INTERNATIONAL JOURNAL OF NETWORK MANAGEMENT
Int. J. Network Mgmt 2012; **22**: 50–64
Published online 24 May 2011 in Wiley Online Library (wileyonlinelibrary.com) DOI: 10.1002/nem.785

GigaManP2P: an overlay network for distributed QoS management and resilient routing

Elias P. Duarte Jr[1,*,†], Lisandro Z. Granville[2], Luci Pirmez[3], José Neuman de Souza[3], Rossana C. Andrade[4], Liane Tarouco[2], Reinaldo B. Correia[3] and Alexandre Lages[3]

[1]*Department of Informatics, Federal University of Paraná, Curitiba, PR 8531–980, Brazil*
[2]*Department of Informatics, Federal University of Rio Grande do Sul, Porto Alegre, RS 91501–970, Brazil*
[3]*Federal University of Rio de Janeiro, NCE, Cidade Universitária, Rio de Janeiro, RJ 20010–974, Brazil*
[4]*Department of Computer Science, Federal University of Ceará, Fortaleza, CE 60455–760, Brazil*

SUMMARY

Management of long-distance, high-speed optical backbones spanning multiple administrative domains requires new solutions for challenging tasks. In particular, it is not trivial to negotiate, monitor and continuously enforce the required quality of service (QoS) for applications that span multiple domains. This paper proposes GigaManP2P: a novel peer-to-peer (P2P) management architecture for high-speed QoS-aware backbones. GigaManP2P peers provide management services in a ubiquitous fashion through modules that interface with both the communication infrastructure and network users. In particular, we describe management services for on inter-domain QoS monitoring and resilient routing. After detecting a QoS constraint violation trend, a proactive rerouting strategy is triggered based on redundant virtual circuits, allowing both full and partial rerouting. The P2P overlay implementation is the basis for allowing transparent communication across autonomous systems. Experimental results showing the overhead of the P2P infrastructure in comparison to raw Simple Network Management Protocol, and the performance of the rerouting strategy, are presented. Copyright © 2011 John Wiley & Sons, Ltd.

Received 11 May 2010; Revised 14 March 2011; Accepted 20 March 2011

1. INTRODUCTION

The proper provisioning of quality of service (QoS) in computer networks depends, among other factors, on the efficient and accurate management of underlying communication infrastructures [1,2]. QoS management has become a key research challenge [1,3]; only properly managed QoS-enabled networks allow the deployment of QoS-sensitive applications such as high-definition digital television (HDTV), telemedicine, real-time remote control, among many others. In the recent past, QoS management was primarily concerned about the proper use of the available and frequently scarce network bandwidth [4]. Nowadays, with the adoption of optical communication technologies, bandwidth is no longer the primary concern. In this scenario, other QoS-related issues have become more important, such as enforcing and monitoring QoS constraints across several autonomous systems (AS). Traditional network management solutions are unable to cope with these issues, and they must be solved by managers of every long-distance, high-speed optical backbone spanning multiple administrative domains.

Optical systems usually present a number of QoS management functions at the lower layers [5–8]. These functions only address optical layer issues [9], and are often loosely integrated with higher-level

[*]Correspondence to: Elias P. Duarte Jr., Department of Informatics, Federal University of Paraná, P.O. Box 19018, Curitiba PR, 81531-980 Brazil.
[†]E-mail: elias@inf.ufpr.br

management systems. Although this has advantages due to the fact that lower-level operations are transparent to the higher layers, there are cases in which a tighter integration could bring benefits for the network operation. An example is rerouting, which, at the optical layer, affects a potentially huge number of flows when a single wavelength is rerouted; furthermore, optical layer rerouting typically triggers a chain of restoration actions in the higher layers that is very difficult to manage and may incur inconsistencies and instabilities in the whole system. In an integrated management architecture rerouting could be more effective by taking into account the requirements of specific applications and their flows.

In this work we present GigaManP2P, a peer-to-peer network QoS management overlay for monitoring and enforcing QoS for applications that span multiple domains. Management services are presented that allow fine-grained rerouting (i.e. enabling specific flows or limited groups of flows to be rerouted without affecting unrelated ones) operating across multiple administrative domains. The QoS management strategy is based on proactive rerouting [10,11], i.e. flows are rerouted prior to the occurrence of a QoS violation. The P2P overlay implementation is the basis for allowing transparent communication across autonomous systems.

The GigaManP2P overlay offers management services considering three different types of clients: network operators, end-users, and end-user applications. Management peers are placed on a set of inexpensive hosts and are responsible for high-level management tasks [12], while conventional Simple Network Management Protocol (SNMP) agents [13] run at managed network devices, such as optical switches, undertaking simpler tasks. Rerouting-aware management peers (also called rerouting agents), operating above the optical infrastructure, dynamically create and monitor Multi-Protocol Label Switching (MPLS) virtual circuits through which affected flows are routed.

GigaManP2P was originally conceived for the multi-AS Brazilian RNP Giga backbone [14]. The RNP backbone is the Brazilian academic network that reaches all 27 Brazilian states with aggregated traffic capacity of 366 Mbps. It has international connections with the USA, summing an external connectivity of 200 Mbps. One of these international connections is dedicated to Internet2, ensuring Brazilian access to the project that gathers academic networks in several countries. Another international connection is to Europe. RNP2 interconnects 329 national education and research institutions, serving around 800 000 users. RNP also acts in the testing and development of pilot applications for the network and in the qualification of human resources to operate the 27 points of presence (PoPs), in areas such as network security, IP management, routing, high-performance networks, system administration, new protocols and services. Experimental results showing the overhead of the P2P infrastructure in comparison to raw SNMP are presented, as well as simulations showing the performance of the rerouting strategy.

The rest of this work is organized as follows. In Section 2, the architecture of GigaManP2P is described. The proactive rerouting strategy is presented in Section 3. Experimental results obtained from simulation, in which we evaluate both the overhead and latency of the proposed approach, follow in Section 4. Section 5 discusses related work, while Section 6 presents concluding remarks and future work.

2. GigaManP2P: A QoS MANAGEMENT OVERLAY

P2P systems [15] consist of nodes (peers) typically running at inexpensive end-user hosts. Peers establish logical connections with one another, forming an overlay network. In this section we describe the architecture and operations of GigaManP2P, an overlay conceived for managing the multi-AS Brazilian RNP Giga backbone [14]. GigaManP2P offers a set of management services that integrates end-users and their applications with the communication infrastructure.

There are several reasons that justify the use of a P2P architecture for network management systems. P2P protocols have been designed considering the peculiarities of the current Internet: for instance, the extensive use of network address translations (NATs) and firewalls. The P2P overlay implementation is the basis for allowing transparent communication across autonomous systems. A P2P overlay allows users to exchange information more easily across NAT/firewalls, and this is not the case for conventional management protocols. The use of a P2P overlay allows for greater

DOI: 10.1002/nem

connectivity between management nodes as multiple paths can be used for pairs of nodes to exchange information, e.g. IP provides, in most cases, only one path between pairs of nodes. P2P overlays also often implement multicast communications, which in several cases are very useful for accomplishing management tasks. When using P2P technologies one 'inherits' a self-organizing network, in which administration of nodes entering and leaving are provided by the underlying network itself. Thus there is no need to implement any control mechanism.

In this section the system architecture is presented. GigaManP2P spans multiple layers, providing means for users to monitor and control the underlying physical system. Management services are then described. In the next section the QoS-based management service is described.

