INTERNATIONAL JOURNAL OF COMMUNICATION SYSTEMS
Int. J. Commun. Syst. 2002; **15**:573–592 (DOI: 10.1002/dac.552)

Recovery from control plane failures in GMPLS-controlled optical networks

Jing Wu[1,*,†], Delfin Y. Montuno[2], Hussein T. Mouftah[3], Guoqiang Wang[2] and Abel C. Dasylva[2]

[1] *Communications Research Centre Canada, 3701 Carling Avenue, Ottawa, Ont., Canada K2H 8S2*
[2] *Nortel Networks, P.O. Box 3511, Stn. C, Ottawa, Ont., Canada K1Y 4H7*
[3] *Queen's University, Dept. of Electrical and Computer Eng., Kingston, Ont., Canada K7L 3N6*

SUMMARY

The health status of the control plane and the data plane of a GMPLS-controlled optical network is independent in the physically separated control network implementation. In most control plane designs, besides the topology information, the entities of the routing protocol only record the number of available wavelengths on each link. However, the status of each wavelength is maintained by the entities of the signalling protocol. Without recovery ability of the signalling protocol CR-LDP, a failure in the control plane will result in the permanent loss of the status information of wavelengths. A mechanism to recover the status information of the wavelengths is proposed. A downstream node maintains a label information database (LID) about assignable (free) labels in each incoming link. A copy of LID is redundantly stored in the upstream node as a label information mirror (LIM). A systematic procedure is proposed to synchronize the contents of a LIM and the corresponding LID. The initialization of a new LDP session with the enhanced recovery mechanism will guarantee the revival of the status information of wavelengths. It can recover multiple control channel failures, but it only applies to single node failure among any pair of adjacent nodes. © Crown Copyright 2002. Reproduced with the permission of Her Majesty's Stationery Office. Published by John Wiley & Sons Ltd.

KEY WORDS: recovery method; control plane reliability; control signalling protocols; generalized multiprotocol label switching; optical networks

1. INTRODUCTION

Multi-protocol label switching (MPLS) has been traditionally used in IP networks for traffic engineering [1,2]. With the development of generalized MPLS (GMPLS) [3,4], the next-generation MPLS has been driven beyond data networks into every corner of the high-performance Internet. The extensive applications of GMPLS provide unified control throughout the whole network, from the optical backbone through core IP and wireless networks [5]. GMPLS addresses challenges such as service provisioning, network evolution and network efficiency. This enables GMPLS to move towards a complete, end-to-end approach of network control.

*Correspondence to: Jing Wu, Communications Research Centre Canada, 3701 Carling Avenue, Ottawa, Ont., Canada K2H 8S2.
†E-mail: jingwu@ieee.org

Published online 27 June 2002

Received 1 September 2001
Revised 26 March 2002
Accepted 24 April 2002

One of the key technologies in GMPLS is to use the signalling protocol of MPLS to control circuit switched connections. Therefore, the signalling protocol of MPLS acts as the unified control signalling protocol for heterogeneous networks encompassing time-division, wavelength and spatial switching [4,6].

The most important and successful application of GMPLS is to control optical networks. In wavelength division multiplexing (WDM) optical networks, the control objects are wavelengths or wavebands [7]. In synchronous optical networks (SONET), time division multiplexing (TDM) time slots at different granularities are controlled [7]. In this paper, we will use WDM networks as examples of optical networks to discuss our proposal. However, the approach proposed is equally applicable to any GMPLS-controlled networks including SONET.

2. THE PROBLEM

2.1. Physical implementations of the control plane of GMPLS-controlled optical networks

Physically, the control plane of a GMPLS-controlled optical network could be implemented in three ways: in-band control channels, separate supervisory channels (also known as out-of-band in-fibre control plane), and physically separated control network (also known as out-of-band out-of-fibre control plane) [8,9]. Typical in-band control channels include frame header bytes (e.g. in SONET/SDH, Digital Wrapper), sub-channel modulation, etc. [10]. Separate supervisory channels use dedicated lightpaths, which most likely go together with the controlled lightpaths in the data plane [8]. An example of physically separated control network is that the control plane runs over an IP/Ethernet network, while the data plane runs over a wavelength routed WDM network (Figure 1) [11]. We will use terminology *control channels* referring to communication media to convey messages among control nodes, no matter they are in-band or out-of-band.

2.2. Reliability issue of the control plane of GMPLS-controlled optical networks

The health status of the control plane and the data plane is more or less independent in the physically separated control network implementation [3,8]. This feature raises the concerns of reliability issue of the control plane [12].

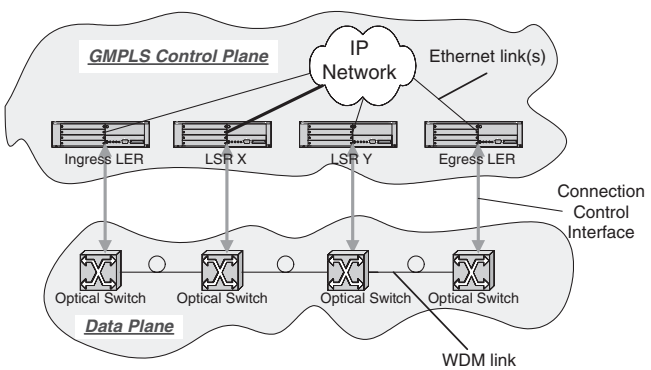

Figure 1. A GMPLS-controlled optical network.

Table I. Key contributors to IP network failures [13].

Outage category	Number of occurrances	Percentage
Maintenance	272	16.2
Power outage	270	16.0
Fibre cut or circuit/carrier problem	261	15.3
Unreachable	215	12.6
Hardware problem	154	9.0
Interface down	105	6.2
Routing problems	104	6.1
Miscellaneous	86	5.9
Unknown/undetermined/no problem	82	5.6
Congestion/sluggish	65	4.6
Malicious attack	26	1.5
Software problem	23	1.3

The control plane of GMPLS-controlled optical networks is a specially designed IP network. Usually only control and management traffic run over it [11]. Although the statistics about the operation of this IP network is not widely available yet, the operations of conventional IP router-based networks can be used as a reference.

Some experimental investigations on the availability of IP networks were conducted for both inter-domain and intra-domain cases [13,14]. Compared to over 99.999% availability of legacy telephony networks [15], the availability of IP networks is much less, because the stability of end-to-end Internet paths is dependent both on the underlying telecommunication switching system, as well as the higher level software and hardware components specific to the Internet's packet-switched forwarding, name resolution and routing architecture. Laboviz *et al.* [13] did an experimental measurement on a deployed wide area IP network and found that the majority of Internet backbone paths exhibit a mean-time to failure of 25 days or less, and a mean-time to repair of 20 min or less.

