Network Working Group
Internet Draft
Expiration Date: September 2000

Jonathan P. Lang (Chromisys)
Krishna Mitra (Chromisys)
John Drake (Chromisys)
Kireeti Kompella (Juniper Networks)
Yakov Rekhter (Cisco Systems)
Debanjan Saha (Tellium Optical Systems)
Lou Berger (LabN Consulting, LLC)
Debashis Basak (Marconi)

Link Management Protocol (LMP)

draft-lang-mpls-lmp-00.txt

1. Status of this Memo

2. Abstract

Future optical networks will consist of optical crossconnects (OXCs) that may be configured with links consisting of a number of user bearer channels and an associated control channel. This document specifies a link management protocol (LMP) that runs between neighboring OXCs and will be used for both link provisioning and fault isolation. A unique feature of LMP is that it is able to isolate faults in both opaque and transparent networks, independent of the encoding scheme used for the bearer channels. LMP will be used to maintain control channel connectivity, verify bearer channel connectivity, and isolate link, fiber, or channel failures within

the optical network.

3. Introduction

Future optical networks will consist of optical crossconnects (OXCs) that use the MPLS control plane to dynamically provision optical trails and to provide network survivability using protection and restoration techniques. A pair of OXCs may be connected by a number of fibers, and each fiber may be used to transmit multiple wavelengths if DWDM is used. Furthermore, multiple fibers and/or multiple wavelengths may be combined into a single logical link, where we follow the convention of [2] and define a link as a logical relationship associating a control channel with zero or more user bearer channels.

This document specifies a link management protocol (LMP) that runs between neighboring OXCs and will be used for both link provisioning and fault isolation. The intent of this document is to lay the foundation for the protocol, and as this document progresses, the messages referenced herein will be explicitly defined. We do propose, however, that the messages are IP encoded so that the link level encoding becomes an implementation agreement and is not part of LMP specifications.

In this document, we will follow the naming convention of [3] and use OXC to refer to all categories of optical crossconnects, irrespective of the internal switching fabric. We distinguish between crossconnects that have electronic cores, called digital crossconnects (DXCs), and those that are all-optical, called photonic crossconnects (PXCs) - referred to as pure crossconnects in [3], because the transparent nature of PXCs introduces new restrictions for monitoring and managing the data channels (see [4] for proposed extensions to MPLS for performance monitoring in photonic networks). The LMP that we propose, however, can be used for any type of OXC, enhancing the functionality of traditional DXCs while enabling PXCs to intelligently interoperate in heterogeneous optical networks.

Due to the transparent nature of PXCs, traditional methods can no longer be used to monitor and manage links. LMP has been designed to address issues faced in managing links in a network with PXCs.

In addition, since LMP does not dictate the actual transport mechanism, this protocol can be implemented on both PXCs and DXCs to allow interoperability. A requirement for LMP is that each link has an associated bi-directional control channel and that free bearer channels must be opaque (i.e., able to be terminated); however, once a bearer channel is allocated, it may become transparent. There is no requirement that the control channel and bearer channels share the same medium; however, the control channel must terminate on the same two nodes that the bearer channels span. LMP consists of four types of functions: a Hello exchange is used to verify and maintain control-channel and link connectivity between neighboring OXCs; link verification is used to verify bearer channel connectivity and exchange Label mappings; a LabelSummary exchange is used to synchronize Label matching and correlate link properties; and a

fault localization technique is used to isolate link and channel failures and initiate protection and restoration techniques.

The organization of the remainder of this document is as follows. In Section 4, we discuss the role of the control channel and the Hello exchange in maintaining link connectivity. The link verification procedure is discussed in Section 5. In Section 6, we show how the LMP will be used to isolate link and channel failures within the optical network.

4. Control channel management

To establish a link between two OXCs, a control channel must first be configured. The control channel can be used to exchange MPLS control-plane information such as link provisioning and fault isolation information (implemented using a messaging protocol such as LMP, proposed in this draft), path management and label distribution information (implemented using a signaling protocol such as RSVP-TE [5] or CR-LDP [6]), and topology and state distribution information (implemented using traffic engineering extended protocols such as OSPF [7] and IS-IS [8]). We require a control channel be associated with each link. We do not specify the exact implementation of the control channel, but rather we assign a (fiber, wavelength) pair to each control channel for identification purposes. This allows the control channel implementation to encompass both in-band and out-of-band mechanisms including the case where the control channel is transmitted separately from the associated bearer channel(s) of a link, either on a separate wavelength or a separate fiber.

