ARCHITECTURES AND ALGORITHMS FOR COGNITIVE NETWORKS ENABLED BY QUALITATIVE MODELS

Balamuralidhar, P.

Publication date:
2013

Document Version
Early version, also known as pre-print

Link to publication from Aalborg University

Citation for published version (APA):
Balamuralidhar, P. (2013). *ARCHITECTURES AND ALGORITHMS FOR COGNITIVE NETWORKS ENABLED BY QUALITATIVE MODELS*.

ARCHITECTURES AND ALGORITHMS FOR COGNITIVE NETWORKS ENABLED BY QUALITATIVE MODELS

**A DISSERTATION SUBMITTED
TO THE DEPARTMENT OF
ELECTRONIC SYSTEMS
OF
AALBORG UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY**

Balamuralidhar P
Center for TeleInFrastruktur Denmark, Tata Consultancy Services Ltd India

Abstract

Complexity of communication networks is ever increasing and getting complicated by their heterogeneity and dynamism. Traditional techniques are facing challenges in network performance management. Cognitive networking is an emerging paradigm to make networks more intelligent, thereby overcoming traditional limitations and potentially achieving better performance. The vision is that, networks should be able to monitor themselves, reason upon changes in self and environment, act towards the achievement of specific goals and learn from experience.

The concept of a Cognitive Engine (CE) supporting cognitive functions, as part of network elements, enabling above said autonomic capabilities is gathering attention. Awareness of the *self* and the *world* is an important aspect of the cognitive engine to be autonomic. This is achieved through embedding their models in the engine, but the complexity and achievable truthfulness of such models are of concern considering the dynamic and non-linear behavior of the network. Moreover the knowledge model should be able to capture and represent the holistic aspect of the network in a scalable manner.

In the present work, I focus on the architectural aspects of the cognitive engine that incorporates a context space based information structure to its knowledge model. I propose a set of guiding principles behind a cognitive system to be autonomic and use them with additional requirements to build a detailed architecture for the cognitive engine. I define a context space structure integrating various information structures that are required for the knowledge model. Use graphical models towards representing and reasoning about context space is a direction followed here. Specifically I analyze the framework of qualitative models for their suitability to represent the dynamic behavior of the wireless network. The motivation behind this novel approach is in the possibility of building the knowledge model from the qualitative information in the form of influence diagrams elicited from human experts. Considering the difficulties of building large scale models through structure learning, the above approach is attractive. After a detailed analysis of the qualitative model I select a set of fitting semi-qualitative extensions with inference mechanisms to overcome their observed limitations. With learning from this exercise I propose a methodology for preparing and using the qualitative models in a cognitive engine.

Further I use the methodology in multiple functional scenarios of cognitive networks including self-optimization and self- monitoring. In the case of self-optimization, I integrate principles from monotonicity analysis to evaluate and enhance qualitative models as part of the methodology. Related to self-monitoring, the proposal is on an architecture for network monitoring and fault diagnostics using qualitative models.

Towards the end I propose a novel cognitive acoustic communication network for short range data communication between devices. I present the design and implementation details along with its interesting applications for near field communications. Further I compare the qualitative models of the acoustic network with an equivalent radio network. The comparison results points to the generality of qualitative models across multi-technology systems.

Major contributions of this research work is in discovering the applicability of qualitative knowledge models and developing mechanisms to efficiently use them in cognitive networks.

Dansk Resume

Kompleksiteten af kommunikationsnetværk er stadigt stigende, og bliver yderligere kompliceret af deres indbyrdes forskelle og dynamik. Traditionelle teknikker udfordres i ”network performance management”. ”Cognitiv networking” er et spirende paradigme for at gøre netværk mere intelligente og således overvinde de traditionelle begrænsninger og potentielt opnå en bedre ydeevne. Visionen er, at netværk er i stand til at overvåge sig selv, på baggrund af ændringer i netværket og miljø, for derigennem at opfylde specifikke mål og lære af erfaringerne.

Begrebet Cognitiv Engine (CE) som understøtter kognitive funktioner, som en del af netværkselementer, så overnævnte autonome elementer får opmærksomhed. Bevidsthed om netværket og omgivelserne er et vigtigt aspekt for at få den ”cognitive engine” til at være autonom. Dette opnås ved indlejring af deres modeller i enginen, men kompleksiteten og værdien af sådanne modeller er problemet, når dynamisk og ulineær opførsel af nettet tages i betragtning. Desuden skal modellen være i stand til at fastholde og repræsentere det holistiske aspekt af netværket på en skalérbar måde.

I det foreliggende arbejde fokuserer vi på de arkitektoniske aspekter af den cognitive engine, der inkorporerer en 3D baseret information struktur til sin videnmodel. Vi foreslår et sæt vejledende principper til et kognitiv system, der skal være autonom og bruges med yderligere information for at opbygge en detaljeret arkitektur for den cognitive engine. Vi definerer en kontekst baseret struktur, der integrerer de forskellige informationsstrukturer, som er nødvendige for videnmodellen. Der bruges grafiske modeller til at repræsentere og ræsonnere om de rummelige sammenhænge vi følger. Konkret analyserer vi inden for rammerne af kvalitative modeller for deres egnethed til at repræsentere den dynamiske adfærd af det trådløse netværk. Motivationen bag denne nye fremgangsmåde er i muligheden for at opbygge en viden model fra kvalitative oplysninger i form af indflydelse diagrammer genereret af humane eksperter. I betragtning af de vanskeligheder, der er ved at bygge store modeller gennem struktur læring, er den ovennævnte fremgangsmåde attraktiv. Efter en detaljeret analyse af den kvalitative model, vælger vi at tilpasse en semi-kvalitativ tillæg med logiske beslutninger for at overvinde de observerede begrænsninger. På baggrund af denne øvelse foreslår vi en metode til fremstilling og anvendelse af de kvalitative modeller i en cognitiv engine.

Yderligere bruger vi metoden i multiple kognitive netværk funktions scenarier, herunder selv-optimering og selvovervågning. I tilfælde af selv-optimering, integrerer vi principperne fra analysen til at evaluere og forbedre den kvalitative model som en del af metoden. Relateret til selv-monitorering, peger forslaget i retning af en arkitektur for overvågning af netværk og fejldiagnoser.

Mod slutningen foreslår vi en ny kognitiv akustisk kommunikations netværk for kortdistance kommunikation mellem enheder. Vi præsenterer design og implementerings detaljer sammen med anvendelsesmuligheder. Yderligere, sammenligner vi de kvalitative modeller af det akustiske netværk med et tilsvarende radionetværk. Sammenligningen resulterer i punkter til den generelle anvendelse af kvalitative modeller på tværs af multi-teknologi systemer.

Hovedresultater af denne forskning er anvendelsen af kvalitative videnmodeller og udvikling af mekanismer til effektivt at bruge dem på kognitive netværk.

Acknowledgment

My inspiration to work in area of cognitive radios and networking has been primarily due to the opportunity to participate in two major EU FP6 Consortium Research projects My Adaptive Global NET (MAGNET) and End to End Reconfigurability (E2R). My interaction with Prof. Ramjee Prasad initiated during this time and since then he has been a prime motivator, mentor and guide for me to pursue this research. I express my sincere gratitude to Prof. Ramjee Prasad. In addition to this there are several researchers, scientists and professors, as part of the above EU projects, with whom I have interacted and developed myself to conduct this research. I thank them all.

Theoretical foundations required to build the cognitive framework as part of this research has been mentored by Late Mr. P.S.Subramanian and his memories will be with me always.

My sincere thanks to my organization “Tata Consultancy Services Limited” where I could do this research work in an encouraging R&D eco-system of TCS-Innovation Labs. Several of my colleagues in this lab have supported at various stages of this research work. Special thanks to Dr.Hemant Rath, Dr.Rajan M.A, Dr. Rahul Sinha and Dr.Hrishikesh Sharma through discussions and other supporting actions. Encouragement from Mr.Ananth Krishnan, CTO of TCS, has been a key enabler.

Many thanks are due to Prof.Albena Mihovska and Prof.Rasmus Hjorth Nielsen for spending their valuable time for giving the feedback and suggestions.

Last but not the least; I thank my family for their full hearted support and understanding for enabling me to complete this research work.

Table of Contents

List of Figures

List of Tables

List of Acronyms

AWGN	Additive White Gaussian Noise
BER	Bit Error Rate
BN	Bayesian Networks
CE	Cognitive Engine
CN	Cognitive Network
CR	Cognitive Radios
CRN	Cognitive Radio Network
CAPRI	Common Application Requirement Interface
CBL	Case Based Learning
D2D	Device To Device
DARPA	Defense Advanced Research Projects Agency
DAML	DARPA Agent Mark Up Language
DL	Description Logic
DUPACKS	Duplicate Acknowledgements
GENI	Generic Network Interface
JOCP	Joint Optimal Congestion Control And Power Control
KP	Knowledge Plane
KPI	Key Perfomance Indicator
LTI	Linear Time Invariant
MIB	Management Information Base
MIMO	Multiple Input Multiple Output Techniques
MSDAG	Monotonic Signed Directed Acyclic Graph
OWL	Ontology Web Language
PER	Packet Error Rate
PHY	Physical Layer
PSD	Power Spectral Density
PU	Primary User
QoE	Quality Of Experience
QoS	Quality Of Service
QPN	Qualitative Probabilistic Networks
RAT	Radio Access Technology
RCM	Regulatory Compliance Monitor
RDF	Resource Description Framework
RKRL	Radio Knowledge Representation Language
RTT	Round Trip Time
SA	Simulated Annealing
SDAG	Signed Directed Acyclic Graph
SDR	Software Defined Radio
SINR	Signal to Interference and Noise Ratio
SNR	Signal to Noise Ratio
SON	Self Organizing Networks
UAB	Upper Audio Band
ULLA	Universal Link Layer API
XML	Extensible Markup Language

1 Introduction

1.1 Research Problem

Knowledge modeling and management is observed to be a key building block for self-management in Cognitive Networks with an end-to-end perspective. The scientific objective of this research is to discover ways though which novel knowledge management can bring-in context awareness and enable autonomic capabilities in networks with an enhanced efficiency. To fulfill this objective, this research work proposes novel architectures and algorithms for a cognitive engine that encapsulates a knowledge model used by the cognitive process that is frugal yet good enough to be implemented on network elements to enable them to be integrated as a cognitive network. Applicability of Qualitative Models is of specific focus.

Several outcomes of this research work have been published in various journals and conference proceedings. A list of 12 publications, where the contributions appeared, is included at the end of this chapter. This is followed by a list of patents filed on some of the novel aspects of the work.

1.2 Motivation

Wireless networks are ubiquitous and of great economic importance to modern connected society. It is also the source of many of the fundamental scientific challenges faced by modern communication networks. A key feature of modern networks is their scale and complexity. It far exceeds the ability of users and operators to optimally configure them.

Future application over wireless networks will require an ecosystem consisting of numerous coexisting Radio Access Technologies (RATs). The multi-RAT networks will provide user-centric communications catering a multitude of services to end-users, including machines, with seamless mobility, application and session management and higher levels of Quality of Experience (QoE). Emerging applications involving Machine to Machine communications and Internet of Things poised to bring in a multi-fold increase in number of communicating nodes into future networks. This along with introducing more spectrum-efficient technologies makes the networks more and more complex, and therefore, more difficult to monitor, control, configure and manage. This poses a challenge to wireless network operators in the task of running their networks while introducing new services and achieving goals in terms of customer satisfaction, benefit, market share, innovation, reputation etc.

Minimization of human intervention in wireless network management is one direction of exploration that is considered in Self Organizing Networks (SONs) [1]. It addresses a family of functionalities used in operating a network in a highly autonomous manner, encompassing self-configuration, self-optimization and self-healing [2][3]. Ericsson estimates the adoption of SON features in 4G Networks has resulted in 40% faster roll outs, and as much as 90% reduction in daily maintenance of new LTE networks (Figure 1.1) [3].

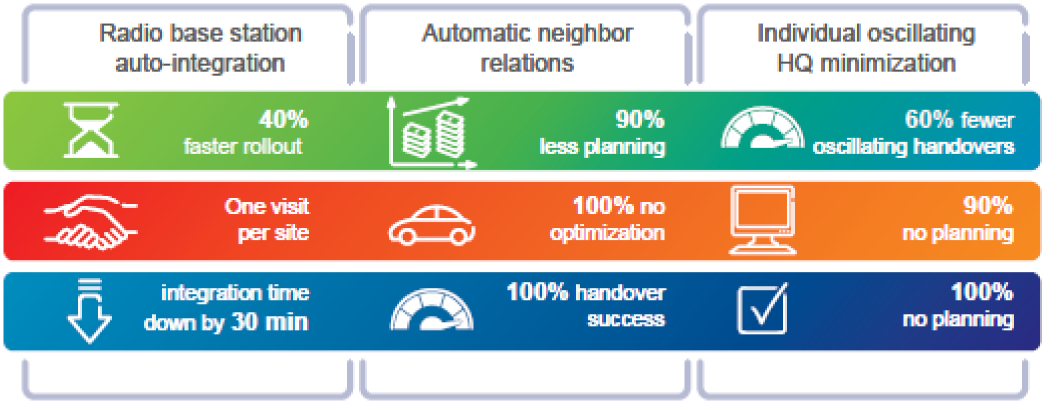

Radio base station auto-integration
Automatic neighbor relations
Individual oscillating HQ minimization
40% faster rollout
90% less planning
60% fewer oscillating handovers
One visit per site
100% no optimization
90% no planning
integration time down by 30 min
100% handover success
100% no planning

Figure 1.1 Measurable benefits from SON features on Live Networks (source: Ericsson).

functionality needs to work across all radio-access, transport and core networks in such a way that the differences between these technologies are masked for higher level operations [Figure 1.2]. This will enable the operators to meet end-to-end service Key Performance Indicators (KPIs).

Cognitive Radio applies the idea of self-organizing to air-interface and generally aims at enabling spectrum sharing. Cognitive Network is extending the self-organization techniques used for radio access subsystem to include entire network with knowledge representation, automated reasoning and learning. This encompasses adaptive mapping of user's requirements, preferences, context and situation onto offered services considering the service provider's resource assignment and other policies. This requires techniques for real-time monitoring and control of situation and context of network and its resources. Unlike traditional network management that is generally statically configured, the paradigm shift with cognitive networks is continuous and dynamic configuration and optimization.

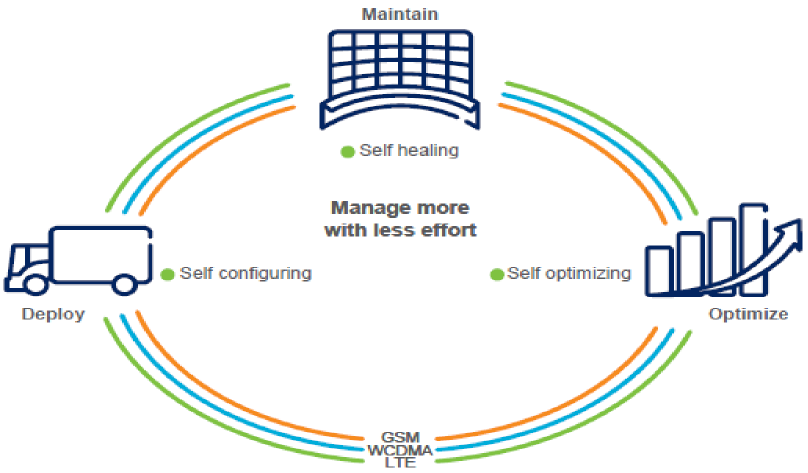


Figure 1.2 SON functionality across all technologies (Source: Ericsson)

In-network management is an approach that supports management operations by the means of a highly distributed architecture. Here the management functions are located in or close to the managed network and service elements. In most of the cases (such as adhoc networks) it is co-located on the same nodes. The distributed network architecture promotes support for self-management features with a suitable trade-off between computing and communication required for management. Many of the self-organization related challenges are addressed in the emerging paradigm of Cognitive Networks.

Knowledge modeling and management is observed as a key building block for self-organization with an end-to-end perspective. From the operators' perspective, a service- and business-driven management needs to be transparent and independent of the underlying network infrastructure, domains, and resources. The network complexity should be hidden from the operators' perspective. A unified knowledge modeling framework that can have an end-to-end scope is still an open problem for research.

.

1.2.1 The emerging paradigm of Cognitive Networks

Much of the cognitive network concepts have been drawn or evolved from that of cognitive radios (CR). An initial definition of cognitive networks has been given by Thomas et al. [5]. In line with the definition of Cognitive Radios, "*A cognitive network is defined as a network with a cognitive process that can perceive current network conditions, and then plan, decide, and act on those conditions. The network can learn from these adaptations and use them to make future decisions, all while taking into account end-to-end goals"*. It seems to have motivated by the Knowledge Plane described by Clark [6] as a "distributed cognitive system that permeates the network".

Though the CN concept draws a lot from the CR, CR need not be a necessary part of cognitive network. A network of cognitive radios is termed as Cognitive Radio Networks (CRN). This view on CN can be seen as CR extended up the stack and across the network. Fig 1.3 illustrates the difference between cognitive radio (CR), cognitive network (CN) and cognitive radio network (CRN). While the scope of CR is the wireless link, CRN spans the network of CRs. CN is the most general concept spanning over the entire communication system, including the core network.

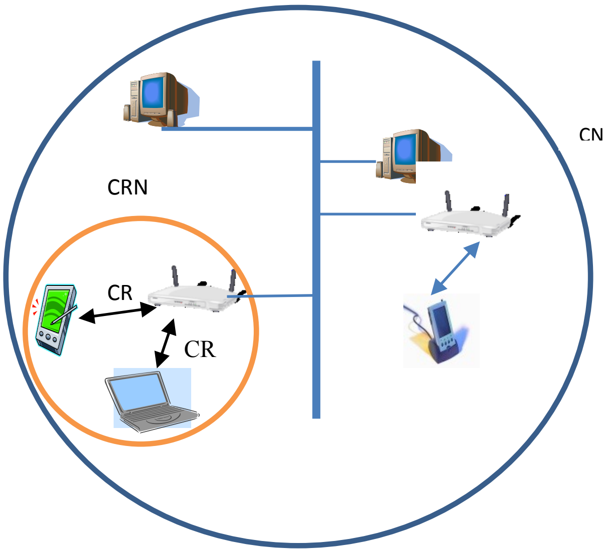
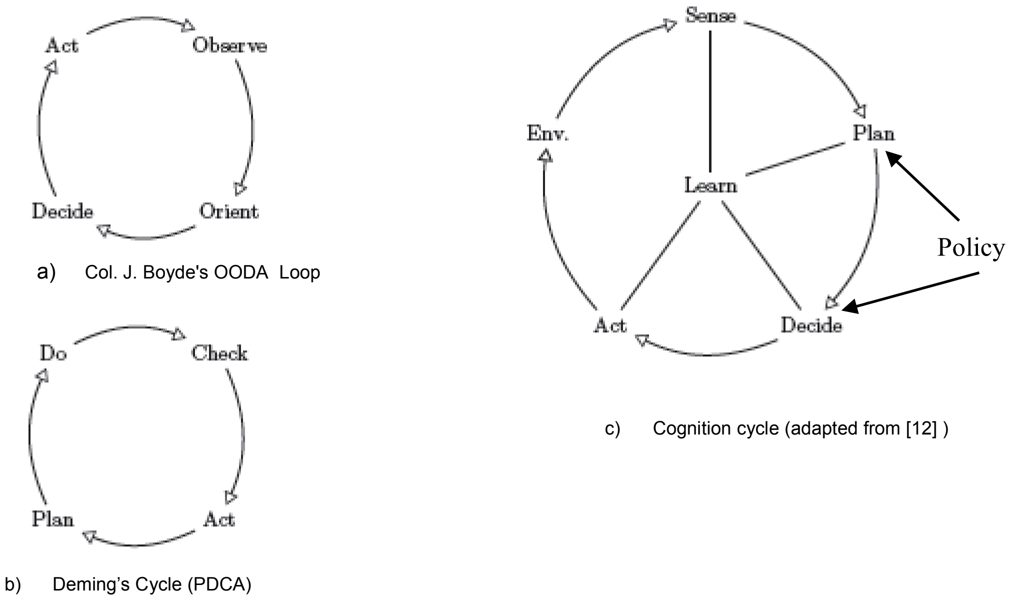


Figure 1.3 Scope of Cognitive Radio, Cognitive Radio Network, Cognitive Network (Adapted from [11])

Looking at some of the recent works related to Cognitive Networks, substantial explorations have been performed by End-to-End Reconfigurability Project E2R II [7], m@ANGEL platform[8], CTVR at Trinity College[9] and the Institute for Wireless Networks at RWTH Aachen University[10]. Architectures at various degrees of maturity for end-to-end oriented, autonomous networks have been proposed as part of the above works. Most of them emphasize the need to have a holistic picture of the network rather than limited local scope. Beyond the high level architectural details, practical considerations for implementing them in real-world devices is still required. A unified theory and set of principles underlying the architectures is desired and should be explicitly identified so that future extensions and enhancements can be consistent and scalable.

In general a cognitive framework incorporates two key components namely Knowledge Management and a cognition loop (Cognitive Process). Knowledge management involves the acquisition, representation and refinement of the knowledge about the network and environment enabled by learning. This brings context awareness capability to the network. The orchestration of the tasks involved in the autonomic network management is done though a cognitive process.



a) Col. J. Boyde's OODA Loop

b) Deming’s Cycle (PDCA)

c) Cognition cycle (adapted from [12])

Figure 1.4 Cognitive cycles as it is evolved

There are different representations of cognitive loop proposed in the literature, but invariably they convey similar ideas. Col. Boyde's OODA (Observe-Orient-Decide-Act) loop in Figure 1.4a, was devised to understand opponents' moves [11]. Deming's cycle (Plan-Do-Check-Act) was formulated for quality improvement (Figure 1.4b). Figure 1.4c incorporates learning as an explicit task present along with other tasks in the cycle and it can be said as more complete depiction of the cognitive cycle. The emerging view is that, the self-aware network will sense the environment (Sense) and the observations captured by the sensors will be further used for planning (Plan) to bring up various solution options. The planning module determines potential actions, i.e. strategies to be followed based on observations and policies stored in the policy module (Policy). The decision module decides on the actions to be taken based on possible moves from planning stage. Learning capability is invoked for each of these modules to learn and remember about respective input/output behavior, so that the experience gathered could be used for improving the performance. Finally, the actuators (Act) are responsible with implementing chosen changes (reconfigurations) in the system.

Many times all the task elements of the cognitive cycles are not co-located. Sensing happens at one or multiple devices and the actuation happens at another. Major aspect that is constraining cognitive cycle is this communication capability. One of the gaps observed in the above discussed cognitive cycles is the absence of a communication capability identified that can achieve synchronization between multiple cognitive cycles distributed over different network elements.

1.3 Challenges Considered

There are several fundamental challenges for a cognitive framework for autonomic networking. The focus of this research is to build architecture and algorithms for cognitive networks with a knowledge modeling mechanism that incorporates following capabilities.

Managing Uncertainty:

Achieving survivable communication services providing acceptable end-to-end quality of service (QoS) is challenging in presence of uncertainty. Some of the uncertainties in wireless networks are perceived mostly at the physical communication layer - due to rapidly varying wireless link qualities, or variable points of network attachment for mobile users. Further application characteristics, users behavior, mobility etc are some of the additional sources of uncertainty with networks. An autonomic management entity should take care of the network management in a robust manner at a reasonable computational complexity.

Context-Awareness:

A context is any relevant attribute of a device that provides information about its interaction with other devices and/or its surrounding environment at any instant of time. Knowledge about the contexts of the participating devices as well as the context of the environment is required to effectively manage the uncertainty. A sequence of device/entity contexts with the underlying interpretation (semantics) defines a situation. Imparting situation awareness through efficient context awareness is a challenge.

Autonomic Resource Management:

Flexible and seamless configuration and delivery of services in large, autonomous, and complex evolving networks is another challenge. How to efficiently and dynamically manage the spectrum (bands) in cognitive radios? How to improve the QoS by preventing the over-provisioning of scarce resources (bandwidth, power) and using learning algorithms to profile and anticipate future resource usage, so that the resulting system performance is optimal or near optimal? And there are many.

Considering the above challenges we explore architectures and algorithms that impart robustness in terms of self-configuration and self-optimization for cognitive networks with a low complexity cognitive engine, in the absence of a high resolution model. Some of the questions we are addressing as part of this research are: How the cognitive engine can be driven with a qualitative or semi-qualitative model that can be developed by an expert initially and the engine can refine it through learning from the experience? What are the suitable architectures that can support it? Further how the cognitive engine operating at a node level can integrate at a network level to achieve network goals?

Further some key questions to answer through this research that can address objectives stated above are:

- How effective is Qualitative graphic models to represent the dynamic behavior of cognitive radio and wireless networks? What are the issues, challenges and possible mitigations?
- How Qualitative graphic models can be effective for the analysis and inference towards solving cross layer problems in wireless networks? What are the enhancements possible to address the correctness of inference, achieve tradeoffs, without the loss of efficiency? Can the combination of monotonicity analysis help in resolving the trade-off? How to handle influences that are not monotonic and can affect the inferences?
- How a Qualitative model based cognitive engine performs for a self-optimizing radio or a network? How QPN based analysis can help in the cognitive controller design?
- Can the model analysis bring out some indicators/guidelines on the design of underlying radio or network so that it can be fed back into the system re-design?

- How can the graph based model be used by the radio for fault detection and self-diagnosis through qualitative inference?

1.4 Contributions and Novelties

The specific research problem being addressed in this thesis is in conceptualizing and enhancing the architecture of cognitive engine (CE) with a novel approach for knowledge representation and management mechanism applicable to cognitive networks. Key requirements for the cognition process and its components have been formulated from the basic principles drawn from cognitive science and developed a detailed architecture and framework of a Cognitive Engine (CE). We proposed the novel concept of using Qualitative modeling with influence graphs to represent the dynamic behavior of cognitive radios and networks. It can evolve to a more quantitative representation through learning while in operation. The specific modeling formalism chosen is based on Qualitative Probabilistic Networks (QPN). Suitable techniques have been incorporated to deal with ambiguities while QPN inference and a methodology to construct, configure and integrate QPN models for wireless networks in the CE have been proposed.

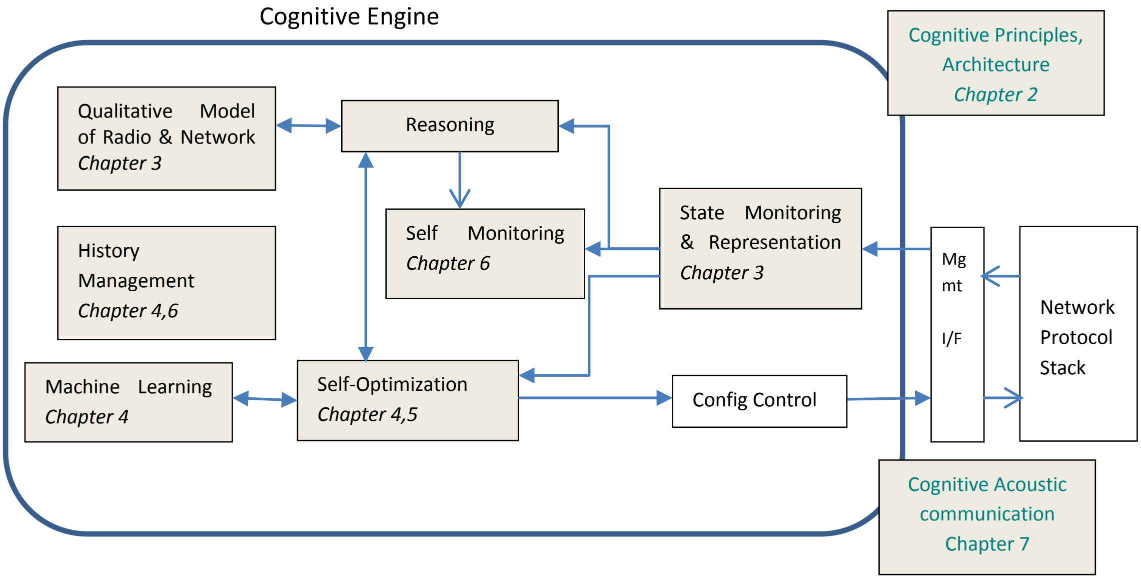

Cognitive Engine
Qualitative Model of Radio & Network Chapter 3
Reasoning
History Management Chapter 4,6
Self Monitoring Chapter 6
State Monitoring & Representation Chapter 3
Machine Learning Chapter 4
Self-Optimization Chapter 4,5
Config Control
Cognitive Principles, Architecture Chapter 2
Mg mt I/F
Network Protocol Stack
Cognitive Acoustic communication Chapter 7

Figure 1-5 Highlighting Areas of Contributions

In addition to this new approach and methodology of using QPN based qualitative models for online dynamic configuration and optimization in a CE, an optimization algorithm encapsulated in an optimization core has been developed. It adopts QPN Model based inference for a sequential optimization approach with innovative techniques for overcoming the problem of ambiguous inference results from conflicting influences. A heuristic method is introduced to determine the sequence of activation of the controls in the optimization algorithm. This is based on their domain property categorization of the state variables into resource, capacity and consumer variables.

Integrating monotonicity analysis with QPN to analyze the fitment of the model to be used in optimization loop is another novel contribution. With this the CE can check if the model is sufficiently constrained and select the right active constraints to consider while optimization in progress.

A Case Based Learning algorithm is incorporated to avoid anomalous configurations that are catastrophic for run time optimization process. A unique compact representation of the situation of the anomalous configuration enabled by the QPN graph structure minimized the size of the database required to support this.

Extending the usage of QPN models to identify network failures towards contributing to self-monitoring capability in CE is another contribution. It includes a proposed architecture and methodology for fault detection and diagnosis based on QPN models.

Finally a novel concept of cognitive acoustic data communication for short range device to device (D2D) communication is introduced. A link design and implementation results are presented. The QPN model of the behavior of this access technology has been observed to be similar to that of a cognitive radio link and we propose that the QPN models can be general enough and technology agonistic.

The contributions topics with respective chapter mapping are depicted in Figure 1.5

Some of the novel contributions which had immediate innovation potential have been filed as patents (Section 1.7)

1.5 Outline of the Thesis

The thesis is a monograph presenting the novel contributions and results on investigations in the area of cognitive networks and is organized as follows.

Chapter 2 analyses the guiding principles behind a cognitive system and defines the requirements of a cognitive engine for imparting autonomic features to cognitive wireless communications. Based on the requirements a novel architectural proposal for the cognitive engine is proposed and that forms the reference for the works in remaining chapters. This Chapter also proposes how to address the need for efficient mechanisms for representing and managing the knowledge of the 'self' and the 'world'.

Chapter 3. Qualitative Probabilistic Networks (QPN) is proposed as a novel approach for modeling the dynamic behavior of cognitive networks and the easy capture of qualitative knowledge from human experts to build the Knowledgebase is the major motivation. The Chapter proposes and designs (or solves) an optimization algorithm based on QPN modeling and reasoning. The use of semi-qualitative extensions in resolving trade-offs and thus enhancing the efficiency of the optimization are analyzed, and based on this analysis, a methodology for incorporating them in the model is proposed. The approach is evaluated through simulation of a cognitive link optimization scenario.

Chapter 4 proposes to extend the use of the QPN Model of wireless networks for driving self-optimization within the framework of the cognitive engine. It uses a sequential optimization algorithm with the novelty of QPN based inference with a heuristic selection of order of the control variables. Further, it demonstrates its application to a cognitive radio link adaptation and throughput optimization. Combining the monotonicity analysis with the QPN based methodology is another aspect of novelty in this chapter.

Chapter 5 develops and analyzes QPN models for capturing behaviors at the network layer and above. The QPN methodology proposed in Chapter 4 is applied here also to prepare the model suitable for inference. To motivate the analysis we focus on a practical problem of TCP congestion management in an adhoc wireless network scenario. We show that a joint congestion management involving TCP and wireless link adaptation strategy can be inferred from the QPN model and the cognitive engine can operationalize it. This follows a simulation study to support and verify the observations.

Chapter 6 proposes a novel scheme for using the QPN model for self-monitoring. The objective here is to identify network failures and provide clues in diagnosing the faults. An algorithm for fault detection is developed and illustrated using an antenna failure scenario in a radio link. Further the monitoring architecture is extended to cover regulatory compliance monitoring for cognitive radios.

Chapter 7 proposes a novel communication scheme using upper audio frequency band for short range communication between devices/appliances over the air. We refer to this scheme as Upper Audio Band (UAB) Communication. Cognitive Radio concepts have been used here to operate this communication modem as a secondary in presence of primary users such as TV, Home Entertainment Systems. We present the design and implementation details of the cognitive acoustic modem and its performance details. Further, the idea of using this in a personal area cognitive network is proposed and analyzed. The QPN model for this radio link is also similar to the one that for the radio from chapter 3, and it points to the use of generality of QPN models across communication technologies.

Chapter 8 Concludes with the research findings and suggests several open problems for future research.

1.6 List of Publications from this Research Work

A. Balamuralidhar P, Ramjee Prasad, "Self-Configuration and Optimization for Cognitive Networked Devices", Springer Journal on Wireless Personal Communications, 2011, DOI: 10.1007/s11277-011-0240-8 *(Chapter 2,3 & 4)*

B. Balamuralidhar P, Rajan M.A, "Signed Graph based Approach for On-line Optimization in Cognitive Networks", Proceedings, COMSNETS-2011, Third International conference on Communication Systems and Networks, 4-8 January 2011, Bangalore India *(Chapter 3)*

C. Balamuralidhar P, "Exploring Qualitative Probabilistic Networks for Knowledge Modeling in Cognitive Wireless Networks", IWCMC-2013 *(Chapter 3, 4)*

D. Hemant K Rath, Rajan MA, Balamuralidhar, "Monotonic Signed Graph Approach for Cross-layer Congestion Control in Wireless Ad-hoc Networks", GlobeCom 2011. *(Chapter 5)*

E. Balamuralidhar P, Ramjee Prasad, "A context driven architecture for cognitive radio nodes", International Journal of Wireless Personal Communications (Springer), Vol. 45 (3), May 2008, pp 423-434 *(Chapter 2)*

F. Rahul Sinha, P. Balamuralidhar, Rajeev Bhujade, "An Upper Audio Band based Low Data Rate Communication Modem", Accepted in ICSPCS-2012 *(Chapter 7)*

G. Rahul Sinha, P. Balamuralidhar, Rajeev Bhujade, "Software defined radio based on the upper audio band for low data rate communications over short distances", SDR12-WinnComm, 2012, Accepted. *(Chapter 7)*

H. Balamuralidhar P, Ramjee Prasad, "A Programming Paradigm for Cognitive Radios", International Conference on Wireless Personal Multimedia Systems (WPMC07), Jaipur, Dec 4-6, 2007. *(Chapter 2)*

I. P.S.Subramanian, Balamuralidhar P, "Intensional description of autonomic use cases", ", ICT '06, 11-12 May 2006, Funchal, Portugal *(Chapter 2)*

J. Balamuralidhar P, Ramjee Prasad, " A Radio Compliance Monitor Architecture for Enabling Regulatory Certification for Cognitive Radios" , WPMC06 , San Diego, U.S.A. from September 17 to 20, 2006 *(Chapter 6)*

K. Hrishikesh Sharma, Balamuralidhar P, "A Context Interpretation framework for Cognitive Network devices", International Conference on Software Defined Radios, SDRF, Denver, USA, Dec 2007 *(Appendix A)*

L. Hrishikesh Sharma, Balamuralidhar P, "Application of Semantic Web technologies for Context Interpretation in Cognitive Communication Devices", WWRF – 19 conference, 5-7 Nov 2007, Chennai, India *(Appendix A)*

1.7 List of Patents Filed

a) Balamuralidhar P, “Dynamic Self Configuration Engine for Cognitive Networks and Networked Devices”, 13/372,636, http://www.freepatentsonline.com/y2012/0209582.html

b) Rahul Sinha, Balamuralidhar P, Rajeev B, “Wireless Data Communication Over Acoustic Channel”, 3487/MUM/2011

References

[1] 3GPP TR 36.902 "Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Self-configuring and self-optimizing network (SON) use cases and solutions

[2] 3GPP TS 32. 541, “Telecommunication management, Self- Organizing Networks (SON), Self-healing concepts and requirements,” v.10.0.0, March 2011

[3] "NGMN Recommendation on SON and O& M Requirements", A requirement Specification by the MGMN Alliance, Dec. 2008

[4] “Smarter Self Organising Networks”, Ericsson Whitepaper , 284 23-3169, February-2012

[5] R. W. Thomas, L. A. DaSilva, and A. B. MacKenzie, Cognitive networks," in IEEE DySPAN, (Baltimore, Maryland), pp. 352-360, Nov. 2005.