Table I shows the network failures during their one-year case study (November 1997–1998) based on the real operational trouble logs of a regional Internet service provider [13]. In the table, maintenance refers to either a scheduled, or unscheduled emergency upgrade of software or hardware, or router/switch configuration changes. A power outage includes either a loss of power to a router, or a power failure in a transport network facility which has an impact on IP links. Unreachable includes intermittent failures which mysteriously resolve themselves before an engineer investigates the outages. These unreachable outages usually result from the maintenance or failures of the transport network. A hardware problem includes a router, switch or power supply failure. A routing problem designation reflects errors with the configuration or interaction of routing protocols. Most routing problems stem from human error and mis-configuration of equipment.

It is well accepted that no single backbone, or snapshot of the Internet provides a valid representation of the heterogeneous and rapidly changing Internet [13]. However, previous studies are helpful to understand the reliability of IP networks. Moreover, after several years' development since the experiments, the quality of hardware and software has been improved and the operation experience has been accumulated. But there is still a long way for the IP network to achieve comparative availability of legacy transport network, i.e. 99.999%

availability or equivalently less than 5 min downtime/year. The challenge here we face is to build highly reliable optical networks controlled by relatively less reliable IP networks.

2.3. Control information dissemination and management

In the control plane of a GMPLS-controlled optical network, the routing and control signalling are two of basic components. Both functions are built on the extensions of IP/MPLS protocols [11,16]. The routing protocol collects information about the network topology, node/link resource availability and provides information for the computation of candidate routes from ingress optical switches to egress optical switches [3]. The signalling protocol is used to set up, maintain, and tear down lightpaths. There are two major control signalling protocols proposed in the Internet Engineering Task Force (IETF) for the GMPLS-controlled optical networks, namely Resource Reservation Protocol with Traffic Engineering extensions (RSVP-TE) and constraint-based label distribution protocol (CR-LDP) [16].

In most implementations, the information management in the control plane is jointly done by both the entities of the routing protocol and the entities of the signalling protocol. Besides the topology information, which describes the connectivity relations of optical switches, the entities of the routing protocol only record the number of available wavelengths on each link. However, the operating status of each wavelength is maintained by the entities of the signalling protocol [3]. There are several fundamental reasons behind this design. The first reason comes from the scalability concern. The routing protocol works based on the periodical advertising of each node's local view of the global topology and node/link resource, and updating each node's local database according to the advertisement of its adjacent neighbours. In order to make the GMPLS control plane scalable, the size of each link state advertisement in the routing protocol should be as small as possible. So the status of wavelengths in a link has to be abstracted. For example, it would be difficult for the routing protocol to trace the status of each wavelength in each WDM link, which may contain more than 100 wavelengths [17]. The second reason relates to the nature of slow convergence of the routing protocol. Even if the routing protocol kept the status of each wavelength on each link, it would not be accurate enough to set up lightpaths.

When a new lightpath is requested from an ingress optical switch to an egress, the route calculation module, which works based on the information maintained by the routing entity, provides a route with available wavelength on each link. The signalling protocol will actually reserve and activate a wavelength channel on each link and thus set up the whole end-to-end lightpath [10]. After the convergence of the routing information, each routing entity will update its knowledge about the number of available wavelengths in each link.

2.4. Importance of the control plane recovery

The routing protocols are fairly fault tolerant. They exchange information through periodical link state advertisement [3,10]. If some failures happen in either control nodes or control channels, they can still recover their routing information after the faults are fixed and the periodical link state advertisement resumes.

Unfortunately, as a major control signalling protocol for GMPLS-controlled optical networks, CR-LDP is vulnerable to hardware and software failures [18,19]. This compares to the fault tolerance of RSVP-TE, which also uses periodical state refreshment and relies on raw IP or UDP instead of TCP. But RSVP-TE has inherent scalability problem. Its periodical state refreshment would cause a huge number of messages in the control plane for a large network. So

CR-LDP still remains one of the most important control signalling protocols for GMPLS-controlled optical networks.

Pulley *et al.* [20] observed that because of the dependence of CR-LDP on TCP, the GMPLS specification requires that all lightpaths associated with a particular session must be destroyed if the TCP session is terminated or fails. They pointed out the necessity of CR-LDP recovery mechanism. Potentially a large number of lightpaths might have been established between two optical switches before the failure. Without CR-LDP recovery mechanism the impact to the network in re-establishing all affected lightpaths would be substantial. There could be a 'signalling storm' when all the affected lightpaths try to be re-established at the same time.

When a fault occurs in the control plane, the default operations of current CR-LDP will discard all the information about the established lightpaths controlled by the affected LDP sessions. The consequence of the permanent loss of status information of channels is critical. Firstly, after the fault is fixed and new LDP sessions are set up, newly established lightpaths could falsely use the in-use wavelengths and interrupt the user communications established before the failure. Secondly, the established lightpaths will either be terminated or have to work in the degraded state without sufficient control. These lightpaths might not be able to be protected against the data plane failures such as fibre cuts. Even after the fault is fixed, they still cannot return to normal operating state [4].

Till now, almost all the protection and restoration mechanisms developed for the GMPLS-controlled optical networks assume the control plane is reliable, and focus on how to handle faults in the data plane [21–23]. But without the recovery mechanism of CR-LDP, it is hard to maintain the established user communications in the data plane in the events of control plane failures.

2.5. Overview of CR-LDP

In MPLS-enabled IP networks, LDP is responsible for the messaging among label switching routers (LSRs) to control label switched paths (LSPs) [24]. CR-LDP is an extension of LDP by using only downstream on demand label distribution mode and adding some other restrictions and extensions [6]. In GMPLS-controlled optical networks, an LSR represents an optical switch or its control node depending on which plane is referred to. An LSP represents a lightpath that bears end-to-end user communications.

Each pair of adjacent control nodes run an LDP session to exchange messages controlling all the lightpaths between the corresponding optical switches. Each side of an LDP session uses an LDP entity, which is a process of software together with a set of state variables and timers.

Within LDP, there are four categories of messages [24]: (1) *Discovery messages*, used to announce and maintain the presence of a control node; (2) *Session messages*, used to establish, maintain, and terminate sessions between LDP peers; (3) *Advertisement messages*, used to setup and teardown lightpaths; (4) *Notification messages*, used to provide advisory information and to signal error information. The operations of LDP are illustrated in Figure 2.

2.6. Default behaviours of LDP when a control plane failure occurs

Figure 3 illustrates LDP sessions and an LSP from the ingress LER to the egress LER through two LSRs. In the normal state, there are three LDP sessions running between the ingress LER and LSR X, LSR X and LSR Y, LSR Y and the egress LER, respectively. LDP specification assumes an LSP is uni-directional based on the fact that most IP channels are one-way [4,24].

