Supporting PIM-SM in All-Optical Lambda-Switched Networks

Marcos Rogério Salvador, Sonia Heemstra de Groot, and Diptish Dey

Communication Systems Group - Telematics Systems and Services (TSS)
Centre for Telematics and Information Technology - University of Twente
P.O. Box 217, 7500 AE, Enschede, The Netherlands
Tel +31 53 489-8013 - Fax +31 53 489-4524
E-mails: {salvador, heemstra, dey}@cs.utwente.nl

Abstract: *We describe a protocol to construct optical multicast trees in lambda-switched networks running PIM-SM. Based on the MPLS architecture, the protocol maps a multicast tree constructed by PIM-SM onto point-to-multipoint LSPs that are connected to form an equivalent WDM multicast tree. LSP set-up request is receiver-initiated in the protocol. A mechanism to collect information about the capability of each WDM node along a path is used to provide receivers with sufficient information to form appropriate LSP set-up requests. This facilitates the set-up of LSPs, in particular in heterogeneous networks.*

Keywords: Protocols, Multicast, PIM-SM, WDM, Lambda Switching.

1 Introduction

Wavelength division multiplexing (WDM) has been the common choice of designers and engineers that need to upgrade communication network infrastructures, either to support the current demands or to meet the demands foreseen in the next generation networks. The main reason for this is the increase in capacity that WDM enables. By exploiting the frequencies of the light, WDM enables up to 25 Tbps of bandwidth, divided into distinct wavelength channels, per optical fibre. Furthermore, as an optical technology WDM allows for the bypassing of intermediate nodes. This results in lower end-to-end delays and bit rate and protocol transparencies.

Despite the huge bandwidth enabled by WDM and the continuous decline in the cost of bandwidth, a good network architecture design is still required. There are many arguments to support such a statement, the most important being the considerable increase in multicast traffic (e.g., tele-learning, Internet radio) that is expected to occur in the near feature. As shown in [1], with a reasonably large multicast group size, optical multicast can reduce bandwidth consumption by approximately 50% and decrease the number of required wavelengths by approximately 60%. This is because a network architecture that was not designed with optical multicast support in mind has to either multicast at the Internet Protocol (IP) layer, unicast at the WDM layer or a use a mixture of both.

The support of IP Multicast in the optical domain has already been investigated in [2], [3] and [4]. However, these studies concentrate on the support of dense mode routing protocols, which assume that bandwidth is plentiful and receivers are densely distributed. To the knowledge of the authors, there is no work in the literature that deals with the support of sparse mode routing protocols, which assume that bandwidth is not plentiful and receivers are sparsely distributed.

In this paper, we focus on the support of Protocol Independent Multicast-Sparse Mode (PIM-SM) [5], one of the sparse mode routing protocols under development by the Internet

Engineering Task Force (IETF). Specifically, we propose a protocol to support PIM-SM in all-optical lambda-switched networks. The protocol maps a multicast tree constructed by PIM-SM onto an equivalent one at the WDM layer.

The rest of this paper is organised as follows. In section 2, we introduce lambda-switched networks, the type of network for which the protocol is meant. In section 3, we explain how PIM-SM works. In section 4, we identify what the problems in supporting PIM-SM in lambda-switched networks are. In section 5, we describe the protocol. In section 6, we conclude the paper.

2 Lambda-switched Networks

Lambda-switched networks [6] are WDM networks that use the IETF's Multi-Protocol Label Switching (MPLS) architecture [7] as its basis and generalise the label concept to incorporate routing granularities ranging from bunch of fibres to a single wavelength time slot or packet flow. The expressiveness of labels provides lambda-switched networks with the potential to support not only several routing granularities, but also traffic engineering (TE) and virtual private networking (VPN).

A label is an identifier of a forwarding equivalence class (FEC), a grouping of packets that present certain commonality[1]. Packets of the same FEC are assigned the same label and forwarded over the same label-switched path (LSP). LSPs are unidirectional. Installation of a LSP is accomplished by proper set-up of the so-called label forwarding information base (LFIB) of each wavelength switch (WS) along the path. A LFIB is a cross-connect table that links input triples of the form <interface, wavelength, label> to output triples of the same form.

An all-optical lambda-switched network may operate in either a circuit switching or a virtual circuit-switching mode. In the circuit-switching mode, the set-up of a LSP leads to the immediate configuration of the fabric of each wavelength switch (WS) along the path. As a packet enters the network, a network layer lookup takes place in order to classify the packet into a FEC. The packet is then forwarded over the FEC's corresponding LSP's output interface (OIF) and outgoing wavelength. Switching at each node along the LSP is done at the optical layer based on the label information that is derived from the packet's input interface (IIF) and incoming wavelength. No electronic processing takes place whatsoever.

In the virtual circuit-switching mode, the configuration of a WS's fabric occurs on-the-fly based on the processing of incoming label information. As a packet enters the network, it is mapped into a FEC and assigned a proper label. The packet is then forwarded over the corresponding LSP's OIF and outgoing wavelength. The label is forwarded on a separate channel. At each subsequent node, the packet is delayed in fibre loops while the label is (currently, electronically) matched with the node's LFIB. If a match is found, the label is replaced with the outgoing label and the controlled WS fabric is set-up accordingly. Otherwise, the packet is discarded. At the egress node, the label is discarded, the packet's wavelength is dropped and the packet is processed as usual.

MPLS defines two ways to set-up LSPs. In independent LSP control, a LSP is set-up in a distributed way, just as with hop-by-hop routing. In ordered LSP control, the set-up of a LSP starts at an egress node and follows in the upstream direction. In either case, the decision to bind a label to a FEC is done at a downstream node with respect to that FEC.

[1] Currently in the Internet, a router maps a packet into a FEC based on the packet's destination address.