[6] D. D. Clark, C. Partridge, J. C. Ramming, and J. T.Wroclawski, \A knowledge plane for the internet," in Conf. on Applications, Technologies, Architectures, and Protocolsfor Computer Communications, (Karlsruhe, Germany), pp. 3-10, ACM, 2003

[7] Bourse, D. and El-Khazen, K. (2005) End-to-end reconfigurability (E2R) research perspectives. IEICE Trans. Commun. (Special Section on Software Defined Radio Technology and Its Applications), 11 4148–4157.

[8] Demestichas, P., Stavroulaki, V., Boscovic, D., Lee, A. and Strassner, J. (2006) m@ANGEL: autonomic management platform for seamless cognitive connectivity to the mobile internet. IEEE Commun. Mag., 4, 118–127.

[9] P. D. Sutton, L. E. Doyle, and K. E. Nolan, A reconfigurable platform for cognitive networks," in ICST CROWNCOM, (Mykonos Island, Greece), June 2006.

[10] P. Mahonen, M. Petrova, J. Riihijarvi, and M. Wellens, Cognitive wireless networks: Your network just became a teenager," in IEEE INFOCOM, (Barcelona, Spain), Apr. 2006.

[11] B. S. Manoj, R. R. Rao, and M. Zorzi, “On the use of higher layer information for cognitive networking," in IEEE Global Telecommunications Conference (GLOBECOM), pp. 3568-3573, 2007.

[12] C. Fortuna and M. Mohorcic, Trends in the development of communication networks: Cognitive networks", Computer Networks., vol. 53, no. 9, pp. 1354-1376, 2009.

2

Cognitive Engine Architecture

In this chapter we propose basic guiding principles to be supported by a cognitive system and formulate the requirements of a cognitive engine for imparting autonomic features to cognitive wireless communications. Idea about a structured context space that incorporates various information structures needed by the cognitive process is a key aspect of the proposal. Based on the key requirements we develop a detailed architecture of a Cognitive Engine.

2.1 Cognitive Engine – Introduction

Cognitive Engine is conceptualized as an executable unit that can be embedded in network elements to provide them cognitive capability individually as well as collectively. The network element is a general term that includes various entities in a network such as radios, base stations, routers etc. Further we define cognitive nodes as a network element with an embedded cognitive engine. In this line we view cognitive network as a system driven by a network of cognitive engines.

The cognitive network research community has proposed several architectural designs for cognitive networks. In this section we review a few key developments towards drawing out a list of major architectural requirements of a cognitive engine in the context of cognitive networking.

2.1.1 Related works

Pioneering work by Clark et.al [1] proposed a cognitive-enhanced networking concept with a major focus on self-recovery from faults. They introduced Knowledge Plane (KP) as a key component of the cognitive system. It followed a closed loop distributed control system structure enabled by learning and reasoning capabilities. It maintains a high level model of the network. Architecturally KP is a vertical plane cutting across the protocol layers and spans horizontally across nodes. The KP should take advantage of the different observations that can be made in various points of the network, implement a unified approach to solve problems, avoid ad-hoc solutions, include network edges, in order to exploit their knowledge; be able to function in dynamical, continuously changing environments, also robustness against misleading and/or incomplete information, and under conflicting high-level goals. It is noted that these basic principles are shared by many of the researchers those followed research work related to cognitive networks.

Thomas et al.[2], introduced cognitive networks more formally and specified the architecture characteristics such as (i) extensibility and proactivity; (ii) the capability by the decision process to use network metrics as input and provide actions as output; (iii) the capability to achieve higher performance levels with respect to traditional networks. They define three layers for the cognitive entity namely behavioral, computational and neuro-physical. Objectives with an end-to-end scope is handled by the behavioral layer, the decomposed local objectives are in the scope of middle layer and implementation of actions are accomplished by bottom layer. A CN is formed of a set of software agents, which have certain reasoning capabilities. These software agents are connected in a network and interact with each other; they can cooperate and act. While functioning in this environment, the agents can learn and take decisions in order to reach an end-to-end goal. These end-to-end goals are dictated by the business and user's requirements.

The work from Trinity College of Dublin presents a framework for cognitive Network [3] that proposes a logical separation between network nodes and the cognitive engine running in the network. The cognitive engine performs learning, orientation, planning, and decision-making functions; the network node performs observation and reconfiguration. While it provides some flexibility, the logical separation of the two entities can potentially limit the benefits from local optimization and can increase signaling overhead.

The concept of a cognitive plane is introduced in CogNet [4], to use information from all layers and run joint-layer optimization algorithms that can be distributed throughout the network. In this proposal, each protocol layer is extended with so-called Intra-layer Cognition Modules for performing intra-layer monitoring, control, and coordination functions. Modules are interconnected through the Cognitive Bus, for the coordination of the cognition modules. While this reduces the complexity of managing cognitive processes, the performance is highly dependent on the coordination of intra-layer modules. Though a distributed operation is conceptualized here the interaction between multiple cognitive planes are not clearly brought out.

Mihailovic et al.[5] investigated two specific topics: i) the definition and formulation of the knowledge lifecycle in self-managed networks, ii) the building of situation awareness in such networks as a ubiquitous concept in dynamic control. In most of these proposals the cognitive entity can comprise all the layers of a node or even just a subset of them, whereas the reconfigurable entity involves reconfigurations not only inside a single node but possibly also in each node of the network. However in the case of multiple distributed cognitive entities, it is not clear how the multiple reconfigurations are coordinated to achieve a network wide goal. Recently the architecture proposed by Fonseca [6] has a hierarchical architecture consisting of three levels of cognitive functionality, namely layer level, node level and network level. While this architecture is good for hierarchically structured networks, it would not be efficient for adhoc networks.

2.1.2 Knowledge representation for cognitive Networks – a gap

One of the major requirements of using a cognitive approach for network wide adaptation is the use of sharable knowledge among network elements to enhance collaboration and cooperation. Most of the reported works choose the use of markup languages, such as the Extensible Markup Language (XML). There are some proponents for the use of DARPA Agent Markup Language (DAML) by the Defense Advanced Research Projects Agency (DARPA), that the features of both XML and the Resource Description Framework (RDF). to support ontologies for web objects and some cases ontology languages such as OWL. Some earlier proposals include Radio Knowledge Representation Language (RKRL), first promoted by Mitola [7] with the objective to represent radio knowledge through the use of structured, yet natural, language. Though these are typical representational mechanisms the real information are gathered from sources/structures such as Management Information Bases (MIBs)[8].

There are practical issues with the use of semantic web technologies to run on typical embedded radio platforms due to the limited resource availability of these platforms. One of our investigations conducted for the evaluation of ontology based context reasoning is presented in *Appendix A* of this thesis. It was concluded that, while XML based languages are recommended for the purpose of standardized information exchange, they may not be efficient for embedded applications since they are more verbose. One possibility of addressing this is by catching the most frequent decision patterns in the history, and using them for faster decisions. In addition to this there can be a hybrid approach towards more efficient and optimized data structures for context representation and interpretation internally, while they are used in XML based formats for standardized information exchange externally. No structured analysis of the information space handled by such representation mechanisms in the context of cognitive networks has been reported in the literature to the best of our knowledge.

2.2 Cognitive Architectures – Guiding Thoughts

As discussed earlier there is a gap in understanding the structural details required for efficient capture and manipulation of information in a cognitive engine. In this section we bring forth some key insights on this aspect and subsequently integrate them in extended cognitive engine architecture to be presented in a later section.

2.2.1 Cognitive Systems – Knowledge of the 'self 'and the 'world'

Here we propose the following fundamental requirements of an object to be *cognitive*.

Cognitive as a prefix applied to an object, which may have a spatio-temporal extension -i.e which may persist in time and may occupy non-trivial domain of space - should satisfy the following:

a) *Within the object a representation of the Ontology of both the internals of the object and the environment the object "lives in" should exist.*
b) *Mechanisms to maintain the global faithfulness (w.r.t temporal variation) and global consistency (w.r.t spatial variations) of these representations should exist.*

Here by ontology we mean the representation of reality or facts, by specifying various concepts and their inter-relationships used in the system. This explicit representation enables multiple systems to share the knowledge and interoperate with a common semantics. A cognitive object should incorporate ontology of the 'self' and the 'world' where it operates. It should also have mechanisms to keep this knowledge up-to-date and relate truthfully with the internal and external states of the object. In the context of a wireless network element the term 'self' means the facts about the internals of the device such as radio parameters, protocol stack configuration, applications, processor, memory, battery power etc. 'World' representation includes the information about the network, network objectives, state of the neighbor devices, communication environment etc.

Maintenance of these representations need not necessarily be completely local; in fact for non-trivial cognitive systems there will be spatio-temporal instances in which the sensed ontology and the internal ontology will be different and a conscious decision will be made not to update the internal ontology if due to prior learning the sensed ontology is known to be in a transient mode.

Indeed what is important in Cognitive systems is the *Persistent* and *Unitary* Ontology which can be called the *conscious knowledge* of the system and which constitutes the *"self"* of the system. The persistent nature of this ontology enables pro-activeness and robustness to "ignorable events" while the unitary nature enables end-to-end adaptations. Robustness of adaptive behavior is ensured by the minimization of the generalization error in learning due to the insistence of a persistent ontology.
Moreover, this feature – of having a core persistent and unitary ontology- is what will distinguish cognitive systems from merely adaptive systems which may also need ontology to function but will not have such a notion of *"self"*. When there is a system constituting many autonomous elements, then there is *"self"* for each of those elements and the *"world"* for each of them is the complement of respective *"self"*s. Further to keep the unitary nature at the system level the ontology should be *communicated between* all elements. This *communication* capability is logically different from the regular network data communication.

Further the ontology also includes the key social behavioral rules that every member of the society of cognitive nodes needs to follow. From the cognitive networks point of view the rules related fairness, non-interference, network utility maximization are such social policies.

With this we see the cognitive process model to have the following basic capabilities, *Sense*, *Analyse*, *Decide* and *Act* supported by two pervasive capabilities ; *Learn* and *Communicate*. Communication will be more important when any of the elementary capabilities in the cycle is distributed.

2.3 Context Space and Ontology

So for a cognitive node, awareness about the 'self' and the 'world' is an essential pre-requisite for self-adaptations and it can be addressed under the general framework of context awareness. Here the term context is used in a more general sense that context is the information that surrounds, and gives semantic meaning to, an entity. Alternately it can be said that context is the raw information that, when correctly interpreted, identifies the characteristics of an entity and it is a function of time and environment. In the case of a network element, this includes the information about the device, network, user and applications.

Conceptual knowledge provides the meaning of fundamental notions in a domain of interest as well as fundamental principles relating to those concepts. This kind of knowledge is referred as ontology and they are formally represented using ontology languages. A discussion on this aspect is given later in this chapter.

To represent and analyze the cognitive system the crucial first task is to get a clear conceptual understanding of the set of contexts. An accepted way of obtaining a conceptual understanding is to strive for a coordinatisation of the set which refines "conceptual" into a "structural" understanding [10]. Here we propose the approach of using a context space to bring that kind of a structural understanding to the system.

2.3.1 Context Space

The basic premise of this approach is that a structure for a robust architecture for a context aware network element should be based on a detailed structural analysis of the spaces associated with this application. A possible strategy is to enumerate a set of contexts; define a (possibly mixed) coordinatisation of this set to make it into a context space. The "axes" of this space will play the role of descriptors of the adaptation loops. These descriptors have to be rich enough so that any "new" adaptation one may come up with at a later point in time can be adequately described. This also can enable the discovery of new adaptations to satisfy a given goal. This also brings the *open adaptation* capability towards a future proof system. A detailed discussion on the coordinatisation and context space structure is given in our paper [11].

On the implementation side, the structure of the context space should help in the identification of generic algorithms applicable for a maximal set of problems addressed by the cognitive engine. It should also be amenable to use formal methods for design and analysis of system behaviors.

The set of contexts relevant to a cognitive network element is considered as inhabiting a multidimensional space as exemplified in Figure 2.1. There are many alternatives possible as the coordinates for the context space. Here we choose to structure major dimensions of the context space in terms of 'Self', representing the internals of the network element (device) and 'World' representing the externals. They are further refined in a hierarchical manner. It should also be noted that some of the dimensions are refinements of other dimensions. It is in this sense we call this coordinatisation as mixed.

We define context space is defined as a triple <*C, IC, DC*> where *C* is a partial order < *C*,' ≤ ' > , where ' ≤ ' indicating the sub-context relation, *IC* is a family of structures indicating the informational relations and *DC* is a family of structures indicating the tradeoff relations. The structure *C* modularizes the context space so that it can be easily accessed, updated and managed. The trade-off relations *DC* is used to identify the relations where resolution of trade-off is required for decision making or evaluating the related information elements. *IC* includes different structures to facilitate the representation of deductive relationships such as dependency, causality and monotonicity. The deductive relations are used to deduce the state of one information element from one or more elements involved in the relation.

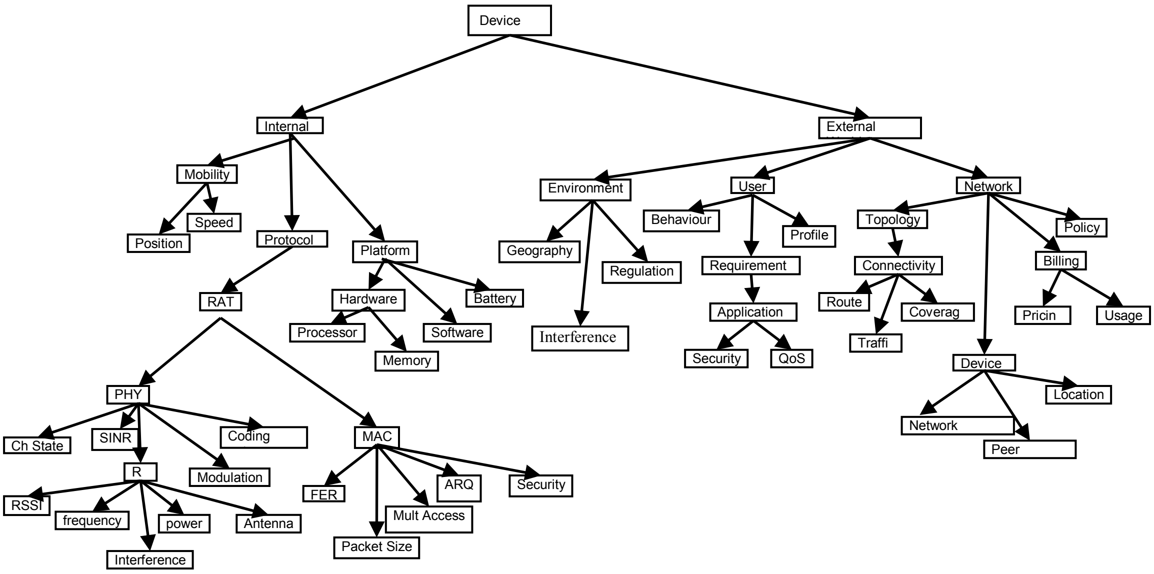


Figure 2.1 Example illustration of the context graph for a Cognitive Radio device

2.3.2 Cognitive Processes

Cognitive Processes are a sequence of executable functions following the meta-process structure *'S-A-D-R'* operating on the Context Space (Figure 2.2).

We introduce an additional meta process element '*Communicate*' to emphasize the requirement of having the communication capability to synchronize with the process elements and other entities involved. This pervasive '*Communicate*' capability is required by a cognitive engine to boot strap and connect with other cognitive elements in the network. As proposed in the guiding principles from previous section the requirement of having a unitary ontology and mechanism to maintain its truthfulness is enabled through this '*Communicate*' capability (Figure 2.3) When the process elements are local then this capability may get implemented through suitable interprocess communication (IPC) mechanisms or Application programming interfaces (API). But more importantly to communicate in non-local manner across multiple CEs the concepts like Cognitive Pilot Channels (CPC) [14] and Cross layer Coordination and Signaling Plane (CCSP)[6] need to be considered.

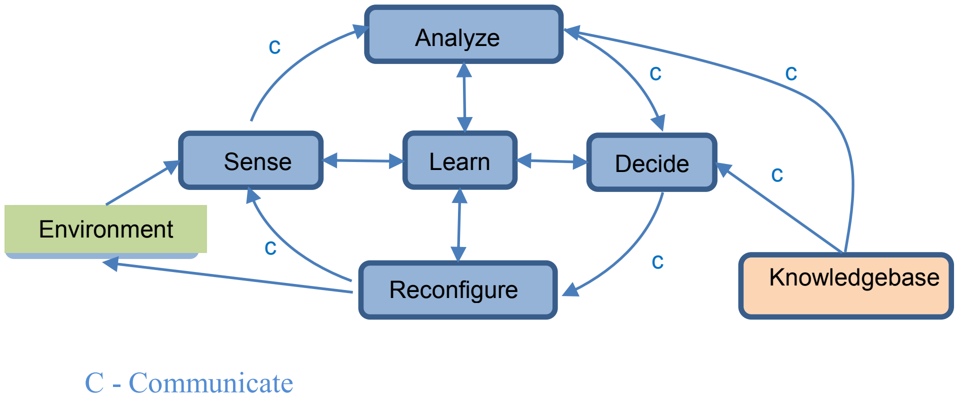


Figure 2-2 Enhanced Cognitive Meta Process

Cognitive adaptations are aimed at achieving goals specified at a higher level related to QoS, connectivity, security, mobility etc. Once these high level social goals are specified by the policy then a suitable plan or process is synthesized or fetched by the execution unit for deployment. More details on this approach is given in our paper [9].

There are two categories of goals for adaptation in the cognitive network system.

1. Maintain the equilibrium in a state/situation
2. Optimize performance and efficiency around an equilibrium

In the first case the system is put into a state of operation which is expected to be stable for a sufficiently long time, but its stability can be challenged by various external factors including channel conditions, traffic load, mobility etc. The adaptation scheme should keep the system in its equilibrium.

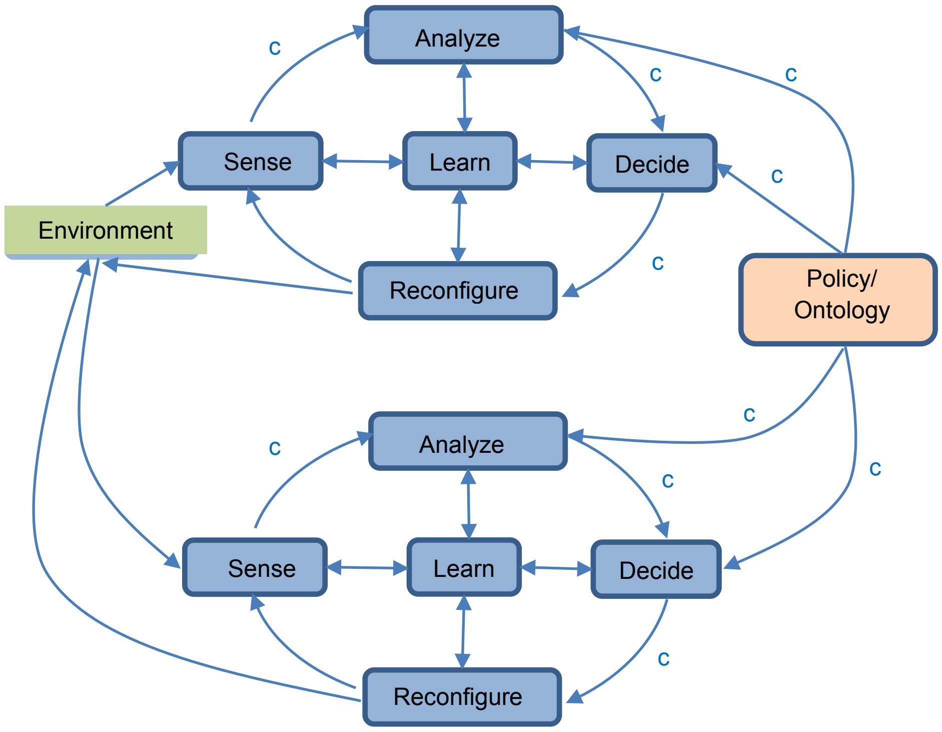


Figure 2.3 Cognitive Process and its interaction between two nodes

In the second case the system is in a state of equilibrium, but there is a quest for optimization of performance and resource consumption dynamically, without affecting the equilibrium. Most cases this will involve trade-off between multiple cost functions for optimizations. For example there is a trade-off between Quality of Service and Power consumption.

A typical context based adaptation loop starts with sensing or measuring the context information, then analyzing the raw information gathered to perform appropriate context abstraction/interpretation, presenting the composite events and abstracted information to take decision on the system configuration and finally implement the reconfiguration decision.

2.3.3 Emerging Architecture

Architecturally a Cognitive Radio can be viewed as the integration of a flexible radio platform with a cognitive management layer. A software defined radio (SDR) can provide the required reconfiguration services and a Cognitive Engine (CE) implements the cognitive capabilities required for the device. The change in SDR to Cognitive Radio to Cognitive Networks is in the scope of intelligent reconfigurations using the cognitive engine as depicted in Figure 2.4. While the scope of SDR is at Radio level, the domain of Cognitive Radio includes the link layer. Cognitive Network has an end-to-end perspective of the network and the scope spans from physical layer to application layer. Figure also shows an emerging view of the cognitive node architecture at a high level. Here the SDR platform is abstracted as a black-box with certain knobs for control/configuration of the device and meters which provides various context measurements. The cognitive entity/engine (CE) consists of a cognitive execution engine which runs a goal driven cognitive process. The meta model of this process is a cognitive cycle – *Sense, Analyse, Decide, Reconfigure* (SADR). It can be easily mapped to various cognitive cycles proposed in the literature (eg: OODA). It uses a knowledge base including ontology, rules, models, plans, and state representations. The cognitive execution engine could be considered as a virtual machine running the cognitive programs that are abstract descriptions of various cognitive processes involved. This module will be responsible for the run-time refinement, composition and scheduling of the processes based on the context space knowledge. A detailed discussion on this architecture and guiding principles are given in our paper [9].

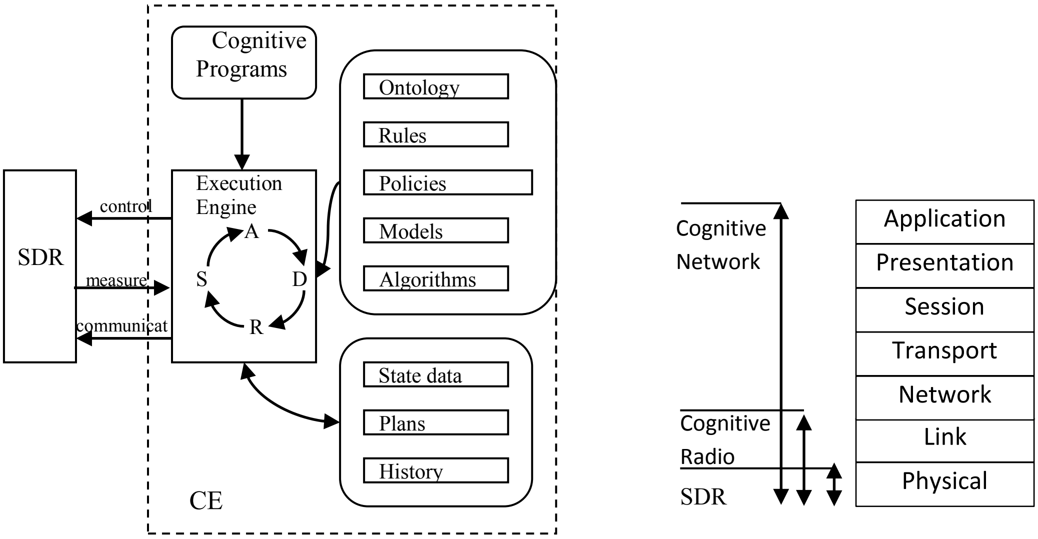


Figure 2.4 Cognitive Engine – High Level Architecture and Scope

2.4 Cognitive Engine Architecture – Key Requirements

Following the high level architecture presented, this section consolidates the key requirements of a cognitive engine (CE) from a practical perspective for applying to cognitive networks. Further it bases these requirements to propose an extended architecture for CE.

Requirements:

Knowledgebase

a) The Knowledgebase of a cognitive engine should have effective mechanisms to represent the ontology related to 'self' and 'world' as defined by a context space.

b) Handling uncertainty is identified as a challenge in Cognitive Networks. The knowledge model should support representation and reasoning under uncertainty.

c) The model should enable the representation of knowledge that is persistent and transient. The representation should have a modular and scalable structure and it should be shareable across nodes.

d) It should facilitate the inclusion of semantic information that is required by the self-configuration and self-optimization engine such as costs, type, relationships (dependency, causality) etc. It should also include the representation of network wide policies.

e) It should support modeling the dynamic behavior of the network towards use in dynamic optimization.

Cognitive Process

a) CE should sense all mandated protocol stack and other device parameters defined by the policy and they should be reported.

b) Sensing should support change detection in the network environment parameters so that efficiency of analysis and decision making could be improved with event triggering on context changes.

c) The sensing process also should enable synchronization of the reported measurements before submitting them to other consumer modules.

d) Need to support standardized interfaces for collecting the measurements from protocol layers

e) Adaptation needs to be considered with an end-to-end perspective

f) Adaptation needs to be completed before any substantial change in the external environment/situation.

g) The adaptation should support planning and operation of control loops running at different time scales.

h) The transients during the reconfiguration process need to be avoided or minimized.

i) The reconfiguration module should synchronize with the protocol stack state and procedures so as to avoid transients and faulty configurations.

General Requirements

a) Architecture should support centralized as well as distributed deployment of cognitive engines

b) It should accompany a life cycle methodology for design, development and management of the cognitive engine.

c) There should be mechanisms to assign roles and responsibilities to nodes for reporting, decision making and reconfiguring. There should be a mandate to *Sense*, empowerment to *Decide*, and authorization to *Reconfigure*.

2.5 Detailed Architecture of a Cognitive Engine

In this section we present an architectural framework for implementing the cognitive engine as a centralized or distributed entity in network elements.

Major modules identified in the architecture reflect the cognitive meta process *Sense-Analyse-Decide-Reconfigure* (SADR) cycle. The modules are grouped as State Sensing, Analysis & Decision, Reconfigure, Learning and Communication as shown in Figure 2.5. State sensing module has a mandate to collect measurement reports from the network protocol stack following the underlying networking standard procedures. It also does periodic measurement of device's internal states such as battery level, memory, processor utilization etc. The list of parameters to be measured is obtained from the context space model. The measurement may arrive at different times and periodicity. They need to be synchronized through suitable methods such as interpolation and resampling to get a common time reference. Further the sensing process detects and reports certain events and situations defined which composite contexts in the context space model are. Context parameters related to environment is monitored for any substantial change, and that 'change' event may be used to trigger the reconfiguration of the related network aspects.

Knowledgebase is the key element that is accessed and shared by all other modules. It houses the persistent and transient knowledge about the 'self' and 'world' related to the network element. A context space model incorporating the state variables, and supporting information structures constitute the key component of the knowledgebase. This is used by the sensing module to plan the measurements, state determination and various event detections. Some of the information structures in the context space model will be used to find alternate information means in case of a sensor failure. Policy and profiles (user and application) are other information elements in the knowledgebase used by Analysis and Decision unit. The Decision database includes an initial set of configurations corresponding to a set of acceptable operating points of the system. This is used as initial points for system self-configuration. This database will be augmented with additional good configurations that are found during the course of operation of the optimization process by the Analysis & Decision module. Additionally boundaries of anomalous configuration spaces, where a configuration resulted in very bad system performance, are also remembered in the database. This anomalous configuration history will help in avoiding disruptive transients during the system reconfiguration. Further policies and profiles from the knowledgebase are used to specify the boundaries of the configuration space. This will manifest in the context space structure in the form of constraints and variable assignments.

The Analysis &Decision engine uses the current state information, consults the context space model to understand the situation, and examines the policy database to choose the goals and constraints to arrive at a set of candidate re-configurations towards achieving the goals. These configurations are further filtered by considering the cost of reconfiguration and possibility of disruptive transients. The context model incorporate necessary cost attributes for computing the overall cost of reconfiguration. Time for reconfiguration is a common metric for the cost. Further there is a check for the feasibility of completing the reconfiguration within the time available for configuration. This time availability will depend on the dynamics of the external environment. The change detection function of the sensing module is expected provide this guidance to the decision maker. For quick decisions, the decision database is consulted to find a best suited configuration for the current situation. After an initial configuration the optimization process can continue based on the statics of the system state. Since wireless networks are known to be highly dynamic the optimization needs to be online and it has to undergo multiple 'SADR' cycles. The underlying optimization algorithm may change based on the specific nature of the adaptation loop. A controller is envisaged to be supporting this orchestration of optimization process.

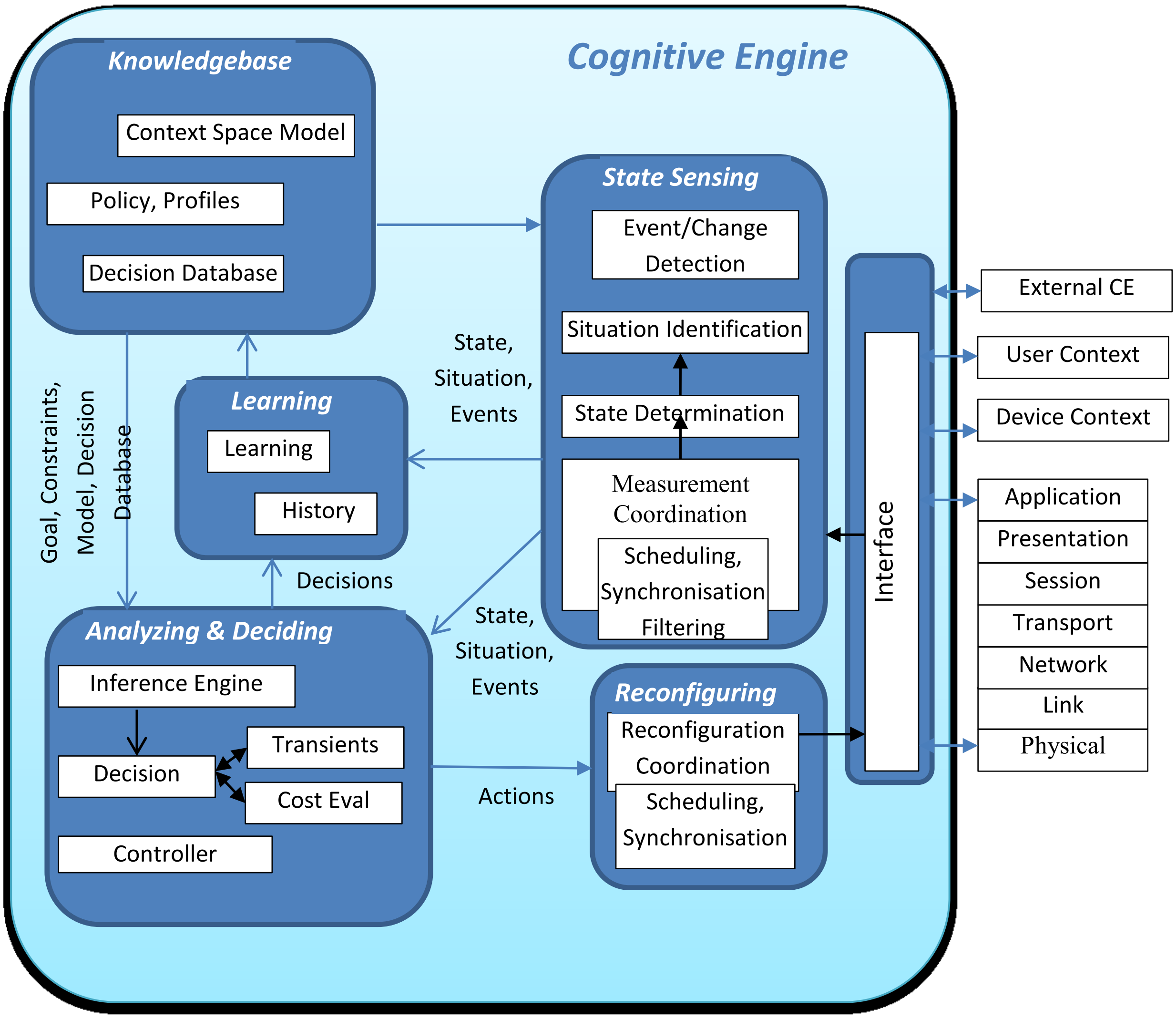


Figure 2.5 Architecture of the Cognitive Engine

Next stage is to implement the configuration decisions on the target network elements. This is required to be synchronized and scheduled with the state machine of respective protocol stacks and device platform. The function of reconfiguration coordinator is exactly this. If it is a local configuration it is done through specific APIs. Remote configuration requires a communication channel to transport the configuration variables. This is common concern for the Sensing module as well. Here we conceive an Interface module which handles interfacing the engine with local and remote configuration and sensing points.

From the interface standards perspective some of the noteworthy proposals are ULLA (Universal Link Layer API) [12], GENI (GEneric Network Interface), and Common Application Requirement Interface (CAPRI) [13]. While ULLA focuses on interfacing the PHY and Link Layer, GENI enables the detailed monitoring and configuration of the transport and network layers. Together, ULLA and GENI provide interface functionalities through the provision of a generic and portable API. For interfacing with the Application Layer the interface specified by Common Application Requirement Interface (CAPRI) is a good choice.

2.6 Embedding the Context Space Structure in the Knowledge base

A context space structure supporting the required informational elements that represents the state and dynamics of the network is a key component of the knowledgebase in the cognitive engine. The structure should support the representation of information dependency, causal relationships, and further support the incorporation of additional attributes of the system to these relation structures. Graphical models are best suited for such requirements. In the next chapter we present an investigation and a proposal in this direction.

2.7 Summary

In this chapter we proposed the basic guiding principles behind a cognitive system that should be considered while designing the cognitive network architecture. The unitary nature of the ontology representing the self & world with mechanisms to keep its truthfulness with respect to the current state of the system is a very fundamental requirement. This proliferates a system wide self-awareness capability. We further formulated important requirements of a cognitive engine in the context of cognitive networking. Based on the stated requirements we developed a detailed architecture of the cognitive engine and will be a basis for the works in the following chapters.

References

[1] D. D. Clark, C. Partridge, J. C. Ramming, and J. T.Wroclawski, A knowledge plane for the internet," in Conf. on Applications, Technologies, Architectures, and Protocols for Computer Communications, (Karlsruhe, Germany), pp. 3{10, ACM, 2003.

[2] R. W. Thomas, D. H. Friend, L. A. DaSilva, and A. B. MacKenzie, Cognitive Networks: Adaptation and learning to achieve End-to-End performance objectives, "IEEE Comm. Mag., vol. 44, no. 12, pp. 51-57, 2006.

