Decentralized Fault Management for Service Dependability in Ubiquitous Networks

Grønbæk, Lars Jesper

Publication date:
2010

Document Version
Early version, also known as pre-print

Link to publication from Aalborg University

Citation for published version (APA):
Grønbæk, L. J. (2010). *Decentralized Fault Management for Service Dependability in Ubiquitous Networks*.

Decentralized Fault Management for Service Dependability in Ubiquitous Networks

By Jesper Grønbæk

Dissertation

Presented to the International Doctoral School of Engineering, Science and Medicine, Aalborg University, in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

October 6th 2010

Department of Electronic Systems - NetSec

AALBORG UNIVERSITET

Institute of Electronic Systems, NetSec
Fredrik Bajers Vej 7
Aalborg Øst
Denmark
http://www.es.aau.dk

Title:

Decentralized Fault Management for Service Dependability in Ubiquitous Networks

Supervisors:

Associate Professor Dr. Hans-Peter Schwefel
Aalborg University, Denmark
Forschungszentrum Telekommunikation Wien, Austria

Professor Thomas Skjødeberg Toftegaard
Aarhus School of Engineering, University of Aarhus, Denmark

Jens Kristian Kjærgård, Chief Technology Officer
Tieto IP Solutions, Denmark

Assessment Committee:

Professor William H. Sanders
University of Illinois at Urbana-Champaign, USA

Professor Mohamed Kaâniche
LAAS-CNRS, France

Associate Professor Henrik Schiøler (chairman)
Aalborg University, Denmark

Moderator:

Associate Professor Tatiana Kozlova Madsen
Aalborg University, Denmark

Abstract

In future generation networks (4G and beyond), the convergence of existing and evolved wired and wireless technologies is expected to create an all-IP ubiquitous networking environment. In this environment, emerging end-user services set new requirements for highly reliable and robust operation. However, such properties are challenging to obtain. Faults threatening service continuity are common due to mobility, unreliable wireless links, and heterogeneous networks with varying properties. In addition, the lack of opportunities to deploy network wide mechanisms for synchronized QoS control, network monitoring, diagnosis and repair challenges traditional network fault management approaches. An alternative, studied in this work, is to consider which options an end-node executing the end-user service may have to mitigate faults in the end-to-end paths. Typically operating in the network edge an end-node may fusion information of available networks and their dependability related properties. Such an end-node driven fault management approach aims to make use of the diversity in the ubiquitous networking environment (various technologies, providers and operational characteristics) to provide improved resiliency without explicitly requiring network support. Predominant challenges in this approach are: i) unobservable and incomplete network state information, ii) unreliable observations based on network traffic, and iii) highly dynamic environments calling for adaptation in the fault management process.

In this thesis solutions to mitigate these issues are studied. In the applied methodology focus is on potential gains in the interaction between the well known components of *Observation*, *Diagnosis*, *Decision* and *Remediation Execution*. Initially, a middleware based framework, called ODDR, is proposed. It aims to hide complexities of fault management to the end-user services, while attempting to optimize for their dependability related requirements. In this framework approaches are studied to improve diagnosis robustness to unreliable observations and dynamics by adopting multiple cross-layer observations and decoding measurement uncertainty information in the diagnosis process. A general case scenario study is conducted encompassing diagnosis of faults in infrastructure networks and remediation by proper access network selection for a time constrained end-user service. The scenario is used to obtain insights on the impact of unavoidable diagnosis imperfections on service reliability. In addition, it is studied to what extend good remediation decisions may be applied to mitigate such imperfections. For this purpose a light-weight decision policy evaluation model is proposed. Its focus is on how to parsimoniously represent the diagnosis imperfections considering both simple memory-less diagnosis mechanisms and more complex mechanisms, which correlate observations in time. The model is applied to evaluate best decision policies and settings of the diagnosis component trading off imperfections. Finally, it is assessed how atomic model parts of the studied decision model may be dynamically re-composed to handle dynamic changes of the networking environment.

Evaluations of the proposed approaches have been conducted in the presented models and a system level simulation environment. It is shown how diagnosis based on network traffic can be highly sensitive to even small changes in the observations caused by dynamics in the networking environment and measurement errors. Obtainable gains in diagnosis accuracy and robustness to changes have been demonstrated using multiple cross-layer observations in a basic probabilistic model. Further, a hidden Markov model based diagnosis mechanism has been proposed which is capable of decoding uncertainty information associated to observations to successfully improve diagnosis performance while balancing impact on different performance metrics. While diagnosis may be improved to a certain extent the case study scenario shows the criticality of addressing remaining imperfections in the remediation decision process. Some of our main findings are: i) certain imperfection trade-off settings of the Diagnosis component can lead to worse end-user service reliability than if no fault management was initiated at all, ii) using end-user service state information can help improve service reliability and minimize remediation overhead by ignoring imperfect diagnosis in non-critical states, and iii) imperfections of complex diagnosis mechanisms can be represented in the proposed policy evaluation model, without increasing its state-space, to identify the best diagnosis trade-off setting of diagnosis imperfections optimizing service reliability. Although remediation improvements in some cases are modest under the limitations of studied setup, it is shown how considering the interplay between fault-diagnosis and remediation in the remediation decision problem provides interesting gains. These findings have all been verified by the system level simulations. A final outlook of our approach and results is provided by showing how the decision model may be re-configured dynamically to changes in the networking environment.

Dansk Resumé

Konvergensen af eksisterende og fremtidige netværksteknologier, trådede såvel som trådløse (FTTx, xDSL, cellular, WLAN, DVB, etc.), er forudset til at skabe et allestedsnærværende netværk baseret på IP-teknologi. I dette netværk afvikles distribuerede brugerapplikationer. Nogle af disse forventes at stille særligt høje krav til pålidelighed. At opnå høj pålidelighed i sådanne netværk er en stor udfordring. Kommunikationsfejl opstår jævnligt som følge af dynamik fra trådløse kanaler, mobilitet og i interaktionen mellem heterogene netværkstyper. Dette kompliceres yderligere af begrænsede muligheder for at stille krav til, samt foretage en koordineret styring af Quality of Service, netværk monitorering, diagnose og fejlretning på tværs af forskellige netværk. Således kan det være vanskeligt at benytte traditionelle centraliserede fejlhåndteringsmekanismer. I dette arbejde præsenteres et alternativ med udgangspunkt i hvilke muligheder en end-node, der eksekverer en del af en distribueret applikation, har for at foretage denne fejlhåndtering. Fordelen for end-noden er et holistisk perspektiv på de netværk, der er til rådighed og dermed en mulighed for at samordne informationer om deres funktion samt pålidelighed. En sådan fejlhåndteringsmekanisme eksekveret i end-noden forsøger således at udnytte diversiteten af tilgængelige netværk (forskellige teknologier, udbydere og aktuelle egenskaber) til at forbedre applikationernes pålidelighed uden at stille krav om særlige funktioner i de tilgængelige netværk. Særlige problemstillinger er: i) uobserverbar og ukomplet tilstandsinformation om netværkene, ii) upålidelige observationer baseret på netværkstrafik og iii) et dynamisk scenarie der stiller krav om en fejlhåndteringsproces, der kan tilpasse sig.

Denne afhandling omhandler løsningsforslag til førnævnte problemstillinger med fokus på at høste potentielle gevinster i interaktionen mellem de velkendte funktioner: Observation, Diagnose, Beslutning og Remediering. Disse studeres gennem udviklingen af et framework, kaldet ODDR. Opgaven for ODDR middlewaren er at skjule netværkets kompleksitet for brugerapplikationer samt forsøge at optimere beslutninger i forhold til de krav om pålidelighed en applikation stiller. I framework'et studeres teknikker til at forbedre diagnoserobusthed overfor upålidelige observationer og dynamiske ændringer. Her benyttes observationer fra flere protokollag samt teknikker til at afkode måleusikkerhedsinformation i diagnoseprocessen. Et case scenarie er defineret. Det antager fejl i trådløse forbindelser og infrastrukturnetværk samt remediering gennem skifte af det trådløse opkoblingspunkt og indeholder som applikationseksempel en pålidelig dataoverførsel med tidsgrænse krav. Case scenariet giver mulighed for at studere indflydelsen af ufuldkommen diagnose på service pålidelighed. Yderligere studeres det, i hvilken grad diagnose-ufuldkommenhed kan afhjælpes med hensigtsmæssige beslutninger om remediering. Til dette formål er fremstillet en systemmodel, der kan benyttes til at evaluere forskellige beslutningspolitikker eller diagnosekonfigurationer. Modellen indeholder en simplificeret model af diagnose, der skal repræsentere ufuldkommen diagnose for simple hukommelsesløse diagnose mekanismer samt mere komplekse mekanismer, der kan korrelere observationer over tid. Endeligt ses der på hvilke muligheder der eksisterer for at opsplitte modellen i generiske dele, der autonomt kan sammensættes eftersom netværksscenarierne ændrer sig.

En endelig evaluering af de præsenterede løsninger er udført i systemmodellen samt en detaljeret simuleringsmodel. Det vises hvordan diagnose baseret på netværkstrafik kan være meget følsom overfor selv små ændringer i observationerne som følge af et dynamisk scenarie samt målefejl. Ligeledes demonstreres hvordan diagnose robusthed og præcision kan forbedres gennem en probabilistisk samordning af informationer fra flere protokollag. Endvidere er det vist i et diagnoseprincip baseret på en hidden Markov model hvorledes information om måleusikkerhed kan forbedre diagnoseydeevnen. Samtidig har forringelser en afbalanceret indvirkning på forskellige diagnosemålepunkter. På trods af forbedringer af observations og diagnoseprocessen er det fortsat nødvendigt at adressere resterende diagnose-ufuldkommenheder gennem hensigtsmæssige beslutninger. Hovedkonklusionerne af disse studier er: i) visse diagnosekonfigurationer, der afvejer givne ufuldkommenheder, kan føre til lavere pålidelighed end hvis ingen fejlhåndtering benyttes, ii) brugen af tilstandsinformation fra brugerapplikationen kan anvendes til at forbedre pålideligheden ved at ignorere ufuldstændig diagnoseinformation når applikationen ikke befinder sig i en kritisk tilstand, iii) den udviklede diagnosemodel kan benyttes til at repræsentere selv visse komplekse diagnosemekanismer uden at føre til en forøgelse af tilstandsrummet af systemmodellen. Samtidig tillader modellen en identifikation af den bedste diagnosekonfiguration hvor en acceptabel afvejning af forskellige diagnose uhensigtsmæssigheder opnås med henblik på applikationens pålidelighed. På trods af at visse forbedringer er små, givet begrænsningerne af det studerede case scenarie, vises det klart hvordan fokus på interaktionen mellem fejldiagnose og remediering byder på nogle interessante forbedringer. Alle disse resultater er blevet verificeret i den detaljerede simuleringsmodel. Konklusionerne perspektiveres slutteligt ved at vise hvordan systemmodellen kan konfigureres dynamisk til ændringer i netværksscenariet.

Acknowledgements

First of all, I would like to direct my warmest gratitude to my PhD Supervisor Associate Prof. Hans-Peter Schwefel at Aalborg University and FTW. Hans has been a great inspirator for me to take on the challenges of research since I first met him during my Bachelor studies at Aalborg University. His very positive approach to every matter, excellent abilities keep the main perspectives and large technical insights have been a tremendous support and great motivation for me to conduct this work.

I would further like to thank my industrial supervisor Thomas Skjødeberg Toftegaard, now at Århus University. With his great enthusiasm, a genuine interests in telecommunication research and a PhD background, Thomas has been a highly valuable partner for both technical and non-technical discussions helping to direct my work. Also thanks to all of my colleagues at Tieto IPS and especially Jens Kristian Kjærgård. He has played an important role for me to feel at home in the company and has provided some highly interesting industrial tasks.

Thanks to Prof. Andrea Bondavalli for enabling me to visit his RCL research group at University of Florence (unifi), Italy and for his time to discuss and provide valuable feedback to my work. In addition, a large gratefulness must be expressed to my research colleagues at unifi, Andrea Ceccarelli and Leonardo Montecchi for their past and current efforts to discuss and contribute to the topics of this thesis.

Finally, I must bring my most heartfelt gratitude to my close family which has kept me energized and positive in all phases of my studies. Especially, I must also thank Johanna Quatmann for her great supporting efforts in my finalization of this thesis.

Jesper Grønbæk,

Aalborg University
October 2010

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In future generation networks (4G and beyond) the convergence of existing and evolved wired and wireless technologies (FTTx, xDSL, cellular, WLAN, DVB, etc.) are expected to create an all-IP ubiquitous networking environment. Innovation in products that make use of digital communication ubiquity will inevitably lead to requirements of increased network performance and flexibility (mobility, battery efficiency, ad-hoc connectivity) but also raise requirements for highly dependable operation. The latter may be associated to end-user services with safety critical elements (heart patient monitoring, assisted vehicle control) or in consumer and industrial applications where efficient operation is required to avoid loss of revenue. However, networks are dominated by mobility, varying traffic conditions and communication paths spanning wired and wireless networks across multiple technologies and service providers. As a consequence faults are likely to occur that may threaten end-user service continuity.

The current generations (2G, 2.5G and 3G) of mobile telecommunication networks are relying on a centralized architecture for data switching, monitoring and control. However, upcoming solutions like 3GPP LTE [2] represent a trend where functionality moves from the core to the edge of the network to provide the flexibility (and cost efficiency) already existing in current flat Internet architectures. The cost is increasing complexity of network management and consequently fault management, which becomes a general issue covering the emerging heterogeneous networking platform. In traditional approaches of system designs for handling fault and ensuring highly dependable operation, limited complexity is generally desirable. Applying such approaches in complex heterogeneous and dynamic networking environments is very challenging if not impossible. Instead, alternative approaches must be devised, which have means to improve the overall end-user service dependability.

In this work, we study a decentralized approach for network nodes to perform autonomous network fault management (observation, diagnosis and remediation). We consider the inherent challenges in: 1) utilizing observations from network traffic, 2) performing diagnosis under unreliable observations, 3) making good remediation decisions under imperfect diagnosis and, 4) how a joint view on these processes may lead to end-user service dependability improvements; also in a dynamic networking system context. While the studied methods are general a distinctive focus is made on what we define as *End-node Driven Fault Management*. In this view it is considered what options an end-node itself has to mitigate faults in the end-to-end path. This approach enables to make use of the diversity in the ubiquitous networking environment (various technologies, providers and operational characteristics) to provide improved resiliency and trust of the end-node network services without explicitly requiring network support.

In this chapter, we further motivate this approach by initially considering the current network evolution, existing visions about Next Generation Networking (NGN) and overall challenges. Next, we introduce the generic scenario assumptions used as a pre-requisite for this work and examples are provided in which end-node driven fault management could apply. Further, details on the envisioned end-node driven fault management approach and its challenges are given followed by a problem description, contributions summary and finally, an overview of how this thesis is organized.

1.1 Emergence of Next Generation Networks

The research presented in this thesis represents a branch of the challenges associated with the realization of ubiquitous networking environments (also referred to as NGN, Next Generation Internet (NGI), Future Internet, etc). As such, the assumptions of this work are based on common visions shared by a large part of research organizations, standardization bodies, equipment manufacturers and network operators. In order to establish relevant NGN scenarios and motivate the end-node driven fault management approach, in this section, the background on recognized challenges in NGN is given. A more detailed analysis on these topics is presented in Appendix A.

1.1.1 Current Networking Scenarios

To provide a starting point for considering NGNs, initially, key properties of current publicly available networking systems are highlighted.

Internet networks

A general principle of the Internet is to form a network of networks with no central entity and thus, no single point of failure. This has let to a highly robust [13],[96] and scalable [79] networking infrastructure that today serves more than 550 million hosts [38]. This success is largely based on some of the good technical properties of TCP/IP protocol stack based networks. In summary, some of these are: *Physical and link layer independence*, which has ensured early adaptation of IP networks on existing copper wire infrastructure. It has also lead to easy introduction of evolving and cost-efficient wired and wireless technologies. *A packet switched architecture*, which adds scalability and shared use of network resources for different data streams. *Elementary transport protocols* like TCP for reliable connection-oriented data transport and UDP for connection-less communication. These are widely applicable to support the wide range of application layer services running on the Internet. *Means for QoS, Security and Monitoring* have been limited in the original Internet protocol stack designs. However, several solutions have over time emerged to enhance the TCP/IP suite such as IPsec for packet level security, DiffServ for

QoS management and control and the Simple Network Management Protocol (SNMP) suite for monitoring and management tasks.

With IP-based networking driving the convergence towards next generation networks its weaknesses and limitations become more exposed. In summary, some of the more important issues are: *Mobility* - A classical issue of IP addressing and routing is that devices are expected to remain stationary in a network. This assumption is not valid for emerging high mobility scenarios and numerous solutions have been proposed and implemented ranging from Mobile IP to end-to-end solutions based on Stream Control Transmission Protocol. *Wireless communications* - Originally designed for reliable wired links, flow control protocols like TCP, have assumed congestion as the dominating cause of packet losses. With the increased use of unreliable wireless links in access and ad-hoc networks, the congestion assumption can severely degrade their efficiency in such scenarios. *End-to-end QoS* - Providing a given QoS level in the end-to-end path remains a significant challenge. Several technical solutions exist (e.g. DiffServ, IntServ, MPLS). A set of these techniques can be successfully deployed within network operator controlled domains. However, some operators retain from implementing these for economical reasons. Further, they are difficult to apply in an end-to-end context due to heterogeneous networks with difference in QoS technologies and prioritization policies supported, if any at all. The significant decentralized network control and lack of incentive for infrastructure network providers to implement QoS means for end-users have lead to a widespread conception that end-to-end guarantees can never be made in an IP network context [97]. This aspects calls for alternative mechanisms that can still help supply needed performance and dependability of end-user services despite the lack of network support. This is a strong incentive for the work of this thesis.

Mobile networks

Mobile networks are designed from completely different principles and presumptions than the networks forming the Internet. First of all, individual mobile networks are closed and controlled completely by their operators. These networks have been designed from scratch to support a single service, namely two-way communication between mobile end-user terminals and between a mobile terminal and existing fixed-line phones. From a traditional mobile network perspective this enables a set of key properties: *High Quality of Service* is ensured by reservation of network resources in the links of the mobile network to obtain a consistent QoS throughout a call. *Security* properties have been integrated to protect against malicious users trying to damage the service provisioning or obtain free service. *Centralized control and monitoring* tasks are manageable from the inherently centralized and hierarchic structure of mobile networks. This centralization, however, also makes the mobile networks weaker to withstand failures in the core network components. *Mobility and Roaming* are fundamental properties of most mobile networking systems.

The mobile networks are under complete control of network operators. Thus, changing the network technologies and mechanisms is somewhat easier than in the Internet where multiple parties must agree on changes and implement these. This also means that mobile networks are under a rapid evolution where new technologies are introduced to improve service capabilities and reduce costs of CAPEX (establishment and upgrade) and OPEX (maintenance, network lease and other operation costs). Mainly these developments are driven by a transition to an All IP Network (AIPN) [3], [9]. This allows operators to make advantage of cost efficient IP/Ethernet infrastructure already established for wired broadband connections (xDSL, cable) for mobile networks data. Further, running legacy mobile communication protocols in IP networks enables: i) better utilization of link resources, ii) traffic switching capabilities at the network edge (to offload the core infrastructure) and new products where GSM/UMTS *Femto* radio base stations can be installed by end-users on existing IP-based broadband connections [9].

Other network types

Other highly relevant publicly available networking technologies that are currently becoming available are *ad-hoc networks* [116] and *terrestrial broadcast* networks [44]. The latter are optimized for digital broadcasting services and could play an important role in future networks. An example could be to deliver the same contents to multiple mobile terminals in a limited geographical area such as video streams at a rock concert. This would require significantly fewer wireless resources compared to sending individual data streams to individual users in a typical mobile network setting.

Summary on current networks

In summary, today mobile networks excel in providing a networking solution with control. This leads to inherent QoS and a profitable content delivery platform due to the strong bounds between the network and the end-user services. However, the need for added flexibility to reduce OPEX and CAPEX costs, while handling the fast evolution of mobile technologies, challenges traditional hierarchical networking architectures based on costly hardware. The solution lies partially in the transition to existing IP based network technologies and architectures. At the same time mobile technology advancements are driven by an incentive to boost the demand for IP/data services in mobile platforms. These developments show that a convergence of mobile networks and current Internet based technologies is happening with IP as a unifying element. This realization drives the ongoing research and development in understanding how a new generation of IP based networks can support future demands for dependable and high performance end-user service provisioning.

1.1.2 NGN Challenges

Visions and developments of Next Generation Networks will offer completely new opportunities to create ubiquitous computing environments. Details on various projects and approaches for the NGN visions in commercial environments and research communities can be found in Appendix A. One of the major actors is the ITU-T organization, which has propelled several projects to develop recommendations for NGN. The latest is the NGN-Global Standards Initiative (NGN-GSI). One of the current contributions is an NGN reference model denominated Y.2011 [80], which has many commonalities to other architectures as e.g. proposed in ETSI-tispan [47]. In the following listing we will briefly consider some of the assumed characteristics of NGNs according to ITU-T, which are relevant to, and shared by this work.

- *Decoupling of end-user service provision from network and independence of service-related functions from underlying transport technologies.*
  To maintain flexibility and decoupled development of the end-user services and the network the end-user service should not be tightly coupled to the network technology on which it is running. A central principle in the Y.2011 reference model is a logical split in the protocol stack between service related functions (e.g. end-user services, billing and signalling) and network data transport functionality (not to be confused with layer 4 transport in the OSI stack). Service related functions must be able to operate without having to consider if the transmission medium is wireless/wireline, which technology is used and what available networking resources exist. These considerations must be made separately in the transport services in an attempt to deliver the network performance required by the service-related functions.

- *Support for a wide range of end-user services, applications and mechanisms based on network service building blocks (e.g. to support services of real-time, non-real time, streaming, etc.).*
  These building blocks are generally offered in a middleware layer architecture providing standardized interfaces to perform functions of defining end-user service requirements and actual data transport. Examples of such frameworks are: *the OPEN framework* presented in reference [103], which enables seamless service migration between devices and underlying networks and the *HIDENETS architecture* [37] enabling functions to raise service dependability in ad-hoc and ad-hoc to infrastructure network systems.

- *Unrestricted access by users to different service providers.*
  With the eased couplings between end-user services and the network itself a common vision of NGNs is that users do not buy all their connectivity services from a single network provider. Instead, an end-node device could have access to multiple networks and technologies across operators. This would enable to make use of the diverse network properties depending on the requirements of a particular end-user service.

Obtaining these properties in practice implies numerous challenges. Some are technical and related to the weaknesses of the flexible but somewhat complex IP network architectures. Other challenges are financial where network operators may only be willing to invest in new solutions that can sustain and in best case improve the market opportunities. Finally, political regulation may be introduced to ensure open and competitive networking markets. In this thesis, these latter aspects are outside the scope to consider in more detail; but they will clearly have a significant impact on whether proposed approaches are viable or not.

The focus of this work lies on the issues of improving dependable end-user service provisioning in converging IP networks. The approach is made in compliance with the expected principles in NGNs and the transition of existing network architectures to get there. Thus, an architecture is defined. It attempts to separate end-user services from complex actions, which may be needed in the network to optimize certain dependability parameters for end-user services defining them. The proposed approach, further, recognizes the following aspects: 1) The convergence of existing networks with the IP-layer as a unifying element. 2) That a network node (end-node or in path node) may make use of the diversity of different network technologies, operator networks and network paths to deliver required performance and improve dependability. 3) That there is a large difference in what functionality may be delivered from different available networks in terms of QoS and information about their properties and conditions (load, faults). Still, all networks may deliver useful connectivity in order to provide needed diversity for dependability and necessary performance properties for a given end-user service. 4) Decentralization, where network control is moved from the network core to the network edge (i.e. in base stations and mobile end-nodes) to provide local optimization. 5) That the solutions proposed for improved end-user service dependability must be complementary to existing approaches. And finally 6), that traditional solutions for proven dependability levels in a highly complex networking environment can be difficult, if not, impossible to apply. However, there will still be applications with high dependability requirements and efforts must be made to define feasible alternatives.

Overall, we propose an approach that may be initiated in current networks in parallel to the evolution of networks. This should facilitate options, already attainable in current networking systems, to attempt to control end-user service dependability properties when network support is limited. In the following section an outline is described of our approach in a NGN scenarios context. Further, an introduction is given to the used terminology and exiting principles to improve end-to-end service provisioning in current and NGNs.

1.1.3 Scenario Outline and Dependability Means

To summarize the previous sections, the envisioned NGN networking scenario is presented in Figure 1.1. It represents an end-node operating in a heterogeneous networking environment. The end-node relies on *end-user service providers*, which may be located in either ad-hoc or infrastructure networks to execute *end-user services* such as such as e-mail, telephony and web-browsing. In the remainder of this work, unless other is stated, the term *service* refers to an end-user service.

In this networking system it is envisioned that a set of end-user services define certain requirements for performance (e.g. goodput, delay and jitter) and explicitly for dependability, i.e. *availability:* readiness for correct service, *reliability:* continuity of correct service. Such QoS requirements are a part of an *end-user service specification*. This means that the service cannot live up to its service specification if the QoS requirements cannot be met by the network or end-user service providers. This situation corresponds to a *service failure*. By definition an end-user service failure is a result of one or more *faults* that have not been recovered or masked successfully before impacting the service. Thus, an overall technical aim in next generation networking systems is to provide useful mechanisms to enable highly dependable, i.e. rarely failing, services when required. Detailed definitions of terminology in dependable computing and fault tolerance can be found in [11].

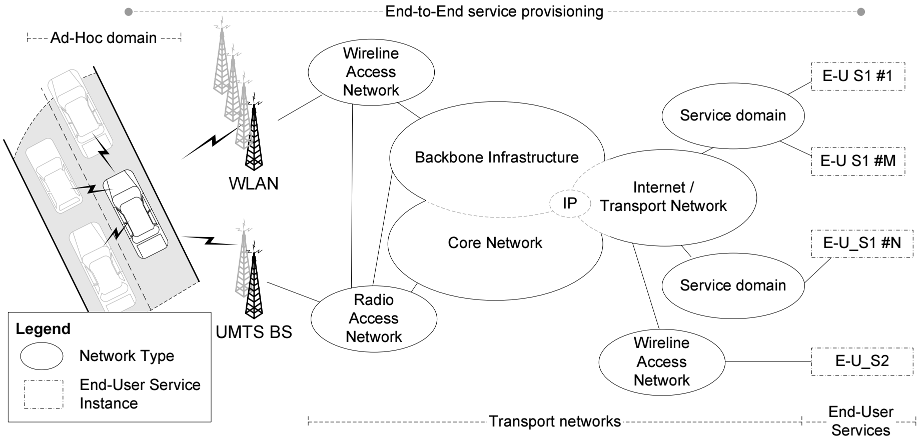

Figure 1.1: *NGN network scenario example with IP as a unifying element.*

Means for dependable end-user services

Obtaining dependable end-user service provisioning is a challenge that is reflected on all parts of the network, the end-user service provisioning and the individual end-node devices. Commonly, these must provide sufficient resources, diversity and redundancy to support overall communication requirements and deliver alternatives when faults occur. From the service provisioning perspective this is typically obtained using redundant access lines and service provisioning replicates placed locally in a data center [58] or other geographic locations [4]. In infrastructure networks efforts are also spend on redundant links and load management in case of faults [42]. Finally, the emergence of ad-hoc networks may provide multiple paths of connectivity if network infrastructure is not present or fails. More elaboration on these issues is provided in Chapter 2.

The approach proposed in this work under end-node driven fault management is seen as an approach to complement these existing techniques. It focuses on which options and end-node itself may have to optimize the dependability properties of end-user services it is a part of executing. The means are to perform observations based on network traffic, diagnose potential faults and decide on the active observation collection and remediation actions needed to optimize relevant parameters as specified in the end-user service specification. The end-node will in this relation try to establish and make use of dependability and performance properties of diverse networks and end-user service providers. An important note in this sense is that an end-node may not have the means to *recover* faults in the networks and at end-user service provisioning ends. Yet, it may have means to *steering clear* of sub-systems affected by them e.g. by selecting another network, changing wireless frequency or another service provider. Thus, in this work the term *remediation* is used as opposed to recovery.

The approach is studied in relation to an end-node middleware fault management framework denominated the *ODDR* referring to it main functions of *Observation*, *Diagnosis*, *Decision* and *Remediation execution*. In the following section we will introduce a set of scenario examples in which the ODDR is envisioned to be applicable. Next, in section 1.2 we present the outline of the ODDR framework and the main challenges associated with this approach.

1.1.4 Cases for End-Node Driven Fault Management

The following three examples are taken from related work in network end-user services imposing certain high dependability requirements. The ODDR is seen to represent an option for implementation and operation of these services.

Car2car and car2infrastructure networks (VANET, Vehicular ad-hoc networks [76])

Vehicles are expected to drive forward the use of wireless communications in mobile ad-hoc networks. Future generations of car electronics may provide information about locations for road pricing or alarm calls to emergency services using vehicle-to-infrastructure (e.g., via UMTS, GSM, WLAN) communication. In addition, inter-vehicle communication may allow exchange of information about road conditions or car control data to enable assisted driving services as platooning [37] or evasive maneuvering. For such services real-time and dependable communication is crucial. The ODDR may attain several roles. An example is the real-time decision on the best communication path between cars: when a car is emergency-braking, it requires to notify its status to cars behind. The reliability and timeliness of the network communication is consequently mandatory: the ODDR module has the ability to diagnose the network state, and decide upon the best end-to-end path (e.g., through UMTS instead of the ad-hoc network) that expectedly is not influenced by the particular fault and provides the needed reliability and capacity.

Industrial use case: Cable Replacement Problem

In many industrial applications cabled solutions are still being preferred to wireless ones, especially in closed networks, due to their consistent performance, security and reliability. Yet, wireless solutions promise to improve flexibility. The problem of maintaining a high reliability and efficiency in wireless communication solutions has been studied in [24]. The work considered software upload and data download for diagnostic purposes in relation to a Driver Machine Interface (DMI) onboard a train. Maintenance personnel may access the DMI via a mobile computer while the train is nearby or in a remote location. The end-user service may simply be an FTP transfer; to maintain efficiency, requirements are set to how long time the file transfer may take. The ODDR may decide upon the needed goodput while minimizing overhead from unnecessary remediation actions. It can benefit from service state knowledge e.g., that a file transfer is nearly complete, and decide not to react to a (potentially falsely) diagnosed fault to avoid jeopardizing the successful transfer by risking a timely and potentially unreliable network fail-over.

Health Care: Emergency Services

Wireless communications are also finding relevant applications in health care. An example is an on-board computer in ambulances, which collects information about a patient during the treatment in the pre-hospital phase [60]. This information must be sent to doctors in the emergency room before the ambulance arrives enabling them to prepare the treatment and initiate it as soon as the patient arrives. During the emergency response and transfer of the patient information via UMTS, the connection may be lost or show degraded performance due to a fault in the provider network. Through exchanging information with other mobile devices in the proximity, and diagnosing the current state of the network, the ODDR knows which local WLANs are most reliable and chooses to probe a few of these to assess their current state and maximize the probability they can convey the information before they get out of reach due to mobility.

1.2 ODDR Framework and Challenges

The ODDR is constituted by a autonomic control loop as depicted in Figure 1.2, which is located in a given end-node. The loop bears strong similarities to the IBM MAPE-K (Monitor, Analyse, Plan, Execute, Knowledge) reference model [75] but is focused at the challenges of end-node driven fault management. Related frameworks are discussed further in chapter 2 while in the following the ODDR components and their interactions are briefly presented. A more detailed overview is further presented in Chapter 3.

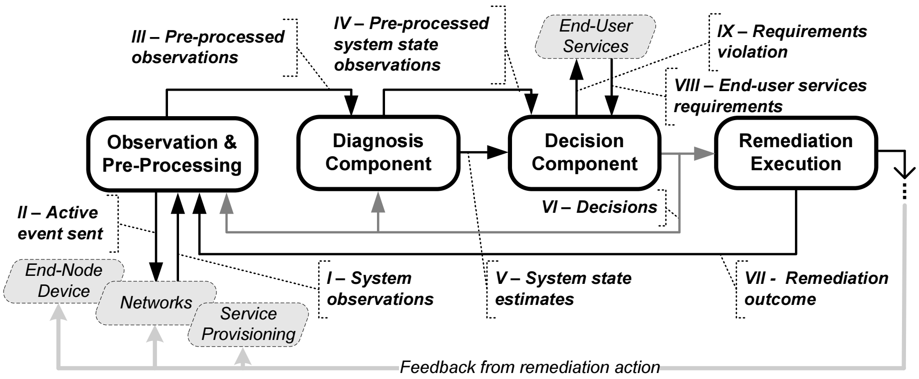
III – Pre-processed observations
IV – Pre-processed system state observations
End-User Services
IX – Requirements violation
VIII – End-user services requirements
Observation & Pre-Processing
Diagnosis Component
Decision Component
Remediation Execution
II – Active event sent
VI – Decisions
End-Node Device
Networks
Service Provisioning
I – System observations
V – System state estimates
VII - Remediation outcome
Feedback from remediation action

Figure 1.2: *Components of the ODDR framework.*

The ODDR is constructed to manage faults in the *system*, which consists of the end-node device itself, networks it uses or may use and the end-user service provider. The main functional components in this process are:

Observation & Pre-Processing (OPP) - The main role of the OPP is to collect observations from the system. This could be done by monitoring the local device variables in the end-node. However, as the end-node may not expect support by monitoring agents located in networks these observations may typically be obtained from existing network traffic (in a *passive* monitoring approach) or self-generated traffic (e.g. probes in an *active* monitoring approach) with the purpose to stimulate or sample parts of the network. As some observations may be of a raw character and/or contain excessive information initial filtering, and pre-processing may be needed in the OPP to only forward relevant information to the other components.

Diagnosis component - The Diagnosis component is essentially the system state estimator. The role of the Diagnosis component is to assess whether the system is operating in a normal state or in a state deviating from normal (fault state) based on pre-processed observations. Diagnosis is implicitly also responsible for separating fault states to the extent needed to provide sufficient remediation actions.

Decision component - The Decision component supervises and leads the execution of the entire ODDR framework. It makes decisions on possible remediation actions. Knowing the end-user services requirements for network communication, the Decision component is in charge of properly configuring the Diagnosis and the OPP components (e.g., setting parameters, changing intensity of active monitoring activities, or modifying accuracy requirements for the diagnosis component), and to identify possible remediation actions. During operation of the end-node, the decision component tracks the states of the system and uses best decision policies to determine when a given action must be executed.

Remediation Execution - This component executes remediation actions initiated by the decision component such as switching to a different wireless channel, selecting a new access network or, in other ways, modifying the communication policies. Remediation actions shall be timely executed and their outcome is monitored in order to obtain information on whether it succeeded and obtain characteristics (remediation time, probability of failing remediation, etc.) for potential adaptation of the ODDR components.

The process flow of between these components starts at the OPP component. Based on observation points in the end-node or different layers of the protocol stack the raw observations are collected (*I*). Events such as probes or stimulation traffic may be generated actively (*II*) to obtain certain observations not available from passive observations based on existing traffic. The OPP emits pre-processed observations (*III*). The Diagnosis Component may use parts of these observation (e.g. mean network delay observations) to estimate the system state (*V*) Directly observable states (e.g. end-user service state, available access networks, etc.) are available to the Decision Component via the Diagnosis Component (*IV*). The decision can be made to initiate several actions (*VI*) to execute remediation or change observation and diagnosis efforts. To initiate service-critical actions (and avoid unnecessary actions) the Decision Component is highly dependent on the end-user service requirements/specification (*VIII*). It may also indicate to the end-user service when requirements cannot be met (*IX*) to allow potential actions in the end-user service itself or by the user (e.g. change video codec or buy access to new access networks). Finally, observations of the remediation outcomes are sent to OPP and forwarded to Diagnosis and Decision components to react.

Challenges in the End-Node driven fault management approach

In this automation framework, the end-node driven fault management approach imposes a strong emphasis on certain challenges in automated fault management. In this section a set of the key challenges are described and motivated.

A) - Unreliable observations - In contrast to many existing fault management approaches the end-node driven approach assumes that the system in which fault management is performed (end-user services and networks) cannot be controlled and instrumented to provide certain observations about their state to the end-node. Instead, the end-node must rely on observations (passive and active) from network traffic. Such observations may, however, be unreliable (e.g. ambiguous, missing, inconsistent [123]), which can lead to a significant impact on diagnosis accuracy and a faulty assessment of available remediation options in the decision mechanism.

Overall, methods in the fault management control loop must be explored to minimize the impact of such unreliabilities and further, good observations must be identified for given diagnosis problems.

B) - Imperfect diagnosis - Diagnosis in the end-to-end path from an end-node is challenging as: i) in many cases observations only provide indirect nondeterministic information regarding the true hidden network state making diagnosis non-trivial, ii) observations may be inherently unreliable. While such unreliabilities are inherent in dynamic networking systems they may also be caused by unreliabilities of instruments performing measurements such as drifting or unsynchronized clocks. Altogether, these issues may lead to imperfect diagnosis where the diagnosis is slow or not fully accurate i.e. causes false alarms or diagnoses the wrong fault.

Challenges are to identify ways of mitigating diagnosis imperfections by improving the decentralized diagnosis process and identify means to deal with unavoidable diagnosis imperfections in the fault management process.

C) - Complex decision problems - The decision process imposes some complex challenges. Overall, the aim is to select a good strategy that may not only optimize the dependability properties of one or more end-user services operating in parallel. It is also a highly relevant property to minimize: *processing*, to conserve energy in battery operated devices and maintain resources for the end-user service itself, and *network traffic resources* to keep perturbation from the fault management at a minimum and avoid faults caused by the management process itself. The latter is particularly important if multiple end-nodes are performing these operations in parallel.

The decision process may execute several actions such as: i) initiating a remediation among multiple options (e.g. change to certain access network), ii) initiate an active observation to improve diagnosis estimates or determine capabilities of a given remediation action (e.g. available bandwidth of an alternative access network), iii) define a required accuracy level of the diagnosis process given available remediation actions, or iv) do nothing as it is expected that diagnosis may become better as more observations are passively collected.

The decision problem itself constitutes a holistic view on the entire fault management process. Thus, several aspects may be included in identifying a good decision strategy like: quantified diagnosis imperfections, level of confidence in diagnosis, the criticality level of the end-user state, available remediation actions and their properties and cost (time, processing and traffic requirements) of decision actions. Finally, the time and order of the initiated actions must be planned, which adds a costly dimension to the decision problem [83]. In summary, the outcome is a complex decision problem, which expectedly require elegantly designed heuristics and/or an efficient modelling approach to solve.

D) - Adaptation to dynamic scenarios - The networking environments are clearly dominated by dynamics from mobility, changing network architectures, varying traffic load, etc. In this sense static approaches to determine the configuration and behavior of the ODDR framework will only be of limited use. The ODDR components must adapt to changes.Providing such adaptive capabilities is a significant challenge well recognized in the networking dependability communities [127]. First of all, changes (deterministically observable as well as hidden) must be reflected in system models affecting the state estimation of the Diagnosis Component and the prediction models of the decision planning in the decision model. Secondly, the system would need to cognitively learn system properties such as dependencies between components [12] (e.g. which fault may lead to end-user service failure and which remediation actions may be applied to mitigate a specific fault) and unobservable system parameters [119] (e.g. fault occurrence rate).

E) - Distributed ODDR The decentralized and distributed architectures of current and next generation networks calls for solutions to solve the challenges in A)-D) in distributed systems. I.e. the ODDR framework could heavily rely on collaboration with other end-nodes e.g. in an ad-hoc network sharing the same environments. This can be beneficial from different perspectives. Initially, an end-node adapting to a given environment may not always need to start from a-priori information, but may utilize knowledge from other nodes with extensive knowledge on operating in a given environment. Next, solving prediction models for best decisions may also be distributed to multiple nodes solving, each, a part of the state space and sharing the solutions. Finally, some awareness of other end-nodes in the system and their expected behavior can be critically important to predict outcomes of remediation actions. E.g. a crash fault in an access network may lead to multiple nodes selecting the same alternative access network causing a contention fault. Clearly, this issue calls for solutions to traditional dependability problems in distributed systems of e.g. security, trust and fairness.

Establishing needed functions of a framework like the ODDR calls for solutions from many research areas including dependability modelling, decision theory, autonomous systems, measurement theory and machine learning just to mention a few. While many of these problems have been studied in an isolated manner for the individual components the overall focus of our view is to consider the ODDR from a holistic approach. By understanding the interplay between the components solutions may be identified making it possible to apply the current state-of-the art for observation, diagnosis and decision making to realize the end-node driven fault management approach.

1.3 Problem Description

Considering the current evolution of publicly available networking systems there is a clear trend that current telecommunication and Internet networks are converging to form a ubiquitous networking environment with IP as a unifying element. Commonly, in new IP based infrastructures innovations are required to maintain a high network stability and performance while providing support for ever increasing performance and dependability requirements of end-user services. High dependability can be difficult to provide as end-user services may operate in end-to-end paths spanning highly heterogeneous wired/wireless networks. Further, a part of these networks may not support specific functions for dependability. Yet, the ubiquity and diversity (different technologies, provider infrastructures and operational characteristics) are from a dependability perspective very attractive to provide different remediation option. This strongly motivates our study of how end-nodes themselves may perform fault management (i.e. observation, diagnosis, decision and remediation execution) in a decentralized manner by addressing and overcoming the main problems associated with this approach.

1.3.1 Problem Statement

The main problems to be studied in this work may be seen from two perspectives: the *decentralized fault management* and *adaptation*. The first refers to the restrictions caused by the decentralized location of the fault management mechanism and lack of network support functions causing unreliable observations and hidden network states. The latter points to the highly dynamic networking environments requiring that means for adaptation to changes must be an inherent part of the fault management framework.

Unreliable observations and hidden network states
Issues of unreliable observations and diagnosis imperfections may be mitigated by improving respectively the observations process (e.g. identifying stronger observations, initiating active probing approaches and filtering data) and the diagnosis process (improve accuracy and promptness). These approaches have been the focus of much existing research work. In this work, we extend such existing approaches by studying the interplay between the fault management components considering: i) for *observation and diagnosis* how to improve diagnosis robustness by using observation uncertainty information and multilayer observations and ii) for *diagnosis and remediation decision*, in relation to given end-user service dependability requirements, how to apply good/optimal decision strategies given quantification of diagnosis imperfections or determine best diagnosis settings considering imperfection trade-offs. This holistic approach encompasses many system parameters and system behavior comprising: networks, fault types, protocols, diagnosis properties and end-user service requirements. Thus, identifying good heuristics for these approaches may be difficult which further motivates model based approaches.

Adaptation to changes
An end-node is envisioned as a mobile wireless device which may be operating in different networking scenarios experiencing various network properties and load conditions. As a result, characteristics of faults, observations, diagnosis performance and impact on the end-user service can change significantly. In addition the end-node must support different end-user service types with varying

dependability and performance requirements. These aspects mean that fault management continually must adapt to the new conditions. This adaptation process must apply to both models used for diagnosis (state estimation models) and models used to determine best decision strategies (prediction models). This implies different challenges which must be addressed. First of all, models need to be lightweight to enable re-planning of strategies as system parameters change while maintaining the expressiveness required to describe essential system behavior. Further, to potentially minimize the model complexity, models may be rebuild autonomously to only cover essential parts of the networking system and avoid state space explosion. The latter requires identification of useful compositions rules and atomic model parts and it is an open question to which extent these can be generalized.

1.3.2 Objectives and Scope of Work

This thesis addresses these main problems through the following objectives:

Establish fault management middleware functions and interactions
A decentralized fault management framework must be specified in order to understand which components are needed in the fault management loop, which functionality they contain and, importantly, which information is conveyed between them. The fault management framework represents the software architecture that may support the implementation of the approaches proposed and assessed in this work.

Improve diagnosis robustness to unreliable observations
Means for performing robust fault diagnosis under unreliable observations must be established. This entails: i) identifying useful passively obtained, low intrusive observations from existing network traffic, ii) using multilayer observations to separate multiple faults and increase robustness to unreliabilities (noisy, ambiguous, missing) in individual observations and iii) specify approaches to increase diagnosis robustness to measurement errors by utilizing observation uncertainty estimates in the diagnosis process.

Study models for assessment of decision strategies under imperfect diagnosis
The interplay between imperfect diagnosis and the remediation decision process is a central issue in this work. Modelling the particular impact on end-user services dependability and cost associated with fault management must help clarify to which extent diagnosis imperfections can be mitigated by good decision policies.

Simplistic model representations should be assessed to ensure lightweight model evaluations. Further, it must be clarified if these models are sufficient to cover the needed policy evaluation cases.

Provide decision model adaptation approach
Based on manually constructed models studied for assessment of best decision strategies it should be considered how such models may be constructed dynamically in response to varying networks and network properties (performance,

dependability and level of knowledge on their properties). This entails defining a framework for construction of prediction models given certain atomic model parts (e.g. generic diagnosis model, generic network model, generic TCP protocol model, etc.). Further, this framework must allow a study on to which degree prediction models should be defined *proactively* or *reactively* considering the state space size and impact on decision policies and end-user service reliability.

1.3.3 Delimitations and Assumptions

In the following assumptions and delimitations made on the scope of this work are presented.

Wireless end-nodes using services in infrastructure networks - We delimit our study to initially focus on a use case of end-user service provisioning in infrastructure networks and service consumption in a wireless end-node. This is a setup that applies to most existing end-user service deployments. Thus, the use of ad-hoc network settings is not considered.

End-node collaboration - The end-node fault management approach could benefit largely from end-nodes collaborating in a distributed manner (see Section 1.2). These topics are, however, not included in the scope of this thesis.

Joint optimization goals - For simplicity, only a single end-user service dependability metric is optimized at a time.

Adaptation and learning - The challenge of adaptation conforms to creating decision model dynamically and similarly learning the real system parameters and dependencies between its components. This problem can, indeed, be considered in this twofold manner. In this thesis, focus is on the model construction while issues of learning are not included.

1.4 Contributions Summary

The contributions in establishing means for end-node driven fault management are summarized in the following. Despite the presented end-node driven fault management context is is expected that these contributions may be generally applicable in other fault management frameworks as well.

ODDR framework
We have defined and detailed an end-node fault management middleware framework denominated the ODDR (Observation, Diagnosis, Decision and Remediation execution). This framework specifies the full fault-management control loop in compliance to generic autonomous frameworks like the IBM MAPE-K reference model [75]. However, its detailed specifications consider functions and interface definitions to support: i) observation with added meta-data (e.g. uncertainty information) ii) a holistic decision component determining components configuration (intensity of monitoring, requirements for diagnosis accuracy) and remediation and monitoring actions to execute, to overall optimize end-user service end-to-end dependability, and iii) an approach to provide dynamic adaptation of decision models.

Increased imperfect diagnosis robustness by multilayer observations

To improve diagnosis robustness to non-adapted changes and diagnosis performance in the networking environment a multilayer observation probabilistic Bayesian Networks (BN) diagnosis model has been proposed. The BN model is derived from basic fault models, TCP behavior and a set of cross-layer observations. To illustrate its diagnosis performance, the BN has been compared to an optimal threshold (OT) approach where observations and network state variables are mapped one-to-one. In comparison to the BN, the OT approach requires less effort to model and parametrize. Even as a basic BN is considered, our results show how utilizing multiple observations in the BN has the potential to improve diagnosis performance. Finally, we evaluate the robustness of the two diagnosis methods toward changed network conditions. These results show how the BN using multiple observations is more robust to changes in network delay compared to the OT.

Increased diagnosis robustness to measurement error by encoding uncertainty in the diagnosis process

Performing measurements in distributed systems of for instance time and packet loss rates can be associated with measurement error due to drifting clocks, clock synchronization uncertainty, host processing and intrusiveness of the measuring system. In this work it has been considered how such sources of observation unreliability can be mitigated. The approach is to quantify diagnosis uncertainty and establish a diagnosis approach that can make use of such uncertainty knowledge to minimize diagnosis imperfections. Three diagnosis components have been proposed based on the Hidden Markov Model formalism: (H0) representing a classical approach, (H1) a static compensation of (H0) to uncertainties and (H2) dynamically adapting diagnosis to uncertainty information obtained for individual observations. Based on uncertainty injection scenarios of unmodelled noise and clock synchronization issues we demonstrate how using uncertainty information can provide a structured approach of improving diagnosis for varying uncertainties.

Imperfect diagnosis policy evaluation model

The potential gains of mitigating diagnosis imperfections by good remediation action decision strategies has been studied in a model based approach. A policy evaluation discrete time Markov chain model has been proposed to capture: i) diagnosis imperfections, ii) remediation capabilities by network fail-over, iii) different potential remediation strategies, iv) cost associated to remediation, and v) time constrained SCTP data-transfer end-user service behavior. Comparison to detailed simulations show that decisions based on the developed Markov model are suitable to maximize end-user service reliability. More generally, we show how restricting fail-over in time can be used to improve reliability under imperfect diagnosis. Our results also show how imperfect diagnosis can lead to cases where performing no remediation is better than initiating remediation at all. Finally, we show how considering the interplay between fault-diagnosis and remediation in the remediation decision problem provides interesting gains.

A parsimonious imperfect diagnosis model capturing imperfections of promptness and accuracy

Extending the contributions of the policy evaluation discrete time Markov chain model a parsimonious Markov model of imperfect diagnosis has been proposed. The model focuses to capture complex behavior of diagnosis approaches which correlate observations over time. it must help identifying in a given end-user service setting which imperfections of promptness and accuracy can be tolerated to provide best reliability.

Capturing complex diagnosis behavior in the model is non-trivial. In our approach, representative diagnosis performance metrics have been defined and their closed-form solutions obtained for the Markov model. These equations enable model parameterization from traces of implemented diagnosis components. From a specific reliability evaluation case study it is shown that the parsimonious diagnosis model sufficiently can be used to identify best trade-offs of imperfect diagnosis performance. From the model-based analysis it is concluded that an over-time diagnosis heuristic proposed in existing work of [25] can improve service reliability and that its configuration will impact the obtainable reliability level. These results are finally shown to be consistent with a similar analysis conducted using extensive simulation-based analysis.

Dynamic model construction approach

To enable decision models to adapt to changes in the networking environment of a give end-node an approach to compose decision models has been proposed. We identify atomic models from the proposed model for policy evaluations. They represent: imperfect diagnosis behavior, remediation network, remediation behavior (delay and probability of failure) and an end-user service. Considering different model composition strategies initial results are provided comparing a proactive and a reactive approach. The results focus on differences in decision policy results, impact on the end-user service reliability and model state space. In an example change case of new access networks, it is shown that the proactive approach despite modelling expectation of changes, provides little improvement over the reactive approach.

List of publications

These contributions are presented in the following publications.

Client-Centric Performance Analysis of a High-Availability Cluster

Grønbæk, J., Schwefel, H.P.; Renier, T.; Frejek, H.P. In Service Availability : 4th International Service Availability Symposium, ISAS 2007, Durham, NH, USA, May 21-22, 2007, Proceedings. Springer Berlin / Heidelberg, 2007. s. 74-93

Safe Wireless Communication Solution for Driver Machine Interface for Train Control Systems

Grønbæk, J., Madsen, T. K., Schwefel, H.P. ICONS 08. Third International Conference on Systems, 2008. Page 208-213

Design and Evaluation of a Safe Driver Machine Interface

Bondavelli, A., Ceccarelli A., Grønbæk, J., Iovino, D., Kárná L., Klapka, S. Madsen, T., Magyar M., Majzik I., Salzo A. International Journal of Performability Engineering, Volume 5, Number 2, pages 153-166, January 2009.

Probabilistic Fault-Diagnosis in Mobile Networks Using Cross-Layer Observations

Nickelsen, A., Grønbæk, J., Renier, T. Schwefel, H.P. In Proceedings of AINA 2009, Bradford, UK, May 26-29, 2009.

Model based Evaluation of Policies for End-Node Driven Fault Recovery

Grønbæk, J., Schwefel, H.P., Toftegaard, T.S. In proceedings of 7th International Workshop on Design of Reliable Communication Networks, 2009, pages 367 - 374.

Assessing the Impact of Imperfect Diagnosis on Service Reliability: A Parsimonious Model Approach

Grønbæk, J., Schwefel, H.P., Kjærgård, J.K. and Toftegaard, T.S. In proceedings of 8th European Dependable Computing Conference (EDCC-8), 2010.

Towards a Framework for Self-Adaptive Reliable Network Services in Highly-Uncertain Environments

Ceccarelli, A., Grønbæk, J., Montecchi, L., Schwefel, H.P., Bondavalli, A. In proceedings of WORNUS 2010.

Improving Robustness of Network Fault Diagnosis to Uncertainty in Observations

Grønbæk, J., Schwefel, H.P., Ceccarelli, A. and Bondavalli A. In proceedings of the 9th IEEE International Symposium on Network Computing and Applications 2010.

Related publications:

Modelling Chain for Throughput Estimation in Wireless Networks

Madsen, T.K., Grønbæk, J., Figueiras, J. and Schwefel, H.P. In Proceedings of the 69th IEEE Vehicular Technology Conference, VTC Spring 20082009, 26-29 April 2009, Barcelona, Spain. IEEE 2009

Cross-Layer Optimization of Multipoint Message Broadcast in MANETs

Nielsen, J., Grønbæk, J., Renier, T., Schwefel, H.P. and Toftegaard, T.S. In Proceedings of WCNC 2009, Budapest, Hungary, April 5-8, 2009. IEEE Computer Society Press.

Planned publications:
Journal paper: *Robust Network Fault Diagnosis to Uncertainty in Observations*.
Conference paper: *Adaptation of Fault Management Decision Models to Highly Dynamic Networking Environments*.

1.5 Thesis Organization

The theses is organized in the following chapters.

Chapter 2 introduces the background on network dependability, existing diagnosis approaches and fault management decisions under uncertainty. Next, a short introduction is given to autonomous frameworks and challenges of adaptation.

Chapter 3 defines in detail the ODDR architecture. Further, an overall scenario is introduced which defines the background for all the studies of this thesis. This includes the definition of an end-user service study case and fault models. Finally, a system level simulation model representation of the scenario is defined along with network state definitions and observation variables.

Chapter 4 introduces to the diagnosis approaches used for the various studies of this thesis. Two groups of diagnosis approaches are studied: one with memory-less properties and one using memory to provide time-correlated diagnosis outcomes (temporal) for improved performance. The different diagnosis components are parametrized to the defined scenario and initial comparisons on their diagnosis characteristics are performed.

Chapter 5 presents a specification and an analysis of the studied end-user service introduced in Chapter 3. This incorporates defining its reliability properties to optimize. The outcome of the analysis is a model, which can be implemented in reliability studies of chapters 7 and 8.

Chapter 6 contains a study of how observation uncertainty may be applied in the diagnosis process. Based on a Hidden Markov Model diagnosis mechanism, different diagnosis variants are proposed to improve diagnosis robustness to measurement errors. A comparison of the variants is finally conducted in a simulation setting. It compares in uncertainty scenarios the HMM based diagnosis approaches with and without the observation uncertainty compensation.

Chapter 7 considers the interactions between diagnosis functions and remediation decisions. The chapter, initially, introduces a joint policy evaluation model consisting of: the end-user service, network state behavior, diagnosis performance and networks. The model is then applied in two studies: i) to identify best decision policy heuristics that can improve end-user service reliability under imperfect diagnosis and ii) to provide a model based assessment of best diagnosis settings (trading off diagnosis imperfections) that lead to improve the service reliability.

Chapter 8 introduces to the topic of adaptation in the ODDR framework. An approach is defined to construct policy evaluation models for changing operating conditions. Next, a model based study is presented on different adaptation model construction approaches. These approaches differ on which assumptions are made on the information of a change and whether the models are constructed in a proactive or reactive fashion. The different approaches are, finally, compared to enable an outlook on how to specify rules for the adaptive model construction.

Chapter 9 contains the thesis conclusion and a view on the open challenges in the end-node driven fault management approach.

Chapter 2

Background on Autonomic Network Fault Management

The end-node driven fault management perspective includes several topics within networking, dependable systems, and autonomous distributed systems. In this chapter, background on the key topics of this thesis work is presented. Initially, focus is on existing means for end-user service dependability in the end-to-end path of communication networks. Next, the ODDR is related to existing work in distributed management frameworks before presenting the main research topics in network fault management. Finally, as the end-node driven fault management approach is related to existing work in network hand-over decision techniques a brief comparison is presented. Note, for each section a brief summary relates the contributions of this thesis to the background topics.

2.1 General Network Dependability

In this section a definition on network faults terminology is presented and existing means for dependable network operation are introduced.

2.1.1 Network Faults Representation

According to the well recognized paper of [11] an activated *fault* is the cause of an *error* (or *deviation*) in the states of the system that may lead to a *service failure*. A service failure is when a service deviates from its service specification. Stated differently, a given fault may lead to service failure if it can cause an error significant enough that the service cannot live up to its specification.

In the following a general fault specification is introduced presenting the fault when it is not activated (*system normal states*) as well as states where it is (*system fault states*). The main aim of this specification is to introduce the fault properties relevant to the following studies. The specification consists of the two main attributes: *fault instance* and *fault impact*. These are defined in the following.

Fault Instance - A fault instance (generally just referred to as a fault in this work) can be described by the properties:

- Fault type
- Fault location
- Fault severity
- Fault process

Fault type describes the event or phenomenon constituting the fault. Examples of such fault types are congestion, contention, crash, radio noise, etc. A given fault instance may occur in different parts of the network which will require different remediation actions. This property is defined as *Fault location*. *Fault severity* defines the level of deviation (error) from the normal operational states. Identified fault severity levels can be mapped to the state space of the fault model representation of the single fault. Severity in the networking environment refers to which extent connectivity, packet loss error, delay and other transmission related parameters are affected. The fault severity levels are controlled by the *fault process* which specifies the dynamics on how a given fault instance may be activated and deactivated (repaired) e.g. using parameters Mean Time Between Failures (MTBF) and Mean Time to Repair (MTTR). In this manner, the fault process controls the nature of a fault i.e. whether it appears permanently, transiently or in an intermittent manner.

Fault Impact - The fault impact determines how a certain fault instance affects a specific end-user services. Thus, to define the full reliability and performance metrics of an end-user service both the *Fault Instance* definitions and its *Fault Impact* must be known. Clearly, the fault impact may be different from service to service; e.g. a TCP stream can be less influenced by jitter than a Voice over IP (VoIP) session.

2.1.2 Existing Means for Dependable Networks

Network faults in the end-to-end path may lead to end-user service failure. Typically, ensuring highly dependent operation despite such faults is addressed in the different parts of the end-to-end path.

Highly Dependable End-User Service Provisioning
In a traditional setting of centralized end-user service provisioning, dependability mechanisms are highly mature. Emerging from service provisioning in PSTNs multiple high availability cluster middleware solutions exist (e.g. Fujitsu RTP4CS [58],[65] and OpenSAF [57]). They integrate solutions for node fail-over, load balancing, fault monitoring and services for end-user services to replicate processes, application state and data among multiple nodes. In recent years focus has been to develop middleware and standardized interfaces enabling cluster functionality for a general range of end-user service types [56]). Also, de-centralized reliable service provisioning platform have emerged such as the open Reliable Server Pooling [43]. It takes advantage of using servers in different network locations to potentially improve both performance and dependability properties. Geographic diversity and service provisioning close to the end-user are principles also applied in the widespread content delivery network provided by the company Akamai today handling a significant part of HTTP traffic in the Internet [4].

Infrastructure networks
Design and deployment of reliability techniques for infrastructure networks is a well studied topic. Much work has been conducted in cost-dependability optimal network architectures [111] and mechanisms for efficient traffic re-direction in case of node failures [42]. More recently, work has emerged in high-level fault management mechanisms (monitoring, diagnosis and recovery) to manage the complex fault scenarios when considering all protocol layers and network services in a modern IP based network architecture [84][12]. Further means in infrastructure networks are to manage all network traffic to ensure sufficient resources for transport of data with high dependability requirements. Multiple means exist for such QoS management as emphasized in the introduction chapter. However, often the general tool applied by network operators is to deploy overprovisioning. I.e. ensuring that there is on average a certain amount of free resources left in dynamically loaded infrastructure links [97]. While this approach does not provide any guarantees on the traffic it has the advantage of simplicity.

Ad-hoc networks
Ad-hoc networks are starting to move from research laboratories into real deployments in applications of e.g. sensor networks [121], flexible Internet connectivity using Mesh networks [116] and vehicular ad-hoc networks VANET [37]. Leveraging high dependability properties in such systems is challenging given high dynamics, unreliable links and often limited processing capabilities. However, the potential high node-count offering collaboration by distribution of tasks and different routes provides means for high redundancy. This has in existing work been used to propose services of distributed reliable data-replication, reliable broadcasting and self-aware clocks [37].

2.1.3 Summary

The end-node driven fault management approach is seen to provide an additional option for improving end-user service dependability properties in the end-to-end link making use of the diversity of difference access networks and potentially end-user service end-points.

2.2 Diagnosis under Unreliable Observations

The background and state of the art in network diagnosis is introduced in this section. Further, background is presented on measurement error and observation uncertainty relevant to the network diagnosis process.

2.2.1 Fault Diagnosis and Detection

System diagnosis is an integral part of fault management to determine if the system is in a faulty state, which fault has occurred and in which part of the system the fault has occurred. In some cases system diagnosis may also focus on determining the nature of a fault, i.e. whether it is *transient*, *permanent* or *intermittent* [39]. In existing literature fault diagnosis covers several focus areas ranging from detection of very basic component failure modes (e.g. up/down) based on basic observation techniques [107] to correlation of a multitude of alarms and observed system metrics [73] (also referred to as event correlation [123]). A detailed view on how diagnosis and detection specifically is defined in the ODDR context is presented in Chapter 4.

In general, performing fault diagnosis in an end-to-end connection can be defined as diagnosis on a black box or gray box through external interfaces. Without explicit network support, alarms and other deterministic fault state information may not be available to the end-node. Instead it must attempt to infer the *hidden* true network state through a set of indirect non-deterministic observations. Performing such diagnosis in complex, heterogeneous and dynamic networks includes several challenges which are highlighted in the following paragraphs.

Unreliable observations and measurements

Observations for diagnosis based on network traffic are inherently unreliable. Observations may be inconsistent, ambiguous, incomplete, missing or delayed [123]. *Inconsistency* may stem from multiple observations providing conflicting information e.g. whether a node is available or not. *Ambiguity* refers to available observations not being rich enough to allow a distinction between relevant faults providing similar observations. *Incompleteness* defines when the obtained observation information is not sufficient to perform diagnosis. Finally, as any traffic in a packet switched network observations may be *missing* due to lost packets or similarly *delayed*.

Many observations will rely on measurements of network related metrics and *measurement errors* may occur. A measurement error is the difference between the measured quantity value and the true value of the considered measurand [20]. It can in general be split in two elements: a *systematic* element and a *random* element [18]. In the distributed networking environment several sources of measurement errors exist. Examples are drifting clocks and low clock resolution [69], clock synchronization uncertainty [22], destination host processing and queuing time (e.g. in case of application based ping) [5],[69] or intrusiveness of the measuring system [22]. In this respect measurement error is another contributing factor to unreliable observations that may affect the diagnosis performance.

Adaptation to dynamics

As a final point network dynamics will also play a significant role in hindering stable and trustworthy diagnosis outcomes. Such dynamics may change the properties of the end-to-end path which may imply: i) new fault instances and observations required or ii) that existing faults and/or observations change characteristics (e.g. higher throughput reduction on a certain fault and a different distribution for a certain observation). Dynamics in the networking scenarios may be attributed to e.g. mobility of other nodes or the end-node itself, load changes (e.g. due to time-of-day), network reconfigurations and sporadic deployment of new access networks (WLAN, femto cells, etc.). Thus, dynamics may be of a slowly or fast changing nature. Such changes calls for both robust and adaptive diagnosis techniques [12], [32].

2.2.2 Overview of Diagnosis Approaches

Performing diagnosis under unreliable observations has been handled using various techniques and with different objectives in existing diagnosis work. For the discrimination of fault nature (transient, intermittent or permanent) under incomplete alarm events (probability of detection below 1) such unreliabilities have been addressed using threshold-based heuristics [25] and probabilistic methods [142]. In existing work of event correlation under unreliable observations some of the previously applied techniques are: *Codebook*, which is a channel coding technique where a signature of alarms is compared to signatures in a database via a most likely match (i.e. some alarms may be missing or false). Codebook approaches have the disadvantage of high configuration complexity and potentially bad performance under ambiguity [123]. *Neural networks* represent another method which is robust to input/observation unreliabilities and have good properties in terms of learning diagnosed system properties although a large amount of training data may be required [139], [123]. A final category considered here are *probabilistic* approaches ranging from *Hidden Markov Models* [142] to Bayesian Networks [73], [124], [117], [104]. Such probabilistic approaches have lately become popular in research on diagnosis as they provide good formalisms for the specification of the diagnosis problem [39], can achieve good accuracy and operate well under uncertainties [124]. The challenge in most cases is that for even small models the computational complexity may be high. Thus, the research motivation is to identify good solution approximations [124] and distributed approaches [125] while maintaining a good accuracy.

More details on the diagnosis approaches studied in this thesis are presented in Chapter 4 in the context of the end-node driven fault management scenario.

