

Multi-Layered Network Survivability – Models, Analysis, Architecture, Framework and Implementation: An Overview

Deep Medhi
Computer Science Telecommunications
University of Missouri-Kansas City
Kansas City, MO 64110-2499
dmedhi@umkc.edu

David Tipper
Department of Information Science
and Telecommunications
University of Pittsburgh
Pittsburgh, PA 15260
dtipper@mail.sis.pitt.edu

Abstract

A major attack can significantly reduce the capability to deliver services in large-scale networked information systems. In this project, we have addressed the survivability of large scale heterogeneous information systems which consist of various services provided over multiple interconnected networks with different technologies. The communications network portions of such systems are referred to as multi-networks. We specifically address the issue of survivability due to physical attacks that destroy links and nodes in multi-networks. The end goal is to support critical services in the face of a major attack by making optimum use of network resources while minimizing network congestion. This is an area which is little studied, especially for large-scale heterogeneous systems. In this paper, we present an overview of our contributions in this area.

1. Introduction

A major attack or a failure can significantly reduce the capability to deliver services in large-scale networked information systems. The drastic effects of communications network failures (even for non-maligned cases) have been demonstrated by several widely publicized network outages in recent years. We broadly classify any type of attack or failure broadly into two groups: physical attack/failure and software attacks. Physical attacks include attacks to destroy switches or transmission systems in some part of a network. Software attacks include attacks such as intruders breaking into systems and destroying, compromising and/or manipulating systems through software change (e.g., injecting false information into network routing tables). In this Darpa project, our work addresses the first group of attacks, i.e., the physical attacks. The obvious question is how to

build and evolve network architecture to cope with any major physical attack so that critical information systems can still communicate by utilizing the available network and system resources. It may be noted that a major attack can be caused by a 'single' failure (such as an attack destroying a single high traffic link or node) or 'multiple' failures (such as an attack destroying multiple links and/or nodes). In the rest of the discussion, the terms 'failure' and 'attack' will be used interchangeably.

Most network-based information systems environments consist of a combination of legacy and emerging technologies. Broadly speaking, the environment can be classified into three layers. The top ('application/service') layer is where services such as voice, data, video, multi-cast video, and other distributed services are provided. These services are provided over the middle layer (referred to here as the 'switched network layer') which consists of the networking environment such as circuit-switching, packet-switching (TCP/IP), ATM, and virtual private networks (for special services). Finally, the middle layer is provided over the network transmission environment layer (the bottom 'physical' layer) which typically consists of a mixed technology infrastructure containing fiber and non-fiber wirebased systems as well as wireless components (microwave, cellular, satellite, etc.). Note that an application/service may traverse several interconnected networks with different physical layer and network layer components. For brevity, such heterogeneous multi-layered systems, which takes a resource-directed network architecture view, will be labeled as "Multi-Networks".

In this work, we address survivability to provide network design and management procedures towards minimizing the impact of failures on multi-networks. Survivability techniques can be classified into three categories: 1) prevention, 2) network design, and 3) traffic management and restoration. Prevention techniques focus primarily on improving component and system reliability. Some examples

are the use of fault-tolerant hardware architecture in switch design, provision for backup power supplies and transmission equipment, use of frequency hopped spread spectrum techniques to prevent jamming in military radio networks and so on. Network design techniques try to mitigate the effects of system level failures such as link or node failures by placing sufficient diversity and capacity in the network topology. For example, the use of multi-homing nodes so that a single link failure cannot isolate a network node or an access network. Traffic management and restoration procedures seek to direct the network load such that a failure has minimum impact when it occurs and that connections affected by a failure are reconnected around the failure. Survivability goals may be accomplished by designing network infrastructures that are robust to malfunctions of nodes and links, and implementing network control systems that are inherently fault-tolerant and self-healing.

Given the multi-networks environment, survivability and restoration can be addressed at multiple levels. Although it may be possible to address survivability in each layer completely independently, our view is that a multi-layer coordinated and integrated survivability and restoration approach is most desirable to make maximum use of available resources. This should also take into account the underlying technology infrastructure so that evolving survivable network architectures can be generated. With this in mind, our goals in this project have been to address the following inter-related issues:

- Development of network design models/algorithms to provide a quality of service (QoS) specified under any failure condition. This work addresses the problem of intelligently designing and evolving a network topology architecture and configuration starting from an existing architecture and legacy networks to improve survivability.
- Development of network management algorithms (e.g., provisioning of backup routes, virtual circuit rerouting algorithms, etc.) which make optimum use of network resources after a failure (both single and multiple types) in support of critical services. This work concentrates on the design and analysis of multiple priority traffic restoration techniques to provide service continuity while minimizing the network congestion. The restoration algorithms will be suitable for automatic invocation by network components, resulting in a self-configuring system that adapts to the changing fault environment. Since emerging multi-casting services such as audio/video conferencing will be critical under an attack, we plan to especially address the issue of the survivability of multi-casting services.

- Emphasis on studying the transient network congestion that occurs after a failure and incorporating its effect into the design of the network and the traffic restoration algorithms. Thus not only will a critical network user be provided service continuity, but the quality of the service can be provided in a graceful manner.
- Functional needs for the network management for multi-networks survivability and demonstration of some of the basic concepts and procedures through a proof-of-concept testbed implementation.

With the above goals in mind, we have divided the work into the following main components:

- Development of design models and tools for survivable networks.
- Development of simulation models and tools that can be used to study of failure.
- Analysis of various network scenarios involving a failure using the tools developed, as well as an evaluation of traffic restoration algorithms.
- A network management framework development for multi-layered network survivability and its proof-of-concept implementation in the MIMIC (Mini Intelligent Multi-networks Information survivability Concepts) testbed at the University of Missouri–Kansas City.

In the rest of the document, we will visit each of these components. It is not possible to cover all of our publications and results in this overview paper. Thus, in most cases, we summarize some of our contributions, while we present a longer description for the network management framework. As applicable, we have indicated the technical reports, theses and publications where more details can be found.

2. Network Design Models

In general survivable network design refers to the incorporation of survivability strategies into the network design phase in order to mitigate the impact of a set of specific failure scenarios. Survivability is typically achieved through, either placing diversity and spare capacity in the network topology (or virtual topology) or adding redundancy to network components (e.g, 1+1 automatic protection switching). In general taking an approach of diversity and spare capacity placement by adopting a mesh type topology with extra capacity is known to be more cost effective and flexible (i.e., can respond to a wider range of failures), than following a redundancy approach. We are interested in network design

models that include issues related to network survivability within the framework of the models. In this section, we describe some of the models and tools we have developed.

2.1. Survivable Virtual ATM Network Design

The details of this work are reported in the doctoral dissertation of R. Cotter[6]. In this section, we have highlighted the main contribution of this work.