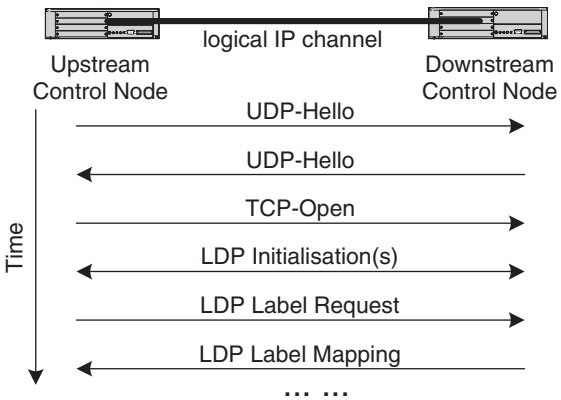

Figure 2. LDP operations.

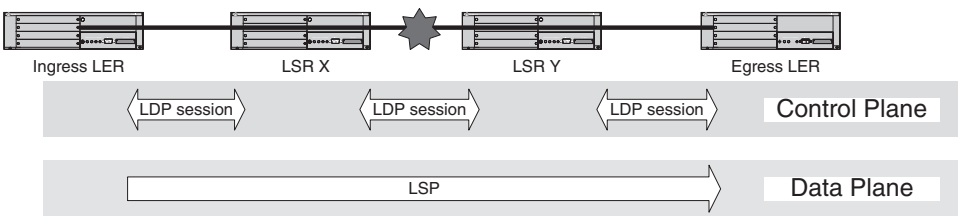

Figure 3. LDP sessions and an LSP in an example network.

Although the concept of bi-directional LSP is later introduced to match two-way communication channels in telecommunication systems [4,17], logically we can still treat bi-directional LSP as a pair of uni-directional LSPs. Thus, all the discussion is also equally applicable to bi-directional LSPs although this paper only considers uni-directional LSPs. The following list summarises the major steps of the default LDP response to a control channel failure. (Refer to Reference [24, Section 2.5.6, Maintaining LDP sessions] for more details.)

1. LDP entities in LSR X and LSR Y detect the failure in a variety of ways.
   - Indication from the management entity that a TCP connection or underlying resource is no longer active.
   - Notification from a hardware management entity of an interface failure.
   - Socket keepalive timeout.
   - Socket sending failure.
   - New (incoming) socket opened.
   - LDP keepalive timeout.
   - Other mechanisms.
2. LDP entities in LSR X and LSR Y report to the management entity about the failure for maintenance purpose.

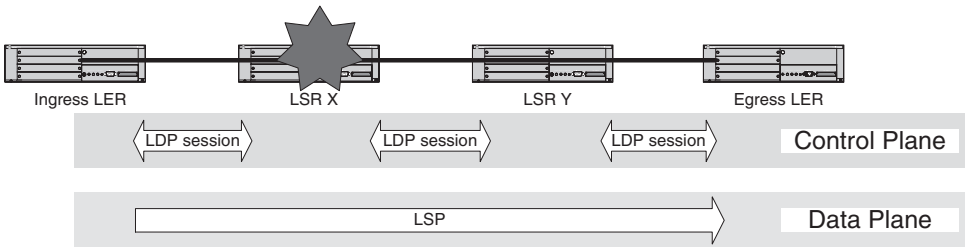

Figure 4. A control node failure in the example network.

3. LDP entities in LSR X and LSR Y clear up the affected LDP session.
   - Delete the hello adjacency associated with the LDP session.
   - Terminate the LDP session.
   - Release all labels and resources associated with the LDP session.
   - Close the transport connection (TCP connection).
4. The management entity co-ordinates the actions of other LDP sessions associated with the affected LSP.

   - The restoration might be performed according to the management policy. For example, if the path protection is used for the LSP, the ingress LER will be instructed to set up a new backup LSP between the same pair of ingress and egress LERs. As soon as the backup LSP is set up, the user traffic is detoured from the failed LSP to the backup LSP.
   - The management entity could decide to release the resources occupied by the failed LSP. In the previous example, the channels between the ingress LER and LSR X, and between LSR Y and the egress LER used by the failed LSP could be forced to release.

Now, let us see the default LDP response to a control node failure as shown in Figure 4.

1. LDP entities in the ingress LER and LSR Y detect the failure of their peer LSR X.
2. LDP entities having detected the failure or experiencing failures report to the management entity about the failure.
3. LDP entity in LSR X may have to further clear up its LDP sessions depending on the nature of the failure. For example, if LSR X encounters a TCP socket failure, it still needs to clear up some LDP related information and resources allocated for the LDP entity.
4. LDP entities in the ingress LER and LSR Y clear up their affected LDP sessions.
5. The management entity co-ordinates the actions of other LDP sessions associated with the affected LSP.

   - Restoration might be performed.
   - LSR Y and the egress LER could decide to release the channels used by the failed LSP.

In summary, without any sufficient scheme, LDP entity clears up all information about the failed LDP session when a control plane failure occurs, causing all LDP sessions to be re-initialized from scratch. The management entity will most likely release all the affected LSPs, because those LSPs are partly out of service or without sufficient control. Unfortunately, the re-initialization of any LDP session without recovery ability may affect previously established LSPs not affected by the failure.

2.7. Other existing solutions

Xu *et al.* [25] mentioned the importance of handling control plane failure and LDP recovery. It was suggested that notification messages should be extended not only for data plane failures but also for control plane failures, and some recovery mechanism should also be introduced in the control plane. They expected that a control node would need to consult with its neighbours to synchronize control channel state information and current lightpaths status in order to successfully recover the LDP. Unfortunately the detailed recovery mechanism was left for further study.

Farrel *et al.* proposed a fault tolerant mechanism for LDP [26]. A procedure was proposed to manage the re-sending of LDP messages and re-synchronizing of LDP session status between two LDP entities affected by a TCP failure. It is controlled by a set of timers. This enables LDP to be able to recover from control channel failures or TCP software failures. However, the coverage is limited to failures that do not affect LDP entities. So it cannot recover from most control node failures, which result in partial or complete loss of LDP session status and information.

The link management protocol (LMP) suggests improving the reliability of the control plane by using backup control channel(s) [8,27]. From the protocol stack point of view, using backup control channel is a method of protection against control channel failures at the IP layer or the data link layer. However, in some applications, backup control channels are not available. Even if backup control channels are provisioned, the TCP layer and the CR-LDP layer could possibly sense the failures in the primary control channels and take actions. In addition, the recovery from control node failures might need more sophisticated co-ordination among redundant processors. So even though providing backup control channels may be the easiest step to improve the reliability of the control plane, it is not sufficient in some scenarios. The recovery mechanism of CR-LDP itself is still necessary.