LSP set-up can be triggered by different events. In the request driven case, LSP set-up is triggered upon interception of appropriate control messages (e.g., routing messages). In the topology driven case, LSP set-up is triggered according to changes in the network layer's routing table. In the traffic driven case, LSP set-up is triggered upon arrival of data.

3 PIM-SM

PIM-SM uses unidirectional shared trees to distribute multicast traffic. Shared trees are also called rendezvous point (RP) trees, or simply RPTs, because traffic of a given session flows through a central point known as RP.

PIM-SM is a soft state protocol based on an explicit join model. To receive multicast traffic of a session, say G, herein denoted as (*, G) traffic, a router must periodically inform an upstream router that the latter should forward incoming (*, G) traffic over the OIF leading to the former. Despite the fact that an OIF is automatically pruned if its state has not been refreshed for a certain time, a router must always inform an upstream router when it is no longer interested in a given session.

A RPT is constructed as follows. A router detects a new group in its membership information base (MIB) and, consequently, creates a multicast route entry of the form (*, G) for that group. An (*, G) entry tells a router to forward source-independent, group G's traffic over the list of OIFs if the traffic's input interface (IIF) is the reverse path forwarding (RPF) interface towards the RP. The list of OIFs, the RP address and the group address G are extracted from the MIB's detected new entry. The RPF interface is derived from the RP address.

The router proceeds by sending a JOIN message over the RPF interface towards the RP. The message consists of the following fields:

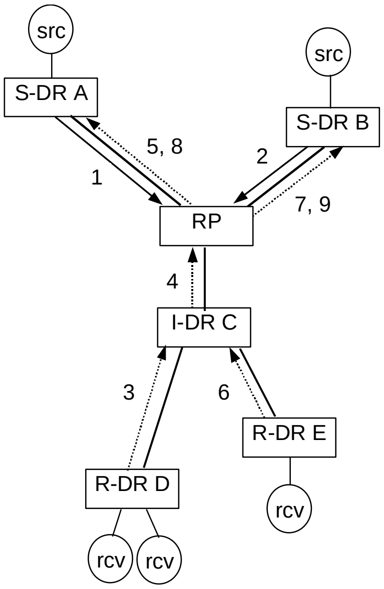
- Multicast-Address: the group address of the multicast session (i.e., G);
- Join: the tree's root address (i.e., RP in this case);
- Wildcard bit (WC-bit): when set to 1 it tells a router that received the message that traffic of the group address identified in Multicast-Address, regardless of its source, must be forwarded over message's IIF;
- RPT-bit: when set to 1 it indicates that the message should flow towards the RP.

At every subsequent router, an (*, G) entry is created. The message's IIF is added into the entry's list of OIFs. The entry's IIF is updated with the interface that the message will be forwarded over, i.e., the RPF interface towards the RP. Eventually, the message reaches either an upstream router that has an (*, G) entry already created or the RP. In either case, only the entry's list of OIFs is updated, as described above. This ends the receiver side of the protocol.

We now look into the sender side of the protocol. Initially, a router with a directly connected source, herein referred to as source-designated router (S-DR), encapsulates each packet in a REGISTER message and unicasts this message to the RP. The RP de-encapsulates each message and forwards the enclosed packet to the downstream members. Aiming at speeding up packet forwarding, the RP may decide to request the set-up of a unidirectional shortest path tree (SPT) from a S-DR, say S, to itself. The RP accomplishes this by creating a (S, G) entry and sending periodic JOIN messages towards the S-DR S. Every subsequent router along the path creates a (S, G) entry and forwards the message upstream. When the message reaches the S-DR S, the latter creates a (S, G) entry and starts sending traffic over the entry's OIF (i.e., the message's IIF).

The RP signalises the S-DR S to send packets natively by sending a REGISTER-STOP message to S-DR S. Upon reception of such a message, S-DR S stops encapsulating packets in REGISTER messages and starts sending packets natively.

Figure 1 illustrates the construction of a RPT.



Messages description
1. REGISTER(S1, G)
2. REGISTER(S2, G)
3. JOIN(*, G)
4. JOIN(*, G)
5. JOIN(S1, G)
6. JOIN(*, G) *
7. JOIN(S2, G)
8. REGISTER-STOP(S1, G)
9. REGISTER-STOP(S2, G)

* I-DR C does not forward message because (*, G) entry already exists.

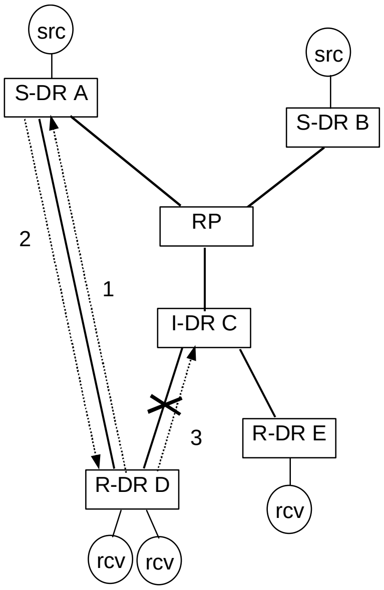
Figure 1 - RPT construction process

A router with directly connected receivers, herein called receiver-designated router (R-DR), can switch to the SPT from a source, say S, after receiving packets from S over the RPT. To accomplish this, the R-DR first creates a (S, G) entry with the SPT-bit cleared. A (S, G) entry tells a router to forward (S, G) traffic over the list of OIFs only if the traffic's IIF is the RPF interface towards S-DR S. The list of OIFs is copied from the (*, G) entry. A cleared SPT-bit indicates that the set-up of the SPT is not complete and, therefore, (S, G) traffic coming over the RPT must still be forwarded over the list of OIFs.

