# Enhancing Resilience for High Availability IP-based Signaling Transport

René Rembarz[1], Stephan Baucke[1] and Petri Mähönen[2]

[1] Ericsson Research, Aachen, Germany. {rene.rembarz, stephan.baucke}@ericsson.com
[2] Department of Wireless Networks, Aachen University, Germany. pma@mobnets.rwth-aachen.de

***Abstract*—With network operators migrating from circuit-switched networks to IP networks, the need for a concept to provide operator grade trunk signaling over IP arises. A solution to enable signaling transport over IP is defined by the IETF SIGTRAN framework that allows for interworking with the Signaling System No. 7 (SS7). SIGTRAN is based on the Stream Control Transmission Protocol (SCTP) that also has a built-in redundancy mechanism, which allows switching to an alternative destination address (failover). Existing work showed that a failover performance comparable to SS7 can be achieved by aggressively setting the SCTP protocol parameters. However, this brings about possible stability issues because these parameter settings seriously impair the protocol mechanisms responsible for reacting to delay variations and congestion situations.**

**We propose a set of network redundancy mechanisms that resolve most network failures transparently to SCTP. In order to eliminate possible stability issues, we use the default SCTP protocol parameters. Simulation results confirm that the proposed architecture can noticeably improve the robustness to concurrent network failures and achieve a similar delay behavior without jeopardizing the protocol stability.**

***Index Terms*—Transport protocol, Communication system signaling, Network reliability, Redundancy**

## I. Introduction

In contrast to most datacom networks, today's telecom networks consist of separate transport and signaling planes. Signaling within the core network is widely based on the Signaling System No. 7 (SS7) [1]. This system allows for the reliable transport of signaling messages over a TDM-based network. One key advantage of SS7 is the ability to quickly resolve failure situations. With its Changeover mechanism, rerouting after a failure is usually completed within 500 to 2000. Ample over-provisioning of link capacities and a high degree of hardware redundancy assure that performance degradation in presence of a failure situation is minimal with SS7.

With the trend to migrate from circuit-switched to IP-based networks, the need for a solution for signaling over IP arises. As the migration cannot happen overnight, an additional requirement to such a concept is to provide interoperability between old TDM-based and new IP-based systems.

The IETF SIGTRAN framework [2] offers such a solution. It is based on the Stream Control Transmission Protocol (SCTP) [3] that is closely related to TCP and allows for reliable transmission of signaling messages over an IP network. A set of User Adaptation Layers (UALs) works as a mediator between SCTP and the legacy signaling applications, providing the same interfaces as the corresponding SS7 layers. Thus, the signaling applications remain unaware over which medium the signaling messages are transported. This is a key factor for seamless migration and interworking between TDM and IP-based systems. Of course, it also brings about the requirement for IP networks to perform similarly to SS7.

In order to increase resilience in IP networks, a key innovation of SCTP is the ability to switch to a different destination transport address in case the current one becomes unreachable (failover). With the default protocol parameters [3], a failover is initiated after more than five consecutive timeouts of the retransmission timer for the current destination address. With a minimum timeout ($RTO_{min}$) of one second and the exponential backoff strategy adopted from TCP, an SCTP endpoint takes at least 1+2+4+8+16+32=63 seconds to detect a failure of its current destination transport address and switch to an alternative one.

In order to achieve reaction times close to SS7, previous work has focused on tuning the SCTP protocol parameters. These efforts mainly rely on lowering $RTO_{min}$ to values near the path round trip time (RTT), restricting the backoff mechanism by limiting the maximum timeout value ($RTO_{max}$) and lowering the number of consecutive errors needed before a failover is performed (Path.Max.Retrans). Results e.g. in [4] showed that it is in fact possible to reduce failover times to values clearly below one second only by adjusting the protocol parameters.