In a multi-network environment, services required by the upper layer can be provided over a logical network. An example of such a logical network is an ATM-based (Asynchronous Transfer Mode-based) virtual network where virtual paths (VP) are defined for different services based on the demand requirement of the application services level. A critical requirement to address is the survivability of such networks, especially taking into consideration the dynamically rearrangeable capability of ATM VPs. In such an environment, to address for survivability, three different strategies are considered: redesign of the entire network for every possible failure scenario, design of the affected parts (VPs) for any failure scenario, and built-in diversity in the initial design of the network.

While addressing these strategies, along with issues such as consideration of multicast group traffic demand that need to be survivable too, we have been able to develop a generic multi-commodity flow based optimization model that can capture them all very well. This is a major contribution in itself. Further, by looking at the structure of the problem, we have developed an algorithmic framework that can also be used for all of the strategies and scenarios. At the core of this algorithmic framework is a decomposition algorithm which is based on the duality-based subgradient optimization algorithm described in [14].

The design model has been implemented and tested on 96 different test networks ranging from 10-nodes in a network to 100-nodes in a network with multiple service classes and traffic load periods. In addition, a protocol message mechanism that works at the ATM VP level has also been developed. It was geared for ATM VP network survivability.

2.2. Book-ahead Guaranteed Survivable Services

In this work, we have considered a best-effort Internet environment with added capability to provide some guaranteed services that are requested ahead of time. An integral requirement for such guaranteed services may be some level of survivability. A possible approach here is to consider a pair of diverse paths for every survivable demand request. At the same time, to assign bandwidth on the back-up for survivability would mean that other best-effort services can not access this bandwidth if there is *no* failure. Thus, this

can lead to consideration of hard requirements and soft requirements.

In addition, such a problem can have conflicting goals; for example, (a) maximization of residual capacity for usage by best effort services, (b) minimization of the cost of the book-ahead survivable provisioning, and (c) possible penalty for not being able to provision for some requests. We have been able to develop an optimization formulation that captures such conflicting goals in a comprehensive manner.

We have done studies on several sample networks and demand requirements to show the effectiveness of the model in meeting different objectives. The details of the model and the results are reported in [23].

2.3. Survivable STM Network Design

STM refers to synchronous transfer mode. Networks based on SONET, and digital cross-connect systems fall under the STM category. In [1, 2], we consider the problem of given a STM mesh type network topology, the normal traffic demand, and the capacity allocation to meet the normal traffic demand, how much spare capacity should be provisioned and where should it be located in order for the network to tolerate a specified set of failure scenarios (e.g., loss of any single link). The term "mesh" does not imply that the network topology is a full mesh, but rather that the network nodes are at least two connected.

Specifically, we present a novel STM survivable network planning technique based on a genetic algorithm formulation of the spare capacity assignment problem for the case of path restoration with link disjoint routes. Genetic algorithms [7] have received considerable attention in recent years for use in solving various combinatorial optimization problems, including the solution of integer programming problems. Genetic algorithms (GA) are stochastic search techniques that mimic the survival of the fittest (or best) paradigm observed in nature.

Our design methodology consists of using the genetic algorithm approach to implement the concept that traffic flows which travel over disjoint routes may be able to share spare capacity on a backup path, since it is unlikely that more than one failure will occur simultaneously. Thus, our approach tries to reduce the cost of spare capacity needed for a particular fault tolerance requirement (e.g., full recovery from any single link failure) by finding a set of backup paths that enables the sharing of spare capacity, which results in reducing the total cost due to the nonlinear economy of scale of spare capacity cost.

The description of the proposed methodology is given in detail in [1, 2] along with a study of numerical results for a variety of network topologies, illustrating the application of the proposed genetic algorithm technique, guidelines for parameter selection and analysis of the computational com-

plexity. Additionally, for the sake of comparison, numerical results for small networks are given for the standard integer programming approaches and a popular heuristic from the literature. It is shown that the GA algorithm is far more computationally efficient while providing near optimal results (2-7%).

3. Simulation Tools

Our interest in studying the effect of a failure on a network and the performance of any newly developed routing and restoration algorithms has led us to the development/enhancement of different simulation tools. Specifically, we have been interested in understanding the implications at the packet-level granularity in the best-effort Internet environment as well as connection/session level granularity in the case of emerging Integrated Services Architecture. Due to our varied interest, it became apparent that just one tool cannot fit every environment. Thus, this led us to the enhancement of the MaRS tool to create the new tool, MoMaRS, which is ideal for understanding packet-level implications due to a failure for multicasting environment. We have also developed extensions to ns simulator [33] for the unicast service environment that addresses fault tolerance. However, to address and understand the connection-level effect, we needed a tool that can simulate several thousand connections in a short period of time.

3.1. MoMaRS Tool

MoMaRS is a packet-level simulation tool. The new tool has been created by extending the MaRS tool[3]. MaRS has very good built-in unicast routing components as well as various service level workload components. Its original design allows the study of a link failure. Our interest was to consider multi-cast services and their performance under failure, and we are also interested in considering an environment that addresses Differentiated-Services Internet[4] and Integrated Services Internet[5].

With this in mind, we have made major enhancements to create the MoMaRS tool. In particular, the tool now has two multicast routing components: MKompella[12] and Multicast Shortest Path First (MSPF) routing protocols, as well as multicast workload components for one-to-many and many-to-many communications. We have further added priority based scheduling at the nodes (routers) for emulating a differentiated-services environment, and Resource reSerVation Protocol (RSVP) along with a classifier, a packet scheduler, and an admission controller to emulate the Integrated-Services environment. In addition, a new TCP component is added to allow a user to study interaction between TCP behavior and routing dynamics due to a failure. In particular, this environment allows us to study a single or multiple

link failure. We are currently working on a user's manual for this tool and it is expected to be available soon.

3.2. MuSDyR Tool

While MoMaRS is a good tool for understanding packet level impact, it is not geared towards studying session or connection level issues for large-number of sessions/connections. Due to lack of an available tool that suits our needs, we have developed a new simulator called MuSDyR[17] (Multi-service Simulator with Dynamic Routing) for this purpose. MuSDyR is designed to consider connection-level impact along with flow/connection-level quality of service routing components. Due to our special interest in studying a network failure, this tool has been designed from the beginning with the capability to study a failure. In particular, it has a partial restoration capability, allowing the user to do staggered restoration after a failure. In addition, this tool has the capability to re-route a connection in the event of a failure. The tool can easily handle simulation of ten thousand simultaneous connections. To consider the fact that network traffic is non-stationary and changes with time, a dynamic traffic generation module has been included in this tool along with the stationary traffic generation module. In addition, this tool has the built-in capability for reservation-based multicast services.