The R-DR next sends a JOIN message over the RPF interface towards S-DR S, specifying S in the field Join and setting both the WC-bit and the RPT-bit to 0. At every subsequent router, a (S, G) entry is created, with the list of OIFs containing the message's IIF and IIF containing the RPF interface towards S-DR S. The message flows upstream until it reaches S-DR S or an upstream router that is already a member of the SPT from S-DR S. In either case, the corresponding entry's list of OIFs is updated, as explained above.

A R-DR with a (S, G) entry and a cleared SPT-bit may receive duplicates of the same packet, one coming via the RPT and another coming via the SPT. If a router with a (S, G) entry and a cleared SPT-bit receives a (S, G) packet and the packet's IIF is not the RPF interface towards the RP, the router sets the SPT-bit and sends a PRUNE message towards the RP. A PRUNE message contains the same fields as a JOIN message. By specifying S in the Join field and setting the RPT-bit, a router informs that it is no longer interested in (S, G) traffic coming via the RPT. At every subsequent router, if an (S, G) entry exists then the message's IIF is removed from the entry's list of OIFs. Otherwise, the router creates an (S, G) entry with the RPT-bit set to 1. The PRUNE message is then sent upstream, eventually reaching the RP or an upstream router that already has an (S, G) entry with the RPT-bit set to 1.

Figure 2 illustrates how this RPT-SPT switchover process works.



Messages description
1. JOIN(S1, G)
2. Data(S2, G)
3. PRUNE(S1, G, WC-bit=0, RPT-bit=0) *

* I-DR C does not forward message because the (*, G) entry's list of OIFs is not empty (i.e., an OIF towards R-DR E exists).

Figure 2 - RPT-SPT switchover example

R-DRs may join a RPT at any time, either with or without S-DRs registered with the RP. Nevertheless, if no R-DR has joined the RPT, the RP sends REGISTER-STOP messages towards the S-DRs informing them that they should stop transmitting. If a SPT has been set-up between S-DR, say S, and the RP, the latter must send a PRUNE message towards the former.

The JOIN message and the PRUNE message were described as distinct messages. However, in PIM-SM these two messages are merged in a single message, the JOIN/PRUNE message. To minimise protocol overhead, a single JOIN/PRUNE message contains both a list of join requests and a list of prune requests.

4 PIM-SM over All-optical Lambda-switched Networks

Before going into the details of the protocol it is important to understand what supporting PIM-SM implies at the IP layer as well as at the WDM layer. To this end, we consider the two main functions of an IP router, namely, routing (calculation) and forwarding. In the context of PIM-SM, routing is the process of consolidating JOIN/PRUNE messages with topology information to create and maintain a multicast forward information base (MFIB), i.e., a multicast tree. Forwarding is the process of matching some fields (e.g., source address and destination address) of an incoming packet's header with a MFIB in order to put the packet into the proper output queue for transmission.

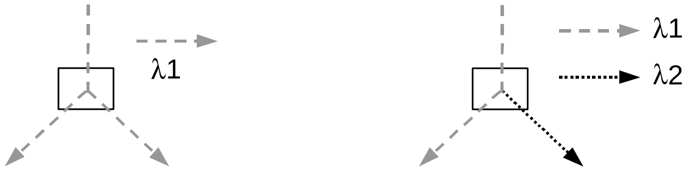
The multicast capability of a network element is given by its forwarding capability. Because forwarding in IP involves memory manipulations only, and memory manipulation is extremely flexible in the electronic domain, every IP router can support the point-to-multipoint, multipoint-to-point forwarding patterns[2] of PIM-SM (as long as the router runs PIM-SM, of course).

The same does not hold for WSs. Firstly, forwarding in WDM is performed in the optical domain and optical technologies (e.g., memories) are still at early stages of research. Secondly, forwarding in WDM is more complex in the sense that it involves both the space domain and the wavelength domain. To explain how limitations in optical technologies affect

[2] Multipoint-to-multipoint is a general case of both point-to-multpoint and multipoint-to-point.

the multicast capability of a WS we consider point-to-multipoint and multipoint-to-point forwarding separately.

Point-to-multipoint forwarding is (currently) accomplished in WDM, in both domains, by two techniques, namely, light splitting and wavelength conversion. The most fundamental, light splitting is a simple technique that splits an incoming light signal into a number N of outgoing light signals, each on a distinct OIF (see Figure 3a). Wavelength conversion is a technique that allows for the forwarding of these outgoing signals using wavelengths others than the incoming signal's wavelength (see Figure 3b).

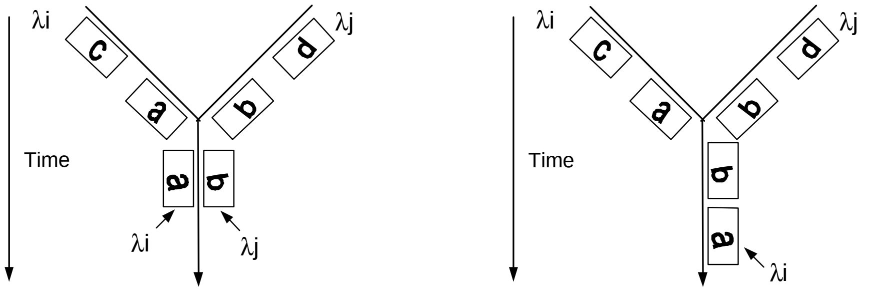


(a) without wavelength conversion (b) with wavelength conversion

Figure 3 - Point-to-multipoint forwarding example

The main problem with light splitting is its impact on the network signal power. The power of each outgoing signal is reduced by a factor of N. Furthermore, network signal power calculation and configuration are done statically while multicast membership is dynamic. The main problem with wavelength conversion is that it is still an immature technology that adds into the complexity and cost of network nodes and network architectures (the reader is referred to [8] for more information on this matter).