2.2.3 State Estimation under Observation Uncertainty

Means to mitigate unreliable observations and improve diagnosis robustness are relevant to explore in the interplay between the observation processes (in ODDR context the Observation & Pre-Processing Component) and the diagnosis process. A particular focus in this thesis is the measurement process where uncertainty estimates may be applied in the diagnosis process. Uncertainty on an observation provides *quantitative* information on the dispersion of the quantity values that could be reasonably attributed to the measurand [18]. Uncertainty has to be included as part of the measurement result and represents an estimate of the degree of knowledge of the measurand. It has to be evaluated according to conventional procedures, and is usually expressed in terms of a *confidence interval*, that is a range of values where the measurand value is likely to fall.

In general, existing work on observation uncertainty applied to the network diagnosis process seems limited. Instead, this topic has been addressed in other similar research fields considering control systems [137], and speech recognition [90], [61].

An important research area in the field of speech recognition is to provide robustness of recognition algorithms to varying noise environments [61]. Commonly, noise robustness techniques have aimed at compensating recognition models to understand noisy speech signals or compensate noisy signals to look like error free signals. In recent years, means of enriching observations with uncertainty information have been studied using ad-hoc and more mathematically founded approaches yielding good results [90]. Gains are that the computational complexity of models using uncertainty is low while maintaining flexibility to varying noise sources. The speech recognition problem is formulated as finding one model among multiple (each representing e.g. a word) that best matches a signal [115]. The defined diagnosis problem differs, as only a single model is defined which must be used to provide the best possible (network) state estimate. Thus, while uncertainty principles are alike it is less clear to what extend the good experiences from speech recognition can be translated to network diagnosis. This provides a key motivation for these studies.

2.2.4 Summary

In the remainder of this thesis the aim is not to propose new accurate diagnosis mechanisms. Rather, it will be considered in the ODDR context how diagnosis robustness to unreliable observations may be improved. In the end-node perspective a comparison between basic thresholding techniques and a multilayer observation based Bayesian network is conducted and a diagnosis mechanism including uncertainty estimates proposed.

2.3 Remediation Decisions under Uncertainty

The backbone of the ODDR is the decision process. In this section a brief view on existing work on joint approaches for fault management in networking environment are provided and a background on principles for decisions under uncertainty is presented.

2.3.1 Joint Observation-Diagnosis-Decision-Remediation

A central part of making an approach as end-node driven fault management viable is to take advantage of the tight coupling and interactions between the functions of observation, diagnosis and remediation actions. To aid manual and autonomous remediation a majority of existing work is focused on improving fault diagnosis accuracy. Interesting gains of combining observation-diagnosis-remediation/recovery is the focus of more recent work. The authors in reference [92] define an approach to plan best sequences of tests (observation collection) and repair actions minimizing time-costs to resolve an end node network connectivity fault. Their focus is on learning such policies, however, under the assumption of perfect tests. In the work of [83] the authors define a central decision mechanism with the objective to plan tests and recovery actions from faults in a service provisioning architecture. They address this decision process under imperfect tests, a probabilistic diagnosis mechanism and a continually calculated decision strategy based on planning a few steps into the future. The authors emphasize how accuracy of diagnosis is only relevant to the extent needed to identify the cheaper recovery action relevant to solve the issue.

2.3.2 Recovery/Remediation Decisions under Uncertainty

Decision making under uncertainty is a general topic which is well studied and an inherent part of probabilistic modelling approaches such as in Bayesian Networks using Decision Graphs [81] and Markov Models using the framework of the Markov Decision Process (MDP) [33]. As previously mentioned Bayesian Networks may offer a strong formalism for the diagnosis task. Its system model may, thus, be shared with the decision process to collect more observations, initiate remediation etc. The Bayesian Network approach is, however, typically limited to modelling static, i.e. non temporal systems. An MDP is based on a Discrete Time Markov Chain (DTMC) and optimal decision policies are derived to minimize a cost function over future system evolutions. As neither of these approaches are applied for decision making in this thesis work, interested readers are referred to the given references for more details. The MDP approach is, however, interesting to derive optimal decision policies in the ODDR framework. As a result the decision problems studies in this work are based on a Discrete Time Markov Chain modelling approach.

DTMCs have successfully been used for performance evaluations of communication systems [55], [51], [16] and dependability evaluation [34]. They provide a useful framework to describe the stochastic evolution of states in complex systems. Assuming system states to satisfy the Markov property only state transition probabilities and initial state probabilities need to be specified to describe system behavior. Some disadvantages of this modelling approach are: The complete state space is a product of all states of system variables. Thus, obtaining solutions can be costly in terms of memory and processing power. Analytical solutions to transient and steady-state evaluations can be difficult to obtain. And finally, not all systems may be well suited for application of the Markov property.

2.3.3 Summary

Many of the challenges in observation collection and remediation policy planning are shared by the end-node approach where existing solutions may apply. In this work additional focus is on decisions to optimize certain reliability metrics of an end-user service model while inherently including imperfection properties of the underlying diagnosis and remediation processes.

2.4 Autonomous Frameworks

The ODDR component falls in to a category of frameworks for autonomous computing. In this section a background on a basic reference model in the research area is provided. Further, issues and existing approaches to enable adaptation and learning in autonomic context are presented.

2.4.1 Autonomic Computing

In the last decade, effort has been made to define a common understanding of autonomous and adaptable systems, with the introduction of Autonomic Computing [74], [127]. The idea of autonomy is inspired by the autonomic nervous system of the human body. It is capable of effectively monitoring, controlling and regulating itself without external intervention. An autonomic system aims to provide such facilities.

The MAPE-K [77] loop depicted in Figure 2.1 has been proposed by IBM as a generic framework for Autonomic Computing. In this framework the *managed*

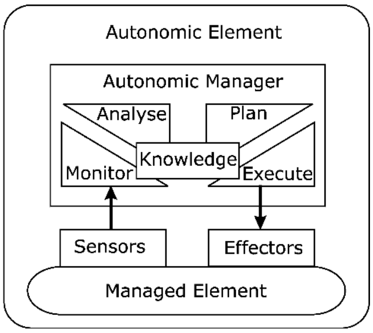

Figure 2.1: *The MAPE-K (Monitor, Analyse, Plan, Execute, Knowledge) reference model for autonomic control loops.*

element represents any hardware or software resource that is given autonomic behavior by coupling it with an *autonomic manager*. *Sensors* collect information about the managed element. Two kind of sensors have been identified; *probes* are system-specific sensors that extract data from a managed element. Probe data is then sent to *gauges*, which may filter, aggregate and process the probes' data before reporting it to higher-level components in the autonomic manager for adaptation planning. *Effectors* carry out changes to the managed element.

The setup is general in the sense that it does not inherently specify the level of autonomicity. I.e. this reference model is seen to cover cases from *level 1. (Basic)* to *level 5. (Autonomic)*. The first level refers to autonomicity with with high human interaction (i.e. as a decision support system). In the latter level the system requires minimal human interaction and can operate and adapt itself to changing operating conditions.

Research in autonomous systems is besides efforts of IBM recognized to have gained momentum from various DARPA programs [74] such as DASADA [136] introducing an architecture for self-managing software and SPS [114] which is designed to sustain dependability of military computing systems to physical and software attacks. This momentum has joined existing and created new research directions in a multitude of areas including the ones introduced in this chapter as well as full frameworks attempting to integrate multiple solutions. According to recent surveys in [74] and [41] current key application areas of full frameworks studies are in wireless sensor networks (power management and self-organization), context management systems and management of complex service provisioning architectures and corporate networks.

Adaptation and Learning

Envisioning the ODDR to operate in an autonomic level, adaptation to new network conditions is a central issue. The problem of creating systems which *adapt* themselves to different environment conditions while learning system properties and dependencies is well known. Belonging to the substantial research area of *artificial intelligence* different techniques have been established in Bayesian reasoning, neural networks, fuzzy logic and dynamic programming. According to the survey presented by the author in [10] adaptation can be split in three overlapping main areas: *Adaptation to a changing environment* - where a system is capable of updating its parameters as a system gradually changes e.g. changes in network load conditions, *Adaptation to a similar setting without explicitly being ported to it* - accounting for cases where the system adapts to a previously unknown environment, and *Adaptation to a new/unknown application* - where adaptation itself must identify its application context, identify potential solutions and apply the most useful.

In general, the definition of the boundaries between *adaptation* and *learning* are not formalized. Depending on the context, the two terms are in literature used interchangeably or together (e.g., *adaptive learning*). From a general perspective, *adaptivity* is often referred to as a characteristic of the system, while *learning* refers to the processes which allow models to evolve and optimize. In this sense, learning is an often useful, but not required, mean to achieve adaptation; for example the term *adaptation through learning* is used in reference [10]. Our main studies in the adaptation approach of the ODDR are relying on adaptation without learning considering an approach to generate models to adapt to varying network scenarios.

Adaptive Modelling

A motivation of studying adaptive modelling approaches is in the ODDR context to ensure that a given ODDR instantiation can adapt to changes in the environment (e.g. fault properties, link load characteristics) and operate in new environments e.g. under end-node mobility actions. The focus of this work is on the latter aspect. Enabling such adaptations while the ODDR is operating is a fundamental problem where some of the main issues are: i) How to bring the system in a safe state where adaptation may occur. ii) How to dynamically construct models as the environment changes, and iii) how and when the adaptation should take place given the current system model, to which extent it matches the system, the current system state and future expectations on change in the system state.

Concerning the issue in i) the authors of reference [48] present a general approach to model and monitor system components to identify states where adaptation may occur or identify actions to drive the system into such a state. The concept is to use and update a Petri Net model to assess such states and actions. Concerning ii), such challenges have been studied in different contexts in the literature. For instance in [46] an architecture for an Intrusion Detection System (IDS) is proposed where, besides a detector, a model generator is employed for learning and generation of new detection models. The authors in [82] deal with the issues in iii) by proposing an adaptation controller exemplified in a multi-server service provisioning infrastructure. To optimize performance and availability metrics, they use an MDP to pre-compute reconfigurations policies to load changes including expectations of future fault events. At fault events, the policies are re-computed to handle the changed resource availability. By this split adaptation approach the authors argue to make the system model computationally tractable while obtaining policies taking future changes into account.

2.4.2 Summary

Adding autonomous capabilities to end-nodes to optimize dependability properties of end-user services independent of network support is a novel approach to facilitate dependable end-user services in future generation networks.

Options of providing adaptation to new operating environments in the ODDR have been explored. We consider the properties of a proactive approach, including expectation on future changes, in terms of the generated policies and reliability metrics. The outcome is compared to reactive cases where no expectation is assumed to simplify the model complexity and consider cases insensitive to non anticipated changes.

2.5 Comparison to Existing Approaches in Handover Mechanisms

Performing hand-overs/fail-overs between various access points and base stations in heterogeneous networks is a dominating part of ensuring end-to-end service continuity. Although the fault management approach sought in this work also considers other remediation options such as wireless channel change, change of service provisioning end-point etc. the hand-overs aspects are central. In the last decade much research has been focused on problems in the domain of *vertical* hand-overs. *Vertical* refers to hand-overs between different access technologies such as WLAN to UMTS to WiMAX whereas *horizontal* hand-overs refer to hand-overs within APs of a given technology. Vertical hand-over/suitable network selection mechanisms are seen as an integral part of future pervasive networking environments where mobile devices can connect to different networks in terms of technology and operators. Existing research has, however, underlined how complex problems of providing good hand-over mechanisms are and that this is still an open research topic [86], [129]. In this work the hand-over problem is seen from a fault management perspective in terms of relating fault diagnosis, remediation options and end-user service requirements to the hand-over decision problem. In this section the aim is to provide an overview on which differences, advantages and disadvantages exist in relation to the proposed fault-centric approach versus existing state-of-the-art (SoA) in vertical handover technology. In Appendix B, a brief overview of existing work in network hand-over approaches is provided. In the following, a set of topics are identified where the fault management perspective proposed in this work has differences compared to the majority of existing work.

Fault-centric vs. Performance centric - The most profound difference between traditional hand-over approaches and the fault-centric view discussed in this work is in the approach to determine when a hand-over should be executed and to which access network. Most existing work attempts to match a set of QoS requirements from a given end-user service to the most suitable network. The decision is typically based on assumed accurate information about available network capabilities. A hand-over is then initiated when QoS requirements cannot be met on the current network or in some cases when a better network is identified. To handle fluctuations in observations, hysteresis and timers are often applied.

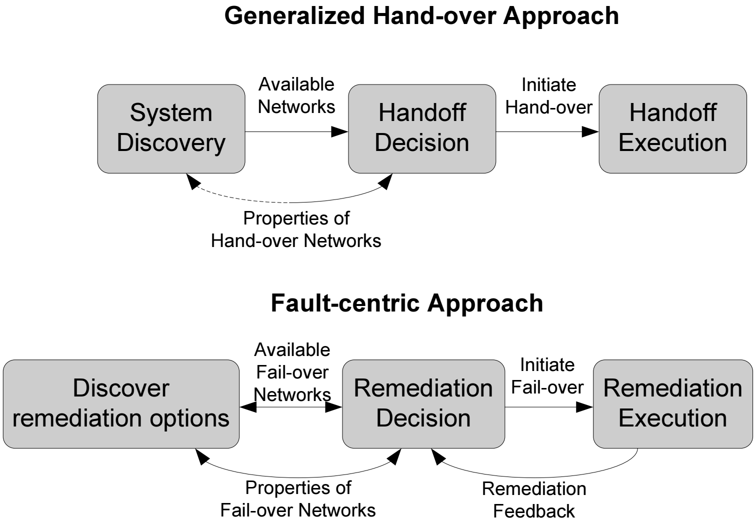

Figure 2.2: *Stages in a generalized hand-over mechanism and the fault-centric approach of this work.*

In typical approaches it is often not considered which fault has occurred but rather that a fault has occurred as indicated by a throughput drop, increase in BER, loss of connection etc. In the proposed approach hand-over decisions are based on end-node driven diagnosis of faults in own and identified networks for remediation. Identifying which fault has occurred provides additional options in the hand-over decision process. Assuming that fault types (congestion, interference, low signal strength) are associated with some fault process (e.g. ON-OFF with exponentially distributed state holding times) identifying a fault can be used to establish its severity on the end-user service and thereby establish if a remediation action is required. Identifying a fault can also be useful to establish the remediation action (available network) which is most likely to mitigate the impact of the fault. The case study of Figure 3.2 is an example of this.

In existing work on hand-over approaches most work considers the aspect of dependability in relation to metrics of BER and PER on particular links (e.g. see [129]). Other important issues of highly dependable service provisioning are availability and reliability of end-to-end connections. For instance an Access Point (AP) may provide good connectivity in terms of bandwidth and delay, however, if it fails and restarts regularly it may not be desirable to use for reliable end-user service provisioning. Clearly, such parameters should be included in the decision process to include performability (dependability versus performance). In the proposed fault-centric approach performability is inherently the central aspect given the inclusion of fault models and diagnosis in the decision process. This will, expectedly, lead to different hand-over decisions increasing dependability of end-user services potentially at the cost of performance (i.e. where lower bandwidth and increased delay must be accepted).

Solve infrastructure network faults - A majority of existing work in horizontal and vertical hand-overs aim at solving first hop wireless link issues. The proposed approach inherently considers hand-overs as a fail-over option to solve infrastructure faults as well.

Unreliable observations and active probing - A topic sparsely addressed in existing hand-over decision mechanism work is how hand-over options (available networks) are discovered and how properties of hand-over options and the current network are determined. Much work in hand-over decision mechanisms assumes that such information can be timely, reliably, and accurately obtained during the system discovery phase. Such reliable information may however be difficult to obtain, especially, when no infrastructure information support exists. As shown in this work, unreliable observations, and thereby diagnosis, may affect the best decision strategy. Thus, hand-over decisions should be based on assumptions of unreliable observations and imperfect diagnosis. The approach will also be extended to consider trade-offs between obtainable diagnosis accuracy and remediation option capabilities in relation to timely requirements. It is assumed that:

- Unreliable diagnosis of the ongoing network connection and remediation options can be improved by spending more time and resources on diagnosis at the cost of delaying the decision.
- Non-complete information about available remediation options can be improved at spending more time and resources on searching for remediation options at the cost of delaying the decision and disturbing the ongoing end-user service. Such disturbances may be unavoidable if only one radio module is available.

A part of including such trade-offs is to include decisions of active probing approaches for information collection in the overall decision process. In hand-over terminology this corresponds to combining tasks of the *System Discovery* and *Handoff Decision* (see Figure 2.2) phases which has not been extensively covered in existing work.

Dynamic QoS requirements - In existing work QoS is often considered in relation to existing traffic class definitions (e.g. see 3GPP specifications [1]) each specifying different required levels for metrics of e.g. RTT, bandwidth and jitter. Such definitions require that application requirements are translated into a QoS class. For instance a data transfer case study considered in Section 3.2.2 would typically belong to the background QoS class. This class aims at a high bandwidth and a low PER. In most work such a prioritization will assign the data transfer end-node to the network with a high bandwidth and low PER. In our work such a hand-over is only performed if the current network cannot live up to the QoS requirements. This is similar to the work in [35] where a lower bound threshold on a given QoS metric must be exceeded before fail-over is initiated. In our work, however, QoS requirements are not defined by static thresholds but rather by dynamic requirements. In the data transfer case the state of the data transfer (amount of data transferred) is related to the time deadline (how much time has elapsed in relation to the deadline). This dynamic approach enables hand-over decisions to be based on an assessment of the end-user service state. For instance if a high throughput has been experienced in the first part of the transfer and much of the data has been transferred, a significant drop in throughput may be irrelevant. The data transfer will end within requirements anyway. In this manner the dynamic QoS requirements can be used to minimize unnecessary hand-overs, minimize overhead and minimize the risk that an unnecessary fail-over will lead to a failure.

Minimize hand-overs for overhead and reliability - In recent work in vertical hand-overs focus is on minimizing unnecessary hand-overs that will not improve the network connectivity significantly. In [128] the authors show how their Markov Decision Process (MDP) based decision algorithm is competitive to existing Multiple Attribute Decision Making (MADM) algorithms while reducing the amount of hand-overs. In [35] the authors try to avoid unnecessary hand-overs from GSM to WLAN to conserve energy. In our work, avoiding unnecessary hand-overs is typically possible in a end-user service like a data transfer where the criticality of the transfer (time elapsed in relation to amount of data transfered) can be used to evaluate whether a hand-over is required. Avoiding unnecessary hand-overs is in our work also in focus to minimize the risk that an end-user service will fail due to a failed hand-over attempt.

Limited or no support from the infrastructure networks - In conjunction with the topic on unreliable observation a large part of existing work assumes that wireless link state, traffic load and network path information can be retrieved from functions in the network. In our work it is assumed that a large part of available networks will not support such functionality. Particularly, WLAN access could be provided by privately maintained access points similarly to the FON concept (see [99]) where QoS control options are not available. Further, in UMTS scenarios the available bandwidth and delay characteristics may not be promised for typical subscribers and access and infrastructure network information will expectedly not be available to the application layers of the end-node software. As a consequence the end-node must use whichever information possible to make the best decisions.

Prior and historical knowledge - Considering metrics of availability and reliability historical knowledge may be very useful to determine how a particular AP, an infrastructure network of a given network provider or an end-user service provider is expected to function in the future. Such information can be imporatant to make the best hand-over decision. Again, immediate observed metrics of a hand-over option with high bandwidth and low delay may be attractive. However, if the particular hand-over option is known to fail with a significant probability it may not be the preferred option. Using prior knowledge of fault occurence rates and repair rates our approach uses such historical information. How such information is obtained at an end-node, updated and made obsolete is a challenge of future work. Some existing work does, however, consider these aspects. In [35] expected mean throughput samples are collected over time to predict future throughput. An approach is proposed to render old samples obsolete to maintain a fresh knowledge of a particular hand-over option. In

[128] it is assumed that knowledge of the network states and state transitions are recorded in the infrastructure network where also new hand-over policies are deduced.

2.5.1 Summary

As seen from the discussed points the end-node driven fault management approach brings a new perspective to the access network selection problem. Not all end-user service functions may need to utilize the complexities of the ODDR. Also, it is not expected that the ODDR will attempt to include all available networks in the decision models consequently, the ODDR functionality is not seen to replace existing hand-over decision techniques, but rather complement these.

Chapter 3

ODDR Framework and Analysis Approach

In this chapter a detailed outline of the ODDR framework architecture is presented. It will establish a reference for the thesis to define where functionalities are located in the fault management control loop and how the individual components interact. Next, a presentation of the networking scenarios to be studied in this work is made considering fault models, remediation options and end-user service assumptions. To enable a framework for evaluating proposed solutions a simulation based analysis methodology is presented. This involves specifying the scenario parameters (system states, link dimensioning, etc.) and observations available in the OPP Component.

3.1 Detailed Outline of the ODDR Framework

In this section the ODDR framework introduced in Section 1.2 is presented in details. Besides clarifying architectural specificities in relation to the MAPE-K framework it is also specified which parts of the framework are addressed in the remainder of this thesis.

3.1.1 The ODDR Components and Modules

The ODDR aims to improve the network resilience for end-user services: depending on the specific service requirements for network communication, the ODDR will operate to maintain the network communication (and consequently the end-user service that depends on the network behavior) within the service requirements. An overall view of the ODDR architecture and of its interfaces is shown in Figure 3.1 and described in detail in what follows. The ODDR framework is a middleware service composed of four different components: the *OPP* (Observation & Pre-Processing), the *Diagnosis*, the *Decision* and the *Remediation Execution* components. Each component is subdivided in a set of optional modules. Different ODDR instantiations may implement a different subset of the ODDR modules, or have different module implementations and settings. Such flexibility of the framework allows to develop different instantiations depending on the available computational and energy resources.

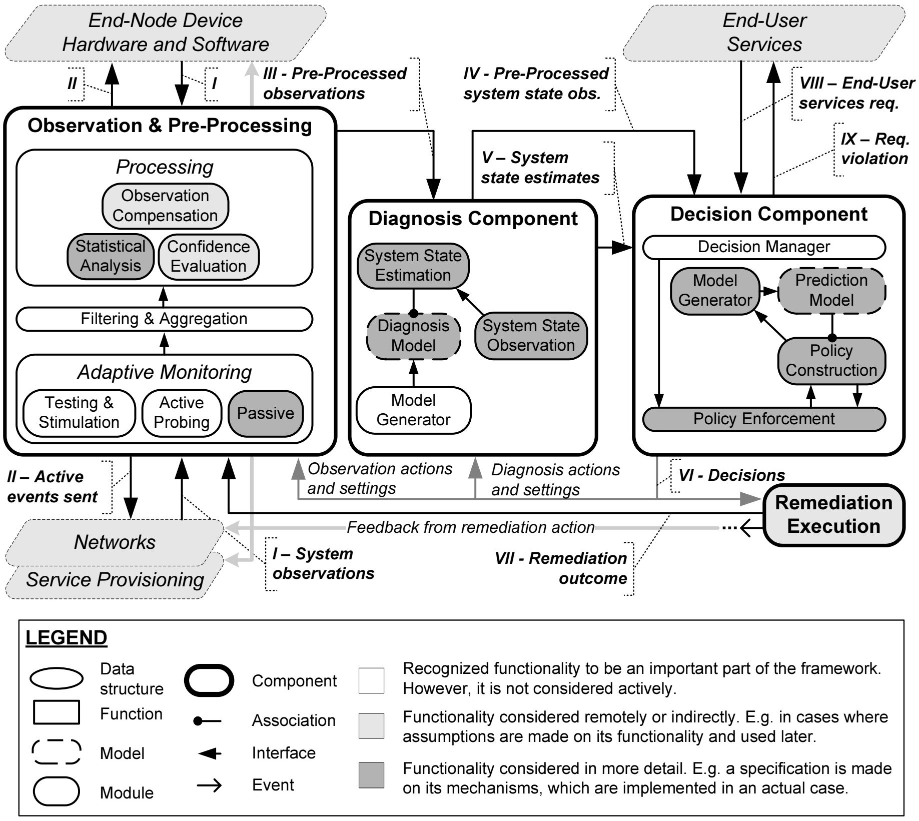
End-Node Device Hardware and Software
End-User Services
II
I
III - Pre-Processed observations
IV - Pre-Processed system state obs.
VIII – End-User services req.
IX – Req. violation
V – System state estimates
Observation & Pre-Processing
Processing
Observation Compensation
Statistical Analysis
Confidence Evaluation
Filtering & Aggregation
Adaptive Monitoring
Testing & Stimulation
Active Probing
Passive
Diagnosis Component
System State Estimation
Diagnosis Model
System State Observation
Model Generator
Decision Component
Decision Manager
Model Generator
Prediction Model
Policy Construction
Policy Enforcement
II – Active events sent
Observation actions and settings
Diagnosis actions and settings
VI - Decisions
Remediation Execution
Feedback from remediation action
Networks
Service Provisioning
I – System observations
VII - Remediation outcome
LEGEND
Data structure
Function
Model
Module
Component
Association
Interface
Event
Recognized functionality to be an important part of the framework. However, it is not considered actively.
Functionality considered remotely or indirectly. E.g. in cases where assumptions are made on its functionality and used later.
Functionality considered in more detail. E.g. a specification is made on its mechanisms, which are implemented in an actual case.

Figure 3.1: *Extended view of the ODDR framework.*

Observation & Pre-Processing

The Observation & Pre-Processing (OPP) component performs network observations (modules grouped under *Adaptive Monitoring*), using: i) *passive monitoring* functions based on existing traffic. Passive monitoring can be configured at run-time (e.g. which observations and sampling rate to use) depending on the information required by the other ODDR components, ii) *active monitoring* functions that can be initiated to collect information not provided sufficiently by passive monitoring, and iii) online testing procedures (module *Testing and Stimulation*), that may allow to accelerate the collection of relevant data about the state of a given network and improve the effectiveness in terms of accuracy and timing for detecting or predicting anomalies.

Observations may be collected from all layers of the protocol stack in the end-node. The output of the monitoring can subsequently be pre-filtered or aggregated (module *filtering & aggregation*) to reduce the forwarding of unnecessary information (in time and space) to the more processing expensive functions in diagnosis and decision components.

It must be noted that the ODDR is the measurement instrument of the ODDR. It represents different active or passive measurement mechanisms (path delay, packet loss rate etc.) to be deployed, configured and enabled depending on the particular end-user services requirements. In general, these mechanisms must be developed with a particular care to collect reliable observations [23], [19]. To achieve this objective, the OPP design and implementation may take advantages of measurement theory basics and well-known practice from monitoring and testing fields. For example, the monitoring and testing functionalities shall be low-intrusive, and computation of the uncertainty in the collected measurement results may help in providing confident and representative results [23]. The latter considerations refer to the remaining three modules: the module *Statistical Analysis*, which provides statistical analysis on the data collected (e.g. mean or standard deviation estimate), and the modules *Observation Compensation* and *Confidence Evaluation*. Observation compensation is the process of removing systematic errors and attempting to provide an estimate of the measured value without measurement error. Considering, however, that the compensation process can never be perfect, the role of confidence evaluation is to estimate the residual uncertainty on a given measurement.

In this thesis the OPP component is studied from a holistic perspective making assumptions on its functions and properties. In particular low-intrusive passive observations are in focus in the following technical chapters. A more detailed description of the assumptions of the observation process are provided in Section 4 regarding observation uncertainties and impact on diagnosis and Section 6 particularly addressing measurement errors and how diagnosis may adapt to these.

Diagnosis Component

The main functionality of the Diagnosis Component is to estimate the actual system state depending on the collected pre-processed observations (*System State Estimation* module). As many states (and particularly fault states) of the network are not directly observable (i.e., hidden), the network state estimation must rely on available observations and on a system model to provide knowledge about the hidden states. Note, in some literature on the diagnosis topic what here is referred to as pre-processed observations is denominated symptoms or alarms. To maintain the generic perspective of the ODDR framework forwarding binary alarms as well as raw data values in this work the term *pre-processed observations* is primarily used.

Diagnosis in the envisioned dynamic system scenarios requires diagnosis models, which can be adapted online. The adaptation may occur in relation to both system changes as well as changes in the requirements of the end-user service and Decision Component. To handle this process the module *Model Generator* has been defined to change both model parameters and potentially its structure.

The *System State Observation* module refers to the process of monitoring and storing the value of certain system variables that can be referred to as being deterministic; i.e. where no uncertainty is associated. Examples of such variables are: end-user service state, available access networks, free system resources etc. The variables may either be discrete or continuous. The module is responsible for making the observations available to the diagnosis process and the Decision Component. It has been located in the Diagnosis Component recognizing it as the overall entity to maintain the current probabilistic and deterministic system state view in the ODDR.

The Diagnosis Component is a central component in the considered work. As emphasized in Chapter 2, diagnosis in networking systems is a highly relevant and intensively studied topic due to integral challenges in performing accurate, timely and adaptive diagnosis on unreliable information. In this work, focus in the Diagnosis Component is on how to apply measurement uncertainty information in the diagnosis process, how to make it robust to changes in observations and, finally, how to characterize the diagnosis imperfections. Issues of how to adapt the *Diagnosis Model* to changing operating conditions are not addressed in this work.

Decision Component

The Decision Component supervises and leads the execution of the ODDR framework. It has the central role of making decisions. In summary, these are: i) *which remediation actions* to initiate, ii) if *active observation* efforts must be initiated, iii) how to reconfigure the OPP and diagnosis components, and iv) to do nothing *waiting* for development in mainly diagnosis, end-user service or network states before initiating any external actions. Taking a starting point in the dependability (and implicitly performance) requirements and active state information of end-user services the decision process has the overall goal to steer the end-node clear of faults that can lead to end-user service failure. In this process it must attempt to minimize: perturbation on the network, use of computational resources and for certain end-node types, the energy consumption. Note, that while all three may be important, the end-node driven fault management approach especially stresses the need to minimize network perturbation. This is due to its centralized nature and inherently, multiple end-nodes attempting to assess network states, which itself could lead to undesirable network fault states.

As mentioned previously, the Decision Component constitutes a holistic view on the entire fault management process. This leads to behavioral characteristics that can optimize end-user service reliability providing advantages, which cannot be provided by individually optimizing the individual parts of the fault management process. A set of key examples are: *Minimize diagnosis effort)* to avoid performing active/passive observations or elaboration of approximate solutions when no gains exists. No gain could be in cases where clearly separating certain fault causes is irrelevant as the remediation action would be the same or where initiating remediation is relatively cheap and resource consuming diagnosis is expensive. *Adapt observation efforts* to diagnosis requirements and only look up remediation options that are relevant to expected faults and end-user service requirements. *Select best diagnosis component setting trading off its different imperfections* related to varying end-user service requirements. Use *characterization of diagnosis imperfections* to select best remediation decision strategies.

Based on the presented aspects to include in the decision process it is clear that multiple sources of *observations* are needed. Most predominant are: estimated system states (network, end-node and end-user service provisioning), available remediation options, end-user service state and time progress. The joint system state view provided by observations must be related to the *system parameters* in order to provide useful decisions. Examples of such parameters are remediation properties (expected delay of application and probability of failure), impact of network states on the end-user service and expected fault/repair process. Finally, additional *cost/reward metrics* may also be needed to capture relevant information not evident by the system state evolution itself (monetary costs, energy costs and good user-perceived quality).

Operating online under varying conditions, the Decision Component needs to adapt its behavior to the environment. In practice, this involves re-computation of the decision strategies to use. However, identifying good decisions in relation to the complex interactions between the managed system states and the actions of the ODDR can be difficult. In this work, model based approaches are studied, which will provide insights into needed aspects to define a good decision strategy and may act as a base to autonomously adapt the strategies. A strategy is defined as a *policy*. A policy is further to be understood as a set of rules defining decision behavior under certain conditions.

Now having established the main role of the Decision Component, a more detailed introduction to the individual modules *Decision Manager*, *Policy Enforcement*, *Policy Construction* and *Model Generator* is given in Section 8.2.

Concerning the Decision Component, in this work, focus is on the interplay between the Decision Component and the diagnostic capabilities. In Chapter 7 it is studied how the impact of diagnosis imperfections may be minimized considering end-user service reliability parameters. The aim is to propose a light-weight model for a given reliability study, which can be used to compare various policy heuristics and may be extended to a dynamic setup given its atomic components. The study is finally extended to consider best settings of complex diagnosis components trading off their imperfections.

Remediation Execution

The execution of remediation actions is performed by the *Remediation Execution* component. Examples of remediations are to switch the end-node association to a different access point (changing access network and potentially end-to-end path in the process), select a different end-user service end-point or change parameters of the communication protocols (frame size, packet size, transmission strength). The component contains the set of rules to initiate a certain remediation action. It has access to interfaces in the protocol stack for execution of these rules and it is responsible for monitoring the remediation process. This monitoring must establish the outcome of remediation (successful or failed) and monitor further performance parameters such as time to effectuate remediation. All monitored information is provided to the OPP component, which further forwards the information to diagnosis and the decision components. The functionality of this component is considered implicitly in the following chapters.

3.1.2 The ODDR Interfaces

In this section the ODDR external and inter-component interfaces are described in detail.

***I* - System observations**

This is the primary interface for the collection of observations to the OPP component. It is a generalized abstraction of interfaces towards monitoring modules in all layers of the protocol stack, operating system counters and potentially monitoring frameworks like SNMP [31]. The observations may be of a raw nature such as network delay measurements, radio signal strength information or already processed information like explicit fault notifications from other monitoring subsystems like health monitors.

***II* - Active events sent**
The active monitoring modules of this component use this interface to send active events. An active event can be stimulation traffic to exert a certain network path, a test request to an end-user service to ensure it provides the expected answers or active probes (e.g. ICMP ping) to perform measurements of path delay and potential bandwidth. The interface generalizes different interfaces towards monitoring modules in all layers of the protocol stack, end-node system drivers and potentially external testing middleware functionality.

***III* - Pre-Processed observations**
Pre-processed observations made by the OPP are conveyed in this interface. Different meta-data may be associated to the observations such as a time-stamp and uncertainty information. The observations may both be of a stochastic nature (e.g. network delay and radio-signal strength measurements) or deterministic (e.g. available access networks and protocol being used).

***IV* - Pre-Processed system state observations**
This interface is similar to interface *III* and in principle provides access to the same information. In practice, however, a sub-set of the observations in *III* needs to be processed into state-estimates before being useful to the Decision Component (through interface *V*). Direct state observations are maintained in the Diagnosis Component but made directly available to the Decision Component such as available remediation actions (e.g. access networks and their properties, end-user service end-points and utilization of frequency bands) and end-user service state information (e.g. progress, criticality level and user experienced waiting times).

***V* - System state estimates**
Interface *V* provides system state estimates for the Decision Component. Each state estimate may have meta-data associated such as confidence in the state estimate or information on which observations or how much time may be needed to improve the estimate for a given amount.

***VI* - Decisions**
The decision interface is composed of several interfaces towards the components *Diagnosis*, *OPP* and *Remediation Execution*. The interfaces are grouped in two lanes conveying *actions* and *settings re-configuration commands*. Actions are in summary: i) initiation of a specific remediation procedure (Remediation Execution Component), ii) actively collect observations (OPP), and iii) elaborate approximate diagnosis (Diagnosis). In addition, the Decision Component issue reconfigurations depending on which end-user service is active. Such re-configurations could be which observations to make, the frequency of observation collection, which observation filtering level to apply, how accurate diagnosis needs to be etc.

***VII* - Remediation outcome**
The Remediation component communicates to the OPP component the outcome of the remediation action. The remediation outcome is provided in order to obtain information for potential adaptation of the ODDR components (including new decisions in case the remediation procedure fails).

***VIII* - End-User services requirements**
The end-user services require certain dependability properties to be maintained by the OPP. These requirements are communicated through this interface. The abstraction level of the requirements may be *high* such as transfer X amount of data reliably within Y seconds, or, *low* e.g. by inclusion of a complete application state profile and restrictions on time or performance in different states.

***IX* - Requirements violation**
The Decision Component operates to satisfy the end-user service requirements. When they cannot be met, the Decision Component may communicate this information to the end-user service, which may then have the option to react. E.g. by increasing its safety margins or informing the user to perform certain manual remediation actions (e.g. insert cable with broadband wired connection). Alternatively, such information may be collected in the application layer to quantify the general QoS of a certain end-user service.

3.2 Scenario, Faults and Remediation Options

Having defined the ODDR and its motivation in this section a derived scenario to be used in the remainder of this work is presented. The scenario must provide a viable framework to study the challenges of end-node driven fault management. This involves the definition of faults, remediation options and an end-user service use case.

3.2.1 Generalized End-to-End Scenario

Based on the general scenario in Figure 1.1 a detailed scenario case has been derived. This case scenario is presented in Figure 3.2. It depicts an end-to-end connection as it will appear in a generalized wireless end-node to infrastructure network setting. In the pervasive networking environment the end-node may have the option to connect to different wireless access networks using private or commercial access points (WLAN), mobile networks (UMTS, LTE) or use a connection via an ad-hoc network. The infrastructure consists of a radio access network and a backbone operated by a given network operator. From the operator network there is a path to the service provisioning infrastructure. It may be established through some arbitrary network architectures (Internet transport) depending on, amongst others, where the end-user service provisioning end-point is situated. Finally, it is considered that an end-user service may have different provisioning end-points located in different networks to deliver the same service.

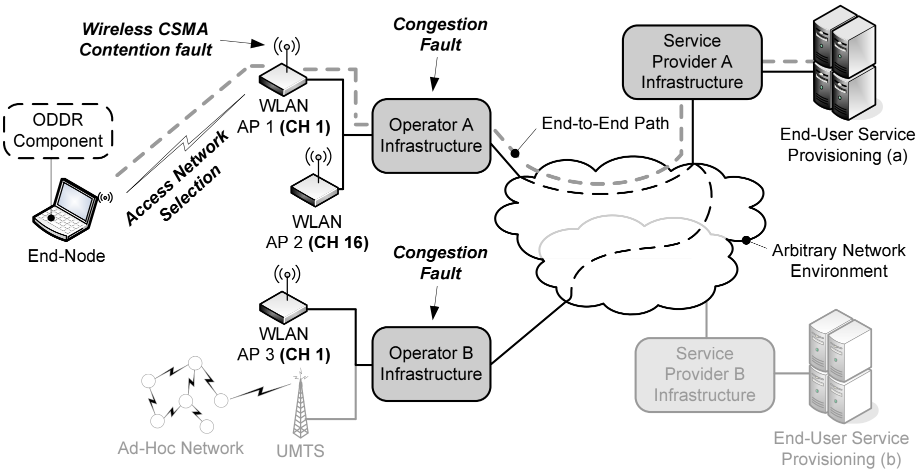
Wireless CSMA Contention fault
Congestion Fault
Service Provider A Infrastructure
ODDR Component
WLAN AP 1 (CH 1)
Operator A Infrastructure
End-to-End Path
End-User Service Provisioning (a)
Access Network Selection
End-Node
WLAN AP 2 (CH 16)
Congestion Fault
Arbitrary Network Environment
WLAN AP 3 (CH 1)
Operator B Infrastructure
Service Provider B Infrastructure
Ad-Hoc Network
UMTS
End-User Service Provisioning (b)

Figure 3.2: *End-to-end scenario with fault cases and remediation by access network selection. Grayed elements are not considered actively in this work.*

Faults and remediation options

Anywhere in the end-to-end path a fault may occur and potentially lead to an end-user service failure. Such faults may be handled in the network part in which it has occurred. However, this is not sufficient to ensure end-user service continuity. Some faults may not be timely recovered such as replacing or repairing a failed component. Examples are private wireless access points that may take hours to weeks to repair or crashed DSL connection that may take some minutes to restart or days to be repaired by a technician. Other faults may not be actively recoverable due to sparse recovery options. An example is high contention in the wireless spectrum caused by multiple asynchronous nodes. Finally, what may be considered as a fault in the end-user service fault model may not be considered a fault in the network part where it occurred and thus, no recovery actions are initiated. In this case an example may be a high router load leading to some level of packet losses and increased delays (congestion). Given the diverse network access options and reconfigurations of the protocol stack parameters the end-node may have several options to remediate such faults. Examples of remediation options could be to: adapt link layer parameters such as reducing the frame size and increasing the transmission strength given contention at the access point is diagnosed, or change end-user service end-point to change parts of the end-to-end path or finally, select another access network. The latter option is particularly interesting considering that the diversity in access network configurations and technologies can help to remediate different faults. An example is depicted in Figure 3.2 for the faults of wireless *contention* and infrastructure network *congestion*. Starting with emphasizing the diagnosis challenges the end-node must identify whether the cause of a reduced throughput is caused by one of the two faults. None of the faults are directly observable as the contention happens at the access point (the end-node may not experience a high contention level itself) and the congestion in the infrastructure. Providing a useful diagnosis can have a significant influence on making the best remediation decision. E.g. assuming the end-node is connected to *Access Point 1* ($AP\,1$) failing over to $AP\,2$ is only useful if the fault is contention in the WLAN channel 1 $CH\,1$. On the other hand, a congestion fault in *Operator A Infrastructure* would require a fail-over to $AP\,3$.

Scenario delimitation

In the remainder of this thesis focus is on a subset of the presented scenario in Figure 3.2 while the grayed parts are not studied in detail. In practice the subset involves the two introduced fault types. Further, a single end-user service provisioning end-point is considered and the remediation option studied is *access network selection*. Despite its simplicity, this scenario is rich enough to support the studies of: 1) how to improve diagnosis robustness to unreliable observations, 2) study various remediation decision strategies and models for these, and 3) discuss decision adaptation approaches. Finally, it is expected that the obtained results may in future work be extended to a larger set of multi-fault and multi-remediation option scenarios.

A more detailed definition of the faults is presented in Table 3.1. Actual parameterizations (i.e. fault process and states) of the faults are introduced as needed in the following chapters.

3.2.2 End-User Service Case Study

A single end-user service case is introduced as a mean to specify a useful end-user service reliability metric. This metric will be applied to assess reliability impacts of imperfect diagnosis and remediation actions.

Examples of end-user service cases to consider could be: a) upload of patient examination results to the emergency room during an emergency response. b) HTTP/TCP based streaming of video/audio (with progressive download) where a certain amount of data must be transferred in a timely manner to avoid buffer underrun. c) File down/upload in time before connectivity is lost or within time constraints to ensure a certain productivity in industrial applications. Such end-user service cases are interesting as the success criteria can be formulated as a deadline in contrast to a minimum throughput requirement. This makes it possible to explore best fail-over strategies (optimizing reliability and minimizing overhead) based on past and predicted progress of the end-user service.

The chosen case is motivated by industrial grade reliability requirements for wireless communication as discussed in the work of reference [24]. It considers a reliable data transfer operation. A certain amount of data must be transferred within a critical time deadline requirement. The reliability parameter considered is, thus, the probability of a successful transfer within the deadline (later referred to as Ω). In accordance to this, the studied end-user service is defined as *time constrained reliable data transfer*. This generic end-user service case covers various end-user services based on Transmission Control Protocol (TCP)/Stream Control Transmission Protocol (SCTP), where the influence of likely network faults may lead to service failure. From a general perspective both TCP and SCTP are considered in the following sections. In cases where one of the reliable transport protocols is used over another this will be mentioned explicitly.

Congestion Fault	
Description	During operation, an end-to-end path may experience different levels of congestion on the links traversed in the network. This may happen from varying cross traffic levels at the congested links or varying paths taken by the packets constituting the end-to-end link. In this respect congestion itself is an expected event and may occur in different severity levels. A *congestion fault* is, thus, in this work defined as a certain severity level that can lead to end-user service failure.
End-node effect	Increased levels of congestion lead to increased round-trip times, packet losses and jitter.
Example causes	On a network component level the fault chain may be initiated by a crashing and restarting router or an unstable link. In both cases traffic over this router or link is rerouted to paths in the end-to-end link leading to a severe congestion level.

Contention Fault	
Description	Contention in this work refers to wireless contention. Multiple wireless nodes are working without a centralized media control scheme. Instead they compete about the network access using Carrier Sense Multiple Access (CSMA) [135]. This media access control scheme is common in open wireless network technologies like IEEE 802.11 (WLAN). Next, assuming that the contention occurs mainly at the access point (due to other nodes using it or other nodes communicating with each other in the same spectrum (channel), the goodput may be severely damaged. The reason is an increased amount of frame collisions (two or more frames sent at the same time causing some to be lost) and the access point waiting for medium access.
End-node effect	Contention at the remote end-point (the access point) is experienced at the end-node as frame losses and eventually packet losses. The end-node may, obviously, also observe some of the contending nodes, which leads to an increased channel access delay (as it waits for the channel to become clear) and consequently increased round-trip time.
Example causes	Crash fault of another access point causing multiple nodes to use the contended until a repair takes place. An alternative cause could be a large group of wireless nodes moving into the area where the access point is operating leading to contention until they move out again.

Table 3.1: *Detailed definitions of the faults considered that may lead to end-user service failure.*

3.3 Analysis Methodology and Setup

A majority of the results produced in this thesis are made in the context of the network scenario presented in Section 3.2. A detailed simulation based approach is introduced. It implements the network scenario to provide observation and parametrization data (network obs. etc.) and an environment for verification and assessment of analytical results. In this section the outline of the simulation setup is introduced encompassing its assumptions, its functions and a baseline parametrization. This parametrization involves defining network states and their impact on the reliable data transfer end-user service. It must be noted that the simulation model structure presented is general to the remaining part of the thesis. However, the parametrization may vary in some of the Chapters. In cases, where such deviations from the model presented here exist, they are explicitly emphasized.

Further details on the simulation verification and setup can be found in Appendix C while Appendix E provides an overview of the common parameters.

3.3.1 Background on Simulation Analysis

To implement the network scenario in a reference setup, a simulation based approach has been chosen. A simulation based approach offers some clear advantages over real experimental test-beds or true end-to-end paths in that it mainly offers: i) stability, full controllability and repeatability, and ii) less experiment setup and execution time and lower cost compared to practical experiments. The main disadvantages are that the proposed simulation environment may be over-simplifying important aspects compared to a real networking environment such as cross-traffic patterns, time-varying changes in the network setup characteristics of observation distributions etc. In the individual studies of this thesis discussion on such simplifying assumptions will be made to put these issues into perspective. In future work, the proposed approaches must be extended to larger experimental Internet test-beds like GENI [63], PlanetLab [113] or experimental test-bed scenarios of ALARP [68], HIDENETS [37] or SAFEDMI [24].

To ensure a realistic simulation environment a system level simulation approach is pursued. System level simulation refers to a realistic implementation of the Internet Protocol stack protocol behavior. This generally includes implementation of layer 4 transport protocols like TCP, SCTP, UDP, RTP etc. down to, and including, layer 1 in the wireless links considering wireless fading, frame losses and media access control. The implementation of the evaluation framework is based on the de facto standard network simulator *ns-2*. This simulator is highly regarded in research communities. It is dominant in simulation based analysis of communication methods in fixed infrastructure and ad-hoc scenarios. Being an open source tool with many different contributing parties ns-2 has been thoroughly revised and partially verified to deliver realistic network simulation results [52]. The network simulator used is in this work is ns-2 version 2.29.

3.3.2 Introduction to Simulation Model

The simulation model of the networking scenario in Figure 3.2 is depicted in Figure 3.3. Initially, its parts are introduced and in the following sections the

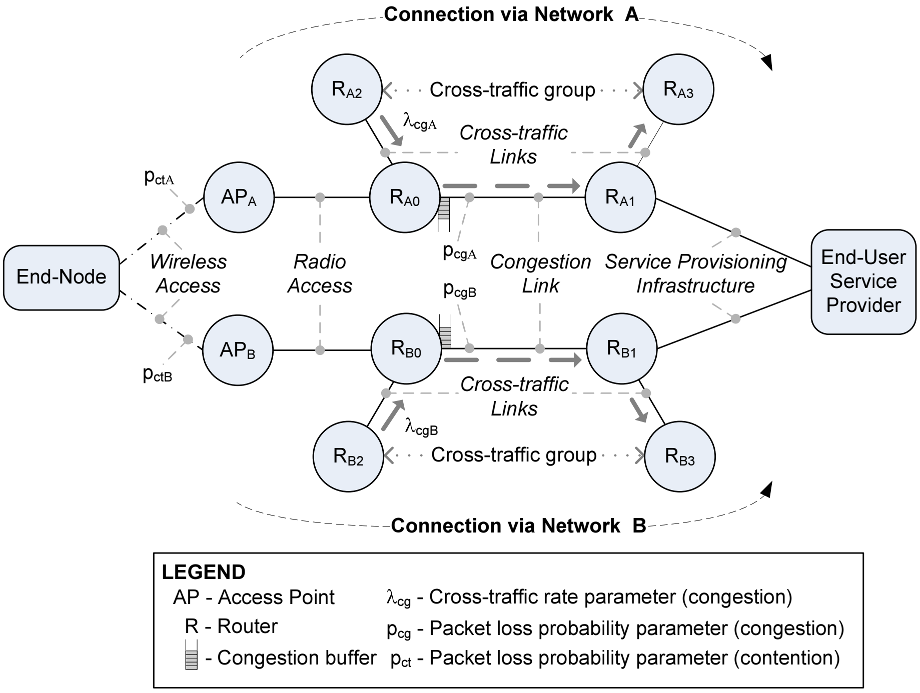

Figure 3.3: *Outline of the simulation model for the studied end-to-end scenario.*

capabilities and limitations of the used simulation model are discussed along with baseline parametrization.

Simulation Model Structure

The simulation model consists of altogether eight infrastructure nodes and two access point nodes in two distinct network paths with an identical network architecture. These paths are denominated *Network A* and *Network B* respectively, referring to two operator infrastructures A and B offering wireless access and independent network paths. The end-node can operate in either of the networks by directing traffic via access point A (AP_A) or AP_B. To describe the model a starting point is taken in network A. From the wireless link, connectivity is offered to the network infrastructure via the *Radio Access Links* representing e.g. a private xDSL connection or cellular network radio access infrastructure. The *Congestion Link* between the nodes R_{A0} and R_{A1} represents the infrastructure network link. It may be seen as a representation of multiple links in the independent end-to-end path resulting from using a certain network provider. Based on cross-traffic generated from R_{A2} to R_{A3} the load on the link from R_{A0} to R_{A1} can be controlled to define various congestion levels. Thus, the buffer in which congestion occurs is in R_{A0} toward R_{A1}. From the potentially congested infrastructure network the *Service Provisioning Infrastructure* is attached representing an independent path to the given End-User Service Provider.

3.3.3 Simulation Model Properties

In the following paragraphs functions and assumptions to realize the studied end-to-end scenario in the similation model are presented. It must be noted that the following descriptions apply to both network A and B, which are initially considered to have equal properties.

Fail-over remediation option and fail-over properties
The remediation option considered is a hand-over (fail-over) from network A to B. I.e. network A is considered as the *initiation* network and network B is the remediation network. General assumptions are: 1) the data upload can only be active on one link at a time, 2) a data-upload session can be resumed after a fail-over, and 3) a fail-over may fail. Failing fail-over must simulate the aspect that a fail-over may fail due to a fault in the connection establishment procedure with the link to which a fail-over is attempted. In the simulation this is simulated by changing access network from network A to network B and disabling the wireless link to AP_B for $200\,ms$ before falling back to access network A. A *link down* duration has been set for $200\,ms$ to correspond to some layer 2 delay in failing over. In practice, this value depends heavily on the wireless equipment and device driver, used authentication/security schemes, link conditions and hand-over implementation. So while $200\,ms$ or less may be achievable [100] in practice this is a highly variable value that may be a factor 2-3 higher. Including the layer 4 behavior, i.e. the stochastic congestion avoidance behavior of SCTP, the mean return delay has been identified to $1.2\,s$ based on simulation results. Details on the obtained results are available in Appendix C.2.

SCTP layer 3 mobility
To simplify issues of layer 3 mobility in this work, SCTP in a multi-homing configuration is used. This has in existing work [95] been considered a potential end-to-end layer 3 mobility solution in hand-over scenarios as an alternative to Mobile IP (MIP)[112]. In the multi-homing setup both the end-node and the service provider has two end-points (with individual IP-addresses). SCTP may now use either of the networks to transfer the data by activating one of the multi-homing paths. As a result an ongoing data transfer session can with insignificant interruption be moved from one network path to another [95]. To disregard complexities of SCTP connection establishment it is assumed that it is always successful and can be conducted with no delay implications.

SCTP configuration
The transmission rate of SCTP is controlled by congestion avoidance mechanisms very similar to TCP. This means that results obtainable from this simulation model may be comparable to a setup based on TCP (e.g. in an MIPv4/6 setting). This is useful for the modelling approach described in Section 7.1 as a starting point can be made in previously well studied TCP modelling techniques. There are, however, some important differences between TCP and SCTP congestion avoidance mechanisms. Most significant differences are pointed out in [29] and [7]. To address some of these differences SCTP has, in the studied simulation setup, been (re-)configured to approach TCP SACK congestion control behavior. Details on these reconfigurations are presented in Section 7.1.

Network state control
The *congestion fault* related network states of the *Congestion Link* can in simulation be controlled either by cross-traffic (for different rates of $\lambda_{cgA/cgB}$) or a setting of an independent loss rate for each link ($p_{cgA/cgB}$). The first option is most realistic in relation to the congestion fault model where losses are correlated due to loss events from queue overflows (assuming drop-tail queues). The latter option ensures independent losses, which may ease later modelling challenges but also be too simplifying. The impact of the two approaches is discussed later in Chapter 7 where a complete model of the scenario is introduced to derive good decision policies. A similar discussion could be made for the *contention fault* in the wireless link considering simulation of contending nodes versus modelling contention as a wireless frame loss probability. Yet, for simplification a fixed frame loss probability is assumed ($p_{ctA/ctB}$) in line with the existing well regarded IEEE 802.11 contention modelling approach in the work of [17].

ODDR capabilities
The simulation model, finally, also implements basic ODDR functionality to support cases where the simulation outcome is determined by ODDR behavior. Besides fail-over execution, such functions are observation pre-processing, diagnosis functions and decision policies for the decision component. Details on which of these functionalities are used are emphasized in the chapters making use of the simulation model.

In the following section a set of network states is defined describing network conditions in normal and fault states. The intermediate results obtained in this process will be used as a background for a model of the end-user service as a function of the network state, which is introduced in Chapter 5.

3.3.4 Network State Definitions

To complete the diagnosis model in this section a realization of the networks states is made. This involves parameterizing the model links, defining fault and normal states and assessing their impact on the reliable data transfer end-user service case. Primary focus is made on the congestion fault case, which is used as reference for a majority of the presented studies.

Fault severity and fault process
In the scenario two likely *fault types* have been introduced and the importance of pin-pointing their *location* in the end-to-end path has been exemplified. Focusing on the fault severity and fault process as defined in Chapter 2, these properties may be used to create a multi-state model that accurately captures the behavior of a certain fault. I.e. a fault may alternate between different severity levels each of which could lead to different decision actions in the remediation framework. In this work, however, a basic two-state fault model is used as a starting point. It allows to represent a *normal state* and a *fault state*. Both the normal and fault state will represent a certain severity level. By definition the normal state *cannot* lead to end-user service failure while the fault state *can* lead to failure. Thus, the fault state may represent a quantified level of

Link Name	**Bandwidth**	**Delay**
Wireless Access	$5\,Mbit/s$	$4\,ms$
Radio Access	$10\,Mbit/s$	$4\,ms$
Congestion Link	$15\,Mbit/s$	$10\,ms$
Service Provisioning Infrastructure	$10\,Mbit/s$	$5\,ms$
Cross-traffic Links	$20\,Mbit/s$	$10\,ms$

Table 3.2: *Applied link properties to the simulation model of Figure 3.3 on page 46.*

degradation/error (e.g. a given throughput degradation) or a hard fault like a crash fault. A common and convenient assumption in fault modelling is that state holding times can be described by an exponential distribution. Convenience stems from the independence assumption that time left in a state does not depend on time already spent in the state. This allows for simple and analytically tractable model representations e.g. in a Markov model [33]. Depending on the type of fault and fault analysis sensitivity to fault model error, richer representation of fault behavior may be needed and obtained using Matrix Exponential (ME) approaches [91]. The ME modelling approach offers the advantage of modelling arbitrary state holding times at the imminent cost of an increase in the state space.

For now, the basic fault model representation can be summarized as an ON-OFF model (ON representing the fault state, OFF the normal) with exponentially distributed state holding times. This fault model may be parametrized to specify both *intermittent*, *periodic* and approximations of *permanent* faults in cases where the expected fault period is significantly larger than the end-user service period.

From the literature of modelling end-to-end links as well as wireless access links Markov models have often been found to provide powerful model representations despite their simplicity. For instance, the authors in [85] show how 802.15.4-based wireless links behavior may be split in a long term component (e.g. link state level) and a fast component (packet level). They propose a model that captures these two levels and shows good similarities to experimental measurements when considering packet level statistics. In the work of [119] the authors attempt to train Hidden Markov Models of various state sizes to network traces based on loss traces. For a diverse set of various Internet end-to-end traces they find that at most four states may be needed to describe the connection state while in a majority of cases only a single or two states would be sufficient. While these models and their results are not tied to network state (fault) causes, they exemplify how simple two-state Markov models, as applied in this work, may be sufficient to describe end-to-end paths properties.

Link dimensioning

The network state definitions are made based on an instantiation of the simulation model of Figure 3.3. The links have been dimensioned to the values of Table 3.2 using the following considerations. Starting from the end-node the *wireless access* links between it and the $AP_{A/B}$ node are represented by wired links in the simulation model. This is due to implementation-wise limitations of SCTP in ns-2. However, the link bandwidth and mean delay parameters have been tuned to correspond to an IEEE 802.11b 11 Mbit/s link under ideal link conditions. As the wired links are full duplex aspects such as layer 2 contention between SCTP data-packets in the uplink and acknowledgements in the downlink are not included in this simulation model. While more recent versions than IEEE 802.11b offer more than $100\,Mbit/s$ [21] (with a theoretical option of moving to $600\,Mbit/s$) the considerations of this work are general and could be applied to various setups and configurations. Moving on to the *Radio Access* link it is assumed that it offers high bandwidth as it may be based on xDSL or a fiber connection. The same is assumed for the *service provisioning infrastructure* bandwidth capacity. To represent a high bandwidth infrastructure the *congestion link* has the highest bandwidth of links in the end-to-end path but also the highest delay to represent multiple intermediate links and nodes. Finally, to realize the congestion issue, *cross-traffic links* have been dimensioned with a higher bandwidth than the *congestion link*.

Congestion network states

To implement the congestion fault model the simulation setup must implement two states: *normal* and *congested* for the *independent losses* setting and *cross-traffic* based setting.

Initially, a normal state has been defined for independent losses specified by a low packet loss probability of $p_{cgA/cgB} = 0.005\,\%$. The impact on data transfer completion times in this state is depicted in Figure 3.4 for 2000 independent simulation runs for a medium-sized data transfer of $data_{size} = 10\,MB$. This data amount is defined from the requirements of the industrial application, which has inspired the studied end-user case study [24] and will be used as a reference throughout this work. Over the simulation runs there is a significant variability in the data transfer completion time distribution, which ranges from $18.6\,s$ to $28.4\,s$. For comparison a data tranfer under perfect conditions ($p_{cgA/cgB} = 0.0\,\%$) has a duration of $16.8\,s$. Clearly, even for the low packet loss probability in a normal state the congestion control mechanisms for SCTP (and similarly TCP) have a significant influence on the data transfer completion times.

A fault state has been defined to provide approximately a halving of the mean throughput under independent losses. The result is a fault state definition where $p_{cgA/cgB} = 0.022\,\%$. The result is depicted in Figure 3.5. As expected, the variability and range of data transfer completion times increases significantly when considering the fault state. It can be observed that there is no overlap from the distribution of the normal state, which is in the range $38.9\,s$ to $59.1\,s$.

The end result is two highly distinguishable network states. Introducing a fault process to alter between the two states will theoretically lead to data transfers in the range of $18.6\,s$ to $59.1\,s$. It should further be noted that the considered mean packet loss probabilities are not rare in an Internet path contexts as several measurement based studies have shown for both low hop count links [40] considered in the order of $0 - 1\%$ and in longer Internet based paths [110],[138] in the order of $0 - 10\%$.

A summary of the state definitions is given in Table 3.3. End-to-end path metrics have been obtained from the simulation which have a dominant impact on the end-user service and that are interesting to understand how the congestion network states may be observed and diagnosed. The upper half depicts

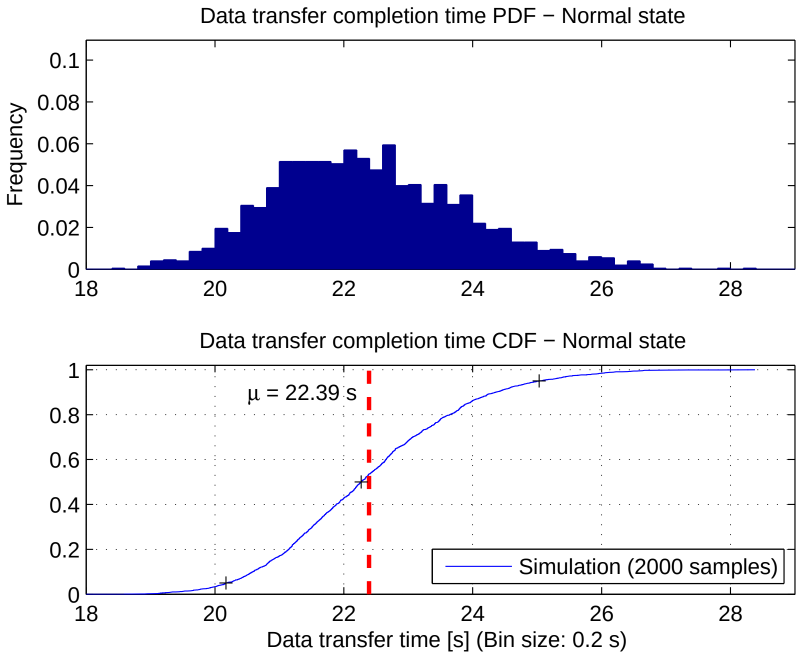

Figure 3.4: *Data transfer completion time distribution for a normal state where* $p_{cgA/cgB} = 0.005\,\%$

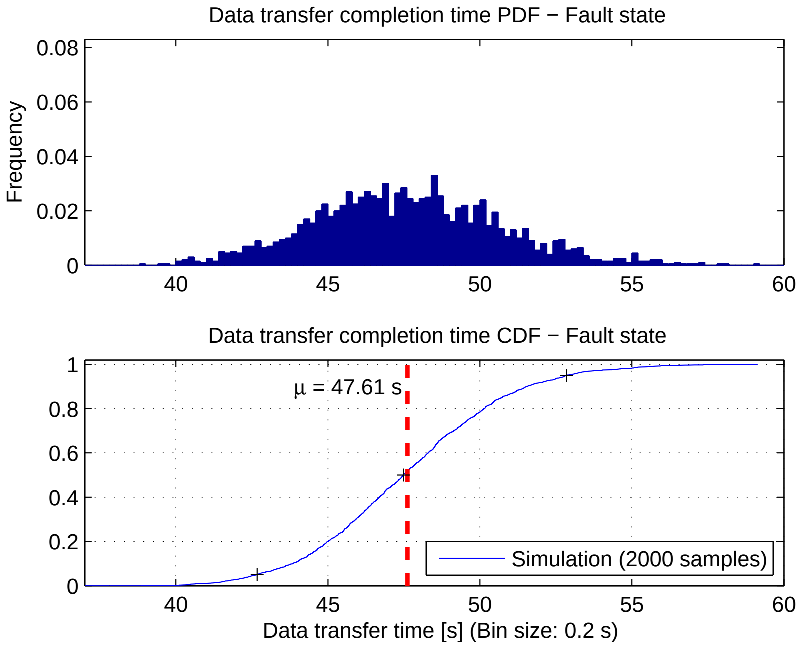

Figure 3.5: *Data transfer completion time distribution for a fault state where* $p_{cgA/cgB} = 0.022\,\%$

results for normal and fault states using independent losses. $\hat{PER}$ is the estimated Packet Error Ratio (PER) obtained from simulation. It describes the ratio between packets sent in the particular state over packets observed as lost. In this respect it just verifies the specified packet loss probabilities to be correct. $\hat{RTT}$ is the mean Round-Trip Time (RTT) estimate made from observations of SCTP acknowledgments at the transport layer. $d\hat{RTT}$ is also an RTT mean estimate. It has, however, been obtained at the link layer by inspecting the payload of the frames to obtain sequence numbers of SCTP. Now registering sending and receiving times of the individual SCTP segments at layer 2, an RTT observation metric can be obtained without including the local end-node network layer to link layer buffer. Eliminating the local buffer provides a better estimate of RTT variations caused by network congestion conditions as local link bottleneck behavior is not included. This is beneficial for the diagnosis component as will be discussed later. It should be noted that for the observed transport and link layer round-trip times the estimates have been made as a mean of means of all raw RTT observations in a trace. That there is a difference between $\hat{RTT}$ and $d\hat{RTT}$ is clear. The local buffer has a significant contribution to the overall RTT estimate. Notice, that $\hat{RTT}$ is lower when a fault has occurred in the independent loss case. This is due to the fact that the data upload transmission rate is decreased in the fault state leading to less queueing in the local queue. Finally, in Table 3.3 the 5, 50 and 95 % quantiles are stated for the data transfer distributions.

