Towards Reliable Integrated Services for Dependable Systems

Schiøler, Henrik; Ravn, Anders Peter; Izadi-Zamanabadi, Roozbeh; Nielsen, Kirsten Mølgaard; Madsen, Ole Brun; Nielsen, Jens Frederik Dalsgaard

Publication date:
2003

Document Version
Publisher's PDF, also known as Version of record

Towards Reliable Integrated Services for Dependable Systems.

Henrik Schiøler
Roozbeh Izadi-Zamanabadi
Ole B. Madsen

Aalborg University,
Dept. of Control Engineering
henrik@control.auc.dk

Anders P. Ravn*
Kirsten Nielsen
Jens Dalsgaard

Aalborg University,
Inst. for Computer Science*
apr@cs.auc.dk

Abstract

Reliability issues for various technical systems are discussed and focus is directed towards distributed systems, where communication facilities are vital to maintain system functionality. Reliability in communication subsystems is considered as a resource to be shared among a number of logical connections and a reliability management framework is suggested. We suggest a network layer level reliability management protocol RRSVP (Reliability Resource Reservation Protocol) as a counterpart of the RSVP for bandwidth and time resource management. Active and passive standby is discussed as well as utilization of passive standby redundancy by background applications residing on alternative routes. Details are provided for the operation of RRSVP based on reliability slack calculus. Conclusions summarize the considerations and give directions for future research.

1. Introduction

Dependable real time systems embrace a variety of application fields from automotive systems to modern large scale SCADA systems. Increase in bandwidth and decreasing prizes drive associated communication subsystems towards less technological diversity. From an applicational/operational view point technological homogeneity is desirable since it furnishes flexibility as well as maintainability. However since technological homogeneity in SCADA communication systems is not a *fait accomplis* and is hardly accomplished in the near future it is natural to focus upon convergence at a higher level. It is well known that internet usability and success owe to the existence of the IP protocol unifying diverse technological platforms, as recognized in the telecommunications community, where IP is adopted for future value added services, e.g. voice and multimedia. In order to encompass the large variety of time related QoS demands, the Integrated Services (IntServ) [1] is suggested as a means for specifying and manageing end-to-end service qualities. Even value added internet services are not likely to be considered highly dependable as for instance alert or real time control traffic in transport systems or nuklear power plants. With the newly released standard for functional safety (IEC 61508) [4] system suppliers face an inevitable challenge in providing confident SIL (Safety Integrity Level) classifications for components and systems, including communication subsystems. In this work we suggest a novel framework for reliability management embedded in IntServ as a natural convergence point for dependable systems.

2. System reliability

The proper working of a complex system relies on the failure state of the set of components it comprises.

2.1. Component reliability

Reliability $R(t)$ is defined as the probability of lifetime above some time duration t since its latest renewal. For exponential lifetime distributions the latest renewal instant is where the system was last known to be alive. Thus for exponential life times no ageing occurs, and reliability measures are uniquely defined by the failure rate λ, i.e. $R(t) = exp(-\lambda\ t)$. Alternatively Mean Time Between Failure $MTBF = \frac{1}{\lambda}$ is given. For irreversible systems, where repair is impossible, reliability $R(t)$ and/or failure rates λ appropriately characterize component reliability. When repair is possible the so called *availability* indicating the uptime of the component may be of higher relevance. Availability is generally given by $A = \frac{MTBF}{MTBF+MTTR}$, where MTTR is Mean Time To Repair. Component providers may be required to provide lifetime data along with the components themselves or system providers record component failures and maintain relevant statistics. For network communication components reliability levels may depend on product type and prize as well as the applied technology, e.g. whether wired or wireless communication is used. Failure rates for physical communication links tend to increase linearly with distance because vulnerability to breakage is measured per length unit. Environmental factors influence link failure rate as well, e.g. whether physical links pass through areas with heavy mechanical or thermal activity.

2.2. System and Subsystem Reliability

An overall assumption is, that systems are divided into independently failing components. In our, case communication networks are divided into physical links connected by routers. Independent components in seriel configuration, like

links $\{l_i\}$ in a communication route r therefore obey the following rules for various reliability measures

$$R_r(t) = \Pi_{i=1}^{n} R_i(t) \quad (1)$$

$$\lambda_r = \sum_{i=1}^{n} \lambda_i \quad (2)$$

$$A_r(t) = \Pi_{i=1}^{n} A_i(t) \quad (3)$$

where subscripts $_r$ and $_i$ indicate quantities associated to the overall route r and the ith. link respectively. For simplicity only link failures are considered. For the failure rate equation (2) component lifetimes are assumed to be exponential. In that case the lifetime of the seriel configuration is itself exponential. For components in parallel the following rules apply

$$1 - R_r(t) = \Pi_{i=1}^{n}(1 - R_i(t)) \quad (4)$$

$$\frac{1}{\lambda_r} = \sum_{i=1}^{N} \frac{1}{\lambda_i} \quad (5)$$

$$1 - A_r(t) = \Pi_{i=1}^{n}(1 - A_i(t)) \quad (6)$$

where exponential component lifetimes are assumed in (2). In equations (4) and (5) no repair is assumed, which yields rather conservative results, when repair is possible. In (3) and (6) independent repair is assumed.