However, this improvement comes at a cost. When using an aggressive $RTO_{max}$, the exponential backoff strategy, which is an important part of the congestion control, is practically disabled. Moreover, with an $RTO_{max}$ close to the RTT, transient rises of the network delay can lead to spurious retransmission and may ultimately result in severe performance degradation. Assuming that these transient delay peaks are likely to occur when there is already a disturbance present in the network, the issue of spurious retransmissions may even lead to a kind of domino effect, jeopardizing overall network stability.

Hence, the parameter tuning approach alone may not be a sufficient answer to the challenge of making SIGTRAN-based systems perform as well as the legacy SS7 networks.

The remainder of this paper is organized as follows. Chapter II describes the approach taken to address the issues listed above. In chapter III, a proposed system concept is presented. Chapter IV describes the setup and results of a set of network simulation that were conducted as a proof of concept. Finally, chapter V summarizes and concludes the paper.

## II. Approach

### A. General Issues

The approach described here starts from the assumption that carrier class networks are unlikely to rely on the end-to-end mechanisms provided by SCTP as their only means of ensuring resilience. Instead, it is common practice to use additional redundancy mechanisms working on various network layers.

While most existing work aims at an isolated discussion of SCTP, we show how the combination of SCTP with other network redundancy mechanisms can contribute to making SIGTRAN networks perform comparably to SS7.

A frequent objection to such an approach is that it violates the design principle described by the "End-to-end Argument" [5]. Quite to the contrary, we argue that our approach complies with its ideas because we do not try to *replace* the end-to-end mechanism of SCTP by something else. Furthermore, it is obvious that SCTP alone has difficulty to perform sufficiently. This justifies supplementing the end-to-end mechanism with local mechanisms and is fully in line with the ideas presented in [5].

### B. Methodology

As the result of a preselection process, a set of redundancy mechanisms was chosen for closer examination.

The mechanisms were evaluated using a predefined set of metrics. Apart from the types of failures addressed by the specific mechanism, its performance, compatibility, scalability and implementation effort were assessed. Identifying possible malicious interactions with the SCTP redundancy mechanism was another central point of the analysis.

## III. System Concept

As a result of the evaluation process, a set of network redundancy mechanisms is proposed that is able to cover almost all failure situations in a typical IP network. Furthermore, the analysis revealed no incompatibilities or other malicious interactions, neither between the individual mechanisms nor between the redundancy mechanisms and SCTP failover. The mechanisms chosen for the system concept are introduced in the next sections.

### A. MPLS Fast Reroute

When a network based on Multiprotocol Label Switching (MPLS) is deployed, network designers have a powerful tool at their disposal to quickly address failure situations. Within the IETF, a framework for MPLS-based recovery has been defined [6]. Tasks are generally divided into detecting a failure and quickly reacting to it.

Fast failure detection is a vital part of the concept. The framework allows for a variety of detection techniques to b used on different layers. Mechanisms on the physical layer allow for a very swift failure detection, e.g. an optical transmission system can be assumed to detect a "loss of light", i.e. a broken fiber, within some 10 ms. Higher layer mechanisms are mostly based on periodically exchanging messages to confirm that the adjacent node is still alive. Here, the HELLO messages used by some routing protocols or a completely new approach like the Bidirectional Forwarding Detection [7] may be employed.

As soon as the failure of a certain Label Switched Path (LSP) is detected, traffic has to be moved to an alternative LSP as quickly as possible. For that purpose, high availability solutions are likely to rely on preconfigured LSPs instead of having to first set them up after a failure. This approach is also referred to as *Fast Rerouting*. Protection LSPs can be set up end-to-end, i.e. between ingress and egress router, or locally, bypassing the next link or router. In both cases, traffic is moved to the alternative LSP as soon as a failure is detected. An optional waiting time may be defined to give lower layer mechanisms like SONET self-healing rings (see below) the chance to react first.

The standard [5] aims at achieving reaction times in the order of magnitude of 50 ms. While this figure appears quite optimistic, it is still sensible to assume that a recovery time of below one second can be achieved.