Another possible method is to let the control plane enquire the status of wavelengths to the data plane. However this poses an additional requirement on the data plane. The data plane would have to add the intelligence to maintain the status information of each wavelength. The synchronization of the information redundantly stored in both planes has to be considered. It is not a favourable system design to assign extra control-related tasks to the data plane.

3. DISTRIBUTED RECOVERY METHOD

There are several contributions in the proposed recovery method: semantics of label information, redundant storage of label information, synchronization procedure of the label information redundantly stored in an upstream control node and a downstream control node, recovery procedure during the initialization of an LDP session [28].

3.1. Semantics of label information

CR-LDP runs in the downstream on demand label distribution mode. When an LSR initiates the setup of a user communication channel to another LSR, the LSR in the upstream side (with respect to the direction of the LSP) explicitly requests from the LSR in the downstream side for a label. This explicit request for a label is implemented in LDP as an LDP label request message.

Then the downstream LSR retrieves the information about the available channels (labels) for that incoming link. If the channels are available and the policy allows, the downstream LSR reserves a channel and assigns a label. In response to the LDP label request message, the downstream LSR sends back an LDP label mapping message to the upstream LSR. After the upstream LSR receives the LDP label mapping message, it can start using the LSP with the indicated label [6].

Normally, there are two kinds of LSP teardown procedures, namely ingress-initiated and egress-initiated. When the ingress LER wants to tear down an established LSP, it sends an LDP label release message to the downstream LSR and stops using the LSP. Then the downstream LSR updates its information about the available channels (labels) for that incoming link. This procedure is repeated by each LDP session along the LSP. In the egress-initiated teardown, the egress LER sends an LDP label withdraw message to its upstream peer LSR. If the upstream LSR decides to tear down that LSP, it sends back an LDP label release message and stops using that LSP. Upon receiving that LDP label release message, the egress LER updates its information about the available resources (labels) for that incoming link and also stops using that LSP. Each LDP session repeats this procedure in the opposite direction of the LSP.

So we can conclude that in the downstream on demand mode, the downstream node maintains the label information for the link. Figure 5 shows this fact under the current definition of LDP and CR-LDP.

In general, the label information includes the label space (usable labels), the status of each usable label. The label space represents the channels in the data plane. The label space can be statically or dynamically configured. The static configuration of the label space of a link involves both the upstream node and the downstream node. It has to be consistent with the actual channel connectivity of the link. The dynamic configuration is achieved via LDP notification messages or other protocols, e.g. the Link Management Protocol. The status of a usable label could be assignable (free), in-use, or reserved (a transit state after receiving a label request and before replying a label mapping). The LDP standard does not specify the semantic meaning of label information, and leaves it to the implementation. There are two possible implementations. One is to manage a label space plus a database of labels in-use or reserved. The other is only to manage a database of assignable labels, which are usable labels excluding labels in-use or reserved. In our proposal, the latter semantics is used, which leads to an efficient recovery of lost information in databases as will be shown.

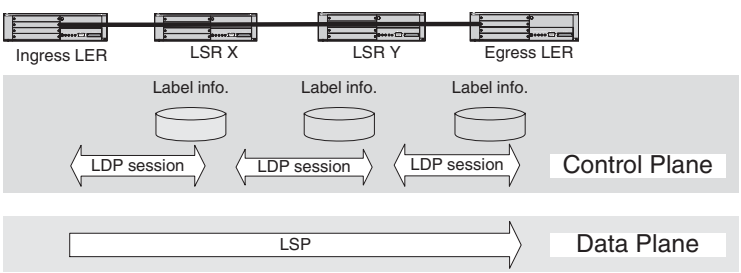

Figure 5. Downstream node manages the status information of a link in the downstream on demand mode of LDP.

3.2. Synchronization procedure of a label information mirror and a label information database

The proposed recovery mechanism of LDP is based on introducing label information mirrors (LIMs) in the upstream LSRs (Figure 6). Each LIM is a copy of the label information database (LID) in the downstream LSR of an LDP session. Because labels only have local meanings regarding the link they refer to, so both a LID and a LIM only store the information about labels regarding a specific link and has no global significance. This makes the recovery mechanism scalable and enables it to be able to be deployed on a per LDP session basis. In this section, we will discuss how to synchronize the contents of a LIM to its corresponding LID.

During the 'cold' initialization of an LDP session (initialization from scratch), a LIM is initialized exactly identical to the corresponding LID, and both are consistent with the actual channel configuration in the data plane. (We will discuss more details about the initialization procedure shortly afterwards.) When an LSR requests a label, it still works as LDP and CR-LDP standards specified in the downstream on demand mode. In addition to the regular procedure, the upstream LSR updates its LIM when it receives the LDP label mapping message from its downstream peer LSR. So both the LIM and the LID are synchronized after the LSP establishment phase, i.e. their contents are identical. In the LSP termination phase, besides the regular procedure, the upstream LSR updates its LIM when it sends the LDP label release message to the downstream LSR peer. In this way, both the LIM and the LID are synchronized after the LSP termination phase. In the dynamic configuration of the label space, an LSR also maintains the LIM or the LID when it receives LDP notification message from its peer LSR about newly available/unavailable channels in the data plane. To conclude, in any stable state of the LDP operations the LIM and the LID are synchronized. Therefore, the label information is also stored in the upstream LSR for each LDP session compared to being only stored in the downstream LSR as in the standard CR-LDP.

3.3. LDP session recovery procedure

In the LDP session initialization, two new type-length-value objects (TLVs) are added into the LDP session initialization message: LIM TLV, and LID TLV. The function of LIM TLV is to notify the downstream LSR peer about the contents of a LIM. LID TLV is to notify the upstream LSR peer about the contents of a LID.

A special flag should be maintained for each LID and LIM to indicate the integrity of the label information. When an LDP session is initialized from scratch, the flag is reset by default. After a successful initialization, the flag is set. When an LSR is reset and re-initialized ('warm'

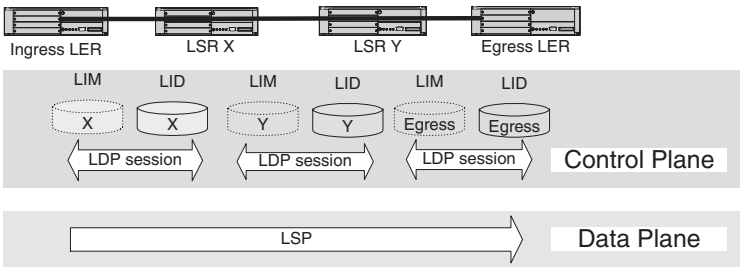

Figure 6. Label information mirrors in upstream LSRs.

initialization), it checks the integrity of the stored label information first. If possible, it will continue the recovery from that information. Otherwise, all its LIMs and LIDs will be set according to actual network configuration in the data plane as initialization from scratch.