2.3. Appropriate reliability measures

In repairable systems like non safety critical network connections or office computer equiptment availabilities are appropriate system reliability measures. Recognizing that fault frequency issues may play a role w.r.t. system failures we shall however in the following consider relatively long fault durations In other words we consider a link breakage model rather than a package loss model. Also we assume individual link MTTRs to be of comparable size. In safety critical systems, reliabilities $R(t)$ or associated MTBF are appropriate system reliability measures. However such systems may comprise replication, i.e parallel configurations of repairable/replacable components, in which case (4) and (5) yield overly conservative results. Therefore we model reliability for safety critical systems also through component and subsystem availabilities and only finally transforming the system availability A into a corresponding system MTBF as shown in equation (7)

$$MTBF = MTTR\,\frac{A}{1-A} \quad (7)$$

where a lower (conservative) bound MTTR is used.

3. Reliable Network Communication

For distributed safety critical systems or in general systems sensitive to reliability (StR-systems) best effort IP services typically do not suffice. Much like the case for distributed real time applications, where best effort end-to-end timing is insufficient, StR-systems require a prespecified level of end-to-end reliability. For real time applications 2 different solutions are suggested on a network layer level; Integrated Services (IntServ) [1] and Differentiated Services (DiffServ) [3]. The latter prescribes how to distinguish services in routers, based on labelling IP traffic in a number of service classes, so service under DiffServ is therefore still to be considered as best effort. The former however suggests a genuine network wide distributed resource management, where timing is negotiated during route setup. In the Resource Reservation Protocol (RSVP) [2] suggested for IntServ every router on a route candidate computes an available throughput rate and its maximum deviation from fluid service, based on local resource balances and scheduling policies. At each hop, flow rate and burst parameters are passed on to the next router in the path. At the destination, required and available rates can be compared along with delay bounds and end-to-end delay. When an StR application requests a reliable service from a service provider a lower reliability bound is requested along with timing related QoS requirements. Provisioning reliable service requires the establishment of a route with at least the required reliability and timing related service qualities. In order to reduce stress on central management facilities and in turn ensure scalability a distributed solution should be preferred. Since service prize should reflect the provided quality, profitability of reliable service provisioning clearly depends on the ability to provide exact service qualities, i.e. not overproviding. Our reliability model of network communication follows the general outline above. A single network route is a serial configurations of components; links, router, switches and power suplies. For simplicity we illustrate ideas by only considering a link breakage model.

4. Reliability Resource ReSerVation Protocol (RRSVP)

Consider the first and simplest case, where an application requests a reliable route to some destination. A first approach is to find the most reliable route through a network of links each equipped with an availability label. We consider route availablity $A(r)$ as a decreasing utilization function, i.e. $C(r) = A(r) \geq A(r\ o\ l) = C(r\ o\ l)$, where r is some route and $\cdot\ o\ \cdot$ denotes concatenation. Route discovery can then be considered an optimal pathfinding problem, with well known algorithmic solutions. However a distributed solution is required, which gives rise to the following *source initiated on demand route request protocol*.
Protocol 1 (Request part):

- When a route is requested a RREQ message is transmitted from the source to every neighbouring node. The RREQ message contains source and destination adresses as well as a sequence number unique to the source and an availability/utility value initially set to 1 by the source. We say that RREQ messages with identical source adresses and sequence numbers are *equivalent*.
- When some node different from the destination recieves a RREQ message on link l', it compares its utility value with a potentially stored value from a previously recieved equivalent message.
- If the utility of the recieved message is higher than the stored value, the recieved value is stored instead along

with l'. In this case new equivalent RREQ messages are forwarded on all links l except l'. The forwarded RREQ messages have utility values updated with the avilability of l, i.e.

$$C(M') = C(M)\ A(l) \tag{8}$$

where M is the RREQ message received on l' and M' is the RREQ message forwarded on link l. In this way every RREQ message carries exactly the availability of the route it travelled so far.

