

Provisioning and Management of Interdomain Connections in Optical Networks: A Service Oriented Architecture-based Approach

Fábio L. Verdi, R. Duarte, F. C. de Lacerda,
Eleri Cardozo, Maurício Magalhães

Department of Computer Engineering and Industrial Automation (DCA)
School of Electrical and Computer Engineering (FEEC)
State University of Campinas (Unicamp)
13083-970, Campinas-SP, Brazil.

Edmundo Madeira

Institute of Computing (IC)
State University of Campinas (Unicamp)
13084-971, Campinas-SP, Brazil.

Abstract—Provisioning of interdomain connections has been a constant challenge for providers. Clients (e.g. access networks) are asking for services that span across multiple different domains and each domain has its local constraints (policies) that must be analyzed before accepting an end-to-end (e2e) connection. In this paper we present a Service Oriented Architecture-based approach by which basic services are modeled and composed to create more complex services. Such approach considers the establishment of e2e interdomain connections in optical networks. The architecture we present here consists of a virtual topology service, an end-to-end connection service and an end-to-end negotiation service. All of them were implemented using the Web services technology.

I. INTRODUCTION

While the provisioning of connections inside a domain is a well known and widely studied problem, interdomain end-to-end (e2e) connections always raise many discussions. The main problem when considering interdomain connections is related to how traffic engineering (TE) is performed and how local domain constraints are respected. Establishing an e2e interdomain connection is not a trivial task. Clients that need an e2e connection crossing multiple domains should have a mechanism to choose the interdomain route taking into account TE parameters such as bandwidth and delay. These parameters can be used to calculate the cost of each edge connecting the nodes among all the domains. Currently, interdomain routing protocols (e.g. Border Gateway Protocol-BGP [7]) do not carry any kind of TE information. Although BGP has some policies by which domains can define simple rules to specific traffic flows, the interdomain routing considers almost nothing about real e2e Quality of Service (QoS) metrics (e.g. bandwidth and delay). Furthermore, BGP was initially developed to exchange reachability information. Adding QoS metrics to it can affect the scalability of the Internet routing [4].

In this paper we propose an approach to support the establishment and management of e2e interdomain optical network connections by using Web services. The main objective of Web services [11] is to allow applications to com-

municate with other applications outside corporate Intranets in an automatic fashion and without the human intervention. This is called application-to-application communication. To achieve this, Web services make use of the Service-Oriented Architecture (SOA) [6]. This architecture is the next step of distributed computing, which enables software components including application functions, objects and processes from different systems to be exposed as services. This concept has been widely analyzed by organizations and consortiums such as OASIS [5] and W3C [11] in order to define specifications that drive the adoption of Web services standards.

Currently, the intra-carrier (intra-provider) E-NNI [2] is being specified by the OIF but there is nothing related to the inter-carrier E-NNI signaling. While this ongoing work tries to reach a consensus about how e2e interdomain signaling could be done, we believe that the use of Web services and XML can act as a facilitator to define how different administrative domains can interact to each other. The idea of having services that operate according to the business and goals of the network provider is also easily addressed by the Web services technology.

The paper is organized as follows. Next section details the architecture, its modules and the interactions between the modules to provide an e2e interdomain connection. Section III shortly presents the aspects related to the implementation of the architecture. Section IV concludes the paper and lists some further works.

II. DESCRIBING THE SOA-BASED MANAGEMENT ARCHITECTURE

The concepts of management, control and transport planes are defined in the context of the Automatically Switched Optical Network (ASON) [13] and are the same for the GMPLS and ASON architectures. In our architecture (see Fig. 1), the management plane is divided into two logical parts. The first one groups the Web services that are responsible for offering to external entities (clients and other domains) an interface to access the functionalities of the management

system. These Web services realize the SOA architecture by abstracting internal details on how each task is implemented. The optical domain is seen as a set of Web services, each one responsible for providing specific functionalities. The services can be as simple as the Virtual Topology Service (VTS) or they can be created by composing simple services so that more complex services are formed (e.g. O-VPN Service).