Figure 1 presents the GigaManP2P architecture. At the highest level, peers communicate in forming the management overlay which provides the client interface. Peers monitor and configure the MPLS-enabled devices at the intermediate level, which establish virtual circuits (Label Switched Paths—LSPs) given the QoS constraints of flows. The lower level represents the optical communication infrastructure. The following subsections describe the internal components of the system in more detail.

2.1. GigaManP2P peer architecture

GigaManP2P offers management services considering three different types of clients: network operators, end-users, and end-user applications. Figure 2 presents the general environment where one can observe the optical infrastructure, the management overlay, and its clients. Peers are placed across the managed optical network to monitor and control network devices found at different administrative domains. Each peer locally offers management services to local clients (i.e. local network operators, local users, and local user applications). In addition, each peer provides additional services to other remote peers forming the management overlay. The architecture of each peer is composed of elements also presented in Figure 2.

The basic communication services (i.e. Hypertext Transfer Protocol/Hypertext Transfer Protocol Secure and Juxtapose—HTTP/HTTPS and JXTA, respectively [16]) are placed at the higher layer of the architecture, enabling communication between peers and clients, and among peers themselves. JXTA is an open source Java framework that enables the development of new P2P applications. The framework consists of modular elements that implement basic P2P functionalities such as service advertisement and discovery, group communication, and application routing. We employed the JXTA framework to construct the GigaManP2P overlay. The P2P framework on which GigaManP2P is implemented must provide the security services required. The system has to guarantee secure

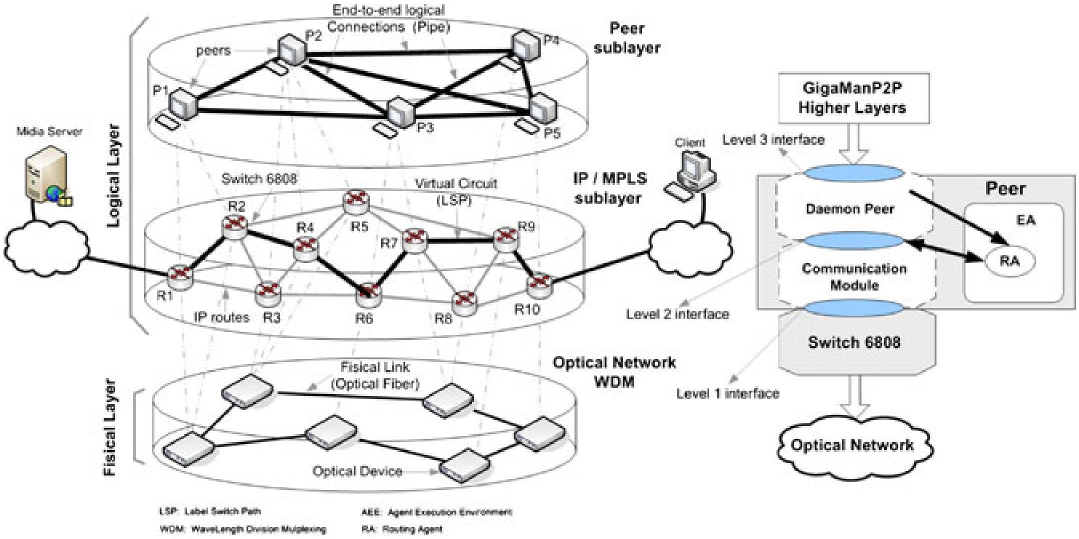

Figure 1. GigaManP2P architecture.

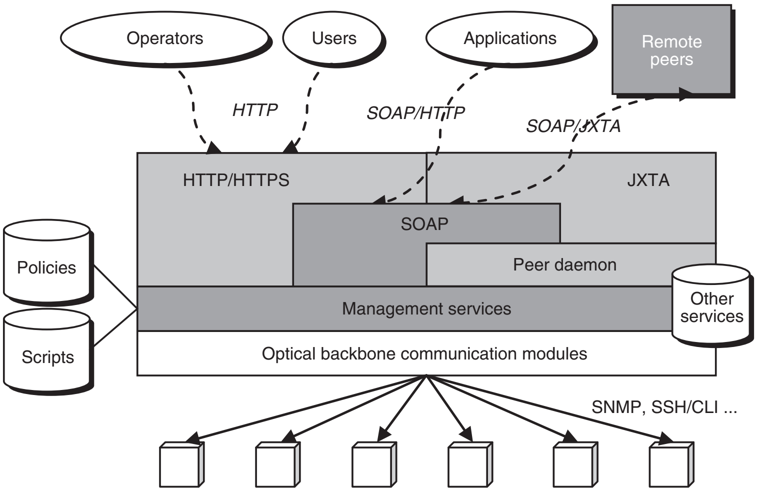

Figure 2. Peer architecture.

communication among peers, even those deployed at different administrative domains—this is the case with JXTA.

The general operation of a peer is controlled by the peer daemon that orchestrates the operations of other internal elements. Each peer exposes a set of management services that are accessed using both HTTP (when users need a Web interface to access the management services) and SOAP (when remote peers need to access the services in a remote procedure call fashion). Since Simple Object Access Protocol (SOAP) [17] supports remote procedure calls, it has been employed for this purpose.

Each management service may require some data repositories for its operations. For example, management policies are stored in a specific repository, as well as management scripts that can be transferred to a peer for future execution. Management services themselves are stored in a service repository. This allows the dynamic deployment of new services which can be transferred to the local service repository of a peer at any time.

The modules for the communication of peers with the optical infrastructure are located at the bottom of the peer architecture. They provide an interface to access optical devices in a transparent way, regardless of the actual management protocol used to access the managed devices. These modules in fact implement an adaptation layer used by the remaining internal elements when communication with optical devices is required. The current implementation of GigaManP2P peers supports SNMP (for instrumentation) and SSH/CLI (for configuration).

2.2. Communication between clients and peers

Clients access GigaManP2P to request the execution of a management service. The set of available services are placed at a level above the adaptation layer mentioned before. If the set of management services needs to be expanded, new services can be installed using the P2P overlay itself. Such new services are stored in the ‘other services’ repository. New services can also be deployed as management scripts, stored in the script repository. The difference between scripts and the services from the ‘other services’ repository is that scripts implement simpler services that tend to have a fixed execution schedule and, after completion, are removed from the repository. Regular services, in turn, are more complex services that extend the basic peer functionality and are not discarded after their execution.

Network operators are GigaManP2P clients that access the management services available to them through either a local GigaManP2P peer or via dynamic Web pages exposed by remote peers using HTTP. One of the key items operators are responsible for is the definition of management policies that are stored in the distributed database formed by the collection of local policy repositories of each peer.

End-users are clients that also access management services through dynamic Web pages. The set of services available to end-users, however, is restricted in comparison to the set of services available to network operators. For example, end-users cannot define management policies. End-user applications are clients able to access the same services available to end-users. The difference resides in the fact that end-user applications use Web services interfaces via SOAP/HTTP requests, while a human user accesses the same services via conventional Web pages.

2.3. Management services

GigaManP2P offers management services to operators, end-users, and applications. These services are organized as follows.

2.3.1. Services for network operators

Network operators are responsible to run the managed network to fulfill end-users needs. The services for network operators allows: handling and enforcing management policies, distributing and controlling the execution of management scripts, distributing mobile agents, and monitoring network connectivity. Handling and using management policies allow the operator to define how the optical network must behave in response to end-user and application requests. Management peers, in this case, act as Policy Decision Points (PDPs) of the Internet Engineering Task Force (IETF) policy-based network management framework [18].