The following is the generic recovery procedure. Assume the upstream node plays an active role in the initialization. Otherwise, some appropriate modifications are needed.

1. The upstream node checks the integrity flag of its LIM for the outgoing link. If it finds the flag is 'False', it initializes the LIM according to the channel configuration of the outgoing link in the data plane. If it finds the flag is 'True', it keeps its stored LIM information.
2. The upstream node advises the downstream node about the contents of its LIM by a LIM TLV in an LDP session initialization message.
3. The downstream node checks the integrity flag of its LID for the incoming link. If it finds the flag is 'False', it initializes the LID according to the channel configuration of the incoming link in the data plane. If it finds the flag is 'True', it keeps its stored LID information.
4. The downstream node calculates the logical intersection of the received LIM with its own LID.
5. The downstream node updates its LID as the calculated intersection.
6. The downstream node sends the intersection calculated back to the upstream node by a LID TLV.
7. The upstream node updates its LIM as the LID TLV indicates.

In the first example, the generic recovery procedure is applied to the case of single control node failure. After a single control node failure happens, e.g., LSR X fails (refer to Figure 4), these are the recovery steps.

1. The ingress LER checks the integrity flag of its LIM for the outgoing link and finds the flag is 'True', it keeps its stored LIM information.
2. The ingress LER advises LSR X about the contents of its LIM by a LIM TLV in an LDP session initialization message.
3. LSR X checks the integrity flag of its LID for the incoming link and finds the flag is 'False', it initializes the LID according to the channel configuration of the incoming link in the data plane.
4. LSR X calculates the logical intersection of the received LIM with its own LID.
5. LSR X updates its LID as the calculated intersection.
6. LSR X sends the intersection calculated back to the ingress LER by a LID TLV.
7. The ingress LER updates its LIM as the LID TLV indicates.

In parallel to these steps, LSR X also recovers its LIM corresponding to its outgoing link to LSR Y. The initialization procedure guarantees that the label information is recovered for all LDP sessions related to the failed LSR.

As the second example, the generic recovery procedure is applied to the case of single control channel failure. When a control channel fails, for example, between LSR X and LSR Y (refer to Figure 3), the LDP session between them will be closed accordingly. In our proposal, both the

LIM residing in LSR X and the LID residing in LSR Y will not be emptied. After the fault is fixed, LSR X and LSR Y re-initialize a new LDP session between them.

1. LSR X checks the integrity flag of its LIM and finds the flag is 'True', it keeps its stored LIM.
2. LSR X advises the contents of its LIM to LSR Y by a LIM TLV in an LDP session initialization message.
3. LSR Y checks the integrity flag of its LID and finds the flag is 'True', it keeps its stored LID.
4. LSR Y calculates the logical intersection of the received LIM with its own LID.
5. LSR Y updates its LID as the calculated intersection.
6. LSR Y sends the intersection calculated back to LSR X by a LID TLV in an LDP session initialization message.
7. Upon receiving the LID TLV, LSR X updates its LIM as the LID TLV indicates.

Therefore, a new LDP session is set up with the label information before the failure happens. It should be noted that the re-initialization procedure is independent of the type of control plane failures. It handles control channel failures and control node failures in a unified manner. However, for some particular failure scenarios, there are some redundant steps in the recovery procedure. For example, in the control channel failure case, in step 4 when LSR Y calculates the logical intersection of the received LIM with its own LID, the result will be exactly the same as its LID. So in step 5, the update of LSR Y's LID as the calculated intersection will be redundant. In practice, usually it is hard to know the exact reasons of failures. So it is critical to design a recovery mechanism not having to distinguish the failure modes. By using some potential redundant steps, the proposed recovery procedure operates independently to failure modes. Moreover, the recovery does not rely on the information stored in the failed LSRs before failures, so it can recover not only from warm status but also from cold status.

3.4. Example

To better illustrate how the proposal works, the recovery of the LDP session between the ingress LER and LSR X after the failure of LSR X will be shown in more details.

The configurations in each node include the port configurations and the channel configurations. The port configuration defines the mapping of ports and fibres. Table II shows the port configuration of the ingress LER. Table III shows the port configuration of LSR X. The channel configuration defines the assignable channels in each fibre. Tables IV and V show the channel configuration of Fibres A and B, respectively. Each node stores the channel configurations of all fibres that connect to it. For example, the ingress LER stores the channel configurations of Fibres A and B because these fibres are connected to it. The configuration information can be obtained by manually editing/setting configuration files, or through the network management system.

When the ingress LER and LSR X are initialized from scratch, both will reset their integrity flags associated with the LDP session between them. This indicates no LIM or LID will be available for the initialization.

Here we assume the upstream node plays an active role in the initialization of the LDP session. However, if the upstream node plays a passive role. The proposal still works with some proper modifications.

Table II. Port configuration of the ingress LER.

Port ID	Fibre ID
Output port 1	Fibre A
Output port 4	Fibre B
●●●	●●●

Table III. Port configuration of LSR X.

Port ID	Fibre ID
Input port 1	Fibre A
Input port 2	Fibre B
●●●	●●●

Table IV. Channel configuration of Fibre A.

Assignable channels
Wavelength 1
Wavelength 2
Wavelength 3
Wavelength 4

Table V. Channel configuration of Fibre B.

Assignable channels
Wavelength 1
Wavelength 2
Wavelength 3
Wavelength 4

After the ingress LER and LSR X finish the neighbour discovery and open a TCP session for the LDP session, the ingress LER sends LIM TLV to LSR X in its LDP session initialization message. Since the integrity flag of the LDP session is 'False', the LIM TLV will include the label information for the maximum label space of the LDP session, which reflects the channel configuration in the data plane. Table VI illustrates the contents of the LIM TLV.

After LSR X receives the LIM TLV, it will also use its maximum label space to calculate the logical intersection, because its integrity flag indicates 'False' for this LDP session. The intersection calculated is identical to the maximum label space of either the ingress LER or LSR X. LSR X updates its LID as the intersection calculated and set the integrity flag of the LDP session. Then LSR X sends its LID (identical to the intersection calculated) back to the ingress LER. The ingress LER updates its LIM as the LID received and set its integrity flag of the LDP session. This ends the initialization of the LDP session.

Table VI. Contents of the LIM TLV.

