PONNI: A routing information exchange protocol for ASON[1]

Sergio Sánchez-López, Xavier Masip-Bruin, Josep Solé-Pareta, Jordi Domingo-Pascual

Advanced Broadband Communications Center (CCABA), Departament d'Arquitectura de Computadors, Universitat Politècnica de Catalunya, Barcelona, Catalunya (Spain)

{sergio, xmasip, pareta, jordid } @ac.upc.es

Abstract

Optical transport networks with automatic switching capabilities (ASON, Automatic Switching Optical Networks) appear as a potential solution to cope with the increasingly growth of Internet traffic demands. This paper focuses in the context of the ASON control plane. In this context, an Optical NNI (O-NNI) has to be defined. For some time it seemed clear that such an O-NNI would be based on Generalised Multiprotocol Label Switching (GMPLS). Nevertheless recently the idea of an O-NNI based on the ATM Private Network to Network Interface (PNNI) paradigm is wining partisans.

In this paper a preliminary attempt to adapt the ATM PNNI protocol to be used as the new O-NNI will be done. In particular, this paper will be addressed to define an ATM PNNI based protocol to cope with the routing information exchange in an ASON. We call such a protocol Private Optical NNI (PONNI). We think that PONNI can be more appropriated for this purpose than the OSPF protocol, the one used in the GMPLS paradigm. The main reason for that is that, while OSPF has to be modified in order to achieve the capabilities needed to distribute both internal and external information on the available resources (e.g. remaining bandwidth, wavelengths, etc.), the PNNI has these capabilities by nature.

1. Introduction

In recent years the introduction of high capacity and reliable transport networks is being necessary in order to cover the needs of Internet traffic demands. New incoming Internet applications increasingly request greater capacity and guarantees of traffic delivery. Optical Transport Networks (OTN) with automatic switching capabilities (ASON, Automatic Switching Optical Networks) appear as a potential solution to cope with such a situation.

ASON has to include a Control Plane able to provide features such as Traffic Engineering. One of the essential components of this Control Plane is the routing algorithm, which has to compute a proper route to the incoming connection demands, for which it has to take into consideration the available network resources. In this way, the routing algorithm requires a Traffic Engineering Database (TED) in order to obtain information on topology and available resources, which must be updated when a change of the topology or the resource utilization is produced. In order that the TED can be updated, the control plane has to provide a flexible, fast and reliable mechanism to disseminate the topology and the available resource information throughout the network. In order to support such a mechanism, an Optical NNI (O-NNI) has to be defined for ASON.

For some time it seemed clear that such an O-NNI would be based on the GMPLS paradigm. Nevertheless recently the idea of an O-NNI based on the ATM PNNI paradigm is gaining partisans. Solutions based on the GMPLS paradigm only consider typical Internet protocols such as link state based

---

[1] This work has been partially funded by the European Commission under the Information Society Technology (IST) program, (LION project, IST-1999-11387), by the CICYT (Spanish Ministry of Education) under contract TIC99-0572-C02-02 and by the Catalan Research Council (CIRIT) 2001-SGR00226

Internet Gateway Protocols (IGPs). In this paper, we deal with a potential use for this purpose of an adaptation of the part of the ATM PNNI protocol [1] designed for routing information exchange. The resulting adapted protocol is called Private Optical NNI (PONNI).

We think that PONNI can be a viable solution because while, for instance, OSPF has to be modified in order to achieve the capabilities needed to distribute both internal and external information on the available resources (e.g. remaining bandwidth, wavelengths, etc.), PNNI has these capabilities by nature. PONNI is defined in the context of pursuing the compatibility between both the ATM PNNI and GMPLS, using the best of each approach. Currently, we only deal with two of the ASON control plane procedures, namely the Routing Information Exchange and the Signalling. Regarding the signalling, we assume to use the GMPLS signalling such as is recommended by the IETF. The compatibility of PONNI with the GMPLS signalling will be part of further studies.