- If the received utility is below an equivalent stored value no further action is taken.
- If no equivalent RREQ was previously recieved the utility value and reception link is unconditionally stored and new equivalent RREQ messages, with updated utilities are forwarded on all links except the reception link.

We may prove the following lemma for optimal availability routes:
Lemma 1: *Every optimal route r from S to N equals $r'\ o\ l$ where r' optimally connects S and n and l is a link from n to N*

so that for protocol 1 defined above:
Lemma 2:
Every node N recieves a RREQ message carrying the availability/utility of the route from destination to N with the highest availability, i.e. an optimal RREQ message.

Since every node recieves an optimal RREQ message it eventually stores the link leading to it through the optimal path, which is finally confirmed by the destination node in the reply part of protocol 1.
Protocol 1 (Reply part):

- When the destination has assured itself that it recieves no further equivalent RREQ messages it backwards a Route Reply (RREP) messages carrying source and destination adresses as well as sequence number on the link on which it recieved the optimal RREQ.
- When some node recieves a RREP it belongs to an optimal route connecting the accompagnied source and destination. It therefore stores the link on which it recieved the RREP along with the source adress and sequence number. In this way RREP messages travel backwards along the optimal route and eventually reach the source. Every node on the optimal route has then stored the link leading forward in the optimal route associated to source adress and sequence number.

After route establishment data is transmitted along the route following the stored links. All together we may state the following theorem valid for protocol 1 :
Theorem 1:
Data are eventually transmitted along the route connecting source and destination with the highest availability.

However this may lead to overly reliable routes and in turn reduced profitability. We suggest to attach a cost $J(l)$ to every link typically increasing with the availability of that link. Thus an overall utility funtion C is defined reflecting both the obtained availability as well as link cost

$$C(r) = A(r) - K \sum_{l \in r} J(l) \tag{9}$$

where K is a positive constant. With the presented definition of C optimal routes are forced not to have overly high availability. In this case it should be noted that RREQ message should carry additional values $\sum_{l \in r} J(l)$ (for the route r so far) as well as the constant K. With C defined as in equation (9) it is ensured to be increasing by concatenation. However lemma 1 is generally not valid, since in this case $C(r) < C(r')$ does not necessarily imply $C(r\ o\ l) < C(r'\ o\ l)$. Thus optimal routes do not in general have optimal subroutes. When lemma 1 is valid, we say that the corresponding utility function is *triangular*. For non triangular utility functions RREQ messages need to carry complete information of the route travelled so far. In that case we suggest the following protocol:
Protocol 2 (Request part):

- When a route is requested, a RREQ message is transmitted from the source to every neighbouring node. The RREQ message contains source and destination adresses as well a sequence number unique to the source, hop counter, a maximum hop value H, and a utility value initially set to 1 by the source.
- When some node different from the destination recieves a RREQ message over a link l', it checks whether the maximum hop count is reached. If not, a RREQ message is created for every outgoing link l except l', where the hop counter is increased, availability and accumulated link costs are updated according to (8) and (9), and the node adress is inserted immediately before the destination adress.

With protocol 2, a RREQ reaches every node n reachable in H hops, through every route of H hops or less leading to n. Every RREQ message carries the utility of the route travelled so far and adresses of all intermediate nodes. Protocol 2 ensures optimality even for non triangular utility functions. However the generated number of messages may be significantly higher than for protocol 1, and the size of RREQ messages is significantly higher, since intermediate node adresses are included in the messages. An alternative approach is to redefine the utility function in (9) recursvely by:

$$\begin{aligned} C_R(e) &= 1 \\ C_R(r\ o\ l) &= A(l)\ C_R(r) - K\ J(l) \end{aligned} \tag{10}$$

For availabilities $A(l_i)$ close to 1 and routes of moderate length C_R closely approximates C. Moreover C_R is triangular, so protocol 1 provides optimal routes for C_R. If only RREQs with insuffient availablity reach the destination it may issue a Route Retry (RRET) message to the source proposing a retry with a lower value of K. If no sufficiently reliable route exists replication by alternative routing is required.