[3] P. Sutton, L. E. Doyle and K. E. Nolan, A reconfigurable platform for cognitive networks, 1st International Conference on Cognitive Radio Oriented Wireless Networks and Communications, 1–5 (June 2006)

[4] D. Raychaudhuri, N. B. Mandayam, J. B. Evans, B. J. Ewy, S. Seshan, and P. Steenkiste, CogNet: an architectural foundation for experimental cognitive radio networks within the future internet," in 1st ACM/IEEE Int. Workshop on Mobility in the Evolving Internet Architecture, (San Francisco, California), pp. 11-16, ACM,2006.

[5] Andrej Mihailovic, G. Nguengang, J. Borgel, N. Alonistioti, "Building Knowledge Lifecycle and Situation Awareness in Self-Managed Cognitive Future Internet Networks" International Conference on Emerging Network Intelligence - EMERGING , 2009DOI: 10.1109/EMERGING.2009.10

[6] Nelson L. S. da Fonseca, "Architectures and Cross-Layer Design for Cognitive Networks", Book Chapter in Handbook On Sensor Networks Edited by: Yang Xiao (University of Alabama, USA),Hui Chen (Virginia State University, USA), World Scientific Publishing Co. Pte. Ltd

[7] J. I. Mitola, Cognitive radio for exible mobile multimedia communications," in IEEE International Workshop on Mobile Multimedia Communications (MoMuC'99), (San Diego, CA, USA), pp. 3-10, 1999.

[8] Dimitris N. Kanellopoulos, "Ontology-driven knowledge management for cognitive networks", Int. J. Enterprise Network Management, Vol. 4, No. 3, 2011

[9] Balamuralidhar P, Ramjee Prasad, "A context driven architecture for cognitive radio nodes", International Journal of Wireless Personal Communications (Springer), Vol. 45 (3), May 2008, pp 423-434

[10] C.L.Nehaniv, "Algebraic models for understanding: coordinate systems and cognitive empowerment", Proc. on second International Symposium on Cognitive Technology, CT'97, pp147.22

[11] P.S.Subramanian, Balamuralidhar P, "Intensional description of autonomic use cases", ", ICT '06, 11-12 May 2006, Funchal, Portugal

[12] M. Bandholz, A. Gefflaut, J. Riihijarvi, M. Wellens, and P. Mahonen. Unified Link-Layer API Enabling Wireless-Aware Applications. In IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), pages 1–5, Sept. 2006.

[13] Adaptive Reconfigurable Access and Generic Interfaces for Optimization in Radio Network (ARAGORN), 2008-2010. FP7 ICT project, EU commission.

[14] Cognitive Pilot Channel Enabling Spectrum Awareness, Sallent, O. , Communications Workshops , Proceedings of ICC Workshops 2009, IEEE International Conference on Communications 14-18 June 2009

3

Qualitative Probabilistic Networks for Context Space Model in Cognitive Engine

In this chapter we propose the use of Qualitative Probabilistic Networks (QPN) for modeling dynamic behavior of cognitive networks. Easy capture of qualitative knowledge from human experts to build the Knowledgebase is the major motivation. We use some of the semi-qualitative extensions of QPN to resolve trade-offs while reasoning with a QPN model of the network. Further we propose an integrated methodology for the development and use of QPN models for cognitive networks.

3.1 Introduction - Graphical Models for Cognitive Networks

Ulrich G. Oppel[14] states that "Every complex system can be determined by a causal probabilistic network without cycles and every such network determines a Markov field. Graphical models have been observed to be very useful for modeling the layered network protocol stack. Bayesian Networks (BN) is a graphical model representing the statistical relationships between random variables that is being widely used for statistical inference and machine learning. Beyond learning and representing the parameters that model the network, BN can be used to make inferences and drive a control loop that can take the network to a stable high performing operating point. However it is to be noted that in general probabilistic inference is known to be NP-hard [1].

3.1.1 Related Works

Graphical models are widely explored for cognitive network optimization. E Meshkova [2] has explored the use of simulated annealing and graphical models (BN) for cognitive network optimization. In their observation complex Bayesian model do not perform consistently better than simple probabilistic models. Moreover large graphical models tend to be computationally costly, esp for resource constrained environments. Later Georgio [3][4] showed that BN can be used for modeling a cognitive network and inference the network behavior. However they also observed the sensitivity of BN model and inference to the sample size required for training. It was noted that in some cases the BN based inference perform better with shorter training data compared to the case where large training data is used. As the number of variables increases the reasoning using BN becomes more computationally intensive. Moreover the construction of a truthful representative BN for the communication network through machine learning is found to be difficult and highly depend on the training data set. In this situation still domain experts are to be relied up on to build the qualitative part of BN and gathering reliable probabilities remains a challenge.

3.1.2 Qualitative Modeling

Traditionally, engineering systems including wireless communication networks uses quantitative models aimed at producing precise numerical results as answers to related queries. Many times such numerical results are overly elaborate and contain much more details than that is required. In some other cases the system is highly dynamic and nonlinear, and then the precision brought-in by the numerical model is misleading in presence of the uncertainty. Emulating the humans' use of common sense to reason about problems qualitatively would be an interesting direction to explore.

Generally qualitative reasoning is viewed as an abstraction of quantitative reasoning. In some cases the information required for the model is elicited from human, and that is often qualitative. There are many ways to present this abstraction. One way is to use symbolic values and intervals to represent a numeric value. This has a bearing on fuzzy sets. For example the Signal to Noise Ratio (SNR) at a radio receiver that has a numeric value of 2.32dB may be stated as '*low*' SNR. Another technique that can be followed is to find time derivatives and abstract them into 'direction of change' such as increasing / decreasing. A measurement time series of SNR having values (2.32, 3.45, 6.57) can be stated as 'increasing'. A third principle for abstraction that can be used is to simplify functions into monotonic relations. For example, the bit error rate (BER) is a non-linear function of SNR and this function can be abstracted as a monotonic causal relation from SNR to BER.

There are advantages and disadvantages of following a qualitative approach for modeling. A qualitative model is easier to specify than a complex numerical model that may have may have many parameters to be defined. They can be easily explained, understood, and verified by human. In presence of uncertainty, a precise numerical model is likely to be highly in-correct than its qualitative counterpart. However major disadvantage of Qualitative model is its limited expressibility and the information is in-sufficient for many decision making tasks where as a numerical model may perform better.

In the view of difficulties faced by the BN to scale up with the precise numerical probability values, it would be interesting to explore on how far their qualitative version can be useful for modeling and reasoning in the context of Cognitive Networks. This aspect has not been explored so far to the best of our knowledge.

3.2 Qualitative Probabilistic Networks and Signed Graphs

Here we consider two related formalisms for the Qualitative Modeling approach. They are Qualitative Probabilistic Networks and Signed Graphs.

Qualitative Probabilistic Networks (QPN) is such a structure in which the probabilistic information captured is the qualitative signs of probabilities and is more robust than exact numbers. Once the QPN is evaluated to be robust then quantitative information may be brought-in through learning process in a systematic manner. While QPN can be the basis of a full-fledged BN, the robustness of the QPN structure (though limited in information) is a reliable fall back for the probabilistic reasoning. With this various possibilities of use it is important to derive as much information as possible from such networks. In our context of knowledge representation in cognitive wireless networks, QPN can be considered as part of the context space model.

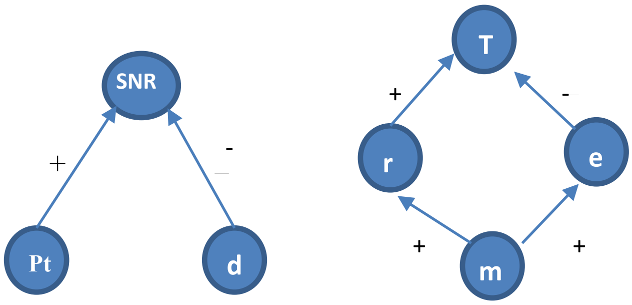
When the link signs/symbols indicating the causal influence are limited to {-,0,+} then the QPN structure can be termed equivalent to a Signed Directed Acyclic Graph (SDAG). The directed acyclic graph causal framework has shown to be a useful tool in thinking carefully about questions of confounding and causal inference [13]. With assumptions on monotonicity as indicated by the signs, we consider both basic QPN and SDAG are structurally equivalent for our discussions unless stated otherwise.

3.3 Qualitative Probabilistic Networks - Preliminaries

Qualitative Probabilistic Networks (QPN), is a special case of BN and introduced by Wellman [5]. QPN encodes statistical variables and the probability of influences between them in a digraph G=(V(G),E(G)). Nodes in the set V(G) represent the variables. We use the notations π(A) to denote the set of all predecessors of node A in G. Similarly σ(A) denotes the set of all predecessors of A. Edges in E(G) express causal influence from one node to the other node of E. It is looked upon as qualitative influences which are either + or – instead of conditional probabilities of BN. Formally the edges can be said to represent the probabilistic independence among the represented variables. A path between two nodes is *blocked* if it includes either an observed node with at least one outgoing edge or an unobserved node with two incoming edges and no observed descendants. Two nodes are said to be *d-separated* if all paths between them are *blocked*. The corresponding variables are considered to be conditionally independent given the entered observations. Though QPN has been explored in the literature for qualitative analysis of systems in biology and chemistry, its application in the context of cognitive networks has not been explored so far to the best of our knowledge.

A qualitative network associates with its digraph a set of qualitative influences such as positive, negative or no influence. A positive qualitative influence of a node B_1 to B_2 is expressed as $S^+(B_1,B_2)$ which says probability of B_2 has a positive influence from that of B_1. In other words if there is an increase in probability of B_1, then probability of B_2 also will increase regardless of any other incoming influence on B_2. This means $\Pr(b_2|b_1x) - \Pr(b_2|\overline{b_1}x) \geq 0$ for any combination of values x for the set $\pi(B_2)\backslash\{B_1\}$ of predecessors of B_2 other than B_1. The notation π is used to indicate the set of predecessors and small letters {b1,b2} indicate the values of {B1,B2}. Considering binary values for variables, b1 denotes b1=TRUE and $\bar{b}_1$ denotes b_1=FALSE. In a similar manner $S^-(B_1,B_2)$ is defined for negative influence. When the influence is non-monotonic or unknown, the representation used is $S^?(B_1,B_2)$ which indicates an *ambiguous* relationship. In QPN the nodes represent the variables and directed edges represent the influences. The edges are labeled by one of the symbols in {+,-,0,?} based on the nature of influence.

Let us consider an example from wireless communications involving three variables Signal to Noise Ratio (*SNR*), Transmit Power (*P*), and Distance from transmitter (*d*). Let us consider these as binary variables taking a value of INCREASE or DECREASE. It can be observed that there is causal relationship from *P* to *SNR* and *d* to *P* . Also it is known that $\Pr(SNR|\mathrm{P}) - \Pr(\mathrm{SNR}|\overline{\mathrm{P}}\,x) > 0$ and can be expressed as $S^+(P,SNR)$. Similarly $\Pr(SNR|d) - \Pr(\mathrm{SNR}|\bar{d}\,x) < 0$ and we get $S^-(d, SNR)$ as shown in Figure 3-1 (a). Another influence diagram involving the causal relationship between modulation (m), bit error rate (e), burst bit rate (r) and throughput T is shown in Figure 3-1(b). It can be explained in a similar manner.



(a) (b)

Figure 3-1 QPN Examples

In QPN the influence relation is assumed to follow symmetry property, that is $S^{\delta}(B_1, B_2) = S^{\delta}(B_2, B_1)$ where $\delta \in \{+, -, 0, ?\}$. According to transitivity property, the net influence of a chain consisting of nodes with only single incoming and outgoing arcs can be computed with a ⊗ operator from Table 1. Composition property mandates that multiple influences between two nodes along parallel chains combine into a single influence using the ⊕ operator from Table 3-1.

Table 3-1 QPN Operators

⊕	+	-	0	?
+	+	?	+	?
-	?	-	-	?
0	+	-	0	?
?	?	?	?	?

⊗	+	-	0	?
+	+	-	0	?
-	-	+	0	?
0	0	0	0	0
?	?	?	0	?

3.3.1 Reasoning using QPN

The reasoning in QPN is based on the idea of propagating and combining signs. It builds upon the properties of symmetry, transitivity and composition of qualitative influences. An efficient algorithm for reasoning with QPN is given by Druzdzel [6]. Subsequently Kouwen [7] has given a revised version of the algorithm and Xiang-Kun Li et.al [8] enhanced the algorithm to support the propagation of multiple observations (Algorithm 3.1). The basic idea of the algorithm is to trace the effect of observing a set of variable's values on the probabilities of the values of all other variables in the QPN by message-passing between neighboring nodes (variables). The sign of the net influence along all active trails between the newly observed variable and the other variables in the network is computed by using sign propagation and combining employing repeated use of ⊗ and ⊕ operators. For each variable, it finds the net influence in a node-sign that indicates the direction of movement in the variable's probability distribution that is caused by the new observation.

Given a QPN structure and a set of observed nodes O with their signs, the function *PropagateObservation (QPN,O,sign)* in Algorithm 3.1 performs a sign propagation based reasoning and updates all the relevant nodes of the QPN. The inferred effect of these observations on other nodes can be obtained by accessing the updated variables corresponding to the nodes of interest. A detailed discussion on the algorithm is available in [8].

Algorithm 3.1 QPN Sign Propagation Algorithm for Multiple Observations

```
O        – set of observations
sign     -  signs of observations

procedure PropagateObservation ( QPN, O, sign )
        for each $V_i \in V(G)$ do
                sign[$V_i$] ← '0'
                oppdir_sign[$V_i$] ← '0'
        end for
        for each $O[i] \in O$ do
                $X_i$ ← Bayes_Ball(V(G),O[],O[i])
```

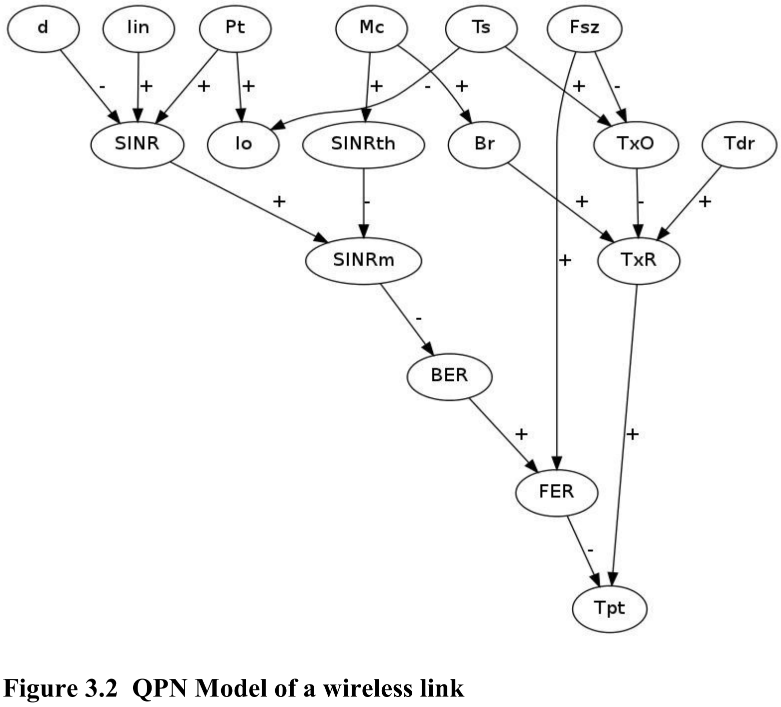
```
            PropagateSign ( Empty, O, O, sign,$X_i$ )
        end for
end procedure

procedure PropagateSign( trail, from, to, message, X)
    sign[to] ← sign[to] ⊕ message
    if to $\notin \sigma(from)$ then
        oppdir_sign[to] ← oppdir_sign[to]  ⊕  message
    end if
    trail ← trail ∪ {to}
    for each active neighbor $V_i$ of  to do
        if $V_i \notin X_i$ then
            linksign ← sign of (induced) influence between to and $V_i$
            if $to \in \sigma(V_i)$ then
                message ← oppdir_sign[to]  ⊗  linksign
            else
                message ← sign[to]  ⊗  linksign
            end if
            if $V_i \notin trail$ and not SignsEqual ( to, $V_i$, message ) then
                PropagateSign ( trail, to, $V_i$,  messagesign )
            end if
        end if
    end for
end procedure

function SignsEqual ( to, $V_i$, messagesign ): Boolean
        signsequal ← false
        if $to \in \sigma(V_i)$ then
            if oppdir_sign[$V_i$] = oppdir_sign [$V_i$]   ⊕  messagesign then
                signsequal ← true
            end if
        end if
        return signsequal
end function
```

This reasoning process can deduce the effect of a change observed by a variable on its successors. In a wireless network model this reasoning process can yield the effect of change in a control variable (such as transmit power) on the goal function (such as throughput). In an optimization setting, repeated queries on the effect of a change in control variables can be used to drive an optimization cycle. In the following sections an example of a wireless link is considered to illustrate the applicability of QPN in modeling and reasoning.

3.4 QPN Model of a wireless link

We develop an example QPN model of a cognitive radio link showing the well-known influence relationships between different cross layer parameters (Figure 3.2). The cognitive radio accesses the medium as a secondary through spectrum sensing in presence of a primary transmitter. The key parameters considered here are *TxPower*, *Signal to Noise Ratio*(SNR), *SINR Threshold*, *SINR margin*, Distance between Tx and Rx (d) , *Bit Error Rate* (BER), *Frame Error Rate* (FER), *Burst Rate*, *Tx*

Bitrate, *Tx Overhead*, *FrameSize*, *Spectrum Sense duration*, *Interference to Primary* etc. In the given model, it says an increase in *TxPower* improves Signal to Noise Ratio (SNR), which in turn reduces the Bit Error Rate (BER). A decrease in BER causes reduction of Frame Error Rate (FER) and finally it improves throughput. Similarly an increase in *modulation* order increases the SNR threshold required to achieve a given BER. At a given SNR this increase in modulation order increases the BER, affecting the throughput negatively. At the same time the increase in modulation order increases the burst rate and results into a positive influence in throughput. Effect of spectrum sensing is captured in this model as the influence of *spectrum sensing duration* on the interference caused by secondary to a primary transmitter. An increase in *spectrum sensing duration* will increase the probability of detection of the primary transmitter and reduce the probability of secondary interference to primary. Further the spectrum sensing duration has an impact on the transmission overhead (reducing the time available for data transmission). An increase in the sensing duration increases the *TXoverhead*. TxBitRate is rate of bits arriving in the transmit queue and it has a positive causal relationship with Throughput. In this model the parameters *Modulation, TxPower, FrameSize and Spectrum sensing duration* are controllable variables, while *distance* is an environmental /external variable.



Tpt – Link Throughput
Io – Interference (Outgoing)
Iin – Interference (Incoming)
TxR – Tx Bit rate
Br – Burst Rate
TxO-Tx Overhead
FSz – Frame Size
SINRm – SINR margin
SINRth – SINR Threshold
Tdr – Transmit Duty ratio
Pt-Transmit Power
Mc-Modulation & Coding
d – Distance
BER – Bit Error rate
Ts – Spectrum Sensing duration

Figure 3.2 QPN Model of a wireless link

Now we analyse the above QPN model to understand the reasoning process and how it can be useful in a cognitive engine in the following sections.

3.5 Qualitative Probabilistic Reasoning

The reasoning process can deduce the effect of a change observed by a variable on its successors. It is based on the idea of propagating and combining signs. It tells us the effect of a single variable change on the whole QPN. The knowledge of the cause-effect relation between control parameters and goal variable of a system model will be useful in system control and optimization. We would use this inference mechanism to see the effect of changing a control in previous example (Figure 3.2) on a performance variable, that is Throughput here.

Case 1: Change of TxPower
Let $S^{\delta}(TxPower, Throughput)$ states the influence of *TxPower* on *Throughput*, where δ is the sign of the influence.
The δ can be obtained by propagating the sign over the chain *TxPower->SINR->SINRMargin->BER->FER->Throughput.*

$$\delta(TxPower, Throughput) = ' + \otimes + \otimes - \otimes + \otimes -' = ' + '$$

This conveys that *TxPower* has a positive influence on Throughput. Alternately this inference is stated as $S^{+}(TxPower, Throughput)$.

Case 2: Change of Modulation
In this case there are two active paths from *modulation* to *throughput*. The net influence is inferred as
$\delta(modulation, Throughput) = \delta(Path\ 1) \oplus \delta(Path\ 2)$

$$= '(+ \otimes + \otimes +) \oplus (+ \otimes - \otimes - \otimes + \otimes -)'$$

$$= ' + \oplus -' = '?'$$

Therefore this inference can be stated as $S^{?}(Modulation, Throughput)$ and it conveys that with the given information it is not possible to unambiguously infer its net influence of *Modulation* on *Throughput*. This type of ambiguity resulting relations are termed as trade-offs.

Spread of the ambiguity sign '?' on inference is a major problem in QPN. Often additional information on the relationship is needed to resolve this crisis. Assigning a relative influence metric that factors-in strength of influence is one approach to address the problem. Renooij [9] handles this problem of achieving trade-off under conflicting influences by enhancing the QPN with the introduction of relative influence strength measures which may be said to be a semi-qualitative approach.

Another aspect to be considered is the dynamism of the strength of influence between two variables, that is not captured in the model. In the present example the influence of SNR Margin on BER is indicated as '-'. While this true in a general sense, the influence strength on BER is high for low SNR Margins and is negligible at high SNR Margins. That is the influence sign changes from '-' to '0' for high SNR Margins. This calls for a mechanism to incorporate conditional signs or situation specific signs.

These two aspects of relative influences and situation specific signs are brought-in in the following sections to enhance the QPN model.

3.6 Enhanced QPN Model for the CR link

Assigning an influence metric that expresses relative strengths is an approach to resolve the problem of ambiguity. We propose to explore the use of enhanced QPN model [9] here. It works on an extended list of signs {-,--,+,++,0,?} representing relative strengths of influences such as positive, strongly positive, negative, strongly negative etc. This provides one more level of differentiating positive and negative influences so that the resultant ambiguities can be reduced while combining them.

In an enhanced QPN the strong and weak influences are partitioned into two disjoint sets and a cut-off value α is used for the purpose. For example a strong influence $S^{++}(A, B)$ is determined when $P_r(b/ax) - P_r(b/\bar{a}x) \geq \alpha$ for any combination of values *x* for the set X of parents of B other than A. Further $S^{+}(A, B)$ expresses $0 \leq (P_r(b/ax) - P_r(b/\bar{a}x)) \leq \alpha$. The sign '$+^{?}$' indicates a positive influence sign that is ambiguous in its relative influence. In a similar fashion the extended signs can be determined for negative influences as well. Practically the cut-off value α need not be established explicitly. The partitioning into strong and weak influences will be elicited from domain experts. A detailed formulation is available in [9]. A review of that work requires extensive introduction to the formalisms involved and is beyond the scope of this thesis. At this point it should be sufficient to say that the enhanced QPN will need the operators $\otimes$ and $\oplus$ to be redefined for the new signs. These operators use an additional information element called *multiplication-index lists* that helps to resolve the trade-off. The relative strengths are expressed as a polynomial in α , and the *multiplication-index list* is the list of indices of α in that polynomial. However this α is used internally by the sign propagation algorithm and human experts who provide the qualitative information need not worry about it.

Table II defines the extended operators $\otimes$ and $\oplus$. *I* and *J* are the *multiplication-index list* corresponding to the two operands and it gets generated during the sign propagation. The same sign propagation algorithm referred in the previous section is used here with the redefined operators. *I* will be set to *0* and *J* will be set to *1* for the first operation of the propagation.

Table 3-2 QPN Operators for extended signs

$\otimes$	$++^{J}$	$+^{J}$	0	$-^{J}$	$--^{J}$	?
$++^{I}$	$++^{I+J}$	$+^{J}$	0	$-^{J}$	$--^{I+J}$	?
$+^{I}$	$+^{I}$	$+^{I+J}$	0	$-^{I+J}$	$-^{I}$	?
0	0	0	0	0	0	0
$-^{I}$	$-^{I}$	$-^{I+J}$	0	$+^{I+J}$	$+^{I}$	?
$--^{I}$	$--^{I+J}$	$-^{J}$	0	$+^{J}$	$++^{I+J}$	?
?	?	?	0	?	?	?

$\oplus$	$++^{J}$	$+^{J}$	0	$-^{J}$	$--^{J}$	?
$++^{I}$	$++^{m}$	$++^{I}$	$++^{I}$	a)	?	?
$+^{I}$	$++^{J}$	$+^{0}$	$+^{I}$	?	d)	?
0	$++^{J}$	$+^{J}$	0	$-^{J}$	$--^{J}$	?
$-^{I}$	b)	?	$-^{I}$	$-^{0}$	$--^{J}$	?
$--^{I}$	?	c)	$--^{I}$	$--^{I}$	$--^{m}$	?
?	?	?	?	?	?	?

Where m = min(I,J)
a) $+^{0}$, if $I \leq J$ else '?'
b) $+^{0}$, if $J \leq I$ else '?'
c) $-^{0}$, if $I \leq J$ else '?'
d) $-^{0}$, if $J \leq I$ else '?'

We propose to enhance the CR link example from Figure 3.2 using this extended sign representation. In the model it should be observed that some of the influences are not static but varying based on the value of their dependent variables. For example the strength of negative influence of *SINRMargin* on BER is strong when *SINRMargin* is very low and it is weak when SINR is much higher. Using the extended signs it can be stated as $S^{--}(SINRMargin, BER)\ for\ (SINRMargin < 0)$ and $S^{0}(SINRMargin, BER)\ for\ (SINRMargin > \varepsilon_{SH})$ where ε_{SH} is a threshold value.

Similarly when FER is very low the influence of FER (below a threshold ε_{FL}) on throughput is negligible. For high FER this influence is strong. This can be stated as

$S^{0}(FER, Throughput)$ for $(FER < \varepsilon_{FL})$
$S^{--}(FER, Throughput)$ for $(FER > \varepsilon_{FH})$
$S^{-}(FER, Throughput)\ otherwise$

where ε_L ε_H are lower and upper threshold values for FER.

The observation is that in many cases the strengths of influences are not static but changes based on certain conditions such as low SNR, high SNR, low FER, high FER etc. The QPN model needs to be extended to incorporate these context specific changes. Towards this we propose to integrate a methodology of context specific sign propagation along with enhanced QPN introduced in [9][10]. The details are provided in the following section.

3.7 Using Context Specific Signs in Extended QPN model

Major aspect of context specific, or alternately situation specific, signs is that the influence sign is not static but dynamic. The sign will be dependent on a context function C(x). A new context node C_i will be added in the QPN structure and its value will depend on the evaluation of a context function associated with it. Let δ(C_i) denote the associated sign of the related influence.

To incorporate the context specific signs for the above CR link example we define two context nodes C_1 and C_2. These context nodes evaluate to a sign based on the contexts they represent. The state of C_1 and C_2 are used to determine the context dependent signs of the edges *E*(*SINRMargin,BER*) and *E*(*FER,Throughput*) respectively and are represented as $\delta(C_1)$ and $\delta(C_2)$. See Figure 3.3.

The state of the context node C_1 is defined as:

$$c_1 = \begin{cases} 0, SINRMargin > \varepsilon_{SH} \\ -1, otherwise \end{cases}$$

Please note that c_1 is the respective context dependent sign is expressed as $\delta(c_1 = 1) = '-$ $-'$, $\delta(c_1 = -1) = '-$ ‘ and $\delta(c_1 = 0)$=0.

Similarly the value of the context node C_2 is defined as $c_2 = \begin{cases} -1, (FER > \varepsilon_{FH}) \\ 0, otherwise \end{cases}$

The respective context depended sign is expressed as $\delta(c_2 = 1) = 0$, $\delta(c_2 = -1) =' -$ $-' and\ \delta(c_2 = 0) =' -'$.

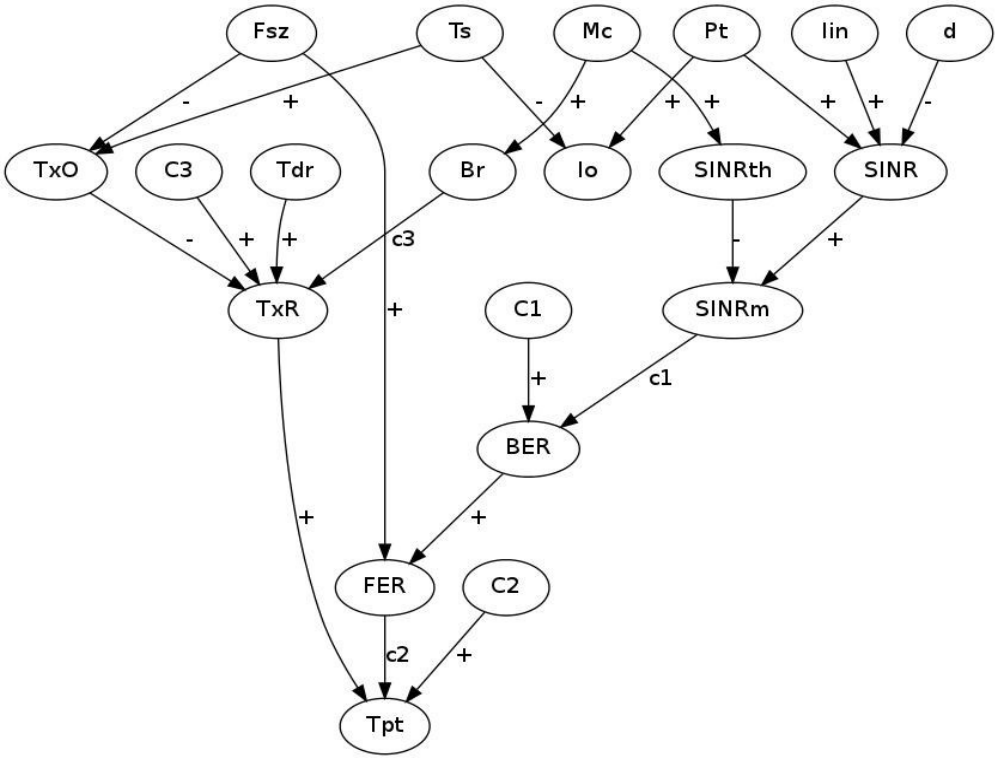


Figure 3.3 QPN Model of the wireless link with context specific signs with two identified contexts C_1 and C_2

3.8 Inference using Extended QPN model

Now let us look at the influence of *TxPower*. From the earlier model discussed in section D, Case 1, the influence of *TxPower* on *Throughput* was inferred to be '+'. But we know that this is not a linear influence and it saturates beyond a point. The key influence of this effect is from the relation between *SINRMargin* and *BER* which comes within the corresponding influence chain. Incorporation of a context node C$_1$ in the modified QPN model can handle this as follows.

$$\delta(Txpower, Throughput) = '\big(+ \otimes + \otimes \delta(c_1) \otimes + \otimes \delta(c_2)\big)' = \begin{cases} '0', & \delta(c_1) = '0' \; or \; \delta(c_2) = '0' \\ & ' + ', otherwise \end{cases}$$

Now let us have a relook at the influence of *modulation* on *Throughput*. The inference process described in section 3.5 (case2) resulted in an ambiguity sign '?'. With the extended sign model shown in Fig 3.2 and using the definition of operators from Table 3.2, we can write

$\delta(modulation, Throughput) = \delta(Path\ 1) \oplus \delta(Path\ 2)$

$= '(+ \otimes + \otimes +) \;\oplus\; \big(+ \otimes - \otimes \delta(c_1) \otimes + \otimes \delta(c_2)\big)'$

$= \begin{cases} ' + ', \delta(c_1) = '0' \; or \; \delta(c_2) = '0' \\ \qquad '?', otherwise \end{cases}$

So it can be observed that there is an improvement in the trade-off resolution. However the inference result is still ambiguous when $\delta(c_1)$ and $\delta(c_2)$ are non-zeros.

To reduce this ambiguity further we explore by assigning stronger signs to all edges with static signs in path 2 (ie *Modulation, SNRMargin,…,Throughput*). That is '+' is changed to '++' and '-' is changed to '--'. But the net influence on the sign propagation still result into the same level of ambiguity:

$$\delta(Modulation, Throughput) = \begin{cases} '0', \delta(c1) = '0' \; or \; \delta(c2) = '0' \\ '?', \delta(c1) = '--', \delta(c2) = '--' \\ \qquad '?', otherwise \end{cases}$$

Here, the fact that at high FER *Throughput* will go down with increase in *Modulation,* is not getting represented by this extended sign representation too. Even if we assign strong signs to all links in the *path2*, in last phase of the sign propagation the composition at Throughput node is '$+^3$' $\oplus$ '$--^5$' = '?'. This is a drawback we find with extended signs when too many arcs are involved and has a major asymmetry in the number of arcs in the conflicting paths.

This problem we propose to solve by an additional context node C$_3$ with it value defined as

$$c3 = \begin{cases} 0; FER > \varepsilon_{FH} \\ 1; otherwise \end{cases}$$

A context depended sign $\delta(C_3)$ is assigned to the edge E(*TxbitRate*,*Throughput*). With this it can be seen that the ambiguity is resolved to be '-' for high FER case.

The procedure of using context depended signs that switches to '0' is equivalent to temporarily removing the edges and that goes along with philosophy of balancing the respective sign graph.

In general enhanced QPN with context specific signs indeed reduce the trade-off problem, though the ambiguity is not completely eliminated. Considering the fact that practically it is not feasible or difficult to remove the uncertainty associated with the resultant inference completely, it is important and useful to state the level of uncertainty explicitly. In that sense QPN based inference has some positive advantage of explicitly stating the ambiguity compared to many quantitative approaches where measure of confidence on the inferences are not easily available.

3.9 Hierarchical Reasoning

While it is relatively easy to go with extending the signs for smaller networks, it is a complex and cumbersome process for large QPNs. Here we propose to adopt a method using network decomposition to sub-nets and then subjecting it to hierarchical reasoning [12]. Each subnet is just a tree or a reduced graph, with one head and one tail each. The head node has its output degree more than 1, and the tail node has its input degree is more than 1. All the other nodes' input degree and output degree are all equal to 1. Here sign propagation is done for each of the sub-graphs individually first. Those sub-graphs under trade-off are selectively applied with sign extensions and other trade-off measures. With the decomposed subnets we construct a hierarchical QPN (h-QPN) model. In h-QPN model the nodes included are only the heads and tails of the sub-graph. The edges hide the respective subnet and carry the sign of the subnet (determined through local sign propagation within the sub-net). In case any of the subnets have a trade-off relation then suitable resolution mechanism is deployed at the sub-net level. This is done through either enhanced signs, or context specific signs or both. Eventually this procedure will result into a simpler QPN model.

While abstracting the causal influence of a QPN variable A on another variable B in terms of a sign, we use an underlying assumption regarding the monotonicity of the said influence. If A is a control variable (such as *modulation*, *transmit power*), its domain is often discretized into an ordered set to make this relation monotonic. An increase or decrease of the variable is implemented by incrementing or decrementing the domain index. In some cases the increase in independent variable may not make appreciable change in the dependent variable for a few settings. This too can be brought inside the monotonicity concept if the relation is non-increasing or non-decreasing. While this is feasible in many cases, there are situations where the relation is not at all monotonic. Next section presents a couple of such cases and proposes an approach to handle the related issues.

3.10 Some Challenges in QPN representation

A few challenges are encountered while modeling the QPN representation of some of behaviors of wireless communication system. Expressing non-monotonic relations and dependencies between control variables are posing some difficulties. In this section we discuss some specific cases and propose problem mitigation approaches that can be generalized for applying in similar cases.

3.10.1 Handling variables that have non-monotonic influences

There are situations where a set of variables have non-monotonic relations. For example let us consider different Multiple Input Multiple Output techniques (MIMO) as part of existing and advanced wireless communication standards.

