GigaManP2P – A Peer-to-Peer Infrastructure for Managing Optical Networks

Lisandro Z. Granville, Luci Pirmez, Elias P. Duarte, Jr., José N. de Souza,
Rossana C. Andrade, Liane R. Tarouco, Reinaldo B. Correia, Alexandre Lages

Abstract—**The management of long distance backbones based on high speed optical networks requires new solutions for challenging tasks. For instance, operators and users located at different administrative domains must communicate with each other in order to configure and monitor agreed quality of service levels. This paper proposes a novel peer-to-peer (P2P) management architecture for optical networks, focused initially on the new RNP Giga backbone. In the proposed architecture, peers provide, in a ubiquitous fashion, management information to modules that interface with both the optical infrastructure and network users. This paper also presents preliminary experiments carried out in order to evaluate the processing overhead imposed by the P2P infrastructure on the SNMP-based management platform.**

Index Terms—**P2P, network management, optical networks.**

I. Introduction

THE proper provisioning of quality of service (QoS) in computer networks, particularly in optical networks, depends, among other factors, on an efficient and accurate network management infrastructure. Traditional management solutions, however, do not provide the proper means for the specific requirements of monitoring and configuring optical network devices. Furthermore, current management solutions do not feature an interface that allows end-user applications to request proper treatment of theirs flows. For instance, applications may require minimum bandwidth or communication security and reliability. A network management solution tailored for optical networks should be easy to update given new requirements, and should be also flexible enough to support the fast and dynamic installation of new management services.

In this paper we present a novel *peer-to-peer (P2P) management system* for optical networks. The proposed solution aims to fill the gap between end-user application requirements and the services exposed by an optical network communication infrastructure. Three types of *clients* can access the management services: network operators, end-users, and end-user applications. In order to access such optical management services, management clients must access *management peers* of the management P2P system.

We present the results of a set of evaluation tests that have been carried out in order to check the feasibility of employing peers to access SNMP (Simple Network Management Protocol) [1] agents. We evaluated the computational load of a peer when it is being used to monitor SNMP variables of optical devices. This computational load is also compared with the computational load required to monitor the same SNMP variables, but using raw command line interface applications instead of management peers.

This paper is organized as follows. Section 2 briefly reviews traditional network management technologies, highlighting aspects of such technologies that are inadequate for the management of optical networks. Section 2 also reviews basic concepts associated to P2P systems. The proposed solution and a P2P management architecture are presented in Section 3. In Section 4 we introduce and describe the management services that our P2P system exposes to its management clients, while Section 5 presents the experimental results obtained from the evaluation of the computational load of management peers. Finally, Section 6 concludes this paper with final remarks and future work.

II. Background

The Simple Network Management Protocol (SNMP) [1], standardized by the Internet Engineering Task Force (IETF) is the *de facto* TCP/IP management protocol widely accepted by both industry and academia. However, some SNMP limitations prevent its usage in all management domains. For example, due to security issues, network operators in almost all cases do not rely on SNMP for configuration management.

Thus, alternative solutions have been proposed in the past years trying to improve or replace SNMP. Mobile agents [2], active networks [3], management by delegation [4] and Web Services [5] are examples of academic alternative solutions.

In international standard bodies, mainly on the IETF, heavy efforts to define and standardize other solutions can be observed as well. The SNMPConf working group [6]

Manuscript received Janyary 21, 2005. This work is supported by the Giga Project of the Brazilian National Research Network (RNP).

L. Z. Granville and L. R. Tarouco are with the Federal University of Rio Grande do Sul, Porto Alegre, Brazil (corresponding author to provide phone: +55.51.3316.6819; e-mails: {granville, liane}@inf.ufrgs.br).

L. Pirmes, R. B. Correia, and A. Lages are with the Federal University of Rio de Janeiro, Rio de Janeiro, Brazil (e-mails: {luci, alexandrelages, reinaldo}@nce.ufrj.br).

E. P. Duarte, Jr. is with the Federal University of Paraná, Curitiba, Brazil (e-mail: elias@inf.ufpr.br).

J. N de Souza and R. C. Andrade are with the Federal University of Ceará, Fortaleza, Brasil (e-mails: {neuman, rossana}@lia.ufc.br).

investigates SNMP adjustments required to provide a proper support for device configuration. Also dealing with device configuration, the NETConf working group [7] has been defining a configuration protocol based on XML (Extensible Markup Language). Finally, Microsoft, Intel, Dell, Sun and AMD have released a new service management specification based on Web Services, named ws-management [8].