Multipoint-to-point forwarding can be accomplished in both domains as well. In the space domain, two or more incoming signals, each on a distinct wavelength channel and IIF, are forwarded over the same OIF (see Figure 4a). In the wavelength domain, two or more incoming signals, each on an arbitrary wavelength and IIF, are forwarded over the same outgoing wavelength and OIF. This form of forwarding is much more complex as it requires some form of optical buffering (to deal with output contention) and, possibly, wavelength conversion (see Figure 4b). Nevertheless, optical buffering is still at very early stages of investigation and, therefore, will not be available in the short term.



(b) Space domain (a) Space and wavelength domain

Figure 4 - Multipoint-to-point forwarding example

For technological (e.g., tuneable receivers get slower as the number of wavelengths increases) or economic (e.g., tuneable lasers are too expensive) reasons, a WS may have either none, limited or full capability with respect to transmission, reception, light splitting and wavelength conversion. Clearly, this may constrain the multicast capability of a network as some WSs might be left out of an optical multicast tree even though they are part of the corresponding IP multicast tree.

Due to technological limitations, economic reasons and the continuous evolution of optical networking and WDM technologies, WDM networks are likely to consist of heterogeneous WS architectures. This heterogeneity, along with the distributed nature of PIM-SM, make the construction of efficient WDM multicast trees somewhat more difficult.

5 Protocol description

We now describe a protocol to construct WDM multicast trees in all-optical lambda-switched networks running PIM-SM routing. Taking into account the issues discussed in section 4 and aiming at network robustness, neither relies the protocol on centralised information nor is the protocol constrained to any specific WS architecture.

Based on the MPLS architecture, the protocol uses MPLS signalling to map a multicast tree constructed by PIM-SM onto LSPs that are connected to form an equivalent tree at the WDM layer (the reader is referred to [9] for further information on the support of multicast in MPLS).

Due to the heterogeneity of WS architectures and the effects it causes, to request a LSP a WS needs to have information about the capability of each other WS along that LSP. Embedding this information in IP (control) routing messages, which are exchanged periodically among adjacent IP routers, is an alternative that is under investigation within the IETF [10]. A drawback of this approach is the potential inaccuracy of the locally stored information, which may increase the chances of a LSP set-up request to fail, depending on the difference between the time that that information was last updated and the time at which a LSP set-up request took place.

Our protocol relies on a mechanism that gathers information about the capability of each WS along a LSP and, consequently, provides a WS with sufficient information to form successful LSP set-up requests. The information gathering process follows the receiver-source direction. The actual LSP installation process follows the opposite direction. Therefore, the whole LSP set-up process takes approximately one round trip delay (RTD).

In the following discussions we assume that the network supports only layer-two forwarding. That is, we assume that there is always at least one (unicast) LSP set-up between any two nodes. The reader is referred to [7] for further information on MPLS forwarding capabilities. We also assume that MIBs are accessible to WSs' controllers.

5.1 WDM information collecting mechanism

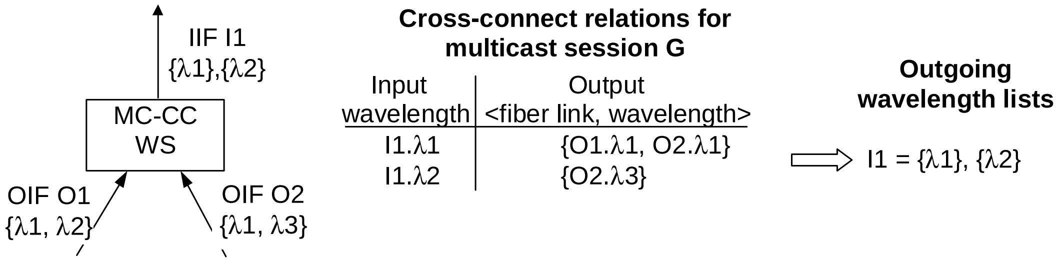
The mechanism used in the protocol to collect information about the capabilities of WSs is derived from the mechanism proposed in [4]. The main difference between them is the direction of operation. The mechanism works as follows.

A receiver-designated WS (R-DWS) willing to join a tree defines a list of wavelengths from which it can receive (plus, possibly, some transmission-specific quality of service requirements, such as bit error rate and signal-to-noise ratio). Based on some criteria (e.g., destination contention probability) defined by that WS, the list is ordered to indicate the WS's wavelength preference. The WS then forwards the list upstream towards the root of the tree that the WS is willing to join.

At each WS along the path, the list of wavelengths is updated so as to describe all the input wavelengths that can be resolved onto the wavelengths contained in the received list. If the outgoing list of wavelengths is empty, the collecting process is aborted. Otherwise, the cross-connect relations between the new list of wavelengths and the old list of wavelengths are stored locally. The WS refers to this information when requested to set up a LSP. If the WS does not receive a corresponding LSP request message within a pre-defined period of time, the cross-connect relation information is deleted.

The mechanism stops when the list of wavelengths reaches either the RP, a SPT's source-designated WS (S-DWS) or an intermediate WS (I-WS) that is already part of the tree and can forward the desired traffic over one of the wavelengths contained in the list.

Figure 5 depicts how the collecting mechanism works. Let us assume that two lists of wavelengths related to a given session, say G, arrive simultaneously at the WS, each over a distinct interface. The list arriving over the interface O1 contains wavelengths 1 and 2. The list arriving over the interface O2 contains wavelengths 1 and 3. Let us assume that the resources available at the WS allow only for i) the splitting of an incoming light signal on wavelength 1 into both OIF O1 and OIF O2 and ii) the forwarding of an incoming signal on wavelength 2 over the OIF O2 using wavelength 3.



Input wavelength	Output <fiber link, wavelength>
I1.λ1	{O1.λ1, O2.λ1}
I1.λ2	{O2.λ3}

Figure 5 - Collecting mechanism example

By matching the two lists with its resource management module, the WS finds out two resulting lists. The OIF O1's list is updated so as to contain the wavelength 1. The OIF O2's list is updated so as to contain the wavelength 2. The WS stores these cross-connection relations for future use and sets an expiration timer.