The second logical part of the management plane is related to the internal modules that are necessary to manage the control and admission of requisitions. In the following, the modules that belong to the architecture are described. First of all, we detail the Web services modules. Then, the internal modules are explained.

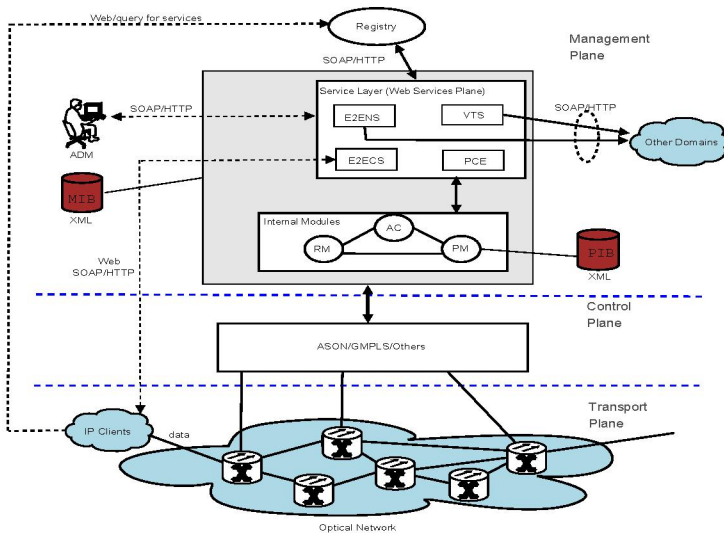


Fig. 1. The Proposed Architecture.

A. External Modules - Web Services Plane

The Web services Plane is a double-faced entity because it provides an interface to access the system and locally it is accessed by the internal modules. The services are exposed offering an interface that is invoked by other applications (e.g. web browser and other services). The domain administrator models the service, defines its behavior and publishes the interface in a Registry (see below) so that the service can be looked up and accessed by external entities. Below we describe in more details the Web services that currently belong to the Web services Plane.

1) *Virtual Topology Service - VTS*: The VTS is responsible for advertising the virtual topology to other domains. The Resource Manager (RM, see below) reads the local domain virtual topology and asks for the VTS to advertise it to other domains. The virtual topology to be announced depends on local constraints (policies) defined by the domain administrator. A given domain may have many different virtual topologies that are advertised following specific rules such as the variation of the amount of traffic during the day, services being offered (VoIP, video conference, O-VPNs being established and so forth) and the availability of the resources (failures, bottlenecks, etc.). Then, the domain administrator

can define a “standard” virtual topology to be used in normal conditions and other virtual topologies that are used when specific conditions are detected. By using this approach, the domain administrator is able to define rules taking into account business strategies. Furthermore, clients are able to contract the VTS by defining rules on when the virtual topology should be advertised and what level of information is to be advertised with the virtual topology.

2) *End-to-End Negotiation Service - E2ENS*: This service is responsible for negotiating the e2e connection with other domains. We adopted the star model by which the head-end domain¹ negotiates with other domains asking for an available resource (lightpath). The E2ENS implements a two-phase protocol. The first phase queries and reserves the resource and the second phase confirms the reservation considering that all the domains involved in the negotiation have the required resource. The reservation follows the first come first served rule. If one of the domains does not have the resource, the reservation needs to be released through the rollback operation. During the first phase the traffic parameters are analyzed in each domain in order to verify if the connection can be accepted. Considering that we are interested in grooming IP/MPLS traffic within lightpaths [1], the E2ENS service needs to exchange at the least the bandwidth, level of protection and the type of traffic. Based on these parameters, each optical domain can choose the best resource to aggregate the incoming flow;