It is important to notice that almost all alternative solutions are focused on technical aspects, and very few concerns exist related to the environment where those solutions are supposed to run on. In addition, one can generally state that the alternative management solutions are devoted to the management of devices, instead of being oriented to the management of networks. Thus, this current context makes the management of new advanced optical networks, composed of different administrative domains, a hard task.

Although the management of different administrative domains has been already investigated [9], due to the introduction of high speed networks, management solutions cannot rely on direct human cooperation anymore. Cooperation support must now be provided by the network management system, in a way that enables different administrative domain operators to interact with each other using proper management system facilities. However, the current management solutions are clearly unable to provide such cooperation facilities.

Peer-to-peer (P2P) systems, on the other hand, have a different purpose in comparison with network management systems. P2P systems were developed to provide, at certain level, cooperation facilities, for instance, through file and bandwidth sharing, or instant message exchange. The investigation presented in this paper extensively uses P2P technology for optical network management, enabling the required cooperation among different administrative domains.

III. A P2P Management Architecture for Optical Networks

P2P networks are logical networks deployed on top of physical networks [10]. The nodes of a P2P network, called peers, run at end-user computers and they have logical connections with other peers of the same network. Peers come in and out of P2P networks as time goes by, making the network topology very dynamic. P2P network models can be characterized in several ways, using parameters such as the presence of a central server. This central server usually holds network information, such as the active list of peers and a list of the resources that each peer carries (e.g. file directory, processing capacity, bandwidth, etc.).

The P2P management infrastructure proposed in this paper is a solution devised to bridge the needs of end-user applications with the facilities of optical networks. This solution provides a set QoS management services directly to the applications. The P2P management infrastructure offers services for three types of clients: networks operators, end-users, and applications.

A. Structure of a Management Peer

In the proposed solution, management peers are distributed along the managed network and take care of local client requests. Each management peer can be seen as a server that takes care of the needs of local clients. Each peer provides, beyond the services to its clients, services to the other peers. This allows the interaction among peers distributed along the different administrative domains of the managed optical networks. The architecture of each peer is presented in Figure 1, and includes the following communication interfaces:

- Communication interface among peers via JXTA [11];
- Interface for scripts transference via Script MIB [9];
- Interface for management policies deployment via Web Services [12] and LDAP [13];
- Interface for mobile agent migration via Aglets [14] or µCode [15].

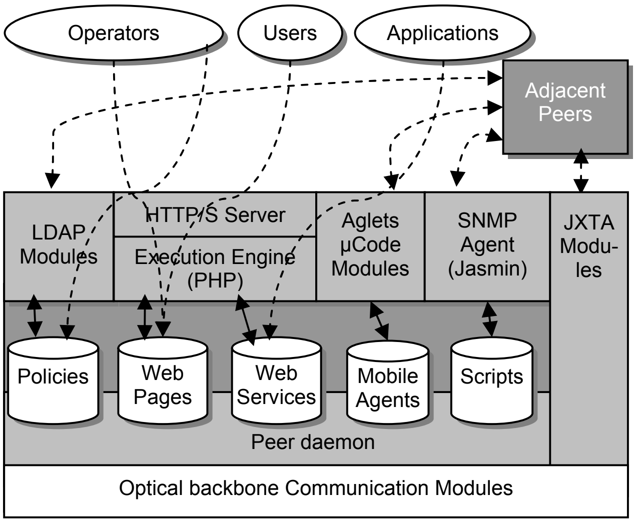

Fig. 1. Management peer structure.

B. Communication between Clients and Peers

Network operators are clients that access their management services through dynamic Web pages created by a Web processing engine, such as PHP. This access allows the operator to create management policies that are stored in the policy repository. The LDAP module is able to send the policies stored in the peer internal repository to other peers.

End-users are clients that also access management services through Web pages. End-users, operators, and applications requests are passed to the peer daemon. The daemon validates the requested service against the client identification. Mobile agents and scripts are provided respectively by Aglets/µCode [15] and Jasmin [16] toolkits.

C. Peers and Optical Infrastructure Communication

The modules for the communication of peers with the optical infrastructure are located at the bottom of the peer structure. They provide an interface to access devices in a transparent way.

The actions executed by the communication modules with the optical infrastructure are controlled by the peer daemon.

This daemon has access to the peer internal management policies, agents, and scripts. In addition to the interfaces presented in Figure 1, other interfaces can be installed through the expansion of each peer via mobile agents.

IV. Management Services

The proposed P2P management solution offers management services to operators, end-users, and applications. These services are presented in the following subsections.

A. Services for Network Operators

Network operators are responsible to run the managed network to fulfill end-users needs. The services for network operators are: handling and use of management policies, management script distribution and execution, distribution of mobile agents, and network connectivity monitoring.