3.3. Extensions to NS simulator

In order to study the effects of failures and evaluate survivability schemes on unicast services for various next generation Internet architectures, we have developed a simulation tool by extending the Network Simulator-NS (version 2) [33]. NS is widely used in the Internet research community. In order to simulate survivability schemes for unicast services, new modules were added to NS including an admission-control agent, a RSVP agent, a flow-routing agent, a resource agent and a fault-tolerant agent. The admission-control agent determines if a connection requesting QoS will be accepted based on the available resources. The RSVP agent will send reservation messages to setup or tear down the flow along the path given by a flow-routing agent. The flow-routing agent at each node maintains path information (set of candidate paths to other nodes) and routing information once the flow is set up. The candidate path set for each node pair is precomputed and loaded into the simulation before execution. Using the candidate path set the flow-routing agent runs the path selection algorithm to find the path that gives the minimum cost route. A resource agent at each node keeps track of resource levels at all ports. The simulation model is constructed so that all nodes share global information of resource levels. A fault-tolerant agent at each node incorporates the different restoration recovery

schemes. Additional modifications to NS include changes to statistics gathering routines to permit the gathering of transient behavior across multiple simulation runs.

4. Some Results

Using simulations, various analyses can be done to understand the impact of a failure, as well as performance, when a new control scheme is introduced to alleviate any problems. In this section, we briefly discuss some preliminary results. We are currently conducting several studies which will be reported elsewhere.

4.1. Transient Behavior with Multicast Services

Using the MoMaRS tool, we have studied the transient network behavior of multicast and unicast connections in both differentiated-service and integrated-service capable Internet architectures under major link failure. Detailed results can be found in [18, 24, 25]. We briefly highlight some of the results here.

Our results show that the performance of multicast routing protocol is essential in reducing the overall network utilization, thereby reducing the overall delay for all packets through efficient network utilization. The priority-based routing algorithm does provide QoS assurance for higher priority traffic. Figure 1 shows a dramatic decrease in the instantaneous delay and jitter for a selected multicast traffic stream when priority-based routing is employed. RSVP along with enhanced queuing and scheduling mechanisms can provide required QoS for real-time traffic in the network with link failure as well as without link failure. Figure 2 shows that in the RSVP-enabled environment, the delay and jitter of the multicast traffic during the failure period remains more or less the same as that during the pre-failure and link recovery periods. This is a very significant improvement in QoS for multicast traffic when compared with base simulations where RSVP is not employed. Simulation results using different routing protocols show that the ability of RSVP to re-establish the affected reserved paths due to network failure depends on the underlying routing protocol.

4.2. TCP Behavior due to Network Dynamics

Using the TCP component of the MoMaRS tool, we have also studied the impact of network dynamics on TCP [21, 22]. Network dynamics refers to the changes in the network state due to route oscillation, link failure and so on. While TCP has been extensively studied over the past decade, surprisingly, very little work has addressed the impact of network dynamics.

Our results show that the impact of network dynamics on TCP depends mainly on its acknowledging (or ACKing)

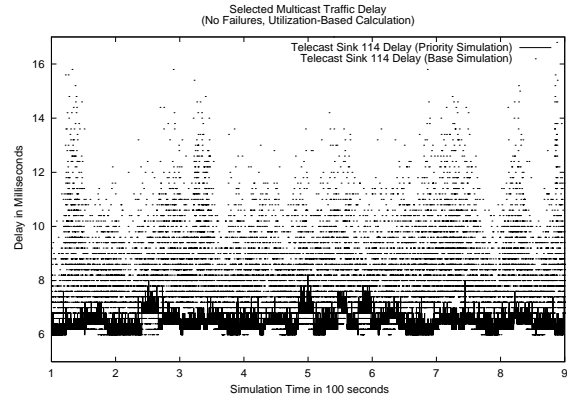


Figure 1. Telecast Sink 114 Instantaneous Delay (No Failure)

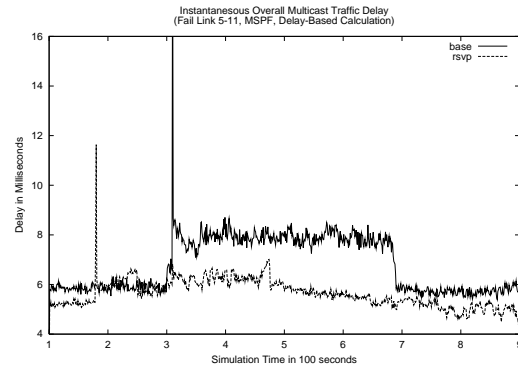


Figure 2. Multicast (Failure, MSPF, Delay-Based)

behavior, where the performance of TCP with Non-Delayed acknowledgment (TCP-NDACK) is noticeably affected as compared with that of TCP with Delayed acknowledgment (TCP-DACK). During route oscillations, TCP flow displays asymmetry and false retransmission. Whereas during overlapping link failures, it displays multiple packet loss, retransmission timeouts, false retransmissions, and flow synchronization.

While false retransmissions can be eliminated in most of the cases by sharing congestion information between the header prediction and fast retransmit algorithm, an improved congestion detection mechanism at the nodes/routers and selective acknowledgement in TCP are necessary in order to handle link failures efficiently. We believe that a better acknowledgement technique would make TCP less susceptible and more adaptive to changes in network dynamics.

4.3. Survivable Services in NGI

In this work, we have investigated schemes for the survivability of guaranteed QoS connections that can be applicable in the Next Generation Internet (NGI) network architecture[30] using the extensions done to the ns tool. We assume the use of RSVP signaling to reserve resources along a fixed route (explicit path reservation) to provide QoS. A comparative study of the performance of two basic survivability schemes within an NGI architecture is presented. In the first scheme, a QoS connection is provided standby backup resources on a disjoint path by reserving resources on both the working and the backup path. In order to reduce the amount of backup resources required a method for sharing backup resources when the working connections have disjoint routes has been included. In the second scheme a dynamic search for restoration resources is conducted over a preplanned set of alternate paths upon notification of a failure.

A simulation based performance study was conducted to quantify the tradeoff in connection blocking from the guaranteed recovery scheme under normal operations versus the connection blocking after a failure from the dynamic search approach. In addition, the speed of recovery in reconnecting sessions is studied along with the transient network congestion produced by retransmission of lost data. The study shows that the first scheme results in much higher connection blocking under normal operations, faster restoration times, and longer transient congestion times due to non-optimal backup routing.

5. Traffic Restoration Procedures

As noted earlier, survivable traffic management and restoration procedures seek to direct the network load such that a failure has the minimable impact when it occurs while the load affected by a failure is restored. The performance of a traffic restoration procedure will largely depend on the combination of the algorithm used for restoration and the spare capacity allocation in the network. In general, simple restoration schemes will lead to allocating more spare capacity, whereas more sophisticated restoration schemes will require less capacity, but may take longer time to restore connections. In a multi-network environment we contend that distributed dynamic restoration schemes need to play a central role in traffic restoration since such networks will typically be operated by various service providers possibly using different technologies at various network layers and the end-to-end spare capacity planning could be difficult. Our work on traffic restoration has concentrated on the development of algorithms for distributed dynamic traffic restoration to make optimum use of whatever resources are available after a failure, in support of critical services. Here

we summarize some of our main results from [9, 10, 34, 35].