Table 3-3 shows high level influences of various MIMO techniques on *Bit rate* and *SINR* with reference to WiMAX standard. This is also subject to certain constraints on the number of Transmit antennas (Nt) and number of receive antennas (Nr) used. Further the total number of antennas also has an impact on the strength of these influences which are not considered in this example.

Table 3-3 MIMO configurations and their influences on bit rate and SINR

MIMO Techniques	Rate	SINR	Nr	Nt
Reliability Enhancements				
Selection Combining (SC)	0	+	≥1	1
Max Ratio Combining (MRC)	0	+	>1	1
Tx Select Diversity (TxS)	0	+	≥1	>1
Beam forming (BM)	0	+	>1	>1
STBC	-	+	>1	>1
Precoding/capacity enhancements				
Linear Diversity Precoding (LDP)	0	+	≥1	>1
Eigen Beam forming (EBM)	+	+	≥1	>1
General Linear Precoding (GLP)	+	0	>1	>1
Spatial Multiplexing (SM)	+	0	>1	>1

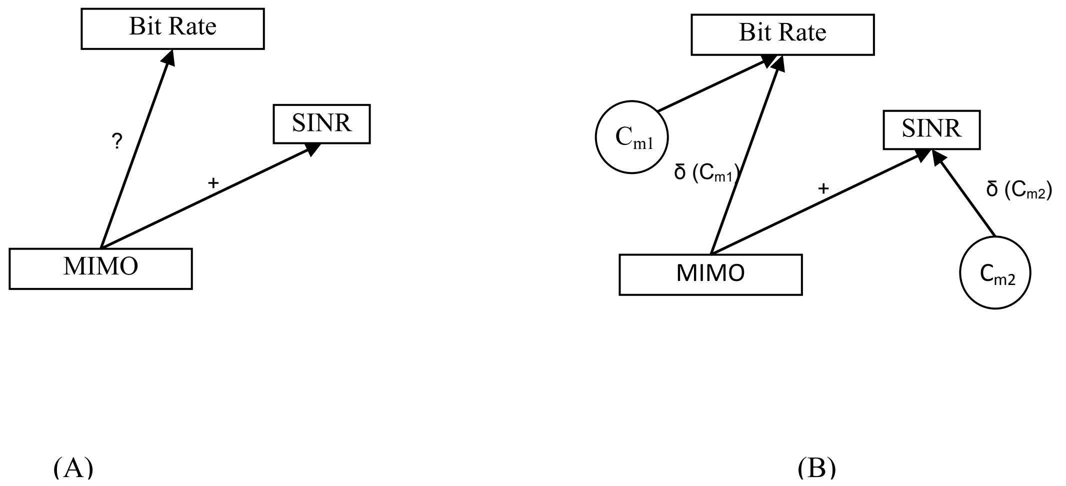


Figure 3.4 Splitting non-monotonic influences to piecewise monotonic influences - example case of MIMO configurations (A) - Relationship with Bit rate is non-monotonic (B) Split up the domain of MIMO into two ordered subdomains with C2 constraints only one of them is active at a time.

The influences listed in Table 3-3 is depicted as an influence diagram in Figure 3.4A. While considering the MIMO configurations as a whole, its influence on *Bit rate* is mixed and cannot be ordered to induce a monotonic relation. However its influence on *SINR* can be termed as weakly monotonic where some configurations have no effect while some others have positive effect.

In order to get a definite signed relationship we split the MIMO variable into four having subdomains as given below. Domains of each split variables are ordered, however the ordering shown in this example is indicative only. In practice the ordering will depend on specific implementation of each of techniques.

$MIMO_1$ = {GLP,SM}
$MIMO_2$ = {EBM}
$MIMO_3$ = {STBC}
$MIMO_4$ = {SC, MRC, TxS, BM, LDP}

Here we propose to use the context specific sign approach. Introduce two context specific nodes C_{m1} and C_{m2} with their values defined as

$$C_{m1} = \begin{cases} 1, MIMO \ \in MIMO_1 \cup MIMO_2 \\ 0, \quad MIMO \in MIMO_4 \\ -1, MIMO \in MIMO_3 \end{cases}$$

.

$$C_{m2} = \begin{cases} 1, MIMO \in MIMO_2 \cup MIMO_3 \cup MIMO_4 \\ 0, MIMO \in MIMO_1 \end{cases}$$

This context specific sign switching takes care of the correct sign propagation (Figure 3.4B). In short what we propose here is to split the domain of the non-monotonic variable so that each subsets can be ordered to induce monotonicity. Then use appropriate context variables to switch them.

3.10.2 Handling Coupled Controls

Sometimes there are situations in which the controls are dependent on each other such that certain combinations of the control values are not allowed. One immediate example is Modulation and Coding Rate. While domain of Modulation is {'BPSK','QPSK', '16QAM', '64QAM'} and that of Coding rate is { ½, 2/3, ¾ }. However these controls cannot be moved independent of each other. The allowed combinations are { (BPSK,1/2), (QPSK,1/2),(QPSK,2/3), (16QAM,1/2),(16QAM, 3/4),(64QAM,2/3)(64QAM,3/4) }. One approach we have used is to combine these two controls and synthesize a joint control variable *Modulation&Coding* and order the domain such way that monotonicity is preserved (Figure 3.5).

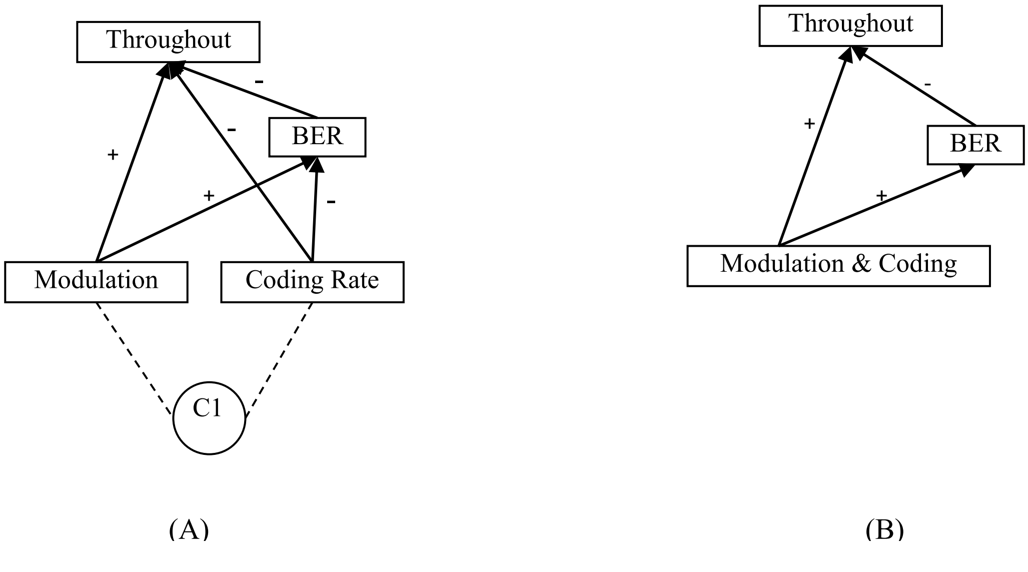


Figure 3.5 Simplified influence diagram depicting coupled control of Modulation and Coding Rate. (A) Two individual controls with a constraint C1 that restricts the combination of assignments (B) Combined control with a re-ordered domain

Another approach is to check if there is a primary secondary relationship between the coupled variables. For example the primary variable provides a coarse control effect while the secondary control has a finer impact on the same set of intermediate or objective variables. In such cases whenever the primary variable is changed the secondary variable is brought to its minimum possible value and fine tuning is done subsequently. In the above case the situation is similar. *Modulation* provide a coarser control on *BER* and *Throughput* while *Coding rate* provide a finer control.

3.11 Additional semantics through attributes to Nodes and Edges

To provide additional semantics to the state variables for context aware manipulation by cognitive engine we propose a set of node attributes. The attributes include classification of variables as 'Control', 'Environment', 'Concept', 'Performance', and 'Observed/Measured'. There is an additional categorization proposed on variables as either 'deterministic' or 'stochastic'. This is to distinguish between some variables those are deterministically computed or set by the policies from others those are stochastic in nature. One application of this categorization is in pruning the domain of QPN sign propagation while reasoning.

Control variables: These are deterministic variables that can be controlled/configured. These variables have a discrete ordered domain so that there is notion of increase/decrease is established by incrementing/decrementing the domain index. This is also required to establish local monotonicity with its child nodes. A cost attribute is used for the arcs incident on the nodes which captures the estimated time taken for effecting the new control setting on the system.

Observed/Measured variables: They are measured or observed from the system and reported to the designated cognitive engine. The measurement is in general a real value that describes a physical state/phenomenon. This variable has an additional binary state value which depicts a change (increase/decrease) from previous state. There is a threshold value specified to detect this change through comparison. In certain cases they can have an associated mathematical expression with which it computes its state from a set of dependent measurements. Such nodes provide a higher level of abstraction to the dependent observed nodes. The arcs going out from a measurement node have an additional attribute on cost of measurement. The cost considered here is the time taken for the measurement process and communicating it to the child node.

Environment Variables: They are external variables that are beyond direct control of the local system. They may change based on their external dynamics and a high level of uncertainty is attributed. One attribute that is captured as part of environment variable is the specification of an expected time duration (interval) before a change can occur. This is useful for the cognitive engine to plan the actions based on the dynamics of environment variables and check the feasibility of completing the adaptation before any appreciable change of context occur.

Concept Variables: They are variables that are included for conceptual explanation. They may derive logical values from their child nodes through sign propagation.

These node classifications and attributes are beyond the standard QPN model, but are used to extend the QPN structure to represent the context space that is used by the cognitive engine.

Now we propose a methodology for preparing the context space based on QPN structure as a knowledge base to the cognitive engine.

3.12 A Methodology for Preparation of QPN Models

Here we propose a methodology for preparing QPN models of wireless communication systems that can be integrated as a knowledge base in the Cognitive Engine (Figure 3.6). There are two approaches available for the construction of QPN structure. One is structure learning through training data. The process starts with the identification of a dictionary of terms that describe related concepts, parameters etc. that can depict the state of the system. They are majorly the state variables that span the system context space. Next step is to identify pairwise causal relations between these variables and assign suitable QPN signs that truthfully describe the relationships. This, we suggest, best be done by a set of human experts to construct the QPN structure in a modular manner. Since the model is human readable and it can be reviewed and certified by the expert team. Further enhancements to the basic model may be made by the cognitive engine through additional structure learning from the data collected while in operation.

Next step is to specify additional semantics to the variables as node attributes. As described in previous section (3.9) the attributes include classification of variables as 'Control', 'Environment', 'Concept', 'Performance', and 'Observed/Measured'. There is an additional categorization proposed on variables as either 'deterministic' or 'stochastic'. This is to distinguish between some variables those are deterministically computed or set by the policies from others those are stochastic in nature.

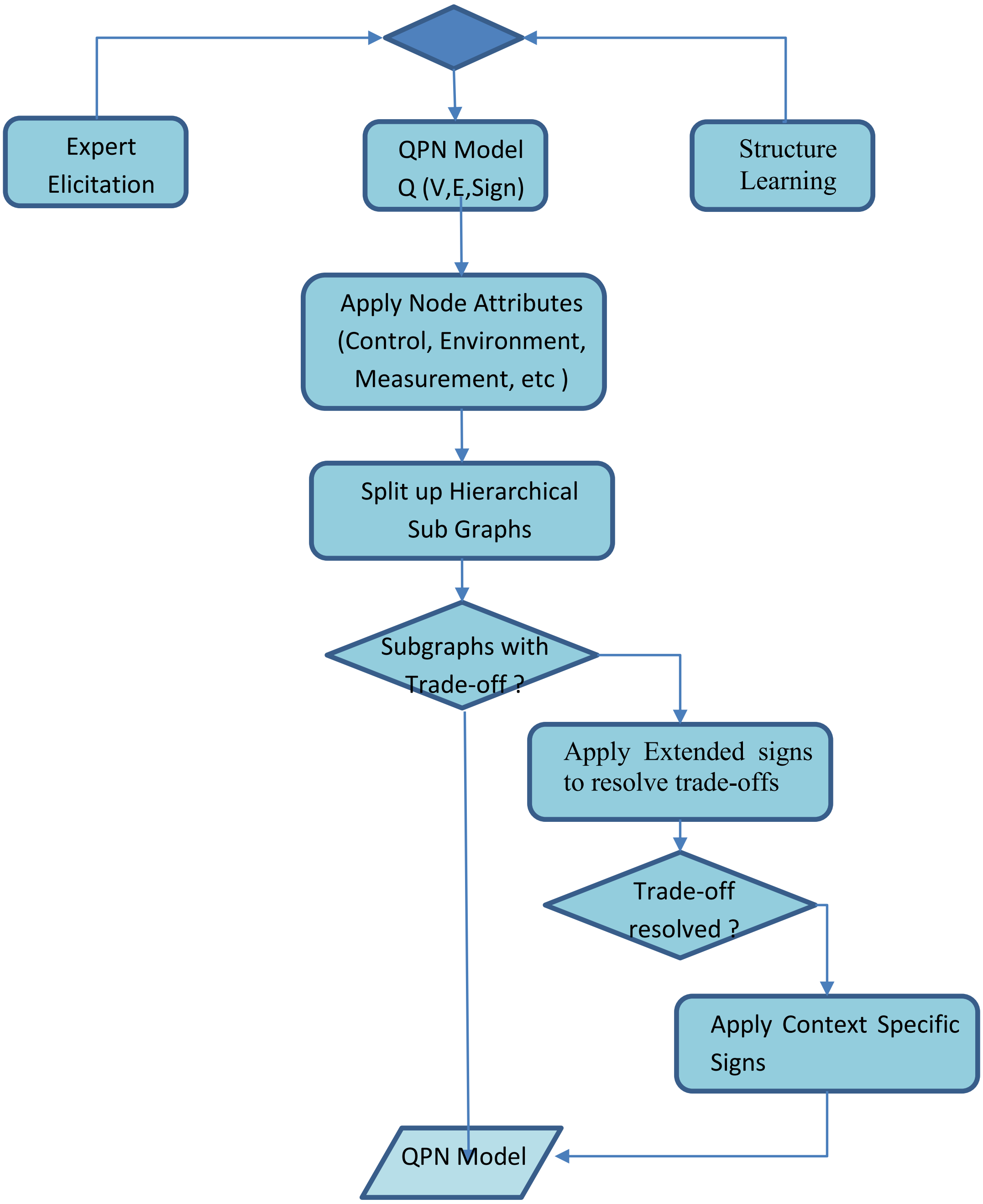

Expert Elicitation
QPN Model Q (V,E,Sign)
Structure Learning
Apply Node Attributes (Control, Environment, Measurement, etc)
Split up Hierarchical Sub Graphs
Subgraphs with Trade-off ?
Apply Extended signs to resolve trade-offs
Trade-off resolved ?
Apply Context Specific Signs
QPN Model

Figure 3.6 QPN Model development methodology

The primary concern with a QPN model is in its ability to provide truthful inferences with reduced ambiguity to the possible extent. For large graphs it is proposed to split the QPN into a set of sub-graphs organized hierarchically. This is followed by identification of sub-graphs that have unresolved trade-offs. Suitable techniques including extended signs, rough-set based weights, context specific signs etc are used to reduce the trade-off ambiguity in those sub-graphs.

With this we can say that the graph structure of the QPN developed can hold additional information structures to constitute the context space that is needed by the cognitive engine to operate its cognitive cycle efficiently and effectively. It can form a unifying structure for defining and executing multitudes of cross layer adaptations and optimizations envisaged in the vision of a cognitive network. We will present some of the use cases later in this thesis.

3.13 Summary

In this chapter we have proposed the novel use of QPN for representing the dynamic behavior of wireless networks. Various QPN inference mechanisms were evaluated and chosen a set of techniques to be used for the application in wireless communication networks. While most of the trade-off situations are addressed by the semi-qualitative enhancements, there is certain level of ambiguity left and that is proposed to be handled by the self-optimization engine in next chapter. We concluded this chapter with a proposed methodology for the construction of QPN based context space as per the requirements of Cognitive Engine laid out in chapter 2.

In the next chapter we propose the use of the above QPN structure to build an optimization engine within the cognitive engine framework.

References

[1] G.F. Cooper, The computational complexity of probabilistic inference using Bayesian belief networks, Artif. lntell. 42 (1990) 393-405.

[2] Elena Meshkova, Janne Riihij¨arvi, Andreas Achtzehn, Petri M¨ah¨onen, “Exploring Simulated Annealing and Graphical Models for Optimization in Cognitive Wireless Networks”, GLOBECOM 2009, IEEE Global Telecommunications Conference (2009)

[3] Giorgio Quer, Hemanth M, et.al, Cognitive Network Interference through Bayesian Network Analysis, Proceedings, IEEE Globecom 2010

[4] Giorgio Quer, Hemanth M, et.al, Using Cognitive Networks for Cognitive Control of Multi-hop wireless Networks, Proceedings, Proceedings of Military Communications Conference 2010

[5] Wellman, M. (1990a), “Fundamental Concepts of Qualitative Probabilistic Networks”, Artificial Intelligence, Vol.44, No.3, August 1990, p.257–303.

[6] M.J.Druzdzel, M.Henrion, “Efficient reasoning in Qualitative Probabilistic Networks”, Proceedings of the Eleventh National Conference on Artificial Intelligence”,1993, pp 548-553

[7] F. van Kouwen, S. Renooij, P. Schot, "Inference in qualitative probabilistic networks revisited," International Journal of Approximate Reasoning, vol. 50, pp. 708-720, Jan. 2009.

[8] Xiang-Kun Li, Shi-Zhong Liao, “Inference in QPNs With Multiple Observations”, Proceedings of the Eighth International Conference on Machine Learning and Cybernetics, Baoding, 12-15 July 2009

[9] Silja Renooij, Linda C Van der Gaag, Enhanced Qualitative Probabilistic networks for resolving trade-offs, Elsevier Journal of Artificial Intelligence, 2008, 172(2008) 1470-1494

[10] Silja Renooij, Simon Parsons, Linda C. van der Gaag, “Context-specific Sign-propagation in Qualitative Probabilistic Networks”, Artificial Intelligence 140, 207–230 (2002)

[11] Janneke H. Bolt, Linda C. van der Gaag, Silja Renooij, “Introducing Situational Signs in Qualitative Probabilistic Networks”, Int. J. Approx. Reasoning 38(3): 333-354 (2005)

[12] Xiangkun Li, and Shizhang Liaa, Hierarchical Reasoning in QPNs Based on Network Decomposition, International Conference on Intelligent Control and Information Processing, August 13-15,2010 - Dalian, China

[13] Tyler J. Vanderweele, “Signed directed acyclic graphs for causal inference”, Journal of the Royal Statistical Society: Series B (Statistical Methodology), Volume 72, Issue 1, Article first published online: 6 JAN 2010

[14] Ulrich G. Oppel, “Every complex system can be determined by a causal probabilistic network without cycles and every such network determines a Markov field”, Symbolic and Quantitative Approaches to Uncertainty Lecture Notes in Computer Science, 1991, Volume 548/1991, 262-266, DOI: 10.1007/3

4

Self-Optimization Driven by QPN Models

In this chapter we propose the use of QPN Model of wireless networks for driving self-optimization within the framework of cognitive engine introduced in chapter 2. A novel scheme for optimization using QPN model based inference is a part of it. It is demonstrated using a cognitive radio link towards link adaptation and throughput optimization. Further we refine the methodology of optimization with several innovative methods including monotonicity analysis and anomalous configuration avoidance using case based learning.

4.1 Introduction - Optimization in wireless networks

Optimization problems involved in wireless networks in general are largely nonlinear. Many of the cause-effect relations involved are analytically in-tractable. Hence a white-box approach for optimization where a detailed model of the wireless network, is not often feasible. There are several distinguishing aspects in wireless network protocol stack optimization from classical optimization. The network environment is highly dynamic and traditional static optimization techniques alone will not work. There will a substantial run-time part for the optimization process which may take several iterations of the cognitive cycle. However the system must conclude the actions before the situation for which the adaptation is performed changes. Further the adaptations involved will have multiple time scales to address the different dynamics of the environment variables.

In this chapter we take forward the QPN based context space model from the previous chapter and propose its use to guide the optimization process within the framework of the cognitive engine.

4.2 Optimization driven by QPN Inference

In QPN the inference process is through propagation of sign generated by an observation to other parts of the network. The sign-propagation algorithm for inference with a qualitative network basically serves to compute the effects of a single observation. However, multiple observations can be incorporated by a sequential updation and a super-positioning of the inference of individual observations [1] .

A high level logical architecture of the optimization engine is shown in Figure 4.1. This maps to the cognitive engine architecture described in Chapter 2. In the QPN model we identify following attributes to the variables

- Control/Configuration: (eg: Frequency channel, modulation, coding rate , Txpower etc)
- Performance : (SNR, PER, Throughput etc)
- Environment (channel conditions, interference, etc)
- Concept Variables (variables those are concepts – not computed or measured)

Control parameters are knobs which can be controlled /configured. The domains of the control parameters discretized and ordered so that their monotonic relations can be established. Performance variables on the other hand are meters which are measurements either raw or estimated. They have associated properties like lifetime, resolution, accuracy, timestamp. Environment variables are generally beyond the control of radio/network controller. Concept variables are hidden variables introduced for the ease of understanding and explaining.

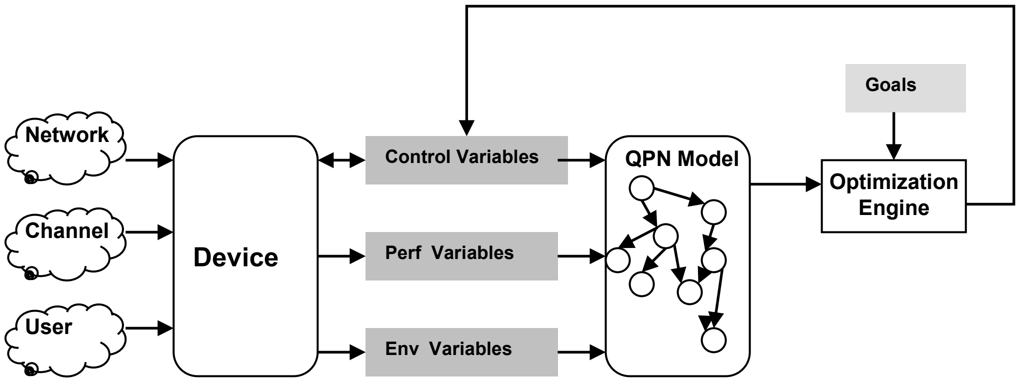


Figure 4.1 High level block diagram of the optimization flow

4.2.1 Discovering Candidates for Optimization

Generally the optimization loops are defined upfront and included as part of the knowledgebase for the cognitive engine to deploy them suitably. Given the context space structure, it is also desirable for the cognitive engine to discover opportunities for optimization and enrich the knowledgebase.

We propose following Lemma to identify candidates for optimization in a QPN Model

We define Q (V,E) as the QPN model, $Z \in V$ as the goal variable and $U \in V$ as a control variable.

Lemma 4.1 : *If an undirected cycle corresponding to a sub-graph $T \in Q$ has an odd number of negative arcs, and T is in a path from U to Z then there exists a candidate optimization cycle O(U,Z).*

Alternately, this Lemma states that if there is a trade-off relation involved in the path from a control variable to a goal variable, then there is an optimization loop to consider.

Importance of this Lemma is that the CE can use this to synthesize optimization loops by analyzing the QPN model. Also from an off-line analysis perspective this Lemma is useful for designing the optimization strategy with the updation of the knowledgebase.

4.3 QPN Model based Optimization Algorithm (QOpt)

The generalized network optimization problem considered is stated as follows:

Given a QPN Model Q with
  U = {U_i} , control variables
  Z , Goal variable

$S = \{S_i\}$, Constraints
$M = \{M_i\}$, measurements / observations

Objective is to find a configuration U^* that minimize/maximize Z, subject to S.

A sequential online optimization approach is adopted in which each control is applied sequentially. The optimization loop is explained as a pseudo code given in Algorithm 4.1. QPN model is used to infer the impact of changing each of the control variables one at a time.

The function *SelectControlVar()* makes the choice of the control variable. Basis of selection can be random, trade-off status, sensitivity or any other prioritization criterion. Here we propose to use the following heuristic approach for the sequence selection of the control variables. First, we categorize the control variables into the following:

V_p - Productivity variables
V_r - Resource variables
V_t - Tuning variables

In the CR link example, *TxPower* is a resource variable, *ModulationCoding* is a productivity variable that uses the resource variable to create link capacity. *FrameSize* is a tuning variable. Once the control variables are categorized as above, we use the following sequence for optimization. Increase resource supply to the system to the maximum possible extent, subsequently increase capacity, followed by consumption of capacity by increasing productivity and finally fine tuning to improve the efficiency. The approach here is to execute a sequential plan involving resource provision, capacity/productivity creation, consummation and finally removal of surplus capacity & resources.

Algorithm 4.1 - QOpt - QPN Model based Sequential Optimization

```
U={u_i} – control variables
U_0 – Initial value
Z – Goal Variable
S={S_j} - Constraints
U ← U_0
Initialise()
while not Optimized(Z) do
        u ← SelectControlVar(U)
        $\delta(u,Z)$ ← PropagateObservation('+',u)
        if $\delta(u,Z) \neq$ '?'
                (u*,Z*) ← ApplyControl (u, $\delta(u,Z)$)
                if ConstraintViolation() or not GoalImproved()
                   (u*,Z*) ←RetractControl((u, $\delta(u,Z)$)
                end if
        else
                δ(u) ←ChooseMove(u)
                (u*,Z*) ← ApplyControl (u, $\delta(u,Z)$)
        (u*,Z*) ←HillClimb(u, δ(u), Z)
        end if
end do
```

The QPN sign propagation algorithm *PropagateObservation()* finds the inferred change that to be made on the selected control variable to improve the goal function. If the inference returns a non-ambiguous sign with respect to a control variable change (increase/decrease), and then move the control in that direction so as to improve the goal function. In case the inference results in an ambiguous sign (‘?’), then the respective control variable is given a perturbation (either increase or decrease) to identify the direction of desired movement for goal improvement. Continue moving the control variable in the same direction as long as the goal variable shows improvement. Basically this is nothing but a local search such as hill climbing. This local search can be turned off if there is not enough time available for the optimization process due to real-time constraints.

We use this optimization approach for a link optimization problem to demonstrate the use of QPN model and inference.

4.4 Example- Link Adaptation in a Cognitive Radio Link

In this simulation scenario (Figure 4.2) two cognitive radio nodes CR_1 and CR_2 are communicating the presence of a Primary transmitter P. The CR nodes are accessing the channel as secondary using spectrum sensing. CR_2 is moving in a specified trajectory. Here the goal is to maximize throughput of the link with a constraint on interference to primary. The QPN model used here is the same as the one discussed in chapter 3. Goal variable is *Throughput* and control variables are *Modulation*, *TxPower*, *Framesize* and *Spectrum sense duration*. We use the following analytical model to simulate the scenario.

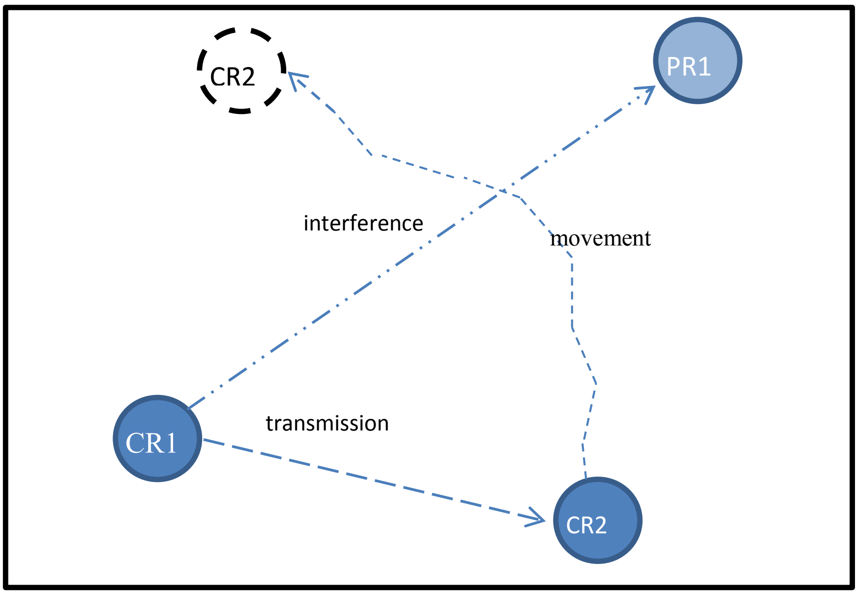


Figure 4.2 Cognitive Link Adaptation Scenario

4.4.1 Radio Link Model

We use the following CR model to simulate the CR link for studying the performance of the QPN based link optimization described above. The CR link works as a secondary in presence of a primary transmitter, Physical layer is an adaptation from WiMax (IEEE 802.16d) and the medium access is done through spectrum sensing.

The parameters used in the model are P_t = Transmit Power dBm; m = Modulation scheme; f = Frame Size , bytes; ρ – SNR; ρ_{th}, ρ_m– SNR Threshold, SNR margin; d – distance between transmitter and receiver; d_0- Reference distance; γ – Burst rate; μ – Transmit duty ratio; ϵ_b – Bit error rate; ϵ_f – Frame error rate; f – Frame size; t_s – Spectrum sensing duration; n_D – Number of samples used for sensing; n_P - number of samples duration between adjacent sensing; P_{FA}- Probability of False Alarm; P_D – Probability of Detection; t_d – Time elapsed between sensing; T – Throughput; φ – Threshold for primary detection; I_{sp}- Interference to Primary (referenced at Tx of secondary) ; β_p = SNR of Primary transmission at Secondary Receiver.

The following empirical propagation model is adapted from IEEE 802.16d standards annexure.

$$\rho = P_t - (83.3 + 20\log\left(\frac{d}{d_0}\right) + 104; \rho_m = \rho - \rho_{th} ;$$

$$\epsilon_b = \begin{cases} 0.5e^{-1.5316*\rho_m + 0.036265*{\rho_m}^2}, \mathrm{\rho m} > 0 \\ 1, otherwise \end{cases}$$

$$\epsilon_f = 1 - (1 - \epsilon_b)^f$$

$$T = \gamma\mu(1 - \frac{(n_P - n_D)}{(n_P)})\ (1\text{-}\epsilon_f)(1 - P_{FA})$$

Where the probability of false alarm P_{FA} and probability of detection P_D of primary using spectrum sensing is computed as [2]

$$P_{FA} = Q\left(\frac{\varphi}{2\sqrt{n_D}\sigma_w^2} - \sqrt{n_D}\right)$$

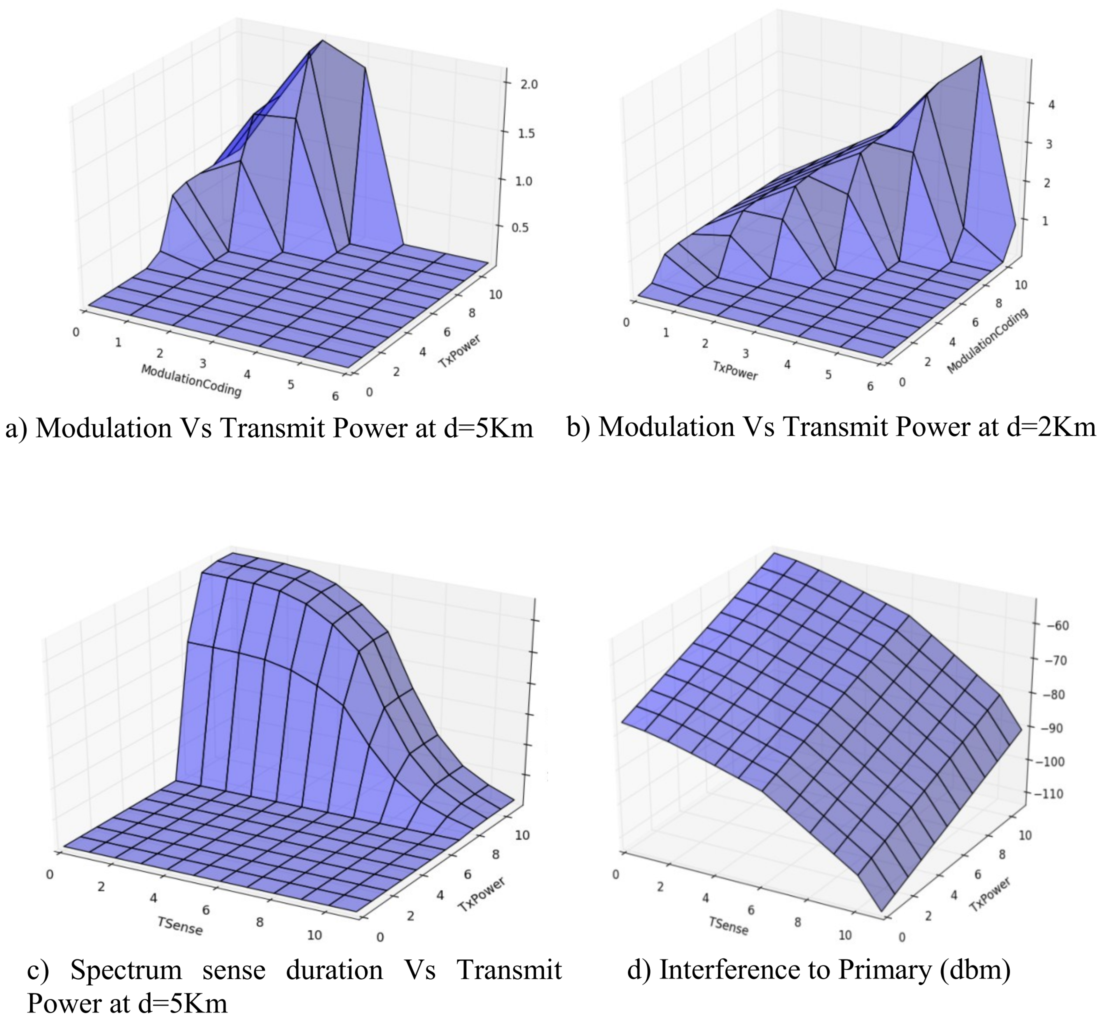
$$P_D = Q\left(\frac{1}{\beta_p}(Q^{-1}(P_{FA}) - \sqrt{n_D}\beta_p)\right)$$

Interference to primary is caused by missed detections and is computed as

$$I_{sp} = (1 - P_D)P_t^{max}$$

Here the control variables are U = { P_t ,m, f, n_D}. Domains of the control variables are ordered and as follows:

m = {BPSK-1/2, QPSK-1/2, QPSK-3/4, 16QAM-1/2, 16QAM-3/4, 64QAM2/3, 64QAM-3/4}
P_t = {2dbm to 24dbm in steps of 2}
f = {200 to 2000 bytes in steps of 200}
$\mathrm{n_D}$ = {2,4,6,8,10,12,14,16,18,20} samples



a) Modulation Vs Transmit Power at d=5Km

b) Modulation Vs Transmit Power at d=2Km

c) Spectrum sense duration Vs Transmit Power at d=5Km

d) Interference to Primary (dbm)

Figure 4.3 Response Surface of various control variables of the link model

The response surfaces of some of the important variables are shown in Figure 4.3 to have a visualization of the reference link model.