The *handling and use of management policies* allows 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 of the IETF policy based network management framework [17].

The *management script distribution and execution* offers a mechanism to the operator that allows him to request to the P2P system the execution of specific tasks (e.g. MPLS paths bandwidth allocation). Scripts support is accomplished via the IETF Script MIB [18]. In this case, each peer acts as an SNMP agent able to execute management scripts.

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

Finally, the *network connectivity monitoring* is the service that enables 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/S. Thus, the P2P system is similar to FreeNet, a concept introduced by the BitTorrent software.

B. Services for End-users

Offering management services to end-users allow them to have some level of control over the network resources required to accomplish their tasks. The services available to end-users are: requesting differentiated traffic, enabling multicast transmission support, and network status report.

Requesting differentiated traffic lets end-users schedule QoS support for mission critical traffic, i.e. each peer acts as a Bandwidth Broker [19]. Although end-users can request QoS support, that is only accomplished if the user requests respect the management policies defined by the network operator.

Enabling of multicast transmission support allows the end-user to inform his or her needs of multicast treatment by the underlying network. Finally, the *network status report* service is the one that asynchronously notifies end-users (e.g. via e-mails) regarding the changes in the optical infrastructure.

C. Services for End-user Applications

The same services available to end-users are available to applications. However, the access to such services is different. Instead of using Web pages, applications access the services via Web Services technology, which is based on SOAP [12].

Management services offered to applications is especially important for cooperative and resource sharing applications such as grids, where the network infrastructure needs to be configured to provide proper communication facilities.

D. Services between Peers

The P2P management infrastructure monitors the optical network performance in order to inform when and why the expected QoS is not being properly provided. Depending on the provided service, the management peer may proceed with pro-active management where QoS maintenance actions are executed prior to a drastic QoS fall. Such actions can be triggered after a peer analysis of management data exchanged with other peers. This analysis may include fast simulations, extrapolations, or artificial intelligence techniques.

The proposed management infrastructure must be highly-available, for management is most important when the network has partially failed.

Peers communicate using direct virtual links that are mapped to IP and physical routes. In order to allow peers to keep communicating even when the regular routes among them are disrupted, we propose a fault-tolerant routing service. This service is based on the use of detours, which are alternative routes selected based on a set of connectivity criteria [20]. Detours have a higher change of being fault-free even when the network has partially failed. Re-routing can also be employed when QoS guarantees are being violated by the route currently in use. This service is transparent to the users.

V. Analysis of the Peer Computational Load

Peers of the proposed management infrastructure were implemented using Sun's JXTA framework, which is open source software. JXTA is a framework for P2P networks development that uses a set of maintenance services already defined and implemented by the network itself. This way, the creation of a P2P network is simplified because the basic services of interconnection, communication and searching between peers are already available. On top of these services, other more specialized services can then be developed, such as pro-active management and rerouting, presented previously. Concerning the use of mobile agents, both Aglets [14] and μCode [15] were employed, which are also open source toolkits.

In this section, experimental results are presented, which were obtained from the evaluation of the overhead of the P2P infrastructure on the SNMP-based management platform. In other words, the computational load of a management peer is examined when it is employed to access optical device variables on top of the SNMP agents which are still used for querying/managing the device. Such measure is then compared with a second measure, which was obtained through a direct access to the SNMP agents of the optical device, i.e. without the use of P2P management infrastructure.

A. Definition of Tests

The objective of the test is to quantify the latency introduced by the extra processing layers to access the optical device's information. For accomplishing these tests, two machines linked by means of a Fast-Ethernet network had been used. The first machine is based on an ATHLON XP1700 processor installed in a SOYO/512 MB motherboard and it is responsible for the execution of P2P management architecture. The second machine is a Pentium II 350 MHz processor on an ASUS motherboard with 256MB of memory. This machine mimics the switch 6808 (a device which will be managed in the future) and it runs a Linux distribution of the SNMP agent version 5.1.1-2. The operating system in both machines is the Linux distribution Fedora 2. The tests were run in two phases: (i) testing the communication module and (ii) measuring the time required for launching rerouting agents. The experience was run 30 times. Results are presented below.

B. Phase 1: Test of the Communication Module

The test of the communication module consists on measuring the latency as perceived by the client when invoking management methods. For such, we measured the time elapsed from the instant just before the method at layer 2 was called and the instant just before the method at layer 1 is called. The methods implement common management functions such as MIB object recovery and MIB object update. The processing time was measured considering two implementation approaches: (i) in the first measurement we employed SnmpAPI object classes supplied by AdventNet (ii) in the second measurement we executed commands provided by the Runtime class of the API Java interface. Table 1 summarizes the processing average times obtained with the two approaches.

