An Architecture for the Self-management of Lambda-Connections in Hybrid Networks

Tiago Fioreze, Remco van de Meent, and Aiko Pras

University of Twente, Enschede, the Netherlands
{t.fioreze, r.vandemeent, a.pras}@utwente.nl

Abstract. Hybrid networks are networks capable of switching data at multiple levels (optical and IP packet level) by means of multi-service optical switches. As a result of that, huge flows at the IP-level may be moved to the optical-level, which is faster than the packet-level. Such move could be beneficial since congested IP networks could be off-loaded, leaving more resources for other IP flows. At the same time, the flows switched at the optical-level would get better Quality of Service (QoS). In order to achieve this beneficial move, huge IP flows have to be properly detected at the packet-level and lambda-connections are to be established for them at the optical-level. Two approaches are currently used for that purpose: the first is based on conventional management techniques and the second is based on GMPLS signaling. Both approaches mostly depend on human intervention, which can be error prone and slow. The idea proposed in this paper to overcome this problem consists of adding self-management capabilities to the multi-service optical switches. The optical switches would then be responsible for automatically identifying IP flows, and establishing and releasing lambda-connections for such flows. The main goal of this paper is therefore to propose an architecture for the self-management of lambda-connections in hybrid networks.

1 Introduction

Hybrid networks are networks that allow data to be switched both at the IP packet-level and at the optical-level [1]. An example of a hybrid network is SURFnet6 [2], the Dutch research and education network. Hybrid networks are composed by multi-service optical switches, which have the capability to perform forwarding decisions at different levels in the protocol stack. As a result of that, big and long IP flows (so called elephant flows) could be moved from the packet-level to the optical-level. It is anticipated that such a move results in a better Quality of Service (QoS) for both elephant flows and remaining IP flows at the packet-level: the former would have less delay and jitter and plenty of bandwidth at the optical-level; the latter would be better served due to the off-loading of elephant flows.

An accurate identification of IP flows and a proper management of lambda-connections are important tasks to achieve the desired move. Two approaches are currently used for that [3]: conventional management and GMPLS signaling. The former is characterized by a centralized management entity (e.g., human manager or an automated management process) that is in charge of establishing lambda-connections and deciding which IP flows should be moved to the optical-level. In contrast, the latter

is characterized by the fact that optical switches coordinate the creation of lambda-connections among themselves after being triggered for that. The decision which IP flows will be moved to the optical-level however is taken by a centralized entity or by the entities exchanging data flows.

Both approaches, however, have some shortcomings. Both approaches require human interaction to detect flows and manage lambda-connections. This interaction may be slow and error prone. Currently, when a lambda-connection is requested within one single domain (intra-domain), several steps are taken (e.g., phone calls and emails exchanging) between requesters and network domain administrators in order to establish the lambda-connection. Evidently, it may take hours or more before a desired lambda-connection can be used. When the requests for a connection spans multiple domains (inter-domain), the lambda-connection provisioning may take even much longer. In addition to that, a troubleshooting process may be needed to solve connection problems, which may delay the connection setup still more.

Moreover, several big IP flows may, for instance, be using resources at the packet-level while the lambda-connection is being established, and therefore possibly congesting the IP-level. Moreover, by the time that the lambda-connection is finally established the elephant flows may no longer exist or not be large enough for a dedicated lambda-connection.

Our solution to overcome these shortcomings consists of extending the GMPLS approach by automatically detecting IP flows eligible for lambda-connections. With this extension, multi-service optical switches automatically detect IP flows and establish/release lambda-connections for them. This can be characterized as a self-management behavior. In this context, the main goal of this paper is to propose an architecture for the self-management of lambda-connections in hybrid networks.

The remainder of this paper is organized as follows. Section 2 shows the current approaches for the management of lambda-connections. Then, sect. 3 starts by presenting the shortcomings of the current approaches, which have motivated our proposal. In addition, sect. 3 also introduces our architecture for the self-management of lambda-connections. Finally, conclusions and future plans are outlined in sect. 4.

2   Current Management Approaches

This section describes the two current approaches used to manage optical networks: the conventional management approach and the GMPLS signaling approach.

2.1   Conventional Management

The conventional management approach is composed by managers and agents [4]. Managers consist of entities that are responsible for managing the network activity by ordering tasks (e.g., configuration or monitoring actions) for agents. Agents are entities in charge of performing the requested tasks. There may also be entities that can play a dual role, acting as both a manager and an agent.

In the optical networks context a manager can be a centralized management entity (e.g., human manager) that is responsible for managing optical switches (agents) in the

optical domain. This centralized management entity is responsible for identifying IP flows to the optical-level and as well the establishment of the optical connections.