Table 3.3 also includes results in the case where fault and normal states are based on cross-traffic. Rates of $\lambda_{cgA/cgB}$ have empirically been adjusted to roughly provide data transfer completion time ranges similar to the independent loss cases. Each cross-traffic event starts an FTP file transfer (using TCP) with file sizes according to a Pareto distribution ($\mu = 10\,KB,\ \beta = 1.5$ [8]). Stability has been ensured in the amount of parallel active transfers (see also Appendix C). Considering $\hat{PER}$, it is significantly higher than in the case of independent losses while producing similar data transfer completion time ranges. The reason is that many of the losses are correlated due to the drop-tail queue packet loss behavior. As SCTP transmission rate reduction efforts are determined based on whether losses have occurred within a transmission window, as opposed to evaluating the individual packet losses, each loss has

					Percentiles		
State	$\lambda_{cgA/cgB}$	$\hat{PER}$, σ	$\hat{RTT}$	$d\hat{RTT}$	5%	50%	95%
Independent losses							
Normal	-	0.5, 0.09	59.1	51.7	20.2	22.3	25.0
Fault	-	2.2, 0.17	53.3	52.5	42.7	47.5	52.9
Cross-traffic - Pareto distributed file sizes							
Normal	86.96	0.8, 0.23	72.1	54.3	18.3	20.4	23.8
Fault	123.46	2.9, 0.44	58.5	57.1	34.3	41.3	50.0
Units	$[conn/s.]$	%	$[ms]$	$[ms]$	$[s]$	$[s]$	$[s]$

Table 3.3: *Network state definitions for state simulation by independent losses and cross-traffic (Each state is based on 2000 independent simulation runs of a data amount $data_{size} = 10\,Mbyte$).*

less impact when it is correlated to other losses. As a result the outcome is higher packet loss ratios for network states when losses are based on cross-traffic compared to true independent losses.

Contention network states
The contention network normal and fault states can be defined in the same manner as the independent losses congestion network states. The aim is, thus, to define a fault state that also leads to provide an approximate halving of the mean throughput by a packet loss probability similar to the one obtained in the congestion state. More details on defining a set of contention network states is provided in Section 4.4 where an approach for multi-fault diagnosis using observations from multiple layers is discussed.

3.3.5 OPP Observations

A final focus is on which observations to make available for the OPP component and how these may be pre-processed. A set of observations has been selected which: 1) can be obtained from existing network traffic to maintain a non-intrusive observation approach with no overhead and 2) enables to diagnose and distinguish between the two faults of contention and congestion despite unreliable observations. The set consists of the observations of *Round-Trip Time* (RTT), *Packet Retransmission Ratio* (PRR) and *Frame Retransmission Ratio* (FRR). The resoning behind their selection and a definition is provided in the following paragraphs.

Round-Trip Time - An indicator for network congestion is round-trip time. The congestion fault is caused by a general increase in the queue length in the infrastructure network router buffers. This leads to increased waiting time for packets in the end-to-end link and consequently the following interpretation: a high RTT observation corresponds to the network being in a *fault* state and a low RTT to the network being in a *normal* state (assuming a reasonable stationarity of the link properties not associated to the fault process).

o_l^{RTT} refers to an observation of RTT, where l is a discrete time step $(1 \leq l \leq L)$ and $O^{RTT} = \{o_1^{RTT}, o_l^{RTT}, \ldots, o_L^{RTT}\}$. Each o_l^{RTT} is obtained by collecting RTT samples readily available from an SCTP based upstream data-stream in a fixed size observation window of the size ω_{RTT} $[s]$. A new window is made for every o_l^{RTT} from the sample time and ω_{RTT} $[s]$ in the past. Each window may consist of $0 \ldots V$ individual RTT samples where V is a random variable depending on the SCTP transmission rate control and its interaction with the network dynamics. This step is a part of the observation pre-processing where relevant statistics are drawn from the raw RTT observations. Being interested in observing the expected value of RTT, for this observation a mean statistic is calculated to obtain:

$$o_l^{RTT} = \begin{cases} \frac{1}{V} \sum_{i=1}^{V} rtt_i & \text{if } V > 0 \\ missing & \text{if } V = 0 \end{cases} \tag{3.1}$$

where rtt_i is the i'th RTT sample obtained in the window of size ω_{RTT} $[s]$ at the discrete time step l. Note, that if the window is empty the observation is defined to be *missing*.

It must be noted that RTT observations are obtained as previously described by recording the end-to-end path RTT at layer 2. This prevents the RTT from becoming affected by local layer 3 to layer 2 queuing effects, which may distort relevant information about the end-to-end path delay and, hence, the congestion level state. Also, it must be emphasized that the RTT observation used to observe the network congestion level, in the simulation scenario, clearly only is valid when states are generated based on cross-traffic. In Table 3.3 this is seen from the $d\hat{RTT}$ estimate where the increased load on the bottleneck buffer leads to a higher mean RTT. Notice, that $\hat{RTT}$ (the layer 3 estimate) actually drops in the fault state due to less buffering in the local link layer buffer due to the drop in transmission rate. This shows why the link layer based RTT observation is needed.

Frame Retransmission ratio - Losses of frames at layer 2 can provide information on contention (also in the case of hidden nodes) based on lack of layer 2 acknowledgments (e.g. see section 9 in the IEEE 802.11 standard [78] as the receiver node (the access point) may not have been able to receive the transmission due to a collision. A retransmission is the indication that the MAC protocol has considered a frame to be lost. To quantify the amount of frame losses this observation variable is defined as a ratio of frames, which are marked as re-transmissions in relation to frames sent. The use of ratios rather than rates is due to the fact that the frame transmission rate is dynamic. The dynamics stem from potential variation in the used data link transmission rate as well as the rate of packet coming from the network layer to transmit. This makes a potential re-transmission rate highly sensitive to the used rate. The observation variable Frame Retransmission ratio is thus obtained as:

$$o_l^{FRR} = \begin{cases} \frac{\Sigma retransmissions}{\Sigma transmissions} & \text{if } \Sigma transmissions > 0 \\ missing & \text{if } \Sigma transmissions = 0 \end{cases} \tag{3.2}$$

where $\Sigma transmissions$ refer to all transmissions including retransmissions and first time transmissions in a window of size $\omega_{FRR}\,[s]$. $\omega_{FRR}\,[s]$ is defined in the same manner as the $\omega_{RTT}\,[s]$ window.

Packet Retransmission ratio - The Retransmission ratio PRR observation ($\omega_{PRR}\,[s]$) is defined similarly to FRR in the sense that packets that are retransmitted by TCP/SCTP are considered in relation to all packet transmissions in a window $\omega_{PRR}\,[s]$. Again, it must be noted that a retransmission by TCP/SCTP is considered as packet judged to be lost by TCP. PRR provides information on whether the end-node is affected by any fault. As the considered faults of contention and congestion per definition will affect the end-node end-user service, packet losses can provide some information. However, packet losses themselves are ambiguous as both of the faults may lead to a similar ratio of packet losses. Thus, all three observations considered in this section may be considered in conjunction to diagnose and separate the contention and congestion faults.

Additional observations which would also provide some information on the network states are: throughput, amount of visible contending nodes and packetpair probe time-gaps [131]. However, the focus is not to identify the best observation options available but rather to handle unreliabilities of available observations. Thus, for the delimitation of this work these three observations RTT, PRR and FRR are in focus.

Example: normal and congestion state RTT observations
As mentioned in the section introduction the congestion fault case is used as a reference for more of the subsequent studies. Thus, concluding this section is an example of the RTT observation as a function of the network state of normal and congested. Round-trip time observations obtained from an SCTP stream in a single end-to-end link (e.g. *Network A*) of the simulation model are presented in Figure 3.6. It contains two large sample sets of observations made in the normal and fault state, respectively, to create the empirical distributions of $P(o_l^{RTT}|Ns = normal)$ and $P(o_l^{RTT}|Ns = fault)$. Ns refers to the network state. The states have been obtained using cross-traffic levels according to the state parameters of table 3.3. In the presented case a window size of $\omega_{RTT} = 0.3\,s$ is considered. This window size provides a reasonable trade-off that can provide initial smoothing of the RTT observations without having a strong impact on the achievable reaction time when it later is used in the context of diagnosis [104].

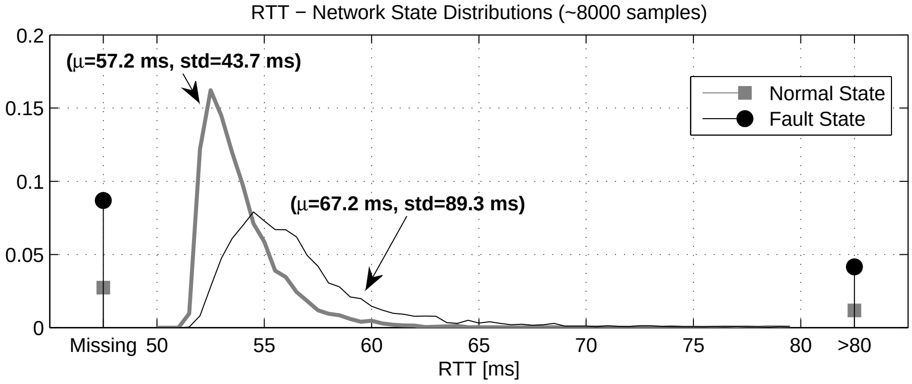

Figure 3.6: *Observation distribution of RTTs for normal and fault network states.*

In the normal state, as expected, most probability mass is located at the lowest obtainable round-trip times. As the fault state is entered the distribution shifts towards longer round-trip times and a long tail. It must here be noted that very large round-trip times can be caused by effects of SCTP (and similarly for TCP) where duplicate acknowledgments delay an acknowledgment of a certain sequence number. As these delays are caused by packet losses, which again are more likely in a fault state, these observation samples contain some useful information on the network state as well. Further, from the distributions it is clear that the two states are not deterministically separable as expected. Finally, it can be observed that the amount of missing observations is significant and that more observations are missing in the fault state. This is the case, as there due to TCP flow control are more periods without transmission when many packet losses have occurred.

Chapter 4

Fault Diagnosis under Unreliable Observations

This chapter is focused on techniques for network fault diagnosis. The aim is to initially present the view on diagnosis as made in this work. This includes defining a basic set of diagnosis metrics, which will be used to classify diagnosis performance in the remainder of the work. Subsequently, a set of diagnosis mechanisms are introduced representing the most basic threshold based mechanisms to more complex probabilistic approaches. The aim is not to present an extensive overview on the vast amount of diagnosis approaches in the literature, but rather to present, specify and parametrize a selection of relevant ones that will be referred to in the remaining chapters. It is further discussed how the presented approaches may handle observation unreliabilities. To enable this discussion, initial experimental diagnosis performance results and comparisons are performed. The insights gained will form a basis for the studies presented in Chapter 6 on improving diagnosis under measurement error and in Chapter 7 on handling diagnosis imperfections in the decision process.

In the following section an atomic view on diagnosis is provided and in the subsequent sections different diagnosis approaches are presented.

4.1 Diagnosis Atomic Model and Metrics

In this section, a generic view on faults and fault diagnosis is introduced. This generic view will help to present diagnosis metrics, which will be used to assess diagnosis performance and provide model representations for the remediation decision problem.

In general the system diagnosis process can be seen as system state estimation where the system state in the studied case is the end-node subsystem, the available networks and the end-user service provisioning end-points. From this perspective the aim of the diagnosis process is to identify system normal and fault states correctly to avoid applying wrong, premature or delayed remediation. To represent this process a general diagnosis atomic model is introduced. The generic system view studied in this work is based on a single independent fault assumption and two network states assumption to form an atomic representation. This representation must help: 1) to define a diagnosis view for quantifying diagnosis performance including imperfections, and 2) to provide a model building block that can be applied in the decision modelling framework to be studied in Section 7. The atomic representation can, thus, represent the different potential faults that may occur in the end-to-end path assuming they occur and can be modelled independently. It may in future work also be scaled to handle more fault states.

The diagnosis process may be *event-driven* [124] where alarms from the network or certain patterns in observations trigger an update of the diagnosis mechanism to produce a new system estimate. It may also be *periodic* where new observations are made and/or collected periodically to produce a state estimate or a combination of both. In this work focus is primarily on the periodic approach. In the studied case it is not considered that network or observation pre-processing triggered events occur. Instead, observations are continually available and the diagnosis process must periodically determine the estimated state of a potential fault case. This periodic diagnostic process is depicted in the atomic representation of Figure 4.1 where periodically (with a fixed period T) a state estimate is produced.

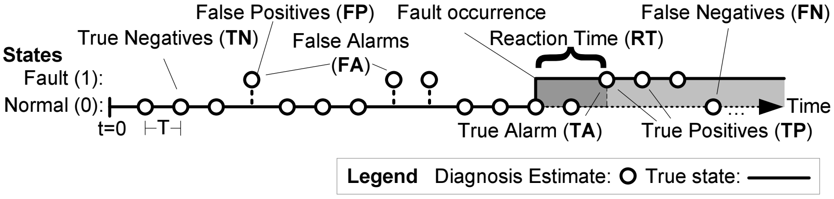

Figure 4.1: *Terminology of diagnosis outcomes.*

State estimation classification and diagnosis metrics

Considering the true fault state process and the diagnostic process in combination enables a classification of the diagnostic process performance. In the following the basic diagnosis performance metrics used in this work are defined.

Using a classical binary classification scheme the true state of the system may be defined as *negative* (N) when in a normal state and as *positive* (P) when in a fault state. When the estimated state and the actual state are equal the estimate is True (T) and False (F) otherwise. This leads to four possible outcomes of the diagnosis component of *True Negative* (TN), *False Positive* (FP), *True Positive* (TP) and *False Negative* (FN). Based on these definitions we may define more contextual metrics to be used in this work:

Accuracy - Accuracy is the amount of true estimates relative to all estimates and provides a metric of the overall state classification accuracy of the diagnosis component.

True and False Alarm - An alarm is the first positive experienced in a group of consecutive positives. An alarm is true if the system state is positive at the time of the alarm. An alarm is false if the system state is negative at the alarm.

Reaction Time - Reaction time represents the time from a fault occurs until it is diagnosed correctly by a true alarm.

Diagnosis Ratios - For every state of the system a ratio between the true/false estimates and all estimates of a particular state can be defined. I.e. the True Negative Ratio is defined as $TNR = \frac{\#TN}{\#FP+\#TN}$, where # refers to an amount. Similarly, it is possible to define $TPR = \frac{\#TP}{\#TP+\#FN}$. From these definitions it is further clear that $FPR = 1-TNR$ and $FNR = 1-TPR$.

These are the main metrics that in the following sections will be used to describe and assess the diagnosis approaches studied in this work. From these, derived metrics can be specified, which may be more representative for the studied fault management case or more practical to model. These derived metrics will be introduced in the following sections and chapters as needed.

Fault detection and diagnosis

With the simplified view on diagnosis presented in this section a close resemblance to fault detection theory is evident. I.e. diagnosing a fault state may implicitly be the detection process itself. In this relation the following chapters will refer to both diagnosis and detection theoretical aspects. However, in the framework of the ODDR fault detection is seen as a functionality in the OPP to identify that an error is present in the system. This may lead to actions such as the OPP intensifying certain observation processes to intensify the diagnosis process accordingly. In this way fault detection may be seen as a mechanism to conserve resources when the system operates in normal states well within tolerated boundaries. In this work, however, these detection mechanisms are not considered actively and, thus, focus is on the diagnostic process and a predetermined observation collection and pre-processing setup.

4.2 Introduction to Diagnosis Mechanisms

In the following chapters a set of diagnosis mechanisms are introduced to provide insights into how to increase diagnosis robustness to measurement errors (see Chapter 6), how diagnosis imperfections may affect end-user service reliability and how such imperfections may be handled in the remediation decision process (see Chapter 7).

Altogether, four diagnosis approaches are considered: i) Basic threshold, ii) a probabilistic Bayesian Network, iii) a heuristic named α-count, and iv) a probabilistic Hidden Markov Model based approach. The diagnosis mechanisms studied range from basic approaches to more complex and potentially more accurate and robust probabilistic diagnosis mechanisms. These different levels of complexity reflect a (non exhaustive) set of options for the system designer to populate the diagnosis component of the ODDR e.g. depending on available resources.

In the following sections first the *basic threshold* is introduced followed by the *Bayesian Network* (BN). In contrast to the basic threshold BN introduces some model structure and correlation of multiple observations. To study these effects a comparison of the two approaches is finally presented. The basic threshold and BN diagnosis mechanisms share the property that they are *Memory-less*. They do not attempt to correlate observations over time. Instead, they rely on a snapshot of the observations available at that moment in time. This simplifies the diagnosis mechanism in the sense that no memory is needed to store previous samples. Another class of diagnosis mechanisms is here defined as *Temporal* diagnosis approaches. They are characterized by implementing some degree of memory, attempting to correlate previous observation samples to the most recent. This may improve their diagnosis accuracy at the cost of increased reactivity. Both the α-count heuristic Hidden Markov Model (HMM) exist in this category where the HMM like the BN introduces model structure. To clarify the considered difference on memory-less and temporal diagnosis approaches a comparison is made between the threshold and α-count heuristic.

The motivation to study both memory-less and temporal diagnosis are that memory-less diagnosis approaches may be simpler to model while temporal may offer better performance. How to model both in relation to the decision component is the key topic of Chapter 7. Aspects of memory in the diagnosis process have previously been discussed in the work of Daidone et al. [39] where these concepts are referred to as traditional *one-shot* and *over-time* diagnosis, respectively.

4.3 Basic Threshold Based State Estimator

A simple and widespread approach of discriminating system states is to use a basic threshold. For simple diagnosis or fault state detection tasks properly configured thresholds may offer a sufficient performance while demanding little implementation and computational complexity. Thresholds are widely used e.g. in network management systems where they may be associated to observation variables of link utilization, link delay measurements, CPU utilization or packet losses rate [134]. Breached thresholds then lead to alarm events that in an event correlation process, as previously discussed, can help stitch together a sufficient system state view to identify the actual fault cause. More than a single threshold can be associated to an observation variable to consider different normal states of operation and similarly different fault states [73].

In the general case, the threshold approach is defined as specifying the threshold $\gamma^r_{variable}$ where *variable* refers to the observation or system variable and r is a threshold identifier on the particular variable for multiple thresholds.

Diagnosis of congestion level fault and parametrization

Applying a threshold based state discrimination approach in the studied scenario the congestion fault is considered as an example. Thus, we define a single threshold on the RTT observation variable o_l^{RTT}. A diagnosis trace J is subsequently specified:

$$J^{(l)} = \begin{cases} 1 & \text{if } o_l^{RTT} > \gamma^r_{RTT}, \textit{ fault} \text{ state hypothesis} \\ 0 & \text{if } o_l^{RTT} \leq \gamma^r_{RTT}, \textit{ normal} \text{ state hypothesis} \end{cases}$$

Considering Figure 3.6 a threshold corresponds to defining a point on the x-axis attempting to separate observations belonging to either of the two distributions in an appropriate manner. Defining a threshold γ^r_{RTT} in this setting will clearly be non-trivial to ensure that most true fault state observations (true positives) are observed without resulting in an intolerable high amount of false fault state observations (false positives). The trade-off options for different threshold settings are presented in Figure 4.2 in a ROC curve [98]. The ROC curve has been obtained from RTT observations of a normal and fault state in 100 independent simulation runs transferring a 10 Mbyte data file using $\omega_{RTT} = 0.3\,s$ and a sample made every $T \approx 0.4\,s$.

Clearly, the threshold based approach is better than a random guess. However, the diagnosis approach is also far from the perfect case, which would be FP=0 and TP=1. Thus, the common challenge is to select a good threshold value that can offer the required diagnosis trade-off between false alarm rate (FP) and mean reaction time (depending on TP) [15][73]. In this work two main approaches for defining thresholds settings are applied:

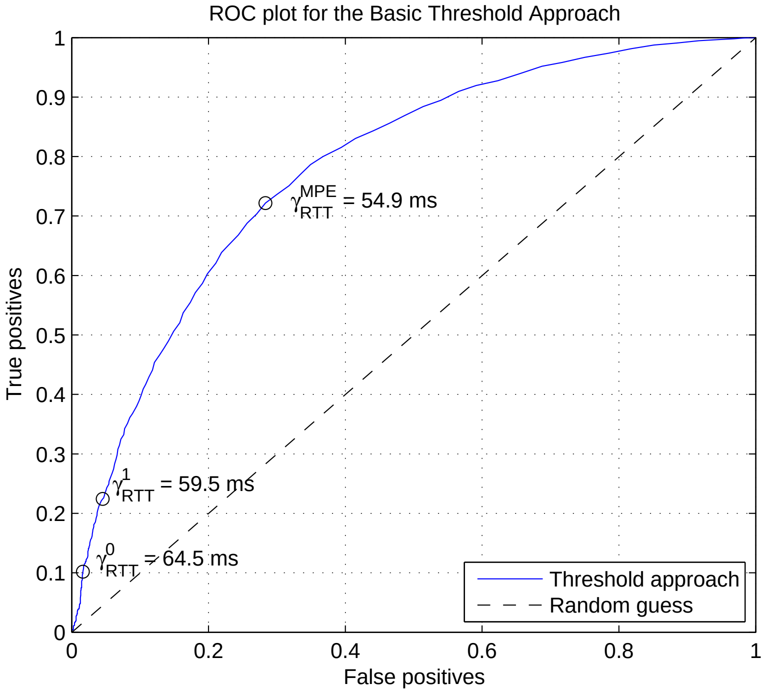

Figure 4.2: *ROC curve showing the trade-off between false positives (leading to false alarms) and true positives.*

Fixed empiric setting - A set of empiric thresholds is selected based on the ROC curve to represent different interesting trade-off settings. The empiric settings will help to assess how a given diagnosis imperfection performance set (i.e. considering reaction time versus false alarms) may be mitigated in the decision process. This aspect is discussed further in Chapter 7. A set of specific settings studied in this work are depicted in the ROC curve being γ^0_{RTT} and γ^1_{RTT}. Details on criteria for selection of these particular settings are presented in section 7.3.

Minimum Probability of Error (MPE) - More formal approaches exist for selecting a useful threshold. An example is the well known Bayesian detector (e.g. see Reference [54]), which enables to encode a-priori information on the fault process (if any) as well as to utilize the conditional distributions $P(o_l^{RTT}|N_s = normal)$ and $P(o_l^{RTT}|N_S = fault)$. A decision rule is used by the Bayesian detector to set the threshold defining which state hypothesis to map an observation to. The *minimum probability of error* (MPE) decision rule is an example of a rule, which equally tries to minimize false positives as well as false negatives. While this approach optimizes for maximum state estimation accuracy it does not necessarily offer best end-user service performance. The accuracy metric may, however, be useful to evaluate diagnosis approaches when no immediate diagnosis context exists (e.g. end-user service requirements) that would give reason to weigh false positives different from false negatives. An example of a threshold obtained by the MPE decision rule (assuming equal prior probabilities on normal and fault states) is depicted in the ROC curve. While the true positives are increased dramatically so are the false positives.

A final relevant parametrization approach could be to reverse the *fixed empiric setting* analysis by identifying the best diagnosis setting given a certain decision strategy. This approach is in Chapter 7 studied for the more complex α-count diagnosis heuristic, which is introduced later in this chapter.

A basic threshold approach may provide useful diagnosis performance at a low implementation complexity. It may, however, be sensitive to even small changes when network conditions change [73]. Also, threshold approaches may have difficulties dealing with unreliabilities such as missing or ambiguous observations (due to multiple potential network fault affecting the observation variable). In the following sections it is considered how robustness to network changes and observation imperfections may be dealt with in a probabilistic framework correlating multiple observation variables .

4.4 Bayesian Network Approach

An approach to overcome unreliable observations and provide robustness to changes is to use and correlate several observations across multiple protocol stack layers. This is, however, not trivial as more observations can lead to contradicting observations and increased ambiguity. To target these challenges we perform a study of using a Bayesian Network (BN) for diagnosis. A BN offers a formal approach of formulating the diagnosis problem by encoding: 1) the basic fault models, 2) a set of cross-layer observations, and 3) system properties such as the impact of faults on TCP/SCTP behavior and further on observations.

The BN inherently enables the inclusion of multiple faults in the diagnosis model. Thus, in the presented and studied model both the congestion and contention faults are considered.

4.4.1 Background on Bayesian Networks

A Bayesian Network is a graphical model that relates stochastic variables of a domain by their causal relations. Formally, a BN $N = (\mathcal{G}, \mathcal{P})$ consists of two basic entities [81]: a *directed acyclic graph* (DAG) $\mathcal{G} = (X, E)$ where X is a set of nodes $X = \{X_1, \ldots, X_\text{n}\}$ and E is a set of edges connecting the nodes. $\mathcal{P}$ is a set of conditional probability distributions (CPD). Each node

X_i represents a variable with a finite set of states and each edge represents a causal relation between two variables. One strength of BNs is that independence between variables can be utilized meaning that only a part of the conditional probabilities present in $P(X)$ needs to be specified in $\mathcal{P}$. This makes it practical to construct and parametrize such models. Most importantly, BNs also make inference in P(X) computationally feasible for large models [133]. More on the background of BNs may be found in [81].

Developing a BN for fault-diagnosis is a three-step process: (1) *obtaining domain knowledge*, (2) *developing a BN structure* and (3) *obtaining probabilities*. This process is described briefly in the following part of the section for a BN considering the previously introduced two-fault scenario.

4.4.2 Diagnosis Model

To construct the BN structure $\mathcal{G}$, variables and their causal relations are mapped from the fault diagnosis scenario to a graphical representation. To provide information on the two faults all of the observation variables introduced earlier are included in the more. These are RTT for congestion, FRR for contention and PRR, which can ambiguously represent information on both faults.

It is a condition that the graph representing the BN is a Directed Acyclic Graph (DAG). In practice this means that the BN cannot model dynamic effects such as the strong influence of TCP congestion control on the faults and observation variables. The consequence is that different effects are difficult to capture in the model. An example is delay between observation variables. It takes time before an increase in FRR leads to increased PRR. Also, throughput could be a useful observation variable. However, there will also be a time delay from a fault manifests itself in the observation variables RTT, PRR and FRR before the throughput is strongly affected by TCP transmission control. Thus, this observation variable cannot easily be included. Details on these considerations for developing a functional BN model can be found in Appendix D and more details are given in references [104] and [102].

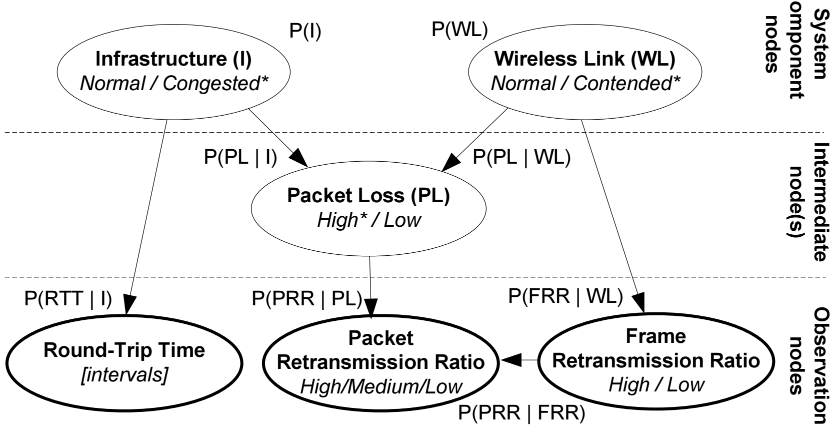

Figure 4.3: *Basic Bayesian Network model for diagnosis of congestion level and contention faults. *Fault states.*

An overview of the final BN for the studied problem is presented in Figure

4.3. In the upper part are the nodes that define the system under diagnosis including its normal and fault states. An intermediate node represents the packet loss level while observation nodes are located in the lower layer. Starting from the system component nodes, two directed edges point to the packet loss node, which they clearly both affect. In addition, the state of the infrastructure independently affects the Round-Trip Time while the wireless link independently affect the FRR. As an example, independence is assumed between the *Infrastructure* variable and the FRR observation. Under this interpretation, a change of the infrastructure variable in the infrastructure networks should not have a significant effect on the ratio of frame retransmissions. As the infrastructure fault does not directly affect the wireless conditions this seems to be a reasonable assumption.

As no knowledge of features in observations are known in advance, simple state spaces have been defined. The RTT is represented by a set of equally spaced intervals representing the thresholds between discrete states. The FRR is defined by high and low which are divided by a single threshold. PRR is defined by high, medium and low. Compared to FRR (which is only depending on wireless link conditions) the introduction of an additional state enables different expressions of when a single or two faults have occurred. Finally, the states of the packet loss variable are high and low.

4.5 Comparison of Threshold and BN Approach

The BN diagnosis performance has been evaluated in a simulation study where its performance is illustrated in relation to the basic threshold based state estimator approach. Note, for the threshold based approach observations and network state variables are mapped one-to-one. I.e. the RTT observation is mapped to *Infrastructure* and the FRR is mapped to *Wireless Link* while no threshold on PRR is considered. The general MPE rule has been used to select Optimal Thresholds (OT) being γ_{RTT}^{MPE} and γ_{FRR}^{MPE}. In comparison to the BN, the OT approach requires less effort to model and parametrize.

Network A of the previously introduced simulation model is used with the parameters summarized in Table 4.1. Notice, as these parameters have been defined in a study independent of the parameters in Section 3.3 there is some deviation between the two. However, they have been created on the same background on fault definitions (method to create fault states, impact of fault states on throughput, etc.). Thus, the conclusions of this section are expected to generally apply to the studied scenario.

Based on the simulation setup the conditional probability tables of the proposed BN have been learned from network data where the true network states are known. Details on learning and actual parameters can be found in Appendix D. It must here be noted that the state-spaces of the individual observation nodes have been defined based on the optimal thresholds γ_{FRR}^{MPE} and γ_{RTT}^{MPE} where the latter aligns the bins of the RTT observation variable. Using these thresholds is done to pursuit good observations for the BN and ensure comparability with the OT approach. As the OT approach does not specify a threshold for PRR, for consistency, optimal thresholds based on the MPE decision rule have been applied for γ_{PRR}^{MPE} as well.

The diagnosis capabilities of the BN and basic threshold based approaches

Link Name	**Bandwidth**	**Delay**
Wireless Access	$10\,Mbit/s$	$1\,ms$
Radio Access	$10\,Mbit/s$	$5\,ms$
Congestion Link	$5\,Mbit/s$	$20\,ms$
Service Provisioning Infrastructure	$10\,Mbit/s$	$5\,ms$
Cross-traffic Links	$10\,Mbit/s$	$5\,ms$

State Definitions		
State	λ_{cgA}	p_{ctA}
Normal	18 conn./s	$1\,\%$
Fault	47 conn./s	$4\,\%$

Table 4.1: *Simulation parameters for the Bayesian Network and the basic threshold comparison study.*

are directly determined by how efficiently they can estimate the state of the network. A measure of timeliness is obtained by the mean *reaction time* and correctness of state estimation capabilities by mean state estimation *accuracy*. Finally, a mean *false alarm count* defines how many false alarms occur in a simulation run on average.

Assuming an application would define a requirement that faults must be diagnosable within few hundreds of milliseconds a diagnosis sample period of $T = 0.1\,s$ has been chosen. In addition, the window size $\omega_{PRR} = 300\,ms$ is set. Also choosing $\omega_{FRR} = \omega_{PRR} = 300\,ms$ helps keeping reaction time low while ensuring some correlation between frame losses and packet losses where the latter is delayed due to TCP detection latency (by timeout or duplicate acknowledgments).

In the following paragraphs, a summary of the results obtained by comparing the basic threshold based approach and the BN is presented. Focus is on the diagnosis performance and the robustness to changes in the system environment when model parameters are maintained. For reasons of brevity only contended wireless link results are presented in detail in the diagnosis performance study. The robustness results are based solely on the diagnosis of infrastructure congestion. In a scenario, a simulation run is conducted of a data transfer for a fixed duration of 35 seconds. At $20\,s$ a fault deterministically occurrence. With a warmup period of $5\,s$ this means that diagnosis time in normal and fault state is equally distributed. This approach is useful to obtain a diagnosis accuracy metric that is well balanced between fault and normal state diagnosis capabilities. Altogether, 30 independent simulation runs are made.

4.5.1 Diagnosis Performance

Both the BN and the OT approaches have been evaluated using the same set of observations provided by observation pre-processing. The Wireless Link diagnosis performance results are listed in Table 4.2. Conveniently, the reaction time results for both approaches are approximately the same given the selected observation window sizes. This allows a fair comparison on the remaining metrics. As seen from *accuracy* the BN, overall, leads to a better state estimation

	Reaction T. [ms]	**Accuracy [%]**	**False Alarms #**
OT	323 (±83)	73 (±1.4)	6.7 (±0.8)
BN	323 (±30)	77 (±0.7)	7.8 (±0.3)

Table 4.2: *Performance results for Wireless Link diagnosis (Ci=95%).*

than the single threshold based approach. The cost, however, is a higher mean amount of false alarms in the simulation runs.

To provide insights into these differences, cases have been identified from all simulation runs of the contended wireless link results where the BN and the OP make different state estimates. The results are shown in Figure 4.4 ordered by *evidence vectors*. An evidence vector consists of the set of observations at a discrete time step. The integer defines the state of the observation where $FRR = 0$ corresponds to low frame retransmission rate, $PRR = 2$ corresponds to high retransmission rate and finally the value of RTT the given RTT-state (see Table D.2) with '-1' corresponding to a missing observation.

From the figure it can be seen that the $FRR = 0$ (low retransmission rate), which leads the OT approach to diagnose a normal state. From the additional information (low RTT, high level of packet retransmissions) this leads the BN to diagnose a fault state. In a significant amount of these cases the BN clearly provides the better state estimate. Altogether, these results explain the increased wireless link state estimation accuracy in Table 4.2. It is, however, also clear that these evidence vectors do not occur very often, only 6.3% in total (568 out of 9012 cases), as it is the occurrence probability of these vectors that determines the improvement of the BN over the OT. For the *congested infrastructure* scenario (not depicted) only 1.8% of cases between the OT and BN differ, as the RTT observation alone is a strong observation for the congestion level.

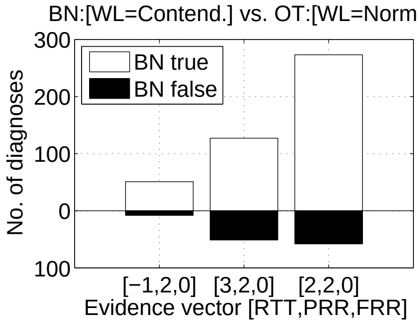

Figure 4.4: *Occurrences of observations that lead to contradicting wireless link state diagnoses.*

The contradicting cases show that the constructed BN based on its structure and learned probabilities is capable of using additional observation to improve diagnosis performance.

	Acc. [%] (before)	**Acc. [%] (after)**	**FP Prob. (after)**
OT	87 (±1.6)	49 (±0.35)	1 (±0.00)
BN	86 (±0.6)	52 (±0.004)	0.95 (±0.01)

Table 4.3: *Robustness results for diagnosis of infrastructure congestion before and after a mean increase of* $4.3\,ms$ *in the RTT observation variable (Ci=95%).*

4.5.2 Robustness to Changes

As discussed previously, networks are highly dynamic and changes may occur from route changes in the end-to-end path, changes of traffic levels etc. Assessing the robustness properties of the two diagnostic approaches has been realized by changing the network delay without re-parameterizing the BN or the OT approaches. The impact has been assessed on infrastructure congestion diagnosis. In the presented example the delay has been increased by $4.3\,ms$ in relation to an original mean RTT around $80\,ms$. $4.3\,ms$ is the bin size of the discretized RTT vector meaning that observations are shifted by a single state.

Results are presented in Table 4.3. From the table it is seen how the accuracy degrades significantly from before the change. The probability of false positives shows that for the delay increase OT estimates '*Infrastructure=Congested*' constantly. The BN is also strongly effected, yet, it remains capable of diagnosing some normal states. This illustrates how the BN can be more robust than the OT to network changes, due to the use of additional observations. However, as the BN is also strongly sensitive to RTT observations its performance good performance is not maintained.

Finally, the BN differs from the OT in one other important property; it can utilize prior belief and additional observations to make a diagnosis even when observations in the evidence are missing. The OT, only using one observation, cannot provide an estimate without an observation sample. Potentially, the previous diagnosis could be used again or the normal state could be selected by default. In this work no assumptions have been made on what would be the best approach, as this may be dependent on the cost of making a wrong diagnosis. Consequently, cases where the OT does not lead to an estimate have simply not been included in the statistics of tables 4.2 and 4.3. This may be disadvantageous to the BN in terms of accuracy and FA. In our results, nonetheless, the BN performance benefits from the property. This has been observed in both performance scenarios. The BN infers infrastructure congestion correctly in 130 out of 150 cases without an RTT observation, and contended wireless link in 6 out of 7 cases without an FRR observation.

4.5.3 Summary on Memory-less Diagnosis Mechanisms

Up until this point, two diagnosis mechanism approaches have been introduced. i) A basic threshold approach, which enables to perform basic state estimation based on a single observation variable, and ii) a more complex approach using a Bayesian Network. In an example design, results of the Bayesian Network approach have been established. Compared to the basic threshold approach the BN offers better accuracy, larger robustness to network changes and, finally, an option to handle missing observations. The cost is an increased complexity to develop and maintain a model and increased resource requirements to solve it. The presented model has been evaluated using exact solution methods with an average inference time of 0.5 ms (on a $1\,GHz$ class PC running Linux). However, in a complete setting with multiple fault scenarios and potentially more observation variables the complexity may increase drastically. Solutions are to use inexact inference methods and letting the decision component control the needed solution accuracy and/or distribute the diagnostic task among multiple end-nodes. The latter approach has been studied in the work of reference [125] as a viable option.

While the BN based approaches may have good state estimation properties and properties to handle observation unreliabilities, in the following chapters focus will be on how to handle diagnosis imperfections. Thus, when applicable, results from the basic threshold based approach will be used as it is simpler to parametrize and implement.

As mentioned in the introduction of this chapter, the diagnosis mechanisms of the threshold and BN have the common trade of not including memory. In this sense the diagnosis components do not correlate observations and knowledge over time. In the following section diagnosis components studied in this work with memory and observation correlation properties are presented.

4.6 Temporal Diagnosis Mechanisms

Including memory in a diagnosis component to enable correlation of knowledge from past observations with latest observations can be a viable approach to track an overall trend of an observation and filter out transient changes. The potential advantage is to obtain good state estimation properties at the cost of an increased response time to changes, which negatively affects the reaction time metric. This again relates to the usual trade-off between false alarms and reaction time, however, there are some differences in how the diagnosis behavior is characterized and its potential to improve diagnosis metrics overall. To study this type of diagnosis mechanism, in this section two temporal diagnosis approaches are introduced: A heuristic known as α-count [25] and an approach based on the Hidden Markov Model formalism.

4.6.1 Heuristic: α-count

The initial diagnosis approach introduced is a heuristic based on the work of Bondavalli et al. [25] named α-count. The general motivation of studying this heuristic is that it is simple in its operation and has little computational complexity. In this respect, the α-count heuristic may be compared to the simple threshold based approach of Section 4.3 with the difference that α-count has memory and temporal correlation of observations. In the following, the α-count heuristic is introduced and compared to the threshold approach in terms of diagnosis properties.

In the original work of reference [25] and further studies in reference [39] the α-count mechanism is introduced as a mean to determine the nature of a fault, i.e. whether it is transient, intermittent or permanent. The mechanism belongs to a scheme of count-and-threshold where fault indication events are counted (for a given time period) and a threshold on the count is used to determine if the fault may be considered as tolerable (transient) or permanent and if an action is required. This principle is directly translatable into the basic diagnosis issue of discriminating a fault state from a normal state based on an observation mechanism. As the α-count mechanism assumes that such observations can only be '1': an alarm event or '0': no alarm event, the basic threshold approach is in this work defined as this observation mechanism. Now, assuming an observation process $J^{(l)}$ consisting of normal (0) and fault (1) state estimates at discrete time l. The α-count is specified as follows:

$$\begin{aligned} \alpha^{(0)} &= 0 \\ \alpha^{(l)} &= \begin{cases} \alpha^{(l-1)} \cdot k & \text{if } J^{(l)} = 0 \quad (0 \leq k \leq 1), \\ \alpha^{(l-1)} + 1 & \text{if } J^{(l)} = 1 \end{cases} \\ M^{(l)} &= \begin{cases} 1 & \text{if } \alpha^{(l)} > \alpha_T, \\ 0 & \text{if } \alpha^{(l)} \leq \alpha_T \end{cases} \end{aligned} \tag{4.1}$$

where k can be defined as a *forgetting* factor, α_T the α-count threshold, and $M^{(l)}$ is the output diagnosis trace. In this manner, the α-count heuristic only requires a single memory variable. This can clearly be useful in very basic hardware configurations as given in sensor nodes where potentially multiple diagnostic functions are implemented.

Diagnosis of congestion level fault compared to basic threshold

The α-count heuristic has been evaluated on the congestion fault scenario using parameters introduced in Section 3.3. It is studied how it compares to the basic threshold approach when considering the transient diagnosis behavior.

When addressing transient behavior in the diagnosis process, focus is on how the diagnosis process reacts to system changes in terms of its performance over time. This behavior is primarily influenced by two aspects: i) the functions of the diagnosis component itself, and ii) how a system change (e.g. network fault occurrence) in time is propagated through system components to, finally, manifest itself in observations. In relation to the latter a given observation may on a timely basis become stronger or weaker in indicating that a fault has occurred. When considering the diagnosis component the transient effects are given by changes in the accuracy of the state estimate as more observations are collected over time.

Figure 4.5 depicts example diagnosis characteristics over time for three different diagnosis traces. They show mean normal state diagnosis probabilities (period $T \approx 0.4\,s$) obtained from 2000 independent simulations of a normal to fault state transition at $t = 30\,s$. These have been obtained from: i) a basic threshold on RTT observations, ii) a model based representation of the threshold approach ensuring independent diagnosis outcomes, and finally iii) an α-count filtering where the observation process $J^{(l)}$ is the output of ii).

Considering, initially the (memory-less) basic threshold approach on the RTT observation it is seen that within around $1 - 1.5\,s$ after the fault occurrence the estimation output transitions to a level near its steady state behavior. Following, is a $5 - 6\,s$ period before it finally settles at steady state. This transient effect is mainly attributed to dynamic interactions between the end-user

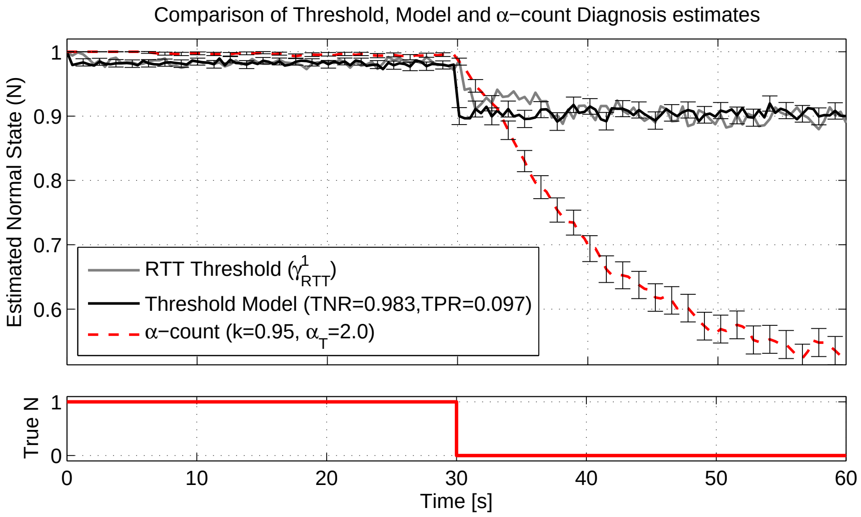

Figure 4.5: *Impact of a fault occurrence on diagnosis estimates.*

service data transfer and the increased amount of cross-traffic when the fault occurs. The model based representation of the RTT threshold approach has been obtained by parameterizing a Bernoulli process for each of the states (parameters p_{TN} and p_{TP}). The diagnosis traces have been generated from stochastic simulations of this model. This model representation is useful as it allows a full characterization of the diagnosis component by the tuple: (True Negative Ratio (TNR), True Positive Ratio (TPR)) where $p_{TN} = TNR = \frac{\#TN}{\#FP+\#TN}$ and $p_{TP} = TPR = \frac{\#TP}{\#TP+\#FN}$. This can be beneficial to ensure a simple diagnosis model representation and parametrization approach when considering decision models. However, as evident from the figure, this also implies the assumption that the transient effects of the threshold based approach on the RTT observation variable can be considered insignificant. The impact of these approximations are discussed further in Chapter 7.

Finally, the α-count results have been obtained using the model based traces as an input to only consider the transient effects added by the diagnosis component itself. The selected α-count parametrization represents an example where a significant improvement in the state estimation capabilities are observed. From these traces important observations can be made regarding the temporal α-count diagnosis component. Compared to the threshold model results, in the normal state the state-estimation accuracy is improved (close to 1). This means that the amount of false alarms can be significantly reduced. In the fault state the estimation accuracy increases dramatically from a ratio of 0.1 to nearly 0.45 in the steady state. However, there is a significant transient phase for the α-count, which can negatively affect the reaction time. These results also show that characterizing the α-count performance by the parameters TNR/TPR under an independence assumption (as in case of the basic threshold) is not possible as they strongly depend on time and cannot in a simple manner be derived from traces of a diagnosis component.

Seen from the holistic perspective of the diagnosis-decision interaction in the ODDR framework, the trade-off options provided by the temporal model approaches are interesting. Remediation decision strategies may be studied that wait for diagnosis to get better before reacting if remediation on a false alarm is costly and the end-user service is not in a critical stage. Identifying good settings for the parameters of the α-count mechanism, and in this setup the basic threshold, is however not straight forward. To assess the potentially improved diagnostic properties of the temporal model approaches the α-count heuristic is revisited in Section 7.4 where a specific service reliability model is used to identify the setting of the diagnosis component that will offer the best trade-off of e.g. reaction time and false alarms. The problem of representing the transient characteristics of temporal diagnosis mechanisms in the decision model is also addressed.

4.6.2 Hidden Markov Models

In this section a short introduction to a more complex diagnosis formalism is provided; namely, the Hidden Markov Model (HMM). Like the previously studied BN, the HMM encodes some system structure, which provides a formalized and straight forward approach of formulating the diagnosis problem. In addition, it has been shown to enable good diagnosis accuracy [39]. In existing work an HMM for diagnosis has been studied in different contexts. In reference [39] the authors study the problem of diagnosing the nature of a fault (and compare it to the α-count heuristic). The authors of [141] apply a HMM in a multi-fault multi-component scenario whey they estimate the most probable system state evolution (including fault states) to perform component failure diagnosis while overcoming unreliable binary tests of the individual components. The authors also demonstrate how learning the HMM parameters can be done on-line to adapt to changes. The HMM formalism has been applied in the work of [93] to classify observations of round-trip times before a packet loss to diagnose if the cause of the loss is congestion or a wireless loss somewhere in the end-to-end path.

For this work, the properties of the HMM approach have been studied in the congestion diagnosis setting to inherently estimate (hidden) network system states based on stochastic observations using the RTT observation variables. This section is focused on the formulation of the studied HMM. In Chapter 6 the model is extended and studied in more detail with approaches to increase its robustness to uncertainties in the observations provided from the OPP component in the ODDR framework.

Background on HMMs

In general, a Hidden Markov Model can be described as a Markov Model (first order, homogeneous and discrete) in the transition probability matrix A with the state space S. For a fully specified model the size of the state space is known as well as the state transition probabilities. However, the process is not directly observable (hidden). It is assumed that the stochastic Markov Model process can be observed indirectly through a stochastic process (given by the observation sequence X). Having knowledge about the conditional probabilities (in the observation symbol probability distribution matrix B) of observing certain observations given states in S, the HMM can be used to derive an estimate of

Symbol	Description
$l, (1 \leq l \leq L)$	Discrete time step l up to L.
$X = \{x_1, x_2, ..., x_L\}$	Error free observation sequence.
$S = \{s_1, s_2, ..., s_N\}$	State space of HMM.
N	The number of states in the HMM.
$Q = \{q_1, q_2, ..., q_L\}$	State of the HMM at time l.
$A = \{a_{ij}\}$, $(1 \leq i, j \leq N)$	The hidden process Markov Model transition probability matrix.
$V = \{v_1, v_2, ..., v_M\}$	Discrete symbol alphabet describing the possible values of observations in X.
M	Number of symbols in V.
$B = (b_j(k))_{j,k}$, $(1 \leq k \leq M)$	The distribution of symbols in V in state j.
$\pi_i = P[q_1 = s_i]$	Initial state distribution.

Table 4.4: *Summary of important symbol definitions for the HMM approach.*

the hidden Markov Model state sequence given a sequence of observations. See [115] for an elaborate description of the HMM formalism. In order to design the HMM it is necessary to specify the model $\lambda(A, B, \pi)$, where π is the initial state distribution. Table 4.4 provides an overview of the HMM symbols and their definitions as used in the remaining part of this work.

Using the HMM for diagnosis of the congestion level fault
To clarify how the HMM may be used for diagnosis in the studied setting an initial instantiation is made based on the atomic diagnosis model in Section 4.1 and RTT observation distributions of Figure 3.6. In a real network diagnosis context the HMM model structure and parameters may be learned [119]. This is a useful property for the studied use case but out of the scope of this work to be considered. In the following paragraphs the steps to define the HMM diagnosis approach are highlighted.

Hidden Markov Model - (A)

For the case study a simple two-state system model is assumed describing a *normal* and *fault state* (congested). The state holding time for a network state is assumed to be geometrically distributed following an ON-OFF process. Thus, the HMM is defined as:

$$A = \begin{pmatrix} 1 - p_f & p_f \\ p_r & 1 - p_r \end{pmatrix} \tag{4.2}$$

where p_f is the fault occurrence probability and p_r is the fault repair probability.

Observation Distributions - (B)

The conditional observation symbol probability distributions are defined as:

$$b_j(k) = P(v_k | q_l = s_j), \quad 1 \leq j \leq N, \; 1 \leq k \leq M \tag{4.3}$$

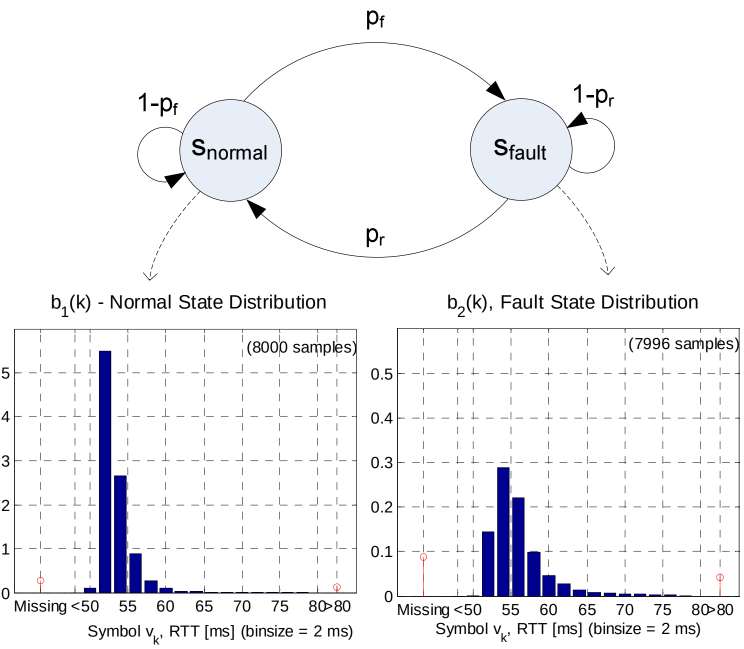

Figure 4.6: *Hidden Markov Model summary.*

where $B = \{b_j(k)\}$, q_l is the current state at discrete time step l and $V = \{v_1, v_2, \ldots, v_M\}$ defines a discrete observation symbol alphabet. Considering b_j as a discrete distribution, rather than a continuous, enables to represent an arbitrary distribution of the networking delays without requiring distribution fitting.

For this realization of a HMM for diagnosis a basic discretization approach is applied where equally sized bins are defined to describe $\{v_1 \ldots, v_{M-1}\}$ in the range from $bin^{lower} = min(x_l)\,[ms]$ to $bin^{upper}\,[ms]$. Note, as RTT delay distribution tails are long (describing rare events) as seen from Figure 3.6, good tail representations may require an impractically large dataset. Thus, for parameterizing the model the symbol v_M represents an accumulated probability of the upper tail of the delay distribution in the interval: $[bin^{upper}, +\infty]\,[ms]$. For this study the initial parameters $M = 16$ and $bin^{upper} = 80\,ms$ have experimentally been found to provide useful results, however, these parameters are expectedly not highly sensitive. Finally, the symbol v_0 is introduced to represent *missing observations*.

Initial State Distribution - (π)

For the conducted studies it is assumed that the network state process always starts in a normal state ($\pi = [1 \quad 0]$), thus, $q_1 = s_{normal}$.

A graphical representation of the diagnosis HMM model is presented in Figure 4.6 representing the system state ON-OFF process and the distributions of the emitted observation symbols for each of the two states using the state definitions of Figure 3.6. Applying this HMM based diagnosis mechanism in online state estimation is realized by applying the *forward algorithm* (details may be found in [115]), which produce the *a posteriori marginal distribution* [119], $\beta(i) = P[q_l = s_i|X]$. To provide a final state estimate a threshold γ_{fault} is introduced on $\beta(fault)$:

$$\hat{s}_{fault} = \begin{cases} 1 & \text{if } \beta(fault) > \gamma_{fault}, \\ 0 & \text{if } \beta(fault) \leq \gamma_{fault} \end{cases} \tag{4.4}$$

This threshold allows to control the diagnosis trade-off for the HMM approach similar to γ_{RTT} for the basic threshold approach or (α_T, k) for the α-count heuristic. To visualize the HMM output and γ_{fault} an example trace is depicted in Figure 4.7 for $\beta(fault)$ using $X = O^{RTT}$ where $\omega_{RTT} = 0.3\,s$ and system parameters of Table 4.5. Λ_f and Λ_r are the fault and repair rates controlling the exponential state holding times of a given ON-OFF network state process. Later in Chapter 7 end-user service usage durations (data transfer time) of 20-60 seconds will be considered. In this relation, the defined rates have been chosen to define a network state process where faults occur and are repaired frequently. For consistency, these rates are also considered in this example along with the diagnosis period $T \approx 0.4\,s$ also applied in Chapter 7.

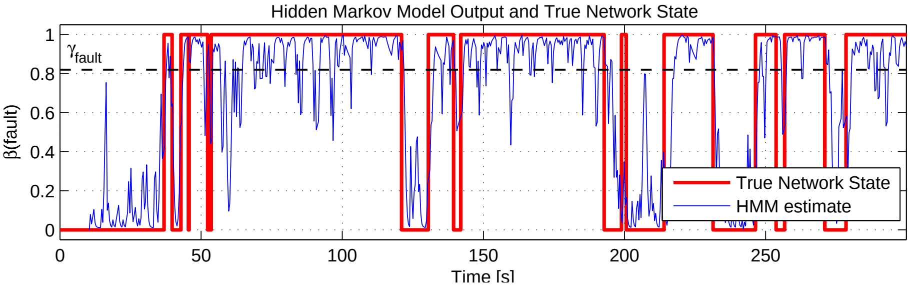

Figure 4.7: *An example HMM trace of $\beta(fault)$ of congestion level diagnosis based on RTT observations.*

The example trace shows how the fault state probability $\beta(fault)$ overall follows the actual network state process. It is also seen that there is some fluctuation in the state estimate that can lead to false alarms. A direct performance comparison to the threshold approach has not been performed for the HMM. However, in chapter 6 details of the diagnosis performance of the HMM based diagnosis component are analyzed for error free observations and observations with observation uncertainty due to measurement error.

Parameter(s)	*Value(s)*
$P[q_1 = s_{normal}]$	1
T	$0.398\,s$
Frequent Fault Event (short repair time)	
$1/\Lambda_f$, $1/\Lambda_r$	$12.42\,s$, $15\,s$
SS probability: *normal*, *fault*	0.453, 0.547

Table 4.5: *Summary of parameters used in the HMM diagnosis example.*

Finally, it must be emphasized that the derivation of $\beta(i)$ is based on a recursive function providing an accumulation of $\beta(i)$ based on most recent observations and their likelihood of belonging to one hidden state over another. Consequently, the HMM approach is similar to the α-count byonly requiring a single memory variable to represent past progress. The use of memory, however, makes it belong to the temporal class of diagnosis components as defined in this chapter. More details can be found in [115].

4.6.3 Summary on Temporal Diagnosis Mechanisms

In this section two diagnosis mechanisms have been introduced and specified to be used for diagnosing congestion level states in the studied scenario. The α-count heuristic represents a simple count-and-threshold mechanism. It requires little effort to implement, but may not be straight forward to parametrize to a particular scenario. A Hidden Markov Model approach partially addresses this problem by adding model structure and knowledge about a given scenario. The cost is increased complexity to specify it and increased implementation effort compared to α-count. As a common element, these mechanisms incorporate memory to correlate past knowledge to the most recent observation. This promises to increase state diagnosis accuracy compared to what can be achieved by the compared memory-less basic threshold. A cost is, though, likely to be paid by an increased reaction time as more observation samples must be collected to achieve a good state estimate.

The introduction of memory in the diagnosis component can lead to significant time-dependent performance characteristics making it difficult to characterize the diagnosis performance by a simple TPR, TNR tuple. This will have an impact when trying to capture the imperfect diagnosis performance in a model used to determine best decision policies or best diagnosis settings given a specific reliability problem. This topic is studied in Chapter 7. Finally, it should be noted that same time dependent diagnosis behavior could exist in a setting including memory-less diagnosis approaches. This could occur in cases where observations used for diagnosis change properties over time. However, in this work it is assumed that the basic threshold approach, without memory, on RTT observations may be approximated by a Bernoulli process ensuring independent diagnosis outcomes and a time invariant diagnosis performance.

4.7 Conclusion

In this chapter the overall challenges of diagnosis in distributed networking systems have been introduced focusing on unreliable observations and the resulting diagnosis imperfections. As a background for the thesis a set of diagnosis mechanisms/approaches have been introduced representing *Memory-less* approaches based on a single-shot view of observations in time and *Temporal* approaches where observations are correlated over time to provide potentially better diagnosis characteristics.

In the following a brief summary of the introduced diagnosis mechanisms is provided emphasizing their basic properties and why they are considered in this work.

Basic threshold

Implementation, parametrization and computational complexity

Very low implementation and computational complexity makes basic thresholds a widespread approach for simple fault diagnosis (and detection) [134].

Provided that historical data is available about a given observation variable in an invariant system a full characterization of the trade-off options of different threshold settings can be obtained in a ROC curve. The always existing challenge is, thus, for the system designer to determine the tolerable amount of false alarms in relation to the reaction time.

Expected diagnosis capabilities and imperfections

Not including any knowledge from past observations or the system structure a basic threshold approach is considered to be the most sensitive to unreliable observations of the three studied approaches. Both the BN and α-count have been demonstrated to provide improved state estimation accuracy in this chapter.

The basic threshold approach is used in this work as: 1) it is simple to implement and configure, 2) it may (under the limitations of the studied scenario) be represented by a simple Bernoulli model, and 3) it offers the basic trade-off options of diagnosis imperfections relevant to study different decision strategies in Chapter 7.

Bayesian Network

Implementation, parametrization and computational complexity

Of the diagnosis mechanisms studied, the BN approach represents the highest complexity of implementation, parametrization and computational complexity. Significant efforts are required to specify the models being system variable dependencies, conditional and a-priori probabilities. Some of these complexities can, however, be mitigated by well established learning mechanisms e.g. for online adaptation [72]. BNs generally impose a high computational complexity, which in worst case is exponential. This issue can partially be mitigated by approximate [124] and distributed solution approaches [125].

Expected diagnosis capabilities and imperfections

As shown in this chapter the BN can inherently correlate information from multiple observations to improve diagnosis accuracy and robustness to changes in the system. This makes the BN approach a highly relevant candidate for the diagnosis component to increase its tolerance to observation imperfections.

Due to the inherent capabilities of performing diagnosis under uncertainties the BN is an interesting candidate for the diagnosis component when computational resources are available. Thus, it has been included in this analysis. As the primary focus of this work, however, is not to obtain the best possible diagnosis capabilities the BN is not analyzed further in the subsequent chapters.

α-count

Implementation, parametrization and computational complexity

The α-count heuristic represents a low implementation and computational complexity.

In terms of parametrization it, however, has two parameters (three if including the threshold as in the studied implementation of this work), which are difficult to map directly to the system properties.

Expected diagnosis capabilities and imperfections

As shown in this chapter, introducing memory has the potential to strongly improve the state estimation accuracy with few means in comparison to the basic threshold. This makes the α-count heuristic relevant to consider in this work as a representative for temporal diagnosis approaches which is also simple to implement. The problem of parameterizing α-count to obtain good diagnosis performance, is studied in Chapter 7.

Hidden Markov Model

Implementation, parametrization and computational complexity

Given that realistic fault models exist that are or can easily be represented as a Markov chain (as the case in this work) the HMM formalism provides a straight forward approach of formulating a diagnosis model and using it to infer the most likely system state. As for any of the diagnosis mechanisms a challenge is to obtain expected observation statistics for a given state. As in the case of BNs also for HMMs well defined approaches exist for online adaptation [142].

Disregarding more advanced mechanisms for online adaptation the implementation complexity of the iterative forward algorithm is moderate. The same accounts for the computational complexity, which is quadratic on the amount of model states.

Expected diagnosis capabilities and imperfections

Similar to the BN, the HMM model imposes some system structure, which in terms of diagnosis performance gives it a principal advantage over e.g. $\alpha-$count. This aspect has, however, not been evaluated in this work.

The structure of the HMM approach and hence, direct mapping to the diagnosis problem makes it a useful candidate mechanism for the study in Chapter 6. It will explore options of improving diagnosis robustness to quantified observation uncertainties.

Future work

An promising future work item is to study the dynamic variant of BNs called a Dynamic Bayesian Network (DBN) [101]. It enables to model system dynamics such e.g. as TCP congestion control and may, thus, lead to better diagnosis performance. A significant complexity increase is expected. In this setting, it will be highly interesting to study if the interplay between diagnosis and the decision component can help mitigate the impact of the increased complexity. The outlook is to apply partial and approximate solution methods for the DBN. It can enable the decision component to request the needed level of diagnosis accuracy and separation of different fault hypothesis given available remediation options and costs of making a bad decision.

Chapter 5

End-User Service Model

In this chapter a model of the time constrained reliable data transfer end-user service is introduced. The introduction of the model will provide background information on the end-user service functionality. More importantly, a central outcome is an end-user service model which can be integrated into the prediction model of the Decision Component for reliability analysis under unreliable observations and imperfect diagnosis. Initially, a background is provided on the assumptions of the prediction model. This background is shared by the modelling approach of the end-user service. Next, the model of the time constrained reliable data transfer end-user service is developed and finally, intermediate results on its performance are discussed.

5.1 Prediction Model Background

The Decision Component is the main control component of the ODDR. It interprets end-user service requirements from the application layer and transforms these into a series of decisions. The primary objective is to ensure high service reliability and secondarily attempt to minimize overhead from observations or unnecessary remediation actions. Which decisions to employ, in which order and when is not a trivial task to perform. A good decision depends amongst others on: the past and predicted development of the end-user service, observed and expected properties of remediation options, consequences of initiating a certain remediation action and the performance properties of the imperfect diagnosis component. Defining good heuristics to drive the decision process may be difficult due to the complex interactions between these aspects. In this work, it is studied how such good decisions may be derived from a model. This model is referred to as the prediction model. Its role is to predict the end-user service reliability given certain decision policies. Thus, it can be used to evaluate and identify best decision policies to apply.

The prediction model relies on a model representation of the end-user service to take certain reliability metrics into account given the end-user service requirements. In a prediction model this end-user service model is envisioned to be interchangeable to different end-user service types. In this thesis, however, focus is on a service model of the time constrained reliable data transfer end-user service. General prediction model parts concerning functions such as networks, fail-over and in particular diagnosis are presented in Chapter 7 and joined with the end-user service model. The resulting prediction model instance is in the remainder of this work referred to as the Policy Evaluation Discrete Time Markov Chain (PE DTMC) model.

General modelling considerations
In the context of the ODDR certain properties of the prediction model are desirable. Initially, the model must posses *sufficient qualitative behavior*. For the optimization of reliability parameters it is not always necessary for the model to be highly quantitatively accurate. More importantly, the model must be able to capture important system behavior that affects studied reliability metrics. Furthermore, it must provide sufficient qualitative behavior in comparisons between different policies to identify the most suitable. Good qualitative behavior is clearly a weaker requirement than good quantitative behavior. The advantage of accepting some quantitative inaccuracies is, however, to allow a potentially simpler model complexity. This supports another desirable property of a *lightweight model*. This property is particularly important in the online and dynamic setting of the end-node. If some parameters of the model change such as end-user service requirements, fault instance properties or diagnosis properties the policy evaluation could need to be re-invoked. In such cases simple lightweight prediction models are considered important in the ODDR context. In the modelling work of this thesis the primary aim is, however, to define initial models and provide insights into the imperfect diagnosis and reliability relations. In practice, this means that simple model constructs are pursued. However, the approach on how to solve the models fast and accurately in potentially low computational power devices is not addressed in details. Besides further model optimizations, viable approaches may be partial solutions, storage of existing solutions and sharing solutions in a distributed scenario of end-nodes. These challenges are left for future work.