Note that outgoing lists of wavelengths of the same group are not merged into a single one. The reason for this is the support of RPT-SPT switchover. If the RP, for instance, merges two LSPs into a single LSP, it is not possible for a WS to prune traffic coming from a given S-DWS because that WS can not determine where the traffic comes from.

5.2 Protocol messages

The protocol consists of four messages (see Table 1). An ADVERT message informs an upstream WS of the intention of the message's sender to be included in a tree. An ADVERT message contains the following fields:

- FEC: identifies a group and, possibly, a given source; FEC is expressed in either forms: (*, G) or (S, G), whereas S is the source's address and G is the group's address;
- Root: indicates the LSP to be traversed by the message; it contains the LSP's label to either the RP or the S-DWS;
- Label: label to be used by incoming traffic of the defined FEC; this label is updated at each WS along the LSP towards the Root of the tree, according to the MPLS architecture;

- LAMBDASET: contains a list of wavelengths and, possibly, some transmission characteristics of each wavelength in the list; the list of wavelengths in a LAMBDASET is updated at each WS.

A SETUP-REQ message is a request for a downstream WS to update its LFIB accordingly, so as to complete the set-up of a point-to-point LSP from the message's sender to that WS. A SETUP-REQ follows the reverse path of a corresponding ADVERT message. The following fields are included in a SETUP-REQ message:

- Label: the label contained in the field Label of the received ADVERT message;
- Wavelength: the wavelength to support the LSP.

An ERROR message informs an upstream WS that a SETUP-REQ message sent by that WS has failed. The following fields are included in an ERROR message:

- Label: the label contained in the field Label of the received SETUP-REQ message;
- Wavelength: the wavelength contained in the Wavelength field of the received SETUP-REQ message;
- LAMBDASET: as defined in the ADVERT message.

A TEARDOWN-REQ message is a request for an upstream WS to remove the message's sender from the tree. This message contains the following fields:

- Label: the LSP's label on which the group G's traffic arrives;
- Wavelength: the wavelength on which the LSP has been set-up.

Table 1 - Protocol messages description

Message	Parameters	Function
ADVERT	FEC, Root, Label, LAMBDASET	Advertises join interest and collects WDM information
SETUP-REQ	Label, Wavelength	Requests the set-up of a LSP
TEARDOWN-REQ	Label, Wavelength	Requests the removal of a LSP
ERROR	Label, Wavelength, LAMBDASET	Indicates a SETUP-REQ failure and returns an up-to-date LAMBDASET

Unlike PIM-SM, where join and prune requests are carried on a single JOIN/PRUNE message, join (i.e., SETUP-REQ) and prune (TEARDOWN-REQ) requests are carried on distinct messages in the protocol. The reason for this is that SETUP-REQ messages follow a source-receiver direction while TEARDOWN-REQ messages follow the opposite direction. Nevertheless, each message may carry one or more requests in order to minimise protocol overhead.

Different rules apply to each message depending on the role and position along a path of a WS. An ADVERT message can be issued by:

- A R-DWS that detects a new entry in its MIB;
- Either an I-WS or a RP in response to an incoming ADVERT message.

A SETUP-REQ message can be issued only in response to an incoming ADVERT message and by:

- Either a S-DWS or a RP;
- An I-WS that is part of the tree and can forward incoming traffic over one of the wavelengths defined in the received ADVERT message.

An ERROR message can be issued only by:

- Either an I-WS, a R-DWS or a RP in response to an incoming SETUP-REQ message that fails;
- Either an I-WS, a R-DWS or a RP in response to an incoming traffic over an unrecognised LSP.

A TEARDOWN-REQ message can be issued by:

- A R-DWS that detects the removal of the last entry of a particular group from its MIB;
- Either an I-WS or a RP if, after processing a TEARDOWN-REQ message, there is no other downstream WS connected;
- A R-DWS if a SPT from a S-DWS has been set-up.

5.3 Protocol behaviour

For the sake of clearness, we describe the behaviour of the protocol in three distinct steps: i) RPT set-up, ii) STP set-up from a S-DWS to a RP, and iii) RPT-SPT switchover. In any of these steps distinction is made regarding minor (non-externally visible) behavioural differences of the protocol. More specifically, in a circuit switching mode network, a LSP set-up request fails if the wavelength defined in the request is already being used by any other LSP on a given link. On the other hand, in a virtual circuit switching mode network, a LSP set-up request fails if the blocking probability caused by other LSPs sharing the same wavelength and the same link exceeds a certain threshold.

For the sake of clearness, each step is illustrated. A network operating in a circuit-switching mode is assumed in the illustrations. The network consists of two S-DWSs, both directly connected to the RP, an I-WS that is also directly connected to the RP and two R-DWSs, both directly connected to that I-WS. S-DWS A can transmit on wavelength 1. S-DWS B can transmit on wavelengths 1 and 2. I-WS C is fully capable of both light splitting and wavelength conversion. R-DWS D can receive on wavelengths 1 and 2. R-DWS E can receive on wavelengths 1 and 3.

Setting up a RPT

Each WS in the network keeps pooling its MIB for new entries or entries that have been removed. If a new entry is detected and (in this case) the R-DWS has at least one reception wavelength available, the R-DWS issues an ADVERT message over the LSP to the RP. A WS finds out the LSP's label by matching the RP address, which is obtained from the new entry, with its LFIB. The LSP's label is assigned to the message's Root field. The WS also binds a label to the message's FEC and assigns it to the message's Label field. FEC is filled with a wildcard * and the new entry's group address G. LAMBADSET is created with the list of the wavelengths from which the WS can receive. The content of ADVERT is stored locally.