4.4.2 Simulation Results

We implemented a prototype simulation system as shown in Figure 4.1 using Python language. The simplified analytical model of a radio link described in has been used to simulate the target radio link in presence of a primary transmission. The distance between transmitter and receiver is varied over a predefined path to generate a dynamic scenario. Transmission duty ratio of Primary is assumed to be stationary. The goal is to maximize secondary throughput with a constraint on interference to primary. The cognitive engine is configured to perform runtime throughput optimization based on the 'QOpt' algorithm described in *Algorithm* 4.1. The optimization engine is triggered periodically in every 10 seconds to reduce the computation overhead compared to that in a continuous operation. Figure 4.4 shows the trajectory of the secondary link throughput achieved by the optimization engine. It shows the average of 50 runs. The total simulation time is 170 seconds and the total data transfer during the time is 630 Mbytes. This is compared with the reference optimum trajectory that is determined from an exhaustive search. Though the exhaustive search option is impracticable in real application it provides as a reference bound for the performance. The reference data transfer is 697 Mbytes and the QPN sequential optimization process shows a performance figure of 90.6% of that reported by exhaustive search.

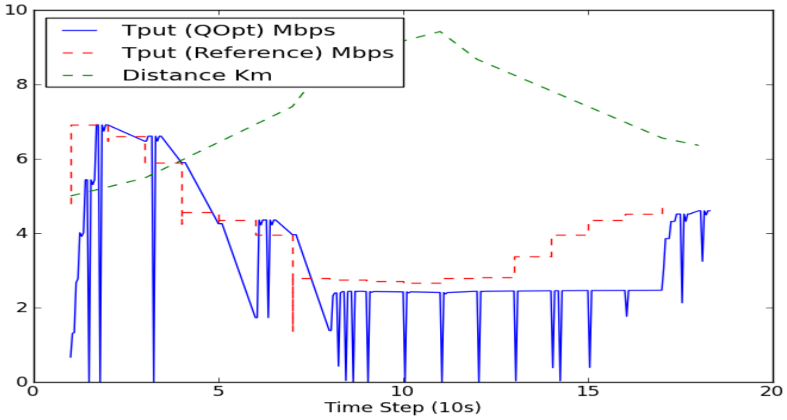


Figure 4.4 Performance in Link Adaptation using QPN sequential Inference. The solid line indicates trajectory for the QPN based optimization(QOpt)

From these two simulation experiments we observe that the QPN model based optimization approach performs reasonably good (better than 90% of the reference bound) with a low computational complexity (It should be noted that a QPN sign propagation require only a few sign multiplication over a chain for inferencing)

4.5 Adaptive FEC for enhancing the Robustness

Here we extend the scope of the above cognitive link model by introducing the use of application level *Forward Error Correction (FEC)* with erasure coding for improving the robustness of the communication link. Motivation for this approach is that in addition or instead of the traditional retransmission scheme erasure coding can be used for reducing the overall packet drops in the communication channel.

Here we use an *(n; k; e)* erasure coding scheme where k packets are encoded to form n packets with an addition of *e* repair packets. If the receiver receives correctly at least *(n-e)* packets of any type, i.e. data or repair, out of n transmitted packets, it can reconstruct the original k data packets. Here, *e* is the erasure correction capability of the code. When the number of erasures is more than e, the recovery is not possible. We consider the use of an optimal erasure correcting code where *e = (n - k).* Figure 4.5 shows the modified QPN model incorporating the FEC scheme.

We enhance the simulation model of the cognitive link described in earlier section with the incorporation of the effect of *FEC* on *Tpt* and *QoS.* We express QoS as a simple utility function of T_{pt} and PktL (packet loss) $QoS = 0.5\left(\frac{Tpt}{Tpt_{max}}(1 - PktL)\right)$. In the following specific simulation study we use the FEC block size as 11 (packets) and *e* takes the values {0, 2,3,4,5}. A detailed discussion on this simulation study is available in our paper [7].

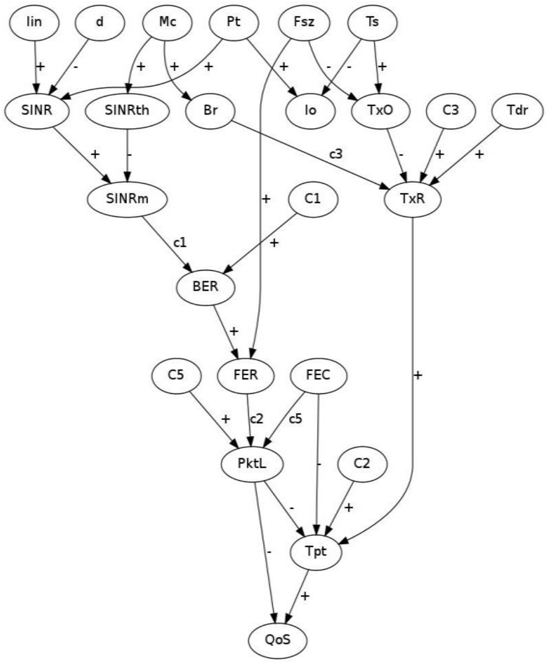


Figure 4.5 QPN Model Enhanced with the FEC Scheme

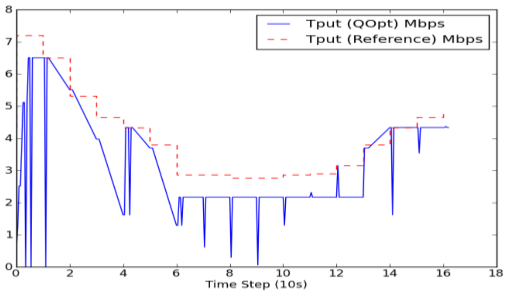


Figure 4.6 Performance in Link Adaptation with FEC using sequential QPN Inference.

We use the same simulation setup described in the previous section to simulate the additional effect of FEC in the overall optimization of link QoS. Figure 4.6 shows the throughput performance along the transmission duration when sequential optimization approach is used. As in the previous simulation study the reference curve shows the upper bound obtained from an exhaustive search. The ideal reference performance bound for the total data transfer is 695 Mb and packet loss is 1e-3 %. The corresponding figures obtained for QPN approach is 568.9 Mbytes (81.8%) at an average packet loss (*PktL)* of 2.3%. Overall the performance of the optimization engine has been in the similar lines as that of the previous case.

4.6 QPN and Monotonicity Analysis

In this section we propose to integrate a few key results from monotonicity analysis theory of signed graphs with the QPN based model framework for optimization.

Identification of monotonicity in system behavior and exploiting it for elegant algorithms for optimization and control is rewarding because they have nice properties of "order" in their dynamical behavior. Monotonic systems do not admit stable periodic orbits or chaotic behavior in a control loop. More importantly, for strongly monotone systems, Hirsch theorem states that almost every bounded solution converges to the set of equilibriums [3]. Both monotonicity and strong monotonicity admit graphical characterizations in terms of the influence graphs and so also in QPN structure. A system is monotone when all undirected cycles of its influence graph have positive sign (i.e., have an even number of negative edges). For an irreducible system it is strongly monotone when the same property holds for directed cycles [4]. From this it can be observed that a sub-graph resulting in a trade-off relation in QPN is not monotonic. Therefore one way to approach for resolving conflicts is by making the undirected cycle positive. This is also termed as balancing the associated sign graph. One possibility is to break the cycle by removing an arc that is comparatively insignificant. We have used this approach in the context specific sign propagation scheme described earlier. Another direction is to explore if the sign of an edge can be changed. A weak influence sign could be reversed if there is no substantial impact of this change.

Monotonicity analysis has been used to simplify the solution or completely solve in closed form design (optimization) problems which have otherwise required extensive numerical computation [5]. Qualitative optimization has been used to decompose an optimization problem into a reasonable number of smaller sub-problems which are readily solved. By analyzing the solutions of these smaller problems, the solution to the original problem can be identified.

The optimization engine in the CE has access to the policy database to deduce the goal variable and associated constraints for the optimization task in hand. Before attempting the optimization process, the optimization engine need to check the problem is well constrained. Towards that we propose to use the following monotonicity principles to analyze the QPN structure and associated constraints [5].

Monotonicity Principle 1 : In a well–constrained minimizing objective function every increasing (decreasing) variable is bounded below (above) by at least one active constraint.

Monotonicity Principle 2: Every strictly monotonic nonobjective variable in a well–bounded problem is either,

a) irrelevant and can be deleted from the problem together with all (inactive) constraints in which it occurs, or

b) relevant and bounded by two active constraints, one from above and one from below

Maximal Activity Principle: The number of independent constraints active in any subset of constraints should not exceed the number of variables on which these constraints depend.

Thus, a problem is said to be well constrained if it satisfies the Monotonicity and Maximal Activity Principles stated above.

By considering a sequential optimization approach of QOpt algorithm, the optimization engine picks up each control variable and performs an inference using QPN model to know which direction the variable to be moved towards optimizing the goal function. Assuming the monotonic property of the path to the goal variable, the control is continued in that direction. According to the monotonicity principle 1, there should be a bounding constraint in the opposite direction of the movement. There should be at least one such constraint involving any of the variables in the chain from control variable to goal variable. In the link model example, the effect of *TxPower* increase on *Throughput* is non-decreasing and its effect on interference is also positive. In this case the increase of *TxPower* is desirable to improve goal function, but is limited by the constraint specified on Interference. So we can say the *TxPower* control is well constrained.

According to principle 2 we need not consider constraints that do not include any of the variables in the path(s) to the goal node. Applying maximal activity principle, we can conclude that the number of independent constraints considered need not exceed the number of paths to the goal node. The optimization engine should verify these requirements on the QPN model before commencing the optimization process.

We introduce an *Active-Constraints-Check* module in the QOpt algorithm which checks if the QPN model is well constrained according to the principles of monotonicity as described above. This is included as part of the *Initialize* function in the algorithm. In this function we take each control variable and relevant constraints to their paths to the goal variable. Apply monotonicity principles to select the active constraints. If there are no active constraints found, then a default constraint is included which limits the value of the control variable to its discrete domain.

Considering the fact that the QPN model encodes only a qualitative knowledge about, the near optimality of the result needs to be appreciated. However there are some immediate concerns too. Since it uses a sequential optimization approach, the time taken for completing the optimization could be a concern. Another issue is the spurious reconfigurations due to random moves selected when QPN inference turns out to be ambiguous (‘?’). In the cognitive engine architecture discussion we proposed to address this problem by a decision database supported by a case based learning scheme. This is described in the following section.

4.7 Avoiding spurious configurations during on-line adaptation

One of the artifacts that can be observed in Figure 4.4 is the periodic downward spikes indicating the fall of objective function (*Throughput*). This is because of a few unavoidable spurious configurations during the solution search process during online optimization. We propose to reduce or eliminate such spurious configurations using a learning mechanism which will remember such situations and avoid any future repetition of those undesirable configurations.

Here we propose a Case based Learning (CBL) approach to create and manage a database of situations (contexts) and configurations that caused drastic fall of the objective function value, so that such bad configurations can be avoided in future. For that a compact representation of the situation is needed.

4.6.1 Compact Representation of a *Situation*

Situation is defined as a stable state of the system at a particular time/event of interest. It can be represented by the system state variables and their stable state values corresponding to the *situation*. Once a reference *situation* is stored in the data base as a reference point, a similar *situation* can be identified by matching the new situation with the reference situation. An appropriate similarity function needs to be used for this matching process. Identifying and storing reference *situations* are part of the learning. This is the central idea behind case based learning.

We propose the following Lemma to capture a compact representation of the *situation* by selecting a set of relevant parameters.

Lemma: *Minimal set of state variables χ_i required to define a situation S, relevant to a control variable u_i and goal variable Z is $\chi_i^S = \{ u_i, \delta_i, z, J(u_i,Z), C_j\}$ where δ_i : direction of change of u_i ; z: Value of goal variable Z; $J(u_i,Z)$ is values of variable nodes in the paths from u_i to Z that have an incident chain from a control variable or external variable, C_j is the set of conditional variables associated with the paths (u_i,Z).*

In the QPN structure the situation χ_i^S is the set of variables those are sufficient to represent the situation so that the influence of u_i on the goal variable can be inferred. This Lemma is used to identify a compact set of state variables χ_i that are required to represent the state change due to a specific control variable change. This avoids the need for storing the state information corresponding to the whole context space and improves storage efficiency of the case database.

4.6.2 Case Based Learning of Spurious configurations

We define the following terms:

Spurious configuration : Indicates a configuration change of u_i in the direction of δ_i which caused the objective function Z to drop below a threshold low value , $Z < Z_{low}$.

Reference situation : Let $\chi_i^r(\delta_{i,})$ be the reference situation that lead to a spurious configuration when u_i changed in the direction of δ_i,

Similarity function : The similarity function is used to check if a candidate situation χ_i^c falls in the neighborhood of any of the bad situations stored in X^R. That is to find out any cases such that $(|\chi_i^r - \chi_i^c|) < \varepsilon$ where $\chi_i^r \in X^r$

The similarity function *similar*(χ_i^r, χ_i^c) is Boolean and will return the result of similarity check.

Learning function: we define the function for learning as *Learn*(u_i, δ_i, z_c): if a *Spurious configuration* on u_i is detected, that results in a drastic degradation of objective function from z_p to z_c, then store the *situation* corresponding this state as a reference *situation*.

Algorithm 4.2 presents the pseudo code for the detection and avoidance of spurious configurations as part of the enhancement to the QOpt algorithm (Algorithm 4.1). The *ApplyControl* function in Algorithm 4.1 is replaced with the enhanced version incorporating the spurious configuration avoidance part. In this enhancement we first perform the check if the selected configuration is similar to any entries in the spurious configuration database. If it is present, then the configuration is skipped. Otherwise we go ahead with the reconfiguration. Subsequently we observe the effect of this new configuration and see if it has generated an anomalous system response. In case a new spurious configuration is detected, then the data base is updated using the *Learn* function.

ALGORITHM 4.2 DETECTION AND AVOIDANCE OF SPURIOUS CONFIGURATION

```
$D_b$ -  Case Database
$u_i$ -  Control variable
Z –  Goal Variable
Zc – Current value of Z
$X_i^c$ – Current Situation

procedure ApplyControlwithLearning($D_b$, $u_i$, $\delta(u_i,Z)$, $z_c$)
    $X_i^c \leftarrow$ CurrentSituation($u_i$, $\delta_i$, $z_c$)
    if not Similar ($D_b$, $X_i^c$) then
        ApplyControl ( $u_i$ , $\delta(u_i,Z)$)
        If Spurious ( $u_i$ , $\delta(u_i,Z)$) then
            $D_b \leftarrow$ Learn ($X_i^c$)
    end
end
```

4.6.3 Simulation Results

The CBL based algorithm for avoidance of spurious configurations described above has been incorporated in the link optimization algorithm and applied it on the CR link optimization problem described in 4.4 . Figure 4.7 presents the trajectory of the optimum throughput with the CBL algorithm. It can be observed that there are some spurious configurations in the first half (as indicated by sharp downward spikes) there are no major anomalous configurations towards the later part of the trace. This shows the effectiveness of the algorithm that is highly beneficial for practical optimization scenarios with CE.

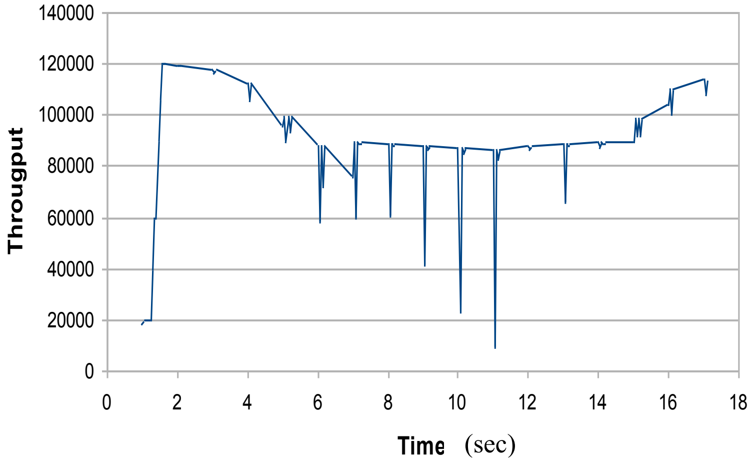


Figure 4.7 Profile of the link throughput with case based learning of spurious configurations (Time in seconds)

In the present optimization scheme, when the QPN inference gives an ambiguous result ('?') we do a local hill climbing search to refine a configuration. This is based on the assumption that the influence relation δ(u,Z) is monotonic in their individual paths to the goal variable. But due to the presence of noise or a deviation on this assumption, the hill climbing algorithm may not perform well due to local minima problem. For such cases considering a global search algorithm would be beneficial however for an online optimization problem the number of steps required for the search is of a high concern. In the next section we experiment the use of simulated annealing algorithm within the QOpt framework and analyse its performance.

4.6.4 Simulated Annealing enabled with Influence graphs

Simulated Annealing (SA) is a very popular meta-heuristic algorithm that is known to be effective for optimizations as a black-box approach. Application of SA for wireless network optimization has been explored by a few researchers and a recent work is available in [6] where the authors use several enhancements to improve the search efficiency. The basic algorithm is simple and problem specific refinements are possible to enhance its performance. The algorithm steps are as follows: Starting from a random position in the search space, the next point is chosen at an arbitrary location within the distance of a jump proportional to a temperature parameter. Initially the temperature is set to maximum. Then the new selection may be accepted with a probability even if it is an inferior solution. The probability of acceptance could be proportional to the improvement in utility and temperature. Every iteration of the search also results a temperature decrease. The algorithm stops when the temperature drops below a threshold or maximum number of iterations reached. Further there will be a reheating process in which the controlled cooling is performed iteratively.

We have used the SA algorithm to the same link optimization problem described earlier. The control variables such as *Modulation Coding*, *Tx Power* and *Framesize* were assigned values selected from their neighborhood in every iteration. The size of the neighborhood was controlled by a decreasing temperature function. To select the neighborhood solution space we have used the influence information from signed graph representation of the link. The solution giving a better utility, that is *Throughput* here , was selected as the initial choice for next iteration. The solution giving inferior utility is selected with a probability determined by the difference in *throughput* values and a temperature variable. The iterations are stopped once the temperature reaches a minimum or maximum number of iterations. On multiple runs it was noted that it took arbitrarily different number of iterations to reach the optimum and in some cases even with 100 iterations an optimum solution was not reached.

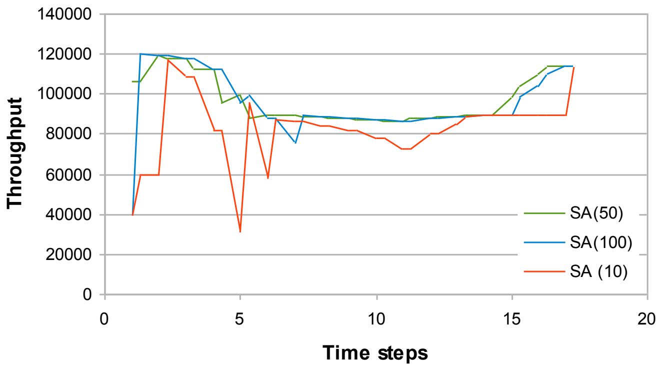


Figure 4.8 Profile of the link throughput with simulated annealing

The Figure 4.8 presents the trajectory of the throughput for different number of maximum iterations settings (10, 50 and 100). It can be observed that there is progressive improvement in tracking the optimal configuration as the number of maximum iterations is increased. With about 100 iterations the optimal solution almost matches with the results of QPN based optimization algorithm. This shows that there is no good return on investment of using SA along with QPN in the QOpt. So we do not consider SA to be integrated QOpt in the proposed Cognitive Engine.

4.8 Summary

In this chapter we proposed the use of QPN Model based inference to drive self-optimization in the Cognitive Engine. A sequential optimization algorithm "QOpt" is used to demonstrate its use in a simple cognitive link adaptation use case. The performance is highly satisfactory with respect to the expectation from a qualitative model incorporating limited information about the system. Integration of Monotonicity analysis with the QPN structure is a novel step to ensure that a well constrained model is available for the QOpt algorithm for a converging optimization. In the next chapter we extend this methodology to build a network level modeling and optimization.

References

[1] Xiang-Kunli, Shi-Zhong Liao, "Inference in QPNS with Multiple Observations", Proceedings of the Eighth International Conference on machine learning and Cybernetics, Baoding, 12-15, July 2009

[2] Gosan Noh and Daesik Hong (2009). "Achievable Throughput Comparison of Sensing-Based and Interference-Constrained Transmissions in Cognitive Radio Networks", Book Chapter: Cognitive Radio Systems, Wei Wang (Ed.), ISBN: 978-953-307-021-6, InTech Publishers.

[3] Hirsch, M. W. (1983). "Differential equations and convergence almost everywhere in strongly monotone semiflows", In J. Smoller (Ed.), *Nonlinear partial differential equations* (pp. 267–285). Providence, RI: AMS

[4] G. I. N Soranzo, F Ramezani and C. Altafini, "Graph-theoretical decompositions of large-scale biological networks," Automatica, pp. 1–17, 2010.

[5] Papalambros, P. and D.J. Wilde (1988), Principles of Optimal Design Modeling and Computation, Cambridge University Press, New York, 1988

[6] Elena Meshkova, Janne Riihijärvi, Andreas Achtzehn, Petri Mähönen, "Exploring simulated annealing and graphical models for optimization in cognitive wireless networks", Proceeding, GLOBECOM'09 Proceedings of the 28th IEEE conference on Global telecommunications, IEEE Press Piscataway, NJ, USA ©2009

[7] Balamuralidhar P, "Exploring Qualitative Probabilistic Networks for Knowledge Modeling in Cognitive Wireless Networks", IWCMC-2013

5

Qualitative Modelling and Inference for Network Scenarios

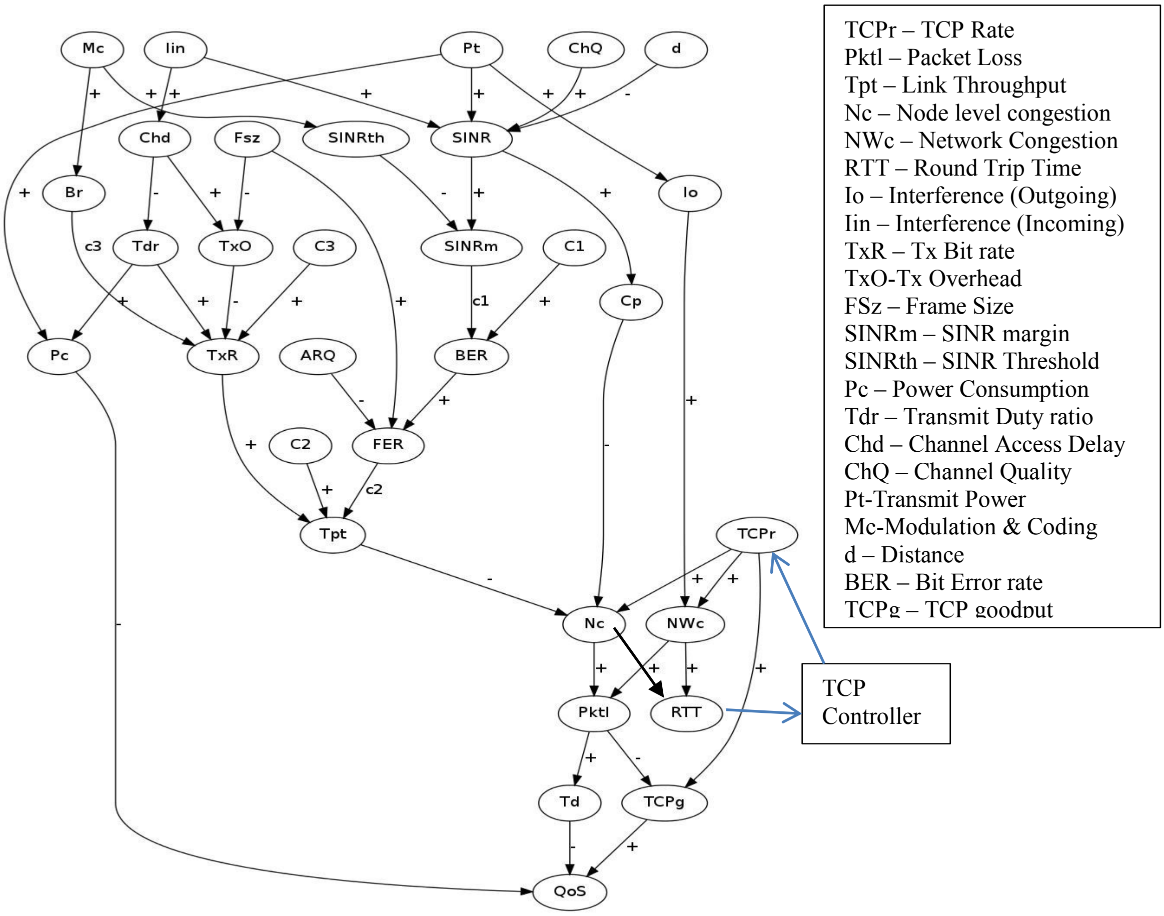
In this chapter we develop and analyze QPN models for capturing behaviors at network layer and above. The QPN methodology proposed in the previous chapter is applied here also to prepare the model suitable for inference. To motivate the analysis we focus on a practical problem of TCP congestion management in an adhoc wireless network scenario. It is shown that a joint congestion management involving TCP and wireless link adaptation strategy can be inferred from the QPN model and the cognitive engine can operationalize it. A Simulation study is conducted and results are presented to support the observations.

QPN Models enhanced with the semi-qualitative extensions have been shown to be useful in modeling the link layer behaviors of wireless networks in the previous chapter. The link layer challenges are very high considering the fact that it is the layer sourcing most of the uncertainty introduced in wireless communications. At network layer the surprise factor comes from the coupled effect of mobility, routing and traffic generated by the applications.

In the following section we enhance the link level QPN model discussed in previous chapter with network level variables. Subsequently we analyze the QPN for issues and challenges of using it for cognitive reasoning.

5.1 QPN Model including Network Layer

One significant addition to the link layer model is the TCP behavior. TCP has good mechanisms incorporated for network congestion control and congestion avoidance. They are specifically designed for wired networks where the links are reliable and capacities are not fluctuating. In the event of a NW Congestion (*NWc*) there will be increase in packet loss (*Pktl*) and round trip time (*RTT*). In TCP, congestion is said to have occurred when the sender receives three duplicate acknowledgments (dupacks) or when a timeout occurs. The TCP congestion control techniques are divided into three broad categories, viz., (i) window based, (ii) equation based, and (iii) rate based. In this discussion we focus on window based schemes because that is more suitable for wireless scenarios. *An Adaptive Window Management technique* is used in the window based congestion control, in which increase and decrease of congestion window (*cwnd*) is based on packet drops and dupacks. This increase and decrease of *cwnds* are based on the principle of Adaptive Increase and Multiplicative Decrease (AIMD). There are many variants of TCP such as Tahoe [1], Reno [2], [3] and Vegas[4] using variants of window based congestion control. In effect when packet loss *(Pktl)* and increased *RTT* is detected TCP adjusts its rate (*TCPr*) by controlling the congestion window (congestion control and congestion avoidance schemes). A decrease in *TCPr* leads to a decrease in network congestion (*NWc*) as well as node level congestion (*Nc*). Node level congestion is influenced by the link capacity and when there is an increased capacity due to good SINR the probability of Node congestion reduces. At the same time an increase in throughput reduces *Nc* assuming the input data rate is constant. Nc and NWc introduces packet loss and that will adversely affect the TCP goodput (*TCPg*). When there is a congestion detected, the TCP congestion controller adjusts TCPr and this forms a control loop. A QoS variable is included which depends on power consumption (*Pc*) and Delay (*Td*). Power consumption has a dependency on transmit power and transmit duty ratio (*Tdr*).



TCPr – TCP Rate
Pktl – Packet Loss
Tpt – Link Throughput
Nc – Node level congestion
NWc – Network Congestion
RTT – Round Trip Time
Io – Interference (Outgoing)
Iin – Interference (Incoming)
TxR – Tx Bit rate
TxO-Tx Overhead
FSz – Frame Size
SINRm – SINR margin
SINRth – SINR Threshold
Pc – Power Consumption
Tdr – Transmit Duty ratio
Chd – Channel Access Delay
ChQ – Channel Quality
Pt-Transmit Power
Mc-Modulation & Coding
d – Distance
BER – Bit Error rate
TCPg – TCP goodput

Figure 5.1 QPN Model depicting the network behavior

A QPN model of the network behavior described above is depicted in Figure 5.1

5.1.1 Analysis

Following the line of QPN based methodology described for link level optimization in the previous sections; same approach can be extended to this network model also. It involves finding of new undirected cycles where trade-offs are present and attempt to resolve them through enhanced signs or context based edge switching. Once the trade-off issues are resolved to the extent possible, the model can be used in the inference engine for driving various network level adaptations as envisaged within the framework of cognitive networking.

First is to lookout for the subgraphs that have trade-off issues. This is done by identifying subgraphs having their undirected cycles negative. Alternately sign propagation from head node to tail node will result in an ambiguous sign ('?') for at least one node. We analyse the graph in Figure 5.1 and observe that there is a new trade-off relation between *Pt* and *Pktl*, in addition to the ones we have discussed and mitigated in previous chapter. There are two chains in the relation. In one chain (P_t-SINR-SINRm-BER-FER-Pktl) the net influence is '+' while in the other chain (P_t-I_o-NWc-Pktl) is '-'. This conveys that an increase in transmit power can reduce packet loss if link capacity is limited. On the other side an increase in transmit power increases interference to other nodes and can cause increased network congestion and thus the packet loss. In the following section we look at closely on how the QPN model at a single link level integrated into a multi-node network.

Let us consider a small adhoc network consisting of three CDMA communication nodes operating in a common radio environment. We use a simplified influence diagram for the node with each node consisting of only a few variables, (P_t,I_{in},I_o,SINR,Pktl,Nc,TCPr), that depicts the salient TCP behavior. The influence diagram representing the interaction of all the three nodes is shown in Figure 5.2. The subscripts of the variables indicate the node id.

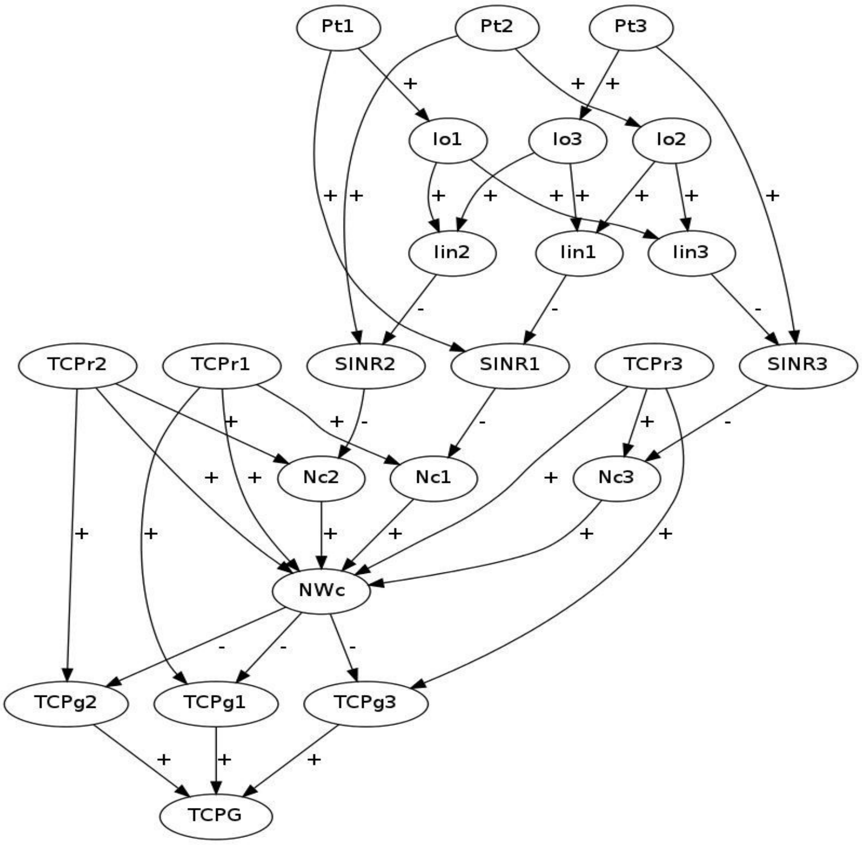


Figure 5.2 Mutual influences between three nodes in a radio network

Here it can be seen that at a node level the TCP controller controls the TCP rate based on the perceived network congestion indicated by packet loss. Following inferences are made using QPN sign propagation.

Table 5.1 List of observations and inferences from Network QPN model

Observation	Inference	Comments

Pt1 ('+')	SINR1('+'), Nc1(-), Io1('+'), NWc('+')
	SINR2('-'), Nc2('+')
	SINR3('-'), Nc3('+')
		Increase in *Pt* of a node decreases the probability of Node congestion and increases network congestion *NWc*.
		Increase in interference creates an increased probability of node congestion for other nodes

TCPr1('+')	Nc('+'),NWc('+'),TCPg('?')
	TCPg2('-'), TCPg3('-')
		When TCP rate is increased there is a probability of increase in *Nc* and *NWc*. While the direct influence of *TCPr* on *TCPg* is positive, its indirect influence through *NWc* is negative and a trade-off relation is created.
		The increased *NWc* will create negative influence for *TCPg* for other nodes.

In the above observations we propose to use the context specific signs and constraints for the reduction of trade-offs and reduction of complexity. In the influence chain of *Pt1,* introducing a limit $Pt1 < Pt1_{max}$, we conditionally remove the influence chain corresponding to the interference *Io1*. With this we observe that the influence of *Pt* on *Nc* is monotonic (at least 'non-increasing').

Looking at the influence of *TCPr1*, we note that it has an ambiguous relation with *TCPg* involving *Nc* and *NWc* in two trade-offs leading to a rate optimization loop. TCP Congestion controller does exactly that. When congestion is detected it reduces the *TCPr*. The mechanism used by TCP to detect congestion (*Pktl* ,*RTT* etc) does not distinguish between the sources of congestion (*Nc* or *NWc*). From the Figure 5.1 and Figure 5.2 it is inferred that TCP need not adapt TCPr when there is a node congestion and it should be primarily handled by link adaptation.

The inferences discussed above also reflect the TCP controller's inability to distinguish between network congestion and node congestion due to wireless link capacity limitation. If δ(SINR) is made '0', that is stable SINR without any change, then *Pktl* will only depend on NWc and thus it reflect the change in network congestion. Therefore an adaptation of *Pt* to make SINR stable will make the TCP congestion control to function as desired.

5.1.2 QPN decomposition through Graph Partitioning

From the analysis of QPN structure in Figure 5.1 it is observed that there are two adaptation loops (controllers) operating over this model. One is the link optimization and the other is TCP congestion control. From a graph theory perspective it is interesting to note that the QPN graph can be decomposed into two by applying a *min-cut* with minimum coupling between the two controllers.

In graph theory, a *cut* is a partition of the vertices of a graph G(V,E) into two disjoint subsets C=(S,T). The *cut-set* of the cut is the set of edges *Ec* whose end points are in different subsets of the partition. That is C=(S,T) is the set $\{(u,v) \in E | u \in S, v \in T\}$. Edges in *Ec* are said to be *crossing* the cut if they are in its cut-set. The *cut* of a graph can sometimes refer to its *cut-set* instead of the partition. There are well known algorithms such as Kargers algorithm [5,6] for efficient solution of min-cut partitioning.

In the QPN model of Fig 5.1, it is found that the minimum *cut-set* C = (S,T) is obtained when S={I_o,C_p,Tpt,P_c} and T={*NWc,Nc,Qos*} such that the two controllers are separated in two partitions. Figure 5.3 presents the decomposition (the variables Qos and Pc are are not shown).