5.1. Virtual Circuit Fault Recovery Routing

In [34] and earlier work, we have studied the problem of routing for traffic restoration after a failure in virtual circuit based wide area networks utilizing source node routing. Note that in such networks, a device failure will typically result in several nodes having many virtual circuits to restore and a critical issue in the restoration is the path chosen for rerouting. Another major factor on network performance after a failure in packet switched networks in general is the transient congestion period that results from restored virtual circuits attempting to send out the backlog of packets accumulated during the time delay between the failure occurring and restoration[32]. Standard routing algorithms in packet networks are normally based on minimizing the steady state network delay and such algorithms may be inappropriate for rerouting affected connections after a failure, since at this time, congestion is a paramount issue.

We present an optimization formulation of the rerouting problem by considering residual capacity in the network as well as the decision on whether or not to reconnect a disrupted virtual circuit[34]. Our formulation allows us to consider several routing schemes and fault scenarios in a unified framework. Note that after a failure many virtual circuits will simultaneously need to be restored; thus, we formulate the restoration problem as a bandwidth packing problem. formulation is based on precise information on the network link status and the decision is done in a centralized manner. In an actual implementation in a network, the source-node based routing makes decision in a distributed manner based on delayed information about network link status. We discuss how our proposed routing algorithms can be implemented in a distributed fashion.

The results of a simulation based performance study are reported comparing the performance of both optimal centralized and distributed implementations of five different routing algorithms in terms of network congestion and traditional survivability metrics, such as the call blocking. Through extensive simulation under different network load conditions, our results indicated that while traditional survivability metrics show little differences between the routing algorithms studied, the transient network congestion behavior is noticeably different. Further, it is shown that at low network loads, when there is enough spare resources so that restoration call blocking is low, the network behavior is the best when the load is evenly distributed throughout the network. In contrast, at heavy loads, when there is high amounts of restoration call blocking, one needs to match the routing scheme to the characteristics of the application to ensure the best network performance.

5.2. ATM Fault Recovery Priorities

In [9, 10], we propose a priority scheme for reconnection of virtual circuits (VCs) in ATM networks that have been disrupted by a failure. ATM networks offer several service categories each designed to handle applications with specific traffic characteristics. A failure typically results in *several* nodes being sources for failed virtual circuits with each having *many* virtual circuits in each service category to *simultaneously restore*, possibly on the order of tens of thousands. The way in which the virtual circuits are processed and routed will determine in part, if the connection is restored, the delay in reconnection and the QoS provided after restoration.

In [10], we have developed a restoration priority scheme based in part on the ATM service classes, which aims at minimizing the impact of a failure on the network while providing users the best possible service. The scheme involves both a priority for reconnection among ATM service classes and a rule for ordering and routing VCs within a service class. The proposed scheme is formulated within the context of switched VC routing but is applicable to virtual path restoration as well. The priority traffic restoration technique proposed is based in part on minimizing the number of dropped cells that need retransmission, thus reducing the transient congestion that occurs after restoration.

Numerical results evaluating the performance of the priority scheme show that it significantly reduces the amount of cells needing retransmission after a failure, thereby reducing network congestion at a cost of longer restoration times for lower priority virtual circuits. More recent work [9] formulates the priority restoration problem for ATM within a optimization framework and compares distributed implementations to the benchmark centralized optimization problem solution, suggesting some improvements to our original distributed scheme. This work clearly shows the benefits to both the high priority network user and the network operator in adopting a multi-priority restoration scheme, in terms of speeding up restoration and reducing network congestion.

5.3. Fault Tolerant Connection Oriented Multicasting

In [35] we examine the potential benefits of dedicated backup route(s) to provide survivability for ATM group communications. Specifically we examine the feasibility of providing survivability using working multipoint routes with disjoint dedicated backup multipoint routes where the multipoint routes are setup using either Virtual Rings, shared multicast trees, or VC Mesh groups. We introduce a optimization formulation which identifies a “Disjoint Steiner Ring” within a general graph in order to construct Virtual Rings. Numerical results show that disjoint backup shared

multicast trees and disjoint backup VC Mesh groups are not always available while self-healing Virtual Rings exist in all cases tested. In addition, experiments comparing the cost of providing survivability between self-healing Virtual Rings, shared trees, and VC Mesh groups show that self-healing Virtual Rings are lower in cost for the topologies considered.

6. Network Management Framework and Tested implementation

In the Introduction, we have given the basic notion behind the multi-network framework that is governed by a resource-directed layered network architecture. Towards developing the network management framework for this architectural context, we now elaborate on motivations behind the framework.

In the resource-directed layered network architecture, it may be noted that each layer will have independent policies regarding routing and resource management. However, the granularity and how often the routing changes are different. For example, in today’s best-effort Internet, the routing can change at the packet level, while the routing in an ATM Network can be either at the connection or the virtual path level. In digital cross-connect networks, routing is either at the T1 or T3 level and does not tend to change in the scale of minutes or sometimes even days. When we consider the SONET environment, the routing can be at OC-3, OC-12 levels and so on. Further, the time scale of response of routing can be different for different environments. While SONET self-healing rings can reroute quickly (about 50 milliseconds), this is not the case in the SONET mesh network environment. Now, consider such a heterogeneous environment and assume that there is rerouting capabilities in each layer. Some important issues arise such as how a failure at the lower resource layer will affect the overall network, what is the best way for overall networks to respond to a failure, and what functionalities are needed for the failure management.

The administration and management of such a resource-directed multi-layered communication network involve some additional complexities in the routing and resource management strategies during link/node failure situations. Although the traditional management systems, which are designed to manage the network of single administrative domain or of homogeneous technology, do deal with issues related to resource management and survivability, the scope of the management information available to the domain-specific management systems is very localized. For example, the management system of a virtual private network may request an additional capacity on an overflowing point-to-point link. Similarly, a link failure at the physical layer may affect some of the virtual channels of service provider networks at the layers above the physical network. Currently

most of these inter-domain issues are handled by human managers. However, with ever-growing multi-layered communications networks, it would be difficult for the human managers to handle the increasing number of such tasks. Another important factor is the desired reaction time for the inter-layer network management issues. For example, when a physical link fails, some of the affected logical channels require immediate attention so that at the user level there will not be any perceivable breakdown of service.

One of the limitations of the domain-specific and technology specific management systems is that the abstraction of the management information is different from one domain to another. Also, each of these management systems will have no knowledge about the abstraction of management information of other domains. The interaction and exchange of management information among the domain-specific management systems are essential in the management systems of multi-layered networks. Hence, the definition of an interface of communication and a common abstraction of management information forms an important factor in the implementation of such management systems for multi-layered networks. The manager at an upper level of hierarchy uses different interfaces of communication to interact with different domain-specific managers at the lower level.