	API AdventNet				Classe Runtime			
	Get	GNext	Bulk	Set	Get	GNext	Bulk	Set
Avg.	1390.27	1390.23	1397.33	1392.87	98.5	97.8	102.17	99.67
Dev.	1.337	0.568	6.233	0.346	8.83	6.57	7.65	5.16
(95%)	0.015	0.006	0.071	0.004	0.101	0.075	0.087	0.059

Table 1. Processing latency (ms).

The values in Table 1 represent the time intervals required for communicating in the test network, that is, as cited previously, on a Fast-Ethernet (100 Mbps) with low traffic. The time intervals measured for the AdventNet SnmpAPI are relatively high due to the overload of the additional processing introduced by this API. For sending a SNMP message to the agent, a set of API classes are instantiated for generating objects which will execute the SNMP requests. This overload corresponds to 92.91% of the total average time (Get method) presented in Table 1, e.g., the effective latency of processing a Java-based SNMP request corresponds, in average, to 98.5 ms, including the processing times and the RTT of the Fast-Ethernet network which is 55.0ms. However, the higher overload obtained using 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 execution of SNMP requests in this situation. A set of 30 requests in one session was executed, with some of the objects of the SnmpAPI already loaded. An average time of 1027.10 ms was obtained for the Get requests, standard deviation of 13.501 ms and a reliability interval (95.0%) of 0.0941 ms.

Therefore, in order to improve the computational performance of the P2P infrastructure, it became evident from the test results that the SNMP infrastructure must be modified. One possibility is the inclusion of a new component, a utility agent, which would visit all the managed devices during the system start up, pre-loading the required SnmpAPI objects. Utility agents would then remain active while the system is running. The inclusion of this new component would allow a reduction of 26.12% in the time spent by the agents for sending SNMP requests.

As it can be observed in Table 1, the implementation with the Java Runtime class resulted in a considerable latency reduction, due to the elimination of the overhead of loading SnmpAPI objects classes.

Additional code development is necessary for measuring the time required for executing the Java-based methods. This code presents an extra overhead on both agent size and increases the communication time. To overcome this problem, utility agents can be again used, which are loaded at start up, enabling also rerouting agents (discussed below) to obtain the information they require without having to incorporate themselves this extra code.

In other words, with the use of the Java Runtime class, the size of the code of rerouting agents becomes bigger, which has a negative impact on their migration latency. Furthermore, portability suffers, because these agents cannot be employed on other platforms. In this aspect, the abstraction introduced by the alternative solution has an advantage because portability is guaranteed, but although the latency is higher.

C. Test of the Rerouting Agents

This test has the primary objective of evaluating the load time of the rerouting agent. Time was measured from the instant where an instance of the rerouting agent is generated until the moment the agent is ready to start its execution at the destination station. During this time interval, the agent suffers the serialization process, migrating to the remote machine (in this test the migration time was not considered, because only one host was employed), and finally, the agent is deserialized. Table 2 presents the average time obtained, as well as the standard deviation and reliability interval.

	Time
Avg.	494.17
Dev.	59.802
(95%)	0.6847

Table 2. Rerougint agent instancing delay.

The latency to launch an agent is proportional to its size, i.e. the size of its code. Thus, these agents must be as simple, and run algorithms as optimized as possible. Routing must be as fast as possible, otherwise all applications may be compromised. In this context, using the previously mentioned utility agents becomes still more attractive. The central idea is to transfer as many rerouting agents functionalities as possible to the utility agents.

VI. Conclusions and Future Work

In this paper we have introduced a novel P2P-based management architecture for cooperative management of optical networks. Cooperative management is required for managing high-speed optical backbones since traditional management solutions, based on SNMP, are not able of providing this functionality. On the other hand, P2P networks were developed with the objective to allow cooperation among users. Thus, it is natural to use P2P technologies for this kind of network management.

The solution proposed in this work has the objective of negotiating the requirements of three different types of clients: network operators, end-users and applications. The proposed architecture offers management services that bridge the communication of the users and the optical infrastructure. Peers communicate even if they are deployed at different administrative domains.

As detailed in this paper, technologies such as JXTA, Script MIB, Web Services, LDAP, Aglets and µCode are employed to build a peer. Besides presenting the general architecture and the peer structure, this paper also presents an evaluation of the impact of the proposed solution in comparison to traditional management systems. The main objective of the tests executed was to check the overhead of management peers on traditional SNMP agents. A prototype was implemented which allowed preliminary tests to be executed.