The management script distribution and execution offer a mechanism to the operator that allows a request to be issued to the P2P system for the execution of specific tasks (e.g. MPLS path allocation). In this case, each peer acts as an environment to remotely execute management scripts.

The distribution of mobile agents is similar to the management script support, except that in this case mobile agents are used as a mechanism to expand the available management services, installing new ones. Therefore, the mobile agent support can be seen as a meta-service whose goal is to allow the operator to install new management services.

Finally, the services for network connectivity monitoring enable the operator to check the current optical infrastructure status in order to take management decisions concerning the allocation of network resources to end-users. All services available to the operators are accessed via a peer Web interface using HTTP/HTTPS.

2.3.2. Services for end-users

Offering management services to end-users allows them to have some level of control over the network resources required to accomplish their tasks. The services available to end-users are QoS negotiation and retrieving network status reports. QoS negotiation allows end-users to schedule QoS support for mission-critical traffic, i.e. each peer acts as a QoS Bandwidth Broker [4]. Although end-users can request QoS support, that is only accomplished if the user request is in accordance with the management policies defined by the network operator. The service of reporting the network status asynchronously notifies end-users (e.g. via e-mail messages) regarding the changes in optical infrastructure.

2.3.3. Services for end-user applications

The same services available to end-users are available to applications. However, the interface to such services is different. Instead of using Web pages, applications access the services via SOAP [17]. Management services offered to applications are especially important for cooperative and resource-sharing applications such as grids, where the network infrastructure needs to be configured to provide proper communication facilities.

2.3.4. Services for peers

QoS monitoring and management is a key service provided and executed by the proposed overlay. The optical network is continuously monitored, and an information service provides reports on when and why the expected QoS is not being properly provided. Management peers proceed with proactive management, where QoS maintenance actions are executed prior to a QoS violation, as described in the next session. Such actions can be triggered after a peer evaluates management data exchanged with other peers. Specific services are employed by the overlay to guarantee resilient routing,

DOI: 10.1002/nem

including a service to establish virtual circuits for applications with QoS requirements, and monitoring and proactive rerouting service; both are described in the next session.

3. QoS MANAGEMENT AND RESILIENT ROUTING IN GigaManP2P

GigaManP2P employs a distributed approach for QoS management. Initially a QoS negotiation service is employed to determine whether a flow can be routed given its QoS requirements and the network conditions. After a virtual circuit is established, the monitoring service is instantiated. Rerouting agents are employed for setting and monitoring virtual circuits, as well as rerouting itself. Rerouting starts with the discovery of a critical path—a subset of the virtual circuit to be replaced. The critical path is determined taking into account QoS metrics relevant to the application. The rerouting strategy selects alternative paths for the monitored flow which replace the critical path whenever a QoS degradation trend is detected, before users perceive a QoS degradation. The strategy is proactive—not reactive—in the sense that it anticipates the violation of QoS requirements. The strategy also allows both full and partial rerouting, depending on whether the broken virtual circuit is completely or partially replaced by the new one.

3.1. Rerouting agents

Three types of agents support the proactive rerouting strategy: *InputNodeAgent*, *IntermediateNodeAgent*, and *AlternativeRouteAgent*. The *InputNodeAgent* operates at the input (or first) device of the virtual circuit. This agent triggers the rerouting of flows belonging to the same virtual circuit and is also responsible for interacting with external modules (e.g. the GigaManP2P service used to create virtual circuits), offering the interface through which routing services are accessed by the rest of the system.

The *IntermediateNodeAgent* has two main goals: (i) to monitor the network devices that belong to the virtual circuit; and (ii) to feed the *InputNodeAgent* with performance information about the QoS metrics relevant to the flow. Finally, the *AlternativeRouteAgent* is responsible for discovering alternative paths. The discovery mechanism operates in a limited area around the critical path called the *search area*. In addition, the *AlternativeRouteAgent* has two other goals: (i) to select the best alternative path; and (ii) to reconfigure the devices in order to establish the new route.

3.2. Rerouting phases

The complete proactive rerouting process consists of five phases (Figure 3) in which the operations required to reroute a flow take place: (1) *agent activation*; (2) *virtual circuit monitoring*; (3) *discovery of alternative routes*; (4) *alternative route monitoring*; (5) *route change configuration*. These phases (Figure 3), except the route change configuration phase, are executed sequentially after a flow with QoS requirements starts. The route change configuration is executed after a QoS violation trend is detected. This strategy implements a proactive rerouting scheme, in the sense that the first four phases are executed before any QoS failure occurs.

In the *agent activation* phase, GigaManP2P activates an *InputNodeAgent* at the peer responsible for the virtual circuit's input (the first) routing device. The *InputNodeAgent* then activates an *IntermediateNodeAgent* at the next peer responsible for controlling the network device in the route to the destination. This *IntermediateNodeAgent* then activates another *IntermediateNodeAgent* at the next peer to the destination, and so on, step by step until all devices have an associated *IntermediateNodeAgent* activated.

The next phase, *virtual circuit monitoring*, starts immediately after the last *IntermediateNodeAgent* is activated at the peer responsible for the output device of the virtual circuit. This phase is concluded only when the flow finishes. During this phase the operations required to obtain relevant information from the virtual circuit are executed, and subsequently sent to the *InputNodeAgent*. A message—generated at the last *IntermediateNodeAgent* and which traverses all *IntermediateNodeAgents* backwards to the *InputNodeAgent*—carries monitoring information, i.e. QoS parameters, which may be configured for each flow depending on its requirements. Periodically the *IntermediateNodeAgent* at

DOI: 10.1002/nem

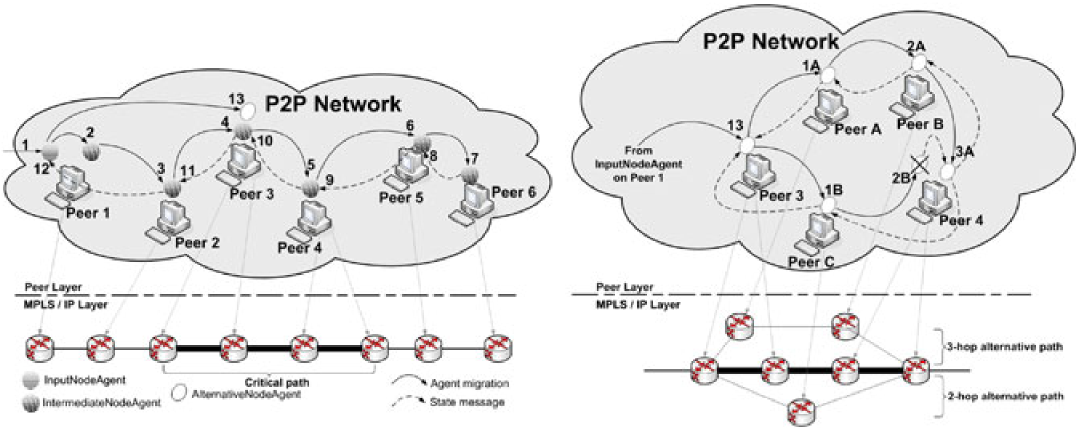

Figure 3. Rerouting phases.

the last node creates this message with monitoring information and sends the message backwards to the previous *IntermediateNodeAgent*, which updates the information and sends the message in turn to the previous agent, and so on, until the message reaches the *InputNodeAgent*.