The control channel of a link can be either explicitly configured or automatically selected, however, for the purpose of this document we will assume the control channel is explicitly configured. A control channel will be assigned a (fiber, wavelength) pair for identification purposes. Note that for in-band signaling, a bearer channel could be allocated to the same (fiber, wavelength) pair as the control channel; however, this is not true when the control channel is transmitted separately from the bearer channels. In addition to a primary control channel, an ordered list of backup control channels can also be specified.

For LMP, it is essential that a control channel is always available for a link, and in the event of a control channel failure, an alternate (or backup) control channel must be made available to reestablish communication with the neighboring OXC. If the control channel cannot be established on the primary (fiber, wavelength) pair, then a backup control channel should be tried. Of course, alternate control channels can (and should) be pre-configured, however, coordinating the switchover of the control channel to an alternate channel is still an important issue. Specifically, if the control channel fails but the node is still operational (i.e., the bearer channels are still passing user data), then both the local and remote nodes should switch to an alternate control channel.

4.1. Hello protocol

Once a control channel is configured between two OXCs, a Hello protocol will be used to establish and maintain connectivity between the OXCs and to detect link failures. The Hello protocol of LMP is intended to be a lightweight keep-alive mechanism that will react to control channel failures rapidly so that IGP Hellos are not lost and the associated link-state adjacencies are not removed. Furthermore, the RSVP Hello of [5] is not needed since the LMP Hellos will detect link layer failures.

The Hello protocol will consist of a single unicast Hello message that is periodically sent along the control channel to the adjacent OXC. Each Hello message will contain two sequence numbers: the first will be the sequence number (SendSeqNum) for this Hello message and the second will be the sequence number (RecSeqNum) of the last Hello message received along the link from the adjacent OXC. The sequence number in the Hello message starts at 1 and 0 is used to indicate a node reset. When a node is brought up (either through a regular boot or through a reboot), the value of SendSeqNum

will be reset to 0. Having sequence numbers in the Hello messages provide a two-fold service. First, the remote OXC will detect that a node has rebooted if the SendSeqNum is 0. If this occurs, the remote node will indicate its knowledge of the reboot by setting RecSeqNum=0 in the Hello messages that it sends and will wait to receive a Hello message with SendSeqNum=1 before proceeding with bearer channel verification. Second, by including the RecSeqNum in Hello packets, the local node will know which Hello packets the remote node has received. This is important because the local node will behave differently to messages based on which Hello messages the remote node is responding to. For example, if the local node has rebooted and receives a message with a RecSeqNum value that is not equal to 0, the local node should discard the message and wait until it receives a Hello message with RecSeqNum=0 indicating that the remote node knows it has rebooted.

5. Verifying link connectivity

In this section, we describe the mechanism used to verify the physical connectivity of the bearer channels. This will be done initially when a link is established, and subsequently, on a periodic basis for all free bearer channels on the link. A unique characteristic of all-optical PXCs is that the data being transmitted over a bearer channel is not terminated at the PXC, but instead passes through transparently. This characteristic of PXCs poses a challenge for validating the connectivity of the bearer channels since shining unmodulated light through a bearer channel may not result in received light at the next PXC. This is because there may be terminating (or opaque) elements, such as DWDM equipment, in between the PXCs. Therefore, to ensure proper verification of bearer channel connectivity, we require that until the bearer channels are allocated, they must be opaque. Furthermore, we assume that the architecture of the OXC is designed so that messages can be sent and received over any bearer channel.

Note that this requirement is trivial for DXCs since each channel (bearer or control) is received electronically before being forwarded to the next DXC, but that in PXCs this is an additional requirement.

To interconnect two OXCs, a link must be added between them, and at a minimum, the link must contain a control channel spanning the two OXCs. Optionally, the attributes of a link may include the protection mechanism for the control channel, a list of bearer channels, and the protection mechanism for each bearer channel.

As part of the link verification protocol, the control channel is first verified, and connectivity maintained, using the Hello protocol discussed in Section 4.1. Once the control channel has been established between the two OXCs, bearer channel connectivity is verified by exchanging Ping-type Test messages over all of the bearer channels specified in the link. It should be noted that all messages except for the Test message are exchanged over the control channel and that Hello messages continue to be exchanged over the control channel during the bearer channel verification process. The Test message is sent over the bearer channel that is being verified. Bearer channels are tested in the transmit direction as they are uni-directional, and as such, it may be possible for both OXCs to exchange the Test messages simultaneously.