At each subsequent WS along the LSP, the ADVERT message is processed as follows. The WS processing the message matches the content of ADVERT (i.e., the message's FEC) with its LFIB (we assume that each LFIB entry has its corresponding FEC information associated). If no match is found and the WS is the RP, the message is no longer processed. If either a match is not found and the WS is an I-WS or a match is found but the WS can not forward on any of the wavelengths described in the received LAMBDASET, the WS follows on by updating the LAMBDASET as described in section 5.1.

If the new LAMBDASET is empty, the message is no longer processed. If the new LAMBDASET is not empty, the WS proceeds as follows. If the WS is the RP then it will start the procedure to have SPTs from S-DWSs to itself installed. Such a process is described later when we describe the set-up of a SPT. If the WS is an I-WS then it binds a new label to the message's FEC and updates the Label field with this label. The cross-connect relation between the old label and the new label, the cross-connect relation between the received LAMBASET and the updated LAMBADSET and the message's FEC are stored locally. The Root's label is swapped accordingly and the message is forwarded over the LSP indicated by the Root's label.

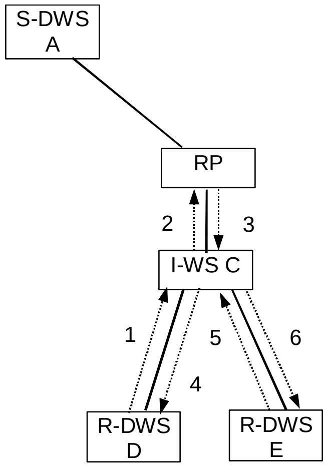
If a match between the content of the ADVERT message and a LFIB's entry is found and the WS can forward on one of the wavelengths defined in the received LAMBDASET, the WS initiates the LSP set-up request process. First, the WS updates its LFIB. The entry's outgoing label is updated with the label contained in the field label. The entry's OIF is updated with the ADVERT message's IIF. The entry's outgoing wavelength is updated with the wavelength that the WS selects from the LAMBDASET. Note that the order in which wavelengths are organised in the LAMBDASET does not mean that the WS processing the message must follow it. The WS is free to select any wavelength if that wavelength suits it more.

Once the WS's LFIB is updated accordingly, the WS sends a SETUP-REQ message over the entry's OIF. The SETUP-REQ's Label field is updated with the entry's outgoing label and the SETUP-REQ's Wavelength field is updated with the entry's outgoing wavelength.

Each subsequent WS processes the SETUP-REQ message as follows. Firstly, the label and the wavelength expressed in the message are matched with the cross-connect relations that have been previously stored. If a match is found, the WS selects one (output) wavelength from the matched entry's list of (output) wavelengths. A new LFIB entry is created linking the incoming label, the input wavelength, and the message's IIF to the outgoing label, the output wavelength and the OIF, as indicated in the matched saved relation. Based on the entry, the WS updates the SETUP-REQ with the outgoing label and the output wavelength and forwards the SETUP-REQ message over the proper OIF.

The SETUP-REQ flows down the RPT until it reaches either a WS that can not set-up the LSP or the WS that issued the ADVERT message. In the former case, the WS does not forward the SETUP-REQ. Instead, it sends an ERROR message back to the SETUP-REQ's sender, containing an updated LAMBADSET. A WS processing such a message releases the committed resources (i.e., deletes the corresponding LFIB entry) and updates the LAMBDASET accordingly. The SETUP-REQ's sender, upon reception of an ERROR message, issues a new SETUP-REQ message.

Figure 6 illustrates the join of R-DWS D and R-DWS E, at different times, into a RPT.



Messages description
1. ADVERT((*, G), LA1, {λ1, λ2})
2. ADVERT((*, G), LA2, {λ1})
3. SETUP-REQ(LA2, λ1)
4. SETUP-REQ(LA1, λ1)
5. ADVERT((*, G), LB1, {λ1, λ3})
6. SETUP-REQ(LB1, λ1)

Figure 6 - RPT set-up example

A WS assumes that a LSP has been successfully set-up if a corresponding ERROR message does not arrive within a certain period of time. The lose of an ERROR message means that the ERROR message's sender will receive traffic over a LSP that the WS does not recognises. A WS responds to this situation by discarding the traffic and sending another ERROR message.

A R-DWS knows that a LSP has been successfully set-up when (*, G) traffic arrives over the proper IIF, the proper input wavelength and with the proper input label. A R-DWS re-sends ADVERT messages until either a LSP is successfully set-up or a pre-defined retransmission threshold is reached.

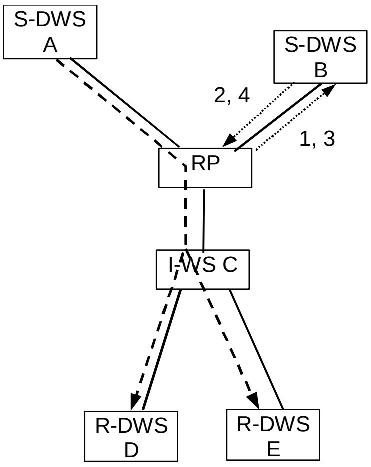
Setting up a SPT from a S-DWS to the RP

One SPT from any given S-DWS to a given RP is required as long as there is at least one R-DWS willing to receive traffic on the corresponding RPT (a RP knows when there is at least one R-DWS willing to receive traffic based on received ADVERT messages). Additional SPTs may be required each time a LFIB's entry's IIF and incoming wavelength cannot be resolved onto any of the wavelengths defined in a received ADVERT message. The behaviour of the protocol is the same in both cases.

A RP behaves as follows. First, it updates the LAMBDASET as usual. If the LAMBDASET is empty then the message is no longer processed. Otherwise, the RP issues an ADVERT message over the LSP to one or more S-DWSs, depending on the case. A WS obtains the LSP's label from the LFIB's matched entry. The LSP's label is assigned to the message's Root field. The WS also binds a label to the message's FEC and assigns it to the message's Label field. FEC is filled with the source address S and the new entry's group address G. The content of ADVERT is stored locally.