The sequential monitoring strategy allows the discovery of the *critical paths*. A critical path is a part of the route which is the bottleneck for the set of monitored parameters across the whole route. After a critical path is detected, an *AlternativeRouteAgent* is activated by the *InputNodeAgent* at the first node of the critical path, i.e. the peer of the critical path that is closest to the *InputNodeAgent*.

The *discovery of alternative routes* phase starts after the *AlternativeRouteAgent* is activated at the peer responsible for the first device of the critical path. This phase concludes after the last *AlternativeRouteAgent* is activated at the peer of the last device of the critical path, indicating the discovery of the last alternative route. A simple algorithm is employed for the discovery of alternative routes: the limited diffusion of AlternativeRouteAgents in a search area with a predefined search radius, *r*. The size of the search area is a parameter that can be set according to the topology; for instance, for sparse networks *r* must be larger than for dense networks. In general, *r* is the size of the critical path times a constant; in our experiments we used this constant equal to 2. Formally, the search area is defined as subgraph $G=(V, E)$ in which the set of vertices in V consist of the first node in the critical path plus all nodes to which the distance is at most r; the set of edges (i,j) in E consist of the links that connect nodes i, j both in V. The generation and association of labels to the new virtual circuit can occur either in this phase or later. If labels are associated in this phase, the approach is called *anticipated*. The other approach, called *on-demand*, is described below.

In the *alternative route monitoring* phase, information about the alternative routes are obtained and sent periodically to the *InputNodeAgent*. Messages flow through the *AlternativeRouteAgents* up to the agent at the last peer of the alternative path. Each agent updates the message with local information. The destination of these messages is the *AlternativeRouteAgent* of the first peer of the critical path. These messages contain information that allows the *AlternativeRouteAgent* to choose the best path to employ given the resources available and the requirements of the QoS flow to be rerouted. Note that if two different virtual circuits have a QoS degradation detected, then it is possible that both *AlternativeRouteAgents* will try to associate labels with the same alternative path, and one or both will succeed depending on the capacity of the involved links and the amount of traffic they are currently carrying.

The *route change configuration* phase consists of the set of operations executed after a rerouting request is issued by an *IntermediateNodeAgent*. This phase is responsible for redirecting the flow to the alternative path. The redirection is executed by the *AlternativeRouteAgent* at the peer of the first node of the critical path. The operations involved in this phase depend on the approach adopted for the generation and association of labels to the new virtual circuit, which can be anticipated or on demand. In the anticipated approach, the labels are generated in the *discovery of alternative routes* phase (proactive phase), as mentioned above. In this case, there is no impact on the latency of the *route configuration* phase, but there is a higher consumption of labels. On the other hand, the on-demand

DOI: 10.1002/nem

approach causes an increase in the latency of the *route configuration* phase, since the operations related to the creation and association of labels must be executed in this phase. Moreover, the latency of the *route configuration* phase becomes dependent on the length of the alternative path, which does not occur in the anticipated scheme.

4. EXPERIMENTAL RESULTS

In this section we present a set of experiments executed using both simulations and actual deployment of our proposed overlay on the top of a real Giga network. We initially present the delays for activating the rerouting agents. Then the overhead of the P2P infrastructure in comparison with raw SNMP is evaluated. The third experiment evaluates the influence of the topology on the rerouting process. The last experiment evaluates rerouting in the presence of different kinds of background traffic.

4.1. Agent activation delays

In order to measure the delays for activating and executing the rerouting agents, a test environment was set up with two hosts running Linux, both of which are based on Intel processors and connected by a 100 Mbps Ethernet. Table 1 shows the average results obtained for this experiment; each one was repeated at least 50 times. The measured delays correspond to agent instantiation, serialization, and configuration.

For the *InputNodeAgent*, the average delay (392.4 ms) corresponds to the interval from the instant the requesting management service starts the agent, until the agent is instantiated at the first node of the virtual circuit. For the *IntermediateNodeAgent*, the average delay (518.7 ms) corresponds to the moment an *IntermediateNodeAgent* is instantiated, executes locally, and sets up the next *IntermediateNodeAgent* in the virtual circuit. For the *AlternativeRouteAgent*, the average delay (259.0 ms) corresponds to the instant an *AlternativeRouteAgent* is instantiated, executes locally, and sets up the next *AlternativeRouteAgent* in the search area in which alternative paths are to be determined. The *AlternativeRouteAgent* migrates to every neighbor, while the *IntermediateNodeAgent* has to access virtual circuit information in order to determine the next hop to migrate to.

4.2. SNMP overhead

In this section, experimental results from the evaluation of the interaction of the P2P infrastructure on the routing devices through SNMP are presented. The test of the communication module consists on measuring the latency as perceived by the client when invoking management methods. More specifically, we measured the time from the instant an agent issues an SNMP request until the response arrives. The methods implement common management functions such as Management Information Base (MIB) object recovery and MIB object update. The processing time was measured considering two implementations: (i) in the first measurement SnmpAPI objects, supplied by AdventNet, were evaluated; (ii) in the second measurement, SNMP commands were executed via a command line interface (CLI) triggered from the Runtime class of the Java application programming interface (API). Table 2 summarizes the processing average times obtained with the two approaches. The environment employed for this experiment consisted of SNMP applications executed on a Pentium-based host running Linux, while the SNMP agent was running on an Extreme Networks BlackDiamond 6808 switch [19]. A dedicated 100 Mbps Ethernet segment was employed. Average results are shown in Table 2; each experiment was repeated 50 times.

Table 1. Agent activation delays.

	Agent			Messages
	InputNodeAgent	IntermediateNodeAgent	AlternativeNodeAgent	
Delay (ms)	392.4	518.7	259.0	4.0

DOI: 10.1002/nem

Table 2. SNMP APIs overhead (ms).

	AdventNet API				Java Runtime Class			
	Get	GetNext	GetBulk	Set	Get	GetNext	GetBulk	Set
Medium	1305.66	1301.13	1349.57	1312.4	51.91	52.36	53.24	47.73
Standard deviation	13.627	23.697	15.348	21.289	1.58	0.70	0.83	6.96
Confidence interval (95%)	±4.876	±8.480	±5.492	±7.618	±0.567	±0.251	±0.298	±2.491

The time intervals measured for the AdventNet SnmpAPI [20] are relatively high due to the overhead of the additional processing introduced by this API. In order to send an SNMP message to the agent, a set of API classes are instantiated to generate objects which will process the SNMP requests. This overhead corresponds to 96.02% of the total average time (SNMP Get command) presented in Table 2, i.e. the effective latency of processing a Java-based SNMP request corresponds, on average, to 51.91 ms. However, the highest overhead of the first approach occurs only when the first request is processed. We assume that this is due to the class instantiation, as mentioned previously. From the second request, due to caching, some objects will be already loaded in memory. Thus, an extra test was carried out in order to measure the average time of the execution of SNMP commands in this situation.

In order to improve the performance of the P2P infrastructure, it became evident from the test results that the SNMP software used in the overlay should be better handled. One possibility is the inclusion of a new component, called *utility agent*, which would visit all the managed devices during the system start-up, forcing SnmpAPI objects to be pre-loaded. Utility agents would then remain active as long as the system is running. The inclusion of this new component would allow a substantial reduction of the time spent by the agents for sending SNMP requests.