MPLS Fast Reroute is able to address all link and router failures within a Wide Area Network (WAN). However, an important exception to that rule applies to edge routers, i.e. routers that have Local Area Networks (LANs) attached. If network topology permits, the Fast Reroute approach makes it possible to reroute messages from the WAN to the LAN. Nevertheless, if the failed component is the default gateway for that LAN, this network becomes isolated because there is no way to deliver messages from the LAN to receivers outside the LAN. This issue will further be discussed in section C.

A general issue with MPLS is the effort related to setting up and maintaining the LSPs. As the solution described in this section implicates that even more LSPs have to be set up, the scalability issue is even aggravated. However, solutions to that problem are being developed, among them the Path Computation Element (PCE) [8]; MPLS Fast Rerouting is explicitly in the scope of PCE.

### B. IP Fast Rerouting

It can of course not be assumed that MPLS is deployed in every network. Many network operators partly or entirely rely on "pure" IP networks. As rerouting after failures is relatively slow with legacy routing protocols and aggressively tuning these protocols to sufficient reaction times may impose stability issues, a dedicated solution seems advisable.

Inspired by the MPLS Recovery Framework, there is ongoing work within the IETF to develop a similar architecture for classic IP networks [9]. It also takes the approach of dividing the problem into failure detection and reaction to a failure. In addition, the problem of packet loss due to transient routing inconsistencies after the repair operation is addressed.

While failure detection is assumed to work analogous to [6], recovery operations naturally differ. The scheme proposed in [9] relies on locally pre-computed alternative routes. For each destination, a router strives to compute an alternative route or find a neighboring router that has an alternative route. This is done *before* a failure occurs. It is anticipated that in around 80% of all failure cases the router itself or an adjacent router can provide an alternative routing.

For the recovery times, similar considerations as in the previous section apply. Assuming quick failure detection and the repair operation merely consisting of uploading a new routing table to the router, recovery times clearly below one second seem realistic.

The scope of IP Fast Rerouting is also the same as for MPLS Fast Rerouting, including the limitation for edge routers. However, one difference is that with the solution presented in this section it cannot be *guaranteed* that an alternative route can be found.

### C. Virtual Router Redundancy Protocol

The issue of a failed first-hop router isolating a LAN has been addressed by various proprietary, vendor-specific solutions. An approach that is based on an open standard is the Virtual Router Redundancy Protocol (VRRP) [10].

The concept is based on the observation that in most LAN environments the IP address of the default gateway is either manually configured or set by the Dynamic Host Configuration Protocol. Consequently, the device that serves this IP address is a single point of failure. VRRP takes the approach of deploying two (or even more) devices that are able to forward packets from the LAN to the WAN. One of them serves as master router and has the forwarding responsibility; the other remains passive and serves as backup router. The master broadcasts its presence to the LAN. When these advertisements fail to appear, the backup router considers the master failed and takes over the forwarding responsibility. An additional benefit of such an architecture is that there is more than one edge router that can reach a LAN from the WAN side. This increases the probability of finding an alternative route to the LAN in case of a failure.

Due to the advertisement scheme in the current VRRP specification [10], a backup router takes at least 3 seconds to react to a failure. The VRRP version 3 [11] that is currently under discussion will however be able achieve recovery times clearly below one second.

VRRP is able to cover all failures of the first-hop router or its LAN interface. This complements the scope of the fast rerouting mechanism described in section A and B above.

### D. Redundant WAN Links

An analysis of failure situations in the network of a major IP network operator [12] has shown that isolated link failures are a very common source of errors. With SONET/SDH being a standard optical transmission technology in today's WANs, a powerful protection strategy is available. With the fast error detection offered by SONET/SDH, it is possible to move traffic away from damaged optical fibers or transmission equipment. A common form of deployment is the use of so-called self-healing rings [13].