3) *End-to-End Connection Service - E2ECS*: This service is invoked by clients to ask for the establishment and removal of an intra/interdomain connection. Clients do not need to know how the domain will provide the e2e connection. They only have to say what kind of connection they want. This service is considered as a compound service since it needs to use other services to perform its task. First of all it needs to find an e2e interdomain route (invoking the PCE service). Note that to find the route, the PCE service depends on the VTS. It is through the virtual topology of each domain that the interdomain route is calculated. Finally, the E2ECS invokes the E2ENS to negotiate with every other domain. After performing these phases and considering that the e2e connection is established across the domains, the client is able to start sending data through the connection;

4) *PCE Service*: The PCE Service is locally invoked by the modules inside the domain. It is responsible for finding an interdomain route so that an e2e interdomain connection can be established. Since the virtual topology of each domain was advertised by the VTS, finding an interdomain route is just a matter of applying a routing algorithm. Although the PCE Service is being currently invoked only by local modules, it can represent a given region or area. Other domains that do not have such service can buy/rent the PCE service from a third-party provider. This is a typical SOA scenario where services are offered, queried and used.

Finally, to have a complete SOA scenario it is necessary to provide a mechanism whereby services are published and

¹The head-end domain is the domain where the requisition was made.

looked up. The Registry is a module where the end point of each service is stored. It is not an Universal Description, Discovery and Integration (UDDI) repository [8] since an UDDI repository has much more functionalities than our architecture needs. Many current works have used the same simpler approach [3]. The Registry acts as a private directory where the service interfaces are registered and a mechanism to query such services is provided.

B. Internal Modules

1) *Admission Control - AC*: It represents the internal interface whereby requisitions coming from the Web Services Plane are forwarded to have access to the management system. It is responsible for receiving connection requests, verifying predefined Service Level Agreements (SLAs) and invoking the Policy Manager and Resource Manager to perform specific management tasks;

2) *Policy Manager - PM*: PM is also known as the Policy Decision Point (PDP) and is responsible for applying the policies defined to the domain being managed. The Policy Manager implements the grooming policies [1], [9], [10] by analyzing a pool of candidate lightpaths (received from the AC) and trying to find one with available resources to accommodate a given IP/MPLS requisition;

3) *Resource Manager - RM*: It is responsible for storing the information model needed to provide and manage e2e interdomain connections. It receives the virtual topology from the VTS (from other domains) and stores it in a local repository. Also, it reads the virtual topology of the local domain and asks for the VTS to advertise it to other domains.

The negotiation between domains is done in a parallel way. The head-end E2ENS invokes the E2ENS of the domains belonging to the e2e interdomain route. Local constraints in each domain are applied in order to accept or not the connection. The E2ENS of each domain returns back confirming or denying the requisition. Currently, the negotiation protocol does not support counter-proposals done by the downstream domains. If the invoked domain does not have the resource required during the reservation phase, the e2e connection will not be accepted and the rollback operation is called to release the resources that were reserved in other domains.

III. IMPLEMENTATION

A. Implementation

The Web services are created using the Apache AXIS 1.2 [12]. The internal modules (AC, RM and PM) are remote objects developed using the Java RMI technology. The communication between Web services is done through XML-based Simple Object Access Protocol (SOAP) messages over HTTP.

The SOA-based computing in our architecture is performed as follows. First of all, a given client looks for the E2ECS in the Registry. After obtaining the E2ECS Web service WSDL, it is possible to create a client application with the correct methods and parameters necessary to invoke the E2ECS service. The information that is stored in the Registry (using XML files) is composed of the name of the service, a

short description of the service, the endpoint (location) of the service and the WSDL location. The WSDL document is not physically copied into the Registry. It must exist somewhere on the Web where it is accessible via an URL, and this URL is what is stored in the Registry. The Registry also offers a mechanism for clients to query services. Such mechanism allows clients to send a description specifying what services they are looking for. The end-point of the WSDL of the service (in case of finding one) is then returned to clients.

E2ECS receives requests from outside the domain and forwards them to the AC module for internal execution. The E2ENS receives requests from the local AC and maps such requests in SOAP messages to invoke E2ENSs of other domains. Remote E2ENSs receive the requests and forwards them to the AC module.