In conclusion, Table 2 shows that the way SNMP objects are used can have a severe impact on the performance of the management overlay. The AdventNet API was employed because of several facilities provided and its high-level programming abstractions. While we did expect that Java's native Runtime class would present lower delays, the difference was actually very significant. The measured delay for AdventNet's implementation was close to 26 times worse than the delay for Java's Runtime class. However, with the use of the Java Runtime class the size of the code of rerouting agents increased, with a direct impact on the activation latency.

4.3. Evaluating rerouting on an example topology

In this experiment we evaluated the feasibility of executing partial rerouting in an example topology, shown in Figure 4, which resembles the academic Brazilian RNP backbone. Two metrics were defined: *NodeReroutingIndex* and *VCReroutingIndex*. The *NodeReroutingIndex* is defined as the number of pairs of nodes that are connected by at least two different paths of a given size divided by the total number of node pairs in the network. The *VCReroutingIndex* is defined as the number of critical paths of size k for which there is an alternative path of a given size, divided by the total number of critical paths with size k. These indices reflect how redundant the network is, i.e. how feasible it is to reroute in the network. The first index considers pairs of nodes, and whether there are alternative

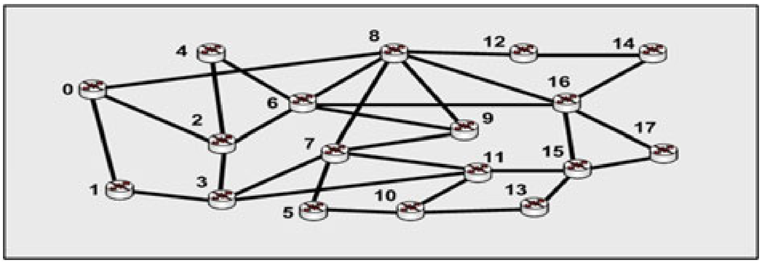

Figure 4. Example topology.

DOI: 10.1002/nem

routes of different sizes between these pairs of nodes. The second index considers critical paths, and whether there are alternative paths of different sizes to reroute around those paths. Figure 5 shows the values obtained for the indices given the example topology.

The graph on the left in Figure 5 shows the *NodeReroutingIndex*, varying the path sizes from 1 to 17. It is possible to see that from 1 to 6 the index increases as the path size increases, meaning that it becomes easier to provide an alternative path when required. Then, from 6 to 14 the index remains nearly constant at 100%, meaning that within this interval it is always possible to reroute when required. For path sizes greater than 14, the index decreases again as we approach the network diameter.

The four curves on the right-hand graph of Figure 5 show the *VCReroutingIndex* computed for critical paths with sizes varying from 2 to 5. For each curve the alternative path sizes varied again from 1 to 17. It is possible to see that for all critical paths we obtained results that are close to each other. The feasibility of finding an alternative path when the critical path size grows from 2 to 5 increases steeply. Then it remains constant at 100% for alternative path sizes varying from 4 to 14. As the network diameter is approached the index decreases.

One can conclude that the search radius used to find alternative paths can be set to the smallest value for which the indices stop growing. In other words, it is useless to use larger search areas, because the ability to find alternative paths remains the same.

4.4. The impact of traffic on rerouting strategy

A simulator was implemented with NS-2 [21] for testing the rerouting strategy. The main metric of interest is the rerouting delay, i.e. the time interval the system takes between a rerouting action starts and completes. We obtained this time by computing the latency of three phases of the rerouting process: *agent activation*, *discovery of alternative routes*, and *reroute configuration*. Furthermore, we measured the *reroute fault rate*, i.e. the percentage of reroute requests that could not be executed due to network traffic conditions. The experiments were run on the topology shown in Figure 4. Each link was configured with a bandwidth of 10 Mbps and a delay of 2 ms. Two types of traffic were configured: constant bit rate (CBR) and Web traffic; 80% of the total traffic corresponds to CBR, while 20% corresponds to Web traffic. We present results considering that the network traffic consumes from 80% to 100% of the available bandwidth. This latency was measured for a representative virtual circuit, consisting of nodes 12-14-16-6-8-7-8-10-13, with a critical path consisting of nodes 6-8-7, and the alternative paths 6-9-7, 6-2-3-7, and 6-4-2-3-7. Results are shown in Figure 6.

The latency of the *agent activation* phase consists of the time interval from the instant the *InputNodeAgent* is activated at the input device to the time instant the *AlternativeRouteAgent* is activated at the first node of the critical path. The graph at the top left of Figure 6 shows that the agent activation delay does not depend on the length of the critical path, but does depend on the amount of traffic on the affected links. The latency of the *discovery of alternative routes* phase corresponds to the time interval from the discovery of the first alternative route, until the last *AlternativeRouteAgent* is activated at the last node of this route. The graph at the top right of Figure 6 shows that the discovery of alternative routes depends, as expected, on the size of the critical paths. The algorithm for finding alternative routes depends on the diameter of the area it will search, and not on the traffic situation of the links traversed.

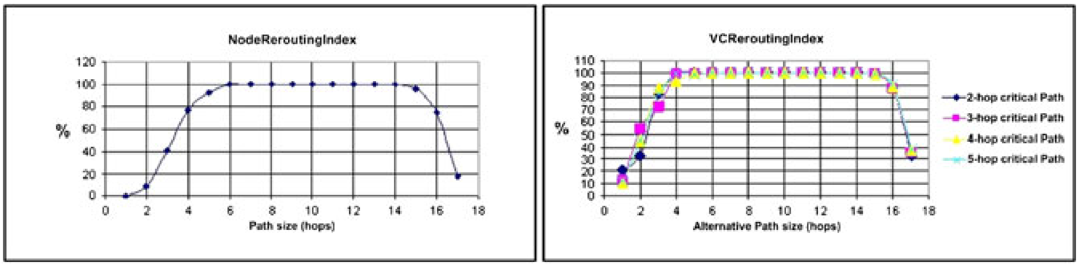

Figure 5. Rerouting indices: example topology.

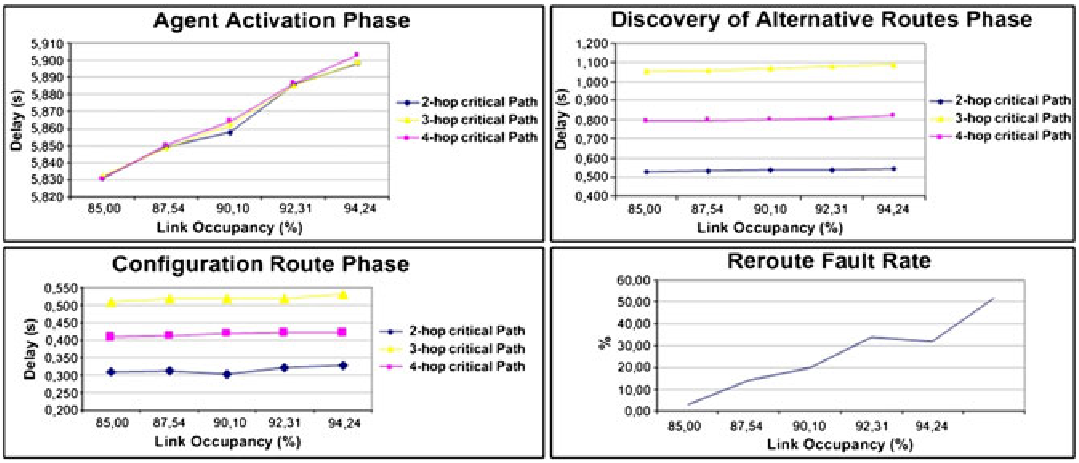

Figure 6. Simulation results with background traffic.