While the failed components themselves may remain isolated, the integrity of the ring is preserved. All isolated single failures in the ring can be addressed by such a setup. Normally, it can be expected that a failure is detected and repaired within 50 ms after the failure.

### E. Rapid Spanning Tree Protocol

Connectivity between the individual hosts and the first-hop routers is commonly provided by a switched network. At larger sites, there is a strong possibility that packets have to run through multiple switches to reach the router. In such a case, special measures have to be taken in order to make sure forwarding within the LAN environment is free of forwarding loops, e.g. by computing a Spanning Tree. If a failure breaks the spanning tree, conventional approaches take a significant amount of time to start a recomputation of the tree. During that time, connectivity in the LAN environment is seriously impaired.

A solution to quickly rebuild the loop-free topology after a failure is offered by the Rapid Spanning Tree Protocol (RSTP) [14]. In contrast to most other approaches that use a timer-controlled recomputation, this protocol starts a recomputation of the tree whenever a link status changes.

The computation effort and, linked to that, the performance of RSTP is strongly influenced by the number of switches. For a number of ten switches it can however be expected that the recomputation is finished within one second.

### F. User Adaptation Layer Redundancy

Although not exactly a network redundancy mechanism, the UALs can also contribute to higher availability. Some adaptation layers, e.g. the MTP3 User Adaptation Layer (M3UA) [15] are able to switch between different SCTP stacks. This does not only offer a last resort repair mechanism in case an SCTP stack fails or loses contact with its peer instance. It additionally provides a means to move signaling traffic from one SCTP stack to another. As effectively employing this method necessitates separate application servers, this also means that traffic can be moved between servers so that e.g. maintenance work can be performed without disrupting connectivity for the signaling protocol working on top of the UAL.

### G. Other Aspects

The only type of failure that remains uncovered by the above redundancy mechanisms is an interface failure at one of the SCTP endpoints. A network redundancy mechanism is not able to repair this type of failure. Thus, when designing the corresponding hardware, this fact has to taken into account. Possible solutions would e.g. be redundant hardware or a detection mechanism that makes it possible for SCTP to immediately perform a failover to another interface.

When considering the different redundancy mechanisms as parts of a system solution, the question arises which repair mechanisms should be preferred. We argue that the network redundancy mechanisms are able to repair failures quickly and almost transparently to SCTP and should therefore be preferred. Reaction times should be stacked in a way that SCTP failover does not react unless it is certain that the network mechanisms were unable to address the failure. The same principle is already applied in the MPLS recovery framework. Here, a configurable hold-off time can be specified that makes sure that MPLS rerouting is only invoked if an underlying mechanism like SONET/SDH failed to counter the failure.

## IV. Network Simulations

To verify the proposed concept, network simulations were conducted. Two scenarios were evaluated:

- A basic, 'plain' IP network with SCTP parameters very close to minimum settings needed for stable operation.
- A network using the proposed enhancements with the standard protocol parameters from RFC 2960.

For both cases, an SCTP association was examined with regard to robustness to concurrent failures and the maximum end-to-end delay perceived during a failure. The SCTP parameters used are shown in table 1.

TABLE 1
SCTP PARAMETERS USED FOR THE SIMULATIONS

| Parameter | RFC 2960 | Aggressive setting |
|---|---|---|
| $RTO_{max}$ | 60 s | 250 ms |
| $RTO_{min}$ | 1 s | 10 ms |
| $RTO_{init}$ | 3 s | 250 ms |
| Path.Max.Retrans | 5 | 2 |
| Assoc.Max.Retrans | 10 | 4 |
| SACK delay | 200 ms | 0 ms |
| Heartbeat interval | 30 s | 30 s |

### A. Simulation Setup

The network simulator ns-2 [16] version 2.27, running on a standard Linux PC, was used for the simulations. An SCTP implementation [17] is already included in the simulator. The ns-2 code has been modified in order to take measurements of the end-to-end delay.