Here, we give an overview of a loosely-coupled hierarchical framework that can facilitate maximal survivability of services in multi-layered networks for various failure situations. More details about this work can be found in [11, 15, 26, 28]. In this discussion, we specifically consider a resource directed two-layered architecture where the networks at the upper layer are called ‘user networks’, while in the lower layer, we have a ‘provider network’. We assume that user networks and the provider network are *not* all completely controlled by the same network administration. In each of the user network or the provider network, we assume that there is at least a domain-specific manager. In this framework, the domain-specific managers are called by the name of the domain for which they are responsible.

All the domain-specific managers will interact with domain independent managers in the upper level of hierarchy. This domain independent manager is called *Across Layer Manager of Managers (ALMoM)*. The communication interface between domain-specific managers and ALMoM specifies the management information that is exchanged and a messaging protocol required for the exchange of information. This architecture is designed to incorporate the management functionalities such as notification of link failures at the provider network and automated expansion of bandwidth of virtual links at the user network. The domain specific manager at the lower layer (provider network) will come to know of the physical link failures almost immediately which may not be the case with domain-specific

managers of upper layers (user networks). Depending on the network technology and the exchange due to the management protocol, the managers will realize the failure of logical virtual links (due to failure of physical link) with a certain amount of delay. In such cases, early notification of the link failure from the managers of the physical layer network is useful in initiating recovery mechanisms for affected links.

6.1. Illustrative Examples

We now discuss two simple illustrative examples to show the need for multi-network coordination. More extensive results along this line can be found in [13, 16].

The first example consists of four nodes in the provider network and three nodes in the user network as shown in Figure 3. The links connecting the nodes U0, U1 and U2 in the user network are logical links which are routed through the physical links in the provider network. Each of these logical links is allocated the requested bandwidth in the physical layer network, although the actual utilized bandwidth can be smaller than the allocated bandwidth. In Figure 3, each logical link is marked with both the allocated and the utilized bandwidth. Now let us assume that physical link P2-P3 with a capacity of 60 units fails and hence, logical link U0-U2 with the allocated bandwidth of 60 units also fails. Of the two alternative paths for this logical link in the provider network, the maximum capacity available is only 50 units, which is less than the allocated bandwidth to this link originally. In the absence of utilization information of this channel, the manager of the provider network cannot restore logical link U0-U2 due to insufficient bandwidth. However, as the actual utilization of this link at the time of link failure is only 45 units, the logical channel can be restored using the alternative path P0-P2-P1-P3 with a bandwidth of 45 units allocated, giving the perception of complete recovery. This recovery mechanism between the layers is possible only with the knowledge of both utilization and all the possible alternative paths.

In the second example network shown in Figure 4, when the physical link P1-P2 fails, the logical link U1-U2 with allocated bandwidth of 15 units and 100 % utilization is affected. There is no alternative path for this affected logical link with residual bandwidth of 15 Units. Hence, the manager of the physical (provider) network cannot restore the affected logical link. The user network will then try to route the traffic between nodes U1 and U2 through logical links U1-U0 and U0-U2. Although the unused bandwidth of logical link U0-U2 is sufficient to carry the additional traffic of U1-U2, the bandwidth of U0-U1 is not. For logical link U0-U1 to carry this additional traffic, it requires an additional bandwidth of 5 units to be allocated. When requested by the manager of the user network, the provider network allocates

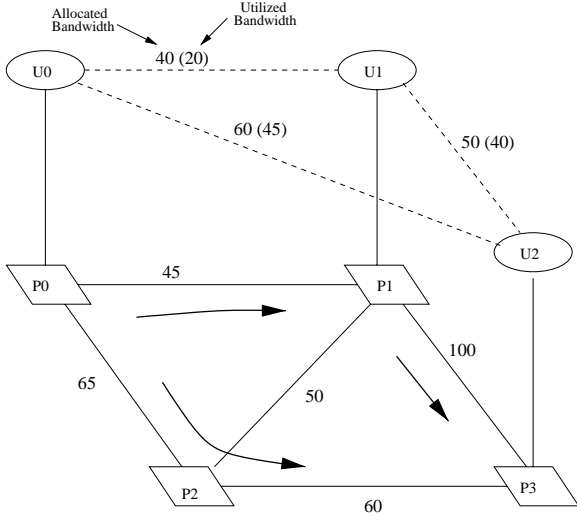


Figure 3. Survivability Mechanism - Example 1

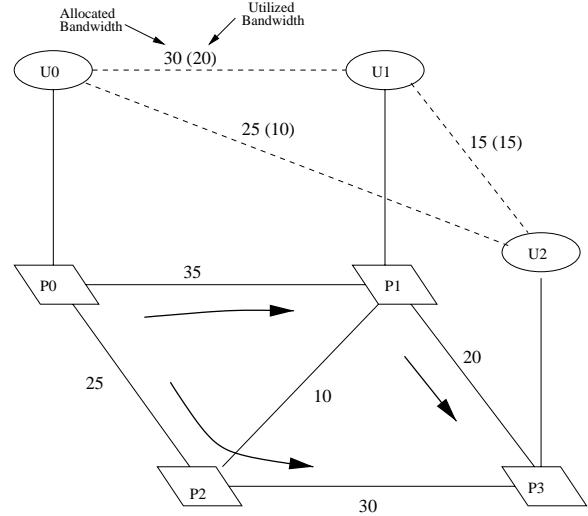


Figure 4. Survivability Mechanism - Example 2

an additional bandwidth of 5 units to logical link U0-U1, so that logical link U1-U2 can be restored. The interaction between the managers of the user network and the provider network is essential in this restoration for two reasons – (1) the manager of the provider network would notify the user network about the failure of the physical link and its inability to restore the affected link, and (2) the manager of the user network negotiates with that of the provider network for the additional bandwidth on logical link U0-U1.

6.2. Architecture of the Multi-layered Network Management System

In the previous section, we have discussed the insufficiency of the domain-specific network managers in a multi-layered network to address a certain level of coordination needed for maximal possible benefits. This type of coordination dictates the need for an integrated management system for multi-layered, multi-domain networks. However, with the increase in size of the network and in number of domains, the amount of management information increases exponentially, hence implementing this integrated management system as a distributed/hierarchical system rather than a single monolithic system is preferable. The choice of a loosely coupled hierarchical distributed architecture is preferred over a flat distributed architecture. In the latter case, each domain specific manager should understand the abstraction of the management information of all other domains, whereas this is not the case with the former.

We propose the loosely coupled hierarchical management system (see Figure 5) where there is an intermediary that interacts with both the upper layer and the lower layer

domain-specific managers. We name such an intermediary component as the *Across Layer Manager of Managers (ALMoM)*. The loosely defined framework allows the possibility that ALMoM may not have control over the internal workings (such as routing) within a specific user network. The capabilities between the user network and the provider network can be negotiated at the time of network manager registration through service level agreements. With ALMoM as the coordinating point, each of the domain-specific managers exchange management information, eliminating the need for the domain specific managers to understand the abstraction of management information of all other domains.