The latency of the *reroute configuration* phase corresponds to the time instant the reroute request is received until a new route is configured, including the selection of the best alternative route to replace the critical path. This latency of the reroute configuration phase (graph at bottom left) depends on the sizes of the critical paths. This is the same as the latency of the discovery phase discussed above, but note that graphs are drawn at different scales and the delay for the configuration of routes is much lower than that of their discovery. Finally, the graph at the bottom right shows that rerouting will succeed unless there is a very high occupancy rate; above 85% of their capacity the fault rate increases.

The main conclusion we can draw from this experiment is that the impact of background traffic on the delays is relatively small. Rerouting delays suffer an increase of less than 10% when that traffic consumes less than 85% of the total bandwidth available. The reroute fault rate only increases when the link utilization rate is very high, causing congestion, and thus implies that several packets are discarded.

5. RELATED WORK

QoS management has been an intensive area of investigation. Aurrecoechea *et al.* [22] surveys the original mechanisms to provide QoS support in IP networks. Flegkas *et al.* [23] present a management architecture for QoS-enabled networks based on policy-based network management. Other management frameworks for QoS-enabled networks include Shankar *et al.* [24] and Wang *et al.* [25], but they are not concerned with multi-domain scenarios. Recently, however, the use of P2P technologies has been considered as a possible and effective solution for the management of highly distributed systems [12]. Wuhib *et al.* [26] use a P2P overlay to detect global crossing thresholds. State and Festor [27] employ a P2P network for the management of wireless networks whose nodes are located in a widely dispersed area. We [28,29] have proposed the use of P2P infrastructure to balance the management overhead among management peers belonging to a single management peer group.

In order to repair degraded services or facilities, restoration techniques are applied [30,31]. A survey of techniques for creating survivable wavelength-division multiplexing (WDM) networks has been presented [10]. The authors start by showing that the automatic protection switching and self-healing ring are the dominant protection techniques used in Synchronous Optical Networks (SONETs) and these techniques may be adapted to WDM networks with some modifications. Nevertheless, besides node and link failures, channel failure is also possible in WDM optical networks, usually caused by the failure of transmitting and/or receiving equipment operating on a given channel. Treating a channel failure as the failure of the entire link may lead to a potential waste of available resources. However, recent algorithms such as Guo [32] still focus on the protection for the single-link failure in optical WDM networks. Funagalli and Valcarenghi [33] consider IP over WDM networks. They claim that both IP and WDM protection schemes should be concurrently employed for different percentages of traffic in order to obtain more effective solutions.

DOI: 10.1002/nem

According to RFC 3496 [34] MPLS-based traffic protection should enable a faster response to faults than is possible with traditional IP-based strategies. Generalized MPLS (GMPLS) [35] extends MPLS [36] to work with optical devices. A protection mechanism using MPLS could enable IP traffic to be placed directly over WDM optical channels providing a recovery option without intervening optical/SONET protection. Proactive MPLS rerouting is called fast reroute [37]. Huang *et al.* [38] propose a reverse notification tree structure for efficient and fast distribution of fault notification messages. Ricciato *et al.* [39] consider path diversity, a key requirement for inter-domain traffic engineering. Three alternative schemes for inter-domain diverse path computation of path diversity in a multi-domain network are compared, in which intra-domain routing information is not disseminated externally.

Rerouting is often classified as either reactive or proactive [10]. Reactive rerouting, as the name implies, is triggered by a failure event; thus traffic is lost until a new routes are defined and configured. Proactive rerouting, on the other hand, employs pre-established recovery paths that employ pre-reserved resources which may never be used if no failures occur but allow faster restoration with little traffic loss when failures do occur. Rerouting can be executed at several layers (optical, MPLS, IP, application), and can be a result of the interaction of techniques deployed in those layers. Rerouting is employed not only to recover from failures but also to solve regular traffic engineering problems, such as congestion. In Puype *et al.* [40] traffic flows are rerouted over a logical IP topology on top of an optical network that does not require manual configuration for provisioning light paths that correspond to the links of the logical IP topology. The proposed solution includes strategies for traffic monitoring and describes a proactive approach that tries to keep the network optimized at all times and thus rerouting whenever it results in better conditions.

In Nelakuditi *et al.* [10] a fast rerouting strategy is proposed specifically for link state routing protocols such as Open Shortest-Path First (OSPF). The proposed strategy is reactive: after link failures link state advertisements are updated, causing routing table recomputations. The strategy is based on local rerouting and prepares for failures using interface-specific forwarding, and triggers local rerouting using a backwarding table. With this approach. When no more than one link failure notification is suppressed, a packet is guaranteed to be forwarded along a loop-free path to its destination if such a path exists. In Liu and Narasimha Reddy [41] another fast rerouting is applied to OSPF; the authors claim that fast rerouting is more appropriate than global routing table update when failures are transient. When a link fails, the affected traffic is rerouted along a pre-computed rerouting path; the strategy allows the local router to signal upstream routers to set up the rerouting path when needed.

Casas *et al.* [42] present an approach involving a combination of reactive and proactive routing based on traffic monitoring. The strategy is based on a new model for the detection of traffic anomalies. After a traffic volume anomaly is detected, the reactive rerouting strategy is triggered. An approach is presented to detect and locate abrupt traffic changes, which was implemented based on SNMP measurements. This is a statistical algorithm which is based on a new model for traffic demand and the measured traffic volume.

6. CONCLUSIONS AND FUTURE WORK

In this paper we presented GigaManP2P: an overlay conceived for solving QoS management in multi-AS optical backbones. The overlay provides a unifying framework that offers management services to three types of clients: network operators, end-users, and applications. Management services act as a bridge between user requirements and the optical infrastructure. The system provides services for QoS monitoring and QoS requirements are enforced by rerouting-aware management peers (also called rerouting agents), operating above the optical infrastructure, that dynamically create and monitor MPLS virtual circuits through which QoS-sensitive flows are routed.

In comparison to traditional proactive rerouting, where back-up routes are reserved from the start, our rerouting strategy is more efficient in terms of network resources consumption because the GigaManP2P management overlay allocates alternative routes only when QoS violation trends are detected. At the same time, our strategy is also more robust than plain reactive routing, in which back-up routes are reserved on demand, but only as a reaction to QoS failures that had already occurred, probably drastically affecting the user applications.

DOI: 10.1002/nem

GigaManP2P was conceived for the multi-AS Brazilian RNP Giga backbone [14], but the proposed architecture and services can be deployed on any multi-AS long-distance backbone. Experimental results showing the overhead of the P2P infrastructure lead to the conclusion that the SNMP APIs employed to contact the final managed optical switches may create a performance bottleneck. The time spent to load SNMP Java classes increases the overall delay of the system, thus requiring a pre-loading workaround that forces Java classes to be loaded before their actual usage. A set of simulations evaluated the impact of network connectivity on finding alternative paths. Results show that it is possible to find such paths for most node pairs using a minimum search radius, which is topology dependent. In our case, since we employed a simulation topology that resembles a national backbone, these results indicate that the actual operational cost to successfully find alternative routes is low. Another final simulation showed that the impact of background traffic over the delays of the management traffic is relatively small, indicating that the rerouting mechanism is sufficiently robust in the daily operations of a national backbone.