4.1. Alternative Routes

We assume that one of the protocols above have lead to an insufficiently reliable primary route candidate r. In that case an

alternative route r' should be found. The route request procedure should ensure sufficient independence between r and r', i.e. they should not share too many common links. For two routes r, we compute the utility $C(r') = A(r \vee r')$ by

$$C(r') = A(r) + A(r') - \frac{A(r)A(r')}{A(I)} \qquad (11)$$

A so defined utility function C allows local update but is non triangular, so only protocol 2 provides optimal routes. We may modify the utility function C in (11) by bounding the dependency ratio $\frac{A(r)}{A(I)}$ between r and r' by some positive constant $0 < \gamma < 1$. Such a bound may be established by realizing that $A(r') \leq A(r)$ and requiring $C(r') > A_R$. Which in turn gives $\frac{A(r)}{A(I)} < 2 - \frac{A_R}{A(r)} = \gamma$. We modify the protocols above to block any routes for which $\frac{A(r)}{A(I)} > \gamma$ by carrying along $A(I)$ in RREQ messages. Along with such a blocking we suggest to use the utility function $C_R(r')$ as defined in (10) with protocol 1, where $J(l) = 0$ for $l \in r$. The rationale behind such a choice is, that the resources used on link l should not be payed twice. All together the resulting protocol tends to produce route candidates, where reuse of primary route links is preferred until some upper limit.

4.2. Passive Standby

When applications accept gracefull service degradation, QoS contract may comprise a list of acceptable services ordered according to quality along with associated reliability levels decreasing with quality. Acceptable service degradation allows utilization of standby resources by less dependable services requesting *background* routes r'' with low but specified reliability requirements and available to a lower cost. The route discovery protocol needs to take into account the risk of failure on primary routes initiating standby transmission on some link of r'' and in turn blocking r'' or at least degrading service. We consider here only blocking. If a primary route r fails, then any backround route r'' partly coinciding with the associated standby route r' is blocked. Proper working of some background route r'' thus relies on all of its own links along with the entire set $L(r'')$ of links of primary routes blocking r'' by failure. Thus

$$C(r'') = A(r'') = A(L(r'')/r'')\ \Pi_{l \in r''} A(l) \qquad (12)$$

As such C is non triangular so only protocol 2 provides optimal routes. Local update of C requires all primary routes to be stored on every node visited during route discovery, as well as every RREQ message to carry unique identification of all primary routes blocking links previously visited. To reduce the complexity we suggest a conservative approximation assuming all blocking primary routes to be disjoint, that is they fail independently. Then utility function updates may be performed locally by

$$\begin{aligned} C(r\ o\ l) &= C(r)A(l) \text{ for } R(l) \subseteq R(r) \\ &= C(r)A(l)\ \Pi_{r \in R(l)/R(r)} A(r) \text{ otherwise} \end{aligned} \qquad (13)$$

where availability of each blocking route is only accounted for once. In (13) $R(\cdot)$ denotes the set of primary routes blocking the argument. Still only protocol 2 ensures optimality and blocking routes should still be carried along in RREQ messages. However in this case primary routes should only be stored in the nodes they block. Due to space limitations we do not consider policies for alternative route discovery taking into account previously allocated background routes.

5. Conclusion

We have argued the need for reliability management in computer communication systems. Inspired by the RSVP protocol for Integrated Services in IP networks we suggest a Reliability Resource Reservation Protocol (RRSVP), where reliability measures are carried through in Route Request messages and modified at each intermediate station and forwarded along route candidates. A number of reliability measures; reliability, availablity and failure rate are presented, their relevance to different systems types is discussed and we suggest availability as a common measure for various system types. We suggest a basic source initiated on demand route retrival protocol based on passing updated route availability along on route candidates. Only if Route Request (RREQ) messages of higher availability are recieved they are forwarded. The basic protocol is proved to produce optimal routes when route utility functions are *triangular*. Since routes of highest availability are not allways required we augment availability with link costs. However the so defined utility function is non triangular and the basic protocol does not provide optimal routes. Two solutions to the problem are presented; a triangular approximation of the utility function and a more elaborate protocol carrying intermediate station adresses along route candidates. We define an appropriate utility function for alternative routing based on the assumption that a primary route of insufficient reliability has been previously retrieved. Finally the case for passive standby routing is discussed along with protocol complexities. Future research includes prototype implementations of the suggested protocols and performance evaluation regarding the load of management traffic generated.

6. References

[1] R. Braden, D. Clark, S. Shenker, IETF RFC-1633, Integrated Services in the Internet Architecture: an Overview., RFC of the Internet Engineering Task Force,IETF,1994.

[2] R. Braden, Ed., L. Zhang, S. Berson, S. Herzog, S. Jamin, IETF RFC-2205, Resource ReSerVation Protocol (RSVP) – Version 1 Functional Specification , RFC of the Internet Engineering Task Force,IETF,1997.

[3] D. Grossman, New Terminology and Clarifications for Diffserv RFC of the Internet Engineering Task Force,IETF,2002.

[4] IEC61508(7 parts), Functional safety of electrical/electronic/programmable electronic safety-related systems, Standard of the International Engineering Consortium, IEC,Geneva,1998-2000.