This paper is a starting point to define a Private NNI as an adaptation of the ATM PNNI protocol for ASON. We focus on the Routing Information Exchange procedure used in IP over ASON environments. In consistence with this, the following issues are taken into account: 1) Providing to ASON with a hierarchical structure in order to assure the scalability for large worldwide networks. 2) Exchange Optical Information. Each node exchanges Hello messages with its neighbours in order to determine what is its local state information. This information includes a node Identifier, a subnetwork Identifier and the status of the links between the node and its neighbours. 3) Exchange Non-Optical Information: PONNI Augmented Routing (POAR), which allows information about non-ASON client networks to be distributed in an ASON.

Although in this paper we consider that the optical transport network clients are IP routers, for the PONNI definition it is also assumed that the protocol can be used for other type of clients such as ATM, SDH, etc.

This paper is organized as follows: Section 2 is devoted to define the PONNI protocol, as a solution for exchanging Routing information in Optical Networks. Section 3 deals with defining the PONNI Augmented Routing (POAR), which have to allow the topology/resource information about non-Optical clients to be distributed through the ASON. Section 4 is devoted to the Proxy POAR, which allows network clients to obtain and register information about non-optical services from and to the ASON. Finally, in Section 5, a case study based on applying the PONNI protocol to an hypothetical configuration of the Pan-European Network is discussed.

2. Topology Information Distribution Protocol

2.1. Hierarchical Structure

Concerning the introduction of a hierarchical structure in the optical network, here it can be used the same structure that was defined for ATM PNNI. That is, a lowest hierarchical level where the optical network is organised in physical subnetworks, and several hierarchical upper levels organised as logical subnetworks. The lower level is organised in groups of peer physical nodes (i.e. Optical Cross Connect (OXC) with similar features) and physical links. Physical links are full-duplex and can have different features in each one direction. Therefore, there are two sets of parameters (i.e. transmission port identifier and node identifier) to define a link. The nodes in a subnetwork exchange information in order to maintain an identical topology database. Moreover, a specific node, called Subnetwork Leader (SL), summarises topology information within the subnetwork. The main task of the SL is to aggregate and distribute information for maintaining the PONNI hierarchy. A Logical Subnetwork Node (LSN) represents a subnetwork in the upper level and the SL executes the functions needed to perform this role (see Figure 1)

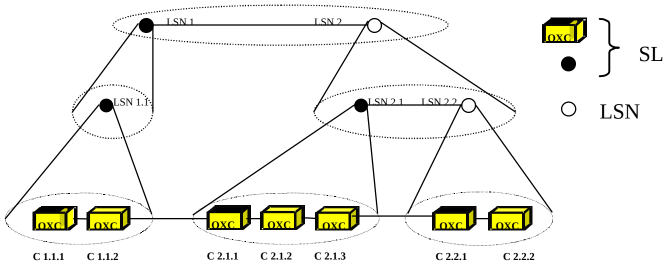

Figure 1. Hierarchical structure for the optical network

On one hand, each subnetwork has to be identified by a subnetwork Identifier (sID). Since GMPLS uses IP addresses, we suggest using the 64 right-most bits of the IPv6 addresses (i.e. the Format prefix, Top-level Aggregation Identifier, Reserved, Next-level Aggregation Identifier and Site-level Aggregation Identifier fields) [2] as sID. Thereby, sID is a prefix of IPv6 addresses such that the organisation that administer the subnetwork has assignment authority over that prefix. Therefore, sID will be encoded using 9 bytes, 1 byte (8 bits) as a level indicator plus by 8 bytes (64 bits) of an identifier information field. The value of the level indicator will be between zero and 64. The value sent in the identifier information field will be encoded with the 64-*n* right-most bits set to zero, where *n* will be the level.

On the other hand, each OXC has to have assigned a node Identifier (nID), which will be composed of 8 bytes distributed as follows: the first byte (level indicator) will specify the level of the subnetwork where the node will be contained. The second byte will take value 100 in order to help to distinguish between physical node and LSN. The remainder of the node ID will contain the 16 bytes IPv6 address of the system represented by the node.

2.2. Exchange of Information

Once the above suggested hierarchy structure is established, each node will be able to exchange Hello messages with its neighbours in order to determine the local state information. This information will include the node Identifier (nID), the subnet Identifier (sID), and the status of its links to the neighbours. Each node would bundle its state in a PONNI Topology State Element (POTSE), which will be flooded throughout the optical subnetwork. Then, a topology database consisting of a collection of all the POTSEs received at nodes would provide the information required in order to compute proper routes for any couple of source-destination nodes belonging to the optical network. The suggested POTSE format is shown in Figure 2.