Future work will consist of improving the QoS negotiation process by providing a QoS advisor agent that will suggest QoS parameters for interested users given the users' applications and the current conditions of the communications infrastructure. In this study, QoS negotiation delay would be reduced because users would not request network resources that are known to be unavailable in advance. Future work will also include employing connectivity criteria [43] to select alternative routes. Connectivity criteria allow network nodes to be ranked according to their path diversity and robustness considering the network topology, reducing the probability that a node with a higher connectivity number gets disconnected. Another project is to work on a GigaManP2P implementation that could improve the overhead of the system, which we perceive is caused by either the JXTA platform or Java itself.

ACKNOWLEDGMENTS

This work was supported by the Brazilian Research and Education Network (RNP) Giga Project. The work was also partially supported by grants MCT/FINEP PLATCOG 01.10.0549.00, and grants 304013/2009-9, 309270 and 4781174 from the Brazilian Research Agency (CNP$_q$).

REFERENCES

1. Evans J, Filsfils C. *Deploying IP and MPLS QoS for Multiservice Networks: Theory and Practice*. Morgan Kaufmann: Burlington, MA, 2007.
2. Mahajan M, Parashar M. Managing QoS for multimedia applications in the differentiated services environment. *Journal of Network and Systems Management* 2003; **11**(4): 469–498.
3. Callegati F, Cerroni W, Raffaelli C, Zaffoni P. Wavelength and time domain exploitation for QoS management in optical packet switches. *Computer Networks* 2004; **44**(4): 569–582.
4. Chieng D, Marshall A, Parr G. SLA brokering and bandwidth reservation negotiation schemes for QoS-aware Internet. *IEEE eTransactions on Network and Service Management* 2005; **2**(1): 39–49.
5. Pinart C, Junyent G. The INIM system: in-service non-intrusive monitoring for QoS-enabled transparent WDM. *IEEE Journal of Selected Topics in Quantum Electronics* 2006; **12**: 635–644.
6. Huang Y, Heritage JP, Mukherjee B. Connection provisioning with transmission impairment consideration in optical WDM networks with high-speed channels. *IEEE/OSA Journal of Lightwave Technology* 2005; **23**(2): 982–993.
7. Marzo JL, Calle E, Scoglio C, Anjah T. QoS online routing and MPLS multilevel protection: a survey. *IEEE Communications Magazine* 2003; **41**(10): 126–132.
8. Ramaswami R, Sivarajan K. *Optical Networks: A Practical Perspective* (2nd edn). Morgan Kaufmann: Burlington, MA, 2001.
9. Strand J, Chiu A. Impairments and other constraints on optical layer routing. *IETF RFC 4054*, May 2005.
10. Nelakuditi S, Lee S, Yu Y, Zhang Z-L, Chuah C-N. Fast local rerouting for handling transient link failures. *IEEE/ACM Transactions on Networking* 2007; **15**(2): 359–372.
11. Lee S, Yu Y, Nelakuditi S, Zhang Z-L, Chuah C-N. Proactive vs. reactive approaches to failure resilient routing. In *23rd Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, March 2004; 176–186.
12. Granville LZ, Rosa DM, Panisson A, Melchiors C, Almeida MJ, Tarouco LR. Managing computer networks using peer-to-peer technologies. *IEEE Communications Magazine* 2005; **43**(10): 62–68.
13. Harrington D, Presuhn R, Wijnen B. An architecture for describing Simple Network Management Protocol (SNMP) management frameworks. *RFC 3411*, STD 62, December 2002.
14. RNP. Brazilian Education and Academic Network. Available: http://www.rnp.br. [October 2008].

DOI: 10.1002/nem

15. Lua EK, Crowcroft J, Pias M, Sharma R, Lim S. A survey and comparison of peer-to-peer overlay network schemes. *IEEE Communications Surveys and Tutorials* 2005; **7**(2): 72–93.
16. Gong L. JXTA: a network programming environment. *IEEE Internet Computing* 2002; **5**(3): 88–95.
17. Curbera F, Duftler M, Khalaf R, Nagy W, Mukhi N, Weerawarana S. Unraveling the Web Services Web: an introduction to SOAP, WSDL, and UDDI. *IEEE Internet Computing* 2002: **6**(2): 86–93.
18. Strassner J. *Policy Based Network Management Solutions for the Next Generation*. Morgan Kaufman: Burlington, MA, 2003.
19. Extreme Networks. Available: http://www.extremenetworks.com [October 2008].
20. AdventNet Inc. Available: http://www.adventnet.com [October 2008].
21. McCanne S, Floyd S. NS-2 Network Simulator. Available: http://www.isi.edu/nsnam/ns [October 2008].
22. Aurrecoechea C, Campbell AT, Hauw L. A survey of QoS architectures. *Multimedia Systems*, Special Issue on QoS Architecture 1998; **6**(3): 138–151.
23. Flegkas P, Trimintzios P, Pavlou G, Liotta A. Design and implementation of a policy-based resource management architecture. In *Proceedings of IEEE/IFIP Integrated Management Symposium (IM 2003)*, March 2003; 215–229.
24. Shankar M, Miguel M, Liu JWS. An end-to-end QoS management architecture. In *Proceedings of the Fifth IEEE Real-Time Technology and Applications Symposium*, 1999.
25. Wang G, Chen A, Wang C, Fung C, Uczekaj S. Integrated quality of service (QoS) management in service-oriented enterprise architectures. In *IEEE International Conference on Enterprise Distributed Object Computing Conference (EDOC)*. 2004; 21–32.
26. Wuhib F, Clemm A, Dam M, Stadler R. Decentralized computation of threshold crossing alerts. In *16th IFIP/IEEE Distributed Systems Operations and Management (DSOM '05)*, October 2005; 220–232.
27. State R, Festor O. A Management platform over a peer-to-peer service infrastructure. In *Proceedings of the IEEE International Conference on Telecommunications (ICT 2003)*, Tahiti, French Polynesia; 124–130.
28. Granville LZ, Pirmez L, Duarte EP Jr, Souza JN, Andrade RC, Tarouco LR, Correia RB, Lages A. GigaManP2P: a peer-to-peer infrastructure for managing optical networks. In *12th IEEE International Conference on Telecommunications (ICT)*, May 2005.
29. Panisson A, Rosa DM, Melchiors C, Granville LZ, Almeida MJB, Tarouco LMR. Designing the architecture of P2P-based network management systems. In *Proceedings of the 2006 IEEE Symposium on Computers and Communications (ISCC 2006)*, June 2006; 69–75.
30. ITU-T Recommendation G.841. *Types and Characteristics of SDH Network Protection Architectures*, ITU-T, October 1998.
31. Medhi D. A perspective on network restoration. In *Handbook of Optimization in Telecommunications*. Kluwer: Dordrecht, 2005.
32. Guo L. A new and improved algorithm for dynamic survivable routing in optical WDM networks. *IEEE Computer Communications* 2007; **30**(6): 1419–1423.
33. Funagalli A, Valcarenghi L. IP Restoration vs. WDM protection: is there an optimal choice? *IEEE Network* 2000; **14**(6): 34–41.
34. Hellstrand F, Sharma V. Framework for MPLS-based recovery. *RFC 3469*, August 2004.
35. Mannie E. Generalized Multi-Protocol Label Switching (GMPLS) architecture. *IETF RFC 3945*, October 2004.
36. Rosen E, Viswanathan A, Callon R. Multi-Protocol Label Switching. *RFC 3031*, January 2001.
37. Raj A, Ibe OC. A survey of IP and Multiprotocol Label Switching fast reroute schemes. *Computer Networks* 2007; **51**(8): 1882–1907.
38. Huang C, Sharma V, Owens K, Makam S. Building reliable MPLS networks using a path protection mechanism. *IEEE Communications Magazine* 2002; **40**(3): 156–162.
39. Ricciato F, Monaco U, Ali D. Distributed schemes for diverse path computation in multidomain MPLS networks. *IEEE Communications Magazine* 2005; **43**(6): 138–146.
40. Puype B, Yan Q, Colle D, Maesschalck S, Lievens I, Pickavet M, Demeester P. Multi-layer traffic engineering in data-centric optical networks. In *COST266/IST OPTIMIST Workshop (ONDM 2003)*, Budapest, 2003; 211–226.
41. Liu Y, Narasimha Reddy AL. A fast rerouting scheme for OSPF/IS-IS networks. In *Proceedings of ICCCN*, 2004.
42. Casas P, Fillatre L, Vaton S, Nikiforov T. Reactive robust routing: anomaly localization and routing reconfiguration for dynamic networks. *Journal of Network and Systems Management* 2011; **19**(1): 58–83.
43. Duarte EP Jr, Santini R, Cohen J. Delivering packets during the routing convergence latency interval through highly connected detours. In *5th IEEE/IFIP Dependable Systems and Networks Conference (DSN)*, 2004; 495–504.