A final aspect of the prediction model properties to consider is the ability to provide *adaptation*. A prediction model may need to change regularly due to for instance new parameters, newly appeared or disappeared access networks and different end-user service models. This adaptation requires model flexibility where parts of the model may be re-used, changes and extended. In this chapter the model parts are presented as such and options for adaptation and extension of the models presented here are discussed in Chapter 8.

5.2 Reliable Data Transfer Model

In this section the basic reliable data transfer end-user service model is defined with a starting point in TCP/SCTP modelling. To start out, two modelling approaches of different complexity to construct and maintain in the PE DTMC are proposed. Subsequently, the two approaches are parametrized and compared resulting in an approach where both models are applicable. It must be noted, that some presumptions are made about the overall PE DTMC model (including fail-over, diagnosis, etc.) at this stage. Details on the overall PE DTMC are, however, left for Chapter 7. As a final step, the data transfer model is extended to also include a model internal notion of time relevant for certain policies as will be shown later in this work.

5.2.1 End-user Service Characteristics

The studied time-constrained reliable data-transfer end-user service case has some characteristics making it worth considering for the service reliability studies. The reliability requirement is formulated as a probability that a data-transfer of a certain amount of data ($data_{size}$) is transferred within a given deadline requirement ($\gamma_{deadline}$). This probability is represented by Ω. In a traditional setting the QoS requirement formulation for such a service would be the *background QoS class* [1] as discussed in comparison to existing hand-over mechanisms. Formulating the requirement as a dependability requirement for this end-user service case allows a different system view compared to maintaining certain throughput, delay and jitter thresholds. The criticality level of the end-user service can be established by monitoring data transfer state and time to deadline. If criticality is low an option may be to wait for diagnosis to improve before reacting. It may also be chosen not to react to a diagnosed fault if a failure is not predicted to occur from it. These aspects are explored in the following chapter through the simulation model formerly introduced. It must be noted that the case of data upload from the end-node to the end-user service provider is considered.

The readily defined network states of Section 3.3 will form the basis for the modelling work in the following sections. Also, as previously, medium-sized data transfer of $data_{size} = 10\,MB$ is assumed as the general case. From these results an interesting setting of $t_{deadline} = 30\,s$ to apply in the following studies may also be defined. It is located between the normal and fault state distributions enabling some margin to optimize Ω.

5.2.2 Modelling of SCTP Behavior

The core aspect of the model is to describe the resulting data transfer completion time distribution for different model parameterizations and different decision policies. Given the requirement to minimize Ω (the amount of data transfers that complete within $t_{deadline}$) the aim is to establish the policy that moves most of the probability mass to or below $t_{deadline}$. Two Markov Chains model approaches are established to describe the data transfer progress and finally the data transfer completion time distribution: A) a basic discrete time Markov chain and, B) a complex discrete time Markov chain. Being basic, approach A) must show what can be achieved for simple model constructs with few states in contrast to B) which attempts to decode more system functionality at the cost of an increased state space. The models are depicted in Figure 5.1.

All models are based on a first order Markov Chain with transient states and a single absorbing state. The transient states represent the accumulated data transfer progress while the absorbing states represent the point at which the last byte of the total data amount has been transferred. Each state can be defined by the transferred data-range it represents as:

$$[..., (j-1)\frac{data_{size}}{n-1} \text{ to } j\frac{data_{size}}{n-1}, ...], \text{ where } j = 1...n \tag{5.1}$$

where n is amount of states. Solving these models in a transient analysis enables a distribution of the data transfer completion probability in relation to time. In the following these data transfer models are described in detail.

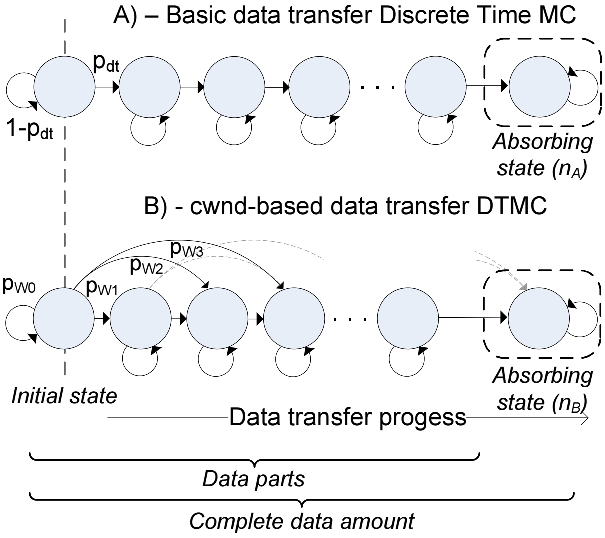

Figure 5.1: *Data transfer progress models.*

5.2.3 A) Basic Discrete Time Markov Chain Model

The data transfer progress may in a simple form be described as a discrete time birth chain with an absorbing state. The model has two parameters, which are p_{dt} the interstate transition probability and n_A the amount of states in the model. Considering that the inter-state transition probability is equal for all state transitions the resulting data transfer distribution formally complies to the Pascal distribution [118]. The model, thus, describes the distribution of the time at which the transition from state $N_A - 1$ to N_A is fired. Defining S as the random variable of the data transfer time, two relevant statistical parameters of mean and variance of the model distribution can be obtained from:

$$\mathrm{E}(S) = (n_A - 1)\frac{1}{\lambda_{dt}} \tag{5.2}$$

$$\mathrm{Var}(S) = (n_A - 1)\frac{1 - p_{dt}}{{\lambda_{dt}}^2}, \text{ where } p_{dt} = \frac{\lambda_{dt}}{\kappa} \tag{5.3}$$

κ is the rate of the discrete Markov chain and λ_{dt} the data part transmission rate. For this chain, clearly, the condition exist $\kappa \geq \lambda_{dt}$ and λ_{dt} is defined as:

$$\lambda_{dt} = (n_A - 1)\frac{\Lambda_{dt}}{data_{size}} \tag{5.4}$$

From these definitions the model is partially defined by Λ_{dt}, which is the mean SCTP goodput (application layer throughput). In the remaining part of this chapter the word *throughput* is used to represent SCTP goodput. Λ_{dt} may be obtained from existing work of TCP modelling [110] or similar modelling methods. It must be noted that SCTP and TCP transmission control algorithms have strong similarities. The model, however still has two free parameters: n_A and κ. In practice, these two free parameters control the distribution variance and consequently, how well the data transfer distribution is modelled. Defining them, is however, not straight forward as they both influence the entire PE DTMC model. κ defines the discrete rate for the entire model (including discrete functions like diagnosis), which introduces some limitations. Further, n_A defines the model state space, which also has an important impact on the product space of the PE DTMC. Defining good settings for these parameters is, clearly, a trade-off between model complexity and accuracy.

5.2.4 B) SCTP Congestion Window Based DTMC Model

A more complex version of the DTMC birth chain is the SCTP congestion window (cwnd) based model. It has the option to more accurately describe the data transfer completion time distribution by incorporating throughput variations determined by the SCTP transmission control mechanisms. Initially, a short summary on TCP/SCTP transmission control is presented followed by the model description. In the following a TCP/SCTP client with data to be sent is referred to as a *source* and a receiver client a *sink*.

Brief background on transmission control mechanisms
TCP and SCTP are both reliable layer 4 transport protocols featuring similar transmission control mechanisms. In short, the transmission control mechanisms throttle the transmission rate of the TCP/SCTP source to avoid network congestion. The throttling process is based on the estimated congestion in the network, which is judged by packet losses. In this respect the SCTP transmission rate and hence, the achievable throughput depends strongly on the packet loss rate. The throttling is based on a sender window of packets (congestion window), which is adjusted continually. The window defines how many unacknowledged packets the transmission control algorithm allows to be in transit. The window size control function can be in either of two phases *slow-start* or *congestion avoidance*:

Slow-start - The slow start phase is active in the beginning of a transfer or during a transfer if significant packet losses are detected; usually by a time-out on missing acknowledgments to sent packets. In the slow start phase the congestion window is initialized to an initial window size i.e. W_0. Every time a new acknowledgment arrives the window is increased by one. This continues until reaching the *slow start threshold* W_{sst} after which, the congestion avoidance phase is entered.

Congestion avoidance - Entering the congestion avoidance phase the source attempts to identify the highest possible transmission rate. The rate can be limited by: i) a maximum allowed window size (source setting) W_{max}, ii) a packet loss under the assumption of congestion, or iii) the receiver window. The latter is set by the sink, i.e. the end-user service provider, if it cannot process the data in the pace it is sent, however, it is not considered in this work. In the congestion avoidance phase the window increase rate is reduced to $1/W_i$ where W_i refers to the current window size when the acknowledgment to packet with sequence number i has been successfully received. In low loss scenarios the most typical packet loss detection method is *duplicate acknowledgments*. In this case a receiver keeps acknowledging the last successfully received packet (packet with sequence number $j-1$) before the lost packet j, based on subsequent received packets ($j<$). The source resends the lost packet (known as fast re-transmission) and reacts by reducing the window size to a half. For both the time-out and duplicate acknowledgment based packet loss detection mechanisms, W_{sst} is halved as well. This means the point where congestion avoidance is initiated, is reached earlier.

These are the general mechanisms of TCP/SCTP transmission control. Both transport layer protocols exist with different variants of the transmission control algorithm. This is especially true for TCP which exist in variants of Vegas [28], Reno, SACK and New-Reno. The latter is most widespread in current IP protocol stacks. The details on these will not be considered further in this work. More specific insights on SCTP and TCP transmission control mechanism can be found in the references [110],[45],[29][7].

Having defined the basics of transmission control mechanisms the cwnd based DTMC model can be introduced. Its starting point is taken in the majority of existing work on TCP transmission control modelling methods. The applied SCTP implementation of ns-2 has been adjusted to approach the TCP SACK variant. This must ensure a minimal divergence to SCTP from the TCP modelling methods. In brief, the following changes and configurations have been made: 1) when $cwnd = W_{sst}$ congestion avoidance is performed as opposed to slow start, 2) fast retransmissions are triggered after three missing packet reports as opposed to four which is typical in SCTP [29], and 3) SCTP specific features of heart-beats and retransmissions on a secondary path have been disabled. It should finally be noted that SCTP has been configured to only use one link at a time to obey a strict fail-over scenario.

Principles of the cwnd-based DTMC model

In the model of Figure 5.1, states represent data transfer progress starting from the initial state, which represents no progress (0 Bytes sent). Further, the transitions (p_{w0}, p_{w1}, p_{w2}, ...) represent different data transmission rates. The actual data transmission rates depend on the state of the transmission control mechanism and can be obtained as follows:

$$\lambda(i) = \frac{\bar{W_i} \cdot \texttt{Pdata}}{R\bar{T}T} \tag{5.5}$$

where $\bar{W_i}$ is the congestion window size during a mean RTT period $R\bar{T}T$, $\texttt{Pdata}$ is the application layer payload in a packet and $i, (1 \leq i \leq R)$ is an identifier for a mean window size. Both $\texttt{Pdata}$ and $R\bar{T}T$ are assumed to be constant for a given network state. $\bar{W_i}$ is clearly dynamic and the needed distribution over $\lambda(i)$ can be deduced from a distribution of $\bar{W_i}$.

Distributions of $\bar{W_i}$ as function of p_{loss} and W_{max} are obtained from an existing basic DTMC model proposed in [45] of the TCP *congestion avoidance* mechanism. This model is based on the following main assumptions:

Independent packet losses - To enable a simple model, independent packet losses with packet loss probability p_{loss} are assumed. This assumption may fit well for links where packet losses are caused by queues implementing the Random Early Detection (RED) mechanism [53], which ensures a fair impact of losses on different flows. In a drop-tail queue, implemented in the simulation setup, this assumption is compromised. Impacts of this assumption will be discussed later.

Only model the TCP congestion avoidance phase - for low loss scenarios ($p_{loss} < 5\,\%$) a majority of time is spent in the TCP congestion avoidance phase. Consequently, including the slow-start model and behavior of packet losses detected by time-out may be left out. Extending this model is necessary if considering higher loss probabilities. It must be noted that this assumption is valid only for longer data transfers where the obligatory initializing slow-start phase has an insignificant role on the data transfer completion time estimate.

Integer window size - The window size remains constant within each $R\bar{T}T$ and is as a simplification counted in integers representing amount of packets.

Recalling W_{max} is the maximum cwnd size and defining W_{bin} as a cwnd discretization bin size to control the modelled resolution of the congestion window values, the following vectors are defined:

$$\begin{aligned} G_r =&(0, \lambda(1), \lambda(2), ...\lambda(R)), \qquad\qquad G_p = (p_{W_0}, p_{W_1}, ...p_{W_R}) \qquad (5.6) \\ R =&ceiling\left(\frac{W_{max} - W_{bin}/2}{W_{bin}}\right) + 1 \end{aligned}$$

G_r are transfer rates and G_p are corresponding probabilities of having state transitions with rates in G_r, which are governed by the distributions of $\bar{W}_i$.

In practice, an upper bound on the achievable rate may exist due to a bottleneck link that has a sufficiently large buffer to avoid packet drops. In the case of the studied scenario this type of bottleneck occurs in the end-user node on the wireless link. It has the least bandwidth resources (see Table 3.2). As the TCP source passes the bottleneck rate (λ_{txmax}) the local link layer interface queue starts to build up leading to an increase in the experienced RTT rather than an increased data transmission rate. To compensate for this aspect in the proposed model the rate index is identified that most closely matches the maximum rate $q = min\{|\lambda(i) - \lambda_{txmax}|\}$. This step assumes that λ_{txmax} is available, which it may be from knowledge of the local link or information obtained from historical observations on a given link. Having quantified q, an update of the probabilities is performed, $G_p(q) = \Sigma_{x=q+1}^{R} G_p(x)$.

The amount of states in the *cwnd-based data transfer model* can finally be defined from:

$$\#states = ceiling\left(\frac{data_{size}}{W_{bin} \cdot \texttt{Pdata}/R\bar{T}T}\right) + 1 \qquad (5.7)$$

Solving the *cwnd-based data transfer model* by transient analysis at a discrete rate of 1, it is now possible to derive data transfer completion time distributions assuming that for low loss scenarios throughput ≈ transmission rate.

5.2.5 Model parameters and numerical evaluation

To provide initial results on the performance of the two models to describing the data transfer distributions they have been parametrized and compared to the fault/normal states that were previously defined. In relation to the basic DTMC model the cwnd-based DTMC model defines more meticulously the data transfer process. Thus, it is presented first as a baseline for the comparison to a simpler approach.

Results of the cwnd-based DTMC model

Practically, the cwnd-based DTMC model can be defined by two parameter sets. The first consists of `Pdata`, W_{max} and W_{bin} and represents, for this work, permanent model parameters. These are summarized in Table 5.1. `Pdata` is obtained from an assumed layer 2 frame size of 1500 bytes of which 38 bytes are MAC and IP headers. W_{max} defines the maximum window size in packets and is obtained from a typical limitation in TCP and SCTP that the maximum announced window size in bytes cannot be more than 65,535 bytes. Finally, W_{bin} has been defined empirically.

Parameter	*Value*
`Pdata`	1452 Bytes
W_{max}	46
W_{bin}	5

Table 5.1: *Permanent cwnd model parameter settings used in this work.*

The second set of parameters is obtained from the network state definitions being $\bar{RTT}$ and p_{loss}. $\bar{RTT}$ is estimated from low transmission rates $\lambda(<q)$ where the windowing dynamics have most influence on the transmission rate estimates. Thus, for both the normal and fault state parameterizations the fault state RTT mean estimate is applied. Referring to Table 3.3 the $\hat{RTT}$ observation is used ($53.3\,ms$). For the p_{loss} case the mean $\hat{PER}$ estimates are applied. The resulting model has a total of $n_B = 75$ states.

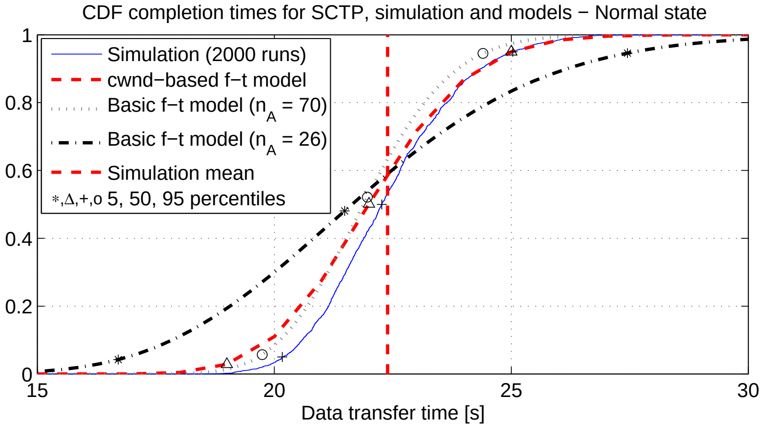

Figure 5.2: *Comparison of data transfer modelling methods in comparison to simulation results for a normal state.*

For the *normal* network state under *independent losses*, the capability of the cwnd-based DTMC model to describe the data transfer completion distribution is depicted in Figure 5.2. For now, the results of the basic DTMC model are disregarded. The CDF derived from the cwnd based DTMC model shows good resemblance to simulation results. An underestimation of $\sim 1\,s$ is observed for fast transfers around the $5\,\%$ percentile. Similar results have been obtained for the fault state (not depicted) where an understimation of $\sim 2.5\,s$ is observed. Three main reasons for this discrepancy have been identified. 1) The RTT used, $R\bar{T}T$, is estimated from the fault scenario (independent losses) where the transmission rate is low. This may give an underestimation of the RTT for large cwnd window sizes, which are likely in the normal state, and thus a more optimistic model. 2) The rate λ_{txmax} is only roughly estimated in the model with a rate resolution of $136\,KB/s$ for the normal state. The maximum rate $\lambda_{txmax} \approx 630\,KB/s$ where $\lambda(q) = 681\,KB/s$. Thus, the maximum throughput is slightly overestimated leading to the underestimation of the $5\,\%$ percentile. 3) The congestion window model is not accurate enough when loss rates increase (fault state) due to amongst other SCTP slow restart being unmodelled. A good resemblence between the congestion window model and observations of the SCTP window have, however, been established as seen in Figure 5.3.

From these results the cwnd based DTMC model seems useful to model the data completion time CDF for cases where the applied assumptions hold. Integral are the assumptions of independent losses and low loss scenarios in the area of $0 - 5\,\%$. In the following, the more basic birth chain is introduced to compare and identify the impact of a simpler and potentially smaller model in terms of state space.

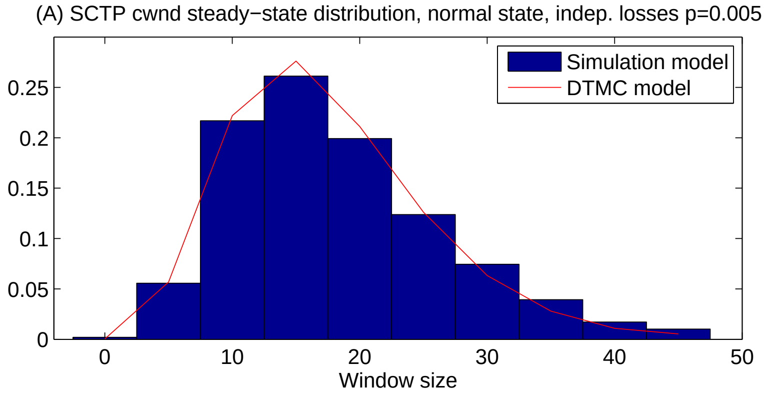

Figure 5.3: *Comparison of results of window size distribution in normal state based on simulated SCTP results and the TCP congestion control DTMC model obtained from the work in [45].*

Results of the Basic DTMC results

In contrast to the cwnd based DTMC model that models TCP/SCTP behavior directly from packet loss rates and RTT, the basic data transfer progress DTMC model needs to be parametrized from the expected mean application layer throughput. Table 5.2 depicts the mean throughput obtained for the simulation environment and for the cwnd-based DTMC model. As it can be seen there is only a small deviation from the simulated to the estimated throughput. Estimated mean throughput results have also been obtained from the well known TCP transmission control model by Padhye et al. [110] using the same input parameters as for the cwnd-based DTMC model. However, in relation to the considered simulation results, the Padhye model error has been found to be approximately 6%. The reason for this difference between the cwnd-based DTMC model and the Padhye model is not central for this work and has therefore, not been studied further. Consequently, the cwnd-based DTMC model is used to derive the mean throughput for the remaining part of this work.

The selection of the discrete time rate parameter κ has been done empirically to comply with the entire PE DTMC. Thus, the results obtained here are consistent throughout the remaining evaluations of this thesis. The approach has been to identify the state Q with the highest flow rate out of it. While details on the full model state space is reported later the rate is defined as:

$$\kappa = \lambda_{dtn} + \lambda_{tp} + \Lambda_f + 1 \tag{5.8}$$

where, λ_{dtn} represents the highest considered data transfer rate (in normal state), λ_{tp} is the time progress (introduced later in this chapter) and Λ_f the fault occurrence rate (which in subsequent studies is higher than the repair rate). Finally, *1* has been selected empirically as a self transitioning rate on Q but also offers headroom to change the current parameters when κ is fixed. This empiric approach has been useful for the results of this chapter, however, more robust approaches may be considered in future model iterations. Depending on λ_{dtn} to determine κ also the state space n_A should be determined. Using equations 5.3, 5.4 and 5.8 a good n_A has been found to provide a standard deviation of the Pascal distribution close to the sample standard deviation for the normal state data transfers used to obtain Figure 3.4. The sample standard deviation is $s = 1.48\,s$. The resulting parameter set is $(n_B = 70, \kappa = 4.51)$. This amount of states is close to the cwnd-based DTMC model state space. To understand if less accuracy can be sufficient, another parameter set is defined considering an approximate doubling of the standard deviation. This allows for the reduction of the needed states on n_B to a third. Thus, another state set is created $(n_B = 26, \kappa = 2.51)$. The results for these two parametrization sets are also depicted in Figure 5.2. It is observed that the parametrization with $n_B = 70$ provides a good characterization of the data transfer CDF, only shifted. The predominant part of this shift is caused by the small error of the cwnd-based DTMC model throughput estimation. The $n_B = 26$ parametrization shows as expected a somewhat higher variance. This parametrization is used in the following studies to understand if it is possible to gain from the reduced state space while still providing useful policy evaluation results. Similar results on the basic data transfer DTMC model on the fault state have been observed (not

State	Simulation (Λ_{dt})	cwnd DTMC (Λ_{dt})	Error
Normal	447	453	1.3
Fault	210	220	4.8
Units	$[KB/s]$	$[KB/s]$	%

Table 5.2: *Simulated mean transmission rate compared to rate obtained from cwnd-based data transfer DTMC (Each based on 2000 independent simulation runs of a data amount $data_{size} = 10\,Mbyte$).*

depicted). It must be noted that the full PE DTMC model will include both a fault and a normal transfer rate data transfer birth chain. Thus, to enable basic translation between these two, the fault state chain is fixed to the same parameters of ($n_B = 26, \kappa = 2.51$). Here, it must be noted that as the mean transmission rate is halved in the fault state the variance increases. This behavior corresponds well with the increased data transfer distribution times when comparing Figures 3.4 and 3.5.

While the cwnd based DTMC model has the potential to provide quite accurate estimates of the data transfer completion times, it requires a high amount of states. As mentioned previously, $n_B = 75$ states in the Markov chain are needed. Considering the alternative basic CTMC DTMC birth chain a realization of it with $n_A = 26$ states has been studied. The cost that must be accepted to obtain the smaller state-space is an increase in the variability of the data transfer completion time estimates. To parametrize the model from state space definitions by RTT and packet loss the cwnd-based DTMC model provides a valid approach for parameterizing the basic DTMC for the considered loss rates. It should be noted that the larger state space of the cwnd base model is no problem for the derivation of the mean transmission rate as the model is solved independently of the full PE DTMC (using transient analysis). It would have a significantly larger impact if it was to be integrated into the PE DTMC model due to its product space.

5.3 Including a Stochastic Clock

In terms of the introduced data transfer completion time models the notion of time exist as the model is solved using transient analysis. This is sufficient to relate the model solution to time. The model itself must, however, also keep an internal notion of time to enable policies based on criticality of the time progress in relation to the time deadline. Thus, the data transfer completion time model is extended to also include time. Modelling deterministic time progress in a stochastic DTMC is, however, not straight forward. The most basic approach is to create another birth chain and model each state as a time epoch. However, as a state is needed for every $\kappa^{-1} \approx 0.4\,s$ this is an expensive solution in terms of states. An alternative is to reduce the state space and accept that time is tracked by a stochastic clock. This approach has been adapted for model simplicity and is depicted in Figure 5.4. The horizontal direction represents the data transfer model introduced before. The vertical direction represents the time progress equally as a birth chain with an absorbing state. Reaching the absorbing state means that the deadline ($t_{deadline}$) has been exceeded and that the data transfer has failed.

The state space of the end-user service model depicted in Figure 5.4 is defined as $S_{EUS} = \{Tp,\, Dp\}$, where Tp=*Time progress* and Dp=*Data transfer progress*. Now as time is included in the model it can be solved using a steady state solution. Thus, we define the model based probability of a successful data transfer as $\Omega_{model} = \Sigma_{r=1}^{m} S_{EUSss}(r,n)$. S_{EUSss} is the steady state solution for S_{EUS}. It must be noted that time progress and data progress is modelled as independent events in the model. This has been done to maintain that the *true time* (not evaluated by the stochastic clock) distribution of Ω_{model} can be obtained from the model in a transient analysis. This feature has been used

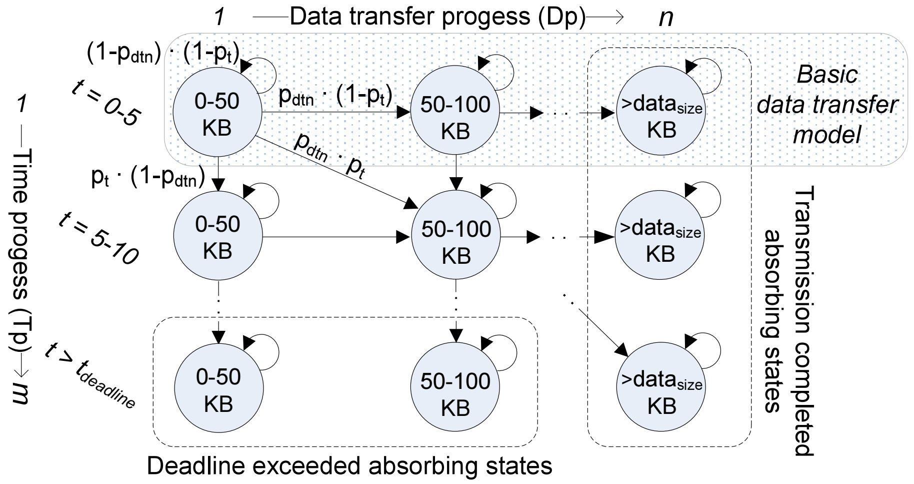

Figure 5.4: *Basic DTMC model representing data progress and time progress tracked by a stochastic clock. p_{dtn} refers to data transfer progress in a normal network state.*

for intermediate verifications. Effects of modelling data and time progress as interdependent events have not been studied further in this work.

Clearly, the approach to model time as a stochastic process in the model will lead to inaccuracies in Ω_{model}. This is demonstrated in an intermediate model result depicted in Figure 5.5 of normal state data transfer results. The time model has been fixed at $m = 10$ states. Considering the last state to be absorbing and $t_{deadline} = 30\,s$, each mean state holding time is $3.33\,s$. The upper part of the Figure depicts the discrete distribution over the time states where the data transfer is completed successfully. Notice, that only time until and including the $> 30\,s$ state is included in the model. It should further be noted that the final time state $> 30\,s$ is considered successful as the data transfer is completed on the transition to this state (see also figure 5.4). Comparing this distribution to the CDF in Figure 5.2 it is evident that the variability is significantly higher when using the stochastic clock. For instance, the simulation results show that 100 % of data transfers are completed within the $t_{deadline} = 30\,s$. However, the model predicts that only 79 % are completed. This shows that the model will lead to a quantitative deviation. It may, however, still be qualitatively correct enabling it to assess useful decision policies.

5.4 Conclusion

In this chapter a model of the time constrained reliable data transfer end-user service has been specified. Modelling data and time progress as birth chains leads to strong quantitative deviations from results obtained in a detailed simulation scenario of the end-user service. The qualitative properties of the model may, however, be sufficient for best policy evaluation. Final conclusions on this topic are obtained in Chapter 7.

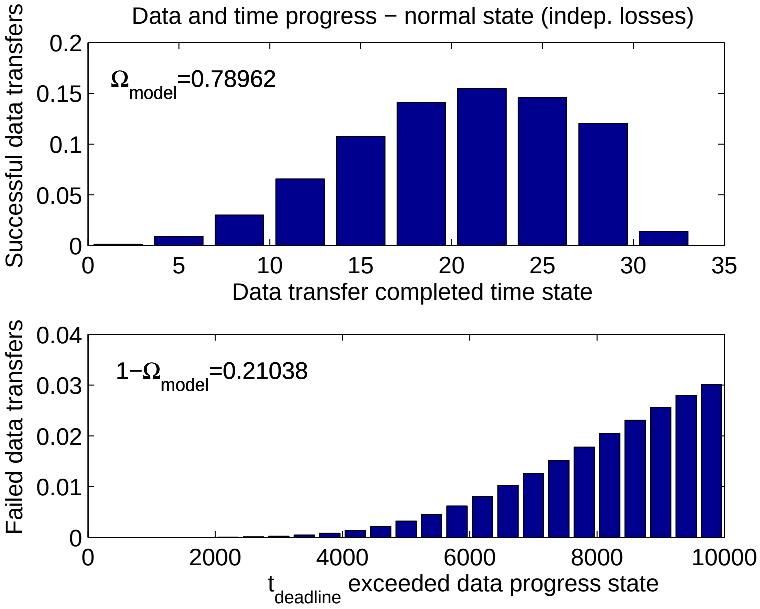

Figure 5.5: *Distributions over data and time absorbing states for a normal state based on independent losses.*

Future work

While the simple birth chain approach has been chosen, especially, leading to a stochastic clock with high variability other modelling approaches have been briefly reviewed. One is to use a *Two timescale* option where more than one epoch clock is used in the DTMC. This may be useful to slow down the stochastic clock to decrease variance at the cost of clock resolution. More on this option can be found in [140]. Another option could be to apply a semi-Markov DTMC to specify other state holding time distributions than geometric. Neither of these methods have been applied here but could be investigated in future work to minimize the time variability if needed.

Also, other end-user service model formulations may be considered. I.e. it would be relevant to consider if the model could be re-formulated to have regenerative properties allowing for the reduction of the current model state space.

Chapter 6

Improving Diagnosis Robustness by Uncertainty Estimates

A part of the unreliability of observations based on network traffic is caused by the monitoring process itself. Measured values that are impacted by non-negligible measurement errors, may lead to significant imperfections of the diagnosis estimates and thereby affect end-user service performance and reliability [67]. Overall, approaches are sought to improve the performance of diagnosis given unreliable and uncertain observations as in the work of [104], [124] and [39]. In this chapter, the disciplines of metrology and diagnosis are joined to identify approaches of improving robustness of system diagnosis to observation unreliabilities from the measurement process.

The body of knowledge of metrology (measurement theory [18]) proposes rules and practices to estimate and possibly mitigate measurement errors through measurement uncertainty [18]. Using approaches of metrology it is assumed that a measure of uncertainty can be provided in the network observation process to enrich measured point estimates. The aim in this work is to propose a diagnosis approach that can make use of such uncertainty knowledge to minimize diagnosis imperfections.

A starting point is made in the Hidden Markov Model formalism. Besides providing a formalized approach to specify the diagnosis problem HMMs are also applied and studied in other application areas such as speech recognition. In speech recognition uncertainty issues from noise have been addressed with good results [61]. In Section 4.6.2 the diagnosis problem has readily been formulated as a Hidden Markov Model. In this chapter, the resulting basic model is referred to as (H0). Based on (H0) two HMM model variants are further proposed: (H1) which assumes an accurate a-priori stochastic estimation of measurement uncertainty to update the model parameters, and (H2) which adapts to uncertainty in observations online by weighing low uncertainty observations over those with high uncertainty in the diagnosis process. The diagnostic performance of the three approaches is compared in the thesis case study for diagnosis of infrastructure network congestion faults based on network delay measurements. To examine how the three HMM diagnosis approaches cope with uncertainties two uncertainty injection scenarios are studied. These cover unmodelled disturbances and clock synchronization issues.

The subject of unreliable observations in HMMs for diagnosis has previously been studied in [39], [141] as an expected stationary coverage probability of conducted tests with a binary outcome. In this work, we extend this approach to handle uncertainty in a continuous observation variable (H1) and propose an approach to associate a different uncertainty to individual observations based on the measurement process (H2).

6.1 Uncertainty in Networking Scenarios

The analysis of including uncertainty in the diagnosis process is based on the generalized end-to-end scenario presented in Section 3.2.1. In this section example uncertainty cases are defined and analyzed. Following, details on the assumptions on the interactions between the ODDR OPP and diagnosis components are introduced.

6.1.1 Case Study View

The case study view taken in this chapter is depicted in Figure 6.1. It complies to the scenario presented in Section 3.2 but introduces some additional components. The following assumptions are made for this chapter. The remediation action is network fail-over. The main weight is, however, on the diagnosis process. Thus, the fail-over action and resulting reliability properties are not considered actively. To simplify the study a single fault is assumed. The fault is the *congestion* fault occurring in the infrastructure network.

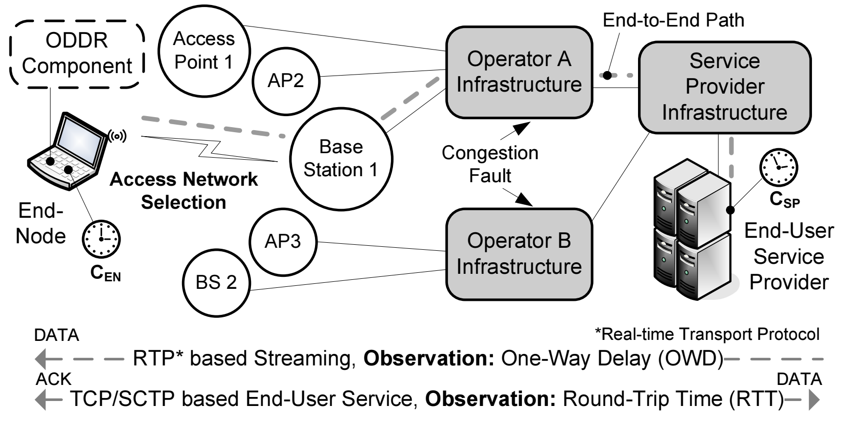
ODDR Component
Access Point 1
AP2
Base Station 1
AP3
BS 2
Operator A Infrastructure
Operator B Infrastructure
End-to-End Path
Service Provider Infrastructure
Congestion Fault
Access Network Selection
End-Node
C_{EN}
C_{SP}
End-User Service Provider
DATA
*Real-time Transport Protocol
RTP* based Streaming, **Observation:** One-Way Delay (OWD)
ACK
DATA
TCP/SCTP based End-User Service, **Observation:** Round-Trip Time (RTT)

Figure 6.1: *Scenario for diagnosis of congestion faults.*

Network Diagnosis Under Uncertainty

The level of network congestion can be observed through the network delay in the path [104], [93]. Depending on the type of end-user service, network delay observations can generally be obtained from *Round-Trip Times* as previously

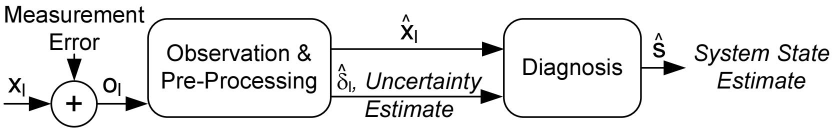

Figure 6.2: *Observation Pre-Processing and System Diagnosis process.*

discussed and *One-Way Delays* (OWDs). OWDs may be obtained in streaming traffic data cases by observing packet timestamps.

RTD and OWD observations have some inherent issues. 1) They provide indirect non-deterministic observations of the true hidden network state making diagnosis non-trivial. 2) The measurement process may, further, be exposed to additional errors leading to observation uncertainty. In this chapter, the focus is on uncertainties caused by: a) Unmodelled network disturbances (noise), as for example queuing delays [5], and b) clock drift and offset, referring in particular to synchronization uncertainty [22]. The first case would apply to both RTT and OWD observations. However, the issue of clock synchronization is most significant for OWD observations where two distributed local clocks (C_{EN} and C_{SP} in Figure 6.1) are used to provide the delay estimate. Clearly, for RTT observations obtained at the end-node only C_{EN} is required. As mentioned initially, other sources of measurement errors and uncertainty exist. However, the chosen are considered sufficiently illustrative in this study to understand how measurement errors and uncertainty based mitigation actions may affect diagnosis performance.

6.1.2 Observation Pre-Processing and Diagnosis

A principal outline of how observation uncertainty is considered in the OPP and Diagnosis components is depicted in Figure 6.2. An *error free observation* x_l represents a perfect measurement, with no uncertainty or undetected systematic errors, at time instant l. The *unreliable observation* o_l is obtained by x_l associated with possible measurement errors; o_l is what the OPP module observes from the network. The functionality of the OPP module is not the focus, but it is assumed to have the following properties:

Observation Compensation: The OPP may attempt to remove any sys-

Symbol	**Description**
$X = \{x_1, x_2, ..., x_L\}$	Error free observation sequence.
$\hat{O} = \{o_1, o_2, ..., o_L\}$	Observation sequence with measurement errors.
$\hat{X} = \{\hat{x}_1, \hat{x}_2, ..., \hat{x}_L\}$	Compensated observations with uncertainty.
$\hat{\delta} = \{(\hat{x_1^{low}}, \hat{x_1^{up}}), ..., (\hat{x_L^{low}}, \hat{x_L^{up}})\}$	Estimate of uncertainty expressed by tuples of upper and lower bounds ($\hat{x_l}^{low} \in \hat{X}^{low}, \hat{x_l}^{up} \in \hat{X}^{up}$).

Table 6.1: *Definition of observation sequences.*

tematic error in o_l to provide an observation point estimate $\hat{x}_l$ [18]. In cases where no compensation is made it is assumed that $\hat{x}_l = o_l$.

Uncertainty Estimate: $\hat{\delta}_l$ represents an estimate of residual uncertainty associated to $\hat{x}_l$. Thus, the OPP provides in discrete steps l (with period T) the measure $\hat{x}_l$ and $\hat{\delta}_l$ as input to the Diagnosis Component. The Diagnosis component may then use both the collected measurement and its uncertainty to estimate the network state.

The expected properties of observation uncertainty $\hat{\delta}_l$ depend strongly on the confidence level required and on the OPP module ability to provide an estimate of uncertainty. Two different assumptions are identified, namely *Confidence Bounds* (CB) and *True Value Interval* (TVI). CB represent probabilistic bounds, typically, considering a confidence level less than 1, where it is possible that the true value of a measurement result may lie outside the uncertainty bounds. CB could be obtained from noise estimation techniques or a model of the residual uncertainty given some *observation compensation* approach [90]. TVI represents bounds, which promise to contain the true value with probability 1 (may be considered a sub-class of the CB bounds). Such bounds may be obtainable from a system model of the uncertainty factor as in [22], where bounds are derived for synchronized clocks in a networking system.

To simplify the coupling between the OPP and the system diagnosis module the weakest possible assumption is made of a uniform probability distribution, i.e. $p(x_l|\hat{x}_l) \sim U(\hat{x}^{low}, \hat{x}^{up})$ within the bound. Stronger assumptions may be made when the mechanisms and properties of the OPP are well known. In this general study this is not the case, thus, these considerations are left for future work. A summary of the observation sequence definitions is provided in Table 6.1. Other symbol definitions relevant in the following sections have been introduced, previously, in Table 4.4.

In the following section, diagnosis components are proposed using processed observations from the OPP. In section 6.3 the properties of the diagnosis components are compared for the CB and TVI bounds.

6.2 HMM Diagnosis Models

A background and initial results on a realization of a HMM used for diagnosis have already been introduced in Section 4.6.2. The resulting diagnosis mechanism is defined as (H0) or the *Basic HMM*, which by definition considers point estimates, $\hat{X}$ for diagnosis, i.e. the a posteriori marginal distribution is given by $\beta(i) = P[q_l = s_i|\hat{X}]$. In this section two variants of (H0) are proposed to compensate for unreliable observations. (H1) is an extension of (H0) where it is compensated statically under the assumption of an accurately specified a-priori model of the measurement error contribution. (H2) represents an observation uncertainty model variant using the uncertainty bounds derived by the OPP.

6.2.1 (H1) - Compensated Basic HMM

This variant is a basic extension of (H0) under the assumption that an accurate a-priori stochastic model of the uncertainty can be provided and used to update B. The uncertainty model is represented by the distribution: $P(\hat{x}_l|x_l)$.

This conditional distribution describes the probability that due to an error contribution an initial error free observation x_l is observed as $\hat{x}_l$ in the HMM. In reality, it may be difficult to obtain this distribution (e.g. by online estimation). Hence, (H1) just provides a comparison basis to the uncertain observation method presented in (H2). For (H1) the observation distributions can be updated as: $B_{(H1)} = B \times C$, where C is the compensation uncertainty matrix defined as: $C = (c_p(k))_{p,k}$ and:

$$c_p(k) = P[\hat{x}_l = v_k | x_l = v_p], \quad 1 \leq p, k \leq M \tag{6.1}$$

These efforts makes C similar to the *translation probability matrix* described in [39] as a mean to define coverage of a given observation.

As the observation distribution model $B_{(H1)}$ is more accurate to the true observation distribution than B in (H0), improvements in diagnosis performance should be achieved for uncertain observations. In the considered form (H1) requires a model update and a complete specification of $P(\hat{x}_l|x_l)$, which is not assumed to be available from the OPP module. Thus, (H1) just provides a comparison basis to the uncertain observation approach presented in (H2).

6.2.2 (H2) - Dynamic Discretization HMM

The (H2) variant is studied to make use of the bounds provided by the OPP module. (H2) dynamically encodes uncertainty of observations into the symbol probability distribution in B. In practice B is updated based on the uncertainty estimates provided by the OPP for each observation in $\hat{X}$. Therefore, the observation received from the OPP module at time step l is $(\hat{x}_l, \hat{x}_l^{low}, \hat{x}_l^{up})$. The approach is based on a dynamic adaptation of the discretization of the continuous observation random variable χ (see Figure 6.3). The continuous observation symbol probability distribution can be specified: $\tilde{b_j}(x) = f(x|q_l = s_j)$. These (continuous) symbol probability distributions form a model of the true distribution of the continuous observation value. They will not be used directly in the HMM model. Instead, a discrete symbol observation probability distribution is defined based on a *new* discrete alphabet to be used in the HMM (see Figure 6.3): $V_{H2} = \{v_1 = \Gamma, v_2 = \Sigma^{low}, v_3 = \Sigma^{up}\}$. The distribution of this alphabet is described by a time varying observation symbol distribution matrix $B(l) = (b_j(k,l))_{j,k}$. The principle is now as follows: it is assumed that the observed symbol always is Γ. Now, the likelihood of observing Γ for state j is in each time step updated as:

$$b_j(1,l) = P[\hat{x}_l^{low} \leq \chi \leq \hat{x}_l^{up}] = \int_{\hat{x}_l^{low}}^{\hat{x}_l^{up}} f(x|q_l = s_j)dx \tag{6.2}$$

It is considered that the true value (x_l) of the observation with uncertainty $\hat{x}$ is located within the bounds for some confidence level P (or outside as marked by example points ϵ^{low}, ϵ^{up}). Further, it is assumed that the true value is anywhere in the interval with equal probability (uniform distribution) as assumed for the OPP.

Intuitively, (H2) can be related to (H0). In a normally formulated setting i.e. (H0), $b_i(k)$ is obtained as follows: the observation x_l discretized to v_k has been made at time step l. What is then the likelihood that v_k was emitted from the system in state s_i? In the setting formulated under (H2) the problem is

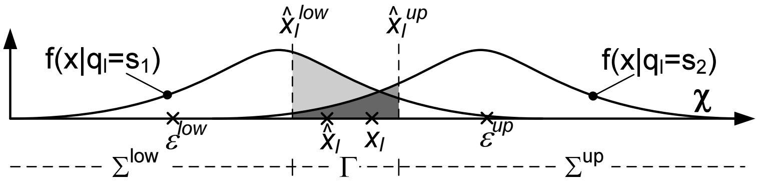

Figure 6.3: *Example distributions of observations for two different system states.*

considered in the following manner: The observation with uncertainty $\hat{x}_l$ has been made at time step l. It is assumed that its true value is equally likely to be anywhere in the interval between $\hat{x}_l^{low}$ and $\hat{x}_l^{up}$. What is then the likelihood that it was emitted from the system in state s_i? Depending on the location of the bounds and their range, the relative likelihood between states in S change. Note, as bounds increase the likelihood increases that the unreliable observation could have been emitted by any of the states. This makes the update on $\beta(i)$ less significant.

6.3 Comparison of Diagnosis Models

To understand the overall properties of the proposed diagnosis models, in this section the proposed models are compared in two different uncertainty scenarios. They represent unmodelled network disturbances on RTT measurements and clock synchronization issues on OWD measurements.

6.3.1 Scenario Setup, Metrics and Model Implementation

Error free observation traces X and traces with uncertainty $\hat{X}$ for diagnosis are obtained from the ns-2 based simulation results of RTT observations an SCTP data transfer (see Section 4.6.2). In addition, statistics from the two networks states of the congestion level are obtained from the distributions of Figure 3.6. Recall, that these are based on mean RTT observations from a window of a fixed size $\omega_{RTT} = 0.3\,s$. These state definitions will be used for model parametrization. Further to govern the ON-OFF network state process and diagnosis period T, parameters of Table 4.5 are reused. To obtain OWD traces, in practice, the actual RTT observations are utilized. In relation to RTT observations, this corresponds to assuming a longer end-to-end path for OWD observations. This step ensures consistency between diagnosis results under uncertainties affecting RTT and OWD, and that only one set of the diagnosis models is needed. Further, this simplification is not expected to have a significant impact on the overall conclusions. For simplicity, uncertainty is added directly to mean estimates obtained from the RTT observation windows, although in practice it would apply to the individual delay observations. It must be noted that the remediation action execution is not studied in this analysis. Thus, diagnosis is only considered on a single active end-to-end connection where the data transfer is being executed.

Comparison Methodology and Metrics

The diagnosis components are compared using 2000 independent simulation runs (provides for $CI = 95\%$ an error bound of $\pm \sim 2\%$ for $p_{RTA} \approx 0.5$). Each simulation run starts the network state process in a normal state and lasts for $300\,s$. To characterize the diagnosis performance metrics describing the trade-off between accuracy and timeliness of the diagnosis case are sought. As shown in Section 4.6, the metrics TNR/TPR may not be a good representation of the performance of temporal diagnosis approaches. Instead, the more intuitive *Mean Fault Reaction Time* (μ_{FRT}) and *Probability of Remediation on a True Alarm* (p_{RTA}) are taken into consideration. Recording in a simulation run the first reaction time observed (for the first correctly diagnosed fault occurrence) μ_{FRT} is the mean reaction time over all simulation runs. Registering in a simulation run if the first alarm raised ($\hat{s}_{fault} = 1$) is true or false, p_{RTA} defines the fraction of simulation runs that leads to a true alarm. The selected diagnosis performance metrics may not uniquely describe all characteristics of the diagnosis component. However, they are considered sufficient for these initial comparisons. A more thorough discussion of such metrics can be found in Section 7.4.1.

HMM model configurations

The two state fault model observation distributions of Figure 3.6 are used to parameterize the diagnosis models as in the example study of (H0). It must be noted, that for all three model realizations missing observations are included in the observation alphabet. It is in this relation assumed that missing observations are not affected by uncertainties introduced in the uncertainty injection studies.

(H1) Setup - To define C, the distribution $P(\hat{x}_l|x_l)$ is discretized into $\{v_1 \ldots, v_M\}$. This is conducted in correspondence to the symbol alphabet of (H0). Observations which fall below the discretization bin corresponding to (v_1) are interpreted as v_1. Observations above the bin v_{M-1} are interpreted as v_M. Recall, that the bin v_M represents $o_l^{RTT} \geq 80\,ms$.

(H2) Setup - The (H2) approach applies in Equation 6.2 sample distribution CDFs of the observations made to obtain the distributions in Figure 3.6. To deal with sample distribution tails that contain limited observations Pareto tails have been fitted to the upper 4% of all observations (highest RTTs). This fraction has been found empirically to provide best diagnosis outcomes. In uncertainty cases where $\hat{x}_l^{up}$ is lower than any observation in the sample distributions, the observation is simply ignored. In the following sections, the defined realizations of the HMM based diagnosis mechanisms are compared in the two uncertainty scenarios.

6.3.2 Uncertainty: Unmodelled Network Disturbances

In this uncertainty scenario the effects of unmodelled network disturbances (noise) are studied on diagnosis performance. Simulation traces of o_l are generated by adding synthetic Gaussian noise to error free observation traces x_l. Gaussian noise is expectedly not highly representative of common network delay disturbances. However, it is useful as a starting point to get valuable insights into the impacts on the diagnosis components.

To parametrize (H1) by defining C it follows that $\hat{x}_l = x_l + N(\mu, \sigma^2)$. For the uncertainty bounds based diagnosis model, in (H2), probabilistic Confidence Bounds are considered based on $\hat{x}_l \pm z\sigma$ where σ is the standard deviation (which we assume to be accurately estimated by the OPP) and z a factor controlling the bound confidence level.

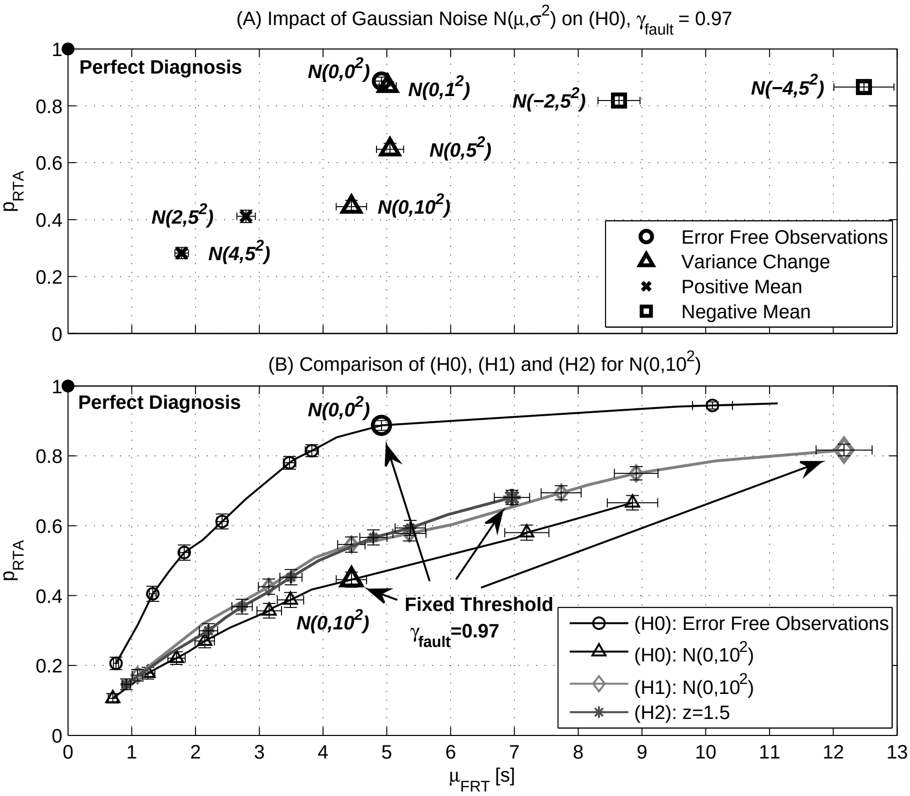

Figure 6.4: *(A) Impact of Gaussian noise on (H0). (B) Comparison of diagnosis trade-off for (H0), (H1) and (H2) for* $N(0, 10^2)$

Figure 6.4 depicts noise impact results and potential diagnosis trade-off options for the different diagnosis components given μ_{FRT} and p_{RTA}. A perfect diagnosis component would enable diagnosis at $\mu_{FRT} = 0$ and $p_{RTA} = 1$; In the imperfect diagnosis cases, characterizing this study, (H0) under error free observations ($N(0, 0^2)$) is defined to represent the best achievable trade-off performance. Trade-off options are given by varying the fault state threshold γ_{fault} of Equation 4.4 in the range $[0.1 \ldots 0.999]$ to characterize the overall capabilities of the diagnosis component. For a practical setup, however, it may not be trivial to adjust this threshold dynamically to maintain some given trade-off. Thus, we also study how a fixed threshold, set at design time for (H0) under error free observations, will affect the diagnosis performance. A fixed threshold of $\gamma_{fault} = 0.97$ provides an example trade-off favoring a high p_{RTA} without a drastic increase in μ_{FRT}.

Considering the different noise conditions in Figure 6.4 (A) on (H0), for increasing variance of the observation noise the probability of a true alarm decreases dramatically (while the probability of a false alarm increases accordingly). This occurs as in a normal state, noise will as one effect cause low RTT observations to become high making the HMM more prone to diagnose a fault state. Interestingly, μ_{FRT} is not strongly affected.

Focusing on changes in the mean, clearly the diagnosis performance is highly sensitive. For $N(2, 5^2)$ performance in p_{RTA} drops below $N(0, 10^2)$ while gaining slightly in improved reaction time. For $N(-2, 5^2)$ μ_{FRT} increases while helping to improve p_{RTA} by nearly 20 percentage points over $N(0, 5^2)$. Moving to the case of $N(-4, 5^2)$ the reaction time is so high that the considered fault cannot properly be diagnosed. This is in practice seen as 1.3% of the simulation traces no longer lead to alarms (although faults appear in all traces). The impact of these degradations in p_{RTA} and μ_{FRT} depend on the end-user service. Later in this thesis when evaluating best imperfect diagnosis settings a sensitivity analysis of the end-user service case reliability to diagnosis performance metrics is conducted. It shows that a degradation of p_{RTA} is most severe and that values below ~ 0.5 will lead to worse service reliability than if not fault management is made at all. An increase in delay similarly means that potential gains of timely remediation become insignificant. More details can be found in Section 7.4.2.

Figure 6.4 (B) shows how (H1) and (H2) perform compared to (H0) considering $N(0, 10^2)$. In the trade-off curve, neither (H1) nor (H2) come close to (H0) under error free observations. Yet, they do offer a similar improvement over (H0). Looking at a fixed state threshold, however, it is clear that (H1) and (H2) perform quite differently. Seemingly, (H1) makes use of the accurate information of the noise to maintain a high p_{RTA} but at the cost of an impractically high μ_{FRT} (in this case nearly the same as the mean fault duration). Compared to (H1), with (H2) a drop in p_{RTA} must be accepted, but at a significantly smaller cost in increased μ_{FRT}. These results are also largely consistent for other noise levels and settings of γ_{fault} not shown here. The studied setting of $z = 1.5$ (Confidence Level of 86.6%) has empirically been identified to provide the best trade-off curve for (H2) under $N(0, 10^2)$ (and $N(0, 5^2)$ as well). An optimum exists as a too small confidence interval would be less likely to contain the true value. Further, a too large interval would make the observation less likely to belong to one hidden state over another (see Figure 6.3). This would lead to long reaction times and diagnosis primarily based on the hidden model behavior in A and missing observations.

6.3.3 Uncertainty: Clock Synchronization Uncertainty

In this section the sensitivity of the (H2) approach to different properties of TVI bounds are studied. As (H1) is not using bounds, it is not evaluated in this section. Practical results of uncertainties due to clock synchronization errors on OWD are further provided. It is shown how (H2) performs by use of statistically estimated bounds on the clock drift and offset.

Figure 6.5 (A) depicts synthetic TVI results considering varying intervals which are symmetric and asymmetric in relation to x_l. The time value refers to the range of the bounds interval ($|\hat{x}_l^{up} - \hat{x}_l^{low}|$). Percentages define how bounds are shifted in relation to x_l. It is observed that as bounds increase from $1\,ms$ to $30\,ms$ for both symmetric and asymmetric cases, μ_{FRT} increases while p_{RTA} drops. For asymmetric bounds of 30%-70%, the bounds include more of the OWD tail behavior favoring the fault state distribution mass. This leads to fast μ_{FRT} and a small penalty of p_{RTA}. The opposite is the case for 70%-30%

bounds maintaining a high p_{RTA} at a high penalty on μ_{FRT}.

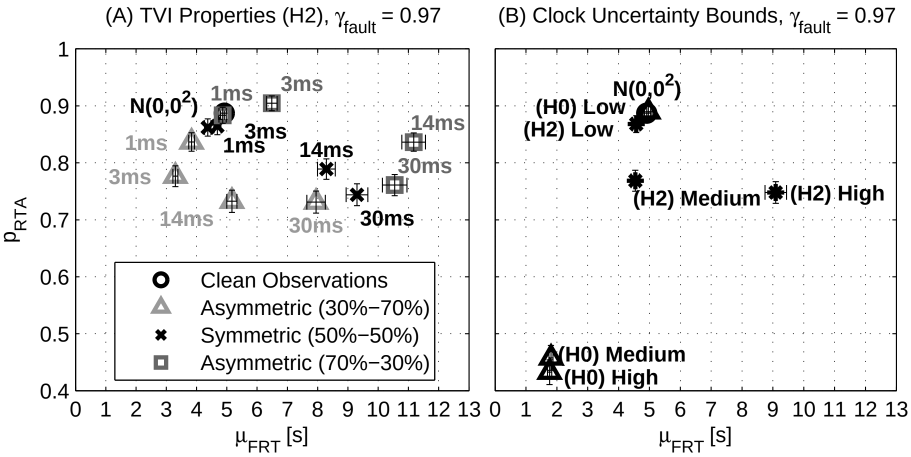

Figure 6.5: *(A) Example, and (B) R&SAClock based True Value Intervals.*

These results are put into practice by studying a real measurement based clock drift experiment. Uncertainty bounds are obtained from the Reliable and Self-Aware Clock (R&SAClock) being a software clock self-aware of its synchronization offset from the reference time. Its internal algorithm considered is presented in [22]. It uses a statistical model of the evolution versus time of clock offset and drift to provide a time value, $\hat{x}_l$, and associated information on the uncertainty to provide $\hat{x}_l^{up}$ and $\hat{x}_l^{low}$. In an experiment considering poorly synchronized clocks, three traces (of $300\,s$ duration) have been selected to exemplify periods with *low*, *medium* and *high* uncertainty (see Figure 6.6). In relation to the scenario, the reference time is defined to belong to the C_{SP} where the relative drift of C_{EN} is considered (see Figure 6.1). The drift offset from the reference time has been added to the error free observations of OWD to get o_l.

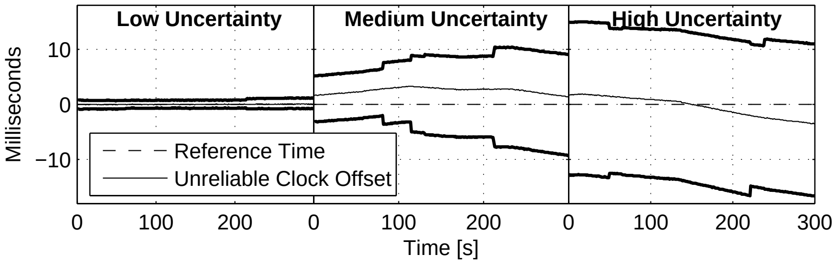

Figure 6.6: *Clock Uncertainty Traces.*

Results comparing (H0) using the drifting clock point estimate $\hat{x}_l$ and (H2) using the bounds are depicted in Figure 6.5 (B). For consistency to Figure 6.4 the exact clock value results (error free observations) for (H0) are still referred to as $N(0, 0^2)$. For a fixed state threshold ($\gamma_{fault} = 0.97$) it is clear that (H0) suffers from the clock offset in both the *medium* and *high* uncertainty cases leading to a low p_{RTA}. As the offset is primarily positive from the start of the trace (where the diagnosis process is initiated) the situation corresponds to mean shifts similar to $N(2, 5^2)$ in Figure 6.4 (A).

Now considering (H2) for *medium* uncertainty in Figure 6.5 (B) a significant improvement in p_{RTA} is observed without leading to a higher cost in μ_{FRT} compared to $N(0, 0^2)$. Looking at the corresponding clock trace in Figure 6.6 it is seen that the lower bound is closer to the reference time value. This corresponds to the 30%-70% asymmetric bounds investigated previously which favor a low μ_{FRT}. For the *high* uncertainty case the p_{RTA} level is also high. However, the large bounds lead to a significant increase in μ_{FRT} as expected. It must further be noted, that (H2) for small bounds performs similarly to error free reference clock observations. These results show that (H2) for this case of a fixed state threshold can provide a significant diagnosis performance improvement. It must, however, also be noted, that (H0) for other settings of γ_{fault} could provide an improvement over (H2) for *medium* and *high* uncertainty bounds (not depicted). In future work, approaches may be sought to adapt this threshold dynamically.

6.4 Conclusion

Performing robust network fault diagnosis based on unreliable observations from network traffic is challenging. Sources of uncertainty may degrade diagnosis timeliness and accuracy impacting reliability and performance of distributed services. In an attempt to improve robustness to uncertainties a new approach has been proposed in network diagnosis to enrich observations by quantifications of measurement uncertainties in a basic Hidden Markov Model (HMM). The properties of the approach have been assessed through three HMM model variants: (H0) representing a classical HMM formulation, (H1) a static compensation of the (H0) model by an accurate a-priori model of the measurement uncertainty and (H2) the alternative HMM formulation utilizing uncertainty enriched observations. To establish how the proposed diagnosis components react to uncertainties a simulation study has been conducted. The study considers diagnosis of a network congestion state fault in an end-to-end connection based on delay measurements. It consists of uncertainty injection scenarios covering unmodelled disturbances (noise) and clock synchronization issues. The diagnosis components have been compared considering a fixed configuration (a fixed threshold on the estimated fault state probability) and varying configurations (varying thresholds) to reveal their trade-off capabilities of mean fault reaction time (μ_{FRT}) and probability of correct diagnosis (p_{RTA}).

Overall, the studied HMM variants have been found to perform very differently. (H0), representing no uncertainty compensation, is as expected highly sensitive to observation noise mean and variance changes. In an increased variance scenario, both the (H1) and (H2) offer similar improvements in the diagnosis trade-off options. For a fixed state threshold configuration (H1) manages to maintain a high p_{RTA} but at a cost of an impractically high μ_{FRT}. In the same setting (H2) provides a more balanced impact on both performance metrics while still improving p_{RTA} over (H0). Using realistic drifting clock traces and corresponding varying uncertainty bounds for a fixed configuration (H2) manages to significantly improve p_{RTA} over (H0) with no or little increase in μ_{FRT} compared to perfect clock measurements.

Future work

The obtained result suggest, that using uncertainty bounds can provide a structured approach to improve diagnosis robustness for an online diagnosis component experiencing varying uncertainty of observations. In future work, it is relevant to study alternative approaches to the (H2) heuristic considering: 1) the impact of the assumption made on the distribution within uncertainty bounds, 2) an (H1) based observation distribution model update approach using uncertainty bounds and, 3) approaches to dynamically adapt fault state thresholds. Also, more realistic network uncertainty injection scenarios remain to be scrutinized. Finally, the impact of diagnosis performance differences must be assessed on the reliability of different end-user service types.

Chapter 7

Ameliorate Service Reliability under Imperfect Diagnosis

After having characterized diagnosis and means to improve its robustness to measurement error and observation uncertainty, this chapter considers which options exist to mitigate unavoidable diagnosis imperfections in the decision process. Thus, in the context of the ODDR framework the interplay between the Diagnosis and Decision components is studied. The aim is to clarify how good decision strategies may help to optimize end-user service reliability and how to configure the diagnosis component considering the imperfection trade-off options of a given diagnosis component.

In this chapter a complete prediction model is developed. In addition to the previously proposed end-user service model the prediction model consists of several functional parts including: time, networks, the fail-over process and a parsimonious diagnosis model. Based on the prediction model two separate studies are conducted: i) a policy evaluation study is made to assess best decision policies given diagnosis imperfections. This study includes modelling of basic memory-less diagnosis behavior. ii) Focusing on the proposed parsimonious diagnosis model it is studied if it can also capture important diagnosis performance properties of temporal diagnosis mechanisms. This resulting model improvement is finally considered to identify best settings of the α-count heuristic for end-user service reliability optimization.

7.1 Introduction of Policy Evaluation Model

A manual approach is taken to construct a prediction model for determining best decision policies under imperfect diagnosis in the case of fault remediation for the reliable data transfer end-user service. In this section the individual model parts are presented and a joint model view is established. Recall that this model is referred to as the Policy Evaluation Discrete Time Markov Chain (PE DTMC) model.

In Chapter 5 the end-user service model of the time constrained reliable data transfer has readily been introduced. Recall, that it integrates both data progress and time progress. In this section other general model functions are introduced with particular focus on capturing performance of the diagnosis process. Thus, the PE DTMC model is in summary developed to include the following functions:

End-user service model - Time constrained reliable data transfer (see Chapter 5).