We also expect that within an administrative domain, there can be different sub-networks with varying network technologies, with each supporting different management protocols such as SNMP and CMIP. In such a scenario, the preferred architecture for the domain specific managers is again hierarchical with two sub-layers. In the lower sub-layer, the management components are technology-specific and will be responsible for interaction with management agents in a limited number of network nodes. These components will collect the management information from the nodes and translate it into a technology independent abstraction before transferring it to the manager at the upper sub-layer. The manager at the upper sub-layer will perform most of the critical management functions such as processing and archiving of management information and decision making. In accordance with the management functionalities, the management components at lower and upper sub-layers are called *technology-specific agents* and *domain-specific managers*, respectively. In an administrative domain, the number of technology specific agents (i.e. management components

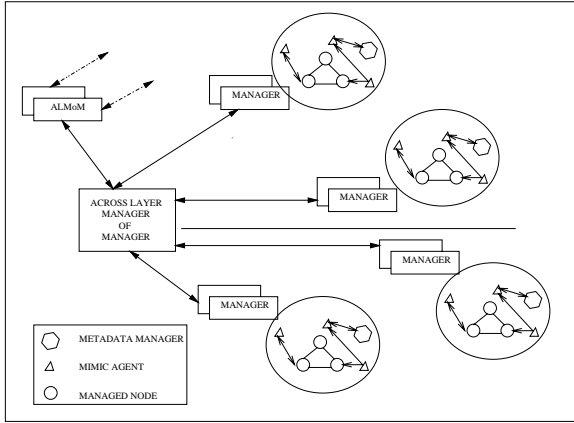


Figure 5. Loosely-Coupled management System for multi-layered Network

at the lower level) depends on the number of subnetworks in the domain and the number of nodes within a sub-network. There can be more than one technology specific agent to monitor a larger sub-network.

The overall architecture of the hierarchical management system for the multi-layered network is shown in Figure 6. This architecture consists of three levels of management components. At the lowest level, technology-specific agents act as proxies between the network nodes and the domain-specific managers at the middle level. The domain-specific managers perform the core of the management functionalities within its administrative domain. The Across Layer Manager of Managers (ALMoM) in the upper-most level is responsible for the inter-domain management functionalities such as survivability and resource management. The exchange of management information between ALMoM and the domain-specific manager is also limited by these functionalities. The role of ALMoM, in this context, can be compared to that of resource trader or broker among the various domains. The interface of communication with management components of the lower level can be of more than one type depending on lower level components. This interface is referred to as the *vertical interface*. Similarly, the interface used for communicating with the management component of the upper level is referred to as the *horizontal interface*. Each management component will interact with more than one management components of the immediately lower level and with only one component in the upper level. The management components in this architecture can also be seen as service providers to the upper level components and as users of service provided by the lower level components. The name *horizontal interface* is used to mean that whichever component in the upper level wants to access the services provided this component, say 'M1', it needs to use

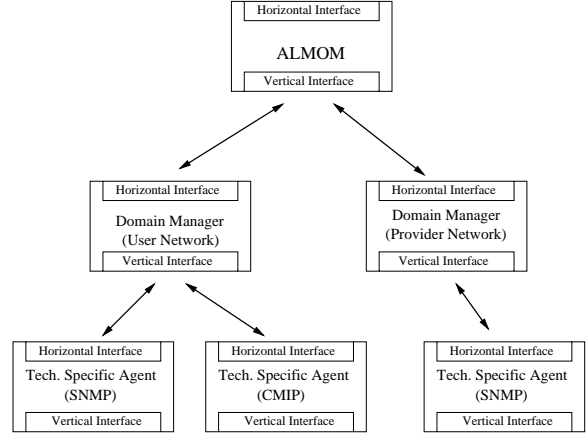


Figure 6. Architecture of Multi-layered Network Management System

the horizontal interface of 'M1'. Similarly, when a management component 'M1' accesses the service offered by a component 'M2' in the lower level, the component 'M1' uses the vertical interface specific to the component 'M2'. In other words, the horizontal interface of another component 'M1' will be the vertical interface of another component 'M2' with respect to the component 'M2'.

The definition of an interface (horizontal or vertical) consists of both management information as well as a set of functions, similar to the definition of a class in object-oriented languages such as C++, Java, etc. The definitions of various interfaces in the hierarchical management system mainly are dependent upon the management functionalities implemented in the domain-specific managers as well as ALMoM. Since the context of this paper is survivability strategies, we shall discuss in more detail the definitions of the interfaces specific to these strategies.

Specific to the survivability mechanisms, the definition of the interface also consists of a set of messages exchanged by ALMoM and the domain-specific managers. For example, when a physical link fails in the provider network, the manager of the provider network sends a message to ALMoM. Then ALMoM would determine the set of logical links affected by this failure in various user networks at the upper layers of the network. Depending on the specific implementation of ALMoM, ALMoM may request the current status of the provider network from its manager, while determining the set of affected components. The ALMoM would inform the managers of administrative domains of user networks whose logical links are affected. This will then initiate a string of message exchanges between ALMoM and domain managers for the restoration of affected areas. Similarly, when one of the administrative domains wants the allocated bandwidth of one of its logical links to be increased, the do-

main manager will send a message to ALMoM to that effect. ALMoM would then determine whether the request can be served. Depending on this determination, ALMoM will initiate a string of message exchanges with domain-specific managers.

Another important aspect of this architecture is that the level of relationship and interaction with ALMoM can be different from one domain-specific manager to another. We envision that each user network may require different levels of survivability requirements. For example, user network-A (UN-A) may require full restoration while user network-B (UN-B) may require partial restoration. What is desirable should be negotiated at the time of registration of the user network domain with ALMoM through a service level agreement (SLA). In such an environment, to address for a failure, ALMoM may be required to correlate notification of failures from different user networks so as to prioritize what restoration to activate in the provider network, so that the user network with the higher priority is restored first, when the bandwidth is limited in the provider network.

Finally, it should be noted that ALMoM, as depicted in Figure 5 does not mean that in practice a single physical entity represents ALMoM. There can be multiple instances of ALMoM for scalability purposes which do distributed networking among themselves, while outside entities such as the user network manager and the provider network manager are transparent to the internal view of ALMoM.

6.3. A Proof-of-Concept Implementation

As a proof of concept, we have implemented our framework for the case of a two layered network with an ATM network as the provider network and an IP network as the user network in the MIMIC testbed at the University of Missouri-Kansas City. Note that this platform is used for a low-cost proof-of-concept demonstration. The basic idea can be implemented in other environment where the user network/provider network paradigm is applicable. In the IP network, we have used FreeBSD routers, while the ATM Network consists of Fore Systems’s LE155 switches.