POTSE Header
Type=64 (POTSE)
Length
POTSEType
Reserved
POTSE Identifier
POTSE Sequence Number
POTSE Checksum
POTSE Remaining Lifetime
POTSE contents
.
.
POTSE contents

Figure 2 POTSE Format

The POTSE contents will be based on the OIF NNI routing requirements [3]. In this way, we suggest the following information to be contained in the POTSE:

- *Nodal Information Group;* consisting of general information about the node to be used for selecting the SL and setting up the PONNI hierarchy. We suggest using the same information included in [1] except the node address, which will be an IP address.

- *Inter-domain link resources to select a path*, which will have to be checked by clients to see if their requested bandwidth requirements shall be provided.

- *Reachability information;* consisting of the reachable client addresses and the next node control plane address, in order to be used by a node to inform to its neighbours about the clients reachable through itself.

- *Directionality attributes*; required to specify if an optical connection is a unidirectional or bi-directional connection.

- *Traffic Engineering (TE) information;* consisting of a set of TE attributes through the different domains. This information is to be used to optimise the network resources utilisation.

- *Transport service information*; required to select a path, which satisfies the client transport service requirements.

- *Protection capability information*; to be used for protection and restoration purposes.

- *Shared Risk Link Group (SRLG)*; required for the end-to-end SRLG disjoint diverse path service to provide that the SRLG information be globally consistent.

2.3. Flooding Mechanism

The flooding is the advertising mechanism to be used by the PONNI protocol. It has to provide a hop-by-hop propagation of POTSEs throughout the optical subnetworks. This mechanism has to assure that all the nodes belonging to the same optical subnetwork have a similar topology database.

We suggest using the same flooding mechanism defined for ATM PNNI. Thus, the PONNI flooding mechanism would consist of two steps, namely encapsulation of POTSEs in “PONNI Topology State Packets” (POTSP) in order that can be transported through the network; and examination of the POTSP components at nodes. The network nodes, when receiving POTSPs would extract the POTSEs from them (each POTSE would be recognised by its header), and would check if the POTSEs are new or more recent than the POTSE still installed in that node. Then new/updated information would be installed in the topology database and would be propagated to the rest of the neighbours except to the node that sent the POTSE. Note that in the flooding process each node would issue POTSPs containing POTSEs with updated information, and that POTSEs installed in the topology databases depends on a refresh time (if POTSEs are not refreshed within a certain time, then they are eliminate from the topology database).

3. PONNI Augmented Routing

We define PONNI Augmented Routing (POAR) as an adaptation of the PNNI Augmented Routing [4]. So then, POAR will be an extension to PONNI routing to allow information defined about Non-ASON services to be distributed in an ASON.

As a consequence, the content and format of the above mentioned information will be specified by POAR but will be transparent to PONNI routing. A POAR-capable device, one that implements PONNI and the POAR extension, has to be able to create POAR POTSEs describing the Non-ASON services located on or behind that device. Since this information will be flooded by PONNI routing, POAR-capable devices have also to be able to examine the POAR POTSEs in the topology database that were originated by other nodes to obtain information on desired services reachable through the ASON.

An important example of how POAR can be used is provided by the overlay routing on ASON backbones. Considering two IP subnetworks connected through an ASON backbone. If the routers are POAR-capable, they can create POTSEs to advertise the routing protocol supported on the given interface (e.g., OSPF, RIP, or BGP), along with their IP address and subnetwork, and other protocol-specific details. The POAR-capable routers can also automatically learn about "compatible" routers (e.g., supporting the same routing protocol, in the same IP subnetwork) active in the same ASON network. In this manner, the overlay routing network can be established automatically on an ASON backbone. The mechanism is dynamic, and does not require configuration. One potential drawback of POAR is that a device must implement PONNI in order to participate. Therefore, an additional set of optional protocols called Proxy POAR has been defined to allow a client that is not POAR-capable to interact with a server that is POAR-capable and thus obtain the POAR capabilities. The server acts as a proxy for the client in the operation of POAR. The client is able to register its own services, and query the server to obtain information on compatible services available in the ASON network.