The ADVERT message flows upstream until it reaches either an I-WS that can forward on one of the wavelengths defined in the message or the S-DWS. Each I-WS processes the ADVERT message as described previously. A S-DWS replies with a SETUP-REQ message. After being processed by each I-WS along the path, as described previously, the message reaches the RP. The RP then updates its LFIB accordingly and discard the message.

Figure 7 illustrates the set-up of a SPT from S-DWS B to the RP. An important aspect that should be noticed from Figure 7, is that a RP does not merge two LPSs onto a single one in circuit switching mode networks. Merging two LSPs onto a single one makes it impossible for a node to distinguish where a traffic comes from and this is necessary if RPT-SPT switchover is to be supported. Merging of two input wavelengths onto a single output wavelength is possible in virtual circuit switching-like networks merging though. This is because in this case, labels can be used to distinguish traffic from different S-DWSs.



Messages description
1. ADVERT((*, G), LN, {λ2})
2. SETUP-REQ(LN, λ1)
3. ERROR(LN, λ1, {λ2})
4. SETUP-REQ(LN, λ2)

* S-DWS B's traffic will flow down the RP only after the RPT is refreshed

Figure 7 - Example of SPT set-up

Figure 8 illustrates how the WDM RPT will look like after it is refreshed.

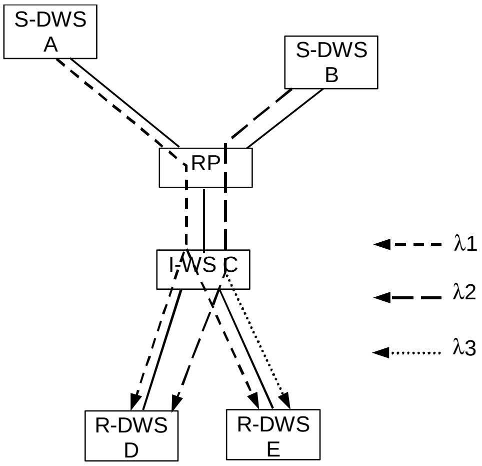


Figure 8 - Resulting RPT

RPT-SPT Switchover

The RPT-SPT switchover process can be divided into two phases: i) the set-up of a SPT from a S-DWS and ii) the prune of the RPT's corresponding LSP. The first phase works exactly as in the set-up of a SPT from a S-DWS to the RP. The second phase is triggered only if the ADVERT message's sender R-DWS detected the arrival of traffic over the requested SPT.

In this case, the R-DWS obtains the incoming label and the input wavelength corresponding to the RPT's LSP to be pruned and then deletes the corresponding LFIB's entry. This results in the issue of a TEARDOWN-REQ message updated with the obtained incoming label and the input wavelength towards the RP. The LSP to be followed by the message is obtained by matching the traffic's IIF coming from the RPT with the LFIB.

At each subsequent WS, the arrival of the TEARDOWN-REQ message leads to the deletion of the corresponding LFIB's entry. The message flows upstream until it reaches either the RP or an I-WS that branches the LSP to another WS. The RP may also forwards the message towards its S-DWS in case no other member of the RPT is interested in traffic coming from that S-DWS over the RPT.

We summarise the protocol by describing its finite state machines (FSMs). A suffix is used in front of each message to indicate the message's direction. FromD indicates that the message comes from the downstream WS. ToD indicates that the message was sent to the downstream WS. FromU indicates that the message comes from the upstream WS. Finally, ToU indicates that the message was sent to the upstream WS.

WSs have different behaviours depending on their roles regarding a particular FEC at a particular moment. We define two roles, namely, upstream and downstream, where a WS playing either one or another role may or may not be an intermediate node.

A node in the upstream role may be in either of the following states:

- DISCARD: a node in this state does not forward incoming traffic but discard it instead;
- FORWARD: a node in this state forwards incoming traffic.

A WS is initially in the DISCARD state. Four events can happen in this state: FromD_ADVERT, ToU_ADVERT, ToD_SETUP-REQ, FromU_Data. A WS in the DISCARD state stays in this state if FromD_ADVERT, ToD_ADVERT or FromU_Data occurs. If ToD_SETUP-REQ occurs then the WS goes to the FORWARD state.

Four events can happen in the FORWARD state: FromU_Data, ToD_Data, FromD_ERROR, FromD_TEARDOWN-REQ. A WS in the FORWARD state stays in this state if either FromU_Data or ToD_Data occurs. If either FromD_ERROR or FromD_TEARDOWN-REQ occurs then the WS goes back to the initial state.

Figure 9 depicts the FSM of a WS in the upstream role.

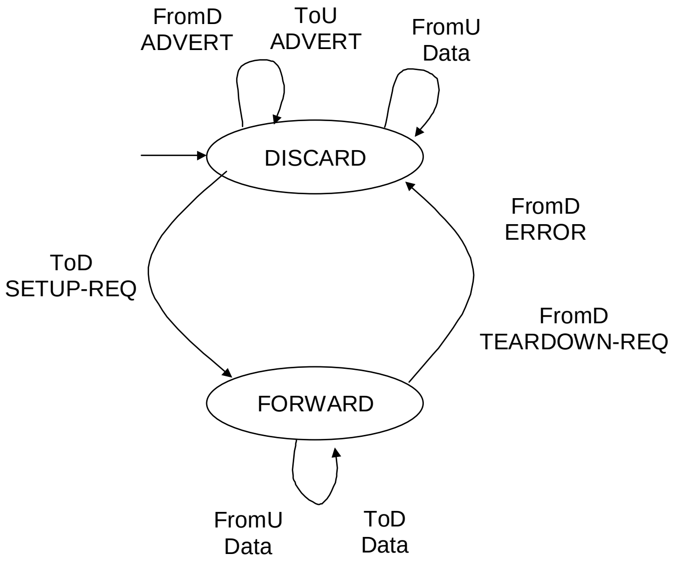


Figure 9 - FSM of a WS in the upstream role

Figure 10 depicts the FSM of a WS in the downstream role. A WS in the downstream role may be in either of the following states:

- DISCONNECTED: a node in this state is not part of the tree and, therefore, does not receive the corresponding traffic;
- CONNECTED: a node in this state is part of the tree and, therefore, may receive the corresponding traffic.