Imperfect diagnosis - Models true and diagnosed (estimated) network state behavior.

Networks - A model of the actively used access network and an additional access network available for remediation.

Network fail-over and protection schemes - The network fail-over process is modelled to enable aspects of a failing fail-over and penalty in fail-over time. This part enables assessment of different protection schemes, which may be available in the access network change process.

These model parts are presented individually and subsequently, the integrated PE DTMC model is defined including the end-user service model. To focus the subsequent studies, diagnosis is delimited to estimate network congestion level states in the infrastructure network.

7.1.1 Parsimonious Diagnosis Model

In this section a general diagnosis model is introduced. As previously emphasized it must be capable of capturing diagnosis performance including its imperfections while being light-weight to pursuit a small state space.

Network states and diagnosis capabilities are modelled by a four-state parsimonious Markov model depicted in Figure 7.1. The network states (Ns) are vertically oriented labeled Normal (N) and Fault (F). The diagnosed network states (Ds) are horizontally oriented representing the estimates $\hat{N}$ and $\hat{F}$. When the estimated state and the actual state are equal the estimate is *True (T)* and *False (F)* otherwise. A fault state estimate corresponds to a *Positive (P)* and normal state estimate to a *Negative (N)*. Thus, the states are named s^{TN}, s^{FN}, s^{FP} and s^{TP}. Transition probabilities between states are named $p_{X|Y}$, where Y describes the state from which the transition initiates and X the transition target state. *Notice*, to avoid complicating this notation further, the s-marker (e.g. s^{TP}) is not used in these probability labels. Further, the labels should not be confused with conditional probabilities.

In the model, the previously defined diagnosis metric of a False Alarm (FA) is given by the transition from state s^{TN} to s^{FP} or state s^{FN} to s^{FP}. Further, Reaction Time is observed as the time from a fault occurrence until a true positive (true alarm) diagnosis in the s^{TP} state given by s^{TN} to s^{TP} or s^{FN} to s^{TP}.

As described in Chapter 4, the diagnosis component operates on a periodic basis with the period T. This corresponds well to the discrete time model. By defining $T^{-1} = \kappa$ (see Section 5.2.5) the DTMC and simulation model versions of the diagnosis component are using the same diagnosis rate. Transitions between true normal and fault states are given by the ON-OFF fault model introduced in Section 4.1 with geometrically distributed state-holding times and transition probabilities: p_{fault} (p_f) and p_{repair} (p_r). In the basic four state model, a total of 16 parameters exist to be defined. However, assumptions can

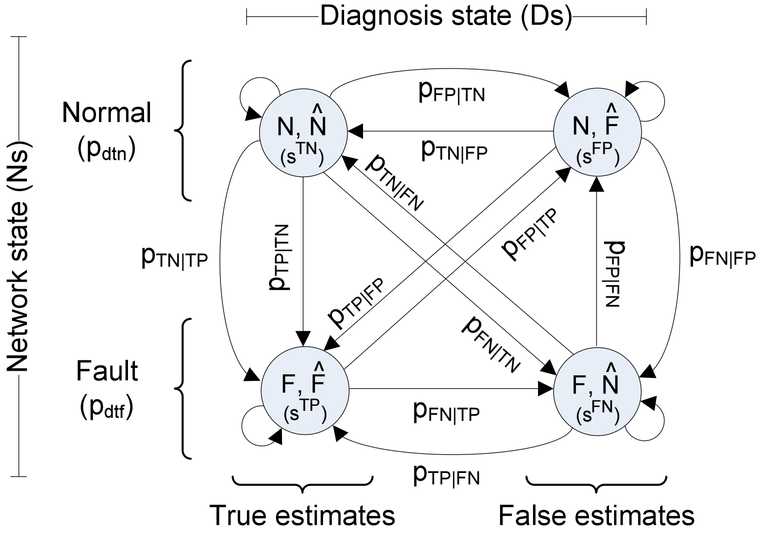

Figure 7.1: *DTMC model of fault occurrence and diagnosis.*

be made to reduce the amount of free parameters. In this work two parametrization approaches are considered: The first is considered in Section 7.3. It utilizes an independence assumption of diagnosis estimates accounting for the introduced memory-less diagnosis approaches. The second is introduced in Section 7.4. It relaxes the diagnosis estimate independence assumption to also represent temporal diagnosis model approaches. The applicability of the introduced parsimonious diagnosis model will be discussed in more details.

Integration with time and data transfer model part

The diagnosis model and the end-user service model are integrated as the network states (Ns) control the rate of transition in the data transfer birth chain (Dp in Figure 5.4). p_{dtn} is the transition probability for the normal state data transfer progress and p_{ftn} for the fault state. It should be noted that all state transition events, i.e. network state change, diagnosis, data transfer progress and time progress, in the overall Markov model are independent and may occur concurrently at a time epoch.

7.1.2 Additional Model Functions

Having established the basic model functions of the end-user service and network state diagnosis, in this section further model functionality is added to cover two networks A and B, the fail-over process and policy support.

Fail-over to other Network

The model contains two network stereotypes to contain the functionality of the fail-over option and properties associated to performing the fail-over. These are a network in which the data transfer is, initially, performed and a second network offering the remediation option. Seen from an initial condition of always starting the transfer in *network A*, *network B* is the remediation network.

To handle the two network stereotypes the model is extended by adding another state space variable called Nw. Nw has three values, namely *network A*, *network B* and *Intermediate*. This corresponds to having three instances of the state space: $\{Tp, Dp, Ns, Ds\}$ (where Tp and Dp are time progress and data progress, respectively). The first two instances describe the operation in network A and B, while the *intermediate* states are used solely to handle the fail-over. These three state sets are depicted in Figure 7.2. Only state variables are depicted that have a probability of progress in the different states of Nw.

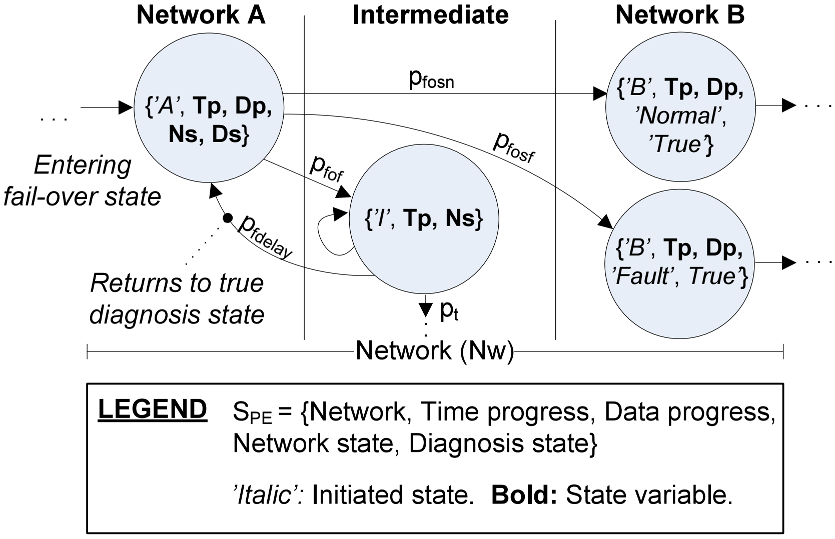
Network A
Intermediate
Network B
{'A', Tp, Dp, Ns, Ds}
p_fosn
p_fosf
p_fof
{'I', Tp, Ns}
p_fdelay
p_t
{'B', Tp, Dp, 'Normal', 'True'}
{'B', Tp, Dp, 'Fault', True'}
Entering fail-over state
Returns to true diagnosis state
Network (Nw)
LEGEND
S_PE = {Network, Time progress, Data progress, Network state, Diagnosis state}
'Italic': Initiated state. Bold: State variable.

Figure 7.2: *Integration of two networks and intermediate states. Apart from initiated states, only states where there is a probability of progress are depicted.*

Starting from *network A* a fail-over is initiated from a state dictated by the applied policy. When a fail-over is initiated it may either fail (with probability P_{fof}) or succeed. If the fail-over *succeeds* the data transfer is immediately resumed in network B in either the normal (P_{fosn}) or fault state (P_{fosf}). These probabilities are determined by the steady-state probability of the network B states. If a fail-over *fails*, the penalty is a geometrically distributed waiting time with the parameter P_{fdelay} where no data transfer occurs. This penalty is handled in an *intermediate* model part. After a failed fail-over, the transfer is resumed in network A to the same (true) normal/fault state as before the fail-over.

A limitation of the model in its presented form is that it does not keep track of the states of *network A* and *network B* simultaneously. Thus, only one fail-over is enabled. This limitation should be removed in future modelling approaches.

Applying and evaluating policies

To implement policies in the model, states given by the policy have their outgoing transitions replaced with transitions to the *Intermediate* and *network B* states as shown in Figure 7.2. Now different policies can be implemented and evaluated in the model by solving for the steady state solution to the full model

$S_{PE} = \{Nw, Tp, Dp, Ns, Ds\}$ to obtain Ω_{model} (see Section 5.3 for how this metric is obtained).

7.1.3 PE DTMC Model Overview

A full overview of the *Policy Evaluation (PE) DTMC Model* is presented in Figure 7.3 along with the simulation model introduced in Section 3.3. Recognizing that the simulation model can be considered a valid, yet comprehensive, approach of performing policy evaluation it is in the remainder of this chapter referred to as the *PE Simulation Model*.

The state space described by S_{PE} represents the complete *PE DTMC model*. A summary of the state space and its dimensioning used in this chapter is presented in Table 7.1.

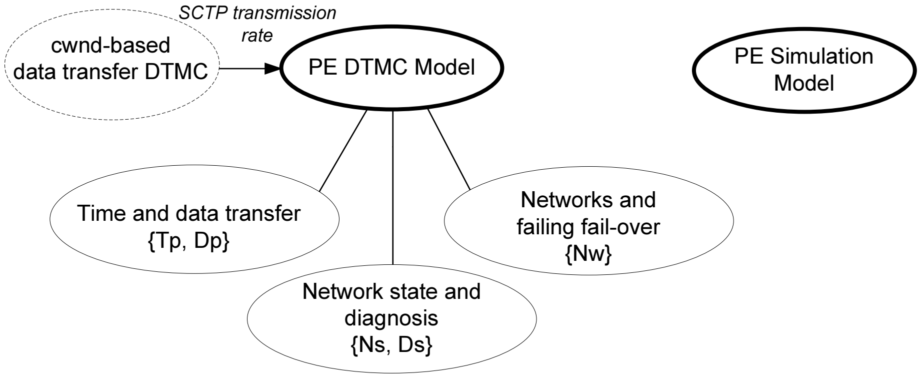
SCTP transmission rate
cwnd-based data transfer DTMC
PE DTMC Model
PE Simulation Model
Time and data transfer {Tp, Dp}
Networks and failing fail-over {Nw}
Network state and diagnosis {Ns, Ds}

Figure 7.3: *Overview of the models applied for policy evaluation.*

State variable	**States**
Data progress (Dp)	$n = 26$ and j = 1...n [..., $(j-1)\frac{data_{size}}{n-1}$ to $(j)\frac{data_{size}}{n-1}$, ...]
Time progress (Tp)	$m = 10$ and i = 1...m [..., $(i-1)\frac{t_{deadline}}{m-1}$ to $(i)\frac{t_{deadline}}{m-1}$, ...]
Network states (Ns)	2 - [$Normal$, $Fault$]
Diagnosis States (Ds)	2 - [$True$, $False$]
Network (Nw)	3 - [$network$ A, $network$ B, $Intermediate$]
Total states	2600 as Ds is not included under the *Intermediate* network states.

Table 7.1: *State-space of the PE DTMC model.*

7.2 Applications of the PE Models

In the remainder of this chapter two main studies are conducted:

Evaluation of Remediation Policies for Imperfect Diagnosis (Section 7.3)
In this study the PE DTMC model is applied to determine best decision strategies for fault remediation under imperfect diagnosis. Besides gaining insight into the properties of a set of policy heuristics for the given end-user service, a comparison is conducted between the PE DTMC model and the more comprehensive PE Simulation model used as a reference in this work. A central part of this study is the diagnosis component and its representation of the diagnosis performance properties. As a first step, a memory-less diagnosis approach based on the *basic threshold* diagnosis mechanism is studied where *independent diagnosis outcomes* may be assumed.

Model-based Evaluation of Trade-offs in Temporal Diagnosis (Section 7.4)
The results of the policy evaluation study are extended to also include diagnosis mechanisms and settings where independence in the diagnosis outcome is an inappropriate assumption. This is the case when considering *temporal* diagnosis components. The focus is, thus, to examine how the parsimonious diagnosis component may capture essential properties of the *α-count heuristic*. The results are, then, used to identify good settings of the *α-count* parameters trading off diagnosis imperfections to optimize the end-user service reliability.

7.3 Evaluation of Remediation Policies for Imperfect Diagnosis

As emphasized in the introduction of this chapter a central focus of the ODDR component is on the interactions between diagnosis and remediation decisions and potential reliability gains in studying these. An important issue is how accurate fault diagnosis needs to be in order to initiate remediation actions that can increase service reliability. Making good decisions to initiate such remediation is not trivial. The decision process must consider: A) *Imperfect diagnosis* where fault estimates may be false which leads to unnecessary or damaging remediation actions or a true fault is not diagnosed in a timely manner. B) *Requirements* from the end-user service. C) *Properties of remediation* such as potential remediation gain, risk of failing remediation and signalling and time overhead associated with the remediation action. In this work, A) and B) are examined primarily.

Approach for identifying best decisions
Remediation decisions are based on different decision rule sets referred to in this work as *policies*. A rule in this context defines a state or ranges of states in which an action is triggered in accordance to the decision policy rules. To identify best policies the method applied in this work has been to evaluate a set of heuristic decision policies. Alternatively, optimal decision policies may be derived by assessing the PE DTMC model in the Markov Decision Process (MDP) framework. However, a clear mapping between a reward function needed in the MDP specification and optimizing for Ω has not been identified. This option should be revisited in future work. Instead, a determination of the best policy is made simply by comparing and identifying the policy that maximizes Ω. This manual approach provides an understanding of the optimization problem

that may be applied in future work to identify optimal decision policies.

In assessing the impact of diagnosis imperfections in this chapter, the main focus is on memory-less diagnosis approaches and the main assumption that diagnosis estimates are independent and only conditioned on the true network state. In the following sections properties of an imperfect diagnosis component are specified. Next, it is defined how they are captured in the parsimonious diagnosis model. Finally, policy evaluation result are evaluated for different diagnosis capabilities.

7.3.1 Imperfect Diagnosis Properties

For the study of various diagnosis imperfections a starting point has been made in the *basic threshold based state estimator* introduced in section 4.3. Using the diagnosis outcome independence assumptions discussed in Section 4.6, the full diagnosis component behavior can be defined for the studied fault model from the tuple:

$$\begin{aligned} D_{th} =& (TNR, TPR) \\ TNR =& \frac{\#TN}{\#FP + \#TN} \text{ and } TPR = \frac{\#TP}{\#TP + \#FN} \end{aligned} \tag{7.1}$$

For the subsequent policy evaluation, a set of diagnosis component settings has been specified to represent different diagnosis properties. This set is introduced in Table 7.2. It is defined from different realizations of the threshold value of γ_{RTT} trading off false alarms and the reaction time. To obtain these realizations a simulation study has been conducted using cross-traffic to provide the required congestion buffer dynamics and similar fault and normal state data transfer performance as obtained with independent losses. The network state definitions of Table 3.3 are used. To delimit our study two thresholds are introduced for *imperfect diagnosis*: $\gamma^0_{RTT} = 64.5\,ms$ and $\gamma^1_{RTT} = 59.5\,ms$. These values have been picked empirically from a ROC curve obtained from simulation results showing possible trade-offs between TPR and FPR for the provided diagnosis component. γ^0_{RTT} has been chosen to ensure a low FPR and False Alarm Ratio (FAR) at the cost of a high Reaction Time (RT). γ^1_{RTT} has been chosen to approximately halve the reaction time at the cost of a higher false alarm ratio. These values will in the policy evaluation show what is most important to the considered system: a low FAR in relation to a low RT. To obtain the diagnosis performance tuples for these results 100 independent simulation runs have been conducted of a data transfer of $data_{size} = 10\,MB$. At $t = 12\,s$ a change from normal to fault occurs providing a period of true normal and fault state to consider. The threshold state estimator is executed on the traces of RTT observations ($\omega_{RTT} = 300\,ms$). Now counting the instances of the different diagnosis outcomes an estimate of TNR and TPR is obtained from Equation 7.1. The results for simulation are depicted in Table 7.2. Before mentioned implications on the FAR and RT can be observed.

7.3.2 Parametrization of Policy Evaluation Model

An open question from the introduction of the *parsimonious diagnosis model* in Section 7.1.1 is how to parametrize it based on the diagnosis performance tuple $D_{th} = (TNR, TPR)$.

Threshold	**(TNR, TPR)**	**FAR**	**RT** (μ,σ) $[ms]$
Sim: $\gamma^0_{RTT} = 64.5\,ms$	(0.984, 0.102)	0.014	3856, 3338
Diagnosis model	(0.984, 0.104)	0.016	3743, 3811
Sim: $\gamma^1_{RTT} = 59.5\,ms$	(0.953, 0.225)	0.041	1983, 1998
Diagnosis model	(0.953, 0.225)	0.044	1573, 1607

Table 7.2: *Key metrics from the diagnosis component in simulation and in the diagnosis model part for two diagnosis configurations.*

From the independent diagnosis outcome assumption studied in Section 4.6 it is defined that a diagnosis outcome only depends on the current network state and not previous diagnosis outcomes. Conditioned on the network state the probabilities are defined:

$$\begin{aligned} p_{TN} =& P(Ds = norm.|Ns = norm.),\ p_{FP} = P(Ds = fault|Ns = norm.) \\ p_{TP} =& P(Ds = fault|Ns = fault),\ p_{FN} = P(Ds = norm.|Ns = fault) \end{aligned}$$

where:

$$\begin{aligned} p_{FP} =& 1 - p_{TN} \\ p_{FN} =& 1 - p_{TP} \end{aligned}$$

Considering that the parameters of the true network state ON-OFF process are known: p_f probability of a fault occurrence and p_n probability of a repair occurrence, the state transition probabilities can be written:

$$\begin{aligned}
p_{FP|TN} =& \quad (1-p_f)\cdot(1-p_{TN}) & = p_{FP|FP} \\
p_{FP|FN} =& \quad p_r\cdot(1-p_{TN}) & = p_{FP|TP} \\
p_{TN|FP} =& \quad (1-p_f)\cdot p_{TN} & = p_{TN|TN} \\
p_{TN|FN} =& \quad p_r\cdot p_{TN} & = p_{TN|TP} \\
p_{TP|FN} =& \quad (1-p_r)\cdot p_{TP} & = p_{TP|TP} \\
p_{TP|TN} =& \quad p_f\cdot p_{TP} & = p_{TP|FP} \\
p_{FN|TP} =& \quad (1-p_r)\cdot(1-p_{TP}) & = p_{FN|FN} \\
p_{FN|TN} =& \quad p_f\cdot(1-p_{TP}) & = p_{FN|FP}
\end{aligned}$$

where it is the case that:

$$\begin{aligned} p_{TP|TN} + p_{FN|TN} = p_{TP|FP} + p_{FN|FP} = p_f \\ p_{TN|TP} + p_{FP|TP} = p_{TN|FN} + p_{FP|TN} = p_n \end{aligned}$$

This leaves two parameters free being p_{TN} and p_{TP}. Now finally considering Equation 7.1, the probabilities may be estimated as: $p_{TN} = TNR$ and $p_{TP} = TPR$.

Intermediate diagnosis model evaluation

The diagnosis model has been simulated stochastically in compliance to the ns-2 based simulation traces of the basic threshold approach used to obtain parameterizations of TPR and TNR in Table 7.2. Thus, a trace of $60\,s$ has been generated with a deterministic transition at $t = 30\,s$ for 2000 independent simulation runs. The model based diagnosis performance results in terms of FAR and RT are seen to correspond to the ns-2 based simulation results for both parameter sets. This indicates that the diagnosis model is a useful representation of the threshold based diagnosis component for the defined normal and fault states.

7.3.3 Heuristic Policy Definitions and Evaluation Setup

In this section, the heuristic policies applied in this chapter are defined. The approach to define these policies has been based on the state-space options of the PE DTMC model and initial empirical trials in the model. The considered policies are:

(PI) No fail-over: No fail-over is commenced during the data transfer. This policy will enable to identify cases where not to do anything may be a better strategy than failing over.

(PII) Fail-over at diagnosed fault state: A fail-over is initiated as soon as the network by diagnosis is estimated to be in a faulty state. This is relevant to investigate how different settings of the diagnosis module affect the remediation performance given the decision component has a high trust in the diagnosis component.

(PIII) Minimum time threshold: A fail-over is initiated when: 1) the conditions in **(PII)** are met and 2) a minimum period of time has passed from a data transfer has been initiated before a fail-over is allowed. This policy has multiple settings represented by the minimum period parameter, $\gamma_{mintime}$. The policy provides options to study decisions based on the end-user service state. It enables to evaluate the time state criticality in relation to the deadline: $t_{deadline}$.

Common for these policies is that they have been found to enable an optimization of the end-user service parameter Ω. Other policy heuristics have also been identified as worth investigating such as a threshold on the data progress state in combination with (PIII). This policy may offer options to avoid fail-overs when the data-progress is far within the time deadline and as a consequence reduce overhead. However, in the following focus is on the Ω optimization problem and such additional policies have been left for future work.

Setup for policy evaluation

To establish the impact on Ω from *imperfect diagnosis* capabilities, results are provided for γ^0_{RTT} and γ^1_{RTT}. For reference, the ideal case of *perfect diagnosis* (TPR=1, FPR=0) is also included. In all cases a data upload action is started in network A. The transfer is started in a normal state with probability 1 assuming that a good pre-diagnosis can be made in the network selection prior to initiation of the data transfer. In each data transfer only one fail-over to network B is allowed. Finally, network B is started in a normal or fault state given by the state steady-state distribution. To focus the impact in the study on policies, diagnosis and hand-over, differences of parameters between network A and B are

PE DTMC and simulation model parameters			
$data_{size}$	$10\,MB$	Fail-over to normal s. p_{fosn}	0.430
$t_{deadline}$	$30\,s$	Fail-over to fault s. p_{fosf}	0.520
Period $\kappa^-1 = T$	$0.398\,s$	SCTP packet payload `Pdata`	1452 B
Failing fail-over prob. p_{fof}	0.05	Congestion window bin size W_{bin}	5
Mean failing fail-over delay	$1.2\,s$	Max congestion window size W_{max}	46

Frequent Fault Event (short repair time)	
Mean periods, $1/\Lambda_f^{(A/B)}$, $1/\Lambda_r^{(A/B)}$	$12.42\,s$, $15\,s$
SS probability: *normal*, *fault*	0.453, 0.547
$P(Ns = normal \mid t = 0 \ldots 30\,s)$	0.09

Table 7.3: *Summary of parameters used in the policy evaluation study.*

eliminated by defining the two networks to have equal parameters. In addition, this ensures the study of a non-trivial case where network B is either significantly better than A (and could always be used) or significantly worse (and should never be used).

The parameters used in the policy evaluation results are provided in the state definitions of Table 3.3, the diagnosis properties of Table 7.2 and finally Table 7.3 which summarizes the parameters introduced in Chapter 5 and in this chapter. In the table five new parameter values are further introduced for p_{fof}, p_{fosn}, p_{fosn} and the fault and repair rates. A value for p_{fof} has been chosen to assume that in 5% of all cases a fail-over will fail. p_{fosn} and p_{fosn} then correspond to the steady state probabilities of failing over to network B provided for the remaining 95% of sucessful fail-overs. For policy evaluations, a single fault/repair process has been specified. It assumes a frequent state change meaning that several ON-OFF periods may be experienced during a single data transfer. In practice this means that an occuring fault may also *self-repair* with a high probability leading to cases where waiting for a repair may be beneficial.

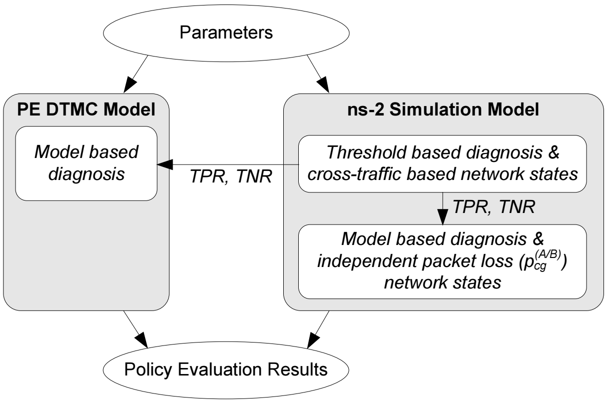
Parameters
PE DTMC Model
Model based diagnosis
ns-2 Simulation Model
Threshold based diagnosis & cross-traffic based network states
TPR, TNR
TPR, TNR
Model based diagnosis & independent packet loss ($p_{cg}^{(A/B)}$) network states
Policy Evaluation Results

Figure 7.4: *Overview of diagnosis mechanism representations in the PE DTMC and ns-2 simulation models.*

Previously, two different state definitions have been defined: one making use of cross-traffic to offer a realistic impact of the fault model and one based on independent packet loss events. The former is necessary for the RTT threshold based approach to enable diagnosis at all. The latter allows for a simulation setup which more closely resembles the PE DTMC model assumptions. Comparing the two can help clarify differences in PE DTMC and the more realistic ns-2 simulation results. The diagnosis model is implemented in the ns-2 based simulation setup as depicted in Figure 7.4 to enable diagnosis in the independent loss event case where no cross-traffic is executed. In the ns-2 implementation of the diagnosis model the state estimates are randomly generated by two separate Bernoulli processes representing the normal and fault state. Which Bernoulli process is active is defined by the fault model ON-OFF process.

To obtain $\Omega_{simulation}$ 2000 independent simulation runs of a data upload scenario are conducted. $\Omega_{simulation}$ is then derived as the fraction of data transfers that were completed within the deadline $t_{deadline}$. Note, that 2000 simulation runs provide for the binomial proportion variable Ω in worst case ($\Omega = 0.5$) an error bound of ± 0.022 (CI 95%) which is useful to demonstrate significant improvements for some settings of the policy evaluations while keeping simulations tractable.

The policy evaluation results are initially considered for *perfect diagnosis* comparing the PE DTMC model and the ns-2 simulation model results. These results provide a baseline for subsequently studying the impact of diagnosis imperfections and to which extent the different decision policies impact the Ω optimization problem.

7.3.4 Perfect Diagnosis

For independent losses, results of $\Omega_{simulation}$ (left y-axis) and Ω_{model} (right y-axis) are depicted in Figure 7.5. The PE DTMC model results have been *scaled* and *shifted* to graphically match simulation results at policies **(PI)** and **(PII)**. Results are presented in this manner to compensate for the differences in quantitative results (discussed in section 5.3) and to emphasize qualitative similarities instead. These qualitative similarities are in this respect most important to establish the capabilities of the model to identify best policies.

In Figure 7.5 varying settings of $\gamma_{mintime}$ (Policy **(PIII)**) are depicted on the x-axis. Notice, $\gamma_{mintime} = 0\,s$ corresponds to policy **(PII)**, and $\gamma_{mintime} > 30\,s$ corresponds to policy **(PI)**. The model results show that failing over immediately when a fault is diagnosed **(PII)** should be done. The same conclusion is true for the simulation results where an actual gain of nearly 10 percentage points is achieved compared to no fail-over (PI). The results show that for the studied settings it is worth taking the chance that network B may be in a good state compared to staying in network A. Should network B turn out to be in a fault state failing over is not worse than staying, as the expectation on the remaining fault time in both networks is equal due to equal parameters and the Markov property. Studying policy **(PIII)** there is consequently, as stated by both the Markov model and simulation results, no gain in waiting from the data transfer has been started before a fail-over is allowed.

The same policies have been studied in a simulation case where losses are caused by cross-traffic. Selected results are depicted in Figure 7.6. A consistent tendency has been observed in the perfect diagnosis case that some gain can be

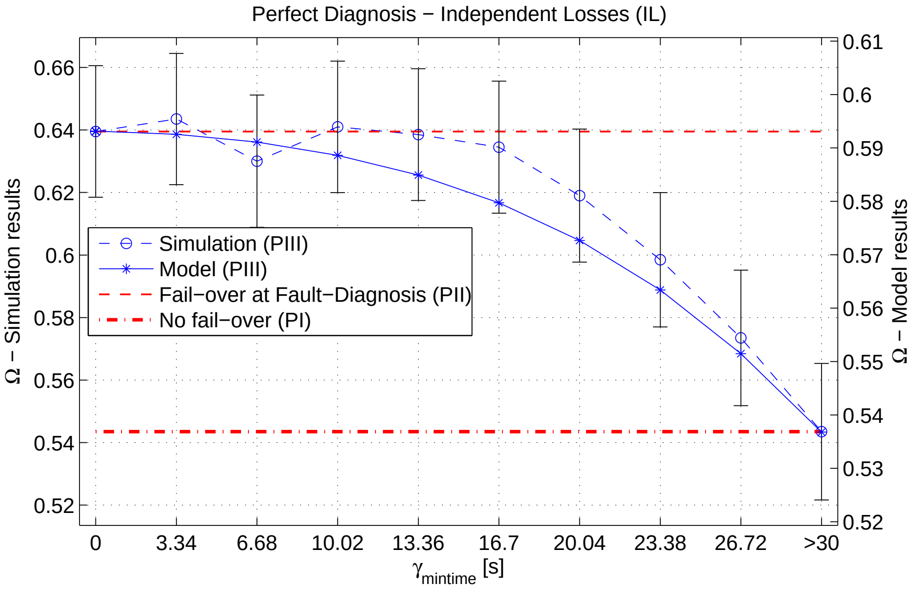

Figure 7.5: *Model and simulation results for remediation policies* ***(PI)****,* ***(PII)*** *and* ***(PIII)*** *in a setting where perfect diagnosis is assumed.*

obtained from waiting in the order of 7-10 s before a fail-over should be allowed (policy **(PIII)**). These results are seemingly conflicting with the PE DTMC model and the ns-2 simulation results with independent losses. A further study

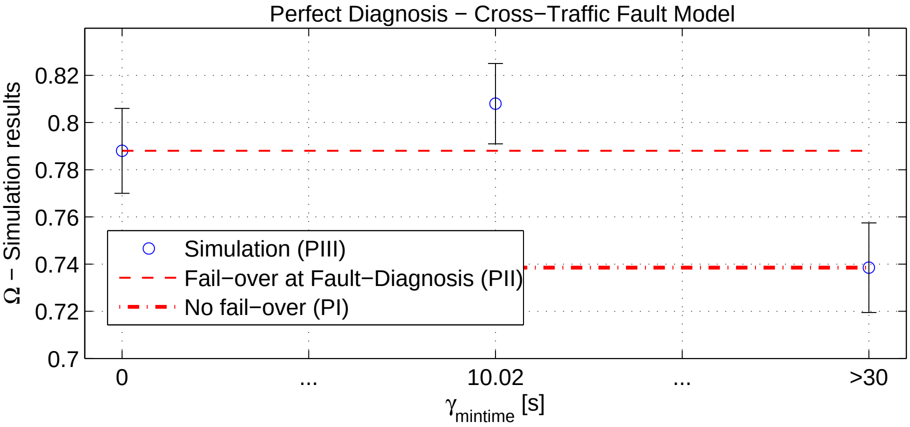

Figure 7.6: *Selected policy settings where perfect diagnosis is assumed and losses are generated by cross-traffic.*

of average throughput correlated to time (2000 runs) in a pure normal and a pure fault state based on cross-traffic reveals that the throughput is higher initially compared rest of the transfer time. E.g. in a fault state the throughput is $271 \pm 2.6\,KB/s$ for the first $8\,s$ and $248 \pm 0.5\,KB/s$ for the remaining. The same tendency is not observed when losses are independent. This indicates that there is some significant correlation between the, in average more than

60, concurrent cross-traffic and the SCTP connection. Waiting, enables the transfer to take advantage of two of such periods with increased throughput in network A and network B. In conclusion, while large transient effects have not been observed in the RTT based diagnosis traces, other transient effects seem to have a relevant impact on the data transfer rate. It is recognized that the data transfer models may need to incorporate these currently un-modelled aspects in a future model iteration. However, in the further analysis, the remaining simulation based policy evaluations are conducted using independent losses. Also, the model based diagnosis component implemented in simulation is used to focus on the aspects of diagnosis imperfections.

7.3.5 Imperfect Diagnosis

Results of imperfect diagnosis are presented in Figure 7.7 for γ^0_{RTT} and Figure 7.8 for γ^1_{RTT}. Two important observations can be made. 1) for γ^0_{RTT}, failing over immediately **(PII)** is still better than not failing over **(PI)**. However, for γ^1_{RTT}, lower reliability is actually obtained from **(PII)**; given that network A is started in a normal state. This shows that the increased probability of false alarms is more damaging than the increased reaction time. 2) The second observation is that **(PIII)** becomes important. There is actually a significant gain in waiting to allow a positive diagnosis to cause a fail-over. In addition our results show that the optimal setting of $\gamma_{mintime}$ increases as the FPR/FAR becomes worse. This is expectedly the case as FAs in the beginning of a data transfer can lead to undesirable fail-over from a normal state to a faulty one. As only one fail-over is allowed there is no chance to recover from a bad decision. This fact clearly weakens **(PII)** compared to **(PIII)**. Waiting serves to initially ignore positives (true or false) to a point where there is a higher probability that the alarm is true and there still is a potential gain in failing over to a normal state.

Also, for imperfect diagnosis the policy evaluation conclusions seem consistent for model and simulation results. Overall, for the studied settings it is shown that the model, despite quantitative deviations, delivers good qualitative results.

The aspects of failing fail-over with the parameters in Table 7.2 have in the considered results had insignificant influence compared to $P_{fof} = 0$. The model, however, enables an evaluation of how significant these parameters need to be, and under which policies, before the reliability is considerably degraded. This should be studied in future work.

The proposed PE DTMC model has been constructed to explore relevant policies for different diagnosis capabilities. Thus, at this stage limited effort has been spent on model optimizations. The model consists of altogether 2600 states and the evaluation of a single policy setting takes $\sim 3\,s$ on a 2.5 GHz x86 based system. State space reduction options should be considered to support other relevant functions and enable model solutions in resource constrained environments. Potential options are to examine the impact on qualitative results by reducing the amount of time and data progress states as well as the aggregation of a subset of these states depending on implemented policies. Further, it should be noted that some relevant model functions could expectedly be added with no (using alternative parameterization) or limited increase of state space such as evaluating best fail-over policies for multiple available remediation networks and support for temporal diagnosis approaches which improve over time

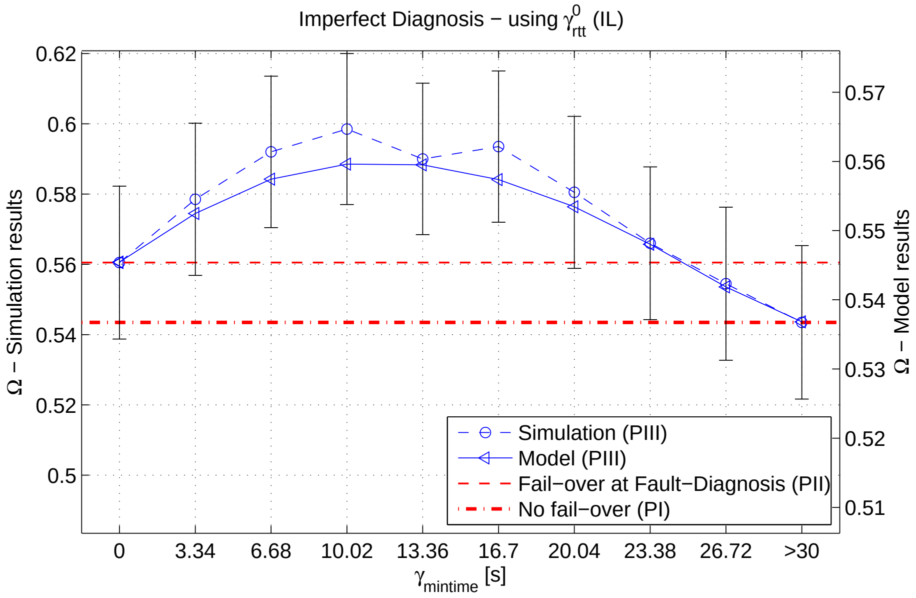

Figure 7.7: *Model and simulation results for remediation policies **(PI)**, **(PII)** and **(PIII)** under imperfect diagnosis using the γ^0_{RTT} diagnosis performance setting.*

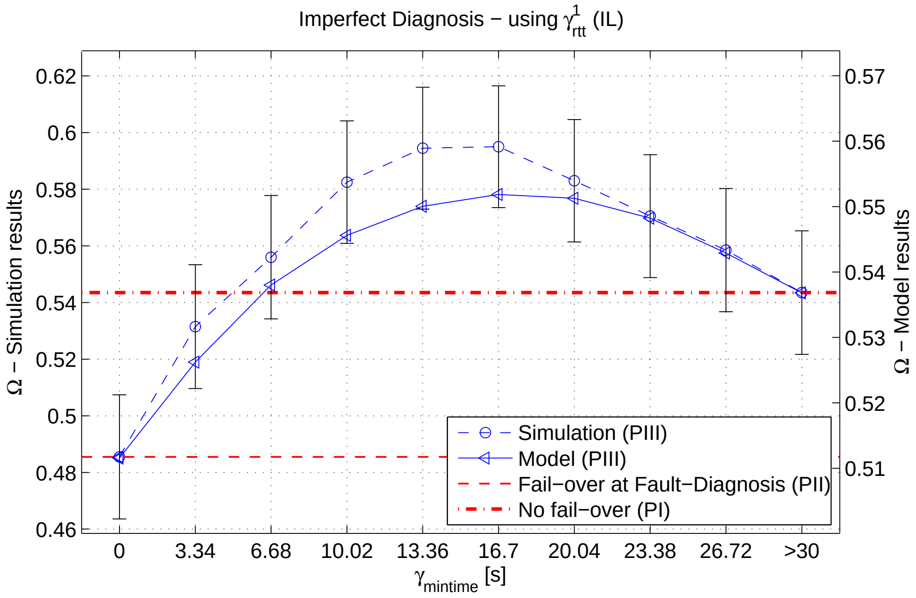

Figure 7.8: *Model and simulation results for remediation policies **(PI)**, **(PII)** and **(PIII)** under imperfect diagnosis using the γ^1_{RTT} diagnosis performance setting.*

(as more observations are made). The latter aspect is the sole objective of the next section where focus will be on extending the analysis of the parsimonious diagnosis model.

7.4 Model-based Evaluation of Trade-offs in Temporal Diagnosis

The parsimonious diagnosis model introduced in Section 7.1.1 is based on the assumption that diagnosis outcomes are independent. While this assumption enables a simple model and parameterization approach, it also hinders assessment of an essential class of temporal diagnosis mechanisms that were presented in Chapter 4, which correlate observations in time to mitigate observation imperfections. In the following, it is examined if the independence assumption can be relaxed by freeing model parameters from two in the previous analysis to six, without increasing the model state space. The approach taken is to define a set of representative diagnosis performance metrics and derive their closed-form equations from these free parameters. The metrics enable: I) a sensitivity analysis to provide valuable insights into the model capabilities, II) the establishment of the metrics' sufficiency to capture main properties of diagnosis for a given service reliability problem, and III) model parameterization from a temporal diagnosis component to assess potential reliability gains and its various trade-off settings.

Based on the improved diagnosis approach the data transfer service reliability problem of improving the data transfer success probability Ω is re-evaluated. However, while focus in the previous study was on selecting the best policy heuristic, attention is now shifted to determining the best parametrization of the temporal diagnosis component trading off its imperfections.

Gains and main challenge

A comparison was introduced in Section 4.6 on the properties of the basic threshold based state estimator and the α-count heuristic representing the memory-less and temporal diagnosis approaches, respectively. Summarizing its conclusions the α-count based diagnosis enables to improve both fault and normal state diagnosis accuracy and reduces the probability of false alarms. The cost is a significant transient phase which affects the reaction time significantly as well.

Besides changing the characteristics of the diagnosis performance, also capturing the diagnosis capabilities of the temporal component in a simple model becomes more challenging. The parameters TPR/TNR in this case depend on time and they cannot in a simple manner be derived from traces of a diagnosis component state estimates as in the previous section. Thus, an alternative approach to characterize such complex diagnosis behavior is the aim of the subsequent analysis. The proposed approach will enable a study of the potential reliability gains obtained from different settings of α-count compared to the previously assessed threshold based diagnosis approach. To delimit the focus to the diagnosis component a fixed remediation policy of *fail-over at diagnosed fault state* corresponding to policy (PII) is considered in the following setup.

7.4.1 Parsimonious Diagnosis Model Revisited

In the initial step, the previously defined 4-state light-weight diagnosis model is revisited to clarify how it may also be used to capture essential temporal diagnosis properties. From the model, closed-form equations are derived of performance metrics considered to be suitable to describe essential diagnosis capabilities. They can subsequently be used to parametrize the model.

Model re-definition

The revised parsimonious diagnosis model is depicted in Figure 7.9. In the general diagnosis model in Figure 7.1.1 all states are connected by a transition. However, for the type of remediation policy studied (fail-over at diagnosed fault state), transitions from positive states (true or false) can be neglected as subsequent diagnosis behavior will have no influence on the reliability analysis (for remediation policies that also depend on other criteria such as criticality of the service, these transitions may be re-considered in future work). Consequently, positive states are absorbing with $p_{FP|FP} = 1$ and $p_{TP|TP} = 1$. The resulting model, thus, has six inter-state transition probabilities.

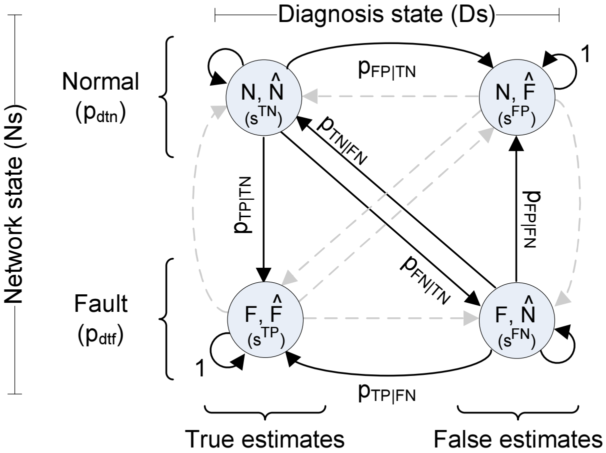

Figure 7.9: *The parsimonious diagnosis model has 6 parameters; 4 are free. Greyed transitions are not considered for the studied remediation policy.*

Now, recalling that $p_f \in (0 \ldots 1)$ is the probability of a fault transition and $p_r \in (0 \ldots 1)$ the probability of a repair transition the system equations for the transition probabilities can be specified:

$$\begin{aligned} p_f =& p_{FN|TN} + p_{TP|TN}, \quad p_{TN|TN} = 1 - (p_{FP|TN} + p_f) \\ p_r =& p_{TN|FN} + p_{FP|FN}, \quad p_{FN|FN} = 1 - (p_{TP|FN} + p_r) \end{aligned} \tag{7.2}$$

As p_f and p_r are defined by the fault-model, they are not free parameters of the model. Thus, only one of the parameters $p_{FN|TN}$, $p_{TP|TN}$ and one of $p_{TN|FN}$, $p_{FP|FN}$ can be considered free. The final free model parameters are summarized in Table 7.4.

Diagnosis metrics

The requirements for the performance metrics characterizing diagnosis are: I) that equations of individual metrics can be derived from the model, II) that metrics in a simple manner can be obtained from traces of the temporal diagnosis component behavior for parameterization purposes, and III) that the set is sufficient to characterize diagnosis when studying reliability assessment problems.

Altogether, a set of two metrics is defined. Each of the metrics can be obtained from an implementation of the diagnosis component in the following manner. A diagnosis trace is obtained by starting the diagnosis process at $t = 0$ (t_0) in a network in normal state with the considered ON-OFF process. The trace is run until an alarm is observed and the true state of the network is recorded at that instant. A set consisting of M repetitions of independent diagnosis traces is created. The number of traces in which remediation is based on a False Positive (False Alarm) are counted by O. The considered metrics are (see also Figure 4.1):

Probability of Remediation on False Alarm, (p_{RFA})

Probability that a fail-over will be commenced based on a False Alarm. Defined as: $p_{RFA} = \frac{O}{M}$

Mean Remediation Reaction Time, (μ_{RRT})

Describes the mean time until a remediation action occurs from t_0. Notice, this metric does not distinguish between whether a fail-over is caused by a false or a true alarm. It must be noted that the metric *Probability of Remediation on True Alarm* p_{RTA} could also be specified. For steady state solutions (where diagnosis is performed until an alarm occurs) it is valid that $p_{RTA} = 1 - p_{RFA}$ and no additional information is provided by having p_{RTA}. This relation changes, though, when considering transient versions of these metrics as will be shown later in this section.

An additional metric p_{miss} has also been studied to capture the probability that a fault is not diagnosed during the limited period of a service usage case. Defining p_{miss} as the probability of missing the first occurred fault, the resulting equation for p_{miss} has been found only to depend on p_{RTA} and μ_{RRT}. Thus, this definition of p_{miss} provides no additional information and has not been considered further.

Summarizing the considered diagnosis performance metrics, p_{RFA} represents the capability of diagnosis to lead to correct remediation (from a true fault state). The metric p_{RFA} is, however, not sufficient alone. Even though a fail-over may occur with $p_{RFA} = 0$ this may not matter if this fail-over occurs after

Parameter	**Range**	**Description**
$p_{FN\|TN}$	$0 \ldots p_f$	Probability of a False Negative (FN) conditioned on being in a True Negative (TN) state and a fault occurrence
$p_{FP\|TN}$	$0 \ldots (1 - p_f)$	Probability of a False Positive (FP) conditioned on being in a True Negative (TN) state and no occurrence of a fault.
$p_{TN\|FN}$	$0 \ldots p_r$	Probability of a True Negative (TN) conditioned on being in a False Negative (FN) state and a repair occurrence
$p_{TP\|FN}$	$0 \ldots (1 - p_r)$	Probability of a True Positive (TP) conditioned on being in a False Negative (FN) state and no occurrence of a transition to the normal state.

Table 7.4: *Free diagnosis model parameters.*

the end-user service has failed anyways. To capture these time aspects μ_{RRT} is needed.

For these diagnosis metric definitions focus has been to ensure simple derivations of closed form-equations from the diagnosis model. The approach has been to specify metrics based on long term behavior of the diagnosis component (where a fail-over is known to happen eventually). However, in reality the diagnosis component is not necessarily operating in accordance with its long term behavior during the limited period of a service usage case. This may influence the quality of these metrics to specify the proper behavior of the diagnosis component. Closer attention will be devoted to this aspect later in the data transfer case study based sensitivity analysis. In the following section the specified metrics are derived from the model. Further, inverted equations are presented using the performance metrics as an input to establish the proper model parameters.

Performance metric equations

The diagnosis Markov Model may simply be considered using phase-type distribution theory [91][64] enabling a straight forward definition of the closed-form equations. Defining $\bar{p}$ as the initial transient state vector,
$Q = [q_{ij}]_{i=\{s^{TN},s^{FN}\};j=\{s^{TN},s^{FN}\}}$ the state transition probability matrix from transient state i to j and $R = [r_{ik}]_{i=\{s^{TN},s^{FN}\};k=\{s^{FP},s^{TP}\}}$ the transition probability matrix from transient state i to absorbing state k, these can be specified as:

$$\begin{aligned} Q &= \begin{pmatrix} p_{TN|TN} & p_{FN|TN} \\ p_{TN|FN} & p_{FN|FN} \end{pmatrix}, R = \begin{pmatrix} p_{FP|TN} & p_{TP|TN} \\ p_{FP|FN} & p_{TP|FN} \end{pmatrix} \\ \bar{p} &= \begin{pmatrix} 1 & 0 \end{pmatrix} \end{aligned} \tag{7.3}$$

Now the metric μ_{RRT} can be derived as the *mean time to absorption* given as:

$$\begin{aligned} \mu_{RRT} =& \bar{p}(\mathbf{I} - Q)^{-1}(T \;\; T)' \\ \mu_{RRT} =& \frac{p_{FN|TN} + p_{TP|FN} + p_r}{\Gamma} \cdot T \\ \Gamma =& (p_{FP|TN} + p_f)(p_{TP|FN} + p_r) - p_{FN|TN} \cdot p_{TN|FN} \end{aligned} \tag{7.4}$$

Where T is the time epoch duration of the discrete diagnosis. Next, the probability that remediation is commenced on a True/False alarm is derived based on the *probability of absorption* in the False Positive state.

$$\begin{aligned} p_{RFA} =& \bar{p}(\mathbf{I} - Q)^{-1}R \cdot \hat{e_1} \text{ (where } \hat{e_1} = (1 \;\; 0)') \\ p_{RFA} =& \frac{p_{FP|TN}(p_{TP|FN} + p_n) + p_{FN|TN}(p_n - p_{TN|FN})}{\Gamma} \end{aligned} \tag{7.5}$$

In the remainder of this work, focus is on the performance metric set: p_{RFA}, μ_{RRT}. To enable a study of the sensitivity of these parameters and subsequently use the model to perform service reliability evaluations, the inverted equations are derived; i.e. the description of the four free model parameters as functions of the two performance metrics. Clearly, this problem is underdetermined as only two equations exist to determine four parameters. While additional diagnosis performance metrics may be introduced, the approach taken in this work is to keep the number of performance metrics small and investigate the impact of the remaining two free undetermined variables. Based on Eq. (7.3) and (7.4) solutions may now be provided for $p_{FN|TN}$ and $p_{TN|FN}$:

$$\begin{aligned}
p_{FN|TN} =& \frac{\Upsilon}{\Psi} \cdot p_{TP|FN} + \frac{p_r \cdot p_f \cdot \mu_{RRT} + (p_{RFA} - 1)p_r \cdot T}{\Psi} \\
p_{TN|FN} =& \frac{\Psi}{\Upsilon} \cdot p_{FP|TN} + \frac{p_r \cdot p_f \cdot \mu_{RRT} - (p_{RFA} \cdot p_f + p_r)T}{\Upsilon} \\
\text{where, } & \Upsilon = p_f \cdot \mu_{RRT} + (p_{RFA} - 1)T \\
& \Psi = p_r \cdot \mu_{RRT} + (1 - p_{RFA})T
\end{aligned} \tag{7.6}$$

Note, that the solution consists of two linear equations where $p_{FN|TN}$ and $p_{TN|FN}$ only depend on $p_{TP|FN}$ and $p_{FP|TN}$ respectively.

Model parameters

To obtain fixed solutions for the underdetermined solutions in Equation (7.6) the additional freedom of the model must be clarified, when p_{RFA} and μ_{RRT} are fixed. This aspect is investigated by defining two different strategies, (M0) and (MI), for model parameterization. Next, the solutions for these approaches are compared in terms of the transient phases of the diagnosis performance metrics. These approaches are:

(M0) - *Equal p_{TP} and p_{FP}* - In this approach the free model parameters are fixed under the assumptions:

$$\begin{aligned}
p_{TN|FN} &= \left(\frac{p_{FP|TN}}{p_f - 1} + 1\right) p_r \\
p_{FN|TN} &= \left(\frac{p_{TP|FN}}{p_r - 1} + 1\right) p_f
\end{aligned}$$

In practice, this corresponds to defining that the probability of a true positive, p_{TP}, is the same independent on whether a transition takes place from s^{TN} (in a fault occurrence) or from s^{FN} (without a repair occurrence). The same is true for p_{FP} from s^{TN} and s^{FN}. This constraint is also applied in the independent diagnosis approach studied in Section 7.3.

(MI) - *Minimize $p_{FP|TN}$ and $p_{TP|FN}$* - In this approach we consider the solution $min(p_{FP|TN})$ and $min(p_{TP|FN})$ where $min()$ defines the lowest value in the legal range for the solution (also considering $p_{FN|TN}$ and $p_{TN|FN}$, see Table 7.4). The interpretation is that the model is forced to minimize direct transitions from s^{TN} and s^{FN} increasing probabilities that such transitions must take place during fault occurrence or repair transitions. This has an interesting impact on the transient behavior of the performance metrics which is considered next. From empirical studies of alternative approaches to (M0) and (MI) performed in the data transfer case study, these alternatives generally provide solutions close to and in the range of (M0) and (MI). Thus, only these two are considered in this paper.

Although both strategies (M0) and (MI) lead to same steady state metrics of p_{RFA} (and p_{RTA}) and μ_{RRT} the obtained solutions show different transient behavior. Thus, results for the transient model behavior of $p_{RFA}(l)$ and $p_{RTA}(l)$ (where l is the discrete time step) have been studied using parameters of Table 7.5 and γ_0 (see Table 7.6). An example is provided in Figure 7.10 in comparison to a simulation trace of the α-count heuristic (mean behavior over 2000 independent runs). Notice, that $p_{RFA}(l) + p_{RTA}(l) \neq 1$ for a low l. This is clearly

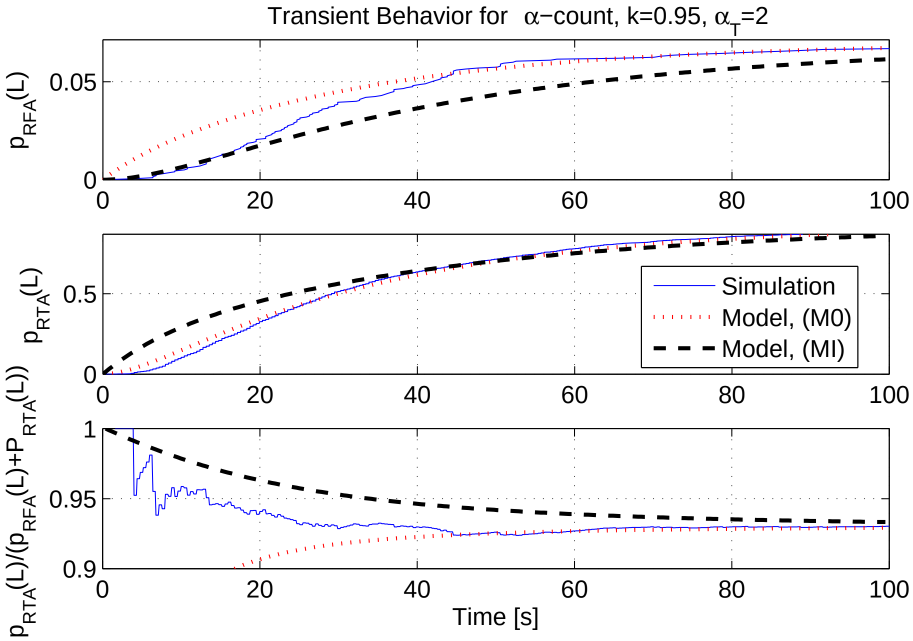

Figure 7.10: *Example of transient behavior for simulation and model settings (M0) and (MI).*

the case as the probability of a fail-over increases with time. For the simulation results it should be emphasized that the relation between $p_{RFA}(l)$ and $p_{RTA}(l)$ also is time dependent. Realizing that a service usage case is inherently limited in time ($< \gamma_{mintime} = 30\,s$) this transience can have some impact.

Comparing the model to the simulation results in Figure 7.10 two different behaviors are observed. While (M0) has a fairly good match in $p_{RTA}(l)$ for low l, $p_{RFA}(l)$ is overestimated. The opposite is the case for (MI). As shown in the lower graph of Figure 7.10, selecting either (M0) or (MI) will allow a shifting of the relation between the two metrics in time. Only (MI) shows the tendency that there is a higher probability of a true alarm for low values of l where (M0) shows the opposite. While the best solution may be found somewhere between (M0) and (MI), (MI) of the two seems to best describe this relation. Same conclusions have been found for other settings of α-count (not depicted).

Parameter	*Value*	*Parameter*	*Value*
Discrete Time Epoch Period, T	$0.398\,s$	p_f	$(1/12.42\,s) \cdot T$
		p_r	$(1/15\,s) \cdot T$

Table 7.5: *Initial analysis parameters.*

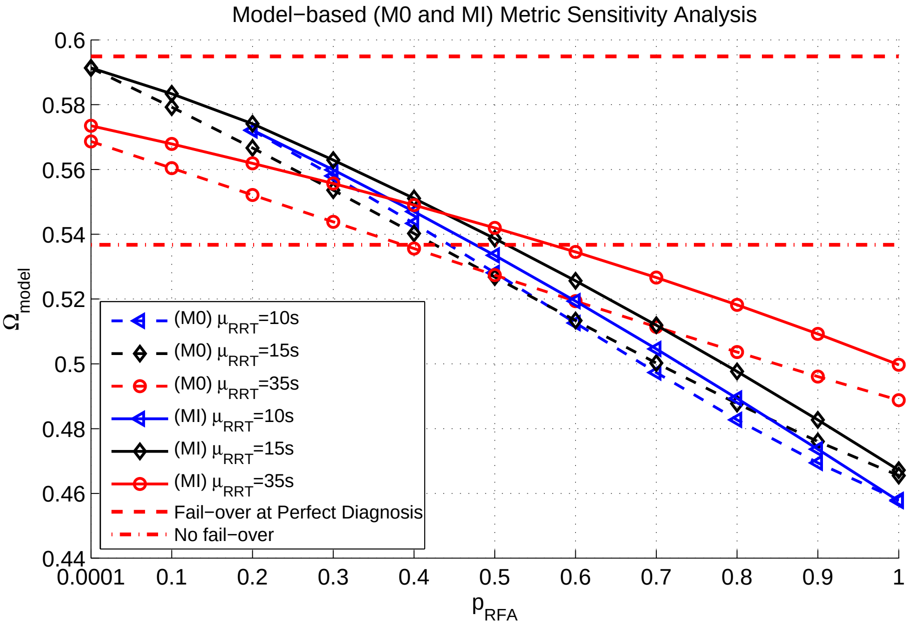

Figure 7.11: *Sensitivity analysis on Ω for performance metrics and parameterization method.*

7.4.2 Model Based Sensitivity Analysis

Having re-defined the parametrization of the parsimonious diagnosis model to the temporal diagnosis approaches, we now return to the time constrained reliable data transfer end-user service. Paring the diagnosis model with the specific reliability problem will help quantifying to what extent the chosen diagnosis performance metrics have an impact on service reliability. This is in the following obtained through a model based sensitivity analysis of the Ω problem. This analysis can in addition enable a comparison of the impact of parameterization approaches (M0) versus (M1), which helps to select the most suitable.

The sensitivity analysis has been conducted based on parameters of Table 7.5 using the PE DTMC model. Recall, that the default remediation policy studied is *fail-over at diagnosed fault state*.

In the study, parameters of p_{RFA} have been varied in the interval: [0.0001...1] for values of $\mu_{RRT} = [10\,s, 15\,s, 35\,s]$ under (M0) and (MI). The results are depicted in Figure 7.11. Besides the sensitivity analysis, results for perfect diagnosis (where $p_{RTA} = 1$ and occurred faults are diagnosed instantly) and no fail-over/no-remediation are depicted. Comparing (M0) and (MI) there is a general tendency that (MI) leads to a higher Ω. This is expected to be caused by the fact that (MI), as shown in our previous analysis, weighs p_{RTA} higher in the transient phase. In terms of the general results for different values of p_{RFA} and μ_{RRT} both (M0) and (MI) show the same tendencies. In fact, preliminary studies (not shown) comparing reliability analysis based on (M0) and (MI) to extensive simulation analysis lead to the same conclusions for best settings of the temporal diagnosis component. Thus, the difference may not be of significant importance for temporal diagnosis evaluation. Qualitatively, though, (MI) provides slightly better results compared to simulation. Thus, (MI) is considered in the remainder of this work. Focusing on the diagnosis performance metrics some general conclusions can be made. Clearly, both p_{RFA} and μ_{RRT} have a significant impact on Ω. A high p_{RFA} (above 0.45-0.55 for the studied settings of μ_{RRT}) and thereby a high level of false alarms can lead to situations in which diagnosis can result in even worse performance of remediation than performing no fail-over at all. Also, a quite low p_{RFA} is needed to get close to the theoretically best diagnosis performance. μ_{RRT} seems to be less influential, but a low value is also required to get close to the theoretical maximum. Notice, that for $\mu_{RRT} = 10\,s$ no solutions exist for $p_{RFA} < 0.2$ in these evaluations. This is because of an assumption that the fault ON-OFF process always starts in the normal state for the conducted analysis in this work. The normal state has a mean duration of $\approx 12.4\,s$ in which a fail-over can only occur due to a false alarm. Thus, it is not possible to have a very low probability of p_{RFA} and a very low value of μ_{RRT}. These facts demonstrate how μ_{RRT} controls promptness of fail-overs, while p_{RFA}/p_{RTA} controls which fail-over type is most likely.

Finally, our results show that the same value of Ω can be achieved for different settings of p_{RFA} and μ_{RRT}. Here, it is important to stress that it is the properties of an applied diagnosis mechanism that specify which values of p_{RFA} and μ_{RRT} can be achieved in practice. This type of analysis is conducted in the next section for the α-count heuristic.

7.4.3 Properties of Diagnosis Component

The next step is to determine the impact of diagnosis imperfections on service reliability in the PE DTMC model for the parsimonious diagnosis model fitted to the α-count heuristic. This analysis will show if the PE DTMC model can also be utilized to identify best diagnosis settings. Initially, performance capabilities of p_{RFA} and μ_{RRT} are investigated based on traces from the extensive ns-2 based system level simulation setup. Subsequently, best diagnosis settings are studied based on PE DTMC results and an equivalent analysis conducted in the ns-2 simulation model.

Evaluation setup

For evaluation, the PE DTMC model with the modified diagnosis model parametrization scheme is applied along with the ns-2 based simulation model. The simulation model is studied with network states based on independent losses, which also requires the use of model based diagnosis. As previously, an independent diagnosis outcome diagnosis model is used to obtain the process $J^{(l)}$ with l being the discrete time step. The α-count heuristic of Equation 4.1 is implemented in the simulation scenario using J^{l} to obtain the temporal diagnosis outcomes M^{l}. The simulation setup, then, enables both to obtain p_{RFA} and μ_{RRT} from an implementation of the α-count component in a realistic setting and an evaluation of $\Omega_{simulation}$.

The applied model and simulation parameters are adopted from the policy evaluation study.

Simulation Model Diagnosis Metrics

As seen from the sensitivity analysis of the proposed diagnosis model p_{RFA} and μ_{RRT} have a significant influence on Ω. In this section, sets of p_{RFA} and μ_{RRT} are extracted from simulation traces of α-count following the methodology previously described with the introduction of the metrics. It must be noted that only a single set of observations is needed from simulation to get the observation process J^l. Subsequently, the dataset can be post-processed for different settings of the temporal diagnosis component. This makes the following model based analysis process significantly less complex than the presented extensive ns-2 simulation based analysis.

Parameter set	*(TNR, TPR)*
γ^0	$(0.983, 0.097)$
γ^1	$(0.953, 0.225)$

Table 7.6: *Parameters used in the independent diagnosis outcomes diagnosis model.*

To carry out the studies two sets of parameters for TNR and TPR are defined for the independent diagnosis outcomes diagnosis model. These are given in Table 7.6 by γ^0 and γ^1. They represent two different possible trade-off options and have been obtained from a simulation analysis of a threshold based diagnosis component on RTT as previously. γ^0 is an example of a setting where the true negative ratio is high leading to few false alarms but at the cost of a low TPR leading to longer reaction time. For γ^1 the TPR is higher at the cost of more false alarms. Both settings correspond to the previously studied imperfect diagnosis cases whereas the parameters of the high true negative cases are slightly different as the estimates are derived from different simulation result sets.

Figure 7.12 depicts p_{RFA} versus μ_{RRT} for γ^0, γ^1 and different settings of the α-count component in simulation. The different settings have been obtained by varying the alpha-count threshold α_T.

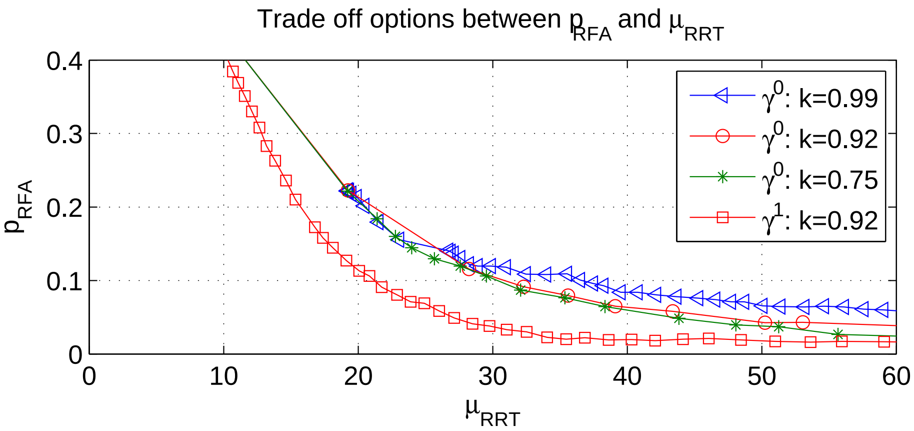

Figure 7.12: *Trade-off options between p_{RFA} and μ_{RRT} for different settings of α-count.*

For γ^0 three different settings of the forgetting factor k are chosen to represent a slow (0.99) medium (0.92) and fast (0.75) rate of reducing the counter variable α. For $k = 0.92$ and $k = 0.75$ the trade-off performance is similar while $k = 0.99$ seems to offer a worse p_{RFA} as μ_{RRT} increases. More interesting is the γ^1 setting (only made for $k = 0.92$) which offers low p_{RFA} values for lower μ_{RRT}. The filtering effect of α-count manages to reduce the amount of false alarms while maintaining the faster reaction time. In the following, the influence of the different settings are studied in the reliability analysis. Due to similar trade-off options of $k = 0.92$ and $k = 0.75$, the latter is not considered further.