Fibre ID	Channel ID
Fibre A	Wavelength 1
Fibre A	Wavelength 2
Fibre A	Wavelength 3
Fibre A	Wavelength 4
Fibre B	Wavelength 1
Fibre B	Wavelength 2
Fibre B	Wavelength 3
Fibre B	Wavelength 4

Table VII. Contents of the LIM and LID after some channels are assigned to lightpaths.

Fibre ID	Channel ID
Fibre A	Wavelength 1
Fibre A	Wavelength 2
Fibre B	Wavelength 2
Fibre B	Wavelength 3

As some channels between the ingress LER and LSR X are assigned to lightpaths bearing user communications, the contents of LIM and LID are updated accordingly. As an example, Table VII illustrates their contents. Please note that according to the semantics we defined for the label information, these labels are usable labels for newly established lightpaths.

Assume now the control node of LSR X fails and is replaced by a new node. The new control node gets its configurations first, then resets its integrity flag of the LDP session and re-establishes the LDP session. With the proposed recovery mechanism, the ingress LER will send its LIM, whose contents are shown in Table VII, to LSR X. After LSR X receives the LIM, it calculates the logical intersection with its maximum label space since its integrity flag of the LDP session is 'False'. After updating its LID as the intersection calculated, which is also identical to the LIM received, the label information before the failure is recovered in LSR X.

In the procedure of the recovery, some steps might be omitted to optimise the recovery. In this example, after LSR X recovers its LID, it might not transmit its LID back to the ingress LER. The ingress LER might not update its LIM as the received LID for this will not affect its LIM at all. However, if these redundant steps are kept, the same recovery procedure will be able to handle all kinds of control plane failures. As a result, it is not necessary to identify the types of the failures. In most cases, this feature will simplify the recovery.

4. ANALYSIS OF INTEROPERABILITY

Now, we will discuss the interoperability of control nodes capable of the proposed recovery method with standard control nodes, which are incapable of the method. This is important in a

multiple-vendor network environment. It is also of significance in the deployment of this enhanced function in a standard GMPLS-controlled optical network. Through this analysis, we will show that the recovery method will not interfere with the normal standardized operations of CR-LDP. It can be predicted that the graceful recovery ability will not be available if any control node in a pair of adjacent control nodes does not support it.

First, let us analyse the single control channel failure case, e.g. the control channel between LSR X and LSR Y fails. Assume LSR X is enhanced by the recovery method but LSR Y is not. When LSR X sends an LDP session initialization message to LSR Y with a new TLV object, LIM TLV, the U bit (i.e. 'Unknown bit') of the new TLV is set. Therefore, LSR Y silently ignores the new TLV and processes the rest of the LDP session initialization message as if the new TLV does not exist. So the LID in LSR Y and standardized CR-LDP operations are not affected.

Assume LSR Y is enhanced by the recovery method but LSR X is not. When LSR X sends an LDP Session Initialization message to LSR Y without encoding LIM TLV, LSR Y will simply assumes the maximum usable labels based on the channel configuration is advertised and does as the recovery procedure requires. When LSR Y sends back an LDP session initialization message to LSR X, a LID TLV is encoded with U bit being set. LSR X simply ignores the new TLV. No standardized CR-LDP operation is affected.

Similar analysis can be made for a single control node failure and will lead to the same conclusion.

5. IMPLEMENTATION AND DEPLOYMENT ISSUES

Network service providers can offer the recovery from control plane failures to their clients as a value-added service. In addition to the attributes of desired protection and restoration of lightpaths against failures in the data plane, users can also specify their preference of recovery service for the control plane failures. Table VIII shows the different reliability enhancement services that a network service provider can offer to its customers. Negotiation of the recovery service for the control plane can be done before a lightpath is established. The maximum tolerant recovery time should also be negotiated, since for some clients, they might want to use other disjoint paths after a certain time period degradation in optical network applications caused by a control plane failure.

Figure 7 shows an example of implementation of the recovery method in a control node. It has one incoming link and one outgoing link. There is one downstream side LDP entity corresponding to the incoming link, and each downstream side LDP entity has a private LID.

Table VIII. Control plane recovery service as a value-added service.

Control plane recovery service	Data plane protection and restoration
Best effort	Best effort
Offered	Best effort
Best effort	Multiple grades offered
Offered	Multiple grades offered

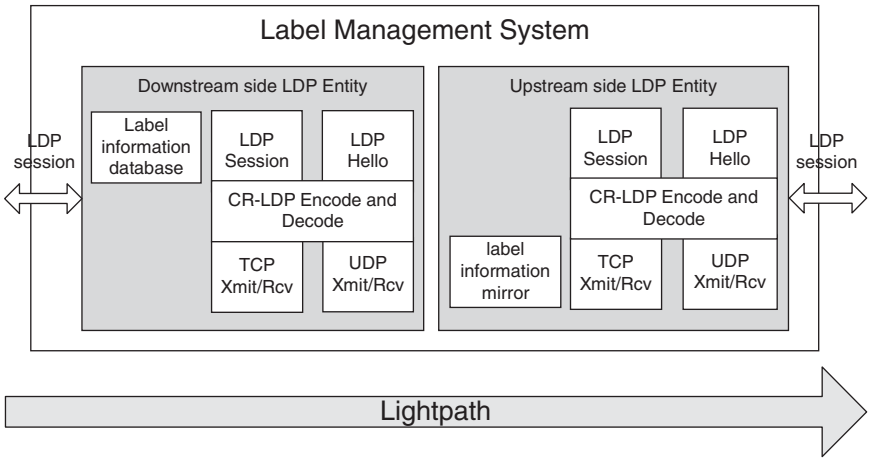

Figure 7. Modules of CR-LDP within a control node.

That means label information is stored on a per session basis. Similarly, there is one upstream side LDP entity corresponding to the outgoing link, and each upstream side LDP entity has it own LIM. So either a LID or a LIM is associated with an LDP entity, thus only with respect to an incoming link or an outgoing link, respectively. This will be more scalable in a large network. Different LDP entities in one control node use different TCP/UDP ports to communicate with their peers.

The recovery method can be incorporated with a centralized recovery mechanism, where all control information is backed up in a centralized network management system. The centralized recovery mechanism could be beneficial in a large area recovery. For example, if the control plane of a whole domain fails, the control information could be recovered through a centralized method. Our proposal is more suitable to a small area recovery. They can be used as a complement to each other in a hierarchical manner.

Our proposal can also include protection of control channels by using backup control channels [8,27]. If backup control channels can protect the failures, LDP sessions will not sense the happening of the failures and will not take any recovery action. To co-ordinate the protection by backup control channels and the recovery in LDP, a timer (i.e. the LDP Keepalive timer) is required. If the protection switching does not succeed within a time bound, the TCP and LDP layer will take action. Then the recovery approach will play a role.