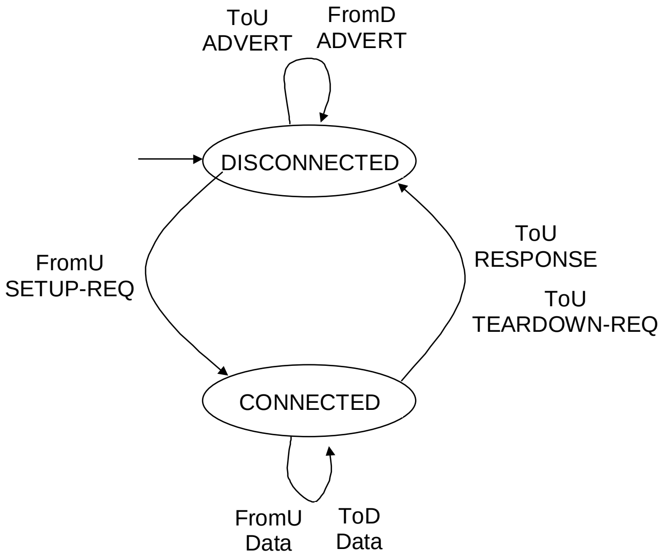


Figure 10 - FSM of a WS in the downstream role

A WS is initially in the DISCONNECTED state. Two events can happen in this state: FromD_ADVERT, ToU_ADVERT, FromU_SETUP-REQ. A WS in the DISCARD state stays in this state if either FromD_ADVERT, ToU_ADVERT occurs. If FromU_SETUP-REQ occurs then the WS goes to the CONNECTED state.

Four events can happen in the CONNECTED state: FromU_Data, ToD_Data, ToU_ERROR, ToU_TEARDOWN-REQ. A WS in the CONNECTED state stays in this state if either FromU_Data or ToD_Data occurs. If either ToU_ERROR or ToU_TEARDOWN-REQ occurs then the WS goes back to the initial state.

6 Final Remarks

We described a protocol to support PIM-SM in all-optical (generic) lambda-switched networks. The protocol uses MPLS signaling to map a multicast tree constructed by PIM-SM into LSPs that are connected to form an equivalent multicast tree at the WDM layer.

Despite being simple, the protocol offers some other desired features. Firstly, no centralized information is required whatsoever. This minimizes point of failures. Secondly, due to its information gathering mechanism, the protocol is not fixed to any specific WS architecture. This is an important feature since WDM and all-optical networking technologies are constantly evolving. Thirdly, WDM multicast trees can be constructed based on QoS parameters defined by receivers, even though the state-of-the-art of WDM and all-optical networking technologies have not tackled this issue yet. As a matter of fact, neither has PIM-SM nor any other IP multicast protocol.

There are still some open issues that need to be solved. Simulation activities will have to be carried out to, for instance, assess the blocking probability of competing SETUP-REQ messages.

Acknowledgements

This research is supported by the Dutch Technology Foundation STW, applied science division of NOW and the technology programme of the Ministry of Economic Affairs of the Netherlands.

References

[1] R. Malli, X. Zhang and C. Qiao: Benefit of Multicasting in All-Optical Networks, Proc. of SPIE All-Optical Networking'98: Architecture, Control, and Management Issues, (October 1998), Vol. 3531, pp. 209-220.

[2] C. Qiao, M. Jeong, A. Guha, X. Zhang and J. Wei: WDM Multicasting in IP over WDM Networks, Proc. of IEEE International Conference on Network Protocols (ICNP), (Toronto, Canada, October 1999), pp. 89-96.

[3] X. Zhang, J. Wei and C. Qiao: On Fundamental Issues in IP over WDM Multicast, Proc. of IEEE IC3N'99, (Boston, USA, October 1999), pp. 84-90.

[4] M. R. Salvador, S. H. de Groot and D. Dey: Supporting IP Dense Mode Multicast Routing in All-Optical WDM Networks, Proc. of the 1st SPIE/IEEE/ACM International Conference on Optical Networks and Communications (OPTICOMM), (Dallas, USA, October 2000), pp. 167-178.

[5] L Wei et al.: Protocol Independent Multicast-Sparse Mode (PIM-SM): Protocol Specification, Internet Draft <draft-ietf-pim-v2-sm-01.txt>, IETF, November 1999.

[6] D. O. Awduche, Y. Rekhter, J. Drake and R. Coltun: Multi-Protocol Lambda Switching: Combining MPLS Traffic Engineering Control with Optical Crossconnects, Internet Draft <draft-awduche-mpls-te-optical-01.txt>, IETF, November 1999.

[7] E. C. Rosen, A. Viswanathan and R. Callon: Multiprotocol Label Switching Architecture, Internet Draft <draft-ietf-mpls-arch-05.txt>, IETF, April 1999.

[8] B. Mukherjee.: Optical Communication Networks, McGraw-Hill Series on Computer Communications, 1997.

[9] D. Ooms et al.: Framework for IP Multicast in MPLS, Internet Draft <draft-ietf-mpls-multicast-00.txt>, IETF, June 1999.

[10] G. Wang et al.: Extensions to OSPF/IS-IS for Optical Routing, Internet Draft <draft-wang-ospf-isis-lambda-te-routing-00.txt>, IETF, March 2000.