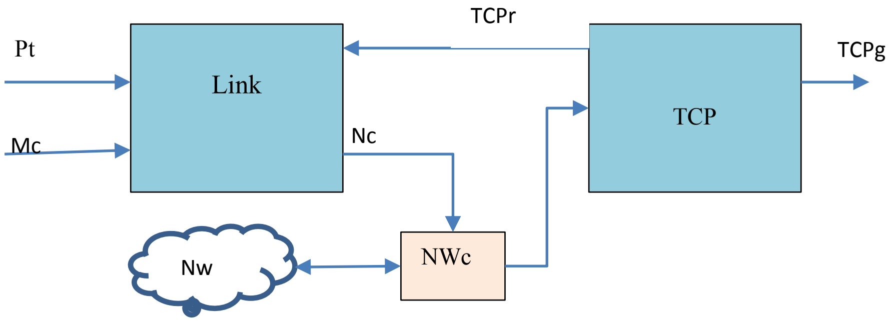


Figure 5.3 Decomposition of QPN model using graph partitioning

This joint power and congestion control has been already recognized as an approach to address this TCP behavior for wireless networks in the literature. We briefly give an overview of the joint TCP congestion control in the following section.

5.2 Related Works - Joint Congestion and Power Control in TCP

For wireless networks packet loss or/and delay can result due to time varying nature of the wireless channel in addition to congestion. Also, the link capacities are not fixed, rather depend upon the signal to interference and noise ratio (SINR) of the link. To address this problem, a joint congestion and power control technique for TCP Vegas is proposed in [1]. Further, a Joint Optimal Congestion control and Power control (JOCP) techniques to address congestion in wireless networks and to improve battery life-time through optimal use of transmission power is presented in [8][9]. JOCP is a cross-layer approach involving TCP and physical (PHY) layer, in which TCP layer performs window based flow control and PHY layer controls transmission power of wireless nodes.

The objective set for JOCP is to maximize the network throughput. The objective function of JOCP consists of two parts involving two optimization loops. First part is the TCP rate control taken care by TCP congestion control mechanism of TCP by increasing/decreasing the window size in each RTT for each flow. The second loop chooses the appropriate transmission power of wireless nodes based on a common variable λ relating both loops. This λ is termed as link price, which indicates the cost of transmission over the wireless link. It plays a significant role in determining the equilibrium window size and transmission power. In JOCP [8], each node requires information in-terms of a message $M_j(t) = \dfrac{\lambda_j(t) SINR_j(t)}{P_j(t) G_{jj}}$ from all other nodes to find this. This message passing for the distributed optimization is an overhead observed of the JOCP algorithm and details are given in [8].

5.3 Time Scales for Adaptations

Another aspect we consider here is the time scales of these two adaptations (congestion control and Power Control). The changes in wireless channel are much faster (milliseconds) where that in TCP networks is comparatively slower (seconds). TCP performs the adaptation step once in every RTT, while power control needs to complete the adaptation with one RTT.

This point out to the need for incorporating additional information related to the dynamics of external context variables in the QPN model and the CE would use this information to schedule various adaptation functions accordingly.

We propose to incorporate an additional attribute on *time scale* of change to the model variables. This information is brought into the model by two ways:

a) Apriori time scale information specified for all external variables (specified by experts)
b) A learning algorithm that estimates and update a dynamics model (a multi-state markov chain for example) linked to the above attribute.

We propose to incorporate this by modifying the *SelectControlVar(U)* function in QOpt (Algorithm 4.1) to make the selection of the control variables based on their timescale attribute.

5.4 Experimental Evaluation

In this section, we describe simulation experiments that have been performed to evaluate the QPN model based congestion control algorithm. We also compare the performance of this approach with that of JOCP approach. All the simulations have been conducted using implementations of cross-layer congestion control algorithm in MATLAB. We consider a CDMA ad-hoc network with six wireless nodes and two pairs of TCP flows (1-5) and (2-6) as in shown in Figure 5.4.

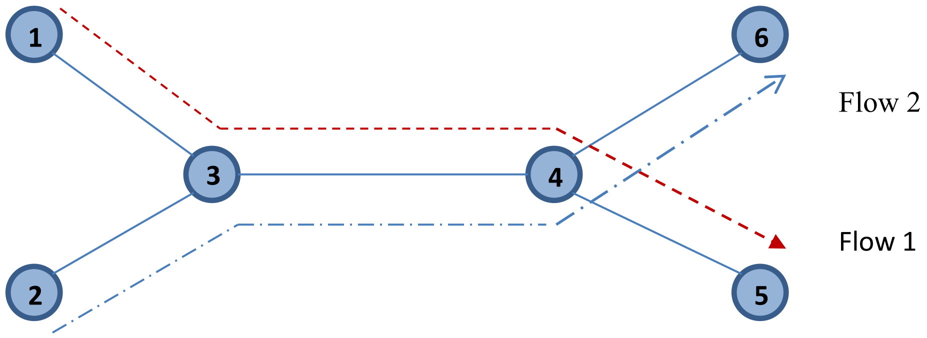


Figure 5-4 Topology of the adhoc network

In the QPN model based approach, all nodes in our simulation are capable of determining the transmission power and modulation index (in adaptive modulation) and in the JOCP approach, all nodes are capable of transmitting and receiving the cost λ_i. All six nodes run TCP NewReno agents and we set the TCP retransmission timeout to be 4 × RTT. We update RTT by using $RTT = \alpha RTT_{estimated} + (1-\alpha) RTT_{measured}$. We assume that α = 0.85 for our simulations. We assume fixed packet lengths of 1000 bits size. We assume that the time required for transmission in each of the segments (Figure 5.3) 1-3, 2-3, 3-4, 4-5 and 4-6 are same. We use fixed as well as adaptive modulation schemes. The path loss exponent due to distance is set as $\gamma = 4$. We consider Additive White Gaussian Noise (AWGN) with Power Spectral Density (PSD) $N_0 = 0.35$ (4.5 dB/Hz). We also simulate shadowing in our experiments. The shadowing is modeled as Log-normal with mean zero and standard deviation (σ) 8 dB. In each simulation run, the channel gain due to Log-normal shadowing is kept fixed for the entire duration of simulation. We also repeat the experiments with different Log-normal shadowing with σ = 4,6,8 and 12 dB. We assume channel reciprocity, i.e., both forward and reverse link gains are the same.

TCP NewReno is implemented in MATLAB. The data rate x_i is computed by using the relation $x_i(t) = \frac{cwnd_i(t)}{RTT_i}$ whereas cwnd_i and RTT_i are updated using TCP NewReno congestion control principle. We also consider step size of $\partial = 0.1$ for changing P_i.

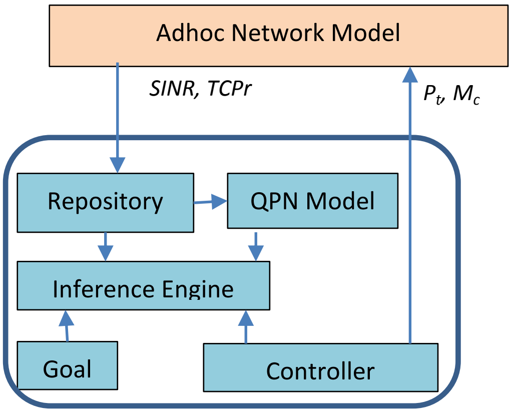


Figure 5.5 Simulation Setup

The simulation architecture is shown in Figure 5.5. Here we use the control variables { *Pt*, *Mc*}. Observation (Measurement) variables are *SINR* and *TCPr*. Goal of individual node is to maximize the rate *TCPr*. Following constraints are set on the QPN variables. {*Pt* < 15W, *BER* <1e-6}. Based on this constraint on BER the threshold for context dependent variable C1 is computed.

We use the domain of the control variables as P_i,= {8:0.25:15} , m_i = {2, 4, 8}. Following are the constraints identified so that it helps to balance the sub-graphs involving these control variables.

An adapted version of the QOpt algorithm (Algorithm 4.1) has been used here for the optimization engine. The system parameters used for simulations are presented in Table 5.1 (For QPN and JOCP). The value of each parameter observed has been averaged over 10 independent simulation runs.

Table 5.2 Summary of Simulation Parameters

Parameters	**Value**
α	0.85
$\text{cwnd}_{\text{initial}}$	3 packets
P_t	{8:0.25:15} watts
Mc	{2,4,8}

5.5 Simulation results

In Figure 5.6 and Figure 5.7, we plot one instance of the *cwnd* variation (corresponding to TCPr) of both flows using QPN based and JOCP approaches respectively. From these figures, we observe that *cwnd* variation over time is uniform for both the schemes, irrespective of fixed and adaptive modulation. Regarding the convergence the observation is that the QPN is somewhat slower than JOCP, however it is not too bad since it converges within 4-5 RTTs. Because of its slightly increased convergence time QPN based algorithm is more suitable for longer flows. Figure 5.8 and 5.9 shows the power control for the flows in JOCP and QOpt methods. We also observe that the average *cwnd* size of QPN approach with fixed modulation is 17.36 packets and with adaptive modulation is 14.06 packets. In addition, we found that the average *cwnd* size of JOCP approach with fixed modulation is 18.10 packets and with adaptive modulation is 14.10 packets. Though the average throughput is more in fixed modulation schemes, there are more packet drops resulting in drop in TCP good-put.

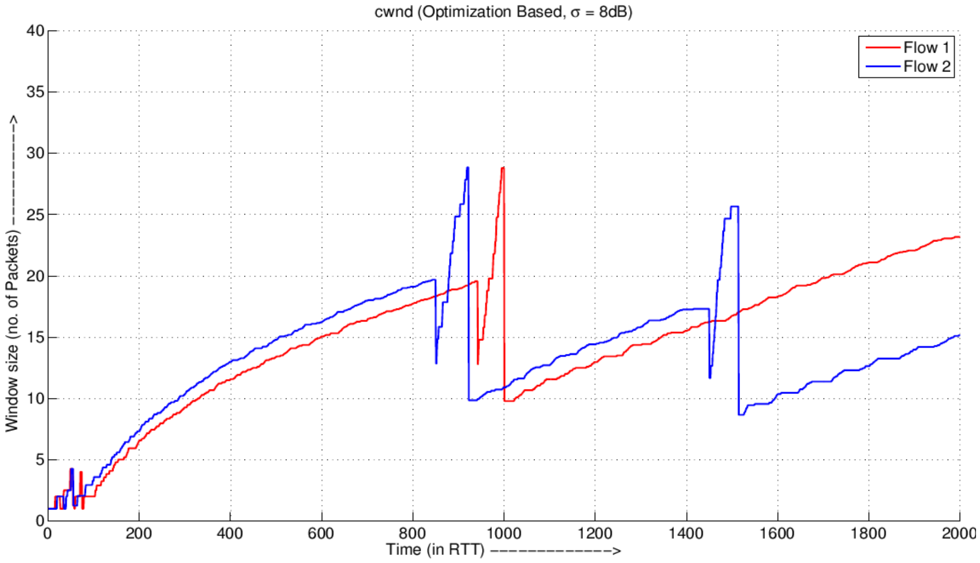


Figure 5.6. Variation cwnd size **for the two flows in JOCP**

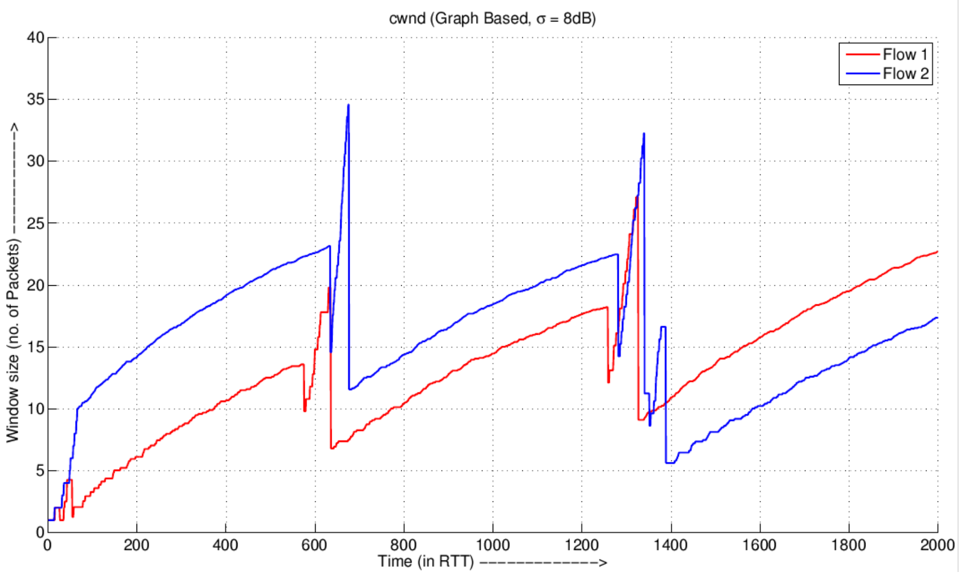


Figure 5.7 Variation cwnd size **for the two flows in QPN**

We also observe that the average transmission power in QPN approach is 11.34 watts, whereas it is 11.42 watts in JOCP approach; average transmission powers are close to each other.

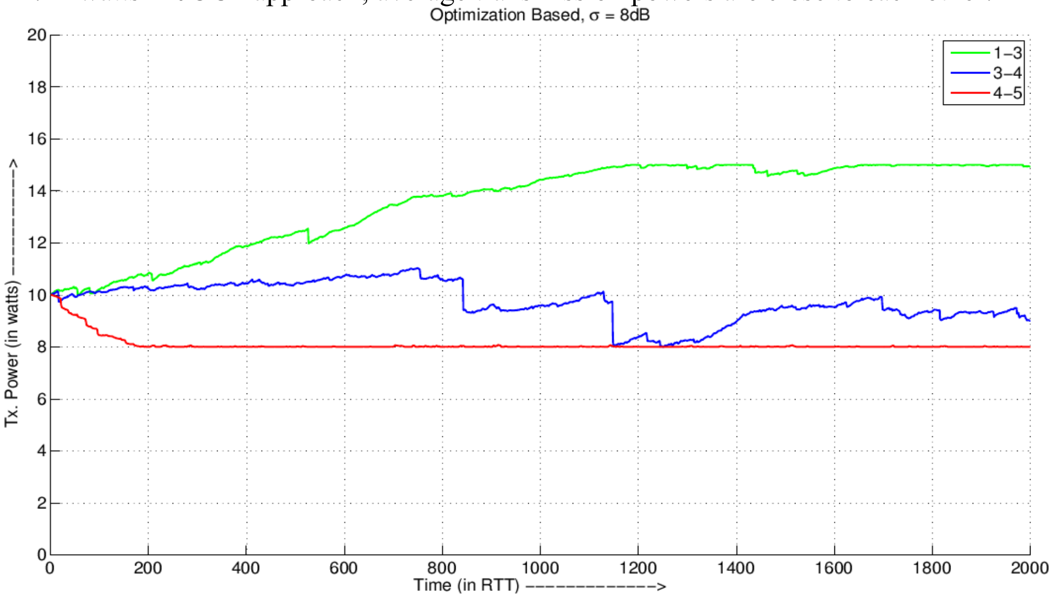


Figure 5.8 Variation of Tx power for three flows using JOCP approach

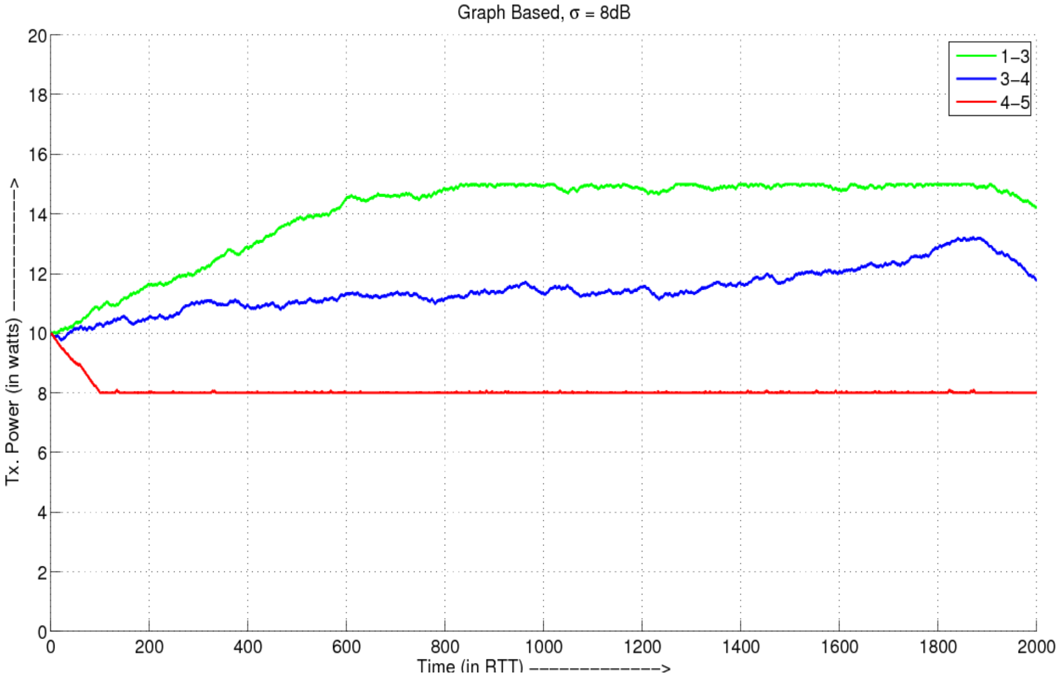


Figure 5.9 Variation of Tx power for three flows using QPN approach

We also conducted experiments with different standard deviation (σ) of Log-normal shadowing and plot the variation of throughput with σ in Figure 5.10. We observe that throughput decreases as σ increases in fixed modulation, whereas throughput remains almost constant in adaptive modulation. The higher rate of throughput drop for the fixed modulation can be attributed to the fact that as shadowing increases, transmission power decreases resulting in decrease in the average throughput. We also observe the average transmission power of individual nodes at different channel states (cf. Figure.5. 11). From this, we observe that the average transmission power decreases when the channel gain deteriorates (similar to Opportunistic scheduling) for fixed modulation, whereas average transmission power increases (at a slower rate) for adaptive modulation.

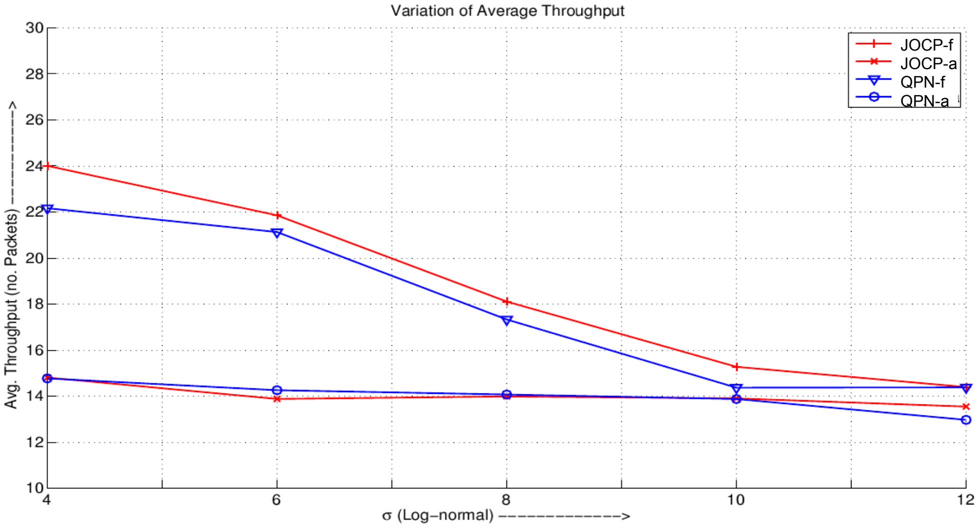


Figure 5.10 Variation of throughput with channel variation

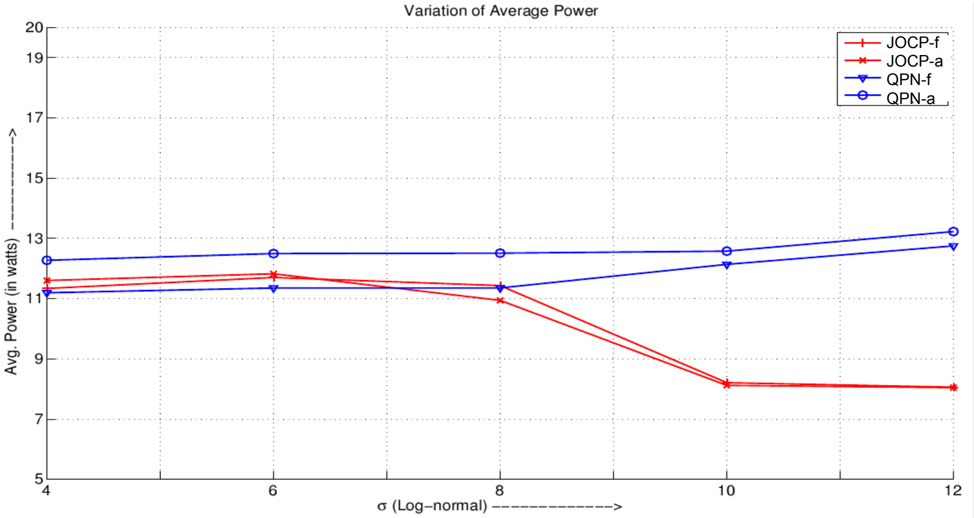


Figure 5.11 Variation of Tx Power with Channel Variation

A detailed discussion on this simulation study is published in [10].

5.6 Summary

In this chapter we proposed the use of QPN modeling the network behavior involving the interaction of multiple nodes. TCP congestion control behavior has been taken as a specific example and shown how the inferences are drawn towards a joint optimization involving TCP and the link layer. A simulation study demonstrating the applicability and performance of QPN model based qualitative optimization approach for a self-optimizing adhoc wireless network. The performance has been compared with a traditional algorithm. The simulation results show that both schemes stabilizes and optimizes system at-par with respect to throughput and transmission power for reasonably good channel conditions. On a positive note, in QPN approach we do not use any form of message passing, whereas in JOCP approach that , we considered here for comparison, we rely heavily on message passing to determine the modulation index and transmission power. Therefore, for a cross-layer implementation perspective, QPN model based approach is better and implementable as compared to that of JOCP approach and is scalable.

References

[1] L. S. Brakmo, S. W. O'Malley, and L. L. Peterson, "TCP Vegas: New Techniques for Congestion Detection and Avoidance," in ACM SIGCOMM, pp. 24–35, 1994.

[2] V. Jacobson, "Congestion Avoidance and Control," in ACM SIGCOMM, pp. 314–329, August 1988.

[3] R. Braden, V. Jacobson, and L. Zhang, "TCP Extensions for High speed Paths," vol. RFC-1185, October 1990.

[4] M. Mathis, J. Mahdavi, S. Floyd, and A. Romanow, "TCP Selective Acknowledgment Options," April 1996. RFC-2018.

[5] Karger, David (1993), "Global Min-cuts in RNC and Other Ramifications of a Simple Mincut Algorithm". *Proc. 4th Annual ACM-SIAM Symposium on Discrete Algorithms*.

[6] Karger, David Clifford Stein (1996). "A New Approach to the Minimum Cut Problem". *Journal of the ACM* 43 (4): 601–640.

[7] M. Chiang and R. Man, "Jointly Optimal Congestion Control and Power Control in Wireless Multi-hop Networks," in Proc. of GLOBECOM, December 2003.

[8] H. K. Rath, A. Sahoo, and A. Karandikar, "Cross Layer Congestion Control Algorithm in Wireless Networks for TCP Reno-2," in Proc. Of NCC, IIT Delhi, January 2006.

[9] H. K. Rath and A. Karandikar, "On Cross Layer Congestion Control for CDMA based Ad-hoc Networks," in Proc. of NCC, IIT Bombay, February 2008.

[10] Hemant K Rath, Rajan MA, Balamuralidhar, "Monotonic Signed Graph Approach for Cross-layer Congestion Control in Wireless Ad-hoc Networks", GlobeCom 2011

6

QPN Model based Self-Monitoring in Wireless Networks

In this chapter we propose a self-monitoring methodology for fault detection and diagnosis in wireless networks based on the QPN model. This uses the QPN inference mechanism along with related performance thresholds to detect a fault and explain the reason behind the failure. Subsequently a network management system can be triggered with the causes to initiate remedial measures automatically or with human intervention. The methodology is illustrated with an example of an antenna failure in a radio link. Subsequently we propose a scheme for ensuring regulatory compliance, another important aspect of self-monitoring.

6.1 Introduction: Self-Monitoring for Wireless Networks

In today's networks, lot of man hours is being spent in fault detection, diagnosis and remedial measures. Maintaining and operating this large and technically complex system is a difficult task that requires operational staff working round the clock. Undetected system faults have serious consequences that will affect network performance and user experience which is very costly for a competitive telecom operator [1]. Diagnosis is currently a manual process left to experts dedicated to daily analysis of a few key performance indicators (KPIs) and the alarms of the network. As the networks grow in size and integrate diverse technologies the task of monitoring cannot be accomplished satisfactorily with conventional methods. In cellular networks problems like detection of automatic cell degradation followed by diagnosis and healing has a huge importance with respect to economy and user experience. Beyond this the fore-sight is for self-managing systems, that self-configure, self-protect, self-heal and self-optimize are the efficiency enhancement approach to handle the complexity of future networks. From a network operational cost perspective this is a very important area, however it has not yet received deserved attention from researchers.

6.1.1 Self-Healing – Related Works

Self-healing process involves majorly three steps, namely, fault detection, fault diagnosis and fault recovery. Identification of the situation that there is a fault with the system is fault-detection. This may be inferred from many of the network performance indicators such as number of dropped calls, access failures, congestion etc. Once a failure is detected, the cause of the problem is to be diagnosed. It is the defective behavior of some logical or physical component of the network that caused the failure. It could be generated from a bad configuration, hardware failure, component degradation etc.

In many cases a fault could be associated with certain symptoms, such as alerts generated from violation of a constraint or threshold, oscillations in configuration etc. A list of common causes and symptoms of faults in wireless networks is given in Barco [2] for GSM/GPRS and in Khanafer for UMTS [3].

Faults in networks generally fall into one of the two categories [4]. Hardware and software subsystems that actively report the failures are one category. The second category faults are more difficult to detect, such as RF failures (antenna orientation and connectivity issues, power amplifier degradation etc), configuration and scheduling problems. “Sleeping cells” in cellular networks that do not service any traffic are especially problematic and difficult to detect.

The first step in fault management is to differentiate between normal and abnormal system behavior. Any “symptoms” of abnormal behavior are linked to a set of potential causes, in most cases based on empirical knowledge supplied to the system by human operators.

One of the traditional approaches followed for fault detection in RAN is based on alarm correlation [5]. Alarm correlation consists in the conceptual fusion of multiple alarms, so that an inferred meaning is assigned to the original alarms. Although alarm correlation can be used for diagnosis of faults, many times it does not provide conclusive information to identify the cause of problems.

Application of Bayesian Networks (BN) to diagnose fault states under uncertainty has been explored for industrial control systems [8][9]. Compared to deterministic rule-based systems BNs provide robustness to unreliable observations [9][6]. In network management also BN based fault diagnosis has been investigated [2][7][9] . The authors in [7] train a BN to learn the normal state of a network based on observations of network traffic in a router. They show how the trained BN is capable of detecting network anomalies but do not diagnose their causes. In [2] diagnosis is performed in an end-node solely to support fault recovery that only uses traffic observations readily available in the end node.

Here we propose a scheme for self-monitoring in cognitive engine that uses QPN for modeling the network behavior. Unlike in BN where quantitative probabilistic information is used to inference the causes of a fault, the QPN based inference engine is used to check the consistency of network behavior with respect to the model. The parameters involved in inconsistent relations are flagged as potential anomalies.

6.2 Fault Detection using QPN model

A high level architecture of the self-healing functionality mapped to the cognitive engine is presented in Figure 6.1. The fault detection module monitors the context sensors and generates fault events whenever a fault condition is evaluated true. These fault conditions are part of a fault & diagnosis model. Once a fault event is generated, the reasoning engine is triggered and possible causes of faults are inferred. If remedial actions corresponding to these causes is available in the Action database then that is deployed by the reconfiguration module. Otherwise suitable alarms are generated.

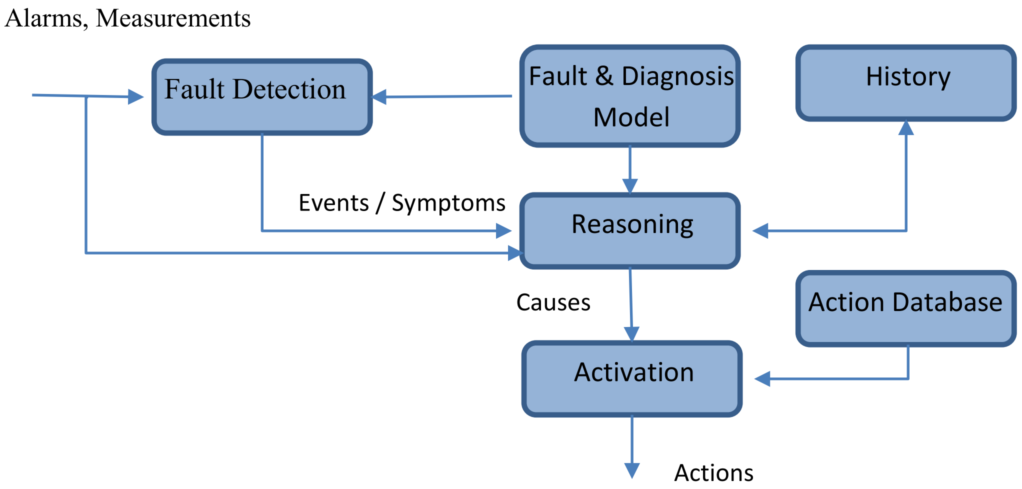


Figure 6-1 High level architecture for self-healing in a Cognitive Engine

Further we propose to realize each of this modular functionality as described below.

6.2.1 Fault & Diagnosis Model

We use the QPN model of the network as the basis for building the fault and diagnosis model. For this model we introduce two additional categories of variables. They are *Fault Cause* variables (F_c) and *Fault Detector* variables (F_d) . F_c is a set of binary variables that indicates the presence or absence of a fault with the respective subsystem represented by the variable. These variables are added in the QPN structure reflecting their causal relationships with other context variables. F_d is a situation specific binary variable with an associated Boolean expression which on evaluation indicates the presence of a fault if F_d=TRUE.

Each of the *Fault Detection* variable F_d is associated with a tuple $< S_i , C_i >$ where S_i represents a situation and C_i is a condition (Boolean expression over the context variables) associated with a fault. This is to state that in any situation similar to S_1 if the condition C_1 becomes TRUE, then a fault F_{d1} is detected. These new variables are added to the QPN structure in a similar fashion as the context specific nodes presented in chapter 3.

6.2.2 Fault detection and diagnosis

The fault detection module periodically updates the fault detection variables and generates events in case they evaluate to be TRUE. The reasoning engine uses the event information and observations to infer the state of the fault cause (F_c) variables. This is done through the QPN sign propagation algorithm described in previous chapter.

6.2.3 Illustrative Example – Antenna Failure

Here we consider a simple example of a radio link (Figure 6.2a) in which the above approach is used for detection and diagnosis of a fault with antenna system. The figure shows the QPN based fault model of the link. P_t is the transmitter power and I_{in} is incoming interference at the receiver. We introduce a *fault cause* variable F_{GA} which represents a fault with the antenna gain (undue reduction of antenna gain due to a fault). When F_{GA} is TRUE, it will have a negative influence on SINR (in the presence of N_o - thermal noise). The separation between transmitter and receiver is represented by d. Here we define undue increase of FER as the symptom of a fault in the radio. It is defined by the tuple $F_d^1 = < S_1, "(FER > \alpha_{FER})" >$ where $S_1 = \{P_t, d, I_{in}\}$, relevant parameters defining the situation. α_{FER} is the reference FER from history corresponding to a situation S_1^* that is similar to S_1. When they are similar situations (current *situation* S_1 and the reference situation S_1^* in the database) the QPN signs of the variables associated to S_1 (change with respect to S_1^*) are $\delta(P_t) = \delta(d) = \delta\ (I_{in}) = 0$. Using this as the observations a QPN sign propagation of δ(FER)='+' yields the inferred sign value of $\delta(F_{GA})$ as '+'. The interpretation of this inference is that while the situation S_1 identical, an increase in FER greater than α_{FER} indicates a failure of Antenna as the cause of the observed fault. The argument used here is that while all the dependent variables of FER, other than F_{GA}, known to be stationary, then any change with FER will be due to F_{GA}.

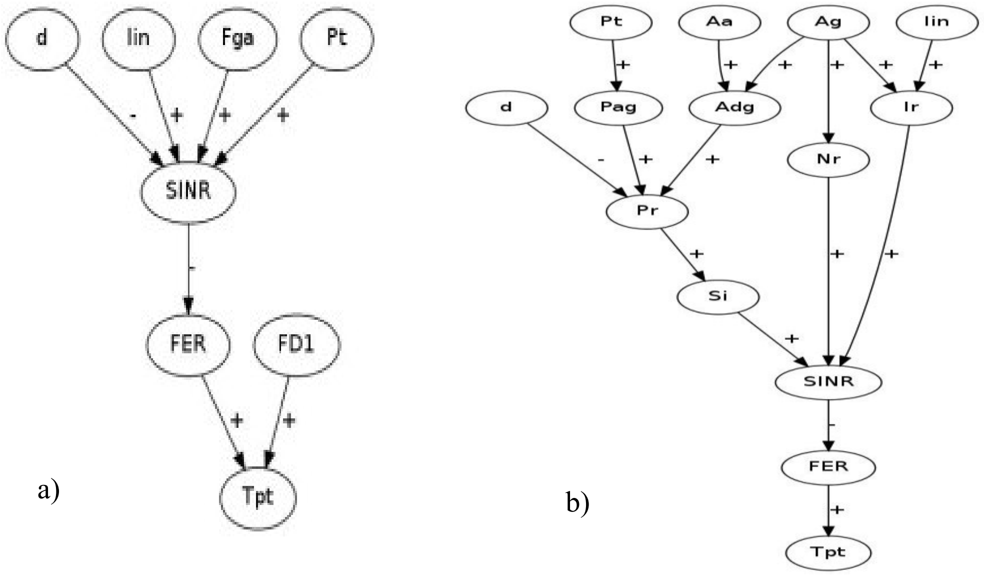


Figure 6.2 Simple influence diagram showing fault model of a radio link as an illustrative example. a) Depicting *Fga* indicating antenna fault b) Depicting two fault points antenna gain (Ag) and Antenna alignment (Aa)

We present an algorithm for the fault detection diagnosis using QPN model in Algorithm 6.1. Here we present the algorithm as an infinite loop checking for events as indicators for possible faults. We prepare a database of reference *situations* those are frequently occurring and consistent with the QPN model. We define a *symptom* variable C_i which is often a Boolean expression of a set of observed parameters. Whenever the symptom is detected and a matching reference situation in database exists, a fault detected flag f_d is set. A list of parent nodes of V, where the symptom variable is attached, which are head nodes form the possible set of fault causes F_c^*. A list O of observation variables are prepared which are parents of V (and head nodes) not including F_c^* . A QPN sign propagation with these observation variables in O set to '0' (that is no change) yields its effect on F_c*. If any of the variables Fc* is set then that is declared as potential causes of faults.

Figure 6.2b presents an influence diagram with some more additional fault parameters. It includes Antenna Alignment (Aa), Antenna directional gain (Adg), Power Amplifier Gain (Pag). Other dependent parametrs are Interference at the receiver (Ir), Noise power at the receiver (Nr), and Signal Power received (Si). The same procedure described above can be extended to this towards determining the related faults.