The identification of IP flows can be done with the help of monitoring mechanisms, such as NetFlow [5]. On its turn, the establishment of lambda-connections can be performed by using management technologies such as command line interface [6], the well-known SNMP protocol [7] and the TL1 language [8]. When the required lambda-connection spans multiple optical switches, the establishment of the lambda-connection involves setting up each optical switch along the desired path.

2.2   GMPLS Signaling

The Generalized Multiprotocol Label Switching (GMPLS) architecture [9] aims at extending the characteristics of the MPLS architecture [10] to support peculiarities existing in today's optical networks. GMPLS extends MPLS in order to provide to the control plane (signaling and routing) with new label capabilities. GMPLS supports spatial (port), lambda, and time-division switching, besides the traditional packet switching. Unlike in MPLS, however, in the GMPLS architecture the labels are no longer carried in the data, but they are defined in the optical switches.

With regard to the configuration process of the optical switches, GMPLS works similarly to MPLS by using signaling messages in order to establish lambda-connections. On the other hand, regarding to its operational model, GMPLS can be deployed as two different operational models [11]: peer and overlay model.

In the peer model, the topology of the core network is not hidden from users (e.g., adjacent IP networks) of the optical networks, which enables the users to see the entire optical network topology and to choose the desired lambda-connection path. Once the desired path is chosen, the users have to communicate with the most adjacent optical switch by informing it with the desired path, the source and destination addresses of the selected IP flow, and also inform GMPLS signaling parameters (e.g., switching type). Once the adjacent optical switch gets this information, it starts then the process of establishing the desired path by interacting with other switches along the path.

In the overlay model, only the adjacent optical switches are revealed to users of the optical network; topology of the core network is hidden. Hence, users are not able to choose their entire desired connection path. Therefore, to create a lambda-connection, the users can inform their adjacent switch only with the source and destination addresses of the selected IP flow and the GMPLS signaling parameters. The adjacent switch then interacts with other switches to decide which path will be chosen (by using interior gateway protocols such as OSPF or IS-IS) based on the connection parameters provided by the users.

3   The Self-management Architecture

This section presents our architecture for the self-management of lambda-connections in hybrid networks. The section starts by showing some shortcomings of the current approaches and by presenting our idea to overcome them. Then, our proposed architecture is introduced by presenting first its functional part and then its physical part.

3.1 Self-management of Lambda-Connections

The management approaches presented in section 2 have some shortcomings. Both approaches depend on human intervention to select and move IP flows to the optical-level and establish/release lambda-connections. This intervention can be therefore prone to errors and take considerable time to be performed (e.g. weeks).

In the conventional management approach, the network manager has the task of deciding which flows will be moved to the optical-level. This decision is mostly made manually by collecting network information and analyze it. Nonetheless, the network manager has also to configure each optical switch along the chosen path in order to establish lambda-connections for the selected flows. In addition to that, he is required to release the connections when no longer needed as well.

On its turn, the GMPLS signaling approach offers some autonomy in the steps of establishing and releasing lambda-connections. However, these steps must still explicitly be triggered by the users or network managers of the optical network. In addition to that, these users and network managers must also to provide the information about which IP flows will be moved to the optical-level.

Our solution to overcome these shortcomings consists of providing self-management capabilities to optical switches. Our self-management solution allow optical switches to be in charge of automatically selecting IP flows to the optical-level and as well creating/releasing lambda-connections for them. Network managers would only be required to configure the optical switches in order to instruct them on which flows to look for and when lambda-connections have to be established and released by using GMPLS signaling protocols. Once configured, the optical switches cooperatively work by their own. Figure 1 depicts how our proposal for a self-management of lambda-connections in hybrid networks would look like.

In Figure 1 IP and optical domains coordinate with one another in order to detect IP flows and manage lambda-connections. Both domains are assumed as already been configured by network managers. IP routers located at IP domain B are
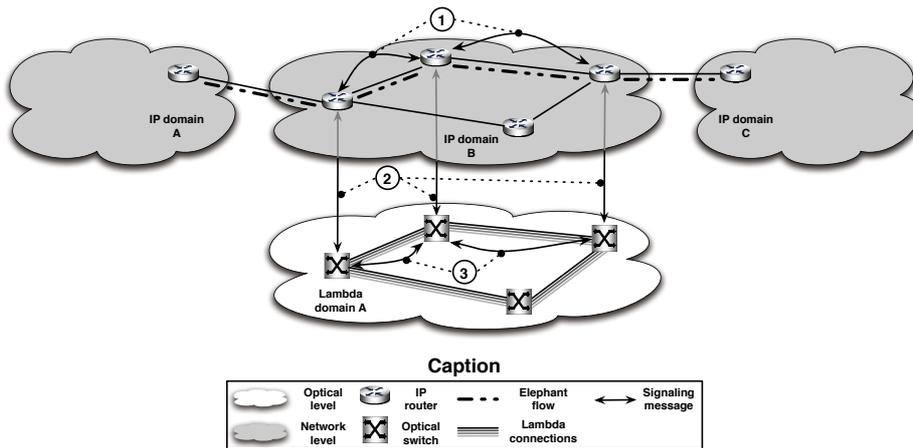