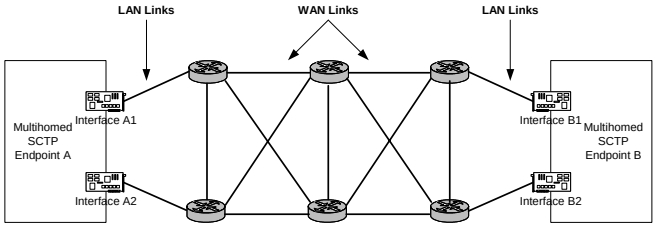


Fig. 1. Simulation Setup.

For the simulations, a network topology according to figure 1 was used. Two dual-homed SCTP-endpoints were connected via a simple network consisting of six routers. Each interface was connected to a dedicated edge router; two additional routers provided a means of simulating the failure of a "core" router that has no endpoint directly attached. The routers were fully meshed with 2 MBit/s WAN links with 40 ms delay. To simulate an access LAN, the links between interface and router were 10 MBit/s links with a delay of 10 ms, resulting in an end-to-end delay of 100 ms. The primary SCTP path was set up between interfaces A1 and B1, in case of a failover the path between A2 and B2 was used. A traffic generator was attached to endpoint A, transferring an average rate of 100 messages per second (fixed message length of 257 bytes) to endpoint B. The low data rate resulting from these settings was chosen to eliminate the influence of congestion effects.

Link or router failures were simulated by setting the link status (or the status of all links attached to one router respectively) to "down" after a specified time.

For the enhanced network, only a subset of the proposed redundancy mechanisms could be simulated in ns-2. The effect of IP Fast Rerouting was emulated by manually triggering a rerouting one second after the failure. The impact of a VRRP repair operation was modeled by activating an additional link between interface and the alternative first-hop router 800 ms after a failure. In case of a WAN link failure, the link was taken back "up" 50 ms after the failure to simulate SONET/SDH performing a repair operation.

### B. Robustness

To analyze the reaction to multiple concurrent network failures, a failure generator was implemented that randomly generates a predefined number of failed network elements (links or routers). The failure generator makes sure that at least one failed element is part of the primary path.

The selected elements were set to "down" after a predefined time. After that, it was determined whether the SCTP association had been closed due to the failure or not. The simulation was run for one to ten concurrently failed network elements; for each case, 1000 simulation runs were performed. The results are shown in figure 2.

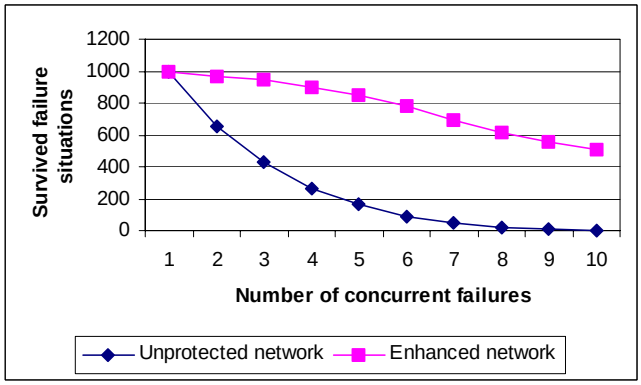


Fig. 2. Robustness to concurrent failures, comparing unprotected with enhanced network.

The curve representing the unprotected network clearly demonstrates the vulnerability of an SCTP association to concurrent network failures, even in a fairly ideal setup as chosen here. Whenever both SCTP paths are affected by a failure, the association will inevitably fail. For two concurrent failures in the network, only 67% of all failure situations can be survived. For three concurrent failures, this survival rate even drops below 50%. Compared with that, the enhanced network performs significantly better. Due to features like the swift rerouting around failed links or routers, the survivability remains above 94% for double or triple failures. Cases where more than three out of the 21 network elements simultaneously failed may surely be considered unlikely. However, the results clearly illustrate how vulnerable the end-to-end failover mechanism of SCTP is with regard to simultaneous network failures.