POAR will use a specific POTSE type to carry this non ASON-related information. The Information Groups (IGs) for the flooding of Ipv4-related protocol information, such as OSPF or BGP, used in ATM PAR, could be used in POAR as well. Moreover, We suggest a new set of IGs in order to be included in POTSEs: PAR MPLS Services Definition IG [4], PAR SDH Services Definition IG and PAR ATM Services Definition IG. These IGs allows no IP clients such as MPLS, ATM and SDH to be able to distribute its topology and available resource information through the ASON.

4. Proxy POAR

In the specific context of IP over ASON, client network nodes have to discover and register the different services offered by all the devices interconnected through the ASON. In order to achieve this, a proxy POAR has to be defined. Our definition of proxy POAR will be based on ATM proxy PAR [4].

According to this, the Proxy POAR would work in client mode on the current IP routers and would interact with the OXC, which would have the Proxy POAR server installed. One of the main advantages is that due to its simplicity in client implementation, it can be immediately incorporated into the existing devices (e.g. IP routers, Label Switching Router, etc.).

The main purpose of the Proxy POAR is to allow Non-ASON devices to use the flooding mechanisms given by PONNI, in order to discover and register the different services offered by all the devices interconnected through an ASON.

Now, we can proceed with an analysis of the Proxy POAR performance and interaction with the PONNI. The Proxy POAR will be asymmetric and it will be based on both a discovery level, such as the Hello PONNI protocol, and registration/query protocols directly supported by POAR functions. The discovery level will be useful to both initiate and maintain the communication between adjacent servers and clients so that the existing connections can always be known. Moreover, the registration/query protocols would be only executed when the adjacencies between the clients and servers were completely established. The configuration information about all the services that the client provides will be given to the server. The server will gather the information about all the clients directly connected to it, and after will assemble the information, this will be flooded through the ASON cloud where the PONNI will be running. This information cannot be used by the OXC in which the Proxy POAR is not installed. Query messages should be sent by the client to the server to allow the client to access the description of the services offered by the rest of the devices, with characteristics registered. So, all the routers with the Proxy POAR client installed will be able to obtain information about any other router, as they are directly interconnected.

Let us consider the scenario depicted in figure 3 as an example of POAR and Proxy POAR application. This example shows the mechanism used to transport MPLS information from a MPLS subnetwork (Domain 1) to other MPLS subnetwork (Domain 2) through an ASON backbone (Optical Backbone). The Proxy POAR client on each Border OXC (BOX) registers the MPLS protocol along with labels, and all the address prefixes, which can be reached. Every server bundles its state information in POAR POTSEs, which are flooded throughout the optical network. Then each client uses the query protocol to obtain information about services registered by other clients.

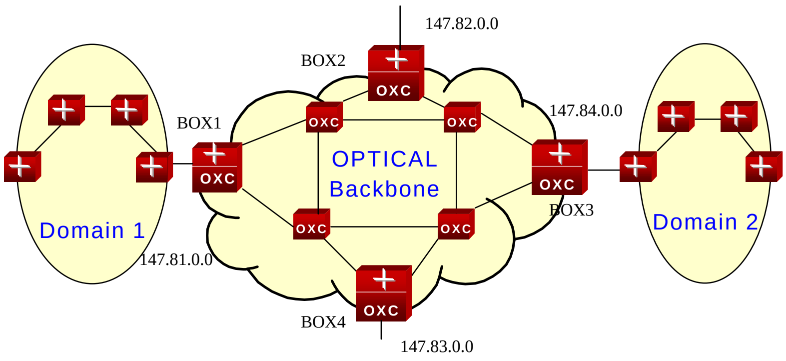

Figure 3 Example of POAR and Proxy POAR application

Using the received POAR MPLS devices Definition IG, every server side generates an MPLS topology database as shown in table 1.