To initiate the link verification process, the local OXC first sends a BeginVerify message over the control channel to indicate that the node will begin sending Test messages across the bearer channels of a particular link. The BeginVerify message contains the number of bearer channels that are to be verified and, for each bearer channel, the local RSVP Label object, represented as a (fiber, lambda) pair as defined in [9]. When the remote OXC receives a BeginVerify message and it is ready to receive Test messages, it sends a BeginVerifyAck message back to the local OXC. When the local OXC receives a BeginVerifyAck message from the remote OXC, it will begin transmitting periodic Test messages over the specified bearer channels. The Test message will include the Label object for the associated channel. The remote OXC will return a TestStatus (Success or Failure) message in response for each bearer channel and will expect a TestStatusAck message from the local node to confirm receipt.

The local (transmitting) node will send a given Test message periodically on the corresponding bearer channel until it receives a correlating TestStatusSuccess or TestStatusFailure message on the control channel from the remote (receiving) node. The remote node will send a given TestStatusSuccess or TestStatusFailure message periodically on the control channel until it receives a correlating TestStatusAck message on the control channel from the local node. Message correlation is done using the local node's (fiber, lambda) pair.

When the Test message is detected at the remote OXC, the Label is recorded and mapped to the remote OXCÆs Label for that channel. The

remote OXC then sends a TestStatusSuccess message over the control channel to the local OXC indicating that the Test message was detected and the physical connectivity of the bearer channel has been verified. The TestStatusSuccess message includes both the local and remote OXCs Label objects for the bearer channel. When the TestStatusSuccess message is received, the local OXC marks the channel as UP, sends a TestStatusAck message to the remote OXC, and begins testing the next bearer channel. If, however, the Test message is not detected at the remote node within an observation period (specified by a timeout value), the remote OXC will send a TestStatusFailure message over the control channel indicating that the verification of the physical connectivity of the bearer channel has failed. When the local OXC receives a TestStatusFailure message, it will mark the channel as FAILED, send a TestStatusAck message to the remote OXC, and begin testing the next bearer channel. When all the bearer channels on the list have been tested, the local OXC will send an EndVerify message to indicate that testing has been completed on this link. An EndVerifyAck is sent as a response.

Both the local and remote nodes will maintain the complete list of Label mappings for correlation purposes, especially in the event of a node reboot.

There is also a LabelSummary message that can be exchanged at any time by the two OXCs. This message contains all the Label associations for a particular link. In addition, each Label [i.e., (fiber, lambda) pair] may have one or more associated protection Labels defined for local (span) M:N protection. If the LabelSummary message received from a remote OXC is accepted and the Label objects match the local Label associations, then the remote local protection definitions are updated and a LabelSummaryAck message is transmitted. Otherwise a LabelSummaryNack message will be transmitted, indicating which Label associations are not correct. If a LabelSummaryNack message is received, the link verification process should be repeated for all mismatched free bearer channels; if an allocated bearer channel has a label mismatch, it should be flagged and verified when it becomes free.

5.1. Example of link verification

The figure below shows an example of the link verification scenario executed when a link between OXC A and OXC B is added. In this example, the link will consist of a bi-directional control channel (indicated by a "c") and three free bearer channels (each transmitted along a separate fiber). The verification process is as follows: OXC A sends a BeginVerify message to OXC B indicating it will begin verifying the bearer channels of the link. OXC B receives the BeginVerify message and returns the BeginVerifyAck message to OXC A. When OXC A receives the BeginVerifyAck message, it begins transmitting periodic Test messages with the Label object

(Fiber 1, 0xffff) across the fiber; the special value of 0xffff for the lambda indicates the whole fiber [9]. When OXC B receives the Test messages, it maps OXC AÆs (Fiber 1, 0xffff) to its own Label of

(Fiber 10, 0xffff) and transmits a TestStatusSuccess message back to OXC A along the control channel. The TestStatusSuccess message will include both the local and remote Label objects for the bearer channel, i.e., (Fiber 1, 0xffff) (Fiber 10, 0xffff). The process is repeated until all of the bearer channels are verified. At that point, OXC A will send an EndVerify message to OXC B to indicate that testing is complete and OXC B will respond with an EndVerifyAck message.