Fig. 1. Self-management of lambda-connection in hybrid networks

exchanging network information (e.g. bandwidth consumed per flow and its duration) regarding to the existence of a elephant flow transiting between IP domains A and C (step 1). Based on this exchanged information and the configuration performed by network managers they make decisions on if a flow is eligible or no longer eligible for a dedicated lambda-connection at the optical-level. If the decision is in favor of creating a lambda-connection (i.e., the elephant flow is eligible to be moved to the optical-level), the IP routers signal the optical switches in lambda domain A (step 2). Then, the optical switches coordinate among themselves in order to create a dedicated lambda-connection to the detected elephant IP flow (step 3). From that point on, the elephant flow is switched at the optical-level and IP routing is accordingly changed. Further information about our self-management approach can be found at [12].

3.2 Functional Architecture

Our functional architecture presents the functional blocks involved in our self-management approach and as well their interaction. Our architecture deals with the layers 1 (Network interface layer) and 2 (Internet layer) of the four-layer TCP/IP network architecture, as can be seen in figure 2.
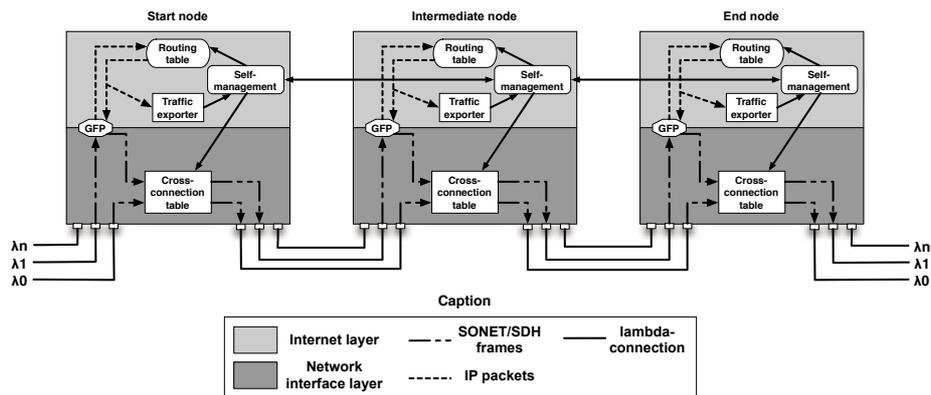
Fig. 2. Self-management functional blocks

Figure 2 shows 4 functional blocks: cross-connection and routing tables, traffic exporter, and a Generic Framing Procedure (GFP) module, which maps IP packets into the underlying transport protocol. In the case of SURFnet6, the underlying transport layer is based on SONET/SDH. Figure 2 also shows our self-management functional block.

The main tasks of the self-management block are to analyze network information and decide when a lambda-connection should be established/released for a certain set of flows. The analysis consists of obtaining network information from the traffic exporter and characterizing it (e.g. by ordering and/or filtering fields). Then, self-management blocks correlate their collected information and cooperatively decide when

Fig. 3. Zooming in into a self-management functional block

establishing/releasing a lambda-connection. Once decided, the self-management blocks involved in the decision process locally adjust the routing and cross-connection tables. Figure 3 shows more internal details of the self-management functional block.

The self-management functional block is composed by 5 elements: traffic characterizer, decision maker, lambda creator, lambda releaser, and active connections table. The traffic characterizer is in charge of collecting network information exported by the traffic exporter (e.g., a NetFlow exporter) and characterizing it. The characterized information is then sent to the decision maker, which cooperatively decides with other decision makers on moving IP flows from the IP-level to the optical-level and vice-versa.

If a decision to move IP flows to the optical-level is taken, then each decision maker contacts its local lambda creator, which is responsible for establishing lambda-connections. The lambda creator performs that by adjusting the routing and cross-connection tables. Once the lambda-connection is established, the lambda creator adds the new entry in the active connections table. The active connections table lists the current active connections locally held by certain network node.

On the other hand, if a decision to move IP flows back to the IP-level is taken, each decision maker contacts its lambda releaser, which is responsible for tearing down lambda-connections. The lambda releaser then reconfigures the routing and cross-connection tables in order to release the connection. Once the lambda-connection is released, the lambda releaser removes the entry in the active connections table.