When an enhanced network is used, the question arises whether SCTP failover or the network redundancy mechanisms were responsible for addressing the failure situation. For double or triple network failure, SCTP failover only had to be invoked in less than 6% of the cases. Even for higher amounts of concurrently failed network elements, this figure remains below 20%.

The results show that the use of network redundancy mechanisms can clearly improve the robustness to concurrent failures in the network. SCTP's end-to-end failover mechanism only comes into play when the network mechanisms are unable to repair the failure.

### C. Delay

Apart from the robustness of an SCTP association, another critical aspect is the delay a signaling application experiences during a failure situation. To measure this value, the simulator was modified to calculate the time that a user message takes from the time it was handed over to SCTP at the sender side until it leaves the stream reordering queue at the receiver side.

Again, the performance of a basic network with aggressive SCTP parameters is compared with an enhanced network with default protocol parameters. Four single failures are examined, covering different combinations of recovery mechanisms. Figure 3 shows the maximum message delay during a failure situation; the figure presents the average over 100 simulation runs and is shown together with the 95% confidence intervals.

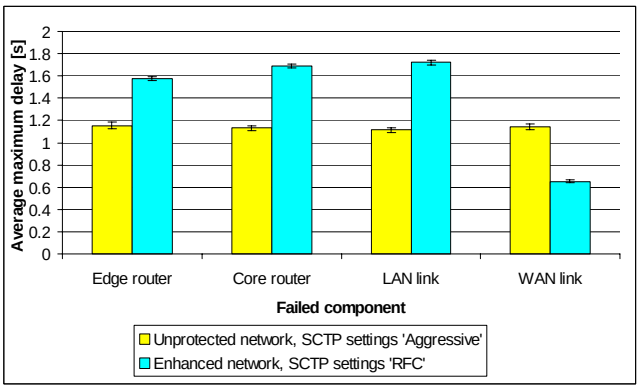


Fig. 3. Average maximum delay during failure situation.

When the end-to-end failover mechanism using the aggressive parameter settings is employed, the highest delay is around 1.1 seconds.

The delay measurements for the enhanced network with standard protocol parameters lead to higher values. For the cases of a failed LAN link, edge router or core router, the results are almost identical. They yield maximum end-to-end delays between 1.6 and 1.7 seconds. Due to the very quick reaction of the SONET/SDH mechanism, the rise in end-to-end delay is even as low as 0.6 seconds for a WAN link failure.

At first glance, the performance of a network employing the proposed enhancements does not seem superior. However, it has to be taken into account that the conservative default protocol parameters have been used that lead to long timeout intervals and therefore to a higher delay. Hence, there appears to be plenty of room for improvements in the parameter settings. In fact, both pure IP and enhanced network show a maximum delay that is already in the same order of magnitude. It therefore seems likely that rethinking the parameter tuning approach in the context of the proposed enhanced network architecture can further improve the delay behavior. These efforts will have to focus on better balancing the requirement for quick reaction to disturbances and the need for leaving the proven congestion control intact in order to maintain stability.

The major advantage of the proposed architecture is that it forms an alternative to the problematic aggressive parameter tuning efforts, rendering a similar delay performance while noticeably improving robustness and stability.

## V. Conclusion

We presented a novel approach to improve the behavior of SIGTRAN-based signaling networks in scenarios with multiple concurrent network failures. Existing research focuses on an isolated discussion of SCTP. In contrast to that, we aimed at developing a system concept and investigated how supplementing SCTP with a set of redundancy mechanisms can contribute to making SCTP perform similarly to SS7 networks.

After identifying and analyzing suitable candidate mechanisms, we combined the different mechanisms to a system solution. As a proof of concept, we conducted network simulations in a simple example scenario. Here, we compared robustness and delay behavior of a conventional approach with our proposed architecture.