Reliability Analysis - Model and Simulation

The reliability analysis for the reliable data transfer end-user service conducted in the model is presented in Figure 7.13. The x-axis defines varying α_T. Note, γ^1 has been evaluated for higher settings of α_T as γ^1 generates more positive estimates. Thus, a higher threshold may be needed to discriminate true positives and false positives. The vertical lines represent the *no fail-over* policy and the upper bound of Ω for fail-over on *perfect diagnosis.*

For the setting $\alpha_T = 0.7$ the results of Ω correspond to the independent outcome diagnosis where the α-count filtering has no effect (see Eq. 4.1). Note, that γ^1 is worse than γ^0 as expected. Observing the three graphs for increasing α_T, it is clear that all settings can provide some improvement over the one-shot diagnosis. The improvement for higher α_T is clearly caused by filtering out false alarms. A tendency for γ^0, $k = 0.92$ and γ^1, $k = 0.92$ is that there seems to be optimal settings where the highest gain in Ω_{model} can be achieved. However, for γ^0, $k = 0.99$ it seems multiple settings can be used without significantly affecting Ω_{model}. Studying for this setting the obtainable diagnosis metrics pairs for $\alpha_t = 1, \dots, 2.8$, μ_{RRT} increases from $19.2\,s$ to $29.9\,s$ while p_{RFA} drops from 0.22 to 0.12. It seems the improvement in p_{RFA} is cancelled out by the increase in μ_{RRT}.

Interestingly, with the temporal filtering the best gain can now be provided by the γ^1 setting $k = 0.92$ for $\alpha_T = 2.5$ where $\mu_{RRT} = 20.8\,s$ and $p_{RFA} = 0.11$. As mentioned previously, this gain is obtained as the improvement in p_{RFA} is relatively higher than the cost in increasing μ_{RRT}.

From these studied settings of α-count in the model it is possible to obtain 69.3% of the theoretical maximum for Ω_{model} compared to the no fail-over policy. Higher gains may be achieved for other obtainable pairs of TNR and TPR. However, as the primary aim of this work is to study the model sufficiency this task is left for future work of applying the model.

The conducted analysis has been repeated in the PE simulation model. The results are depicted in Figure 7.14. From these results it is clear that the same conclusions are made as in case of the model based results. Note, that quantitatively there is a difference in terms of Ω_{model} and $\Omega_{simulation}$, while they are strongly qualitatively alike. The quantitative difference can again be contributed to the fact that time is modeled stochastically in the model. This leads to higher variability in the model data transfer completion time estimates and consequently lower estimates of Ω. However, this aspect does not effect the qualitative performance of the model in terms of defining best settings of the studied temporal diagnosis component. Thus, the model is considered highly

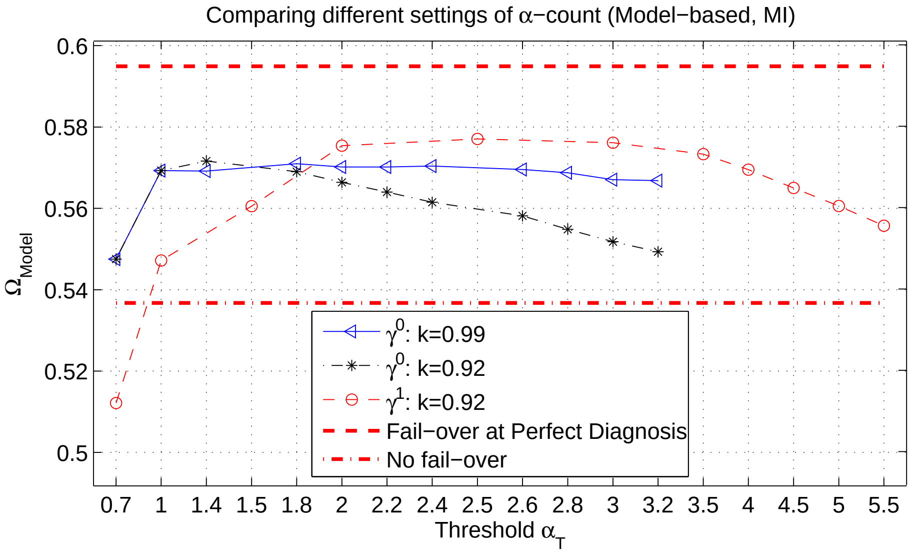

Figure 7.13: *Model results for reliability analysis of α-count for different settings.*

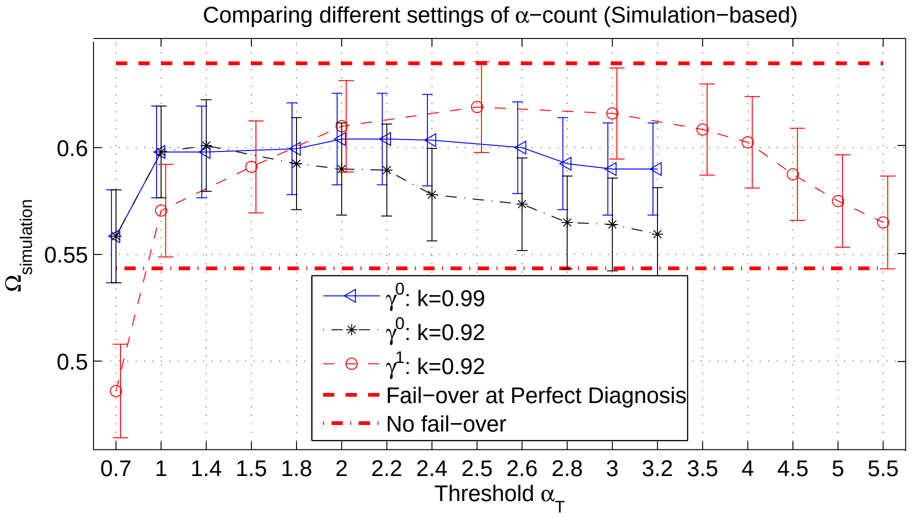

Figure 7.14: *Simulation results for reliability analysis of α-count for different settings. Confidence bounds of γ^0, $k = 0.99$ and γ^1, $k = 0.92$ have been shifted respectively left and right for improved visualization. .*

useful. It can be applied to evaluate the best settings that trade off imperfections of temporal diagnosis to provide improved remediation in the context of end-node driven fault management. Further, this is done without increasing the state space (and complexity) of the PE DTMC model initially defined.

7.5 Conclusion

In this chapter the interplay between the diagnosis and decision component has been studied. The main aim has been to consider how good remediation decisions may mitigate diagnosis imperfections and similarly how to identify qualified diagnosis settings trading off diagnosis imperfections for a fixed decision policy. These aspects have been considered for a time-constrained data transfer end-user service case attempting to optimize the reliability parameter of probability that a data transfer will complete within the time deadline. To gain insights into the decision problem and define a prediction model for use in the decision component, a DTMC based policy evaluation model (PE DTMC model) has been created. This model provides evaluation of best policies for remediation in optimizing the end-user service reliability and consists of the model-parts: end-user service model (simplified data transfer and time model), a parsimonious diagnosis model (network states and diagnosis capabilities) and remediation to an alternative network. To estimate SCTP data transfer distributions and mean throughput a self-contained SCTP congestion window based DTMC model has been proposed.

The four state parsimonious diagnosis model parametrization approach has been specified to support basic memory-less diagnosis mechanisms where independent diagnosis outcomes can be assumed. In addition, the model has been extended, without increasing its state space, to also capture essential properties of more complex, and potentially better performing, temporal diagnosis approaches with strong correlation in the diagnosis outcome. To achieve the latter, closed-form equations of representative diagnosis performance metrics (mean Remediation Reaction Time, μ_{RRT}, and probability of Remediation on a False Alarm, p_{RFA}) have been successfully established leading to: a) a sensitivity analysis of their impact on service reliability metrics for a given reliability problem, b) an assessment of their sufficiency to capture the main diagnosis properties and c) a model parameterization from traces of an implemented diagnosis component (in simulation).

Finally, the prediction models have been parametrized from an extensive ns-2 based system level simulation setup offering model parametrization and validation of the model results.

Main Results

The results from the PE DTMC model have been compared to the extensive system level simulation analysis. Due to a stochastic clock, model based estimates deviate quantitatively from simulation results. The model is, however, qualitatively good as it can be used to derive the same best-policy conclusions as obtained from considerably more tedious system level simulation results. For a perfect and imperfect diagnosis component it is shown how the best policy differs in relation to diagnosis capabilities. For some levels of diagnosis accuracy *no fail-over* is a better policy than failing over when a fault is diagnosed. Further, it is shown how waiting to enable fail-overs can improve reliability significantly. Although improvements in the studied scenario are modest (5.5-10 percentage points) under the given limitations it is demonstrated how including diagnosis capabilities in the remediation decision can be useful to improve reliability. The Markov model for optimal remediation strategy evaluation in this paper targets a specific SCTP upload application with completion-time constraints. Several aspects of the modelling approach can benefit also other application scenarios: 1) The inclusion of passed time as input parameter for the remediation decision. 2) The abstraction of the fault-diagnosis properties. 3) The abstraction of the SCTP congestion control behavior.

Based on the proposed diagnosis model a reliability analysis of an temporal heuristic (α-count) from [25] has been conducted in a scenario of time-constrained SCTP-based data transfers. Under a remediation policy of *fail-over at diagnosed fault state* the sensitivity analysis shows how both μ_{RRT} and p_{RFA} are essential diagnosis performance metrics. It is also shown how the model, despite its simplicity, may represent different transient behaviors of diagnosis metrics; however, a fixed parameterization approach based on steady state behavior of metrics is sufficient for the considered reliability study. The model based results show that the α-count diagnosis mechanism can provide significantly improved reliability compared to simpler diagnosis components not using time correlated observations. Moreover, it is shown how best diagnosis-settings can be identified maximizing reliability trading off μ_{RRT} and p_{RFA}. Finally, a comparison has been conducted of the model based analysis to an equivalent analysis performed in the extensive simulation setup. Despite expected quantitative differences the qualitative similarities are high. As a consequence, exactly the same conclusions regarding best diagnosis settings can be derived from the model as well as simulation. These results emphasize the usefulness of the proposed parsimonious model and the sufficiency of the chosen diagnosis performance metrics to capture essential diagnosis performance.

Future work

The study of this chapter has provided some basic insight into the decision problem under imperfect diagnosis. However, for the model to have relevance in a full setup, additional fundamental model functions are relevant to address: i) diagnosis of remediation options to assess their state (presumingly with imperfections), ii) decisions about collecting (actively and passivly) more information to improve diagnosis before making decisions to remediate, iii) multiple fail-overs during an end-user service, and iv) inclusion of multi-fault multiple remediation option scenarios. These functions clearly add to the complexity of policy evaluation. This encompasses defining good policy heuristics as well as obtaining fast and efficient model solutions. The first problem may be addressed by deriving optimal policies in and MDP like approach (for confined parts of the state space) while the latter requires a thorough analysis of how the model state space may be reduced further or re-used by solving the model for different parameterizations in independent policy evaluation cases.

For the extended model functions proposed in future work also the generality of the parsimonious diagnosis model should be studied further. This is relevant when evaluating more complex policies than in the case of temporal diagnosis approaches where remediation is initiated on a diagnosed fault event.

The PE DTMC model in its current form is also viable for future studies of formulating the optimization problem differently than optimizing reliability parameters. Also, new policies to minimize unneccesary hand-overs accepting a given drop in reliability could be relevant. This could be achieved by a threshold on the data transfer states.

Finally, future studies should be made for different end-user service reliability formulations to understand to which extent the results of this chapter can be generalized to these as well. Examples of such services could be: IP voice service where a certain quality must be maintained for a given fraction of time or arbitrary distributed services where function execution location (end-node or in the cloud) policies must be determined based on (imperfectly) diagnosed network states, reliability and resource cost.

Chapter 8

Decision Model Adaptation in Dynamic Network Scenarios

The preceding sections have focused on how to mitigate unreliable observations. Applied means have been to increase diagnosis robustness and make good remediation decisions taking into account the diagnosis imperfections and properties of the remediation options. The presented studies until now have been based on a static system setting where the network conditions are well known and do not change over time. In the highly dynamic environments of the end-node, assumptions of a static system view may only be valid for limited periods of time. Thus, the ODDR components and their sub-functionalities will need to adapt as changes occur. This is needed to obtain near-optimal or sufficient operation that can increase dependability to an acceptable level for the end-user service being executed. In this chapter, it is studied how the ODDR may facilitate the adaptation to changes. The primary focus lies on the models used in the *decision component* to derive decision policies. An equally important topic is how to properly adapt the diagnosis component model. However, for reasons of delimitation this topic is left for future work.

In the chapter, initially, an introduction is given to which categories of changes exist. Next, details on functions in the decision component to handle adaptation are presented. Finally, a study is presented on how to construct the prediction models. The study takes into account when adaptation is made and which presumptions are available on the change. This involves an implementation of the PE DTMC model principles in a Stochastic Activity Network (SAN) modelling framework, Möbius [36], which may act as support for the adaptation process.

8.1 Introduction to Changing Scenarios

In this section a brief introduction is provided introducing which parts of the operating scenario of the ODDR typically are expected to change over time. Based on these changes, requirements for adaptations in the end-node driven fault management setup are presented.

8.1.1 Changes in the End-to-End Path

From the end-node perspective, changes may apply to different system parts including *system parameters* (e.g. fault occurrence rates, channel settings, bandwidth assignment), *network configuration* (e.g. topology, available access networks) and the level of *information* of these properties. Not directly attributed to changes in the components of the end-to-end path, the information level refers to how well the properties of a component are known by the ODDR and can be used accordingly in the decision process. Thus, the information level is a property that may change over time as a component is used and more information about it is gathered.

Based on the scenario introduced in Figure 3.2 a set of high-level parameters has been identified representing components in and in relation to the end-to-end path as seen from the end-node perspective. These are depicted in Figure 8.1. Each of these parameters represent parts that can change. In the figure an example is given on how the parameters may be interdependently related. Details on the individual parameters can be found in Appendix F.

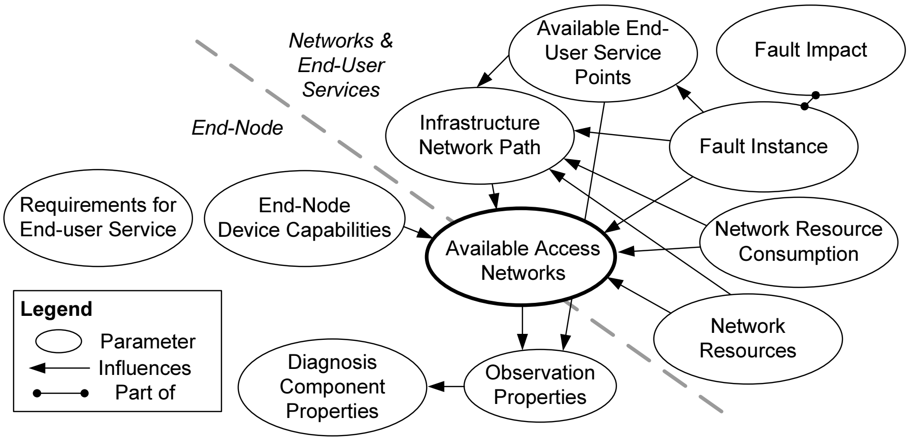
Networks & End-User Services
End-Node
Available End-User Service Points
Fault Impact
Infrastructure Network Path
Fault Instance
Requirements for End-user Service
End-Node Device Capabilities
Available Access Networks
Network Resource Consumption
Network Resources
Legend
Parameter
Influences
Part of
Diagnosis Component Properties
Observation Properties

Figure 8.1: *A high-level summary of parameters in the end-node driven fault management scenario and their dependencies as seen from an end-node device perspective.*

It can be utilized that a part of these parameters can be attributed to the *Available Access Networks*. Thus, a starting point of defining and representing a change is in the following study made from an access networks perspective.

8.1.2 Challenges for Adaptation in the ODDR

In the scenario of the end-node driven fault management many events can trigger a change in the availability of access networks and their properties. Examples of such triggers are end-node *mobility*, *evolution of the access options* in certain geographical locations (as private or public access points/base stations are installed and removed) and network load patterns. How to regularly adapt to such changes in the ODDR is an open question. It leads to some fundamental challenges shared with any autonomously adaptive system [74]. Some of these are:

Detection of prediction model insufficiency - Determining when adaptation is needed, is a function of how critical an inconsistency is to the end-user service reliability performance and the deviation between the real world system and the model state space representation and its parameters. Finding a good approach for identifying such model insufficiency can be crucial to ensure a good trade-off between the needed amount of costly adaptation tasks and the overall performance of the decision process.

Pre-filtering changes - As emphasized, changes may occur frequently. This leads to a significant amount of system information continually floating into the ODDR component. Assuming that functions of data processing become increasingly complex as information moves into the ODDR component modules, early pre-filtering is important. For example in an urban environment a mobile end-node will continually discover new access networks (even in a stationary setting). Pre-filtering by selecting only networks with good signal strength, networks that are known in advance or networks, which are expected to have the needed properties for a particular end-user service, can significantly reduce adaptation efforts.

Learn system properties - Learning in the ODDR context is a topic that independently applies to several functions. Learning may be used to determine system parameters that are not directly observable (e.g. amount of fault model system states [119]), system dependencies (e.g. which remediation action can solve which fault) or rules to autonomously construct prediction models. While learning may be considered a mean to provide adaptation it is not necessarily a prerequisite as discussed in Chapter 2.

Dynamically construct models - When system changes have been detected and system properties discovered, models used to obtain decision policies need to be re-computed from scratch or incrementally. This problem imposes sub-challenges on: i) how to dynamically build models for newly discovered environments, and ii) how to ensure that constructed models are efficient to solve, and iii) how to derive good or optimal policies in these.

Constructing models dynamically can be seen as a challenge of identifying system building blocks and rules on how to stitch these together depending on the discovered scenario. Model efficiency is depending on how large part of the system state space to include in the model. Also, it must be identified how including and excluding different parts of the state space may affect the properties of the derived policies.

In the remaining part of this chapter focus is on dynamic construction of models for policy derivation. Thus, with a starting point in the ODDR, an approach is proposed on how this could be realized. Using the basic mechanisms of the proposed model construction approach, an initial study is made on a change case of a *new access network arrival*. More specifically, implications on policy evaluation results are considered when including an expected change in the model in advance (leading to a larger state space) compared to waiting to adapt until the change occurs (leading to a smaller state space).

8.2 Adaptation in the ODDR

The main role of the Decision Component is to determine which actions (remediation, active observations, reconfiguration) should be initiated to optimize reliability of end-user services. In this section, a baseline architecture is introduced defining how adaptation could be enabled in the Decision Component and thereby the ODDR.

The key issue to obtain decision outcomes that can sufficiently improve end-user service reliability, is to ensure that decisions made are based on a correct and up to date system model. When the system model and real world scenario deviates to an *undesirable* extent, the system model needs to be updated. In this chapter it is studied how the adaptation of the prediction model may be conducted using online model generation. The role of the prediction model is to describe the expected progression of the system states including potential changes. This enables to identify the decision policies that are most likely to lead to a highly reliable operation of the end-user service. An ambitious goal could be to construct a prediction model that contains all anticipated changes and resulting policies in the system design phase. However, in a highly dynamic and ubiquitous environment, such as the one considered in ODDR context, this is clearly not a feasible option. Instead, the prediction model should represent only a part of the system and anticipated changes. Subsequently, the model should be updated to new system evolutions when needed. This is where online model generation comes into play. An online model generator must be capable of modifying the structure and parameters of the model. This is needed when system changes are discovered as the arrival of a new network or if new knowledge is obtained such as the maximum throughput of a particular access network. In fact, the model generator should always be able to provide autonomously a new model for the system, regardless of how and how much the networking environment has changed. For this reason, in the following we introduce a compositional modelling approach to generate models for different configurations of the system with a starting point in a set of basic atomic models.

8.2.1 Model Composition Approach

A Stochastic Activity Network (SAN) represents a high-level modelling formalism based on Petri Nets. The formalism is based on four graphical primitives which are: places, activities, input gates and output gates. In brief, places can contain zero to multiple tokens and represent states. Activities define how transitions of tokens take place over time, while gates describe changes in the amount of tokens for different places. Input gates further define when activities are enabled. SANs are a class of stochastic Petri nets and under certain assumptions (i.e., only exponentially distributed or instantaneous activities) the underlying stochastic process is a Continuous Time Markov Chain (CTMC); in this case analytical solution methods exist. More details can be found in references [36], [120]. One of the features of SAN is the ability to create a model of the system by composing atomic sub-models through the *join* and *replicate* operators. The join operator allows to compose (possibly different) sub-models by setting some places as shared; the replicate operator creates a predefined number of replicas of the same sub-model.

In the work of [26], this compositional modelling is exploited by the use

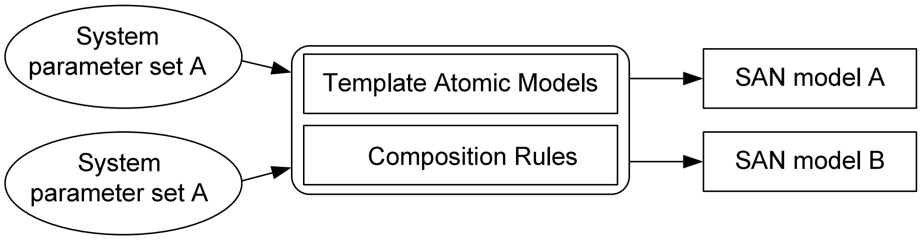

Figure 8.2: *Composition of different system models from pre-defined atomic model building blocks in accordance to a system parameter set.*

of parametric models following an approach, which recalls the Object-Oriented programming methodology. After identifying the basic *building blocks* of the relevant modelling domain, the respective parametric atomic models are created. Using multiple instances of these templates with different parameters and following specific compositional rules, a model for a wide range of possible system parameters can be obtained as depicted in Figure 8.2. Notice, the SAN model is an intermediate version of the prediction model, which can be converted into a Markov model, given the previously mentioned conditions can be met.

A similar approach can be used to implement online model generation in the ODDR framework. In particular, when a new model generation step is triggered, ad-hoc composition rules will allow to autonomously build the corresponding prediction model and from that establish a new active policy set. In this setting, the currently studied PE DTMC model and its sub-models can be directly translated into this approach considering individual models for: the end-user service, a network, an active diagnosis component and fail-over properties. Thus, relatively simple rules may be established. They must change the model to accommodate to new and disappearing networks, different diagnosis models for different faults and importantly, different end-user service models as the end-user changes the active service.

8.2.2 Decision Component Modules

In Chapter 3, the overall ODDR framework has been introduced and a brief overview of the Decision Component and its modules has been presented. In Figure 8.3, a more detailed realization of the Decision Component is introduced in the context of the model composition approach. It must be emphasized that the approach presented here still requires a significant amount of future work to become applicable. It, however, defines a setting for the subsequent studies of the model construction approaches.

In the following, a short introduction to the individual modules and their interfaces is provided.

Decision Manager - Its main role is to manage the information flow to the remaining modules. I.e. it is responsible for filtering out changes that are not relevant to the decision process. It also stores and updates a global set of system parameters (including end-user service requirements) that are used in prediction models and model generation process. Finally, it must be able to deliver information to the end-user service layer if end-user service requirements cannot be met.

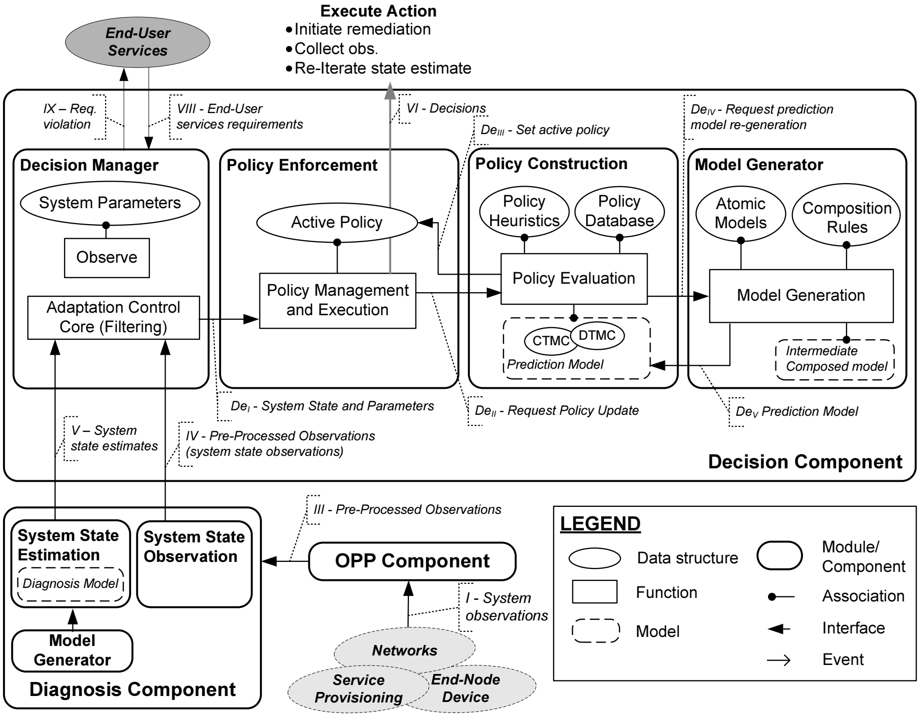

Figure 8.3: *Rich image of the ODDR components, their functions and Decision Component to enable decision making and adaptation to system changes.*

Policy Enforcement - The policy enforcement module is the policy engine of the Decision Component. This is where observations based on discrete or continuous, deterministic or probabilistic states (interface De_I) are mapped to an *Active Policy*. The active policy contains the currently applied policy rules for making decisions and rules defining its validity. In cases where adaptation is not required (i.e. observations are within the validity of the active policy) decision actions are executed as defined by the policy.

Policy Construction - The module Policy Construction performs the main task describes in the thesis as *policy evaluation* by evaluating a set of policy heuristics in the currently defined prediction model. A *policy database* can help to avoid re-calculating policies of already encountered system configurations.

Model Generator - The model composition approach is located in the Model Generator component. It uses system parameter information to generate models as the system changes or moves into states for which no policies have been derived. The *Intermediate Composed model* corresponds to the SAN model.

In brief, the adaptation loop in the Decision Component can be summarized as follows. Adaptation can be initiated if the *Active Policy* validity does not correspond to the observed system states. When the policy is no longer valid a new policy evaluation is triggered (De_{II}). In the Policy Construction Component a new active policy may be obtained from the policy database or from a re-evaluation in the *prediction model* if it includes the state space needed to calculate a new policy. If this is not the case, the Model Generator is activated (De_{IV}) to generate a new prediction model (De_V), which then will lead to a new policy evaluation and finally, a new active policy (to be set via De_{III}). This loop contains some of the inherent challenges in the task of adaptation as previously introduced. In the following sections, the impact of performing the prediction model generation in a proactive or reactive manner is studied. The change case of a new access network is introduced in the following section.

8.2.3 Change Case Study

As a background to study different prediction modelling approaches in the ODDR adaptation approach a *change case* is introduced in this section. A change case refers to a system change that requires adaptation and thus, a prediction model update. The starting point is made in the arrival of a *new access network*. An arriving new access network can be understood as a new access network that is discovered by the OPP component due to a mobility action or simply, a new access point that becomes available. This corresponds to a fault model scenario where an access point becomes available and certain properties are known about for how long it is expected to remain available. Finally, the arrival of a new access network may also in a next step be used to model changes of parameters of already available access networks at a certain point in time.

Study background
In the *new access network* change case the following aspects are in focus of the study.

Varying Network capabilities - Considering *new access networks* to have different capabilities in terms of performance and reliability it must be established how these parameters impact the decision process and overall end-user service reliability metrics.

Varying Time of Arrival - The *new access network* may appear before an end-user service is initiated or during an end-user service execution phase. It must be clarified when the properties of a newly arrived network are beneficial to improve reliability.

Information on Arriving Networks - Operating in unknown or known scenarios can make a difference on how much information an end-node has on the potential available networks that may arrive. This information can both be on when the network is expected to arrive and which properties it is expected to have. The impact of the availability of such information on decision policies and reliability parameters must be clarified.

The change case studies will, primarily, refer to these points. In the subsequent paragraphs, a scenario and basic assumptions made for the change case are introduced.

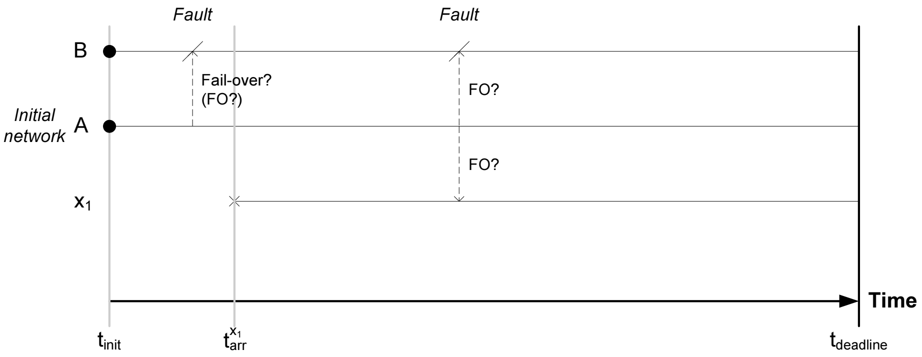

Figure 8.4: *Scenario of a new access network x_1 arriving during end-user service execution.*

Change case scenario and assumptions

A change case scenario has been specified with a starting point in the well known setup introduced in Section 3.3. An end-node is operating on a good or optimal strategy on using networks A and B as depicted in Figure 8.4. The end-user service is initiated at $t = 0$, defined as t_{init} where operation starts in network A in a normal state. During the end-user service executions a fail-over may be initiated to network B. Before ($t \leq 0$) or during an end-user service execution case, a new network, network x_1 appears representing new or other properties, which may or may not improve the end-user service reliability if included in the prediction model. This point in time is referred to as $t_{arr}^{x_1}$. This allows for new options regarding the fail-over and potentially also the re-calculation of policies or the activation of a pre-computed policy depending on the prediction model approach. To delimit the following study, a set of assumptions have been made about the new access network change case scenario.

Fault Model - For all networks the single congestion fault ON-OFF model is assumed.

Properties of Networks - Networks A and B are defined to have the same parameters as in the previously conducted studies of Chapter 7. However, different parameterizations of network x_1 are provided to study the decision impact for different cases. An introduction of the applied parameterizations is presented in Section 8.5.

Imperfect Diagnosis - For the diagnosis process on network A, imperfect diagnosis is assumed. A single setting assuming independent diagnosis outcomes for $\gamma^0_{mintime}$ in Table 7.2 is used (TNR=0.984, TPR=0.102).

Single Remediation Action - To ensure consistency to previous studies, only a single remediation action, i.e. fail-over, is allowed during the end-user service execution. As a result the decision problem becomes an issue of which network to fail-over to from network A.

Single New Access Network - The change case of this work includes different parameterizations of the arriving network x_1. However, it is assumed that there is only one new network that may arrive at a time. Thus, it is network x_1 that changes parameters. Also, when the new network has arrived it is defined that it will remain available throughout the service execution period (along with networks A and B).

Remediation Cost - In previous studies a cost of executing remediation has been implemented as the probability of a failing fail-over ($p_{fof} = 0.05$) and a random delay drawn from an exponential distribution (Mean $fdelay = 1.2\,s$). These properties have been maintained for the following studies and the same parameters are assumed for fail-over to network B as well as network x_1.

Arrival Time - The following analysis methods allow assessment on the decision policies for different arrivals times of x_1 (before and after end-user service initiation). The primary focus of the studies is on arrival during the end-user service execution phase to study end-user service state dependent adaptation.

Finally, a simple formalization of the new access network change case can be made considering different sets of access networks: The *World* (Φ) refers to all possible configurations of the arriving network that the end-node may encounter. Note, in the subsequent studies it is defined that $[A, B, x_1] \in \Phi$. Further, the set of *Available* (θ) refers to access networks that the end-node in practice can access where $\theta \subseteq \Phi$. The new access network change case refers to network x_1 being included in θ where network A and B already exist. These definitions are helpful when describing properties of models and changes in the following sections.

Delimitation to model based analysis
The new access network change case is studied based on an implementation of the PE DTMC model as a SAN model in Möbius. The primary focus lies on the different modelling approaches for the construction of the prediction model. Thus, the analysis is purely based on the model results. Clearly, in future work the obtained results must be validated in simulation and or in a realistic network setup.

8.3 PE DTMC Model SAN Version

This section describes the background on implementing the complete PE DTMC model of Chapter 7 in a Stochastic Activity Network version. This model will provide the background to investigate different realizations of the prediction model to handle policy evaluation in the new access network change case. First, the model is introduced and some details are provided on the differences from the PE DTMC. Then it is defined how policy evaluation is performed in the SAN version. Finally, the expected validity of the results of the SAN model is discussed based on the re-evaluation of policies previously made in the PE DTMC.

8.3.1 Möbius Implementation

The implementation of the PE DTMC model in a SAN is depicted in Figure 8.5 including the case of a new access network. This model is referred to as the

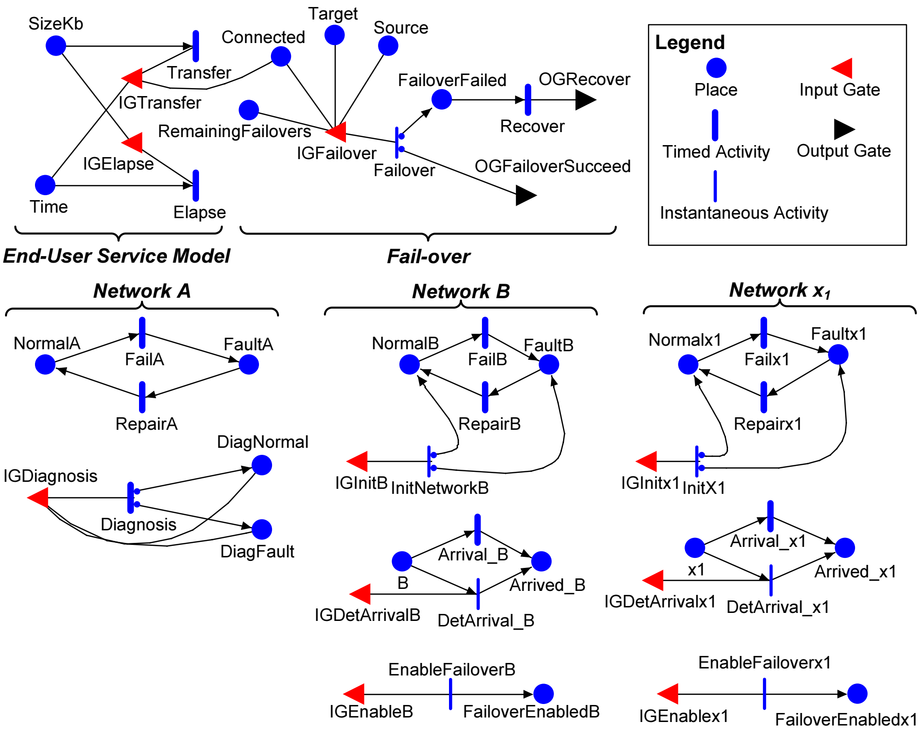

Figure 8.5: *Möbius implementation of the PE DTMC model in a SAN.*

Policy Evaluation SAN model (*PE SAN*). For simplicity the model has been implemented as depicted. This means that it has been implemented as a single atomic model. However, the model clearly includes several repeated structures corresponding to diagnosis, networks, etc., which as previously mentioned could be identified as atomic models that may be dynamically stitched together. In this work, this process has been conducted manually. Further studies on dynamic model construction are left for future work.

As depicted, the model is presented in five different parts: Network A, B and x_1, an end-user service model and the fail-over process. In the following paragraphs the individual model parts are described overall. Unless other is specified, all the activities (transitions) are exponentially distributed.

Network A

The column of model components under *Network A* consists of two main parts from bottom up: The diagnosis process, and the ON-OFF fault model, simply implemented as a token alternating between the normal and fault state. Note, as network A is defined to start in a normal state the place *NormalA* contains the token, initially. The PE DTMC model diagnosis process is based on a deterministic interval with the period T. In the SAN this interval has to be implemented to follow an exponential distribution. This ensures that the model can be solved using analytic methods. It still uses a mean period T. When the diagnosis activity is fired, a token is added to *DiagNormal* or *DiagFault* as defined by the TNR and TPR and conditioned on the true network state.

Note, in this setup the diagnosis process is only active on network A as only a single fail-over is allowed. An individual diagnosis model could be added to other networks for cases of multiple fail-overs.

Networks B and x_1
Networks B and x_1 are identical and here referred to as remediation networks. Considered from top down, as Network A they implement an ON-OFF fault process and additional logic to initialize it with some probabilistic distribution. This is typically the steady state probabilities as applied in the PE DTMC studies. Next part enables to control the arrival of the remediation network; either at a deterministic point in time (with an instantaneous activity) or probabilistically. The deterministic point in time is activated by the model internal stochastic clock (the *Time* place in the End-User Service Model). The lower model part of a remediation network controls when a fail-over to the network can be allowed. A fail-over is enabled when the network has arrived and a fail-over is allowed given by the remediation policy applied.

End-user service model
An end-user service model has been implemented in a similar manner as presented in Chapter 5. It contains both a stochastic clock and the data transfer progress. The resulting model structure of both data progress and time progress corresponds to a CTMC birth chain. As in the PE DTMC model the time or data progress processes are inhibiting each other depending on which first reaches the absorbing state (corresponding to the respective place containing no tokens).

The rate of the data state transition rate is governed by the state of the network in which the transfer is made (governed by the *Connected* place). The amount of data progress and time progress states have been set equal to the PE DTMC model (see Table 7.1).

It should also be noted, that for this model the mean transfer rate is used to define the property of the network states. Thus, the PE SAN model does not inherently include functionality like the cwnd-based data transfer model to estimate the mean transfer rate from the end-to-end path RTT and packet loss rate.

Fail-over
The final model part handles the fail-over between network A and remediation networks B and x_1. As in the PE DTMC model a fail-over failure may occur (*FailoverFailed* place) leading to a continuation of the transfer in network A after some penalty delay defined by the *Recover* activity.

8.3.2 Model Solution and Results

As in the PE DTMC model in the PE SAN model the Ω_{model} reliability measure is of interest. Ideally, the measure would be evaluated at steady-state. Möbius features different steady-state solvers, however, these are restricted from solving models with multiple absorbing states [120]. An alternative method is to perform an *instant-of-time* transient analysis with a sufficiently large time, assuming that the steady state has been reached. To obtain this the Ω_{model}

measure in the SAN model, is expressed as an *instant-of-time reward variable* with the following rate reward:

```
if(SizeKb->Mark()==0) {
    return 1;
}else{
    return 0;
}
```

This means that the variable should have the value 1 if the predicate holds (that the entire amount of data has been transferred), and 0 otherwise. By evaluating the reward variable mean at a sufficiently late instant of time, the probability of successful data transfers completion within the deadline is obtained. To select this instant of time the typical $t_{deadline}$ is used as a reference i.e. $t_{eval} = 10 \cdot t_{deadline}$ to ensure a minimal error. When all the activities in the SAN model are exponentially distributed or instantaneous, the stochastic process underlying the SAN is a CTMC, and can thus be solved analytically.

Policy Evaluation Approach
Reflecting the implemented SAN PE model back to the architecture proposed in Figure 8.3 the SAN model in Figure 8.5 corresponds to the *Intermediate Composed Model*. Further, the underlying state space generators of Möbius construct the CTMC used for the prediction model. In the following studies for simplicity the policies to evaluate are included in the SAN model and Möbius is used to apply the policies, define the state space, and perform the solving. In future work, it may however, be considered if generating a new CTMC is needed every time a new policy setting is to be assessed as currently done by Möbius.

Policy Evaluation Validation Results
The PE SAN model has been partially validated by making the same model evaluations as in Section 7.3 for the evaluation of remediation policies under imperfect diagnosis. Details on these results can be found in the Appendix in Section F.2. Summarizing the results, there seems to be a good match between the two result sets when comparing the evaluated policy evaluation results. Some numerical differences apply, however, where the PE SAN model produces lower values for Ω_{model}. These differences may partially arise from differences between the CTMC underlying the time and data transfer model and the DTMC of the PE DTMC model. However, these differences have not been studied in further details. Based on the results, it is assumed that the PE SAN model can provide policy evaluation consistently to the PE DTMC model.

8.4 Prediction Model Options

Using the presented PE SAN model structure as a starting point, in this section a set consisting of three modelling approaches is defined for the prediction model under adaptation. Common for the model approaches are that they handle the change case of a new access network arriving. The models, however, differ in whether the adaptation is handled *proactively before* an expected change or *reactively when* the change occurs. These two approaches are referred to as Modelled

Adaptation (MA) and Dynamic Adaptation Model (DAM). Our study is limited to a single DAM model approach; simply referred to as DAM, and two MA approaches: MA_{PEXP} and MA_{DA}. Which approach is more applicable, depends on which information is available to the adaptation process and/or which approach leads to the better trade-off of computations required and end-user service reliability. Thus, in the following sections the approaches are compared on: differences in policy outcomes, performance in a given change case scenario and some overall results on their state space. These studies are a preliminary step to understand how model generation rules should be established. The three models are presented in details in the following sections.

8.4.1 Modelled Adaptation - Probabilistic Expectation

The MA_{PEXP} approach is based on the assumption that the ODDR in advance can obtain knowledge about which new access networks (represented by their properties) may become available. This knowledge represents both the stochastic arrival characteristics and the rate which can be combined to a probabilistic expectation. Such information could be possible to obtain when the end-node may have some prior knowledge of the scenarios it is operating in. Using the terminology of sets introduced in Section 8.2.3, the MA_{PEXP} is assumed to include the entire world, Φ from the beginning, which in this view is t_{init}.

Properties

- Including a full-state view on the local area, the MA_{PEXP} has the advantage that derived policies can take expectations of changes into consideration. This may potentially lead to better policies and higher overall reliability.

- Requires that reliable estimates of the arrival probability of a new access network exist. These may not be easily obtainable.

- Including one or more expected networks in the prediction model may render policy evaluation too complex and intractable. This could become an issue if the MA_{PEXP} has been computed based on a wrong or insufficient definition of Φ and would need to be re-computed.

8.4.2 Dynamic Adaptation Model

The DAM model differs from the MA_{PEXP} model approach, as it waits to perform adaptation until the new network arrives. Thus, the DAM model makes the assumption that no knowledge of the new access network exists until the time point it actually arrives ($t^{x_1}_{arr}$). Thus, from the end-user service initiation time t_{init} to $t^{x_1}_{arr}$ the active policy (see Figure 8.3) for network A to B is used. Assuming the parameters of the new access network can be obtained at $t^{x_1}_{arr}$, a new active policy including x_1 can be derived based on a new prediction model. This prediction model now only needs to be derived in the interval from $t^{x_1}_{arr}$ to $t_{deadline}$. Note, it is assumed that solution time is zero in relation to the end-user service progress. In future work, a time penalty for model re-computation, should be included in the assessment.

Properties

- Does not consider expectation on arrival of a new access network regarding time and its parameters. Compared to the MA_{PEXP} this can help reduce the state space of the prediction model.
- Not including new access networks leads to re-assessment of new policies more often than the MA_{PEXP} model.
- As the *DAM* model does not make use of any pre-characterization of the arriving network, it is not sensitive to cases where there could be inconsistency between networks, which are expected to be in Φ and the actual networks encountered.
- The *DAM* model approach inherently conditions its evaluated policy on both the time and data transfer progress. This occurs as the new policy evaluation performed at $t_{arr}^{x_1}$ is based on a re-initialization of the prediction model to the actual system states.

8.4.3 Modelled Adaptation - Deterministic Arrival

The MA_{DA} model is defined as an intermediate modelling approach between the MA_{PEXP} and *DAM* models. It has been defined to bridge the understanding of these two opposite modelling approaches.

This approach assumes availability of knowledge about the properties of the arriving network before the end-user service initiation time t_{init} and can, thus, be applied to pre-compute policies. In correspondence to the *DAM* model the MA_{DA} model is, however, assumed to contain no probabilistic knowledge on the arrival of a new access network. Instead it evaluates the best policy for different deterministic arrivals of x_1. As in the case of the *DAM* model, it is assumed that the active policy from t_{init} to $t_{arr}^{x_1}$ is based on a scenario only consisting of networks network A and B with no expectation on the arrival of x_1.

Properties

- Does not consider any expectation on the arrival of a new access network.
- Assumes the new access network properties are known before end-user service initiation. This enables a pre-computation of the policy for different arrival times of network x_1.
- In comparison to the *DAM* model, the policy evaluation is only conditioned on the time progress given different arrival times of network x_1.

8.5 Model Comparison Background

After having defined the three modelling approaches, the next step is to compare their properties in terms of the policies they produce and the resulting impact on reliability of the end-user service. In this section, the individual model evaluation approaches are introduced. Next, a set of x_1 networks is defined to represent different interesting cases for policy evaluation. Finally, details on how the individual model approaches are evaluated in the previously defined Möbius model are given.

8.5.1 Evaluation Approach

The model evaluation approach is split in two parts: *Policy Evaluation* and *Scenario Evaluation* as presented in Figure 8.6. The aim of policy evaluation is to determine the best policy for a given x_1 access network arriving. Recall that the model generation in this evaluation inherently includes the construction of the prediction model. In *policy evaluation*, Ω_{PE} values are calculated internally for each individual model approach to compare the policy heuristics and identify the best. In *scenario evaluation*, the individual model approaches and their best policies are compared in a scenario. The scenario is characterized by a particular x_1 network configuration and an arrival rate of the new access network. The scenario must enable a relative comparison of the modelling approaches to assess Ω_{model}.

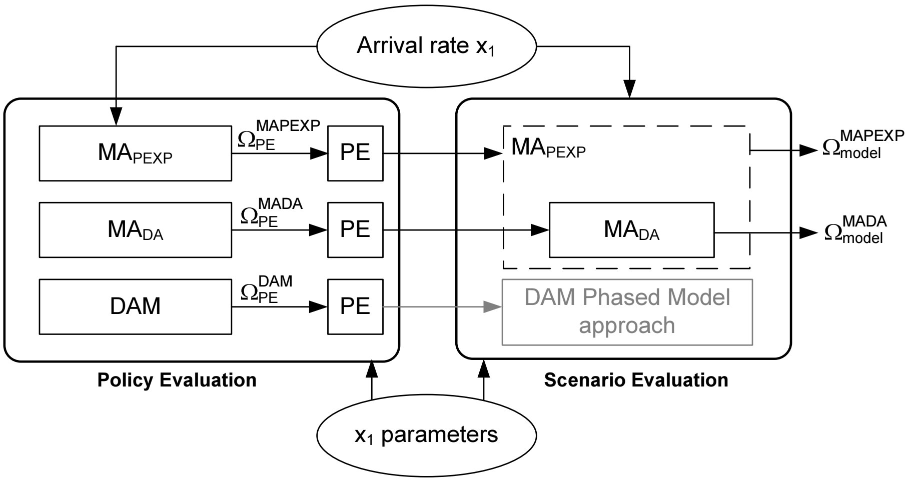

Figure 8.6: *Evaluation approach encompassing policy evaluation and evaluation in a scenario with best policies.*

Model based results for the scenario and the three prediction model generation approaches are obtained in the following way. The MA_{PEXP} is based on knowledge on the probabilistic arrival, already in the policy evaluation. Thus, its policy evaluation setup and the scenario are equivalent. The MA_{DA} approach leads to a policy set which can be assessed in the same setup as MA_{PEXP}. Finally, the DAM scenario evaluation needs to be conducted in a phased modelling approach. This is necessary as the policy changes during the end-user service execution. For reasons of delimitation this approach is however, not considered in details in the following comparisons. Thus, these evaluations are left for future work. Instead, the DAM results are compared to the MA_{DA} and MA_{PEXP} at the policy evaluation level.

Further information on how the policy evaluation and scenario evaluation is conducted for the individual models is provided after the following section where the different x_1 network types to study, are introduced.

8.5.2 Arrival Networks

To conduct a study comparing the individual modelling approaches of the new access network change case, a set of x_1 networks with different parameters are introduced. Each of these networks are studied individually as the newly arrived network.

To obtain an interesting set of x_1 networks, several configurations have been studied. The study approach has been to examine a setup of fail-overs from network A to x_1 where x_1 is available from t_{init} (replacing network B). Starting from parameters of network B, new parameter sets for x_1 have been empirically sought. They provide other $\gamma_{mintime}$ settings and stereotypes of slightly and significantly better/worse networks. For reasons of delimitation only a part of these have been selected. They all allow some improvement over network B to ensure they will lead to a different policy than fail-over to network B. Their parameters are depicted in Table 8.1. Model based study results as conducted in Section 8.3 under policy **(PIII)** (see Section 7.3.3) are further depicted in Figure 8.7. The selected parameterizations are:

$\mathbf{x_1^\alpha}$ - The setting is based on an increase in the fault/normal occurrence rate compared to network B while maintaining the mean throughput of normal and fault states. This network setting is more attractive than that of network B as higher Ω_{model} results are obtainable. The reason is expectedly that the risk of ending up in a long fault period is reduced in the time limited window of the end-user service execution. This leads, for a policy **(PIII)** on network x_1, to a $\gamma_{mintime}^{x_1^\alpha} \approx 9.99$. I.e. a fail-over becomes more attractive sooner than for network B.

$\mathbf{x_1^\beta}$ - The setting is based on a decrease in the fault/normal occurrence rate compared to network B. Opposite $\mathbf{x_1^\alpha}$ this negatively affects the obtainable Ω value compared to network B. To compensate for this aspect, the fault state mean throughput has been increased to reduce its negative impact. The result is a network $\mathbf{x_1^\beta}$, which also can provide improvements over network B at a $\gamma_{mintime}^{x_1^\alpha} \approx 6.66$.

$\mathbf{x_1^{SUPER}}$ - Finally, a super network is defined for reference, which for sure is significantly better than network B by multiplying the fault/normal state mean data transfer rates by 1.5. The outcome is a network, which always can provide significantly better results than network B leading to a $\gamma_{mintime} = 0$.

Evaluated from t_{init}, all of the selected networks lead to $\gamma_{mintime}^{B} > \gamma_{mintime}^{x_1}$. Cases where $\gamma_{mintime}^{B} < \gamma_{mintime}^{x_1}$ is true, while still providing higher reliability over network B have not been identified. If such cases exist has not been clearly determined. However, an intuitive argument that this could be the case is, that a better network than B will also be more beneficial to fail-over to earlier. Such relations could, however, change if other than steady state information was available for the fail-over network e.g. if remote diagnosis was enabled.

Scenario Settings

To define different scenario settings, arrivals of x_1 are considered to occur after an exponentially distributed time after t_{init}. The arrival rate parameter is

Parameters	**A/B**	$\mathbf{x_1^{\alpha}}$	$\mathbf{x_1^{\beta}}$	$\mathbf{x_1^{SUPER}}$	**Unit**
Repair rate - Λ_r	0.0667	0.202	0.0167	0.0667	$occ./s$
Fault occ. rate - Λ_f	0.0805	0.242	0.0202	0.0805	$occ./s$
Mean transfer rate (normal state) Λ_{dtn}	453	453	453	679	KB/s
Mean transfer rate (fault state) Λ_{dtf}	220	220	265	330	KB/s
SS prob. normal s.	0.45	0.45	0.45	0.45	-
SS prob. fault s.	0.55	0.55	0.55	0.55	-
$E[\Lambda_{dt}]$	324	324	350	487	KB/s

Table 8.1: *Parameters of different x_1 access network configurations.*

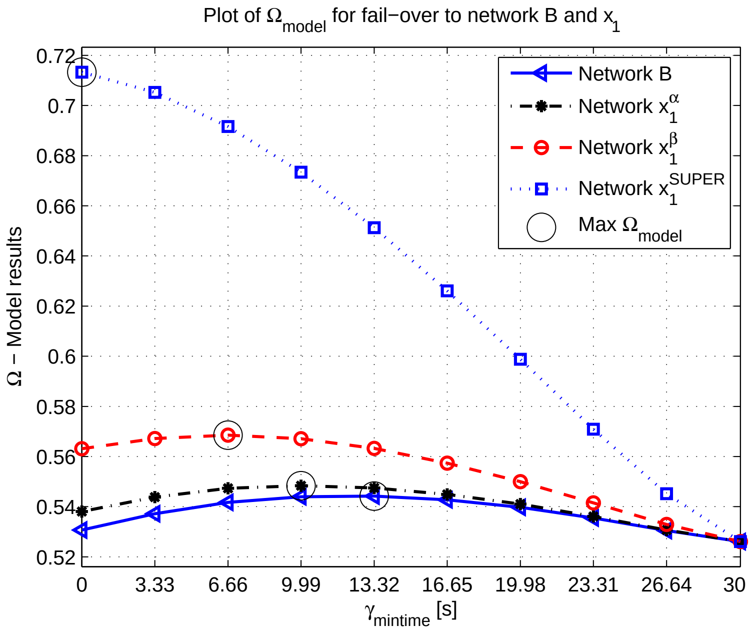

Figure 8.7: *Different x_1 network configurations in comparison to network B in a setting of the traditional $\gamma_{mintime}$ policy and fail-over from network A to x_1/B.*

defined as: Λ_{x_1arr}. Three different rates have then been specified with a starting point in the required maximum end-user service duration time of $t_{deadline} = 30\,s$ to form an arrival rate of $1/30\,arrivals/s$ (defined as a medium rate) and fast (factor 3 increase) and slow (factor 0.5 decrease) rates:

$$\Lambda_{x_1arr}^{fast} = \frac{1}{10}\,arr/s \qquad \Lambda_{x_1arr}^{medium} = \frac{1}{30}\,arr/s \qquad \Lambda_{x_1arr}^{slow} = \frac{1}{60}\,arr/s \quad (8.1)$$

8.5.3 Policy Heuristics

Introducing a new access network arrival change case leads to explore new policy heuristics on which network to fail-over to. To delimit the scope of the subsequent studies, new policy heuristics are based on extending the policies already introduced in Section 7.3.3. The starting point is made in policy heuristic **(PIII)** *Minimum time threshold*. To include fail-over to either network B or network x_1 the policies $(\mathbf{PIII^B})$ and $(\mathbf{PIII}^{x_1})$ are introduced. They define a fail-over to network B using the threshold $\gamma^B_{mintime}$ or a fail-over to network x_1 using $\gamma^{x_1}_{mintime}$. It must be emphasized that either policy is also conditioned on the availability of the network to which it is associated. I.e. a fail-over to network x_1 can only be commenced given it has arrived. How these policy heuristics

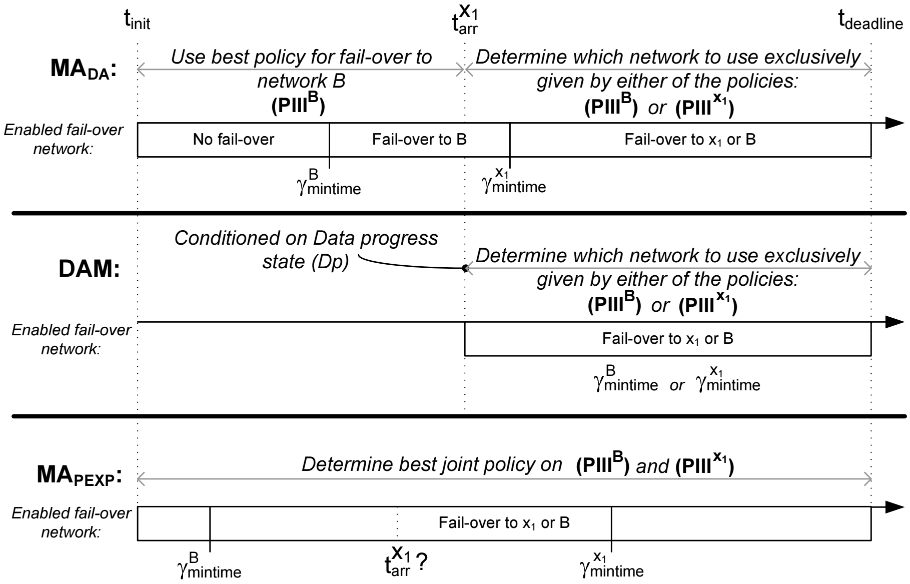
t_{init}
$t^{X_1}_{arr}$
$t_{deadline}$
MA_{DA}:
Use best policy for fail-over to network B (PIII^B)
Determine which network to use exclusively given by either of the policies: (PIII^B) or (PIII^{x1})
Enabled fail-over network:
No fail-over
Fail-over to B
Fail-over to x1 or B
$\gamma^B_{mintime}$
$\gamma^{x_1}_{mintime}$
DAM:
Conditioned on Data progress state (Dp)
Determine which network to use exclusively given by either of the policies: (PIII^B) or (PIII^{x1})
Enabled fail-over network:
Fail-over to x1 or B
$\gamma^B_{mintime}$ or $\gamma^{x_1}_{mintime}$
MA_{PEXP}:
Determine best joint policy on (PIII^B) and (PIII^{x1})
Enabled fail-over network:
Fail-over to x1 or B
$\gamma^B_{mintime}$
$t^{X_1}_{arr}$?
$\gamma^{x_1}_{mintime}$

Figure 8.8: *Policy evaluation under the different prediction model construction approaches.*

are interpreted and evaluated in the individual prediction model construction approaches is depicted in Figure 8.8. For use in the following policy evaluation descriptions, a notation of the best policy for fail-over from network A to network B is defined (under the assumption of x_1 never arriving) as: $(\mathbf{PIII^B_{BP}})$. It refers to a policy where $\gamma^B_{mintime} = 13.32$ as seen in Figure 8.7.

MA_{DA} - Policy evaluation in this model is made under the assumption of the *DAM* model, that the arrival of network x_1 is not known until it appears. As a result, $(\mathbf{PIII^B_{BP}})$ is used until $t^{x_1}_{arr}$. After the arrival of x_1, it is determined if this access network can provide better results in the remaining period by evaluating $(\mathbf{PIII}^{x_1})$ conditioned on that $(\mathbf{PIII^B_{BP}})$ has been applied until $t^{x_1}_{arr}$. In the period from $t^{x_1}_{arr}$ to $\gamma^{x_1}_{mintime}$ fail-overs to network B are allowed while from $\gamma^{x_1}_{mintime}$ to $t_{deadline}$ fail-overs to x_1 may be commenced. Initial experiments on restricting fail-overs to network B in the period from $t^{x_1}_{arr}$ to $\gamma^{x_1}_{mintime}$ showed no difference in the best policy of $(\mathbf{PIII}^{x_1})$.

DAM - Introducing a re-computation of the prediction model during end-user service provisioning, the *DAM* model can be re-computed taking into account the actual data progress state. The *DAM* policy evaluation is conducted assessing and applying the best policy of $(\mathbf{PIII^B})$ and $(\mathbf{PIII}^{x_1})$ in the period from $t_{arr}^{x_1}$ to $t_{deadline}$.

MA_{PEXP} - The policy evaluation approach is similar to MA_{DA} with the main difference that the prediction model includes a probabilistic expectation on the change. The outcome is that the best setting of $\gamma_{mintime}^{B}$ now also depends on the arrival of x_1. A joint evaluation of $\gamma_{mintime}^{B}$ and $\gamma_{mintime}^{x_1}$ is conducted with a precedence to fail-over to network x_1 if fail-over to both networks is enabled by the $\gamma_{mintime}$ thresholds.

Limitations of the studied policies
The studied policy heuristics represent a delimited set and may not contain the optimal policy. Some of the limitations to be explored in future work would be:

- Include a policy enabling immediate fail-over as network x_1 arrives. This could be relevant for a superior network like x_1^{SUPER}.
- Determine if a joint policy of fail-over to x_1 and B after x_1 has arrived. I.e. could it make sense to fail-over to B for a certain period and x_1 for another.

8.6 Policy Evaluation Results

In this section comparisons on the policy evaluation outcomes of the introduced prediction models and the applied policy evaluation approaches are provided.

8.6.1 MA_{DA} Policy Evaluation

A central aim of policy evaluation in the MA_{DA} prediction model is to identify the best policy setting of $\gamma_{mintime}^{x_1}$ under $(\mathbf{PIII}^{x_1})$ and clarify if it provides an improvement over what can be achieved under $(\mathbf{PIII_{BP}^B})$. Recall, that the evaluation of $(\mathbf{PIII}^{x_1})$ is derived assuming the use of $(\mathbf{PIII_{BP}^B})$ in the interval from t_{init} to $t_{arr}^{x_1}$.

For the studied policies, fail-overs to network B are not dependend on the arrival time $t_{arr}^{x_1}$, but rather on $\gamma_{mintime}^{x_1}$. In practice, this means that the policy evaluation can be based on a study where only $\gamma_{mintime}^{x_1}$ is varied and $t_{arr}^{x_1} = 0$. Thus, a settings where $t_{arr}^{x_1} > 0$, will not change the policy evaluation results.

Policy evaluation in the MA_{DA} are depicted in Figure 8.9. It represents the case of network x_1^{β}. To keep results consistent to the MA_{PEXP} evaluation, in the following section, not only $(\mathbf{PIII_{BP}^B})$ is evaluated but also different settings of $\gamma_{mintime}^{B}$. Focusing initially on $(\mathbf{PIII_{BP}^B})$ (where $\gamma_{mintime}^{B} = 13.32\,s$), it is clear that network x_1^{β} maintains its global best policy irrelevant on when it arrives or becomes available. This means that a fail-over to network x_1 should always be commenced from $\gamma_{mintime}^{x_1} = 6.66$ and onwards, if network x_1^{β} has arrived. These results indicate that a conditioned evaluation is not needed to identify the best decision strategy. An independent evaluation of the arriving network as in Figure 8.7 lead to the same conclusions.

From the results in Figure 8.9, another observation can be made. Provided that $(\mathbf{PIII^B})$ would be performed by assuming that a deterministic arrival time (or $\gamma_{mintime}^{x_1}$ setting) of network x_1 is known, then the best setting of $\gamma_{mintime}^{B}$

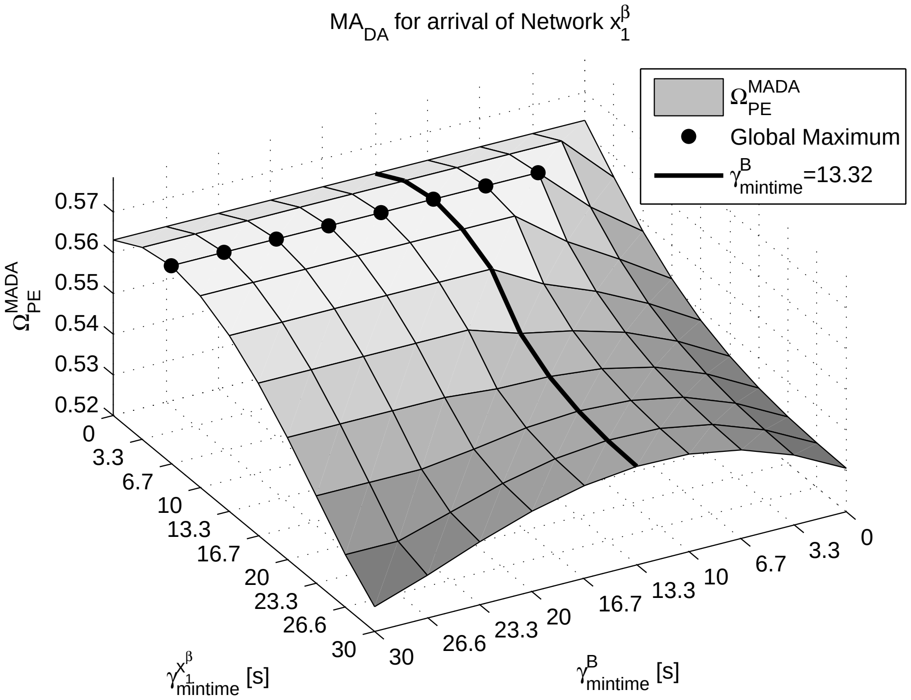

Figure 8.9: *Comparison under MA_{DA} of different $\gamma_{mintime}$ settings for fail-over to network B versus network x_1^{β}.*

is different from $(\mathbf{PIII_{BP}^{B}})$. An example assuming an arrival of $t_{arr}^{x_1} = 23.32$ (also corresponding to $\gamma_{mintime}^{x_1} = 23.32$) the setting $\gamma_{mintime}^{B} = 23.32$ ensures that fail-overs to network B are restricted further. However, also in this case the same conclusions can be obtained from the independent evaluations of the two networks.

The general conclusions made here for x_1^{β}, are also consistent with what has been observed for x_1^{α} and x_1^{SUPER}.