In our current implementation of ALMoM, the survivability mechanism is the only management functionality that is implemented. To make this implementation operating system and location independent, we have used a CORBA-based object management environment and Java programming language. With the CORBA based approach, we have modularized the management system not only at the component level, but also at the management functionality level. For example, the modular implementation of the manager component of the provider network is shown in Figure 7.

An important advantage of modular implementation is that new management functionalities can be added into each

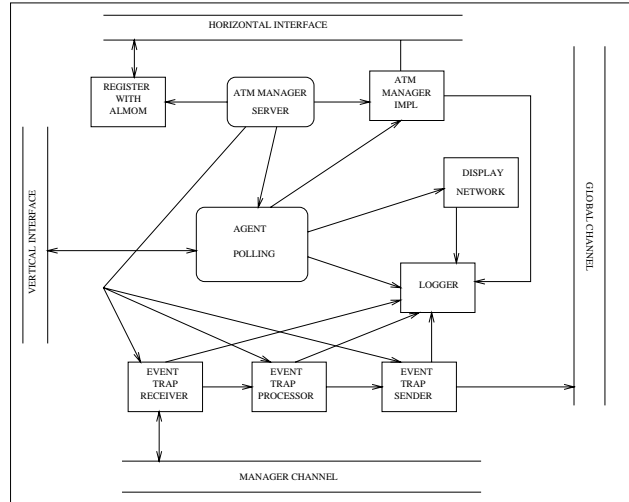


Figure 7. Implementation of the Domain-Specific Manager

of the manager components without much modification to the rest of the implementation. Similarly, *EventChannel* service, which is readily available in the CORBA environment, is used for broadcasting critical events that may have been occurring in the network to various management components. Notably a physical link failure is one such critical event. Depending on the size of the network, there can be more than one *EventChannel* active in the system. In this implementation, we have used two channels, the manager channel and the global channel, providing broadcast service between technology-specific agents and domain-specific managers, and between domain-specific managers and ALMoM, respectively. Each of the events broadcasted in these channels contains enough information, such as event identification, time stamp, origin network and so on, in its header to uniquely identify the event.

7. On-going work

We are currently at the final phase of this project. Several efforts are on-going.

We are currently developing multi-network survivability study tool sets where each of the simulation tools, MoMaRS and MuSDyR, is independently connected with a lower level network optimization-based reconfiguration component. This would then allow us to study a lower layer link failure and its impact on higher-level services (e.g. provider network failure impact on user networks).

In addition, in the MoMaRS tool itself, we are investigating addition of a fault-tolerant multi-cast feature. Similarly, in the MuSDyR tool, we are in the process of adding new routing schemes.

We are also working on the next version of the MIMIC testbed implementation to incorporate some enhancements.

8. Acknowledgments

This work is supported by DARPA and Air Force Research Lab, Air Force Materiel Command, USAF, under agreement No. F30602-97-1-0257. It is also partially supported by NSF Grant NCR-9506652. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Defense Advanced Research Projects Agency (DARPA), Air Force Research Laboratory, or the U.S. Government.

9. Contributors

The work accomplished here would not have been possible without the involvement of various personnel and their dedicated effort. The overall effort at the University of Missouri was led by Deep Medhi while David Tipper took the lead at the University of Pittsburgh.

9.1. University of Missouri-Kansas City

Robert Cotter was instrumental in the development of the survivable virtual ATM network design models, algorithms, and the prototype tool; this is all reported in details in his recently completed Ph.D. dissertation [6].

Several graduate students have been involved in development of MoMaRS simulation tools. Before the project started, Samir Shah started working on the first enhancement to MaRS[3] to consider the multicast shortest path routing component [27]. Then, Gerald Rogers initiated the development of the MKompella multicast routing algorithm and priority-based routing to consider differentiated services environment (reported in his MS thesis [24]; also, see [25]); Suresh Muppala added the RSVP-component to this tool [18] while Ulka Ranadive added the TCP component as well as HTTP workload to this tool [21, 22]. Wen-Jung Hsin as the post-doctoral research associate co-ordinated overall effort related to MoMaRS as well as in the analysis of different network failure scenarios using MoMaRS. Vamsi Valluri and Chengning Lu are currently working on the next version of this tool.

MuSDyR [17] connection-level routing simulation tool to study a failure was designed and developed from scratch by Shishir Ramam, Shankar Subramaniam, and Joshi Sivasankar; their MS theses have reported various aspects

of the tool as well as analysis results [20, 29, 31]. Currently, Aekkachai Rattanadilokchai, Loren Rard and Yiming Huang are working on the next version of this tool.

Mark Hieber is finishing up developing a study environment where MoMaRS is coupled with a lower layer network reconfiguration module to allow multi-layer survivability study. Brenda Groskinsky is developing a numerical differential equation model to analyze the impact of reconfigurability in a dynamic traffic environment [8].

The initial set up and testing of the MIMIC testbed, Fred Summa led the effort while Aanand Ramachandran conducted some of the initial testing [19]. In the next phase, when the multi-network management framework was developed and then implemented, Sanjay Jain, Deepa Shenoy and Mukunda Saddi were heavily involved and their contributions have been documented in their respective theses [11, 26, 28]. Currently, Kaushik Deka, Chetan Desai, Jane Zupan and Dijiang Huang are involved in the updated version of the testbed and the framework.

Finally, Post-doctoral fellow T. Srinivasa Rao and Research Associate Wen-Jung Hsin have been key players who plunged into several aspects of the project, specifically new optimization model development, simulation analysis and in the test-bed implementation while Fred Summa has managed to bring sanity to the project by making sure that we have a working environment.

9.2. University of Pittsburgh

Several individuals have contributed to this project at the University of Pittsburgh. In particular Adel Al-Rumaih developed the genetic algorithm based spare capacity planning method and its implementation into a design tool in his Ph.D. dissertation [1]. Graduate student Yu Liu has contributed to the relative comparison study of the genetic algorithm approach and mathematical programming techniques reported in [2]. Ph.D. candidate Anotai Srikitja developed the extended version of the NS simulator and conducted the study on survivable service in NGI reported in [30]. Graduate student Wei-Ping Wang has contributed to the development of fault recovery routing algorithms reported in [34]. Ph.D. candidate William Yurcik developed the survivable ring approach to providing fault tolerant group communications detailed in [35] and his forthcoming Ph.D. dissertation. Visiting scholar Bjorn Jager has contributed to the development of fault recovery routing algorithms and priority restoration schemes as detailed in [34, 10] and his forthcoming doctoral dissertation [9].