3.3   Physical Architecture

The physical architecture consists of showing the physical location of the functional blocks. In our physical architecture, the functional blocks are located at two different physical locations: in the multi-service optical switches and in an external management device. The traffic exporter and routing and cross-connection tables are already supported inside current multi-service optical switches, so they will be kept where they are. This assertion is based on discussions with network managers. The remaining blocks are located in the external management device. Figure 4 shows our physical architecture.
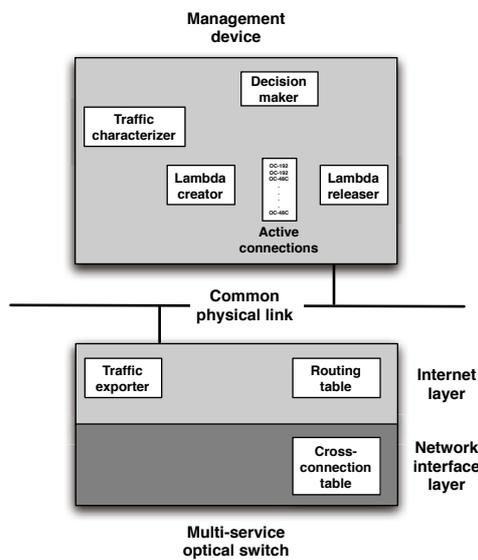
Fig. 4. Physical architecture

Even though all functional blocks could internally be implemented into the multi-service optical switches, doing that is not trivial because vendors may not be willing to change the implementation of the optical switches operating systems. That is the reason our new functional blocks will be implemented in an external management device.

4   Conclusions

Section 2 of this paper identified the current approaches to manage lambda-connection in hybrid networks. In practice two approaches are being used: conventional management, which is based on SNMP or TL1 manager-agent interactions, and GMPLS. Both approaches, however, require human interaction to detect flows and create / release lambda-connections. As discussed in Section 3.1, traditional approaches for lambda-connection management are therefore slow and error prone. To overcome these problems, the remainder of this paper proposed a functional and physical architecture for self-management of lambda-connections.

The functional architecture defines which functions and interactions are needed to perform self-management in a logical and comprehensible way. The physical architecture defines how these functions and interactions can be implemented; an important goal hereby is to avoid as much as possible modifications to current implementations of multi-service optical switches. This decision allows us to create, in later stages of our research, a testbed to evaluate our architecture.

This paper describes work that is still in progress. More research is needed, for instance, to determine proper configuration parameter values for the self-management functional blocks in order to decide when to establish and release a lambda-connection. A next goal is therefore to simulate our architecture in order to find answers for this question.

Acknowledgments

References

1. Gauger, C.M., Kuhn, P.J., Breusegem, E., Pickavet, M., Demeester, P.: Hybrid Optical Network Architectures: Bringing Packets and Circuits Together. IEEE Communications Magazine 44(8), 36–42 (2006)
2. SURFnet: SURFnet6 lighpaths mark start of the new Internet area (press release), Available in: `http://www.surfnet.nl/info/en/artikel_content.jsp?objectnumber=107197`
3. Bernstein, G., Rajagopalan, B., Saha, D.: Optical Network Control: Architecture, Protocols, and Standards. Addison-Wesley, Reading (2003)
4. Schoenwaelder, J., Quittek, J., Kappler, C.: Building Distributed Management Applications with the IETF Script MIB. IEEE Journal on Selected Areas in Communications 18, 702–714 (2000)
5. Claise, B.: Cisco Systems NetFlow Services Export Version 9. Request for Comments: 3954 (RFC 3954) (2004)
6. Schoenwaelder, J.: Overview of the 2002 IAB Network Management Workshop. Request for Comments: 3535 (RFC 3535) (2003)
7. Case, J., Mundy, R., Partain, D., Stewart, B.: Introduction and Applicability Statements for Internet Standard Management Framework. Request for Comments: 3410 (RFC 3410) (2002)
8. Man, F.-T.: A Brief History of TL1 Journal of Network and Systems Management 7 (1999)
9. Mannie, E.: Generalized Multi-Protocol Label Switching (GMPLS) Architecture. Request for Comments: 3945 (RFC 3945) (2004)
10. Rosen, E.C., Viswanathan, A., Callon, R.: Multi-Protocol Label Switching (GMPLS) Architecture. Request for Comments: 3031 (RFC 3031) (2001)
11. Banerjee, A., Drake, J., Lang, J., Turner, B., Kompella, K., Rekhter, Y.: Generalized Multiprotocol Label Switching: An Overview of Routing and Management Enhancements. IEEE Communications Magazine 39(1), 144–151 (2001)
12. Fioreze, T., Pras, A.: Using self-management for establishing light paths in optical networks: an overview. In: Poster session proceedings of the 12th EUNICE Open European Summer School 2006 (EUNICE 2006), pp. 17–20 (2006)