8.6.2 MA_{PEXP} Policy Evaluation

Policy evaluation results for MA_{PEXP} have been made under the same settings as in the case of MA_{DA} with the additional assumption that arrivals are probabilistically expected. These results are depicted in Figure 8.10 for different arrival rates using x_1^{β}. Compared to MA_{DA} in the model now only a single global maximum of Ω_{PE} exists for any of the arrival rates. As expected, this maximum is sensitive to the arrival rate. For $\gamma_{mintime}^{B}$ there is an increase in its best setting for increasing rates of Λ_{x1arr}. I.e. it makes sense to be a bit more conservative in the fail-over to network B waiting for the better x_1^{β} to arrive compared to cases without expectation.

Studying the $\gamma_{mintime}^{x_1}$ policy setting, the gains of waiting to fail-over seem to have been significantly reduced. This is observed from a flatter characteristic of Ω_{PE}^{MAPEXP} for low values of $\gamma_{mintime}^{x_1}$. An explanation may be that the probability of an early arrival of x_1^{β} is relatively low, compared to the cumu-

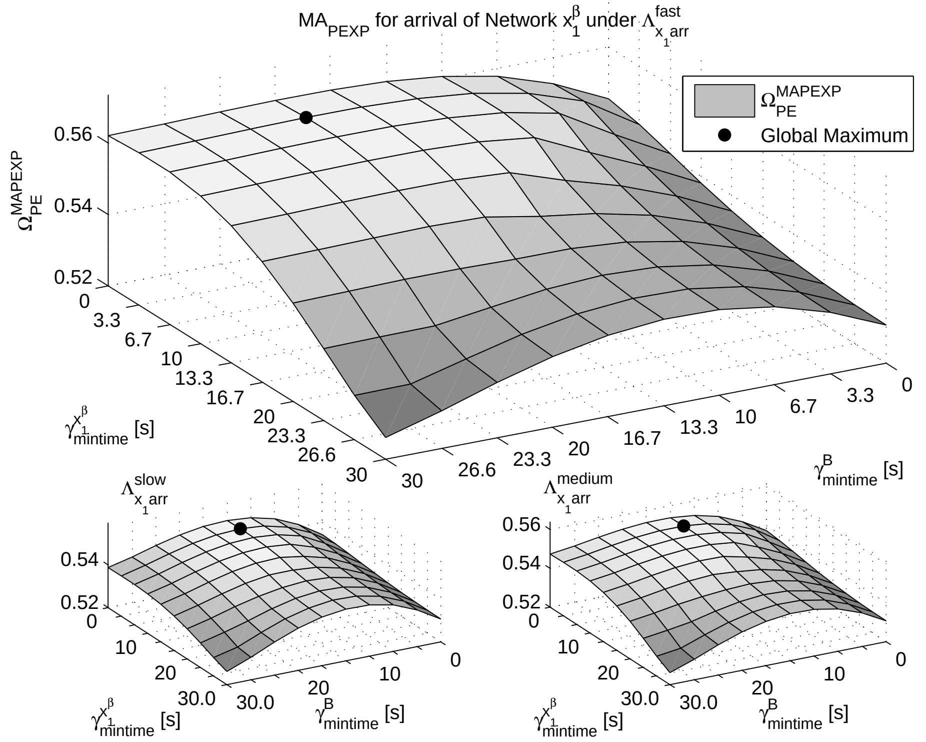

Figure 8.10: *Comparison of different $\gamma_{mintime}$ settings for fail-over to network B versus network x_1^β with under varying arrival rates.*

lative probability than an arrival has occurred later in the transfer. Thus, the sensitivity to changes in $\gamma^{x_1}_{mintime}$ is low early in the transfer and increases with time; even for the high arrival rate scenario. This is supported by the results showing that there are still small gains in waiting on $\gamma^{x_1}_{mintime}$. In most cases a decrease of a single time state has been observed compared to results in Figure 8.7. The reason has not been clearly determined. However, the impact overall seems to be of little significance to Ω_{PE} under the studied network types.

Also MA_{PEXP} conclusions have been found to be consistent for the different x_1 access network configurations.

8.6.3 *DAM* Policy Evaluation

The *DAM* prediction model represents a reactive approach where the policy is recalculated at a certain time point during the end-user service execution. This results in an adaptation approach where the policy can be re-calculated based on the actual or estimated system state. The following results must clarify if this impacts the previously derived policy in the comparable MA_{DA} case. The *DAM* model re-computation is not performed until the new access network arrives. Thus, it must be derived on the basis that in the interval from t_{init} until $t^{x_1}_{arr}$ ($\mathbf{PIII^B_{BP}}$) is applied.

For the *DAM* approach, policies have been calculated conditioned on both the data transfer state and the time state. This tuple is defined as the arrival state of x_1. As a consequence the policy also only needs to be calculated starting from the time of x_1 arrival as seen in Figure 8.8. In practice, the DAM policy evaluation is conducted by initializing the PE SAN model in a given start state. This start state is defined by:

(Dp, Tp) - These are the states on which the policy evaluation is conditioned where: $Dp \in [1 \dots 26]$, $Tp \in [1 \dots 10]$ and $Dp = 26$, $Tp = 10$ are the absorbing states.

(Ds) - It is assumed that diagnosis is re-initialized to diagnose no fault (negative state) for each evaluation.

(Nw) - Only evaluation for transfers which are active in network A is relevant. This is the case as only a single fail-over is allowed and a transfer always starts in network A.

(Ns) - This state is the most complicated to specify as it is not directly observable. Instead, it may be initialized as a probability distribution over its states. In a deployment of this approach in the ODDR, an HMM may be used to obtain the most probable state distribution at $t^{x_1}_{arr}$. How it is obtained for these studies is described in the following.

The network state distribution can be derived from the PE SAN model at different arrival times of network x_1. Note, as the network state process in the model is represented by true time and the arrival process is based on the stochastic clock, this may lead to some result irregularities. The normal state probability under different conditions is depicted in Figure 8.11. Note, only the network states of active transfers still in network A are relevant. Observing initially $P(Ns = Normal|Nw = A, Tp = x)$ this is the transient behavior of the ON-OFF process. Conditioning it on active transfers, it is seen how the normal state probability increases after $\gamma^B_{mintime} = 13.32\,s$. This is due to $(\mathbf{PIII^B_{BP}})$ where remaining transfers are more likely to be in the normal state. It can be observed that the experienced network state also is sensitive to the data transfer progress state. Thus, in the DAM evaluation the results of the conditional $P(Ns|Nw = A, Dp \in [1 \dots 25], Tp = x)$ are used.

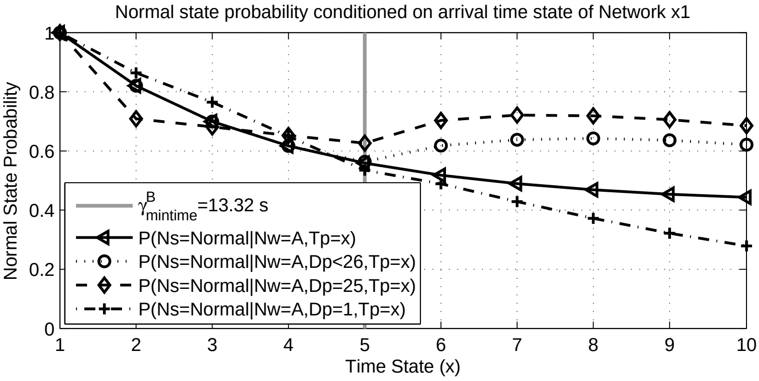

Figure 8.11: *PE SAN model predicted normal state estimates for different arrival times of network x_1 conditioned on active transfers in network A.*

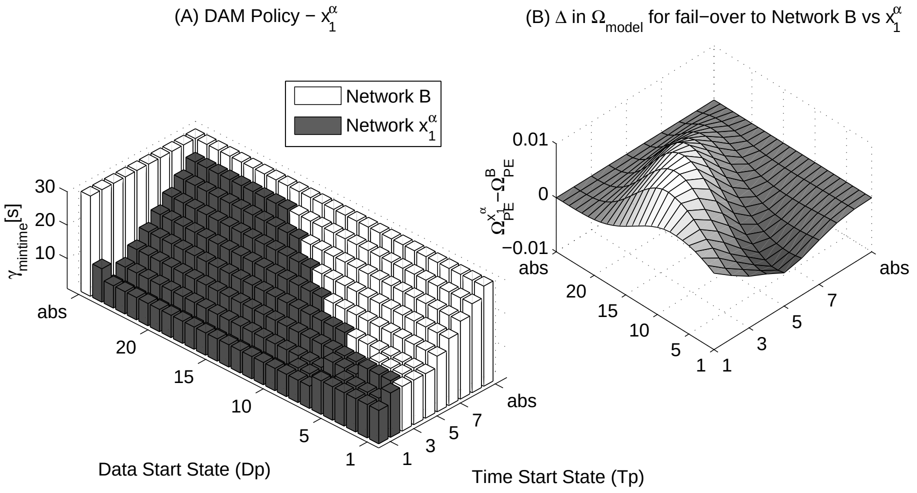

Figure 8.12: *(A) Best policy as a function of data transfer progress and time progress starting states. (B) Difference in Ω_{PE} for using network B versus network x_1^α.*

All of the x_1 access network configurations have been evaluated in the DAM model approach. For reasons of brevity results are only presented for network x_1^α. The outcome of the policy evaluation is depicted in Figure 8.12 (A). It represents the applied policy ($(\mathbf{PIII^B})$ or $(\mathbf{PIII}^{x_1})$) and its setting based on which provides the best Ω_{PE} for the particular starting state of (Dp, Tp). The stair effect occurs from letting $\gamma_{mintime}$ to follow the particular starting state if no additional waiting is beneficial.

The results show that it may be beneficial to fail-over to network B for some cases and x_1^α for others. This effect has only been observed for the network x_1^α configuration which performance wise is closest to network B. In the other cases network x_1 is preferred for all cases. As shown in figure 8.12 (B), depicting the difference of applying the two policies, it is, however, also clear that network B does not offer a significant advantage.

For time progress states where a waiting policy is relevant $Tp \in [1 \ldots 4]$, a tendency is observed that as the data progress state increases a smaller $\gamma_{mintime}$ can be accepted. An interpretation under x_1^α is that a fail-over to a potential fault state becomes less critical if there is not much of the data transfer missing and still substantial time before the deadline.

In summary, these results show that the DAM model does lead to different policies depending on the data progress state. It is yet to be clarified if these differences will lead to an improvement over MA_{DA} policies in the final Ω_{model} results. It should be noted that states where the DAM model leads to interesting variations are expectedly also the least likely to occur: i.e. that a lot of the time has passed and little of the data has been transferred or opposite. As a result the obtained policies expectedly lead to reliability results similar to what can be achieved using MA_{DA}. To clarify these aspects in future work the DAM modelling approach must be implemented in the scenario evaluation as well (see Figure 8.6).

	MA_{PEXP}	MA_{DA}	DAM	A, B and A, x_1^{β}
#Cases	100	100	2600	20
#States *Min*,*Max*	2072, 17176	1036, 10444	5, 6267	1036,6266
#States μ, σ	10342, 3438	5500, 1879	1278, 1183	3871,1627

Table 8.2: *State space statistics for calculation of policy evaluation results under arrival of network x_1^{β} as provided by Möbius. For MA_{PEXP} results, Λ_{x1arr}^{medium} is used.*

8.6.4 Model State Space Results

A representative comparison on the computational resources required for the modelling approaches is expectedly highly scenario dependent. I.e. a computationally expensive proactive approach may not be expensive over time if re-computations are rare. A reactive approach may in the same setting need to perform many re-computations, overall, being the more expensive solution. Such considerations have been left for future work. However, to provide a basic idea on how the different model approaches compare, overall state space statistics are presented in Table 8.2 for arrival of network x_1^{β}. These results are based on the state space that Möbious generates for each individual policy setting studied.

The number of cases relates to the amount of policy settings to be evaluated. This number depends on the state space chosen for time progress (Tp) (to calculate $\gamma_{mintime}$ results) for all model approaches. The DAM model, in addition, depends on the data progress states (Dp). This results in significantly more cases to evaluate if the DAM is pre-computed.

Focusing on the mean state space, the DAM model in general can offer the smallest, as it can scale to the state space of the end-user service. In a scenario setting, it will, however, be sensitive to when network x_1 arrives and the progress of the end-user service. Also, the independent evaluation of policies for networks A, B and networks A, x_1^{beta} leads to a relatively small mean state space amount. Clearly, the MA_{DA} evaluation is more costly due to the conditional evaluation of policies for fail-overs to networks B and x_1^{β}. Finally, the probabilistic expectation in MA_{PEXP} leads to the largest state space.

8.7 Scenario Comparison Results

Concluding the comparison of the prediction model approaches Figure 8.13 shows the Ω_{model}^{MAPEXP} in comparison to Ω_{model}^{MADA} for the studied x_1 configurations and different arrival rates. As all the studied x_1 network configurations provide some improvement over network B, an increase in Ω_{model} can be observed. Now studying the gain in including probabilistic expectation in the prediction model the improvement is limited for the studied configurations. This should be seen in relation to a significant model size and complexity increase of the model MA_{PEXP} model compared to the MA_{DA} model; especially when a solution to the latter does not require a conditional solution of the available fail-over networks. Finally, expecting the DAM model to perform similarly as well this means that proactively pre-computing the new access network arrival will not

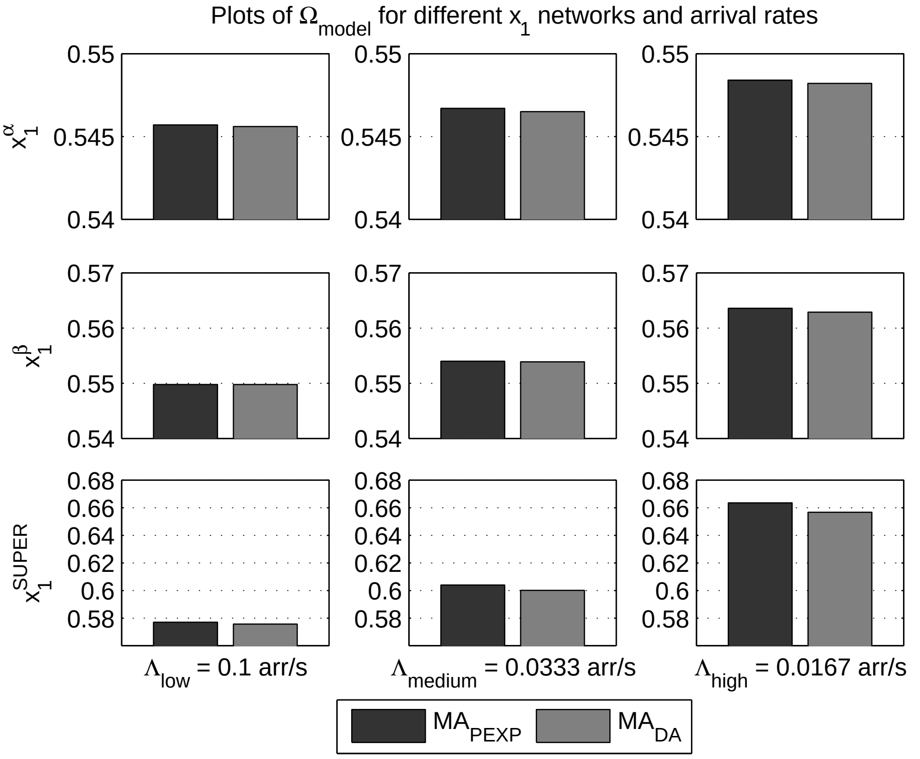

Figure 8.13: *Comparison of scenario evaluation results for varying x_1 network configurations and arrival rates. Including knowledge on the probabilistic arrival has a limited impact on the $\Omega_{modelresults}$.*

provide a significant advantage in these cases.

8.8 Conclusion

Adaptation of models used for prediction of best remediation policies is required for the end-node to operate in a near optimal setting. A proposal on how such model adaptation may be performed takes advantage of the already modular structure of the PE DTMC model. Utilizing a SAN model representation model parts may easily be removed, added or modified to adapt to new environments. A manual model construction approach has been made studying the arrival of a new access network.

An open question is how this model construction should be organized in terms of a proactive (Modelled Adaptation, MA) versus a reactive adaptation approach (Dynamic Adaptation Model). A pre-study has been conducted on the comparison of two proactive model approaches (MA_{PEXP} and MA_{DA}) and a single reactive (DAM). They must adapt to the arrival of a new access network x_1 during an end-user service execution. The proactive approaches are based on the assumption that properties of x_1 are known leading to pre-computation of policies. MA_{DA} is based on no knowledge on the arrival of x_1 while MA_{PEXP} assumes the arrival can be probabilistically known. The reactive DAM model, refrains from making any of these assumptions but uses the advantage of knowing the exact state of the end-user service at the time of x_1 arrival to make the best policy from this point. Under the studied scenario it has been shown that the DAM evaluation approach leads to different policies conditioned on a particular end-user service state. For a large part of the state space, these policies are however, consistent with the MA_{DA} policy evaluation results, which can be obtained from a relatively small model state space. Including expectation in policy evaluation the MA_{PEXP} model does lead to different policies compared to MA_{DA} demonstrating additional gains in waiting on failing over for a good network to arrive. However, in most cases the improvement in the reliability Ω_{model} is limited questioning if the significant increase in the state space paid for MA_{PEXP} is providing sufficient gain.

Future work

Further assessment of the proposed prediction modelling approaches is relevant to study how composition rules should be constructed in the model generation function of the ODDR. Particularly, further studies should focus on for which assumptions on network arrival probabilities, network properties and the accuracy of this knowledge, which approach offers best performance-complexity trade-offs. I.e. in some cases a DAM-like approach may be beneficial if the arriving networks often change characteristics. In other cases, where the operating scenarios are more static, proactive pre-computations like in the MA_{PEXP} case can offer a better performance where more complex computations are less critical if re-computations are rare.

In terms of evaluation, the different policies generated by the DAM model must be compared to the MA_{DA} and MA_{PEXP} results under scenario assumptions. This is relevant for both the model based analysis as well as in the PE simulation model. The latter is interesting in the DAM case where the true end-user service progress states can be applied, which may lead to different results than the model based analysis.

Chapter 9

Conclusion

9.1 Summary

For end-nodes operating in emerging ubiquities networking scenarios a high diversity of available access networks (different technologies, provider infrastructures and operational characteristics) may enable some attractive properties to resolve fault issues. However, the lack of opportunities to deploy network wide mechanisms for synchronized QoS control, network monitoring, diagnosis and repair challenges traditional centralized network fault management approaches. Instead, control decisions must be made at the network edge, which motivates the approach of end-node driven fault management studied in this thesis.

This approach raises several issues related to unobservable faults and unreliable observations based on existing network traffic. The consequences are diagnosis imperfections and remediation actions that can negatively effect end-user service dependability properties. Solving these issues requires good techniques for observation collection, diagnosis and decision making. However, imperfections are unavoidable. This calls for a joint view on the processes in the end-node driven fault management loop. Such a joint view must help to optimize for robustness and end-user service oriented reliability, which can be difficult to obtain if focusing on the individual components alone.

In summary, this thesis has contributed to several parts of the fault management loop including:

Overall architecture - An architecture has been proposed known as the ODDR (Observation, Diagnosis, Decision and Remediation Execution). It identifies the main functions of the fault management control loop including adaptation to changing scenarios. The ODDR has been studied in a context of diagnosing network contention and infrastructure congestion faults while providing remediation by access network fail-over.

Joined view on observation process and diagnosis - Two studies have been conducted providing an integrated view on the observation process and the diagnosis component: i) A basic Bayesian Network has been designed and implemented for diagnosis using multi-layer observation options to improve its robustness to unreliable observations and improve diagnosis state estimation accuracy. ii) Measurement errors may occur in the observation process. To ameliorate the negative impact of such errors on diagnosis performance a Hidden Markov Model diagnosis approach has been proposed and evaluated to include observation uncertainty supplied by an observation component.

Joined view on diagnosis imperfections and decisions - A reliable data transfer end-user service model is defined. It is based on a deadline requirement rather than a traditional QoS service formulation on minimum throughput requirements. This enables to study remediation policies taking into account diagnosis imperfections and end-user service state (criticality) to: i) improve remediation decisions by policy evaluation that can maximize reliability, and ii) identify diagnosis settings that best trade off diagnosis imperfections based on historic traces of diagnosis component performance. Central to these evaluations is a developed policy evaluation model. It includes a parsimonious diagnosis model enabling assessment of diagnosis components with varying properties.

Approach for adaptation - To accommodate changes in the scenarios of the ODDR, an outline for dynamic model generation has been introduced in the ODDR context. As the approach is based on the Stochastic Activity Network (SAN) formalism the policy evaluation model has been re-implemented in this setup. This has enabled a study comparing the impact of proactive and reactive model generation approaches on policy evaluation results, end-user service reliability and basic complexity properties.

Main results

In the end-node perspective a basic threshold diagnosis mechanism and a proposed Bayesian Network approach are compared on the same diagnosis cases based on unreliable observations. The BN in the comparison demonstrates how the correlation of multi-layer observations in the protocol stack can lead to improved diagnosis accuracy, some robustness to changes in the observations and an option to handle missing observations. In the observation process measurement uncertainty may be attributed to observations. An introduced HMM based diagnosis mechanism (H2), using this uncertainty information is also shown to provide improved diagnosis performance trade-offs of timeliness and accuracy over a classical HMM diagnosis model formulation (H0). Compared to a static measurement error compensation of (H0) to form (H1), for a practical setup of a non-adaptable diagnosis configuration, (H2) further provides a more balanced outcome of timeliness and accuracy. These results suggest, that using uncertainty bounds can provide a structured approach to improve diagnosis robustness.

Diagnosis imperfections of timeliness and accuracy in the end-node setting are, however, unavoidable. A characterization of such imperfections can be applied in a policy evaluation system model to predict appropriate decision policies. In a study of the reliability properties of the time constrained data transfer service it is shown how the best policy differs in relation to diagnosis capabilities. For some levels of diagnosis accuracy *no fail-over* is a better policy than failing over when a fault is diagnosed. Also, waiting to enable fail-overs can improve reliability when there is uncertainty to the correctness of the diagnosis outcome and end-user service criticality is low. Although improvements in the studied scenario are modest (5.5-10 percentage points) under the studied limitations it is demonstrated how assessing diagnosis characteristics in the remediation decision is a highly relevant option to improve reliability.

To pursuit lightweight models for assessment of diagnosis imperfection trade-offs in policy evaluation, a parsimonious diagnosis model approach has been proposed. It has been found useful to capture properties of simple diagnosis behavior, where independent diagnosis outcomes can be assumed. A more interesting class of temporal diagnosis mechanisms promises to offer significant accuracy improvements using memory to correlate multiple diagnosis outcomes. By applying a parametrization method using diagnosis performance metrics, it has been shown how the parsimonious diagnosis model can also represent this class without an increase in the state space. These capabilities have been applied in a reliability study implementing a temporal diagnosis heuristic (α-count) from [25]. Under a remediation policy of *fail-over at diagnosed fault state* the model based results show that the α-count diagnosis mechanism can provide significantly improved reliability compared to simpler diagnosis components not using time correlated observations. Moreover, it is shown how best diagnosis-settings can be identified to maximize reliability trading off reaction time and false alarms. The policy evaluation model based results have been found to be consistent with equal studies performed in a system level simulation model.

Finally a study has been conducted on how policy evaluation models may be adapted to changing conditions; particularly in the arrival of a new access network. Proactive and reactive model composition approaches for the previously defined reliability studies have been compared on generated policy results, resulting reliability and basic state space sizes. Early results show how a reactive approach can make use of the current end-user service state to compute a unique policy from this point. In most cases, however, these policies are consistent with pre-computed policies obtained under the same assumption of no expectation on the arrival of the new access network. A proactive approach enabling to probabilistically include expected arrival of a well performing new access network, only provides limited reliability gains. This questions the application of this proactivity given the accompanying large state space requirements.

9.2 Outlook

Focusing at the delimitations and discoveries made in this work, several interesting topics have been identified in the ODDR end-node driven fault management loop for future work.

Diagnosis robustness and feasibility of complex diagnosis

The inherent delimitation that an end-node may not rely on infrastructure network support imposes significant challenges to fault diagnosis. For common network and end-user service conditions the impact of measurement errors on different observations should be established. For relevant observations corresponding uncertainty estimation techniques must be developed. Finally, these estimation techniques should be studied in a setting together with the observation uncertainty diagnosis approach and its improvements proposed in this thesis.

The multi-fault scenarios and lack of reliable observations motivate the application of complex diagnosis mechanisms. These use cross-layer observations and apply some system structure in the diagnosis model. In general, such mechanisms are highly complex and may not be applicable in a limited resource setup of the end-node. Future studies may focus on if efficient approximate and partial solution techniques can be dynamically applied under control of the decision process. Based on end-user service requirements, cost and availability of remediation actions, it has the holistic view to determine the needed level of accuracy and timeliness.

Decision models and policy evaluation

Even for the delimited scenarios and lightweight system models of this thesis, policy evaluation is computationally costly. Solutions should be sought in identifying further lightweight model components, as in case of the parsimonious diagnosis model, to keep the state space small. The current policy evaluation approach calculated all permutations of a certain policy heuristic to identify the best. In future work a more systematic and delimited policy evaluation approach must be defined. It may be based on a Markov Decision Process approach to derive optimal policies for only confined parts of the state space.

End-user service models and optimization aims

In this thesis, the starting point has been made in a time constrained reliable data-transfer end-user service. It has resulted in an interesting end-user service model formulation, where its criticality level can be used in the decision process. Future work should study for which end-user service cases similar end-user service formulations could be relevant. An example is a cloud computing service. A model of its stochastic storage and computation actions defines its reliability depending on network conditions and where actions are carried out (in the cloud or locally). Another example is to model reliable voice and video services, by setting requirements to what fraction of different quality levels can be accepted during a session. Finally, based on the end-user service model and reliability requirements, additional optimization aims of the decision component should be to minimize overhead from unnecessary actions of remediation and active observations.

Adaptation and learning

A dynamic model generation approach has been outlined in this work. In future work composition rules must be specified to generate models for different encountered scenarios of access networks. The rules must take into consideration: i) how much of the system state space to include proactively for better remediation decisions, and ii) which parts to include reactively to minimize the state space complexity.

Such adaptation may initially be based on deterministically available observations such as: access network technology, wireless channel used, packet loss rates, observable contention levels etc. Progressing the approach, gradual learning of system parameters, which have been assumed to be available in this thesis, must be introduced. Some of these parameters are: amount of network states in an end-to-end path, the network state process and the imperfection properties of a diagnosis component.

Another integral aspect of learning, readily studied in existing autonomous network management settings [12], is how to learn system dependencies. In the end-node driven fault management approach this would be needed to establish the relation between experienced network (fault) states, operator networks and end-user service provisioning end-points. This information would be required to assess and initiate correct remediation actions.

Future scenarios

Future investigations should explore the studied principles of the ODDR in a realistic network setting to clarify to which extent the assumptions made in this work can be applied. Also, a central issue is scalability to clarify how multiple end-nodes operating in a decentralized manner will affect the networks in lack of centralized control. A part of such a study could include distributed approaches where end-nodes collaborate to share decision policy solutions, network parameters, models etc. This could help to minimize required computation at the individual nodes and conflicting decisions across multiple nodes i.e. where all select the same access network in the case of a fault.

Appendix A

Current and Future Networks

This appendix provides more detailed background on the analysis presented in the introduction on current networks and visions of next generation networks.

A.1 Current Networking Scenarios

To provide a starting point for considering NGN and beyond, initially, key properties of current publicly available networking systems are highlighted. The open *Internet networks* and closed *mobile networks* represent two different worlds that are predicted to converge. Making this distinction is, however, already difficult as Internet access is becoming an inherent part of data services in mobile networks. Yet, from a technical perspective there is still a clear distinction in the architecture of the two networking systems that has a practical impact on their properties. In this sense it is relevant to consider NGN from both perspectives to provide an understanding of how the convergence may take place in relation to the technical challenges but also economical and consumer market interests.

A.1.1 Internet Networks

A general principle of the Internet is to form a network of networks with no central entity and thus no single point of failure. From a high level perspective this has made the Internet very flexible and expandable giving it its popularity and global spread encompassing more than estimated 550 million hosts [38] and 1.5 billion users [79] by mid 2008. The Internet as a whole is also surprisingly robust. Parts of the network are continually stressed by increasing traffic amounts and malicious attacks. While the latter is often found to affect Internet performance [13], [96] it has continued its operation. Since the beginning of the publicly available Internet in the eighties numerous new protocols, hardware technologies and end-user services have been deployed. Yet, the basic principles of combining IP packet switching technology with reliable transport options delivered by TCP have been left largely unchanged. Clearly, the key reason for this success is the TCP/IP protocol stack where a strong layered architecture has had a significant role in promoting its properties. In summary these properties are:

Physical and link layer independence - The Internet protocol stack is basically independent of applied physical and link layer technologies which clearly has a part in its success. Thus, physical and link layer technologies widely in use to provide *last mile* connectivity, are results of utilizing existing infrastructure such as xDSL on PSTN copper wire and IEEE 802.1D based cable modems on cable broadcast systems. In the backbone networks telecommunication providers are utilizing legacy T-carriers/E-carriers and SDH (fibre) in ATM based networks and recently more cost efficient Ethernet based solutions offering up to 10 Gbit/s using optical fibers. Moreover, some network providers have started to offer fibre-to-the-(home/Building/Curb) (FTTx) enabling Gbit/s bandwidth at the last mile as well. Requirements for mobility and flexibility has also let to the introduction of multiple wireless technologies targeted at IP-traffic. Last mile connectivity may be provided by WLAN or WiMAX. In the mobile domain WiMAX, EDGE and UMTS are also starting to play a significant role in the provisioning of Internet connectivity.

Flexible addressing scheme and routing - The cornerstone of the Internet is its packet switched architecture where single IP-packets are routed through the network from a sender to one or multiple destination hosts. In general the network layer provides no guaranties of packet delivery. Packets may be received in another order than they have been sent, packets may be corrupted, lost or even duplicated in the path. Reliable delivery must be provided in other layers.

Elementary transport protocols - To provide basic interfaces for end-to-end transport and application multiplexing (by ports) the TCP/IP protocol stack offers UDP and TCP for connection-less and connection-oriented data transport, respectively. While UDP is central for streaming end-user services where losses can be tolerated, TCP offers reliable data delivery and transmission control to mitigate network congestion. Recent statistics as in [27] have shown that TCP/UDP convey in the area of 90% of all data in the Internet where the remaining part largely consists of control message such as ICMP ping. While newer transport protocols with extended functionalities have been proposed, such as SCTP proposed by IETF [130] in 2000, these have not gained widespread penetration. This partially underlines the versatility of TCP and UDP.

Means for QoS, Security and Monitoring - While originally being designed for confined non-public networks with best effort service delivery the original design of protocols and network services (e.g. Domain Name System (DNS)) have only to a limited degree considered issues of QoS and security. However, several solutions have over time emerged to enhance the TCP/IP suite such as IPsec for packet level security, DiffServ for QoS management and control and the SNMP suite for monitoring and management tasks. While such solutions have been pushed to improve the manageability and control of IP networks they also highlight how difficult it may be to ensure proper end-to-end support in the Internet. For instance DiffServ may not supported on all parts of a path or use different/conflicting policies in different domains. Further, claims have been made [50] that the IPsec design in order to handle a large variety of end-to-end configurations has become too complex to be certified secure. A closer look on such shortcomings which eventually are associated to the Internet architecture are discussed in the following.

With IP-based networking driving the convergence towards next generation networks its weaknesses and limitations become more exposed. In summary, some of the more important are: *Mobility* - A classical issue of IP addressing and routing is that devices are expected to remain stationary in a network. This assumption is not valid for emerging high mobility scenarios and numerous solutions have been proposed and implemented ranging from Mobile IP (MIPv4/6, layer 3) to end-to-end solutions based on Stream Control Transmission Protocol (layer 7). *Wireless communications* - Originally designed for reliable wired links, flow control protocols like TCP, have assumed congestion as the dominating cause of packet losses. With the increased use of unreliable wireless links in access and ad-hoc networks, the congestion assumption can severely degrade their efficiency in such scenarios. *End-to-end QoS* - While being an integral part of IP network design since the eighties, providing a given QoS level in the end-to-end path remains a significant challenge. Several technical solutions exist (e.g. DiffServ, IntServ, MPLS) which can be successfully deployed within network operator controlled domains. However, some operators retain from implementing these for economical reasons [97] while they further are difficult to apply in an end-to-end context due to heterogenoius networks with difference in QoS technologies and prioritization policies supported, if any at all. The significant decentalized network control and lack of incentive for infrastructure network providers to provide QoS means for end-users have lead to a widespread conception that end-to-end guarantees can never be made in an IP network context. This aspects calls for alternative mechanisms that can still help supply needed performance and dependability of end-user services despite lack of network support. This is a strong incentive of the work of this thesis.

A.1.2 Mobile Networks

Mobile networks are designed from completely different principles and presumptions than the networks forming the Internet. First of all, individual mobile networks are closed and controlled completely by their operators. These networks have from scratch been designed to support a single service namely two-way communication between mobile end-user terminals and between a mobile terminal and existing fixed-line phones. From a general perspective this enables the following key properties of mobile networks:

Quality of Service - When a call is made to or from a MS necessary resources are reserved in the entire mobile network to ensure a consistent QoS throughout a call. If these resources cannot be assigned the call is not accepted in the first place. Having complete control over the network, operators can continually improve the capacity of the network to cope with an increasing amount of users and user demands. In this manner a high fraction of accepted calls can be ensured as well.

Security - Currently deployed mobile networks have inherently been designed with security in mind. The operators must be protected against malicious users trying to damage the service provisioning or obtain free service. The users must also be protected from malicious attackers trying to listen in on their calls or use their identity to make calls. Network functions like data encryption and SIM-card based authentication are used to address these security risk scenarios.

Centralized control and monitoring - Mobile networks have initially been designed to have a centralized and hierarchic structure. This simplifies network control and monitoring but also makes the network weaker to withstand failures in the core network components.

Mobility and Roaming Mobile networks have been inherently designed to cope with high mobility of end-user terminals which may operate while moving between cells in *home* provider networks and to cells of other *visited* provider networks (roaming).

Service model and billing - In contrast to the end-user service model in Internet networks mobile networks are inherently strongly tied to provided services which may be speech and data, as well as more high level end-user services such as mobile TV or music streaming. This also enables full control of billing.

The mobile networks are under complete control of network operators. Thus, changing the network technologies and mechanisms is somewhat easier than in the Internet where multiple parties must agree on changes and implement these. This also means that mobile networks are under a rapid evolution where new technologies are introduced to improve service capabilities and reduce costs of CAPEX (establishment and upgrade) and OPEX (maintenance, network lease and other operation costs). New developments are:

High bandwidth services - Other mobile services than speech and text messages have been deployed. These employ mobile television, music downloads as well as mobile alternatives to fixed line Internet access. Such services require extensive improvement and changes in the network infrastructure in more ways. 1) Besides new air interfaces capacity in backbone networks must be increased to manage the increasing traffic amounts. 2) Decentralization of core network functions to avoid aggregating the high bandwidth traffic in a few central points causing congestion and inflexibility to further bandwidth increases.

Introduction of IP and Ethernet - The latest releases of mobile standards in 3GPP are targeting a mobile system architecture based on an All IP Network (AIPN) [3]. However, IP-based communication in the RAN and core network is readily being supported in new infrastructure components conveying GSM and UMTS traffic. In addition to this, network sites are being upgraded/installed with cost efficient and high performance Ethernet equipment replacing more expensive E1/T1 lines. Advantages of IP enables networks are:

- Operators can make advantage of cost efficient IP/Ethernet infrastructure already established for wired broadband connections (xDSL, cable) to also transport data between the radio Base Station (BS) and the mobile core network controllers.

- In the original specifications of 2G mobile networks like GSM timeslots are pre-reserved in the radio network infrastructure when a call is made; also when no traffic is conveyed. Optimization of network resource use. can be achieved by packetizing datagrams from multiple mobile terminals and transmit these only when data is available.
- Call switching capabilities may be introduced at the network edge to: 1) Avoid usage of expensive and long latency satellite links for local calls. 2) Reduce load in the *mobile core* network. 3) Provide direct packet routing for IP enabled services.
- Enable new products where GSM/UMTS *Femto* radio base stations can be installed by end-users on existing IP-based broadband connections [9].

A.1.3 Other Network Types

While mobile and Internet networks are main drivers of current digital communication there are other highly relevant publicly available networking technologies currently being rolled out such as *terrestrial broadcast* networks and *ad-hoc* networks.

Considering terrestrial broadcast networks, currently, large investments in and deployment of digital broadcast networks are made. Providers of TV and radio are motivated to abandon analogue techniques to deliver improved content such as higher quality audio/video, meta-data and interactive services. In several countries counting e.g. Sweden, Denmark, the Netherlands and Finland, the terrestrial analogue network has already been switched off and replaced by a digital counterpart. While multiple standards exist (DVB-T, ATSC, ISDB and DMB), in general such networks have some interesting properties [89]: a) High coverage; typically operating in low frequency bands (e.g. VHF 170-230 MHz) the associated low signal attenuation means that good coverage can be provided with few masts. b) High bandwidth; At a signal bandwidth of 8 MHz using 64-QAM up to 31.67 Mbit/s can be supported in DVB-T. c) A terminal can combine signals from multiple masts to provide robust reception and simple mobility support. d) Solutions for IP over e.g. DVB have been proposed [62][44] to enable IP-based broadcasting services.

Inherently, these networks are optimized for broadcasting service and could play a significant role in future networks. An example could be to deliver the same contents to multiple mobile terminals in a limited geographical area such as video streams at rock concert using significantly fewer wireless resources than what a typical mobile network broadcasting individual data streams would.

In summary, today mobile networks excel in providing a networking solution with control leading to inherent QoS and a profitable content delivery platform due to the strong bounds between the network and the end-user services. However, the need for added flexibility to reduce OPEX and CAPEX costs, while handling the fast evolution of mobile technologies, challenges traditional hierarchical networking architectures based on costly hardware. The solution is partially in the transitioning to existing IP based network technologies and architectures. At the same time mobile technology advancements are driven by an incentive to boost the demand for IP/data services in mobile platforms. Based on these developments it is clear that a convergence of mobile networks and current Internet based technologies is happening with IP as a unifying element. This, realization drives the ongoing research and development in understanding how a new generation of IP based networks can support future depends for dependable and high performance end-user service provisioning.

A.2 NGN Initiatives and Expectations

Next Generation Networks is a term describing how current networks expectedly will be replaced with new network technologies and approaches. The resulting networks will offer completely new opportunities of creating ubiquitous computing environments. Many of the technologies used in the first generations of the Internet and mobile networks were developed quietly in research environments establishing the basis for entirely new ways of interaction and markets. Today, the amount of actors have increased significantly including large equipment manufacturers, telecommunication companies, digital service providers, governments and research communities. These actors are all trying to get their share as new requirements to communication systems emerge.

Seen from an overall perspective the transition to NGNs could follow two, not mutually exclusive, paths; through *clean-slate designs* or *evolutionary designs*. Starting with clean-slate designs this approach means to create entirely new network designs without seeking to achieve compatibility with existing technologies. This way of thinking is typically popular in research communities where concerns regarding existing equipment investments are smaller compared to industry. Some researchers believe that clean-slate designs are necessary to ensure global networks capable of handling future requirements. Evolutionary designs are based on an approach where current technologies are used as a starting point to improve the networks. This approach is not surprisingly most interesting from an industry perspective. Existing investments must be utilized while new technologies slowly are adopted. The transition of Mobile Networks to use Ethernet/IP is a good example of this. Altogether, both clean-slate and evolutionary design approaches are valid. Clean-slate designs may contribute with completely new concepts while evolutionary designs will ensure that transitions to new network technologies will be smooth for industry and users.

A.2.1 NGN Projects

Driven by the various actors, a large part of umbrella projects have been established in the last decade to support and organize research within NGNs.

Within *clean-slate* research can be mentioned: *AKARI* [6], a project under the Japanese National Research Institute for Information and Communications Technology (NICT) aiming to demonstrate a clean-slate design by 2016 focusing on high-bandwidth optical transmission links and robustness via autonomous network adaptation. *FIND* (Future Internet Design) [106] under the US NSF is a project with similar objectives trying to separate future network design from legacy Internet technologies while maintaining requirements of a secure, open and highly available Internet. FIND is associated to another NSF project *GENI* [108] which is a large scale networking framework with heterogeneous nodes enabling to test clean-slate design implementations also with no presumptions on IP technology.

In the group of *evolutionary design* projects, more focus is on developing new specifications and standards to drive the development of technologies that can provide immediate increments to existing network technologies. As example is work made in 3GPP which is a joint collaboration between ETSI [3] and multiple equipment manufacturers and network operators. Besides having specified the 3G/UMTS standards 3GPP is also a forum in which 4G (LTE and LTE-Advanced) [2] technologies are developed aiming to deliver radio access at 50-100 Mbit/s and transitioning mobile networks into All-IP networks. ETSI are also involved in the *tispan* (Telecom and Internet converged Services & Protocols for Advanced Network) [47] project based on existing efforts in 3GPP. It aims to adapt a comprehensive IP-centric architecture for end-user service delivery called IP Multimedia Subsystem which is focused on maintaining a viable business platform for future networks covering simultaneously mobile and Internet based networks. Finally, it is also relevant to mention the substantial effort made in the ITU-T context where several projects have been conducted to create recommendations for NGN where the latest is the NGN-Global Standards Initiative (NGN-GSI). One of the current contributions is an NGN reference model denominated Y.2011 [80] which has many commonalities to other architectures as e.g. proposed in ETSI-tispan. In the following section we will briefly consider some of the assumed characteristics of NGN according to ITU-T which are relevant to, and shared by this work.

A.2.2 Key Characteristics of NGN

- *Decoupling of end-user service provision from network and independence of service-related functions from underlying transport technologies.*
  To maintain flexibility and decoupled development of the end-user services and the network the end-user service should not not be tightly coupled to the network technology on which it is running. A central principle in the Y.2011 reference model is a logical split in the protocol stack between service related functions (e.g. end-user services, billing and signalling) and network data transport functionality (not to be confused with layer 4 transport in the OSI stack). Services-related functions must be able to operate without having to consider if the transmission medium is wireless/wireline, which technology is used and what available networking resources exist. These considerations must be made separately in the transport services in an attempt to deliver the network performance required by the service-related functions.

- *Support for a wide range of end-user services, applications and mechanisms based on network service building blocks (including real time /streaming /non-real time services and multi-media.)*
  These building blocks are generally offered in a middleware layer architecture providing standardized interfaces to perform functions of defining end-user service requirements and actual data transport. Examples of such frameworks are: *The Open framework* presented in [103] which enables seamless service migration between devices and underlying networks and the *HIDENETS architecture* [37] enabling functions to raise service dependability in ad-hoc and ad-hoc to infrastructure network systems.

- *Unrestricted access by users to different service providers.*

> With the eased couplings between end-user services and the network itself a common vision of NGNs are that users do not buy all their connectivity services by a single network provider. Instead, an end-node device could have access to use multiple networks and technologies across operators to make use of their diverse properties depending on the requirements of a particular end-user service.

Obtaining these properties in practice implies numerous challenges. Some are technical and related to the weaknesses of the flexible but somewhat complex IP network architectures. Others are financial where network operates are only willing to invest in new solutions that can sustain and in best case improve the market opportunities. For instance, currently flat-rate IP telephony on mobile handsets is a threat to the valuable market in time charged voice services. Finally, there are clearly also a political influence to regulate which restrictions network operators may or may not be allowed to introduce. In this thesis these latter aspects are only considered superficially; but they will clearly have a significant impact on whether proposed approaches are viable or not.

Appendix B

Hand-over Technique Analysis

In this appendix a brief review on existing work in existing mechanisms and principles for network hand-over is presented. The aim is to provide a high-level comparison of such approaches to the fault management oriented perspective to access network selection made in this work. A summary of the considered differences is presented in Section 2.5 on page 30.

B.1 Existing Hand-over Approaches

In this section an overview of existing hand-over related work is presented. Relevant shortcomings are emphasized in relation to the fault management approach considered in this work.

B.1.1 Horizontal and Vertical Hand-over Scenarios

From an overall perspective existing hand-over mechanisms may be split into two categories: vertical hand-overs and horizontal hand-overs [126]. The horizontal case refers to hand-overs within a single radio technology while vertical cases refer to hand-overs across different radio technologies. In much existing work [126], [122], [70] horizontal hand-overs are characterized by the challenge of providing best connectivity to the Access Point (AP) of a given network providing a sufficient signal strength, signal to noise plus interference ratio (SNIR) and low BER. This is typically solved by establishing threshold based hand-over decision mechanisms of filtered observations of SNR with timers and hysteresis to minimize unnecessary hand-overs [126]. It is further argued how vertical hand-over mechanisms are increasingly complex as:
1) Observations used for hand-over decisions have different implications depending on technology used. E.g. a low SNR in one technology may lead to a higher transmission rate than a high SNR case in another. Thus, SNR observations are not directly comparable.
2) Hand-overs are not necessarily conducted only to provide connectivity but may also be made to achieve certain connectivity aims such as low cost, high throughput and/or low jitter [86]. In this case different properties of different access technologies can be utilized.
3) Much signalling, synchronization and time overhead may be required to move

traffic and session data from one access technology to another.

In comparison to the fault-centric approach considered in this work the distinction between vertical and/or horizontal hand-overs is not a central aspect. The approach may be applied in purely horizontal scenarios as well as in vertical scenarios. The crucial aim is to use a connectivity option that delivers the required transport service and chooses the best alternative (remediation option) when a given fault is diagnosed. The advanced objective of optimizing dependability metrics, given faults in both access networks and infrastructure, has many similarities to challenges of providing vertical hand-overs. Thus, in the following sections primarily mechanisms of vertical hand-overs are reviewed to identify relevant comparison cases, differences in approaches and their advantages/disadvantages.

B.1.2 Components of Vertical Hand-overs

In general the functional elements of a hand-over mechanism can be split in three phases [129][86] as depicted in the upper part of Figure 2.2.

System Discovery concerns identifying which access networks are available and typically also identifying their properties considering QoS levels available, basic transmission rates, current utilization etc. Many issues are related to system discovery. In end-nodes with a single radio interface end-nodes may be required to periodically stop their association with the current AP to scan for alternative networks. To avoid end-user service disruption there is a limit on for how long channel scanning can be performed and thereby how many networks may be discovered. Further, information about the properties of available networks may be limited. If they are not provided the end-node may need to make its own assessment. More on these issues is discussed in sections B.1.3 and B.1.4. A common synonyms for the process of *system discovery* is *handover initiation*.

Handoff Decision is the central aspect on deciding if and when to initiate a hand-over to one of the available networks discovered in the system discovery phase. Making good decisions is a central challenge in establishing a strong hand-over mechanism and is consequently a well studied topic. Basic aspects of decision methods studied in existing work are discussed in section B.1.3. In Table B.1 a non-exhaustive overview of decision input parameters (e.g. observations) and hand-over decision objectives identified in existing work are presented.

Handoff Execution is the phase in which the actual handover is made. A handover can be a fairly basic process of changing the physical layer attachment solely to a highly complex process including renegotiation of authentication, signalling between core networks, redundant data streams, buffering of data and IP address change. Issues on hand-over executions are shortly addressed in section B.1.5.

The lower part of Figure 2.2 depicts the fault-centric approach presented in this work with hand-over as a remediation option. Clearly, the phases are very similar. In section 2.5 identified differences between the fault-centric approach and existing hand-over methods are discussed.

<table>
<tr><th colspan="2">Decision Input Parameters</th></tr>
<tr><td>• Networks
- Transmission rates
- Service provider charge [109]
- Power usage requirements
- Delay, Jitter
- PLR, BER
- QoS classes supported [129][86]
• End-node device
- Battery status [86]
- Available network interfaces
- Processing load [128]
• Context-information
- Location
- Velocity [86][132]</td><td>• User preferences
- Preferred networks (technology, provider) [86]
• End-user service
- QoS requirements (delay, jitter, throughput, PLR)
- Session duration [128]
• Available networks
- Available bandwidth/delay [143][128]
- Transfer completion time [109]
- RSS [70]
• Connected network
- Connection lifetime prediction [143]
- RSS [70]</td></tr>
<tr><th colspan="2">Hand-over objectives</th></tr>
<tr><td>• Reduce unnecessary hand-overs [128][143][35]
• Ensure required QoS in terms of delay, jitter and bandwidth [129][122]
• Minimize delay and jitter. Maximize bandwidth</td><td>• Maintain connectivity
• Processing load [128]
• Minimize battery consumption
• User preferences</td></tr>
</table>

Table B.1: *A set of decision input parameters and hand-over objectives used in existing work of vertical and horizontal hand-overs.*

B.1.3 Multi-criteria Decision Methods

In this section a short overview of existing hand-over decision methods is presented. While most of these methods are general and could apply to network controlled hand-over mechanisms the primary focus is on end-node controlled/driven hand-over mechanisms.

Making a decision of when a particular node should perform a hand-over is typically a result of a complex mechanism that weighs multiple input parameters (parameters of available networks, end-user service requirements, end-node capabilities) against each other. The mechanism must produce a decision that will benefit multiple objectives such as increasing throughput and decreasing delay while reducing battery consumption, cost of connectivity and minimizing unnecessary hand-overs (see also Table B.1). This leads to a multi-criteria and multi-objective problem where an optimal solution must be found.

Multiple attribute decision making techniques

Several well established mathematical approaches exist to provide a solution for the multi-criteria decision problem. In [129] the authors establish a basic presentation of such decision mechanisms and compare their performance. A set of traffic classes are defined in accordance to 3GPP specifications [1] considering:

Conversational, Streaming, Interactive and Background. Given a weighing of QoS metrics (BER, Delay, Jitter and Bandwidth) for each class the individual algorithms are evaluated based on their capabilities of assigning a wireless node, using a certain QoS class, to a network with the appropriate capabilities. Specifically, vertical hand-over cases are considered in terms of evaluating the algorithms across UMTS, GPRS and WLAN access networks with different and varying QoS capabilities. The studied algorithms in [129] belong to a method category called Multiple Attribute Decision Making (MADM) referring to the weighing of multiple attributes of multiple decision options to choose the most suitable option. In practice each algorithm describes a score function enabling a priority ordering of the options which in this case are available networks. In short the studied algorithms are:

Simple Additive Weighting (SAW) - For each attribute j of a network option a sub-score is calculated by multiplying an attribute weight (ω_j) with a metric (r_j) of how well the attribute matches the requirement. A total score is calculated by summing all sub-scores from each parameter and identifying the network with the highest score as the most suitable network.

Technique for Order Pref. by Similarity to Ideal Solution (TOPSIS) - A total score for each network option is calculated based on how close the individual attributes are to the actual requirements. The most identical solution (network) to the requirements will be selected.

Multiplicative Exponent Weighting (MEW) - This method is similar to SAW. However, weights are defined as an exponent ($r_j^{\omega_j}$) and the sub-scores are multiplied to form the overall score. A ratio is calculated between the obtained score and a theoretical ideal score where all attributes match requirements. The network with a score closest to 1 is chosen.

Grey Relational Analysis (GRA) - The method enables a comparison of decision options to an ideal decision option case based on sequences of normalized QoS parameters. Normalization can handle larger-the-better, smaller-the-better or nominal-the-better to consider different comparison cases to the ideal solution. More on GRA used for network selection can be found in [122].

To parametrize these approaches for a comparison weights ω_j have been assigned using a technique called Analytic Hierarchy Process.

Analytic Hierarchy Process (AHP) - This technique can be used to map e.g. QoS parameters and their importance to each other in a matrix. The Matrix is generated from a human assessment of the importance of different QoS factors to each other for each QoS class. Based on this matrix a set of weights can be deduced. Further, the consistency of the human assessment can be derived analytically to verify the validity of the obtained weights. Besides being used by the authors in [129] to parametrize the different studied algorithms AHP is also applied as a part of the solution in [122].

The overall conclusions of [129] are that these proposed solutions provide very similar network selection solutions in a scenario of one UMTS, one GPRS and two WLAN access networks. Best performing is however GRA where higher bandwidth and lower delay is obtained for Interactive and Background QoS classes.

Some generalized characteristics on solutions based on MADM techniques are:

- Efficient handling of decisions were a large number of attributes must be considered.
- Assumption of *perfect* and available observations of option network capabilities. Thus, such methods may not perform well under uncertainty.
- Primary focus on ranking of available networks to determine most optimal point of connectivity.

Decision function based techniques

Another survey made in [86] introduce a slightly different decision making category compared to MADM approaches. The category is denominated *decision function based techniques*. Basically the aim is still to rank available networks calculating utility, cost or score functions for available network options. Focus is, however, on creating such functions as a sum of weighted sub-functions. The sub-functions are dynamically updated and may encompass:

- QoS parameters of available bandwidth, delay, BER, ...
- Load balancing between optional access points (to avoid contention due to similar selections of multiple end-nodes).
- Dynamic user input to weigh network service charge to experienced QoS.
- Dynamic context information to establish need for a hand-over. The authors of [86] emphasize location, velocity end-node capabilities as important context parameters.

Details on specific solutions considering these aspects are presented in further details in [86]. The authors of the survey in [129] have proposed a new decision approach based on Markov Decision Processes (MDP) in [128]. The authors consider classical QoS metrics such as delay and bandwidth. However, by modelling network behavior in time and specifying costs of different QoS metric levels the authors can derive optimal policies on when to hand-over to optimize an overall reward function. The authors focus on minimizing the amount of hand-overs while providing a high reward given an assumption of the connection lifetime distribution and mean duration. Compared to mechanisms based on MADM the presented MDP approach provides the overall best reward while reducing needed hand-overs significantly.

As the category defines a broad scope of decision functions based on various criteria it is difficult to make general statements on these techniques. Based on the emphasized work in [86] it may however be stated that *decision function based techniques*:

- Cover a set of dynamic hand-over decision mechanisms determining when to perform a hand-over and to which network. Other criteria than static QoS metrics are considered.

- Do in general not consider unreliable observations.
- It is often assumed that network functions exist to support information provisioning to end-nodes. E.g. see [14].

Methods managing uncertainty
The authors of [86] also emphasize the use of Artificial Intelligence (AI) techniques such as Fuzzy logic and Neural Network based mechanisms. These mechanisms are providing robust mathematical frameworks to map different types of observations together while handling unreliable observations. As argued in [86], often, good decision mechanisms can be obtained using AI. However, methods such as Neural Networks increase complexity as they need to be trained from network data sets in order to provide useful decisions. These methods can be characterized as:

- Useful for combining and comparing data of different kinds to make best decisions.
- Handle non-numeric and complex parameters in fuzzy sets by assigning membership functions to them.
- Good decision strategies may be trained from collection of network data and evaluation of decision outcomes. The cost is, however, increased complexity.
- Capable of handling imprecise and unreliable data input.

B.1.4 Best Path Selection

The studied hand-over mechanisms in the previous sections are primarily concerned with ensuring connectivity to networks which can live up to certain QoS requirements. A large part of the proposed mechanisms focus only on solving QoS issues in the one hop link to the access point or base station. Less focus is on the entire end-to-end path from the end-node to the end-user service provider. A hand-over can, however, also be used to change the entry point to the infrastructure network and thereby potentially solve issues in the network path. This requires that the considered attributes for the decision process also include end-to-end path aspects. This leads to a brief description of best path selection mechanisms. Such mechanisms are typically characterized by a dynamic end-to-end path assessment.

The work in [59] uses a best end-to-end path selection mechanism to optimize the use of SCTP in a multi-home configuration. The multi-home option in SCTP is originally used to provide backup connectivity should the primary connection fail entirely. Moreover, the backup path is used to send retransmissions for added reliability. The authors in [59] modify SCTP to optimize the overall throughput by making sure the path with the highest throughput is always used for transmission. Thus, they specify a mechanism to estimate the bandwidth. It is based on a back-to-back probing approach where a packet train (multiple packets sent immediately after each other) is sent on a path. Based on the time inter-spacing between the packet pairs (obtained from acknowledgments) an estimate of the bottleneck bandwidth can be made as also demonstrated in

[30]. It is shown in [59] how the overall robustness and performance of SCTP can be improved. Another example is provided in [105] where both bandwidth estimation and delay estimation is applied in vertical handover scenarios to choose between best paths in an end-node driven approach. Such approaches provide network and end-user service provider independent path assessment and can, despite accuracy trade-offs, provide useful performance improvements. As also emphasized in [59] the obtainable accuracy is typically a trade-off in relation to the amount of probing traffic sent. This may be taken advantage off in bandwidth probing techniques by adaptively weighing resources (timely and data) required to accuracy needed. Existing work in best path selection can be characterized as:

- Useful approaches for network path estimation with no network or end-user service provider support.
- Enables trade-off of estimation accuracy in relation to overhead of time and data resources.
- From the review work presented in this section it appears that end-to-end best path selections techniques have not be thoroughly investigated in relation to handover decision approaches.

B.1.5 Handover Execution

An important aspect of providing both vertical and horizontal hand-overs is the actual *hand-over execution phase*. When a decision has been made to initiate a hand-over it must be executed while ensuring acceptable metrics of hand-over delay, packet loss and data overhead. Other important metrics can be reliability such as the probability that a fail-over will fail and a new must be initiated. Much work is focusing on designing new protocols and mechanisms that can be used to improve such metrics. Much of the work is focused on Layer 3-7 mobility aspects where typically many different network components like routers, access points and network agents are involved in the hand-over process. A large part of the work is focused on Mobile IP v. 4/6 (MIPv4/6) which solves IP mobility challenges. Extensions of MIP, called FMIPv6 [88], have been made to facilitate fast and potentially seamless hand-overs. In addition, much recent work attempts to optimize characterize and improve the performance of FMIPv6 e.g. see [94]. Another relevant IP mobility option considered in this work is SCTP. Also improved versions of SCTP have been proposed to strengthen its performance in mobile hand-over scenarios e.g. mSCTP in [87] and [95].

In this work focus is not on the actual hand-over process. Thus, optimizing or in other ways improving handover execution aspects is not in focus. More relevant is which properties existing and readily applicable handover mechanisms have and how they affect the hand-over reliability. Thus, establishing such properties will lead to parameters to be used in the decision model where they may affect decision outcomes.

B.2 Summary

A summary highlighting the difference between traditional hand-over approaches and the fault-management based approach of this work is presented in Section 2.5 on page 30.

Appendix C

Simulation Model

This appendix introduces details of the ns-2 based simulation setup introduced in Chapter 3. Section C.1 describes the approach for generation of network states (congestion level) by cross-traffic. In Section C.2 the Layer 4 experienced delay penalty is obtained for use in analytic models. Finally, Section C.3 presents the background on how simulation results generally are obtained in ns-2 for the simulation results of this Thesis.

C.1 Cross-traffic Generation

In this section details on the congestion network states based on cross traffic are defined. As shown in Figure 3.3 the state of the infrastructure network is controlled by traffic between cross-traffic nodes R_{A2} and R_{A3} in Network A and R_{B2} and R_{B3} in Network B. The traffic model is defined as follows:

- Traffic is generated as multiple concurrent FTP connections from node $R_{(A/B)2}$ to $R_{(A/B)3}$. An FTP connection is established to transfer a file of the size X where X is a sample from a Pareto distribution with parameters $\mu = 10\,KB,\ \beta = 1.5$ to mimic Internet traffic [8].
- The FTP connection generation process is Poissonian with rate $\lambda_{cg(A/B)}$.
- File data traffic is only conveyed from nodes $R_{(A/B)2}$ to $R_{(A/B)3}$. Thus, for sufficiently high traffic generation rates congestion will occur in the buffers at nodes $R_{(A/B)0}$ toward $R_{(A/B)1}$.

For the analysis in this work, a single network state configuration set is defined. The set specifies two network states a *normal state* and a *fault state*. The simulation parameters for these two network state configurations are introduced in Table C.1. Details on their impact on the time constrained reliable data transfer end-user service are presented in Table 3.3.

Stability of cross-traffic generation
The cross-traffic methodology can cause instability in the amount of concurrent FTP connections. Clearly, in cases with high congestion the throughput for each connection will drop. At a certain point a higher rate of connections will be generated than connections terminated. This means that the congestion is

Network state configuration			
State	$\lambda_{\mathbf{cA/cB}}$	**Buffer length** ($\mathbf{R_{A0/B0}}$)	**Approximate performance**
	Basic set		
Normal	86.9565	20	Approximately 600 KB/s
Fault	123.4568	20	Medium packet loss (Leads to approximately 1/2 of throughput in normal state
Units:	conn./s	packets	-

Table C.1: *Two sets of cross-traffic parameters controlling infrastructure network states normal and fault.*

increased continually leading to an undesired time dependent and inhomogeneous effect on the simulation results. As a part of the solution, to avoid this situation, a limit has been introduced to the amount of concurrent connections allowed. If this limit is reached new connection establishments are ignored until existing connections have terminated.

To establish whether instability is a problem with the selected parameters in Table C.1 the amount of concurrent FTP connections between $R_{(A/B)2}$ to $R_{(A/B)3}$ has been monitored throughout the transfer of a $10\,MB$ data amount from the end-node to end-user service provider via network A. One transfer has been made where the network is in the normal state throughout the transfer and another where the network is in the fault state. The results are depicted in Figure C.1. Clearly, there is not a strong tendency towards instability. In practice there may, however, be cases where the limit is reached ultimately reducing the rate of generated cross-traffic transfer events.

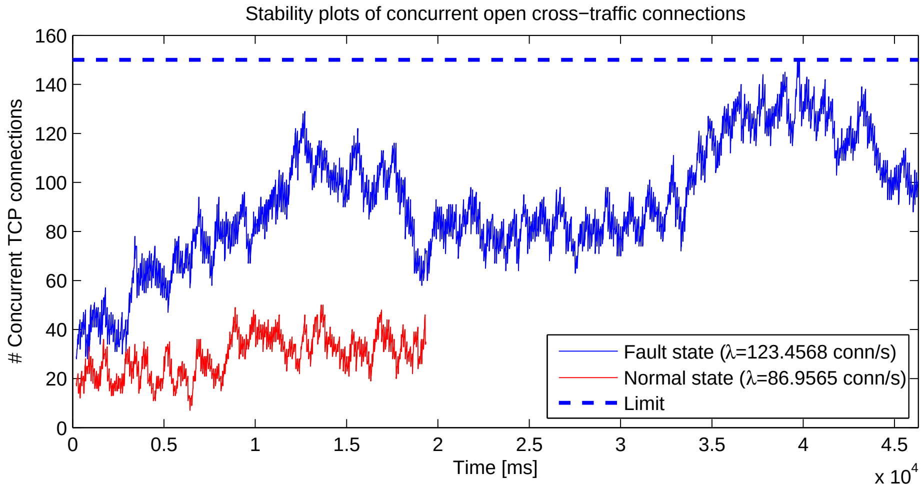

Figure C.1: *Amount of active cross-traffic connections during a data transfer for normal and fault network conditions in the Basic set.*

C.2 Failing Fail-over Time

In this section we attempt to establish the distribution and mean time delay penalty from a failing fail-over in an SCTP multi-homing setup. In the simulation this is simulated by changing access network from network A to network B and disabling the wireless link to AP_B for $200\,ms$ before failing back to access network A. A *link down* duration has been set for $200\,ms$ to correspond to some layer 2 delay in failing over. In practice this value depends heavily on the wireless equipment and device driver, used authentication/security schemes, link conditions and hand-over implementation. Including the layer 4 behavior in the delay is difficult as many parameters influence the impact. Some are:

- What the congestion window size is when the fail occurs.
- How many packets are lost in the failing network interface buffer.
- Whether the change in the congestion window size minimizes the impact on the SCTP stream from buffer overflow in the network.

The methodology used to establish an approximation of the mean delay is to compare data transfer times (data amount: $10\,MB$) of two sets of simulation runs: In 1) the network operating in a normal state where a data, and in 2) the data transfer stream experiences a single failing fail-over. In the second set a failing fail-over is generated manually after 10 s. In the two sets the simulation random generators are seeded equally. This means, that the difference in transfer time between two traces en each set is only caused by the failing fail-over. Figure C.2 depicts the distribution of data completion time *differences* for 50 simulation runs in each set.

As seen some of the differences are negative. These are cases where a simulation run experiencing a failing fail-over actually leads to a shorter data transfer time than the same simulation run where no failure occurs. The reason for this reduction in data transfer time is that the SCTP sender actually avoids some packet losses that occurs in the normal network state run. This can happen as the time of the transferred packets after the failing fail-over are time shifted compared to the pure normal state run. Further, the congestion window is correspondingly lower due to the failed fail-over which minimizes the impact of the packet losses. Still, in most cases the failing fail-over leads to a significant increase in the data transfer times. The mean delay impact is $\mu = 1206\,[ms]$.

C.3 Simulation Methodology

In this section the general methodology used to conduct simulation based tests in the ns-2 environment is introduced. The terminology of simulation based testing is introduced in Figure C.3.