Fig. 2 shows the sequence diagram to establish an e2e interdomain connection. It represents the successful case, i.e., the scenario in which all the domains have the required resources. The boxes in the diagram depict the group of steps that should be executed in each domain to reserve lightpaths.

Some preliminary results show that the time to establish an interdomain connection in a scenario with 5 domains varies from ≈ 80 ms to ≈ 105 ms, what in our point of view is a very feasible time. The quantity of bytes to communicate all the SOAP messages that are necessary to establish a single e2e connection is approximately 8Kbytes (7893 bytes), and the time to exchange this quantity of data is approximately 52,85 ms. This time belongs mostly to the Web services engine that is responsible for parsing the XML-based message SOAP, creating the HTTP message, sending it to the remote Web service and making the reverse operation.

IV. CONCLUSION

In this paper we presented a SOA-based architecture for providing e2e interdomain connections. Our solution adopts the idea of having a virtual topology over the optical physical topology. Such virtual topology represents only the ingress and egress FA connections inside the domain preserving the confidentiality of each domain. The e2e interdomain negotiation service offers a mechanism by which domains can negotiate an e2e interdomain connection. The PCE and the VTS work together to offer a mechanism to advertise and find a route to attend customers requirements. Moreover, the idea of composing simple services to form more sophisticated ones seems to be effective in the optical network management context.

The architecture proposed in this paper can be used not only in optical scenarios but also in any scenario where the physical topology is mapped to a virtual topology. The virtual topology idea abstracts the details on how such mapping is done and the details about the selection of the resources are left to each local domain.

The implementation of the architecture has showed that the modules we have been defining are feasible and the integration of different technologies are very useful in the context of optical network management. Currently we are defining an