Algorithm 6.1 Algorithm for fault detection and diagnosis

```
FD – Fault Database
$F_d$ –     Set of faults
$F_c$ –     Set of Fault causes
Fd, S, C – Fault id, Situation, Fault condition
$f_d$ -     Fault detected
$f_c$ -     Fault cause
FD = {< $F_{di}$, $S_i$, $C_i$ >}

$S_c$ – Current situation
while TRUE
        if similar ($S_c$ ,$S_i$ ) and eval($C_i$)=TRUE then
                $f_d = F_{di}$
                $F_c{}^* \leftarrow \pi^*(\sigma(F_{di})) \cap F_c$
                $O \leftarrow \{< v_i,' 0' > | v_i \in \pi^*(\sigma(F_{di})) \backslash F_c^* \}$
                $\delta(F_c{}^*) \leftarrow$ signpropagate ($F_{di}$,'+',O)
                if $f_j \in \delta(F_c^*)$ and $f_j$='+' then
                        $f_c \leftarrow f_j$
                        InitiateAction($f_d$,$f_c$)
                end if
        end if
end while
```

6.3 Radio Monitoring for Regulatory Compliance

Flexibility provided by the radio part of a cognitive network node is another important point for monitoring from a regulatory perspective. The implementation of a cognitive radio in general follows a Software Defined Radio (SDR) approach providing sufficient flexibility for supporting cognitive capability. Ensuring regulatory compliance of the cognitive radio is a very important aspect to be addressed for commercial deployment.

Regulatory aspect of the cognitive radios - that is, how to make sure that any radio terminal designed would operate properly in the ecosystem of CR – is an aspect needed for certification. Cognitive Radio may use the spectrum differently based on its multiple modes of operation. The transmission modes may change from what is anticipated at design time. These changes may be due to the reconfiguration action of the CR or some software failure. So there is a possibility of interferences from the operation of CR which need to be contained.

6.3.1 Regulatory Compliance

While it is mandatory for the radio to implement the regulatory policies, the compliance needs to be established for regulatory certification. The existing communication devices are type approved by certifying agencies with the assumption that they have tested all possible modes of operation of the device and there is a mechanism exists in the device to prevent it from going to any new modes. With the new dynamically reconfigurable communications devices this is not true, there are innumerable modes that are difficult to cover through testing and there are unanticipated modes of operation that the device may configure itself into. Here we investigate the structure that is required for a CE to address this requirement.

6.3.2 Regulatory Compliance Monitor for Cognitive Radios

We conceptualize a regulatory compliance monitor (RCM) for cognitive radios to be incorporated in the cognitive engine. Following are the major requirements of RCM considered. An authenticated and truthful regulatory policy information need to be available to the RCM. The sensors and context measurements that provide the information elements for policy evaluation need to be trustable.

The policy updation mechanism provides the current and authenticated machine readable policy to the RCM. For the evaluation of these policies the required context information is provided from the knowledge repository. On the event of a reconfiguration the control parameters are scrutinized by the RCM as guided by the regulatory policy. Based on the conformance status RCM gives out a go/nogo signal that enables/disables the reconfiguration controller from using the proposed control parameters. Figure 6.3 gives a high level architecture of integrating the RCM functionality in the cognitive engine.

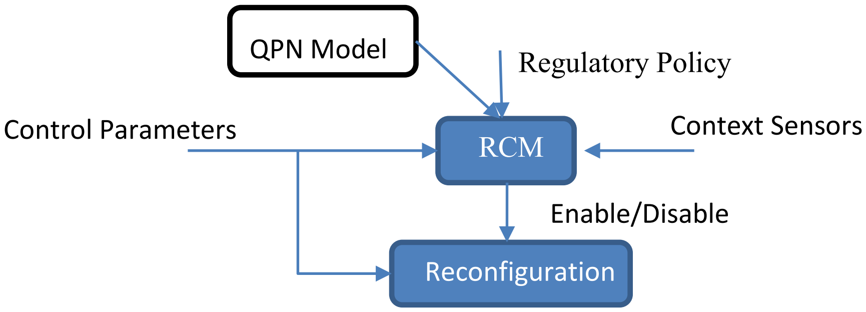


Figure 6.3 Integration of Regulatory Compliance Monitor (RCM) in cognitive engine

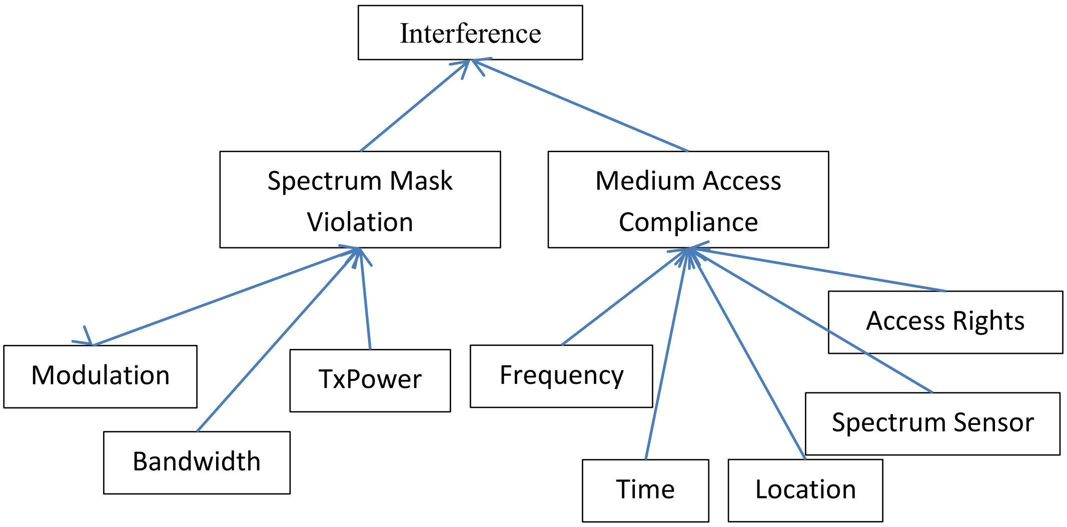


Figure 6.4 Dependency diagram for Interference as a compliance measure in Cognitive Radios

Interference is a useful metric to monitor the regulatory compliance of a radio. Figure 6.4

presents a view of the dependency diagram showing major parameters that determine the interference generated by a radio to other radio receivers. Majorly the interference is due to the spectrum mask violation and deviation from medium access etiquette. The dependent parameters include frequency band, bandwidth, transmit power, modulation, time & location of the radio and performance of the spectrum sensor. The medium access scheme and dynamic access rights are also part of the MAC etiquette. Location information is needed for a cognitive radios with universal roaming, and the spectral policies may be different for countries, regions, localities (urban Vs rural). In future it may have a dependency even on Time of Day. The dependency diagram giving the related control and observation parameters is extracted from the QPN model by the RCM. This can be mapped to the fault detection & diagnosis architecture where a constraint violation/fault is acted upon. In a similar manner a violation of regulatory policy is also considered as a fault event.

6.3.4 Approach for Regulatory Certification based on RCM

As explained above, responsibility of the RCM is to evaluate the constraints based on the measured or estimated constraint variables. These measurements are done through the *Sensing* module of the CE. Accuracy of sensor measurements has an impact on the constraint evaluation. Some constraints are enforced by embedded design ensuring respective compliance. We propose the procedures listed below to assess the compliance of radio device:

a) Correctness of the policy statements and the QPN Model
b) Authentication mechanism for the policy management
c) Validation of the mechanism for constraint formation from policies
d) Test the accuracy of the sensors
e) Verification of correct provision of device context information
f) Implementation of the control mechanism in case of a violation
g) Black box testing of implicit constraining by design

6.3.5 Policy Management

Spectrum policies have to be expressed in a machine readable form to enable autonomic reasoning and decision making. A policy language could be used to express the policies published by regulators which are generally in a human readable form. A relevant work towards this has been reported for DARPA XG program in which a language framework of Ontology Web Language (OWL) has been used [9] . This is a declarative language enabling merging of multiple policies. OWL can be seen to be equivalent to description logic (DL), which allows OWL to exploit most of the existing DL reasoning approaches.

A CR will use these policies with appropriate reasoning to select a potential compliant mode of transmission. From a regulatory point of view what needed is a framework for detecting and stopping policy violations. This can be seen as detection of violation of certain boundary constraints derived from the above policies. These constraints may be derived statically in a constraint repository or dynamically based on the device context.

6.3.6 Security

Policies are to be updated in a secure manner. Agencies involved in the specifications are Regulators, Manufacturers and Operators. Digital certificates can be one of the methods to authenticate the policy files and related processing commands. Architecture of a Radio Security Module enabling global roaming of an SDR terminal presented in [11] takes a similar approach. Simpler schemes involving light weight cryptosystems can also be considered. A policy based framework could be used as a secure boundary, similar to the concept of trusted kernel in secure systems. Critical performance validations are kept within this kernel and future innovations can happen outside this boundary. The secure module should not be modifiable by software reconfiguration and it should be sufficiently tamper proof.

6.4 Summary

Self-monitoring for fault detection, diagnostics and reconfiguration is an important requirement in cognitive networks and that needs to be facilitated by the cognitive engine. An approach based on QPN model to support the feature of fault detection and diagnosis is proposed in this chapter. The QPN sign propagation based inference along with the historical information is used to detect a deterioration of a key performance metric that indicating a possible fault condition and causes. Further a scheme for monitoring of radio for regulatory compliance has also been discussed. The proposed approach of using Radio Compliance Monitor (RCM) integrates well with the CE architecture. The RCM works inside a secure policy management boundary. Validation of the RCM could be a basis for ensuring radio compliance and certification.

References

[1] “Designing, Operating and Optimizing Unified Heterogeneous Networks”, Whitepaper from nokia-siemens networks

[2] Barco, R., Wille, V., & Díez, V. (2005). System for automated diagnosis in cellular networks based on performance indicators. European Transactions on Telecommunications, 16(5), 399–409.

[3] Khanafer, R., Solana, B., Triola, J., Barco, R., Nielsen, L., & Altman, Z., et al. (in press). Automated diagnosis for UMTS networks using bayesian network approach. IEEE Transactions on Vehicle Technology. doi:10.1109/TVT.2007.912610.

[4] Raquel Barco, Pedro Lázaro, Volker Wille, L. Díez, Sagar Patel, Knowledge acquisition for diagnosis model in wireless networks, Elsevier journal on Expert Systems with Applications 36 (2009) 4745–4752.

[5] Wietgrefe, H. (2002). Investigation and practical assessment of alarm correlation methods for the use in GSM access networks. In Proceedings of the IEEE/IFIP network operations and management symposium, Florence, Italy.

[6] M. Steinder and A. Sethi, “Probabilistic fault localization in communication systems using belief networks,” IEEE/ACM Transactions on Networking (TON), vol. 12, no. 5, pp. 809–822, 2004.

[7] C. Hood and C. Ji, “Probabilistic network fault detection,” Global Telecommunications Conference, vol. 3, 1996.

[8] E. Horvitz and M. Barry, “Display of information for time-critical decision making,” Proc. of the Eleventh Conference on Uncertainty in Artificial Intelligence, 1995.

[9] Z. Yongli, H. Limin, and L. Jinling, “Bayesian networks-based approachfor power systems fault diagnosis,” Power Delivery, IEEE Transactions on, vol. 21, no. 2, pp. 634–639, 2006.

[10] Charlotte Adams, “Policy Controlled radios, Making room in the Spectrum”, Avionics Magazine March 2005

[11] Chih Fung LAM, Kei Sakaguchi, Jun-ichi Takada, Kiyomichi Araki, “Radio Security Module that Enables Global Roaming of SDR Terminal while Complying with Local Radio Regulation”, IEEE VTC, 2003

7

Cognitive Acoustic Communications

In this chapter we propose a novel communication scheme using upper audio frequency band for short range communication between devices/appliances over the air. We refer this scheme is Upper Audio Band (UAB) Communication. Cognitive Radio concepts have been used here to operate this communication modem as a secondary in presence of primary users such as TV, Home Entertainment Systems. This acoustic modem has been implemented and its performance details are discussed. Further the idea of using this in a personal area cognitive network is presented.

7.1 Introduction - Short Range Acoustic Communications

Here a novel and interesting use case with a practical implementation of an acoustic cognitive radio is introduced. This addresses the need of a low cost software modem that can be used for short range low data rate device-to-device (D2D) communications. A major requirement that has been put on the system is that it should use audio, but should be imperceptible for human ears, at the same time the frequency band used should be supported by the devices involved. This stringent requirement has been addressed by choosing the upper audio band in the range of 16KHz to 20KHz for communication. Achieving meaningful data rates in this narrow band required us to adopt innovative modulation techniques along with error correction coding for robustness. Further the intended spectrum might be occupied by transmissions from entertainment appliances such Hi-Fi audio systems, HD-TV etc. Therefore our design is to use the cognitive acoustic radio as a secondary user with the above appliances as primary.

7.2 Related Works

New methods of communication, protocols and signal processing and accompanying standards are needed for devices in the Internet of Things so that they can connect to a variety of access points even if conventional RF interfaces are not available. Prior work in [1,2,3] has used the audio band for communicating data. In [2], multi-carrier modulation has been proposed for transmission of data over the air using the perceptible audio band of 6.4-8KHz. The frequencies in this band are first filtered out from the original sound. A multi-carrier modulation with 32 sub-carriers is then used to communicate data. In [3], data is embedded in a sound signal over the perceptible audio band. This method performs a Lapped transform on an audio signal, and then modifies the phase of the transform coefficients to embed the data. The inverse of the transform coefficients with modified phase gives a time-domain signal, which is then played by a speaker. The error rate performance reported for the systems in [1,2,3] is unsuited for data communication. In addition, there is an inevitable distortion of the original audio signal in which the data is embedded.

A method of communicating data over the audio band without distorting existing sound signals is needed. Signaling methods suited for the audio channel need to be designed. Audio interfaces on mobile devices are typically sensitive to frequencies upto 20KHz. Since the Upper Audio Band (UAB) in the range 16-20 KHz is almost imperceptible to the adult human ear over short time duration, it can be used for data communication. Within this bandwidth, it is possible to communicate at a low data rate over a short range depending on the sensitivity specifications of the speaker and microphones.

7.3 UAB Physical Layer Design

To design signaling schemes for communication over the audio channel, it is important to study the channel characteristics. In this section we characterize the UAB channel and propose signaling schemes matched to this channel. The signal processing operations at the receiver are optimized for the audio channel.

7.3.1 UAB Channel Model

The audio channel can be modeled as a Linear Time Invariant (LTI) system. Eigen functions of LTI systems are complex exponentials which pass through the channel undistorted. Multi-tone FSK signals which are linear combinations of sinusoids are eigen functions, which can be used for communication over the audio channel. The signal model in the next section elaborates on the choice of frequencies for a multi-tone FSK signal. Furthermore, in a closed room environment, without ambient sound, the recorded signal on a microphone was observed to have an approximately Gaussian distribution.

7.3.2 Signal Model

For this application we propose the use of Multi-tone FSK modulation with tones arranged in a lattice structure for communicating data over the audio channel. A dense packing of circular contour plots with a given radius in 2 dimensional space is a hexagonal lattice packing [4]. If two tones (f_1, f_2) are chosen from a Hexagonal lattice, the correct decision region (Voronoi region) for each lattice co-ordinate point is a hexagon. The generator matrix for a hexagonal lattice is given by [4]:

$$G^{(2)} = \begin{bmatrix} 1 & 0 \\ 0.5 & \sqrt{3}/2 \end{bmatrix} \tag{2}$$

With a base pair of frequencies, M1=13000 Hz, M0=12000 Hz and a translation of 6500 Hz, we compute the 4 pairs of tones for a 2-tone FSK modulation to obtain the 4 adjacent frequency lattice co-ordinates:

F_1 =[M0 M0] $G^{(2)}$ +[0 6500]= [18000,16892]

F_2 =[M0 M1] $G^{(2)}$ +[0 6500]= [18500,17758]

F_3 =[M1 M0] $G^{(2)}$ +[0 6500]= [19000,16892]

F_4 =[M1 M1] $G^{(2)}$ +[0 6500]= [19500,17758]

The co-ordinate points $[F_1, F_2, F_3, F_4]$ are labeled using a binary reflected Gray code of order 2. Adjacent co-ordinate points are spaced 1000Hz apart in the frequency plane. The frequency lattice constellation is shown in Figure 7-1.

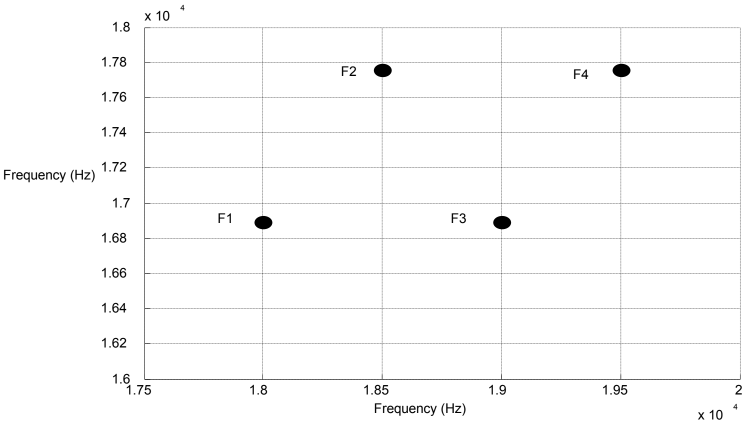


Figure 7-1 Frequency lattice constellation for 2-tone FSK modulated signal

We define a channel as the frequency band spanned by a set of 4 adjacent lattice co-ordinate points. Multiple channels can be introduced by translating the lattice co-ordinates in a default channel. The hexagonal lattice packing is a spectrally efficient packing in the frequency plane allowing a larger number of channels to be introduced, compared to any other arrangement of frequency co-ordinate points.

The signaling model can be extended to a 3-tone FSK. In three dimensions, a laminated hexagonal lattice is the densest packing of spheres[4]. The generator matrix can be derived as:

$$G^{(3)} = \begin{bmatrix} 1 & 0 & 0 \\ 0.5 & \sqrt{3}/2 & 0 \\ 0.5 & 0.5/\sqrt{3} & \sqrt{2/3} \end{bmatrix} \qquad (3)$$

7.3.3 Pulse Shaping

The multi-tone FSK modulated signal is pulse shaped using a Dolph-Chebyshev window to reduce the perceptibility of spurious frequencies to human ears caused by sharp transitions across signaling intervals and to create silence zones between adjacent signaling intervals. The silence zone should be larger than the maximum expected delays of acoustic echoes in a closed room environment to avoid errors in decoding due to inter-symbol interference. Figure 7-2 shows the Power Spectral Density (PSD) of a 2-tone FSK signal. The Chebyshev pulse shaped signal has a lower out-of-band spectral response compared to the non-pulse shaped signal. This reduces the perceptibility of interference to human ears due to the out-of-band frequencies.

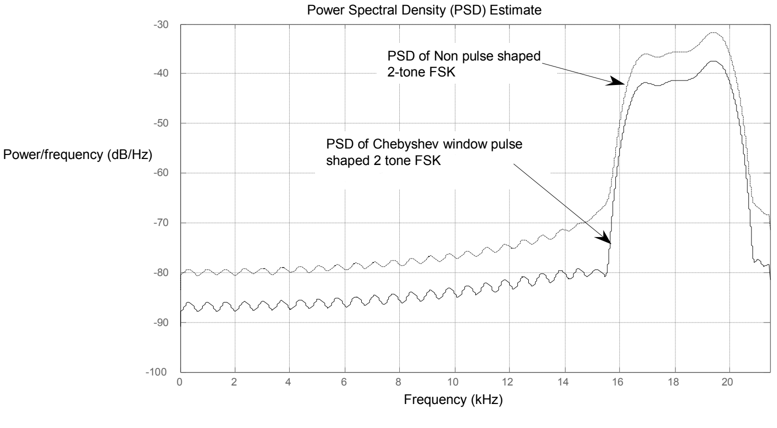


Figure 7-2 Power Spectral Density estimate comparison for a 2-tone FSK modulated signal (Chebyshev window shaping v.s no pulse shaping)

7.3.4 Receiver

The received signal is filtered using a band pass filter with a pass-band range of 16-22 KHz. With a default sampling interval $t_s = 1/f_s$, where f_s=44100 samples/s. A signal-onset detection is performed at the receiver using the Teager-Kaiser (TK) energy operator on the received signal samples. The TK energy operator [5] is useful to detect transient signals in the presence of noise. The output of the TK energy operator when applied to a 2-tone FSK modulated signal over a time interval [0,T/2] is shown in Figure 7.3. The TK energy operator acts as a filter, reducing the noise in the received signal.

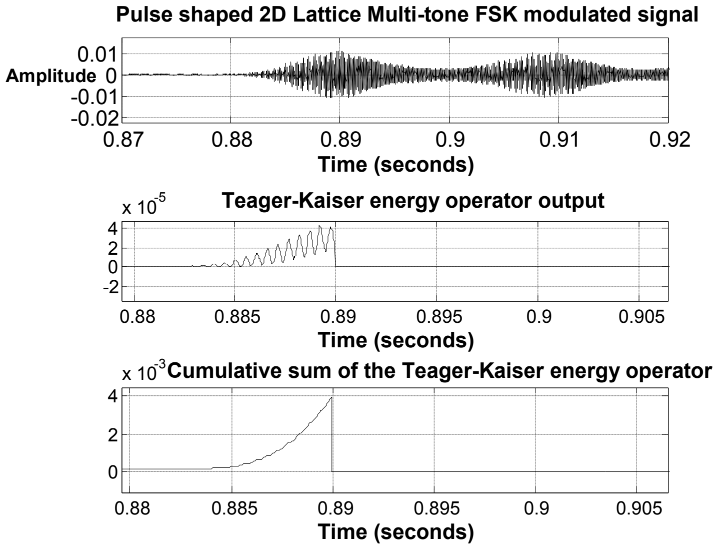


Figure 7.3 The TK energy operator output when applied to a pulse shaped 2 tone FSK modulated signal over a time interval [0,T/2].

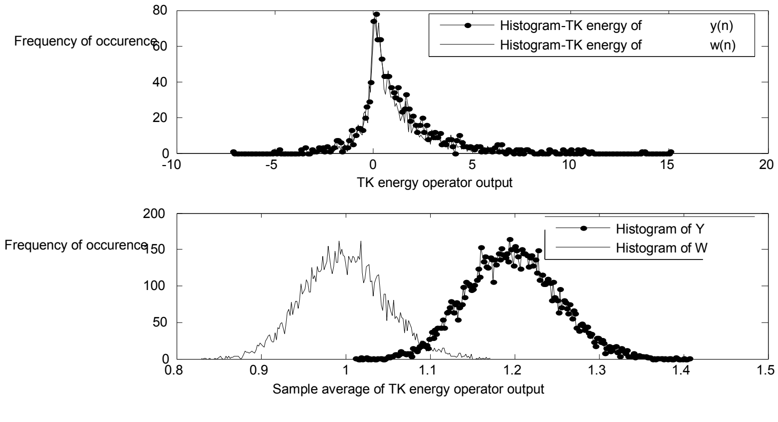


Figure 7-4 Histogram of the TK energy of the signal and the sample average of the TK energy.

Figure 7-4 shows a histogram plot of the TK energy for the signal and the noise. The histograms are observed to have overlapped Meijer G distributions. In contrast, the sample averages Y and W were observed to have well separated Gaussian distributions. The p.d.f $f_W(w)$ is centered on the variance of the noise sequence $\{w_C(n)\}$. The signal onset detector now has to test for two hypotheses:

$$\begin{cases} W < \gamma & : H_0 \\ Y > \gamma & : H_1 \end{cases} \qquad (4)$$

The detector threshold γ must be chosen such that:

$$P[W > \gamma \mid H_0] \leq p_{FA}$$
$$P[Y < \gamma \mid H_1] \leq p_{MD} \qquad (5)$$

An optimal threshold γ is chosen as the point where $f_Y(\gamma) = f_W(\gamma)$. Since the p.d.f $f_W(w)$ is centered about the variance of the noise $\{w_C(n)\}$, the threshold γ is dependent on the noise variance. The lowest Signal to Noise Ratio (SNR) for which the constraints in (5) are met is the detector sensitivity. Figure 7-5 shows the BER performance for different multiple tone FSK modulated cases. In the multi-tone lattice a smaller spacing between frequency lattice points may allow the use of additional channels, but will degrade the BER performance. Hence the tradeoff between the allowed number of channels and the BER performance needs to be taken into consideration when choosing frequencies for the multi-tone FSK modulation.

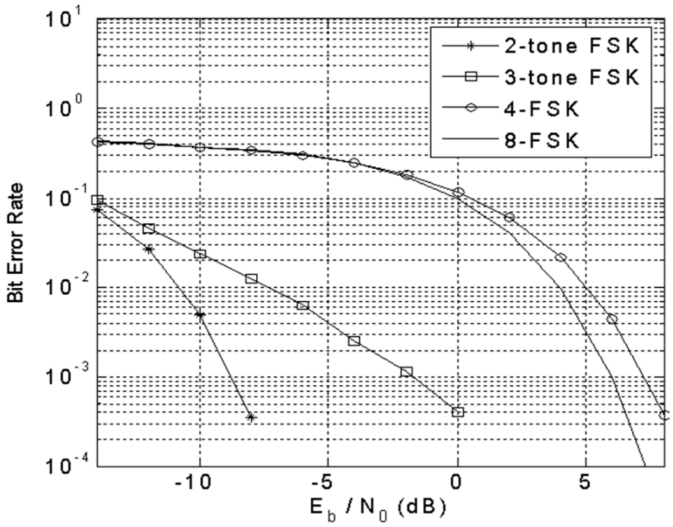


Figure 7-5 Comparison of Bit Error Rates for a 2/3 dimensional lattice multi-tone FSK modulated signal and 4, 8 FSK over AWGN

7.4 UAB modem prototype design

A prototype modem was developed (Figure 7.6) using the upper audio band of frequencies and the proposed multi-tone modulation format. A three symbol error correcting (15,9) Reed-Solomon (RS) code over the field GF(2^4) was used for data encoding/decoding. A data sequence $D = \left[d_{1,} d_{2,.} .., d_{32} \right]$ with $d_i \in \{0,1\}$was appended with a flag sequence [0000], giving a 36 bit message sequence [D, [0000]] which was encoded using a (15,9) Reed Solomon code. The field elements of the RS code over the field GF(2^4) were generated using a primitive polynomial $p(X) = 1 + X + X^4$. The encoded symbols were mapped to a 2-tone FSK signal for data transmission over the UAB. The modulated signal is played using compatible speakers.

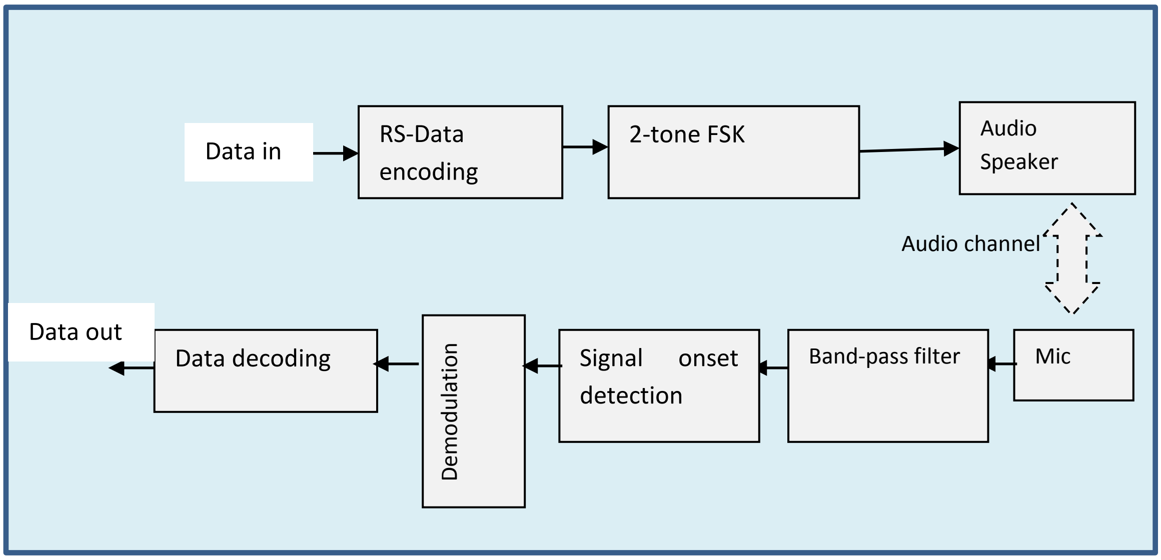


Figure 7-6 Transmit/Receive chain for the UAB modem

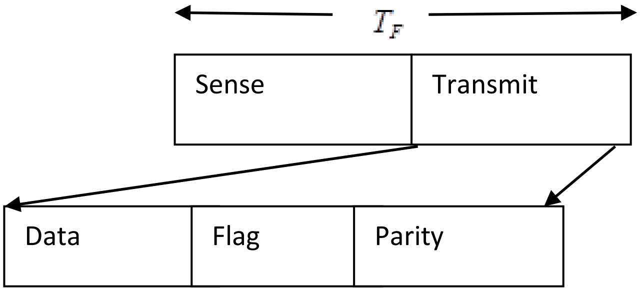


Figure 7-7 Physical layer frame structure

For the medium access we follow a simple scheme of repeated cycle of two slots, namely Sense and Transmit [6]. The time slot structure is designed by dividing each fixed-size cycle of length T_F into the following two phases:

1. ***Sensing Phase***- During the sensing phase, all active UAB devices perform spectrum sensing on the channel. Only links whose sensing results in *Not Busy* status proceed to the next phase. A channel is usable if the computed periodogram of the received signal does not cross a pre-determined threshold at the tone pairs/triads for the 2/3 tone FSK modulation.

2. ***Data transmission phase*** -A 32 bit data sequence and a 4 bit flag sequence is RS encoded. The physical layer frame structure is shown in Figure 7.7. The Flag sequence of 4 bits can also be used for indicating the modulation scheme (2/3 tone FSK modulation) to the receiver, for a reconfigurable modem, which can adapt its data rate.

There is a tradeoff between interference avoidance and sensing efficiency. Sensing efficiency is determined by the following parameters:

1. Sensing periodicity
2. Detection sensitivity

A sensing slot is scheduled every data frame. This period can be varied (scheduled for every set of data frames instead) depending on the environment. The thresholds estimated for identifying valid tones determine the false alarm and misdetection probabilities and depend on the sensing time interval. A larger sensing time interval allows a better detection at the cost of reducing the throughput. The signaling interval for a 2-tone FSK modulation was taken as 20ms, giving a data rate of 100 bits/s. One RS codeword over the field GF(2^4) is represented by 60 bits. An encoded data frame of 60 bits spans a time interval of 1.2 seconds. The Beacon broadcast and Quiet period were each allocated a time interval of 40ms. For a 3-tone FSK modulation, a RS encoding over GF(2^3) was used. The field elements of a two error correcting (7,3) RS code over this field were generated using a primitive polynomial $p(X) = 1 + X + X^3$.

7.5 Implementation and Experimentation

The UAB modem is implemented on an *Apple iPad* in software. The microphone of the iPad is used as the receiver. A portable mp3 player used as a transmitter which played a pre-computed transmitter waveform stored in its flash card. For a data rate of 100 bps, the data could be decoded successfully at a receiving iPad upto a distance of 10 meters from a portable speaker with an 80 dB/W/m Sound Pressure Level (SPL) specification playing an mp3 encoded transmit waveform file. The receiving iPad was able to decode data correctly in the presence of background instrumental music in a closed room lab environment with furniture and equipment. It was observed that receiving microphones need to have sensitivity in the range of 50-100 mVolts/Pascal.

A practical demonstration on indoor location identification using a UAB modem implemented on Apple iPad is shown in Figure 7.8. Here the location code is converted to a UAB modulated audio stream in mp3 and an mp3 player is used to transmit this periodically. An Apple iPad or iPhone having the UAB application installed, receives this audio location beacon and detects the location id.

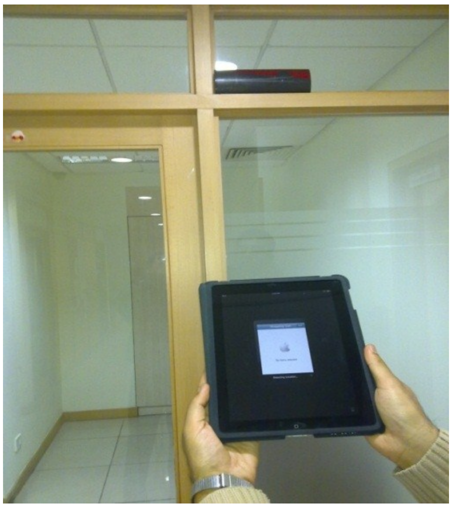


Figure 7-8 Cognitive UAB modem implemented on Apple iPad receives location messages transmitted by a UAB transmitter (an mp3 player is used) kept at the top.

7.6 Cognitive Communications using UAB

The UAB modem can be positioned to work as a secondary in presence of various primary transmitters. The possible primary transmitters are TV, Hi-Fi music systems and other audio systems. The UAB transmission should not create interference to humans who are listening to those primary transmissions.

Figure 7-9 shows the spectrogram recorded during the UAB transmission in presence of a TV and Hi-Fi music system as primary. Since there is a strong interference from the audio/music the UAB modem schedules on only one channel near 20KHz that is relatively interference free. Other channels those are below the first one is turned off. In Figure 7.4b it is observed that the UAB modem schedules another channel when the primary transmitter goes off.

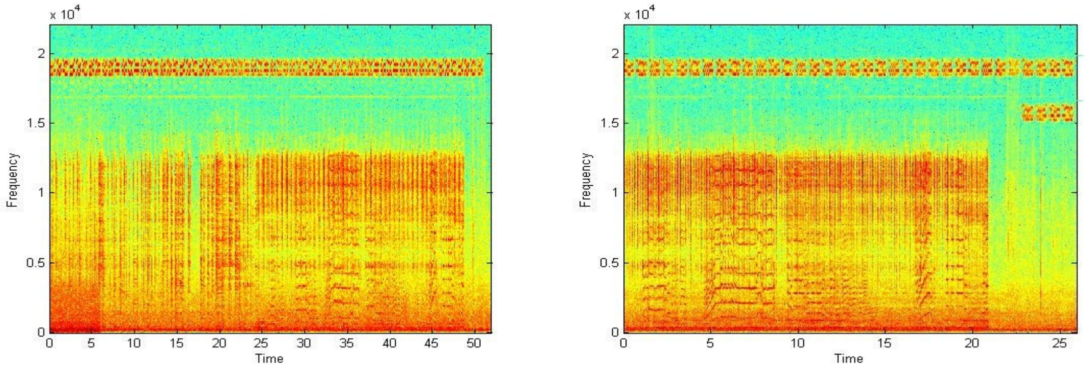


Figure 7-9 Spectrogram corresponding to UAB transmission in presence of TV and Hi-Fi Music.

For an efficient cognitive operation the performance of spectrum sensing plays an important role. Some observations based on practical experiments using the above prototype are presented in the following section.

7.6.1 Sensitivity of Sensing Duration

We have conducted a practical experiments involving the measurement of throughput of UAB link for different sensing durations in presence of a primary transmission. A slotted data transmission from the Primary User (PU) was generated on a single channel. The receiving audio modem was tuned to synchronize with the empty time slots to schedule its transmission. The total number of bits it can transmit in the sensed empty time slots determines the throughput of the audio modem. The throughput is hence defined as the ratio of the total time of sensed empty time slots to the total time of observation. The effective throughput for different sensing durations and a 2-tone FSK signal model is given in tables 2a, 2b, 2c.