Our proposal can also be incorporated with an LDP fault tolerance mechanism based on timer-controlled re-sending of LDP messages [26]. One possibility is to let the timer-controlled re-sending of LDP messages try first during the recovery. If sufficient information is preserved during a control plane failure, the timer-controlled re-sending of LDP messages could be faster to recover the LDP session, because it does not need to re-establish an LDP session. However, if it cannot recover because of lack of necessary information, e.g. in most control node failures, then our proposed recovery mechanism can take over. In this case, the integrity flag of a LID or a LIM needs to be enhanced to reflect the integrity of not only label information but also related session state variables and session timers.

6. RECOVERY CAPABILITY AND LIMITATIONS

The recovery method applies to multiple control channel failures, but it only applies to single node failure among any pair of adjacent nodes. The limitation of single node failure arises from the fact that if two adjacent nodes fail simultaneously, the label information of the link between them will be lost. The recovery method will not have this trouble in multiple node failures provided that none of them is adjacent. If the recovery method needs to be extended to the failure of two adjacent nodes, the label information might need to be redundantly stored in control nodes two or more hops away along a lightpath. This is out of the scope of this paper.

Generally, in the stable state of the CR-LDP operations before a control channel failure happens, the LIM in the upstream LSR and the LID in the downstream control node are synchronized. But when the control channel failure happens before the CR-LDP operation reaches the stable state, their contents could be slightly different. This will cause the so-called 'over-booking' problem.

One example is when an upstream control node finishes updating its LIM and sends out a CR-LDP label release message to its downstream peer, a control channel failure happens and prevents the message from arriving at the downstream node. Therefore, the LIM and the LID are not synchronized now. The upstream control node treats that label as being freed, but the downstream peer treats that label as in-use. After the recovery, the LIM will be restored as the LID in the downstream control node according to the proposed recovery procedure and that label is treated as in-use, because we assume the upstream node plays an active role in the session initialization. However, the lightpath will no longer be usable because the upper layer application has already released it. That label becomes dangling. Thus, the over-booking problem occurs. Another example is when a downstream control node finishes assigning a label for its incoming link and sends out a CR-LDP label mapping message to its upstream peer, a control channel failure happens and prevents the message from arriving at the upstream node or the upstream peer itself fails.

The over-booking problem will cause some resources being reserved falsely after the recovery and no traffic can use them any longer. This could be remedied by administrative methods or by periodical network level re-configuration. In fact, recalling that CR-LDP runs in the downstream on demand mode, an upstream control node is always conservative about the resources. More specifically, an upstream control node will not use a label assigned for it until it receives an explicit acknowledgement from its downstream peer through a CR-LDP label mapping message. When an upstream control node frees a label, it does so immediately and then notifies its downstream peer through a CR-LDP label release message. Regarding the fact that the signalling procedure to set up or tear down a lightpath only lasts for tens of milliseconds to several seconds, while the lifetime of a lightpath is days to months, the chance of over-booking problem is fairly rare. Anyway, although the over-booking problem will waste some network resources, in most cases it is better than terminating the lightpaths unnecessarily. How to solve the over-booking problem is a topic for further study.

7. CONCLUSIONS

We proposed a recovery method for control plane failures in GMPLS-controlled optical networks. It helps to maintain user communications transported by lightpaths during control

plane failures and restores full control functionality without any interruption of user communications. This method can recover from any number of control channel failures and control node failures provided that none of the failed node is adjacent. Our proposal also has some significant features.

1. It is a fully distributed mechanism. So it is more reliable and scalable and it has no potential bottleneck of centralized control mechanisms.
2. It handles all kinds of control plane failures in a unified manner, so there is no need to distinguish modes of control plane failures.
3. It inter-operates seamlessly with the standard LDP (LDP without the improved recovery method). So it does not interfere with the normal standardized operations of LDP. This merit makes it easier to be progressively deployed in the existing standard LDP implementations.
4. It offers transport service providers more options to provide value-added services. In addition to multiple-grade of protection and restoration against failures in the data plane, now transport service providers can also offer recovery from failures in the control plane.
5. We suggest implementing LIDs and LIMs on a per session basis in order to be scalable in large networks.
6. There is no fundamental need to use additional hardware.

In addition, our proposal can be extended to work with other reliability enhancement mechanisms, e.g., centralized recovery mechanisms, the protection of control channels by using backup control channels, and the fault tolerance for LDP based on the timer-controlled re-sending of LDP messages.

The reliability of the control plane in networks is critical for proper networking. This research is only a start to this complicated issue. More research is needed to improve the reliability of the whole protocol suite running in the control plane.

REFERENCES

1. Armitage G. MPLS: the magic behind the myths. *IEEE Communications Magazine* 2000; **38**(1):124–131.
2. Xiao X, Hannan A, Bailey B, Ni LM. Traffic engineering with MPLS in the Internet. *IEEE Network* 2000; **14**(2): 28–33.
3. Banerjee A, Drake J, Lang JP, Turner B, Kompella K, Rekhter Y. Generalized multiprotocol label switching: an overview of routing and management enhancements. *IEEE Communications Magazine* 2001; **39**(1):144–150.
4. Banerjee A, Drake J, Lang J, Turner B, Awduche D, Berger L, Kompella K, Rekhter Y. Generalized multiprotocol label switching: an overview of signaling enhancements and recovery techniques. *IEEE Communications Magazine* 2001; **39**(7):144–151.
5. Hache L, Li L. Unified control infrastructure for carrier network evolution. *IEEE Communications Magazine* 2000; **38**(11):74–77.
6. Jamoussi B, *et al*. Constraint-based LSP setup using LDP. IETF draft draft-ietf-mpls-cr-ldp-04.txt, work in progress, July 2000.
7. Green P. Progress in Optical Networking. *IEEE Communications Magazine* 2001; **39**(1):54–61.
8. Lang JP, Drake J. Link Management Protocol (LMP), *Proceedings of 16th Annual National Fiber Optic Engineers Conference* (*NFOEC*), vol. 2. Denver, Colorado, August 2000; 368–377.
9. Ghani N. Lambda-labeling: a framework for IP-over-WDM using MPLS. *Optical Networks Magazine* 2000; 45–58.
10. Rodriguez-Moral A, Bonenfant P, Baroni S, Wu R. Optical data networking: protocols, technologies, and architectures for next generation optical transport networks and optical internetworks. *IEEE Journal of Lightwave Technology* 2000; **18**(12):1855–1870.
11. Awduche D, Rekhter Y. Multiprotocol lambda switching: combining MPLS traffic engineering control with optical crossconnects. *IEEE Communications Magazine* 2001; **39**(3):111–116.
12. Li G, Yates J, Wang D, Kalmanek C. Control plane design for reliable optical networks. *IEEE Communications Magazine* 2000; **40**(2):90–96.