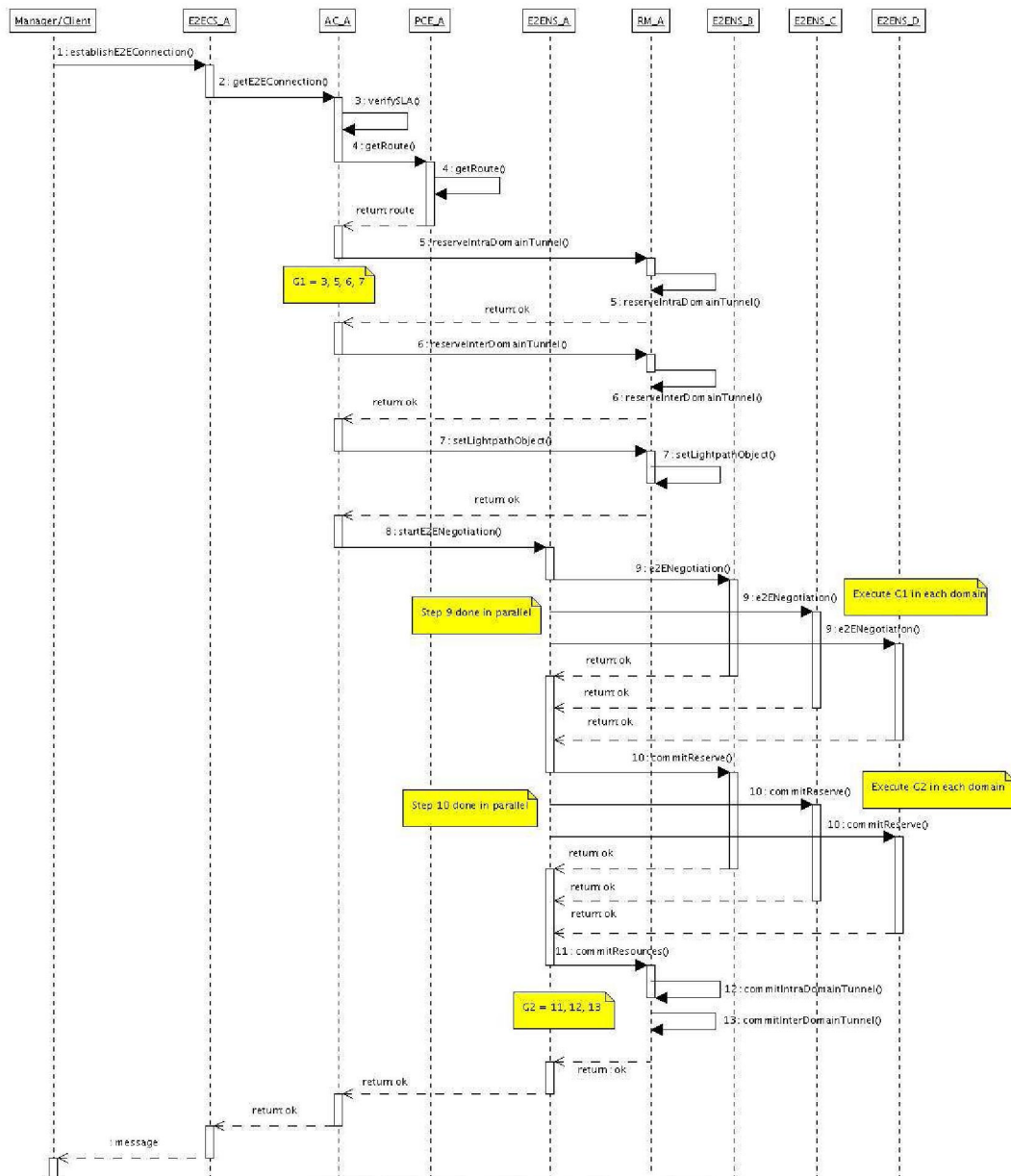


Fig. 2. Provisioning of a successful e2e interdomain connection.

O-VPN service which, to the best of our knowledge, is one of the main services to be offered by ISPs in optical networks. The Web services technology appears again as a facilitator to provide this service.

REFERENCES

- [1] C. Carvalho, F. L. Verdi, E. Madeira, and M. Magalhães. Policy-based Fault Management for Integrating IP over Optical Networks. *The 5th IEEE International Workshop on IP Operations & Management (IPOM'05)*, LNCS-Springer-Verlag, 3751:88–97, October 2005.
- [2] OIF Intra-Carrier E-NNI 01.0 Signaling Specification.
- [3] D. L. Truong et al. A Policy-based approach for user controlled Lightpath Provisioning. *IFIP/IEEE NOMS*, pages 859–872, April 2004.
- [4] L. Xiao et al. Advertising Interdomain Qos Routing. *IEEE Journal on Selected Areas in Communications*, Vol. 22, No. 10, pages 1949–1964, December 2004.
- [5] OASIS Consortium: <http://www.oasis-open.org/>.
- [6] M.P. Papazoglou and D. Georgakopoulos. Service Oriented Computing. *Communications of the ACM*, 46 No. 10:25–28, October 2003.
- [7] Y. Rekhter and T. Li. A Border Gateway Protocol 4 (BGP-4). *RFC 1771*, March 1995.
- [8] OASIS UDDI Specification: <http://www.uddi.org/>.
- [9] F. L. Verdi, C. Carvalho, E. Madeira, and M. Magalhães. Policy-based Grooming in Optical Networks. *4th IEEE Latin American Network Operations and Management Symposium (LANOMS 2005)*, pages 125–136, August 2005.
- [10] F. L. Verdi, E. Madeira, and M. Magalhães. Policy-based Admission Control in GMPLS Optical Networks. *First IEEE Broadnets'04 (formerly OptiComm)*, San Jose, USA, pages 337–339, October 2004.
- [11] Web Services Activity: <http://www.w3.org/2002/ws/>.
- [12] <http://ws.apache.org/axis/>, 2005.
- [13] ITU-T: Architecture for the Automatically Switched Optical Network (ASON), G.8080/Y.1304.