The simulation results clearly pointed out the vulnerability of a network entirely relying on the SCTP failover mechanism. One third of all concurrent double failures in the example network led to a disruption of the signaling association. For three concurrent failures, only half of the associations survived. In contrast to that, the enhanced network withstood over 94% of both double and triple failure situations. At the same time, only minor impact on the delay behavior could be observed. In addition, the enhanced network can be expected to be more stable as it operates with standard SCTP protocol parameters. The responsibility for reacting to failure situations shifts from the SCTP failover to the network redundancy mechanisms, turning SCTP failover into a “last resort” mechanism. Moreover, our studies revealed no malicious interactions, neither between the redundancy mechanisms and SCTP failover, nor between the different redundancy mechanisms themselves.

Our work has shown that the proposed system solution is capable of noticeably improving robustness to concurrent failures while achieving a comparable delay behavior and without modifying the default SCTP protocol parameters. The promising results of the simulations conducted in our simple reference network topology clearly motivate further studies on this issue. Possible study items could be a broader and more realistic range of topologies, an extended set of redundancy mechanisms or the influence of cross-traffic in the network.

## VI. Acknowledgments

The work leading to this paper was partially sponsored by the German Ministry for Research and Education (BMBF) within the 3GET project (research grant 01BU355). The paper solely contains the view and opinions of the authors.

## References

[1] ITU-T Recommendation Q.700, “Introduction to CCITT Signaling System No. 7”.
[2] L. Ong, M. Garcia, et al., “Framework Architecture for Signaling Transport”, RFC 2719, October 1999.
[3] R. Stewart, Q. Xie et.al., “Stream Control Transmission Protocol”, RFC 2960, October 2000.
[4] A. Jungmaier, M. Tüxen, “On the Use of SCTP in Failover Scenarios”, *SCI 2002*, Orlando, Florida, USA, July 2002.
[5] J. Saltzer, D. Reed, D. Clark, “End-to-end Arguments in System Design”, *Second Conference on Distributed Computing Systems*, April 1981.
[6] V. Sharma, F. Hellstrand, “Framework for MPLS-based Recovery”, RFC 3469, February 2003.
[7] D. Katz, D. Ward, “Bidirectional Forwarding Detection”, <draft-ietf-bfd-base-00.txt>, Internet Draft, work in progress, July 2004.
[8] A. Farrel, J. Vasseur, “Path Computation Element (PCE) Architecture”, <draft-ash-pce-architecture-00.txt>, Internet Draft, work in progress, September 2004.
[9] M. Shand, “IP Fast Reroute Framework”, <draft-ietf-rtgwg-ipfrr-framework-02.txt>, Internet Draft, work in progress, October 2004.
[10] R. Hinden, “Virtual Router Redundancy Protocol”, RFC 3768, April 2004.
[11] R. Hinden, “Virtual Router Redundancy Protocol for IPv6”, <draft-ietf-vrrp-ipv6-spec-07.txt>, Internet Draft, work in progress, September 2004.
[12] A. Markopolou, G. Iannaccone, et al., ”Characterization of Failures in an IP Backbone”, *IEEE INFOCOM 2004*, Hong Kong, March 2004.
[13] T. Wu, “Emerging Technologies for Fiber Network Survivability”, *IEEE Communications Magazine*, February 1995, pp. 60-74.
[14] IEEE 802.1w, “Rapid Reconfiguration of Spanning Tree”.
[15] G. Sidebottom, K. Morneault, J. Pastor, “Signaling System 7 (SS7) Message Transfer Part 3 (MTP3) - User Adaptation Layer (M3UA)”, RFC 3332, September 2002
[16] The Network Simulator ns-2, version 2.27, http://www.isi.edu/nsnam/ns/.
[17] A. Caro, SCTP module version 3.4 for ns-2, http://www.cis.udel.edu/~acaro/research/ns2sctp/.