Table 1 Topology Database

	@IP Dest	@OXC	Label Out		@IP Dest	@OXC	Label Out
BOX1				BOX3			
	147.82.2.1	@BOX2	0.50		147.82.2.1	@BOX2	0.50
	147.84.0.0	@BOX3	0.40		147.81.0.0	@BOX1	0.20
	147.83.2.0	@BOX4	0.30		147.83.0.0	@BOX4	0.30
BOX2				BOX4			
	147.81.0.0	@BOX1	0.20		147.82.2.1	@BOX2	0.50
	147.84.0.0	@BOX3	0.40		147.84.0.0	@BOX3	0.40
	147.83.2.0	@BOX4	0.30		147.81.0.0	@BOX1	0.20

5. Case Study

Once PONNI has been defined, in this Section, we are going to evaluate its performance considering a case study. This case study is based on the Pan-European Network, and consists of comparing the flat topology of this network with a hypothetical hierarchical structure depicted in Figure 4.

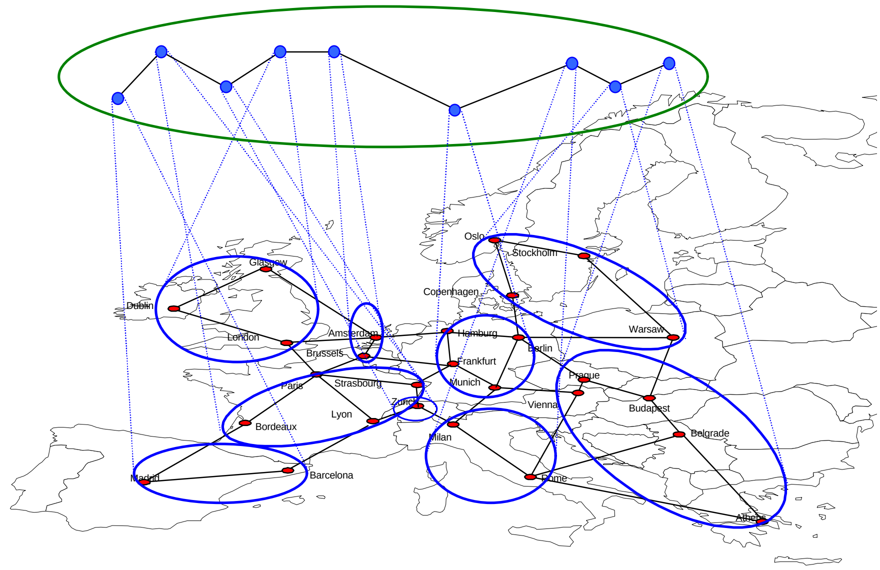

Figure 4 Hypothetical hierarchical structure based on a Pan-European Network topology

As can be observed in Figure 4, in order to obtain the hierarchical structure the Pan-European Network is divided in several physical subnetworks, - each of them represented at the upper level by a LSN. The resulting structure is composed by 27 nodes, 39 links, 9 subnetworks, and 2 hierarchical levels. The flat network is composed of the same physical connections and the same number of nodes, but all its nodes belong to a single subnetwork and a routing hierarchy does not exist.

In order to study he performance of the PONNI, both the flat network topology and the hierarchical network topology were simulated using the ATM PNNI routing Protocol Simulator (APRoPs) [7]. This simulator allows the scalability, robustness and maintainability of PONNI routing status, which is required for carrying out the evaluation.

The results of this simulation are shown in Table 2. This includes the time and the amount of data required for the PONNI routing to reach the stability, and for maintaining the stability. The stability is the completion of database synchronisation, i.e. all the nodes in a same subnetwork have identical topology databases. Maintainability means to observe the amount of data required in order to maintain the PONNI routing status after initial stability is reached.

Table 2 PONI routing stability

	Database Synchronization		Maintain PNNI Routing status	
	FLAT	Hierarchy	FLAT	Hierarchy
Time (seconds)	6,1	4,1	600	600
Total-Data (KBytes)	1,576	105	2,08	810
Data (no Hello) (KBytes	1,551	75	1,551	102

Note that the difference between the flat and the hierarchical structure is about one order of magnitude in favour of the hierarchical structure.

Furthermore, we have simulated the scalability of the PONNI protocol from a flat network to a hierarchical network. In order to perform this simulation, we have added 12 nodes to the initial configurations. Concerning the hierarchical structure, the following logical configurations were considered: 1) subnetwork with 12 nodes, 2) subnetworks with 6 nodes each, 3) subnetworks with 4 nodes each and 4) subnetworks with 3 nodes each. The results obtained with these simulations are shown in Figure 5.