13. Labovitz C, Ahuja A, Jahanian F. Experimental study of Internet stability and backbone failures. *Digest of Papers of 26th Annual International Symposium on Fault-Tolerant Computing*, Madison, Wisconsin, June 1999, 278–285.
14. Paxson V. End-to-end routing behavior in the Internet. *IEEE/ACM Transaction on Networking* 1997; **5**(5):601–615.
15. Kuhn DR. Sources of failure in the public switched telephone network. *IEEE Computer* 1997; **30**(4):31–36.
16. Rajagopalan B, Pendarakis D, Saha D, Ramamoorthy RS, Bala K. IP over optical networks: architectural aspects. *IEEE Communications Magazine* 2000; **38**(9):94–102.
17. Moral AR, Bonenfant P, Krishnaswamy M. The optical Internet: architectures and protocols for the global infrastructure of tomorrow. *IEEE Communications Magazine* 2001; **39**(7):152–159.
18. Griffith D. A Comparison of RSVP-TE and CR-LDP. Optical Internetworking Forum (OIF) contribution OIF2000.179, August 2000.
19. Brittain P. MPLS traffic engineering: a choice of signaling protocols. *White paper*, Data Connection Limited, http://www.dataconnection.com, January 2000.
20. Pulley R, *et al*. A comparison of MPLS traffic engineering initiatives. *White paper*, NetPlane System, Inc., http://www.netplane.com, 2000.
21. Gerstel O, Ramaswami R. Optical layer survivability: a services perspective. *IEEE Communications Magazine* 2000; **38**(3):104–113.
22. Chen TM, Oh TH. Reliable services in MPLS. *IEEE Communications Magazine* 1999; **37**(12):58–62.
23. Ramaswami R, Sivarajan KN. *Optical Networks*: *A Practical Perspective*. Morgan Kaufmann Publishers Inc.: San Francisco, 1998.
24. Andersson L, *et al*. LDP Specification. IETF RFC 3036, January 2001.
25. Xu Y, *et al*. Generalized MPLS Control plane Architecture for Automatic Switched Transport Network. IETF draft draft-xu-mpls-ipo-gmpls-arch-00.txt, November 2000.
26. Farrel A, *et al*. Fault tolerance for LDP and CR-LDP. IETF draft draft-ietf-mpls-ldp-ft-01.txt, work in progress, February 2001.
27. Ashwood-Smith P, *et al*. Generalized multi-protocol label switching (GMPLS) architecture. IETF draft draft-many-gmpls-architecture-00.txt, work in progress, February 2001.
28. Wu J, Montuno DY, Mouftah HT, Wang G, Dasylva AC. Improving the reliability of the label distribution protocol. *Proceedings of the 26th Annual IEEE Conference on Local Computer Networks* (*LCN*), Tampa, Florida, November 2001, 236–242.

AUTHORS' BIOGRAPHIES

Jing Wu received his BASc in 1992 and his PhD in 1997 from Xi'an Jiao Tong University, China. He worked at Beijing University of Posts and Telecommunications (Beijing, China) as an assistant professor, Queen's University (Kingston, Ontario, Canada) as a postdoctoral fellow, and Nortel Networks Corporate (Ottawa, Ontario, Canada) as a systems design engineer. He is now a research scientist in Communications Research Centre Canada, His research interests mainly include control and management of optical networks, protocols and algorithms in networking, network performance evaluation and optimisation, etc. Dr Wu is a member of the IEEE and the Society of Computer Simulation.

Delfin Y. Montuno obtained the BSc in Physics from Ateneo de Manila University, Philippines in 1972; MSc in Electronics from Yamanashi University, Japan in 1976; Doctor of Information Engineering from Nagoya University, Japan in 1980; and PhD in Computer Science from University of Toronto, Canada in 1985, Since 1984, he has been with Nortel Networks (formerly BNR) where he worked on a number of projects including the development of VLSI layout algorithms. He is currently involved in the development of high capacity switch fabrics, and resource management algorithms and mechanisms for high-speed networks. His other research interests include constraint logic programming, neural networks, fuzzy logic, and computational geometry, and their applications to network traffic control. He is also an Adjunct Research Professor in the School of Mathematics and Statistics. Carleton University, Canada.

Hussein T. Mouftah joined the Department of Electrical and Computer Engineering at Queen's University in 1979, where he is now a Full Professor and the Department Associate Head, after three years of industrial experience mainly at Bell Northern Research of Ottawa (now Nortel Networks). He has spent three sabbatical years also at Nortel Networks (1986–1987, 1993–1994, and 2000–2001), always conducting research in the area of broadband packet switching networks, mobile wireless networks and quality of service, over the optical Internet He served as Editor-in-Chief of the IEEE Communications Magazine (1995–1997) and IEEE Communications Society Director of Magazines (1998–1999). Dr Mouftah is the author or co-author of two books and more than 600 technical papers and 8 patents in this area. He is the recipient of the 1989 Engineering Medal for Research and Development of the Association of Professional Engineers of Ontario (PEO). He is the joint holder of a Honourable Mention for the Frederick W. Ellersick Price Paper Award for Best Paper in Communications Magazine in 1993. Also he is the joint holder of the Outstanding Paper Award for a paper presented at the IEEE 14th International Symposium on Multiple-Valued Logic. He is the recipient of the IEEE Canada (Region 7) Outstanding Service Award (1995). Dr Mouftah is a Fellow of the IEEE (1990).

Guo-Qiang Wang is a technical manager working for Nortel Advanced Technology group. With over a decade working experience in telecommunication, he has involved network products and system engineering including SONET, ATM and IP routing. Since 1999 he has led a team working on GMPLS control platform technology for switched DWDM networking. His research activities cover the protocols and algorithms for optical routing, signalling control and mesh topology protection. He has co-authored several papers for international conferences, and owned couple of pending patents related to above fields, G. Q. Wang held a MS in Computer Science.

Abel Dasylva graduated from l'Ecole Nationale Superieure de Techniques Avancees, Paris, with a degree in general engineering, and from the University of Illinois Urbana-Champaign with a master in electrical engineering in 1998. From 1998 to 1999, he worked as a research engineer for Nokia, in the Broadband Network group, Burlington MA, where his activities concerned admission control and quality of service routing for internet and 3G wireless networks. Since 1999, he has worked as a research engineer for Nortel Networks, Ottawa ON, with the Advanced Technology Investments group, in the area of optical internet control plane protocols.