Table 7-1 Effective throughput for different sensing durations for different speaker-microphone distances

Sensing duration	Throughput (kbps)		
	2 ft	4 ft	6 ft
10ms	0.55	0.54	0.95
13ms	0.54	0.53	0.93
20ms	0.52	0.50	0.90

It can be seen that a better throughput is obtained with smaller sensing duration. With an increase in distance (e.g. 6ft), the signal power of the PU reduces and the received signal at the spectrum sensor does not meet the set thresholds. This makes the spectrum sensor flag more ‘empty’ time slots resulting in more transmission opportunities and an increase in the throughput. With weak PU signals, there is a low probability of incorrect decoding by the SU due to interference and hence the throughput increases.

The details of the UAB communication modem design, implementation and experimentation are published in [7][8]

7.6.2 Range Performance

An experiment is conducted to observe the range performance of the UAB link. The received signal strength measured at the receiver for various distances is plotted in Figure 7.10.

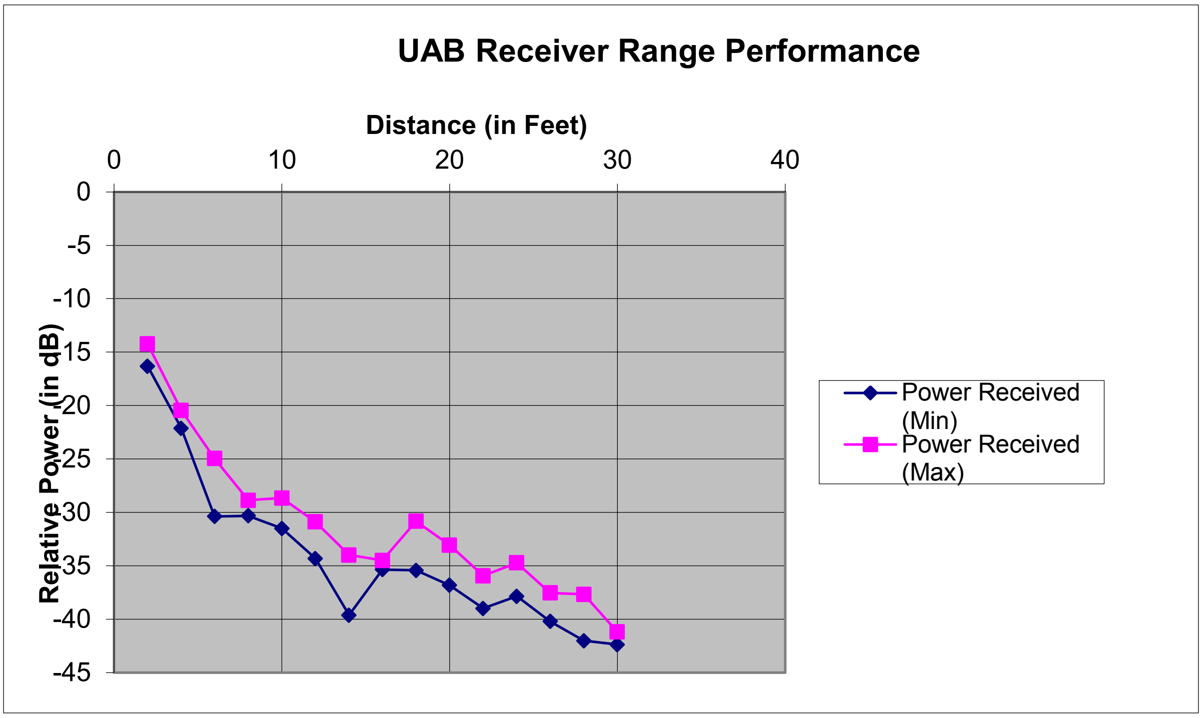


Figure 7.10 Variation of received signal strength with range

The characteristic is similar to the response of a short range radio link such as IEEE 802.15.4. In spite of the local undulations the overall characteristics is observed as a monotonic causal relationship from *distance* to *received signal strength* (or SNR).

7.7 QPN Model for the UAB Link

Based on the above performance characteristics and experimental observations the behavior of the UAB link is modeled as follows. As first step we list the observed facts followed by the construction of a corresponding QPN representation.

Let us define following terms:

m – modulation (number of tones , {2,3,4,8})
Ts – spectrum sensing duration ({10,13,20} ms)
P_D – Probability of Detection of Primary
d_P – Distance from Primary Transmitter
d - Distance between secondary Tx and Rx
Pr - Received signal strength
BER – Bit Error Rate

Table 7-2 Influences Observed for UAB Link

From	To	Sign
m	BER	+
Ts	P_D	+
Ts	Tpt	-
d_P	Tpt	+
d	Pr	-

With these observations and using additional well known background information we construct the following QPN model for the link. We have added two known variables TxR (Tx bitrate) and TxO (TxOverhead) for a clearer interpretation. The constructed QPN model is shown in Figur. By comparing with the QPN model of a generic CR introduced in Chapter 3, it can be observed that they are consistent. It also points to the fact that there is a high probability of QPN models being consistent across multiple technologies. These technology agonistic models are of great value add to the cognitive engine to be relevant for handling heterogeneous networks (Hetnets).

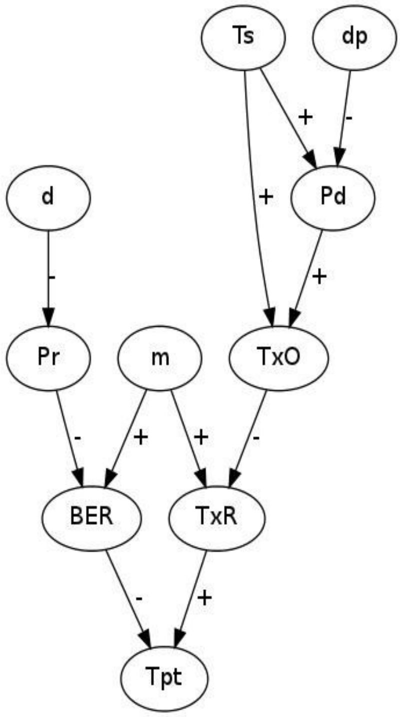


Figure 7.11 QPN Model for the UAB link constructed from observations

7.8 Summary

In this Chapter the design and implementation of a novel communication scheme using upper audio band (UAB) for short range low data rate communication presented. We used a novel multi-tone lattice modulation scheme along with error correction codes to achieve a robust communication. This technology can be used as a low cost, low data rate, short range software communication modem for devices those have integrated audio interfaces. This is envisaged to be used for communications between devices/appliances including cell phones over the air and has several promising applications. Indoor location tracking is one such application. Further it has been shown that Cognitive Radio concepts can be used to operate this UAB link as a secondary in presence of primary users such as TV, Home Entertainment Systems. Further QPN model for the UAB link from the experimental observations is found to be consistent with the generic CR model presented in chapter 3. It gives us a positive feel that these QPN models are technology agonistic and of great value add to the cognitive engine to be relevant for handling heterogeneous networks (Hetnets) of future.

References

[1] Aerial acoustic communications, Lopes, C.V. Aguiar, P.M.Q. 2001 IEEE Workshop on the Applications of Signal Processing to Audio and Acoustics.

[2] Evaluation and Demonstration of Acoustic OFDM, Nakashima, Y. Matsuoka, H.Yoshimura, T. Fortieth Asilomar Conference on ,Signals, Systems and Computers, 2006. ACSSC '06.Oct. 2006.

[3] Acoustic Data Transmission Based on Modulated Complex Lapped Transform, H. Yun, K. Cho, N. Kim, IEEE Signal Processing Letters, Vol 17, Jan 2010.

[4] Sphere Packings, Lattices and Groups, J.H Conway, N. Sloane, Springer, 1998

[5] Detection of transient signals using the energy operator, R.B. Dunn, T.F. Quatieri, J.F. Kaiser, ICASSP 1993.

[6] Lifeng Lai; El Gamal, H.; Hai Jiang; Poor, H.V, "Cognitive Medium Access: Exploration, Exploitation, and Competition," IEEE Transactions on Mobile Computing, vol.10, no.2, pp.239-253, Feb. 2011.

[7] Rahul Sinha, P. Balamuralidhar, Rajeev Bhujade, “An Upper Audio Band based Low Data Rate Communication Modem”, ICSPCS-2012

[8] Rahul Sinha, P. Balamuralidhar, Rajeev Bhujade, “Software defined radio based on the upper audio band for low data rate communications over short distances”, SDR12-WinnComm

8

Conclusions and Future Works

8.1 Conclusions

Imparting autonomic features to communication networks has been the basic driving force behind the concept of Cognitive Networks. The approach to enable cognitive functionality in existing networks is to incorporate a knowledge plane with an embedded cognitive engine. The cognitive engine has the functionality of facilitating autonomic features including self-configuration, self-optimization and self-stabilization to the system. These self-x features along with others can be put in a single basket of self- management. We believe that a systematic approach to architecting the cognitive network should consider the guidance from cognitive science.

In chapter 2 we proposed key guiding principles behind a cognitive entity. One of the strong point emphasized here is that a cognitive entity needs to have a model of 'self' along with a model of the 'world'. Use of ontology is a key aspect here. The persistent nature of this ontology enables pro-activeness and robustness to 'ignorable events' and our arguments on this have been appreciated in several citations of our publication on this. Bringing context awareness through the conceptualization of a context space with necessary information structures is positioned as a key basis for knowledge modeling and management. In the underlying cognitive process, in addition to Sense-Analyse-Decide-Reconfigure (SADR) supported by underlying Learning, the significance of 'Communicate' as a new process element is to be noted. This 'cognitive communication' capability is required to boot strap and synchronize cognitive engines distributed across multiple network elements. Based on the guiding principles of cognition a set of requirements for the cognitive engine were stated and a detailed architecture was developed.

For the representation and manipulation of context space with required information structures graphical models are well suited. By considering the complexity of popular Bayesian Networks (BN) we chose to explore qualitative approaches including sign graphs and qualitative probabilistic networks (QPN). While these qualitative techniques are used in other science streams including biology and chemistry, its use in wireless communication networks is new. In chapter 3 we introduced QPN modeling in the context of wireless networks and developed a QPN model for a generic cognitive radio link. Structurally the QPN is similar to signed directed acyclic graphs (SDAG) which has a good theory behind it for systems analysis. While analyzing the QPN model for the CR link using sign propagation we found that trade-off is a major problem where the inference results in ambiguity. This posed serious limitation on the usability of QPN in our intended application. We found that by the combined use of situation based signs and enhanced signs the problem of trade-off can be addressed to a great extent. Situation specific switching of signs of certain edges is found to be good strategy and this switching often requires certain state measurements and thresholding. Initial experimentations can estimate these thresholds, however they needs to be adapted through learning. While developing the QPN model we came across situations where the causal relation is not monotonic and cannot be assigned a single sign. Example case discussed is based on the behavior of MIMO on SINR and rate. Another situation is the case of coupled controls where there is a coupling between control variables. In both these cases approaches for restructuring the relations are suggested.

We also proposed the use of hierarchical decomposition of larger graphs to smaller sub-graphs for improving the efficiency in addressing the trade-offs and reasoning process. Finally we introduced a methodology to develop an extended QPN model for its use as a knowledge model in the cognitive engine.

One key application of the above dynamic model is in self-optimization. We devised an online optimization algorithm that sequentially exercised the control variables based on reasoning with the QPN model. In case of an ambiguous inference regarding the decision on any control variable a perturbation is made and a local search is initiated. On performance evaluation using a link optimization problem in a dynamic network scenario, it has been observed that the algorithm is giving satisfactory results. Complexity is much less than a simulated annealing based optimizer, also giving better performance. There was a problem anomalous configuration observed occasionally giving sharp deterioration of the goal function during the search process. Since the optimization is on-line, there is no way to know if the configurations are going to be good or bad. For this purpose a case based learning scheme was developed and integrated with the optimization engine to remember such bad configuration boundaries and avoid them at future run time.

In Chapter 5 we extended the scope of the QPN model to cover the network layer aspects. The link level model was enhanced with incorporating the TCP behavior and QoS measures. It also depicted the interaction behavior of multiple nodes in the same access network, and their coupling through mutual interferences. One of the major problems known with TCP is its poor performance of congestion management in wireless networks. We chose to see how the QPN model interprets this aspect and experiment an optimization strategy for a joint congestion control based on the model. A CDMA based adhoc network supporting TCP was considered as a simulation Performance of the QPN model based distributed optimization was analyzed for a few test cases. The results were compared with a state of the art approach and found to be on par. This is really an encouraging performance for a qualitative model with limited knowledge embedding. However it leaves a clear indication on its relevance to model network behaviors and efficient reasoning for various cognitive functionalities.

Beyond self-optimization, self-monitoring is another important capability that a cognitive engine need to facilitate. In chapter 6, architecture for self-healing incorporating fault detection, diagnosis and recovery in the context of using the above cognitive engine is presented. An illustrative example is used to show the flow for the detection and diagnosis using QPN model in the case of an antenna failure. Radio monitoring from the perspective of regulatory compliance is another important aspect of a cognitive radio that can even impact its business case. The requirements to incorporate a radio compliance monitor (RCM) are presented. Further a certification strategy for cognitive radios with embedded RCM is proposed. However in our opinion, it is not easy to support certification of radios that has an unlimited flexibility in terms of dynamism in waveform design, spectrum mask and medium access techniques. The configuration modes need restriction until there is a practical mechanism to identify violators and penalize them.

Chapter 7 introduced a novel and interesting use case with a practical implementation of an acoustic cognitive radio. This addresses the need of a low cost software modem that can be used for short range low data rate device-to-device (D2D) communications. There was a stringent requirement of imperceptibility to human ears has pushed us to choose the upper audio band in the range of 16KHz to 20KHz. Achieving meaningful data rates in this narrow band required us to adopt innovative modulation techniques along with error correction coding for robustness. Further the intended spectrum might be occupied by transmissions from entertainment appliances such Hi-Fi audio systems, HD-TV etc. Therefore our design is to use the cognitive acoustic radio as a secondary user with the above appliances as primary. We have conducted some experiments to understand various behavioral aspects of the radio and use these facts to build a simplistic QPN model. It was observed that the qualitative behavior is same as that of a generic cognitive radio model introduced in chapter 3. This prompt us to argue that the Qualitative models of wireless networks can be technology-agonistic and are of great value add to the cognitive engine to be relevant for handling heterogeneous networks (Hetnets).

We believe that this research work has brought significant insights on the architectural aspects of cognitive engine and applicability of qualitative techniques for knowledge modeling in the cognitive engine for cognitive radios and networks. Many of the information structures that are required for the cognitive engine is possible to be incorporated around the QPN structure and can be an integral part of the context space representation. Further we observe that there is a potential for positioning the network influence graph in the form of QPN, possibly integrated with a standard ontology language, as a knowledge modeling standard for representing the dynamic behavior of cognitive radios and networks.

8.2 Future Research Directions

The applicability of cognitive networking has a rich opportunity and impact in the area of networked embedded systems. The uncertainty, variability, heterogeneity, scale etc are at the major concerns for such systems and cognitive networking has a major role to bring in efficiency and robustness. Moreover the CE as a generic module could be used to enable devices with autonomic capabilities.

The influence graph framework is a good formalism to represent the system dynamics across network, device, application, and user experience in a scalable manner. Graph theory can help in simplifying large graphs through decompositions to get a set of manageable interconnected subsystems.
The QPN and sign graph based Qualitative models and inference mechanisms can be taken forward in many directions. Improving expressability of the model can help in reducing the ambiguity in inferences. This means the incorporation of more semi-qualitative to quantitative information in the structure and thus the complexity. There are several semi-qualitative extensions to QPN that are proposed in literature, that considers various ways of introducing measure of influence strengths. This includes strength measurements in terms order-of-magnitude kappa values, interval probabilities and rough sets. While introducing more precision in the probability values, they pause increasing challenges in getting these information from human experts. Alternately there are schemes to estimate them from training data. Evaluating and refining these metrics as candidate representations is an exercise to be done.

Some of the specific enhancements suggested for QPN based modeling of Cognitive networks are:

- Use of semi-qualitative influence information in the form of intervals, rough sets
- Exploration of the use of synergetic influences
- Non-symmetric influence strengths
- Hybrid models with deterministic nodes, expressions evaluating the influence strengths
- Systematic decomposition approaches for large networks
- Distributed algorithms using QPN models
- Integration of serial-parallel inference algorithm
- Analysis and approach for global optima

Another direction is to look at QPN as the stepping stone for a more powerful mechanism of BN. As the CE gathers experience from the network, the QPN can be refined and enhanced to a full-fledged BN with the learned conditional probability measures. This enhancement need not be done at a global level, but at sub-graph level as per the performance requirement.

Another requirement is to consider the dynamism of the ‘world’ variables that may have different time scales. In QPN model, the environment variables can be given an additional attribute on an expected time before a change. This information is useful in planning adaptation loops in multi-time scale based on the dynamics involved. Suitable markov models can be embedded in those nodes to learn and predict this additional attribute.

Extending the monotonicity analysis to identify possible monotonic decompositions for a large network and integrate it as a cascaded system of monotonic subsystems will be very useful from a control theory perspective to analyze and ensure stability of various network adaptations.

Integration of context space structure in the network ontology that is represented in a standardized ontology language can be a potential standard in itself. This will also have a compact graph based representation that can be embedded inside network elements with constrained resources.

Test and validation of cognitive network system is an important aspect to be considered. There is a potential for designing systematic test and validation process guided by the context space structure. Moreover the cognitive engine can use the QPN models to plan conduct self-tests autonomically.

Further we believe that the use of influence diagrams to explain the dynamics of wireless networks to students has a great potential to contribute towards wireless standards education. Several professors have supported this view and suggested to include them in text books and even in wireless standards documents.

"Cognitive" as a design paradigm is being explored in many areas such as pervasive computing, vehicular communications, networked embedded systems etc. The strength of cognitive framework is in its systematic knowledge management and cognitive meta process cycle. It has developed very well in the context of autonomic communications and spreading in other fields as well.

Appendix A

Context Interpretation using Ontology

A1 Contexts and Ontology

A context model is needed to define and store context data in the required machine-processable form. A survey by Strang et al[1] classified the most relevant context modeling approaches. The conclusions of their evaluation show that ontologism is the most expressive model and fulfills most of their six requirements for context-aware systems.

Ontologies allow machine-processing of context data by enabling formal reasoning over them. There are wide varieties of languages for explicit specification of context information. The graphical notations include UML and RDF while the (description) logic-based notations include DAML+OIL, OWL etc. While the concepts and their relations are captured using ontologies, the actual instantiated-from-concepts context data is specified using notations such as RDF. Typical features of such instances can be found in [2]. OWL Language is an extension of RDF.

While the context information can be stored and represented using ontologies, the “intelligence” in the context aware system is implemented by context interpretation engine, which reasons over the context information. Classically, context and ontology based methodologies have been well explored for performing interpretation to systems predominantly at application level in the semantic web domain. But while mapping, the approach could be extended further to cover all other entities in a cognitive network by integrating information and interaction of radio resources, communication protocols and user actions.

A2 Context Interpreter (CI) Architecture

The detailed architecture evolved out of mapping context interpretation function which can be incorporated as a part of the cognitive management entity shown in Figure A.1

Raw context sensor data is pre-processed to remove outliers in terms of data formatting, noise etc. Suitable data abstraction techniques such as aggregation and fusion are applied on the pre-processed data to extract higher level information elements as defined by the Ontology. Validation of the data is done through appropriate consistency check with the ontological model and stored in the knowledgebase. Some of the knowledge representation can be done through rule base as well. This requires the context interpretation engine to include both ontology based reasoner as well as rule based reasoner. Typically events are easier to be detected by rules and situations are identified by ontology.

Appendix A - Context Interpretation Using Ontology

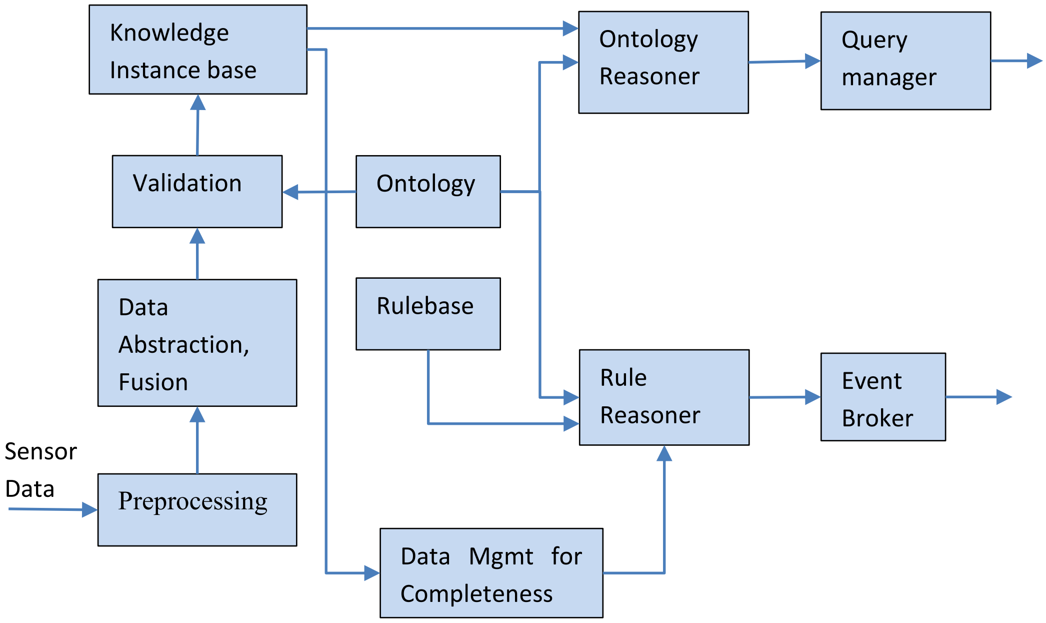


Figure A.1 Architecture for Context Interpretation

A3 Prototype

Using various semantic web tools and technologies, an initial proof-of-concept system has been implemented to study and understand its feasibility of incorporating context interpretation as required by a cognitive engine. The system was implemented using a tool-chain shown in Figure A.2. OWL was chosen as the ontology language, RDF as the context data language, while SWRL was chosen as the rule language. Being extension of OWL, it is easy to store SWRL rules with OWL, RDF information. The popular query language to RDF database is SPARQL, and we chose this language for our use.

Appendix A - Context Interpretation Using Ontology

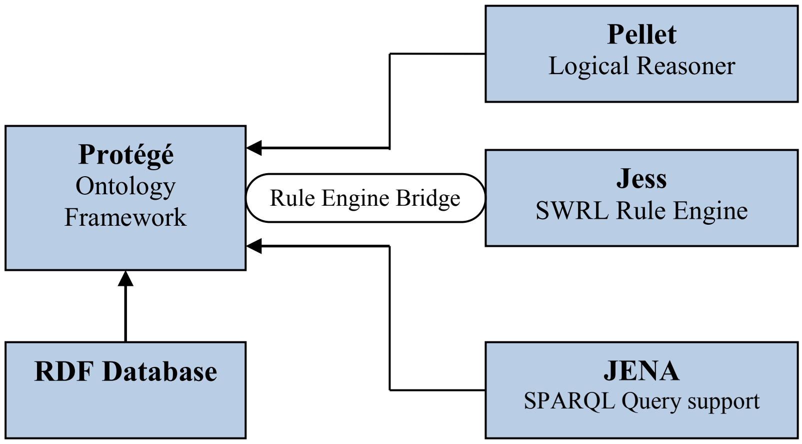


Figure A.2 Architecture of Ontology based Context Interpretation Prototype

Protégé-OWL plugin was used as the basic CI framework. Protégé is based on Java, and it supports OWL-DL. It has a built-in knowledge base container. Protégé allows editing and storage of SWRL rules in the knowledge base. It supports rule engine integration, especially with JESS, and also attaches to various OWL reasoners easily, such as Pellet or RacerPro.

We chose Pellet as ontology reasoner for ease of integration as well as license issues. Similarly, JESS was chosen as the rule reasoner. To support SPARQL query, we used JENA framework. The entire integration work was carried out in Java language.

A3.1 An Example Use Case Scenario

Following representative use case scenario was chosen to evaluate the above CI architecture.

This scenario considers the work of Rahul, a sales executive of a retail company who has to travel frequently between multiple locations and use multi-modal transport for the travel. For his enterprise network connectivity he uses a laptop that has 3G cellular data connectivity, a GPS based location system, and an RFID affixed for asset management. A context management agent sitting on his laptop takes care of identifying his situation and trigger appropriate application reconfigurations on the laptop.

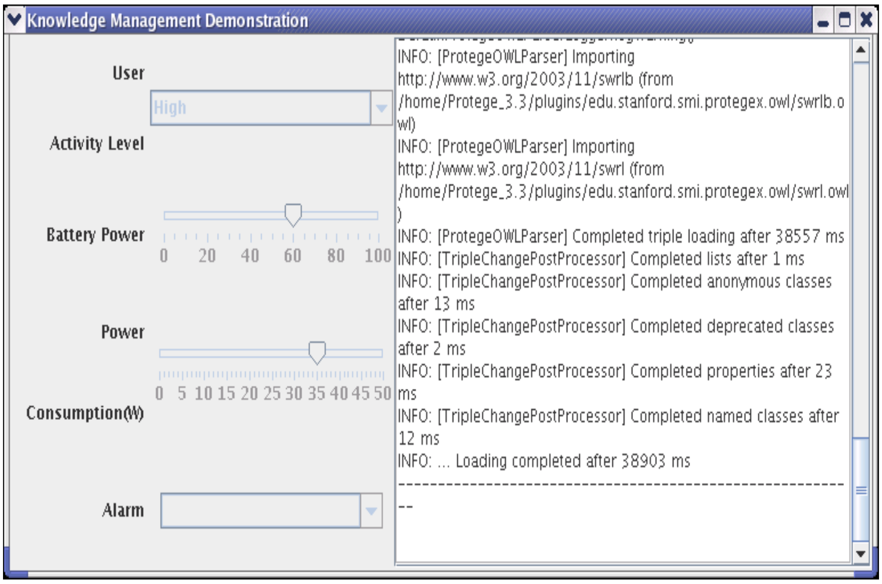
When Rahul enters his office and hooks his laptop to the company's network, some of his enterprise applications are automatically enabled. The enterprise presence server is updated with his location information.

Next, Rahul goes for a trade fair by catching a train. While on train, he connects his laptop to 3G network and launches an enterprise application to complete certain workflows. In addition there are a few other applications also launched including a song streamed from a server, download of an ebook and web browsing. With the present application load, the estimated laptop battery life is 2 hours but his travel time to reach the destination is 4 hours. Also he is not yet aware of the fact that his train does not have a facility for battery charging. But, for his consolation, the context management entity detects the situation and advises him to take necessary steps and in certain cases takes proactive steps to reconfigure the applications by itself.

A3.2 Context Interpretation for the Use Case Scenario

The CI prototype described in section 7.3 has been configured for the use case scenario described in the previous section. The implementation was tested for various events and situations for expected context interpretation as outlined in the use case. A screen shot of the GUI is shown in Figure

First task was to build the ontology that describes the needed concepts, relations and properties pertaining to this specific use case. This included entities like Employee, User terminal, laptop, battery, battery charging, Software Applications, Travel, vehicle, train, location, mobility etc. There are sensors to gather data on location, laptop battery charge level, battery drain rate, network QoS, Application status etc. Further the data sources like user's activity plan, travel itinerary, train schedule etc are also consulted. Subsequently various event rules and rules for data abstractions are defined. This include the detection of battery low condition, condition to be maintained for sustained operation of the laptop (ie battery life time should be greater than the time to connect to an alternate power source), user mobility, travel mode, estimating battery life time and time to get alternate power source etc. Details of this implementation and evaluation are presented in our publications [3][4]. Various situations could be specified by setting appropriate context variables through a GUI and the interpreted events are observed to reflect the intended use case scenario.



User Activity Level: High

Battery Power

Power Consumption(W)

Alarm

```
INFO: [ProtegeOWLParser] Importing
http://www.w3.org/2003/11/swrlb (from
/home/Protege_3.3/plugins/edu.stanford.smi.protegex.owl/swrlb.o
wl)
INFO: [ProtegeOWLParser] Importing
http://www.w3.org/2003/11/swrl (from
/home/Protege_3.3/plugins/edu.stanford.smi.protegex.owl/swrl.owl
)
INFO: [ProtegeOWLParser] Completed triple loading after 38557 ms
INFO: [TripleChangePostProcessor] Completed lists after 1 ms
INFO: [TripleChangePostProcessor] Completed anonymous classes
after 13 ms
INFO: [TripleChangePostProcessor] Completed deprecated classes
after 2 ms
INFO: [TripleChangePostProcessor] Completed properties after 23
ms
INFO: [TripleChangePostProcessor] Completed named classes after
12 ms
INFO: ... Loading completed after 38903 ms
---------------------------------------------------------
--
```

Figure A.3 Prototype developed for the context interpretation framework

A3.3 Discussion

The prototype was tested by us for some use cases for functional as well as performance studies. The first performance bottleneck was found in the time taken for repeated validation of knowledge base, whenever an update is made to the knowledgebase. This is addressed by changing the runtime parameter of Pellet to make it do a quick, incremental reasoning. We experimented with iterations over rule-based reasoning, but could not gain clarity over when to stop.

The time-performance for our medium-sized ontologies drastically increased. The Protégé-JESS bridge exports only those facts that are relevant to the set of rules being executed, and hence speed was satisfactory. Overall, the time taken at various steps for a database of 23 concepts, 39 properties and 149 concept/property instances was found to be of order of 100 ms on a Pentium-IV processor-based desktop.

One of the major issues with this CI implementation is the large memory requirement of such an architecture which might be infeasible for a handheld cognitive device to provide. This may not be the issue with architecture, but relates to the tool chain involved in the prototyping. An alternative could be to provide this interpretation functionality by a more powerful network device as a service.

A4 Conclusions

Ontologies are indeed a good mechanism for providing robust interoperable context management and provisioning to drive the cognitive process for an intelligent terminal. But the concern is in implementing the same in the constrained embedded environment of a network element such as handheld terminal. But the use of a relatively powerful network device can support this and it could be outsourced with the CI functionality required by attached cognitive terminals in the network.

References

[1] T. Strang and C. Linnhoff-Popien, “A Context Modeling Survey”, Sixth International Workshop on Advanced Context Modelling, Reasoning and Management, September 2004.

[2] M. Baldauf et al, “A survey on Context-aware Systems”, International Journal on Ad-hoc and Ubiquitous Computing, Vol.2, No. 4, 2007

[3] Hrishikesh Sharma, Balamuralidhar P, “A Context Interpretation framework for Cognitive Network devices”, International Conference on Software Defined Radios, SDRF, Denver, USA, Dec 2007

[4] Hrishikesh Sharma, Balamuralidhar P, “Application of Semantic Web technologies for Context Interpretation in Cognitive Communication Devices”, WWRF – 19 conference, 5-7 Nov 2007, Chennai, India

Appendix B

Balamuralidhar P - Resume

Balamuralidhar P obtained his Bachelor of Technology (BTech) in Electrical Engineering from Kerala University (1980-1985) and Master of Technology (M.Tech) in Electrical Engineering from Indian Institute of Technology, Kanpur (1985-1987). He has over 24 years of research and development experience in Digital Signal Processing, Embedded Systems and Wireless Communications. He has over 50 publications in various international journals and conferences and over 20 patents filed for inventions in related disciplines.

From 1987 to 2000 he was with Society for Applied Microwave Electronics Engineering & Research (SAMEER), as a Scientific Officer. He was responsible for the development of digital signal processing system of first Indian MST Radar. Subsequently he was leading an R&D group in Signal processing and Information Technology.

In Sasken Communication Technologies, Bangalore (2000-2003), he lead two major programs in wireless communications towards development for reference designs for 3G-WCDMA and IEEE 802.11g baseband.

Balamuralidhar is currently a Principal Scientist and heading TCS Innovation Lab at Tata Consultancy Services Ltd (TCS), Bangalore. Major areas of research are in Networked Embedded Systems, Intelligent Sensing & Communications and Internet of Things. He was the lead of TCS participation in two EU FP6 research consortium programs namely “My Adaptive Global NET” (MAGNET) and “End to End Reconfigurability (E2R)” in the area of next generation wireless communications.

Balamuralidhar is the Chair of the “Internet of Things Workgroup”, in Global ICT Standardization for India (GISFI). He has given Keynotes, Invited talks and Panelist in several conferences and workshops including CTIF Annual Workshops, ANTS, EmTech, ICRTIT, UWorld, ICoAC. He is also in the technical program committee member of several conferences/workshops including Globecom and ICC. He is in the advisory panel of India ICT Vision-2035 constituted by TIFAC. He is also a member of IEEE and CII.

Appendix C

Mandatory Declaration

Thesis Title : Architectures and Algorithms for Cognitive Networks Enabled by Qualitative Models
Name of PhD Student : Balamuralidhar P
Name and Title of Supervisor : Prof. Ramjee Prasad, Department of Electronic Systems, CTIF, Aalborg University
List of Published Papers:

1. Balamuralidhar P, Ramjee Prasad, “Self-Configuration and Optimization for Cognitive Networked Devices”, Springer Journal on Wireless Personal Communications, 2011, DOI: 10.1007/s11277-011-0240-8
2. Balamuralidhar P, Rajan M.A, “Signed Graph based Approach for On-line Optimization in Cognitive Networks”, Proceedings, COMSNETS-2011, Third International conference on Communication Systems and Networks, 4-8 January 2011, Bangalore India
3. Balamuralidhar P, “Exploring Qualitative Probabilistic Networks for Knowledge Modeling in Cognitive Wireless Networks”, IWCMC-2013
4. Hemant K Rath, Rajan MA, Balamuralidhar P, “Monotonic Signed Graph Approach for Cross-layer Congestion Control in Wireless Ad-hoc Networks”, GlobeCom 2011.
5. Balamuralidhar P, Ramjee Prasad, “A context driven architecture for cognitive radio nodes”, International Journal of Wireless Personal Communications (Springer), Vol. 45 (3), May 2008, pp 423-434
6. Rahul Sinha, P. Balamuralidhar, Rajeev Bhujade, “An Upper Audio Band based Low Data Rate Communication Modem”, ICSPCS-2012
7. Rahul Sinha, P. Balamuralidhar, Rajeev Bhujade, “Software defined radio based on the upper audio band for low data rate communications over short distances”, SDR12-WinnComm, 2012.
8. Balamuralidhar P, Ramjee Prasad, “A Programming Paradigm for Cognitive Radios”, International Conference on Wireless Personal Multimedia Systems (WPMC07), Jaipur, Dec 4-6, 2007.
9. P.S.Subramanian, Balamuralidhar P, “Intensional description of autonomic use cases”, ”, ICT ’06, 11-12 May 2006, Funchal, Portugal
10. Balamuralidhar P, Ramjee Prasad, “ A Radio Compliance Monitor Architecture for Enabling Regulatory Certification for Cognitive Radios” , WPMC06 , San Diego, U.S.A. from September 17 to 20, 2006
11. Hrishikesh Sharma, Balamuralidhar P, “A Context Interpretation framework for Cognitive Network devices”, International Conference on Software Defined Radios, SDRF, Denver, USA, Dec 2007
12. Hrishikesh Sharma, Balamuralidhar P, “Application of Semantic Web technologies for Context Interpretation in Cognitive Communication Devices”, WWRF – 19 conference, 5-7 Nov 2007, Chennai, India

Appendix C Mandatory Declaration

This thesis has been submitted for assessment in partial fulfillment of the PhD degree. The thesis is based on the submitted or published scientific papers which are listed above. Parts of the papers are used directly or indirectly in the extended summary of the thesis. As part of the assessment, co-author statements have been made available to the assessment committee and are also available at the Faculty. The thesis is not in its present form acceptable for open publication but only in limited and closed circulation as copyright may not be ensured.