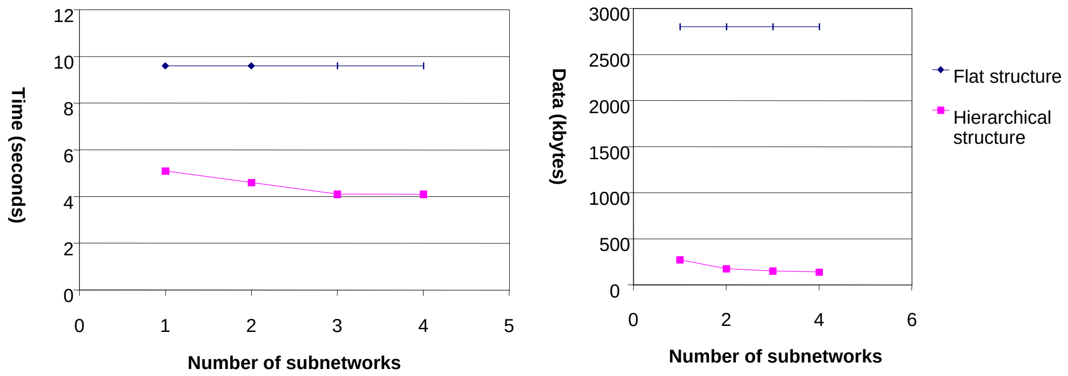

Figure 5 Scalability from flat network to a hierarchical network

Figure 5 shows a significant difference between the flat and the hierarchical network regarding the topology database stability. When we only add one logical subnetwork to the hierarchical network the time needed to obtain a stable topology database is around 50% lower than the time needed in the flat network. Moreover, the amount of data needed to reach that stability in the flat network is around 90% higher than in the hierarchical network. In addition, if the nodes are organised in more than one subnetwork, then as the number of subnetworks increase both the time and the amount of data of stability decrease.

6. Conclusions

In this paper an adaptation of a part of the ATM PNNI protocols has been suggested to be used in Automatic Switched Optical Networks. In particular it has been proposed a routing information exchange protocol called PONNI, which provide the ASON with a hierarchical structure, a topology information flooding mechanism and a protocol extension to distribute topology and resource information about non-optical clients through the ASON. This proposal makes use of the ATM PNNI features, which can be adapted better to the ASON and, at the same time, it takes into consideration the ASON control plane based on GMPLS.

According to the simulation results obtained in a case study based on the Pan-European network simulation, the PONNI protocol improves the topology database stability compared with a flat network,

This paper is a preliminary attempt to a complete adaptation of the ATM PNNI protocol to ASON, which is a task being carried out within the Layers Interworking in Optical Network (LION) project [8].

7. References

[1] ATM Forum. “Private Network-Network Interface Specification Version 1.0”, af-pnni-0055.000, March 1996.

[2] R. Hinden, M. O'Dell, S. Deering “An IPv6 Aggregatable Global Unicast Address Format” Rfc2374, July 1998.

[3] Angela Chiu, et al., “NNI routing requirements”, oif2001.508, October 2001.

[4] ATM Forum. “PNNI Augmented Routing (PAR) Version 1.0”, af-ra-0104.000, Jan. 1999

[5] Sergio Sanchez-Lopez, Xavier Masip-Bruin, Jordi Domingo-Pascual, Josep Sole-Pareta, Juanjo Lopez-Mellado, “A Path Establishment Approach in an MPLS-ATM Integrated Environment”, IEEE GlobeCom’01, San Antonio, Texas (USA), November 2001.

[6] http://www.carrier1.net, http://www.colt-telecom.com, http://www.ebone.com, http://www.globalcrossing.com, http://www.i-21future.com, http://wwwkpnqwest.com, http://www.viatel.com

[7] Song, Y., “ The NIST ATM PNNI Routing Protocol Simulator (APRoPS), Operation and Programming Guide”, version 1.0, December 1998.

[8] Information Society Technology (IST) program, ”LION project”, IST-1999-11387, 1999-2001