As expected, the performance of peers is inferior to direct accessing management information, taking into account that the number of peer software layers is higher. On the other hand, more management services are supplied by peers who are also more sophisticated. Future work includes completing the full implementation of the proposed architecture and executing additional tests, including checking the time for deploying management policies through peers and the delay to install new dynamic services using mobile agents. Finally, an analysis of the QoS guarantees is also necessary.

Acknowledgment

The authors thank the RNP (National Network of Education and Research), CPqD (National Center of Research) and FINEP (Financier of Studies and Projects) for supporting the development of the related work in this paper.

References

[1] J. Case, R. Mundy, D. Partain, and B. Stewart, "Introduction and Applicability Statements for Internet Standard Management Framework", RFC 3410, IETF, 2002.

[2] S. Berkovits, J. D. Guttman, and V. Swarup, "Authentication for Mobile Agents". In:Vigna G. Mobile Agents and Security. 1. Berlin, Heidelberg. Springer-Verlag, 1998.

[3] K. Psounis, K. "Active networks: Applications, security, safety, and architectures". IEEE Communications Surveys, pages 1–16, March 1999.

[4] G. Goldszmidt, and Y. Yemini, "Distributed Management by Delegation". 15th International Conference on Distributed Computing Systems. IEEE Communication Society, 1995.

[5] R. Neisse, R. Vianna, L. Z. Granville, M. J. Almeida, and L. Tarouco, "Implementation and Bandwidth Consumption Evaluation of SNMP to Web Services Gateways", In: 9th IFIP/IEEE Network Operations and Management Symposium (NOMS 2004), 2004.

[6] M. MacFaden, D. Partain, J. Saperia, and W. Tackabury, "Configuring Networks and Devices with Simple Network Management Protocol (SNMP)", RFC 3512, IETF, 2003.

[7] R. Enns, "NETCONF Configuration Protocol". IETF Internet draft <draft-ietf-netconf-prot-04> (work in progress), 2004.

[8] A. Geller, et al. "Web Services for Management (WS-Management)". Available at http://www.intel.com/technology/manage/downloads/ws_management.pdf, 2004.

[9] D. Levi, and J. Schoenwaelder, "Definitions of Managed Objects for the Delegation of Management Scripts", RFC 2592, IETF, 1999.

[10] A. Oram, "Peer-to-Peer for Academia". Published on the O'Reilly Peer-to-Peer & Web Services Conference, 2001.

[11] L. Gong, "JXTA: A Network Programming Environment", In: IEEE Internet Computing, Vol. 5, Issue 3, p. 88-95, 2001.

[12] F. Curbera, M. Duftler, R. Khalaf, W. Nagy, N. Mukhi, and S. Weerawarana, "Unraveling the Web Services Web: an Introduction to SOAP, WSDL, and UDDI", In: IEEE Internet Computing, Vol. 6, Issue 2, p. 86-93, 2002.

[13] M. Wahl, et al. "Lightweight Directory Access Protocol (v3)", RFC 2251, 1997.

[14] Oshima, Mitsuru & LANGE, Danny B. "Mobile Agents with Java: The Aglet API". Available at: http://aglets.trl.ibm.co.jp/information.html.

[15] G. Picco, "µCode: A Lightweight and Flexible Mobile Code Toolkit". In Kurt Rothermel and Fritz Hohl, editors, Proceedings of the 2nd International Workshop on Mobile Agents, LNCS, pages 160–171, Springer-Verlag, 1998.

[16] J. Quittek, J. Schoenwaelder, F. Straus, "Jasmin - A Script MIB Implementation". Available at: http://www.ibr.cs.tu-bs.de/projects/jasmin.

[17] J. Strassner, "Policy Based Network Management Solutions for the Next Generation", Morgan Kaufman Publishing, ISBN1-55860-859-1, 2003.

[18] J. Souza, A. Rocha, C. Rocha, "Script MIB Extension for Resource Limitation in SNMP Distributed Management Environments", Fortaleza-Ceará, Brazil, ICT 2004, Springer-Verlag, LNCS-3124 pp. 835-840.

[19] S. Blake, et. al. "An Architecture for Differentiated Services". RFC 2475, IETF, 1998.

[20] E. Duarte, R. Santini, J. Cohen, "Delivering Packets During the Routing Convergence Latency Interval through Highly Connected Detours," Proceedings of the IEEE/IFIP International Conference on Dependable Systems and Networks (DSN'2004), Dependable Computing and Communications (DCC) Symposium," pp. 495-504, Florence, Italy, 2004.