References

- [1] A. Al-Rumaih. *A Spare Capacity Planning Methodology for Wide Area Survivable Networks*. Ph.D. dissertation, Department of Information Science and Telecommunications, University of Pittsburgh, May 1999.
- [2] A. Al-Rumaih, D. Tipper, Y. Liu, and B. Norman. *Spare Capacity Planning for Survivable Mesh Networks*. Technical Report, Department of Information Science and Telecommunications, University of Pittsburgh, September 1999.
- [3] C. Alaettinoglu, A. U. Shankar, K. Dussa-Zieger, and I. Matta. Design and implementation of MaRS: A routing testbed. *Journal of Internetworking — Research and Experience*, 5(1):17–41, March 1994.
- [4] Y. Bernet *et. al.* *A Framework for Differentiated Services*. Internet Draft draft-ietf-diffserv-framework-02.txt, February 1999.
- [5] R. Braden, D. Clark, and S. Shenker. *Integrated Services in the Internet Architecture: an Overview*. Internet RFC 1633, June 1994.
- [6] R. Cotter. *Network and Protocol Design for Survivable Wide-Area Virtual ATM-based Networks*. Ph.D. dissertation, Computer Networking and Telecommunication Networking, University of Missouri-Kansas City, September 1999.
- [7] D. E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, Reading, Mass., 1989.
- [8] B. Groskinsky, D. Medhi, and D. Tipper. *An Investigation Of Capacity Control Schemes In A Dynamic Traffic Environment*. Technical Report, Computer Science Telecommunications, University of Missouri-Kansas City, July 1999.
- [9] B. Jager. *Traffic Restoration in Survivable Wide Area Communication Networks*. forthcoming Ph.D. dissertation, Department of Informatics, University of Bergen, Bergen, Norway, December 1999.
- [10] B. Jager and D. Tipper. On fault recovery priority in ATM networks. *Proc. of IEEE ICC 98*, June 1998.
- [11] S. Jain. *Design and Implementation of Management Components for Multi-Layered Survivable Networks*. MS Thesis, Computer Science Telecommunications, University of Missouri-Kansas City, May 1999.
- [12] V. Kompella, J. Pasquale, , and G. C. Polyzos. Multicast-routing for multimedia communication. *IEEE/ACM Transaction on Networking*, 1:286–291, 1993.
- [13] D. Medhi. A unified approach to network survivability for teletraffic networks: Models, algorithms and analysis. *IEEE Trans. on Communications*, 42:534–548, 1994.
- [14] D. Medhi. Multi-hour, multi-traffic class network design for virtual path-based dynamically reconfigurable wide-area ATM networks. *IEEE/ACM Trans. on Networking*, 3(6):809–818, December 1995.
- [15] D. Medhi, S. Jain, T. Srinivasa Rao, D. Shenoy, M. Saddi, and F. Summa. *A Network Management Framework for Multi-Layered Network Survivability*. Technical Report, Computer Science Telecommunications, University of Missouri-Kansas City, July 1999.
- [16] D. Medhi and R. Khurana. Optimization and performance of network restoration schemes for wide-area teletraffic networks. *Journal of Network and Systems Management*, 3(3):265–294, September 1995.
- [17] D. Medhi, S. Ramam, R. J. Sivasankar, and S. P. Subramaniam. *Design of MuSDyR: version 1.0*. Technical Report, Computer Science Telecommunications, University of Missouri-Kansas City, July 1999.
- [18] S. Muppala. *Network Performance under Failure in the presence of RSVP-based Multicast Services*. MS Thesis, Computer Science Telecommunications, University of Missouri-Kansas City, May 1999.
- [19] A. Ramachandran. *On Restoration in ATM Networks and Measurements in a Testbed*. MS Thesis, Computer Science Telecommunications, University of Missouri-Kansas City, June 1998.
- [20] S. Ramam. *Multicast Routing with Reservation in Wide-Area Networks*. MS Thesis, Computer Science Telecommunications, University of Missouri-Kansas City, August 1999.
- [21] U. Ranadive. *Effect of Network Dynamics on TCP: Observations and Possible Solutions*. MS Thesis, Computer Science Telecommunications, University of Missouri-Kansas City, August 1999.
- [22] U. Ranadive and D. Medhi. *Some Observations on the Effect of Network Dynamics on TCP*. Technical Report, Computer Science Telecommunications, University of Missouri-Kansas City, July 1999.
- [23] T. S. Rao and D. Medhi. *Book-Ahead Guaranteed Survivable Services in Wide-Area Networks: Problem Formulation and Results*. Technical Report, Computer Science Telecommunications, University of Missouri-Kansas City, July 1999.
- [24] G. Rogers. *Analysis of Multicast and Priority based Routing Algorithms with Physical Failures in a Packet Switched Networking Environment*. MS Thesis, Computer Science Telecommunications, University of Missouri-Kansas City, July 1998.
- [25] G. Rogers, D. Medhi, W.-J. Hsin, S. Muppala, and D. Tipper. Performance analysis of multicast and priority-based routing under a failure in Differentiated-Services Internet. *Proc. of IEEE MILCOM'99*, October 1999.
- [26] M. Saddi. *Network Management System with User Network/Provider Network Paradigm for Multi-Layered Survivable Networks : IP based Implementation*. MS Thesis, Computer Science Telecommunications, University of Missouri-Kansas City, August 1999.
- [27] S. Shah and D. Medhi. Performance under a failure of wide-area datagram networks with unicast and multicast traffic routing. *Proc. of IEEE MILCOM'98*, October 1998.

- [28] D. Shenoy. *A Network Management Framework for Multiple Layer Survivable Networks: Protocol Development and Implementation*. MS Thesis, Computer Science Telecommunications, University of Missouri-Kansas City, May 1999.
- [29] R. Sivasankar. *A Study on Dynamic Traffic in a QoS Routing Environment*. MS Thesis, Computer Science Telecommunications, University of Missouri-Kansas City, August 1999.
- [30] A. Srikitja, D. Tipper, and D. Medhi. On providing survivable services in the Next Generation Internet. *Proc. of IEEE MILCOM'99*, October 1999.
- [31] S. Subramaniam. *Performance of Dynamic Quality of Service Routing under the Influence of Link Failures*. MS Thesis, Computer Science Telecommunications, University of Missouri-Kansas City, July 1999.
- [32] D. Tipper, J. Hammond, S. Sharma, A. Khetan, K. Balakrishnan, and S. Menon. An analysis of the congestion effects of link failures in wide area networks. *IEEE Jnl. Sel. Areas Comm*, 12:179–192, 1994.
- [33] UCB/LBNL/VINT. Network simulator-ns (version 2). <http://www-mash.cs.berkeley.edu/ns/ns.html>.
- [34] W. Wang, B. Jager, D. Tipper, and D. Medhi. *On Virtual Circuit Fault Recovery Routing*. Technical Report, Department of Information Science and Telecommunications, University of Pittsburgh, September 1999.
- [35] W. Yurcik, D. Tipper, and D. Medhi. Providing network survivability to atm group communications via self healing survivable rings. *Proc. of 7th International Conference on Telecom. Systems*, March 1998.

This paper appears in the Proceedings of the *DARPA Information Survivability Conference and Exposition (DISCEX'2000)*, Hilton Head Island, South Carolina, January 25-27, 2000.