AUTHORS' BIOGRAPHIES

Elias Procopio Duarte Junior is an Associate Professor at Federal University of Parana (UFPR), Curitiba, Brazil, where he is the leader of the Computer Networks & Distributed Systems Lab (LarSis). He obtained his Ph.D. in Computer Science from Tokyo Institute of Technology, Japan, 1997, M.Sc. in Telecommunications from the Polytechnical University of Madrid, Spain, 1991, and B.Sc. and M.Sc. in Computer Science from Federal University of Minas Gerais, Brazil, 1987 and 1991, respectively. Research interests include computer networks and distributed systems, their dependability, management, and algorithms. Prof. Elias has has served as chair or member of committees of several conferences and workshops in his fields of interest. He chaired the Special Interest Group on Fault Tolerant Computing of the Brazilian Computing Society (2005-2007); the Graduate Program in Computer Science of UFPR (2006-2008).

DOI: 10.1002/nem

Lisandro Zambenedetti Granville is Associate Professor at the Institute of Informatics of the Federal University of Rio Grande do Sul (UFRGS), Brazil, General Director of the Center for Research and Development of Information and Communication Technologies (CTIC) of the Brazilian federal government, Events and Special Interest Groups Director of Brazilian Computer Society (SBC), and member of the Brazilian Internet Committee (CGI.br). He has served as a TPC member (2003-2011), General Co-Chair (2004), and Steering Committee member (2005-2011) for the Brazilian Symposium on Computer Networks and Distributed Systems (SBRC). He has been serving as a TPC member for NOMS/IM, DSOM/CNSM, LANOMS, and APNOMS. Lisandro has served as TPC co-chair for DSOM 2007 and NOMS 2010, as program vice-chair for CNSM 2010, and as co-chair CSSMA of ICC 2011. Lisandro is future workshop co-chair of ICC 2012 and CNOM representative for GLOBECOM 2012.

Luci Pirmez is an Associate Professor at the Institute of Informatics of the Federal University of Rio de Janeiro (UFRJ), Brazil where she is the leader of the Computer Networks & Distributed Systems Lab (LabNet). She received her M.Sc. and Ph.D degrees, both in Computer Science from the Federal University of Rio de Janeiro, Brazil in 1986 and 1996, respectively. She is a member of research staff of the Computer Center of Federal University of Rio de Janeiro. She has served as a TPC member (2003-2011), and Steering Committee member (2005-2011) for the Brazilian Symposium on Computer Networks and Distributed System (SBRC)(SBC/LARC SBRC). Her research interests include wireless networks, wireless sensor networks, network management and security.

Jose Neuman de Souza is an Associate Professor at the Federal University of Ceara in the Computer Science Department. Prof. Neuman holds a Ph.D. degree from Pierre and Marie Curie University (PARIS VI/MASI Laboratory), 1994. He worked in the European projects PEMMON (ESPRIT programme), ADVANCE (RACE I programme) and ICM (RACE II programme), as a technical and administrative member and his contribution was related to the heterogeneous network management environment with emphasis on the TMN (Telecommunication Management Network) systems. From 1999 to 2005 He was a board member (Directory) of the Computer Networks National Laboratory (LARC). He has been a First Class Invited Professor at : UMR CNRS 8144 PRISM-Universite de Versailles Saint Quentin-en-Yvelines, France (2001); UMR CNRS 7030 LIPN-Universite de Paris 13, France (2005, 2006, 2008, 2009); IMAGINE Lab-University of Ottawa, ON, Canada (2007); IBISC Lab-Universite d'Evry Val d'Essonne, Evry, France (2011). He has been as well a CNRS invited researcher at LABRI laboratory - Bordeaux 1 University, France (2010). Since 1999 He has been the Brazilian representative at the IFIP TC6 (communication systems).

Rossana M. C. Andrade is an Associate Professor at Federal University of Ceará, Fortaleza, Brazil, in the Department of Computer Science. She received her Ph.D from School of Information Technology and Engineering (SITE) of the University of Ottawa, Ottawa, Canada, in May 2001. Her Ph.D. thesis focused on the capture, reuse and validation of software patterns for mobile systems. She received her master´s degree from Federal University of Paraíba (currently called Federal University of Campina Grande), Campina Grande, Brazil, in 1992, and her bachelor´s degree in computer science from State University of Ceará, Fortaleza, Brazil, in 1989. Andrade does research in the areas of computer networks and software engineering. Specifically, she is looking at ubiquitous computing and software reuse in these areas. Alternative solutions to increase systems security as well as the application of formal and semi-formal techniques to specify and validate systems are also her research interests. Besides, she is interested in applying formal and semi-formal techniques to specify and validate systems.

Liane Margarida Rockenbach Tarouco is a full professor at the Center for Innovation and Use of New Technologies in Education of the Federal University of Rio Grande do Sul (UFRGS), Brazil. She holds a M.Sc. degree (1976) from UFRGS and a Ph.D. degree (1990) from the University of São Paulo (USP), both in Computer Science. She chaired the 7th Brazilian Symposium on Computer Networks (SBC/LARC SBRC 1989) and has been serving as a TPC member since 1983. From 1983 to 1992 she was a representative of Brazil at IFIP TC6 (Communication Syste ms). Her areas of interest include distributed network management and expert systems to support fault correlation.

Reinaldo de Barros Correia received his B.Sc. degree in Electronic Engineering and his M.Sc. degree in Computer Science from the Federal University of Rio de Janeiro, Brazil in 1984 and 2003, respectively. Research interests include computer networks and network management.

Alexandre Gomes Lages received his both his B.Sc. and M.Sc. degrees in Computer Science from Federal University of Rio de Janeiro in 2005 and 2007, respectively. Research interests include computer networks and network management.

DOI: 10.1002/nem