In a *test* a simulation is used to investigate a certain aspect. As an example the aim of a test may be to compare results from the system model to simulation results. In a test various *simulation scenarios* may be defined where a *simulation scenario* describes a specific configuration of the simulation environment, i.e. network state configuration, fault/repair rate, link bandwidth etc. In a simulation scenario the desire may be to examine the effects of varying one or more specific parameters like *diagnosis threshold* γ_{rtt} and *remediation*

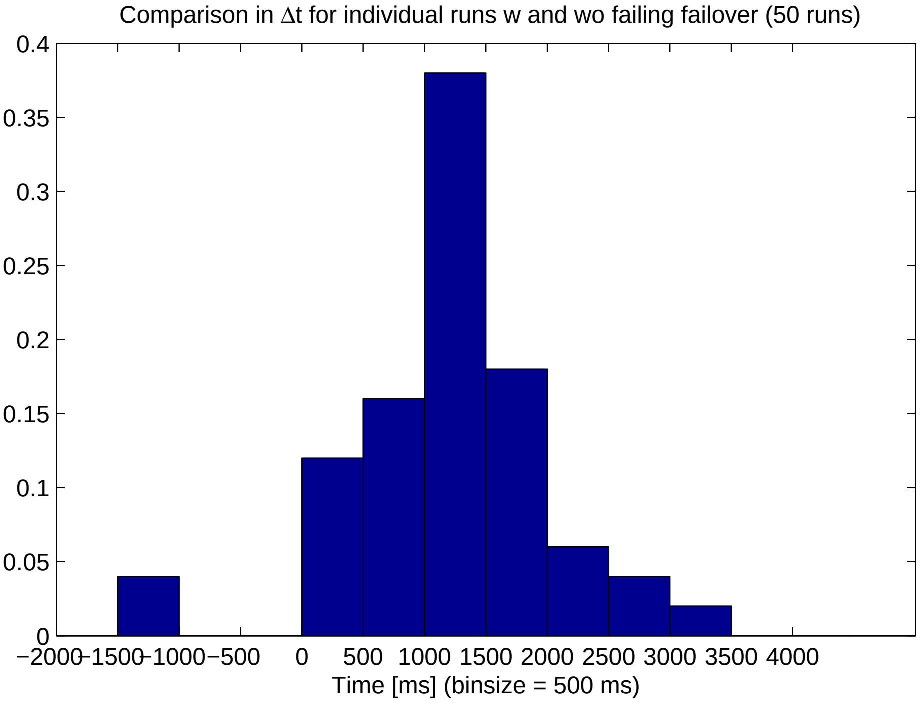

Figure C.2: *Distribution of difference in data transfer completion time between a normal state transfer and a normal state transfer where a failing fail-over occurs.*

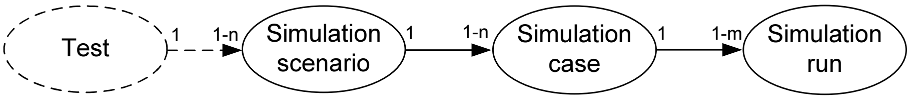

Figure C.3: *Terminology of tests conducted in the evaluation framework.*

policy. Each examined setting of a set of parameters leads to the definition of a *simulation case*. A simulation scenario may consist of one or more simulation cases. Finally a simulation case may consist of one or more *simulation runs*. A simulation run is the actual conduction of a simulation.

- The statistical independence of simulation runs are ensured by following the guidelines in the ns-2 manual [49]. Each random variable used in the simulation is based on an individual random generator. These random generators are seeded differently and consistently based on a main seed given to the *default random generator*. From each random generator (pseudo-)independent sub-streams can be drawn. A new sub-stream is drawn for each simulation run.

- For creation of the main results of this Thesis all simulation cases are re-seeded compared to runs used for training (establishment of state and model parameters). *Seed 1* is used for parameters. *Seed 7* is used for tests.

- The consistent use of seeding and sub-streams means that although different simulation runs in a simulation case are statistically independent this is not the case across simulation cases. I.e. random process values in simulation run i of simulation case A are the same as in simulation run i of simulation case B. In this setting this means that e.g. actual fault-occurrence and repair times are identical in the same simulation run across different simulation cases. As a result variations in runs across simulation cases are largely due to changed behavior from controlled variables.

Appendix D

Bayesian Network Approach

Section 4.4 introduced a Bayesian Network model for fault diagnosis. This Appendix contains details on how the model is obtained and further, how the model is parameterized and used for diagnosis.

D.1 Intermediate Model

To construct the BN structure $\mathcal{G}$, variables and their causal relations are mapped from the fault diagnosis scenario (Figure 3.3) to a graphical representation. The approach in this work has been to construct an intermediate model describing basic system components and in which order these components influence each other, i.e. their causal relations. Next, the intermediate model has been formed into a BN. The intermediate model is depicted in Figure D.1 and has been specified from the following structured method: (1) Variables have been identified that represent system components where faults may occur. E.g., *congested* is a fault state of the *Infrastructure* component. (2) Next, observable variables have been identified where useful information about the states of the unobservable system components can be obtained. These are RTT and packet/frame retransmission rate as described in Section 3.3.5. (3) Intermediate variables have been specified that describe system behavior and relations between observations and the system components. As an example the condition of the component *wireless link* affects the observation *packet-retransmission rate* through the intermediate variables *packet loss.* (4) Finally, variables are represented as nodes in $\mathcal{G}$ and edges are identified from causal relations between variables. In the following a short description of intermediate nodes are given. More details can be found in [104].

Cross traffic load - The cross-traffic load is the load on the bottleneck drop tail queue router ($R_{(A/B)0}$) in Figure 3.3) which is assumed to have a constant service rate.

Upstream - The transmitted data is modelled by *Upstream*, that represents the actual rate of outgoing packets from the sender. It is assumed that there is always data to be sent from the application.

TCP - The *TCP* node covers the transmission rate control mechanisms of TCP. TCP controls the upstream based on RTT and detection of packet loss.

Throughput - The rate of successfully acknowledged data.

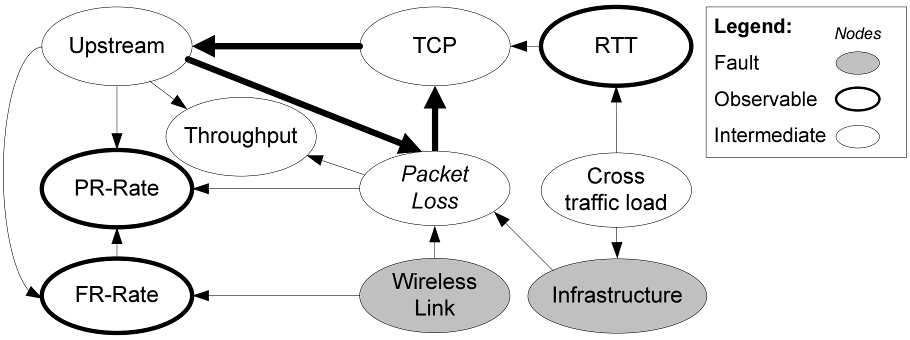

Figure D.1: *Intermediate model.*

Unconnected nodes in the model are interpreted as causally independent variables. Some important independence assumptions are:
Upstream→Cross traffic load - It is assumed in this model, that *Upstream* $\ll$ *Cross traffic load*. Thus, load on the bottleneck link is not significantly influenced by the upstream of the sending application. This also means that the influence of *Upstream* on *Congestion* is considered insignificant.
Wireless link→Infrastructure - The two components of the network have no (significant) influence on each other, thus the nodes are regarded as being independent.

D.2 BN Model

The model in Figure D.1 is not a BN. It contains a cycle *Upstream→Packet loss→TCP→Upstream*. For this type of problem a dynamic BN could be introduced (see [101]). However, the focus in this work is to use a regular BN and thus the cycle must be eliminated. Moreover, some of the intermediate nodes may be disregarded for simplicity. The main steps to achieve this are:
1) The *TCP*-node is removed to eliminate the cycle. This means that the impact from the control mechanisms of TCP are left unmodeled.
2) Being controlled by TCP the *Upstream*-node is eliminated from the model by converting packet retransmission rate and frame retransmission rate $(\frac{retransmissions}{s})$ into ratios $(\frac{retransmissions}{all-transmissions \cdot s})$ between sent and retransmitted packets/frames (PRR and FRR). Thus, upstream is contained in these observations.
3) Maintaining *Throughput* as an observation is an option. However, immediate TCP actions (e.g. reducing window size) can have a delayed impact on throughput observations. As the BN does not express such causal relations in time having throughput as an observation can be difficult.
4) The state of the *Infrastructure* node is directly defined by the *Cross traffic load* node. Thus, these two nodes are joined.
5) To maintain the relation that both infrastructure congestion and a poor wireless link lead to packet loss, the intermediate variable representing packet loss remains in the model.

The final outcome is the BN of Figure 4.3 defined as the *basic model*. Both the defined faults are modeled as two-state discrete random variables: Infrastructure, $I = \{Normal,\ Congested^*\}$ and Wireless Link,
$WL = \{Normal,\ Contended^*\}$. States marked with asterisk are the identified fault states. As no knowledge of features in observations are known in advance, simple state spaces have been defined. The RTT is represented by a set of equally spaced intervals representing the thresholds between discrete states. The FRR is defined by *high* and *low* divided by a single threshold. PRR is defined by *high*, *medium* and *low*. Compared to FRR (which is only depending on wireless link conditions) the introduction of an additional state enables different expressions of when a single or two faults have occurred. Finally, the states of the packet loss variable are *high* and *low*. For an overview of the states see Table D.1.

D.3 FDD Framework and Parameters

Data is needed to parametrize the BN and the OT to perform fault-diagnosis. A framework has been developed that contains functionality for supplying observation data. The framework is illustrated in Figure D.2. The framework has been realized in an ns-2 simulation environment that implements the network model of Figure 3.3 and using the parameterizations of Table 4.1.

Initially, data from the observation points can be generated by simulation as done in this work, or in practise, read from a network log file or monitored real-time from a communication process. Next, the observation point data is processed into evidence e. After processing, the evidence is propagated in the BN and network state estimates are inferred. For the OT the same processed observations are used and state estimates are given directly from evidence (cf. section D.3.2).

Inference in the BN is done by calculating the posterior probabilities $P(I|e)$ and $P(WL|e)$. Posterior probabilities enable an estimate of which state a variable is in and we assume that a fault state is diagnosed if $P(R = \text{fault-state}|e) > 0.5$. Inference can be performed using either exact inference [133], which is NP-hard or alternatively approximate inference methods [124]. The small BN model considered, implies only little computational overhead. Based on 9000 inference cases the average inference time is 0.5 ms (standard laptop running Linux) and

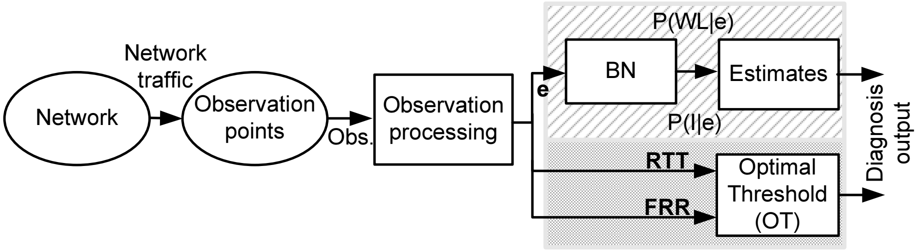

Figure D.2: *Overall diagnosis framework for comparing the BN and basic threshold (OT) diagnosis approaches.*

thus exact inference is applied.

D.3.1 Observation Processing

As emphasized previously BNs (and the OT) are not inherently capable of handling causality in (infinite) time. Instead, functions for processing observations and extracting timely features are utilized in the observation pre-processing. Examples of such functions are simple mean or variance estimates or more advanced auto-regressive functions [73]. In this work mean estimators are used, sampled in discrete steps using a moving average. It is assumed that mean estimates in this initial approach can provide useful statistics to infer about network states.

The size of the window for the mean estimators is set to 300 ms equally for all observations. Fixing this value is done to decrease the size of the parameter space and ensure that fault diagnosis is provided within hundreds of milliseconds. It should be noted that the value of the window size influences the performance of the BN significantly [102]. Collecting observations and performing state estimation is done periodically. The choice of sampling period is determined by the dynamics of the observations. In this work, sampling is done with a sampling period of $T = 100\ ms$. From early simulation and test results, this setting proves reasonable to capture the dynamics of RTT and retransmission ratios.

D.3.2 State-spaces and CPDs

The state-spaces of the individual observation nodes have been defined based on the optimal thresholds γ_{RTT}^{MPE} and γ_{FRR}^{MPE}. This is done to pursuit good observations for the BN and ensure comparability with the OT approach. As the OT approach does not specify a threshold for PRR, for consistency, optimal thresholds based on the MPE decision rule have been applied for γ_{PRR}^{MPE} as well.

The optimal thresholds are derived from two marginal distributions of each observation; one distribution where no fault occurs and one where it does. As all observation nodes have a different number of states, the optimal thresholds are applied differently for each observation node in the BN. The RTT has 12 states. In this case, γ_{RTT}^{MPE} is separating state 3 and 4, based on $Infrastructure$ fault occurrences. In the PRR case, the observation is influenced by both faults which can be generalized into representing 0, 1 or 2 faults occurring (3 states). Thus, these states are separated by two optimal thresholds, $\gamma_{PRR,1}^{MPE}$ and $\gamma_{PRR,2}^{MPE}$. To obtain these, an additional marginal distribution is used for PRR when both faults occur. The FRR observation node has two states separated by γ_{FRR}^{MPE} based on $Wireless\ Link$ fault occurrences.

The conditional probability distributions (CPDs) in the BN are elicited based on *learning*. By using the Expectation-Maximization (EM) algorithm it is possible to estimate the parameters of a probabilistic model [71]. In the BN, the parameters are the specific probabilities of the causal relations, e.g. $P\left(RTT \mid I\right)$.

Data sets for learning have been generated from the simulation containing information from both observations and network states. The probabilities of the

	RTT	PRR	FRR
States	12	low/medium/high	low/high
Thresholds	62-105 ms Interval: 4.3	0.02, 0.05	0.029

PL	States	2 (Low, High)
I, WL	P(I), P(WL)	(0.5, 0.5), (0.5, 0.5)

Table D.1: *BN node and parameter setup.*

	P(PL | I, WL)		P(PRR | FRR, PL)		P(FRR | WL))		
	Low	High	Low	High		Low	High
0,0	0.93	0.07	0.97	0.03	0	0.85	0.15
1,0	0.63	0.37	0.45	0.55	1	0.41	0.59
0,1	0.35	0.65	0.18	0.82			
1,1	0.13	0.87	0.07	0.93			

0 → *normal/low*, 1 → *fault/high*

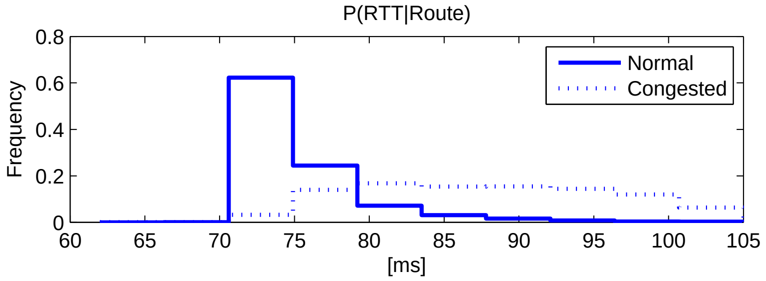

Table D.2: *CPDs for PL, PRR and FRR and RTT.*

network states, i.e. infrastructure and wireless link, represent the prior *belief* of faults occurring. Here, they have been specified equally as 50%/50% as we assume to have no prior belief of whether normal states are more dominant than fault states. All basic model configuration parameters are shown in Table D.1. Learned conditional probabilities are depicted in Table D.2.

Some results on the BN diagnosis capabilities compared to the OT are presented in Section 4.5. More details can be found in [104].

Appendix E

PE Parameter Summary

This appendix summarizes a set of the main parameters used in the thesis to describe: the *Scenario* of Section 3.3 in Table E.1, diagnosis performance metrics applied throughout the thesis in Table E.2 and the common parameters used for policy evaluation in Chapter 7 and parameters of the initial setup of Chapter 8 in Table E.3.

Scenario Parameters		
Parameter	***Name***	***Description***
End-user Service		
$data_{size}$	Data size requirement	Application requirement for amount of data to be transferred before $t_{deadline}$.
$t_{deadline}$	Deadline requirement	Time deadline requirement in which a data transfer must be completed.
Ω	Probability of successful transfer	Reliability metric; probability of completing the data transfer of size $data_{size}$ within $t_{deadline}$.
Fault Model (congestion)		
Λ_f	Fault occurrence rate	Parameter of a geometric distribution defining the OFF state/normal duration.
Λ_r	Fault repair rate	Parameter of a geometric distribution defining the ON state/fault duration.
λ_{cg}	Cross-traffic rate	Network state control parameter *option 1*: Rate of generated infrastructure cross-traffic connections. Cross-connections are TCP based transfers of data with sizes following a Pareto distribution ($\mu = 10\,KB,\ \beta = 1.5$ [8])
p_{cg}	Independent losses rate	Network state control parameter *option 2*: Parameter controlling the independent loss process for different network states.
$RTT/dRTT$	Mean round-trip time.	RTT for a given network/end-to-end path. Obtained at Layer 4 (RTT) or by frame inspection at layer 2 ($dRTT$).
Remediation Option		
p_{fof}	Probability of failing fail-over	Probability that a fail-over will fail and that the transfer is continued in the network from which a fail-over has been attempted. The failing fail-over will lead to an end-user service disruption for a period defined by p_{fdelay}.
p_{fdelay}	Failing fail-over delay	Geometric distribution parameter defining the end-user service disruption period when a failing fail-over occurs.

Table E.1: *Summary of general scenario parameters.*

Diagnosis Performance Metrics		
Metric	***Name***	***Description***
General Metrics		
RT	Reaction Time	Time from a fault occurs until it is correctly diagnosed with a true positive.
FA	False Alarm	A transition from a True Negative (TN) to a False Positive (FP) diagnosis estimate.
TA	True Alarm	A transition from a False Negative (FN) to a True Positive (TP) diagnosis estimate.
Independent Diagnosis Estimates (Memory-less Diagnosis Mechanisms)		
TPR (FNR)	True Positive (False Negative) Ratio	Describes the steady state relation between True Positives (TP) and False Negatives (FN) as: $TPR = \frac{\#TP}{\#TP+\#FN}$ ($FNR = 1 - TPR$)
TNR (FPR)	True Negative (False Positive) Ratio	Describes the steady state relation between True Negatives (TN) and False Positives (FP) as: $TNR = \frac{\#TN}{\#FP+\#TN}$ ($FPR = 1 - TNR$)
Application Specific Metrics (Temporal Diagnosis Mechanisms)		
μ_{RRT}	Mean Remediation Reaction time	This metric describes the mean time until a remediation action occurs since end-user service initiation (t_{init}). The metric does not distinguish between whether a fail-over is caused by a false alarm or a true alarm.
μ_{FRT}	Mean Fault Reaction Time	Describes the mean time from a fault occurs until it is correctly diagnosed. Is only valid for a continued fault period (i.e. without intermediate repair and fault events).
p_{RFA} (p_{RTA})	Probability of Remediation on False (True) Alarm	The metric p_{RFA} defines the probability that the fail-over will be commenced based on a false alarm under the following assumption: an end-user service operating in infinite time will eventually lead to a diagnosis case and a fail-over. Notice: $p_{RTA} = 1 - p_{RFA}$ (Remediation on True Alarm).

Table E.2: *Summary of diagnosis performance metrics.*

Parameter	**Description**	**Value/Range**
$t_{deadline}$	Deadline where the complete file must be transmitted.	$30\,s$
$data_{size}$	Size of data amount to be transferred	$10,000\,KB$
p_{fdelay}	Mean return delay (geometrically distributed) when a fail-over fails	$1.2\,s$
p_{fosn}	Probability that a fail-over from current to to remediation network succeeds and that the remediation network is in a good state.	0.43
p_{fosf}	Probability that a fail-over from current to to remediation network succeeds and that the remediation network is in a congested state.	0.52
p_{fof}	Probability that a fail-over fails and the remediation returns transmission to the *primary* network.	0.05
T	Diagnosis Period.	$0.398\,s$
γ^0_{RTT}, (TNR, TPR)	Diagnosis performance set γ^0_{RTT}	$(0.984, 0.102)$
γ^1_{RTT}, (TNR, TPR)	Diagnosis performance set γ^0_{RTT}	$(0.953, 0.225)$
Λ_f	Fault Occurrence Rate	$0.0805\,s$
Λ_r	Repair Occurrence Rate	$0.0667\,s$
Λ_{dtn}	Normal State Mean Transfer Rate	$453\,KB/s$
Λ_{dtf}	Fault State Mean Transfer Rate	$220\,KB/s$
n	File transfer progress birth chain states.	26
m	Time progress birth chain states.	10

Table E.3: *Summary of common parameters used in the results of the policy evaluation models.*

Appendix F

Adaptation and Changes

This Appendix provides background material on dynamic scenarios and adaptation presented in Chapter 8. Section F.1 presents details on components in the end-to-end path, which of their parameters are expected to change and which adaptation tasks may be required in the ODDR. In section F.1 validation results on the proposed PE SAN model are presented for evaluation of known remediation policies.

F.1 Changes in the End-to-End Path

In this appendix the aim is to provide an overview of which parts of the end-node driven fault management scenario are dynamic, what the potential impact is on end-user service provisioning and how the ODDR must adapt to such changes. From the perspective of the ODDR component a new or changed environment can affect for instance available recovery options and their properties, fault types and their impact on end-user services, diagnosis capabilities, etc. Essentially, such changes may lead to non-optimal remediation decisions and as a worst case result in an increased amount of failures if the changes are not accounted for in the decision process. The outcome of this section is used to clarify which elements require adaptation and how various adaptation tasks can be grouped.

F.1.1 Definition of the End-to-End Path

In the following a definition of the components considered to compose the end-to-end path is provided. These definitions are in Section F.1 applied when describing the individual parameters that are considered to lead to changes.

Overall End-to-end Path Components

1. End-node device - *Defines End-Node capabilities. These capabilities may vary in terms of available hardware and software resources.*

   (a) Software - *Software in the End-Node having an impact on end-to-end performance and dependability.*

      i. End-user service - *End-user service type and requirements.*

ii. Protocol Stack - *Implementation of protocols (performance and dependability properties) and availability of observations. Also includes Network Interface drivers.*

iii. Subscriptions - *Defines which networks and service may be accessed and potentially their cost.*

iv. Other - *Other software in the end-node which may indirectly affect networking and end-user service performance due to faults or excessive resource consumption. These aspects are not handled by the ODDR component.*

(b) Hardware - *Defines processing and memory resources as well as physical network access options (technology and supported settings).*

2. Wireless access networks - *Entails the wireless access part of the end-to-end connection which may consist of a single hop to a base station/access point or multiple hops in an ad-hoc network.*

3. Radio Access Network (RAN) - *The infrastructure that connects the access point and/or base station to the core network.*

4. Core Network - *The physical/logic core network operated by one or more end-user service providers. Core networks are defined to offer connectivity between RANs and End-user Service Provider Networks.*

5. End-User Service Provider Network - *Network associated with the used end-user service.*

6. End-User Service Provisioning Architecture - *The architecture of potentially several end-points which may deliver end-user service provisioning. May be organized as a central or distributed cluster with a single or multiple entry points.*

7. End-User Service Provider End-Point - *The end-user service provisioning system. It is defined by its capabilities in terms of hardware and software resources.*

   (a) Software - *Software in the End-User Service Provider End-Point having an impact on end-to-end performance and dependability.*

   (b) Hardware - *Hardware in the End-User Service Provider End-Point having an impact on end-to-end performance and dependability.*

F.1.2 Identification of Changing End-to-End Parameters

In this section a list of parameters of the end-to-end path are identified. These parameters are expected to change over time which requires adaptation of the ODDR component in the end-node to provide useful fault management. The identified parameters take a starting point in a generalization of the scenario depicted in Figure 3.2. A graphical representation of how these parameters may be associated in the interdependencies is presented in Figure 8.1.
Each parameter is described using the following paragraphs:
Part of end-to-end path - Defines in which part of the end-to-end path the parameter reside. This also involves parts of the end-to-end path which have an

impact on the changes of the given parameter.

Description - A general description of the parameter and some examples of its role in the end-node driven fault management approach.

Expected impact - Expected impact on the end-user service provisioning and the ODDR behavior as the parameter changes.

Required Adaptation - An overall description of which adaptations may be required in the ODDR component to handle when this specific parameter changes.

Available Access Networks

Part of end-to-end path - Wireless access networks, end-node device hardware (technologies and configurations) and software (protocol stack and subscriptions)

Description - The available access networks typically define different network access opportunities for an end-node. The network access is considered to be wireless using some arbitrary technology. A different wireless access network is not inherently defined by also offering another access path in the radio access and core network; i.e. two different WLAN access points in range may make use of the same physical infrastructure to the core network. The end-node may choose the most optimal access methods from a variety of properties such as cost, QoS requirements and dependability. From an ODDR perspective changing the access network is also one of the more significant approaches to attempt to correct faults. Thus, it is considered a central remediation option to change the access network.

The properties of individual access networks may vary significantly based on technology (frequency band, channel access scheme, range), load conditions and, in the context of end-node driven fault management; to what extent they provide alternative end-to-end paths to provide remediation of certain faults.

Expected impact - Clearly, at least a single available access network is required for a wireless end-node to initiate and maintain any end-user service in an infrastructure or ad-hoc network (an active link). The extent of additional available access networks is expected to have a significant influence on which options an end-node has to initiate remediation actions. If (alternative) available access networks (redundant paths towards the end-user service) do not exist there may be no options to resolve a fault in the active link. Further, the diversity of the offered access networks (in terms of technology, configuration and architecture they provide access to) also has a significant impact on which fault may be remediated.

Altogether, the end-node must be closely updated with which networks are available; at least to an extent where remediation solutions may be available for the most likely faults that are bound to lead to end-user service failure.

Required Adaptation - The ODDR component must maintain a *sufficient* set of available access networks in its list of available remediation options. When available access networks appear or disappear alternatives must be reevaluated as determined by the *decision module*. *Sufficiency* in this context may be defined by the requirements of end-user services, cost considerations of searching for new access networks and resource consumption considerations of reevaluating new decision policies in the end-node.

Network Resources

Part of end-to-end path - End-node device, software, subscriptions. End-node device hardware. Wireless access network. Radio access network. End-user Service Provider Network. It is assumed that the core networks have sufficient static network resources to not become a bottleneck.

Description - Maximum available network resources for a given end-node path. Note, this definition does NOT consider *available* network resources. Only the maximum amount of resources that may be used by an end-node considering no impact from other users sharing the same resource.

Expected impact - A change in the available network resources may have a significant impact on the properties of certain end-to-end path options. E.g. an available WLAN access point which has its xDSL RAN link upgraded may offer improved connectivity options into the infrastructure network.

Required Adaptation - Update expected maximum performance of a single or multiple available access network options (depending on where the change apply). Re-evaluate certain fault-type and impact definitions which may be affected by the change in available resources.

Network Resource Consumption

Part of end-to-end path - All excluding end-node.

Description - This parameter considers long-term evolutions of network resource consumption. The parameter may be defined as a mean resource consumption on a certain end-to-end path. Long-term here refers to changes that occur over hours, days and months. Hours can be considered as time-of-day changes, days can be considered as time-of-week changes whereas monthly changes may be considered as changes in individual user requirements as new services are used or as load moves to other parts of the network.

Notice, short-term changes in terms of network resource consumption are considered from a fault perspective as defined by e.g. *congestion faults* or *contention faults*.

Expected impact - Network resource consumption will clearly have an impact on available network resources for a given end-node. Thus, it may change properties of available access networks and the network paths they offer as well as the end-user service impact of certain fault types.

Required Adaptation - Update expected maximum performance of a single or multiple available access network options (depending on where the change applies). Re-evaluate certain fault-type and impact definitions which may be affected by the change in available resources.

Fault Instance

Part of end-to-end path - All excluding end-node (for delimitation reasons only network-related faults are considered).

Description - A fault instance (generally just referred to as a fault in this work) may be described by the properties:

- Fault Type
- Fault Location
- Fault Process

These properties are defined as follows:
Fault Types describe the nature of a fault. Examples of such fault types are congestion, contention, crash, radio noise, etc. It is assumed that a specified set of fault types are handled by the ODDR component. Which fault types and in which links of the error-fault(-failure) chain they need to be specified are up to the system designer and/or autonomic fault definition approaches. A given fault instance may occur in different parts of the network which will require different remediation actions. This property is defined as *Fault Location*. The *fault process* specifies how a given fault instance may be activated or not. An example of a simple fault process is an ON-OFF process with rates Λ_{repair} and Λ_{fault} parameterizing exponentially distributed holding time distributions for the ON and OFF process respectively. The fault process may also be parametrized to specify intermittent and permanent fault processes.

Expected impact - A given end-to-end path will be dominated by a subset of likely faults. As such faults become more or less severe (in terms of location and process) this will also influence most optimal remediation decisions. E.g. a fault which changes from a long ON-cycle (fault state) to an expected short ON-cycle may have a significantly reduced impact.

Required Adaptation - Fault instance parameters in the diagnosis model and the decision model should be adapted to correspond to observed fault processes.

Fault Impact

Part of end-to-end path - All excluding end-node (for delimitation reasons only network-related faults are considered).

Description - The fault impact determines how a certain fault instance affects various types of end-user services. For now these different end-user types remain unspecified. Thus, to define the full effect of a fault on the reliability and performance metrics of an end-user service both the *Fault*

Instance definitions and its *Fault Impact* must be know. As an example of changing impact of a fault could be a router restarting occasionally. This may in occasional periods lead to traffic streams being diverted leading to congestion on a link in the considered end-to-end path. As the load characteristic on the restarting router may be reduced/or increased this also changes the impact on the congested link and thereby the end-user service.

Expected impact - A change in fault impact may affect the observations used to diagnose the fault as well as the impact on the end-user service performance and reliability.

Required Adaptation - Update end-user service impact definitions to ensure proper remediation decisions. Also update the diagnosis component based on changes in the observations. Potentially invalidate a fault instance if it has an insignificant effect on most considered end-user service in the end-node.

End-Node Device Capabilities

Part of end-to-end path - End-node.

Description - Device capabilities covers the hardware and software of the end-node device as described in Section F.1.1. Changing device capabilities may also significantly change the options of the ODDR component to collect observations and perform remediation.

Expected impact - Hardware changes such as adding or removing wireless interfaces can significantly change the extent of available remedation options and their diversity. Also, having multiple radio interfaces (even within the same technology) may enable new properties such as searching for remediation options while continuing service provisioning on the active path. Software changes may lead to different performance and/or reliability characteristics. E.g. an end-node protocol stack may become better at handling a specific fault type.

Required Adaptation - Hardware changes are expectedly rare (if not caused by a fault) and may be deterministically determined in the end-node. Software changes may also be deterministically registered if they lead to new functionality. More subtle changes such as performance optimizations may be more difficult to identify. Both cases may, however, require a re-evaluation of fault impact, remediation option capabilities and consequently best decision strategies in the decision module.

Available end-user Service Points

Part of end-to-end path - End-User Service Provider End-Point,End-User Service Provider Network and End-User Service Provisioning Architecture.

Description - Defines available end-points where a given end-user service can be obtained. An example is an FTP-download service where a copy of a certain file can be obtained from several provisioning points.

Expected impact - Changes in available end-user service points also changes potential remediation strategies for an end-node to change the end-to-end path in addition to changing access network. Additional end-user service points in different locations of a network may therefore provide a significant option of improvement in end-user service performance and dependability. It is, as in case of the *Available Access Networks* necessary that the end-points deliver sufficient diversity to provide remediation actions for different faults.

Required Adaptation - When applicable the ODDR component must maintain a list of available end-user service points which may be used interchangeably. When available end-user service points appear and disappear alternatives must be reevaluated as determined by the *decision module*.

Requirements for End-user Service

Part of end-to-end path - End-node software.

Description - An end-node makes use of the ODDR component to provide optimization of primarily dependability aspects of end-user services. The end-user service may set its requirements to the ODDR component at the initiation of an end-user service and potentially during the lifetime of the end-user service session.

Expected impact - End-user service requirements have a significant impact on the decision behavior of the ODDR. I.e. if requirements are loose (e.g. low throughput and high delay can be tolerated) the ODDR component will minimize communication overhead (e.g. observations and execution of remediation actions) and perform close to best effort. If requirements are tight the ODDR component will more aggressively attempt to identify best networks and assess faults and their impacts.

Required Adaptation - Redefinitions of end-user service requirements is a change that is expected. The decision rules need to reflect different settings of requirements.

Infrastructure Network Paths

Part of end-to-end path - Radio Access Network. Core Network. End-User Service Provider Network.

Description - An end-to-end path may span multiple different networks maintained by different service providers. Considering the end-to-end path from the Wireless Access Network to the end-user service provisioning point the actual path may change. Such changes may occur on a packet level, during the provisioning of an end-user service and on longer time scales as infrastructure networks are reconfigured. In addition, changes may occur independent on uplink and downlink paths.

Expected impact - It may be difficult to diagnose and recover from faults in non-stationary paths as changing paths may lead to an instant change in observations as well as the actual state of the network.

Required Adaptation - For short-term changes these may be modeled as a part of the fault-process on a given end-to-end path. For longer term changes the end-to-end part may need to be re-assessed in terms of infrastructure network faults.

Observation Properties

Part of end-to-end path - All.

Description - The diagnosis component uses observations to determine the state of the network used by the active end-to-end path as well as remediation region networks. These observations may change character as other parts of the network change. A simple example is if the network path changes. In that case a different hop-count may lead to a different mean round-trip time (RTT).

Expected impact - As observations change properties this will affect the performance of the diagnosis component.

Required Adaptation - The diagnosis component need to adapt to changing observation properties.

Diagnosis Components Properties

Part of end-to-end path - End-node.

Description - Diagnosis component properties describe the performance of the diagnosis component(s) under imperfect observations. Relevant performance metrics may be False Alarm Ratio and Reactivity Time, As exemplified under *Observation Properties* the properties of a diagnosis component may change as observations from the network change.

Expected impact - Changing diagnosis properties have a clear impact on which remediation decision policies to apply as shown in [66]. A non-optimal decision policy may lead to reduced end-user service reliability or even worse performance than obtained by no fault management at all.

Required Adaptation - The applied decision policies must be updated.

F.2 Möbius Policy Evaluation Results

This section introduces evaluation results of policies in the PE SAN introduced in Section 8.3. The evaluated policies and policy evaluation results are known from the PE DTMC model in Section 7.3. Thus, the following results serve as a validation of the PE SAN model to deliver consistent policy evaluation results. A summary of the parameters used for evaluation can be found in Appendix E.

The outcome of the evaluations are depicted in Figure F.1 and Figure F.2, for the imperfect diagnosis settings of γ_{rtt}^{0} and γ_{rtt}^{1}, respectively. Overall, there seem to be a good match between these PE SAN results and the PE DTMC results in Section 7.3 when considering the evaluated policy evaluation outcomes. Some numerical differences apply however where the PE SAN model

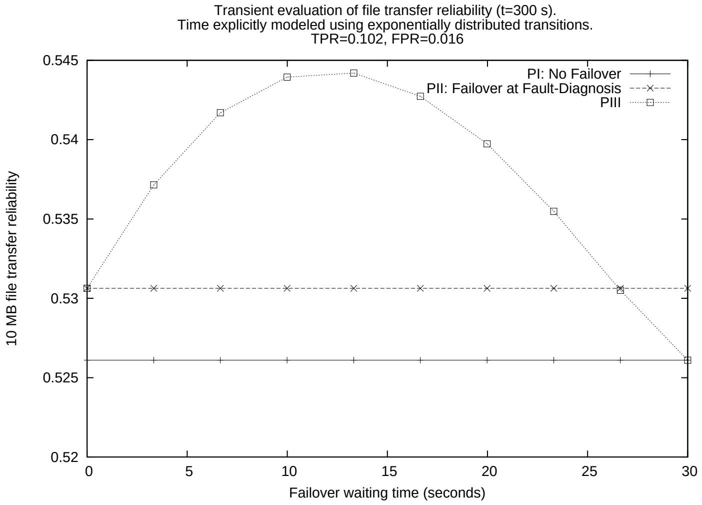

Figure F.1: *Results of SAN evaluation for remediation policies **(PI)**, **(PII)** and **(PIII)**. (γ^0_{rtt})*

provides lower values for Ω_{model}. These differences may partially arise from differences between the CTMC underlying the time and data transfer model of the SAN model solution and the DTMC of the PE DTMC model. However, these differences have not been studied in details. In this respect it is assumed that the PE SAN model can provide policy evaluation consistently to the PE DTMC model.

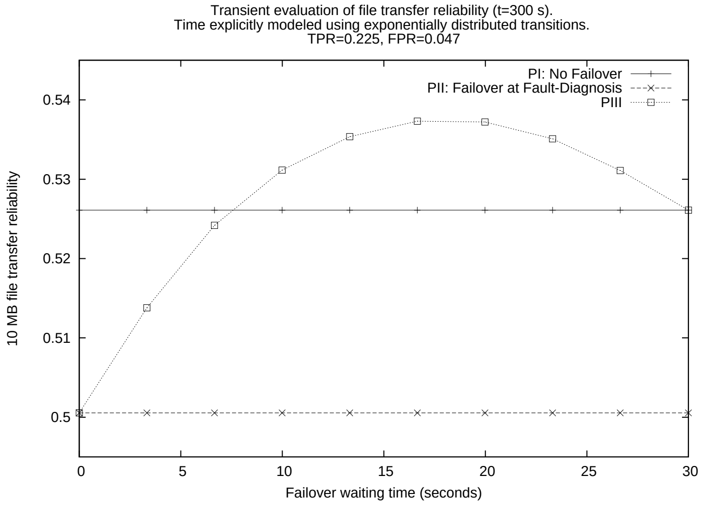

Figure F.2: *Results of SAN evaluation for remediation policies PI, PII and PIII.* (γ^1_{rtt})

Appendix G

Abbreviations

AP Access Point

BS Base Station

CSMA Carrier Sense Multiple Access

DAG Directed Acyclic Graph

DNS Domain Name System

ETSI European Telecommunications Standards Institute

HMM Hidden Markov Model

IMS IP Multimedia Subsystem

ITU-T International Telecommunication Union

LTE Long Term Evolution

VANET Vehicular Ad-Hoc Network

MADM Multiple Attribute Decision Making

MDP Markov Decision Process

ME Matrix Exponential

MPLS Multiprotocol Label Switching

MTBF Mean Time Between Failures

MTTR Mean Time to Repair

NGI Next Generation Internet

NGN Next Generation Networking

RTT Round-Trip Time

ROC Receiver Operation Characteristic

PER Packet Error Ratio

PSTN Public Switched Telephone Network

SCTP Stream Control Transmission Protocol

SNMP Simple Network Management Protocol

TCP Transmission Control Protocol

UDP User Datagram Protocol

VoIP Voice over IP

QoS Quality of Service

Bibliography

[1] 3GPP. Quality of service (qos) concept and architecture, January 2009. TS 23.107 version 8.0.0 Release 8).

[2] 3rd Generation Partnership Project 3GPP. 3gpp: Lte. e-utran architecture description 3gpp ts 36.401. Technical Report V8.4.0 (2010-02), ETSI, 2010. Technical Specification.

[3] 3rd Generation Partnership Project 3GPP. All-ip network (aipn) feasibility study. Technical Report V9.0.0 (2010-02), ETSI, 2010. Technical Specification.

[4] M. Afergan, J. Wein, and A. LaMeyer. Experience with some principles for building an internet-scale reliable system. In Proceedings of the 2nd conference on Real, Large Distributed Systems, volume 2, 2005.

[5] Jay Aikat, Jasleen Kaur, F. Donelson Smith, and Kevin Jeffay. Variability in tcp round-trip times. In IMC '03: Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement, pages 279–284, NY, USA, 2003. ACM.

[6] AKARI. New generation network architecture akari conceptual design. Technical report, AKARI Project, April 2007.

[7] R. Alamgir, M. Atiquzzaman, and W. Ivancic. Effect of Congestion Control on the Performance of TCP and SCTP over Satellite Networks. In NASA Earth Science Technology Conf, 2002.

[8] E. Altman and T. Jiménez. Ns simulator for beginners. Technical report, Univ. de Los Andes, Mérida, Venezuela and ESSI, 2003.

[9] Per Ola Andersson, Håkan Asp, Aldo Bolle, Harry Leino, Peter Seybolt, and Richard Swardh. Gsm transport evolution. Ericsson Review No. 1, 2007.

[10] Davide Anguita. Smart adaptive systems: state of the art and future directions of research. In Proceedings of the European Symposium on Intelligent Technologies, Hybrid Systems and their Implementation on Smart Adaptive Systems - EUNITE 2001, pages 1–4, 2001.

[11] Algirdas Avizienis, Jean-Claude Laprie, Brian Randell, and Carl Landwehr. Basic concepts and taxonomy of dependable and secure computing. IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, 1(1):11–33, 2004.

[12] P. Bahl, R. Chandra, A. Greenberg, D.A. Maltz, and M. Zhang. Towards highly reliable enterprise network services via inference of multi-level dependencies. In Proceedings SIGCOMM'07, pages 13–24. ACM Press New York, NY, USA, 2007.

[13] M. Bailey, E. Cooke, F. Jahanian, D. Watson, and J. Nazario. The Blaster Worm: Then and Now. IEEE SECURITY & PRIVACY, pages 26–31, 2005.

[14] S. Balasubramaniam and J. Indulska. Vertical handover supporting pervasive computing in future wireless networks. Computer Communications, 27(8):708–719, 2004.

[15] Michèle Basseville and Igor V. Nikiforov. Detection of Abrupt Changes: Theory and Application, pages 25–65. IRISA/CNRS Rennes, France and Institute of Control Sciences Moscow, Russia, 1998. Online edition: http://www.irisa.fr/sisthem/kniga/.

[16] G. Bianchi. Performance analysis of the IEEE 802.11 distributed coordination function. Selected Areas in Communications, IEEE Journal on, 18(3):535–547, 2000.

[17] G. Bianchi et al. Performance analysis of the ieee 802. 11 distributed coordination function. IEEE Journal on selected areas in communications, 18(3):535–547, 2000.

[18] BIPM, IEC, IFCC, ISO, IUPAC, and OIML. Guide to the expression of uncertainty in measurement (GUM), 1993.

[19] BIPM, IEC, IFCC, ISO, IUPAC, and OIML. International Vocabulary of Metrology–Basic and general concepts and associated terms (VIM), JCGM 200, 2008. 3rd edition.

[20] BIPM, IEC, IFCC, ISO, IUPAC, and OIML. ISO international vocabulary of basic and general terms in metrology (VIM), third edition, 2008.

[21] IEEE-SA Standards Board. Part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications, amendment 5: Enhancements for higher throughput. Technical report, The Institute of Electrical and Electronics Engineers, Inc., October 2009.

[22] A. Bondavalli, F. Brancati, and A. Ceccarelli. Safe estimation of time uncertainty of local clocks. In Proc. of Int. IEEE Symp. on Precision Clock Synch. for Measur., Contr. and Comm., ISPCS 2009, pages 47–52, 2009.

[23] A. Bondavalli, A. Ceccarelli, L. Falai, and M. Vadursi. Foundations of measurement theory applied to the evaluation of dependability attributes. Proc. IEEE/IFIP DSN, I, Washington, DC, 2007, pages 522–533, 2007.

[24] A. Bondavalli, A. Ceccarelli, J. Grønbæk, D. Iovino, L. KÁRNÁ, Š. KLAPKA, T.K. Madsen, M. Magyar, I. Majzik, and A. Salzo. Design and evaluation of a safe driver machine interface. International Journal of Performability Engineering, 5(2):153–166, 2009.

[25] A. Bondavalli, S. Chiaradonna, F. Di Giandomenico, and F. Grandoni. Threshold-based mechanisms to discriminate transient from intermittent faults. IEEE Transactions on Computers, 49(3):230–245, 2000.

[26] Andrea Bondavalli, Paolo Lollini, and Leonardo Montecchi. QoS Perceived by Users of Ubiquitous UMTS: Compositional Models and Thorough Analysis. Journal of Software, 4(7), 2009.

[27] Pierre Borgnat, Guillaume Dewaele, Kensuke Fukuda, Patrice Abr, and Kenjiro Cho. Seven years and one day: Sketching the evolution of internet traffic. In proceedings of IEEE Infocom 2009, pages 711–719, 2009.

[28] L.S. Brakmo, S.W. O'Malley, and L.L. Peterson. TCP Vegas: new techniques for congestion detection and avoidance. ACM SIGCOMM Computer Communication Review, 24(4):24–35, 1994.

[29] A. Caro, K. Shah, J. Iyengar, P. Amer, and R. Stewart. SCTP and TCP Variants: Congestion Control Under Multiple Losses. submitted to ACM Computer Communication Review, February, 2003.

[30] R.L. Carter and M.E. Crovella. Measuring bottleneck link speed in packet-switched networks. Performance evaluation, 1996.

[31] J. Case, R. Mundy, D. Partain, and B. Stewart. RFC3410: Introduction and Applicability Statements for Internet-Standard Management Framework. RFC Editor United States, 2002.

[32] A. Casimiro, P. Lollini, M. Dixit, A. Bondavalli, and P. Veríssimo. A framework for dependable QoS adaptation in probabilistic environments. In Proceedings of the 2008 ACM symposium on Applied computing, pages 2192–2196. ACM, 2008.

[33] C.G. Cassandras and S. Lafortune. Introduction to discrete event systems. Kluwer Academic Boston, 1999.

[34] D. Chen, S. Garg, C. Kintala, and KS Trivedi. Dependability enhancement for IEEE 802.11 wireless LAN with redundancy techniques. Dependable Systems and Networks, 2003. Proceedings. 2003 International Conference on, pages 521–528, 2003.

[35] W.T. Chen and Y.Y. Shu. Active application oriented vertical handoff in next-generation wireless networks. In 2005 IEEE Wireless Communications and Networking Conference, volume 3, 2005.

[36] G. Clark, T. Courtney, D. Daly, D. Deavours, S. Derisavi, J.M. Doyle, W.H. Sanders, and P. Webster. The M
"obius modeling tool. In Proceedings of the 9th International Workshop on Petri Nets and Performance Models, pages 241–250. Citeseer, 2001.

[37] HIDENETS Consortium. Highly dependable ip-based networks and services - final evaluation, consolidated results and guidelines, January 2009. Deliverable 1.3.

[38] Internet Systems Consortium. Isc domain survey: Number of internet hosts, January 2010. Number of Hosts advertised in the DNS.

[39] A. Daidone, F. Di Giandomenico, S. Chiaradonna, and A. Bondavalli. Hidden Markov models as a support for diagnosis: Formalization of the problem and synthesis of the solution. In 25th IEEE Symposium on Reliable Distributed Systems, 2006. SRDS'06, pages 245–256, 2006.

[40] M. Dischinger, A. Haeberlen, K.P. Gummadi, and S. Saroiu. Characterizing residential broadband networks. In Proceedings of the 7th ACM SIGCOMM conference on Internet measurement, page 56. ACM, 2007.

[41] S. Dobson, S. Denazis, A. Fernández, D. Ga
"ıti, E. Gelenbe, F. Massacci, P. Nixon, F. Saffre, N. Schmidt, and F. Zambonelli. A survey of autonomic communications. ACM Transactions on Autonomous and Adaptive Systems (TAAS), 1(2):259, 2006.

[42] R. Doverspike and B. Cortez. Restoration in carrier networks. pages 45–54, oct. 2009.

[43] T. Dreibholz and E.P. Rathgeb. Reliable Server Pooling–A Novel IETF Architecture for Availability-Sensitive Services. In Proceedings of the 2nd IEEE International Conference on Digital Society (ICDS), Sainte Luce/Martinique, pages 150–156. IARIA, 2008.

[44] DVB-Project-Office. Internet protocol datacast. Technical report, DVB-organization, April 2008. Standard Architecture described in ETSI TR 102 469 v1.1.1.

[45] N. Ehsan and M. Liu. Analysis of TCP transient behavior and its effect on file transfer latency. Communications, 2003. ICC'03. IEEE International Conference on, 3, 2003.

[46] Eleazar Eskin, Matthew Miller, Zhi-Da Zhong, George Yi, Wei-Ang Lee, and Salvatore Stolfo. Adaptive model generation for intrusion detection systems. Workshop on Intrusion Detection and Prevention, 7th ACM Conference on Computer Security, November 2000.

[47] ETSI-TISPAN. Telecommunications and internet converged services and protocols for advanced networking (tispan) - ngn release 1. Technical report, ETSI, March 2006. ETSI TR 180 001.

[48] J.C. Fabre, M.O. Killijian, and T. Pareaud. Towards On-line Adaptation of Fault Tolerance Mechanisms. In 2010 European Dependable Computing Conference, pages 45–54. IEEE, 2010.

[49] Kevin Fall and Kannan Varadhan. The ns Manual. UC Berkeley, LBL, USC/ISI, and Xerox PARC, May 2010.

[50] Niels Ferguson and Bruce Schneier. A cryptographic evaluation of ipsec, January 2003. http://www.schneier.com/paper-ipsec.html.

[51] D.R. Figueiredo, B. Liu, V. Misra, and D. Towsley. On the autocorrelation structure of TCP traffic. Computer Networks, 40(3):339–361, 2002.

[52] S. Floyd. Validation Experiences with the NS Simulator. Proceedings of the DARPA/NIST Network Simulation Validation Workshop, Fairfax, VA, May, April 1999.

[53] S. Floyd and V. Jacobson. Random early detection gateways for congestion avoidance. IEEE/ACM Transactions on networking, 1(4):397–413, 1993.

[54] N. Fonseca and M. Crovella. Bayesian packet loss detection for TCP. In Proceedings IEEE INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies, pages 1826–1837, 2005.

[55] S. Fortin-Parisi and B. Sericola. A Markov model of TCP throughput, goodput and slow start. Performance Evaluation, 58(2-3):89–108, 2004.

[56] Service Availability Forum. Service availability forum - service availability interface, February 2010. SAI-Overview-B.05.03.

[57] OpenSAF Foundation. An open source high availability middleware solution based on the application interface specification from the service availability forum, June 2009.

[58] OpenSAF Foundation. Rtp4 continuous services, June 2009.

[59] R. Fracchia, C. Casetti, C.F. Chiasserini, and M. Meo. WiSE: Best-Path Selection in Wireless Multihoming Environments. IEEE Transactions on Mobile Computing, 6(10):1130–1141, 2007.

[60] J. Gade, M. Dahl, P. Thorhaard, and F. Knudsen. Amphitm ambulance record keeping system. ournal of Clinical Monitoring and Computing, 20:117–144, 2006.

[61] M. Gales and S. Young. The application of hidden Markov models in speech recognition. Foundations and Trends in Signal Processing, 1(3):195–304, 2008.

[62] G. Gardikis, A. Kourtis, and P. Constantinou. Dynamic bandwidth allocation in DVB-T networks providing IP services. Broadcasting, IEEE Transactions on, 49(3):314–318, 2003.

[63] Global environment for network innovations (geni).

[64] C.M. Grinstead and J.L. Snell. Introduction to probability. American Mathematical Society, 2 edition, 1997. isbn:978-0821807491.

[65] J. Grønbæk, H.P. Frejek, T. Renier, and H.P. Schwefel. Client-Centric Performance Analysis of a High-Availability Cluster. Service Availability, pages 74–93, 2007.

[66] J. Grønbæk, Hans-Peter Schwefel, and T. S. Toftegaard. Model based Evaluation of Policies for End-Node Driven Fault Recovery. In 7th International Workshop on Design of Reliable Communication Networks, 2009. Proceedings., pages 367–374, 2009.

[67] Jesper Grønbæk, Hans Peter Schwefel, Jens Kristian Kjærgård, and Thomas Toftegaard. Assessing the impact of imperfect diagnosis on service reliability: A parsimonious model approach. In proceedings of 8th European Dependable Computing Conference (EDCC-8), 2010.

[68] ALARP Project Group. Alarp. a railway automatic track warning system based on distributed personal mobile terminals. annex 1 - description of work. Technical report, Ansaldo STS S.p.A. AND Forschungszentrum Telekommunikation Wien AND Università degli Studi di Firenze AND ResilTech S.r.l. AND Elbit Systems AND PROPRS Ltd. AND Darmstadt University of Technology - Institute for Ergonomics, October 2009.

[69] Network Working Group. RFC 2681 - A round-trip delay metric for IPPM, 1999.

[70] Q. Guo, J. Zhu, and X. Xu. An adaptive multi-criteria vertical handoff decision algorithm for radio heterogeneous network. In 2005 IEEE International Conference on Communications, 2005. ICC 2005, volume 4, 2005.

[71] D. Heckerman. A tutorial on learning with Bayesian networks. Technical report, Microsoft Research, Advanced Technology Division, March 1995.

[72] D. Heckerman. A tutorial on learning with Bayesian networks. Innovations in Bayesian Networks, pages 33–82, 2008.

[73] C.S. Hood and C. Ji. Proactive network-fault detection [telecommunications]. IEEE Transactions on reliability, 46(3):333–341, 1997.

[74] Markus C. Huebscher and Julie A. McCann. A survey of autonomic computing—degrees, models, and applications. ACM Computing Surveys, 40(3):1, 2008.

[75] Markus C. Huebscher and Julie A. McCann. A survey of autonomic computingŮdegrees, models, and applications. ACM Computing Surveys, 40(3), August 2008. Article No. 7.

[76] B. Hughes, R. Meier, R. Cunningham, and V. Cahill. Towards real-time middleware for vehicular ad hoc networks. VANET Š04, pages 95–96, 2004.

[77] IBM. An architectural blueprint for autonomic computing. Technical Report Fourth Edition, IBM, 2006. White Paper.

[78] IEEE. 802.11, part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications. IEEE Standard for Information Technology, June 2007. ISBN 0-7381-5656-6.

[79] Computer Industry Almanac Inc. Worldwide internet users top 1.5 billion in 2008. china tops 235m internet users., May 2009. Press Release.

[80] ITU-T ITU-T SG13. General principles and general reference model for next generation networks, itu-t recommendation y.2011. Technical report, ITU-T, October 2004.

[81] Finn V. Jensen. Bayesian Networks and Decision Graphs. Springer-Verlag New York, Inc., 2001.

[82] Kaustubh R. Joshi, Matti Hiltunen, Richard Schlichting, William H. Sanders, and Adnan Agbaria. Online model-based adaptation for optimizing performance and dependability. In WOSS '04: Proceedings of the 1st ACM SIGSOFT workshop on Self-managed systems, pages 85–89, New York, NY, USA, 2004. ACM.

[83] K.R. Joshi, M.A. Hiltunen, W.H. Sanders, and R.D. Schlichting. Automatic model-driven recovery in distributed systems. In 24th IEEE Symposium on Reliable Distributed Systems, 2005. SRDS 2005, pages 25–36, 2005.

[84] C.R. Kalmanek, Ihui Ge, Seungjoon Lee, C. Lund, D.a. Pei, J. Seidel, J. van der Merwe, and J. Ates. DarkStar: using exploratory data mining to raise the bar on network reliability and performance. Proceedings of Design of Reliable Communication Networks, pages 1–10, October 2009.

[85] A. Kamthe, M.Á. Carreira-Perpiñán, and A.E. Cerpa. M&M: Multi-level Markov Model for wireless link simulations. In Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems, pages 57–70. ACM, 2009.

[86] M. Kassar, B. Kervella, and G. Pujolle. An overview of vertical handover decision strategies in heterogeneous wireless networks. Computer Communications, 2008.

[87] S.J. Koh, M.J. Lee, M. Riegel, M.L. Ma, and M. Tuexen. Mobile SCTP for transport layer mobility. IETF draftsjkoh-sctp-mobility-03. txt, 2004.

[88] R. Koodly. Fast Handovers for Mobile IPv6 (FMIPv6). Technical report, IETF RFC 4068, July 2005.

[89] U. Ladebusch, CA Liss, N. Rundfunk, and G. Hamburg. Terrestrial DVB (DVB-T): a broadcast technology for stationary portable and mobile use. Proceedings of the IEEE, 94(1):183–193, January 2006.

[90] H. Liao and MJF Gales. Issues with uncertainty decoding for noise robust automatic speech recognition. Speech Communication, 2007.

[91] Lester Lipsky. Queueing Theory – A Linear Algebraic Approach. Springer, 2 edition, 2009.

[92] M.L. Littman, N. Ravi, E. Fenson, and R. Howard. An instance-based state representation for network repair. In Proceedings of the National Conference on Artificial Intelligence, pages 287–292. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2004.

[93] J. Liu, I. Matta, and M. Crovella. End-to-end inference of loss nature in a hybrid wired/wireless environment. In Proceedings of WiOptŠ03: Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks. INRIA, 2003.

[94] H. Lu, X. Zhou, and P. Hong. Improving the Performance of Fast Handovers in Mobile IPv6. In IEEE Global Telecommunications Conference, 2007. GLOBECOM'07, pages 1808–1812, 2007.

[95] L. Ma, F. Yu, VCM Leung, and T. Randhawa. A new method to support UMTS/WLAN vertical handover using SCTP. IEEE Wireless Communications, 11(4):44–51, 2004.

[96] Danny McPherson. 2% of internet traffic raw sewage, March 2008.

[97] A. Meddeb. Internet QoS: Pieces of the puzzle. IEEE Communications Magazine, 48(1):86–94, 2010.

[98] C.E. Metz. Basic principles of ROC analysis+. In Seminars in nuclear medicine, volume 8, pages 283–298. Elsevier, 1978.

[99] C. Middleton and A. Potter. Is it good to share? A case study of the FON and Meraki approaches to broadband provision. In International Telecommunications Society 17th Biennal Conference, Montreal, June 2008.

[100] A. Mishra, M. Shin, and W. Arbaugh. An empirical analysis of the IEEE 802.11 MAC layer handoff process. ACM SIGCOMM Computer Communication Review, 33(2):102, 2003.

[101] K.P. Murphy. Dynamic Bayesian Networks: Representation, Inference and Learning. PhD thesis, UNIVERSITY OF CALIFORNIA, 2002.

[102] A. Nickelsen and J. Grønbæk. Probabilistic fault detection in network communication. Technical report, Aalborg University, May 2006.

[103] A. Nickelsen, F. Paternó, A. Grasselli, K-U. Schmidt, M. Martin, B. Schindler, and F. Mureddu. OPEN: Open pervasive environments for migratory interactive services. In To appear in Proceedings of the 12th international conference on Information Integration and Web-based Applications and Systems (iiWAS2010). ACM, 2010.

[104] Anders Nickelsen, Jesper Grønbæk, Thibault Renier, and Hans Peter Schwefel. Probabilistic Fault-Diagnosis in Mobile Networks Using Cross-Layer Observations. In Proceedings of AINA 2009, Bradford, UK, May 26-29, 2009.

[105] J. Noonan, P. Perry, and J. Murphy. Client controlled network selection. In it IEE 3G Conf. 5th Intl. Conf. on 3G Mobile Communications, 2004.

[106] NSF. Find (future internet design), 2006. Homepage in operation fall 2010.

[107] R.C. Nunes and I. Jansch-Porto. QoS of Timeout-Based Self-Tuned Failure Detectors: The Effects of the Communication Delay Predictor and the Safety Margin. In Proceedings of the 2004 International Conference on Dependable Systems and Networks, pages 753–761. IEEE Computer Society, 2004.

[108] The GENI Project Office. Geni system overview. Technical report, BBN Technologies, December 2007. initiatives/GENI/GENISysOvrvw1.1.pdf.

[109] O. Ormond, J. Murphy, and G.M. Muntean. Utility-based Intelligent Network Selection in Beyond 3G Systems. In IEEE International Conference on Communications, 2006. ICC'06, volume 4, 2006.

[110] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose. Modeling TCP throughput: a simple model and its empirical validation. Proceedings of the ACM SIGCOMM'98 conference on Applications, technologies, architectures, and protocols for computer communication, pages 303–314, 1998.

[111] E. Palkopoulou, D.A. Schupke, and T. Bauschert. Capex and availability tradeoffs of homing architectures in multi-layer networks. pages 70–77, 2009.

[112] C. Perkins. Rfc 3344: Ip mobility support for ipv4, August 2002.

[113] Planetlab. an open platorm for developing, deploying, and accessing planetary-scale services.

[114] Calton Pu and Douglas Blough. Reflective self-regenerative systems architecture study. Technical report, Georgia Institute of Technology, July 2006.

[115] L.R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. Readings in speech recognition, 53(3):267–296, 1990.

[116] V. Rastogi, V.J. Ribeiro, and A.D. Nayar. Measurements in OLPC mesh networks. In Proceedings of the 7th international conference on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks, pages 577–582. IEEE Press, 2009.

[117] I. Rish, M. Brodie, Sheng Ma, N. Odintsova, A. Beygelzimer, G. Grabarnik, and K. Hernandez. Adaptive diagnosis in distributed systems. Neural Networks, IEEE Transactions on, 16(5):1088–1109, September 2005.

[118] Christopher Rose. Derivation of pascal distribution, November 2007.

[119] K. Salamatian and S. Vaton. Hidden markov modeling for network communication channels. In Proceedings of the 2001 ACM SIGMETRICS international conference on Measurement and modeling of computer systems, pages 92–101. ACM, 2001.

[120] William H. Sanders. Möbius Manual. The PERFORM group, University of Illinois at Urbana-Champaign, May 2010. Version 2.3.1.

[121] J. Song, S. Han, A.K. Mok, D. Chen, M. Lucas, M. Nixon, and W. Pratt. Wirelesshart: Applying wireless technology in real-time industrial process control. IEEE RTASŠ08, pages 377–386, 2008.

[122] Q. Song and A. Jamalipour. A network selection mechanism for next generation networks. In 2005 IEEE International Conference on Communications, 2005. ICC 2005, volume 2, 2005.

[123] M. Steinder and A.S. Sethi. A survey of fault localization techniques in computer networks* 1. Science of Computer Programming, 53(2):165–194, 2004.

[124] M. Steinder and A.S. Sethi. Probabilistic fault localization in communication systems using belief networks. IEEE/ACM Transactions on Networking (TON), 12(5):809–822, 2004.

[125] M. Steinder and A.S. Sethi. Multidomain diagnosis of end-to-end service failures in hierarchically routed networks. IEEE Transactions on Parallel and Distributed Systems, pages 379–392, 2007.

[126] M. Stemm and R.H. Katz. Vertical handoffs in wireless overlay networks. Mobile Networks and Applications, 3(4):335–350, 1998.

[127] Roy Sterritt. Autonomic computing. Innovations in Systems and Software Engineering, 1(1):79, 2005.

[128] E. Stevens-Navarro, Y. Lin, and VWS Wong. An MDP-based Vertical Handoff Decision Algorithm for Heterogeneous Wireless Networks. IEEE Transactions on Vehicular Technology, 57(2):1243–1254, 2008.

[129] E. Stevens-Navarro and V.W.S. Wong. Comparison between vertical handoff decision algorithms for heterogeneous wireless networks. In Vehicular Technology Conference, volume 2, pages 947–951, 2006.

[130] R. Stewart. Stream control transmission protocol, September 2007. RFC: 4960.

[131] J. Strauss, D. Katabi, and F. Kaashoek. A measurement study of available bandwidth estimation tools. In Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement, pages 39–44. ACM, 2003.

[132] C. Sun, E. Stevens-Navarro, and VWS Wong. A Constrained MDP-Based Vertical Handoff Decision Algorithm for 4G Wireless Networks. In IEEE International Conference on Communications, 2008. ICC'08, pages 2169–2174, 2008.

[133] M. Suojanen, S. Andreassen, and KG Olesen. A method for diagnosing multiple diseases in MUNIN. Biomedical Engineering, IEEE Transactions on, 48(5):522–532, 2001.

[134] M. Sylor and L. Meng. Using Time Over Threshold to Reduce Noise in Performance and Fault Management Systems. Services Management in Intelligent Networks, pages 145–156, 2000.

[135] Andrew S. Tanenbaum. Computer Networks, 4. ed., pages 292–302. Pearson Education, Inc., 4 edition, 2003.

[136] Giuseppe Valetto and Gail Kaiser. Using process technology to control and coordinate software adaptation. Software Engineering, International Conference on, 0:262, 2003.

[137] H. Wang and S. Daley. Actuator fault diagnosis: an adaptive observer-based technique. Automatic Control, IEEE Transactions on, 41(7):1073–1078, Jul 1996.

[138] Y. Wang, C. Huang, J. Li, and K. Ross. Queen: Estimating Packet Loss Rate between Arbitrary Internet Hosts. Passive and Active Network Measurement, pages 57–66, 2009.

[139] H. Wietgrefe. Investigation and practical assessment of alarm correlation methods for the use in GSM access networks. In 2002 IEEE/IFIP Network Operations and Management Symposium, 2002. NOMS 2002, pages 391–403, 2002.

[140] G. Yin and Q. Zhang. Discrete-time Markov chains: two-time-scale methods and applications. Springer, 2005.

[141] J. Ying, T. Kirubarajan, KR Pattipati, and A. Patterson-Hine. A hidden Markov model-based algorithm for fault diagnosis with partial and imperfect tests. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, 30(4):463–473, 2000.

[142] Jie Ying, T. Kirubarajan, K.R. Pattipati, and A. Patterson-Hine. A hidden markov model-based algorithm for fault diagnosis with partial and imperfect tests. Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on, 30(4):463 – 473, nov 2000.

[143] AH Zahran and B. Liang. Performance evaluation framework for vertical handoff algorithms in heterogeneous networks. In 2005 IEEE International Conference on Communications, volume 1, 2005.