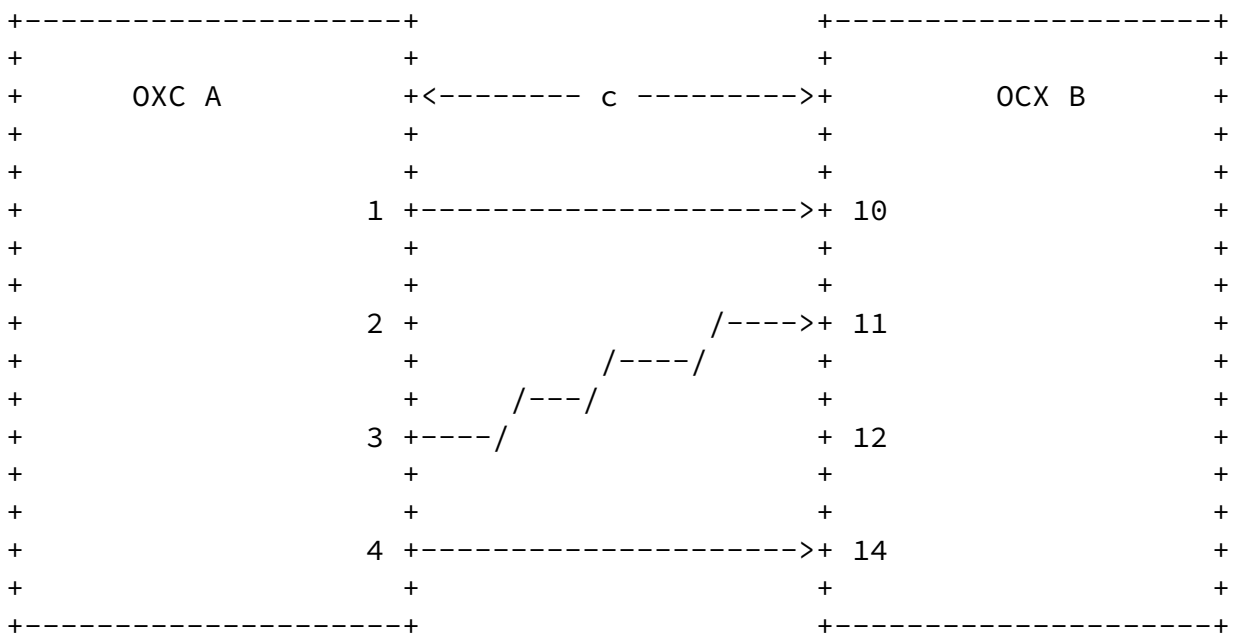


Figure 1: Example of link connectivity between OXC A and OXC B.

6. Fault localization

In this section, we describe a mechanism to rapidly isolate link or bearer channel failures in an optical network. As before, we assume that a bi-directional control channel is always available for inter-node communication and that the control channel spans a single hop between two neighboring OXCs. The case where a control channel is no longer available between two nodes is beyond the scope of this draft. The mechanism used to rapidly isolate link and bearer channel failures is designed to work for unidirectional optical trails, and can be easily extended to work for bi-directional

trails; however, for the purposes of this document, we only discuss the operation when the optical trails are uni-directional.

A link connecting two OXCs consists of a control channel and a number of bearer channels. If bearer channels fail between two OXCs, a mechanism must be used to rapidly locate the failure so that appropriate protection/restoration mechanisms can be initiated. An important implication of using PXCs is that traditional methods used by DXCs to monitor the health of allocated bearer channels may no longer be appropriate since PXCs are transparent to the data bit-rate and format. Instead, fault detection is delegated to the physical layer (i.e., loss of light or optical monitoring of the data) instead of the layer 2 or layer 3.

6.1. Fault detection

As mentioned earlier, fault detection must be handled at the layer closest to the failure; for optical networks, this is the physical (optical) layer. One measure of fault detection at the physical layer is simply detecting loss of light (LOL). Other techniques for monitoring optical signals are still being developed and will not be further considered in this document. However, it should be clear that the mechanism used to locate the failure is independent of the mechanism used to detect the failure, but simply relies on the fact that a failure is detected in the optical layer.

6.2. Fault localization mechanism

If bearer channels fail between two PXCs, the power monitoring system in all of the downstream nodes will detect LOL and indicate a failure. As part of the fault localization, a monitoring window can be used in each node to determine if a single bearer channel has failed or if multiple bearer channels have failed.

As part of the fault localization, a downstream node that detects bearer channel failures across a link will send a Channel_Fail message to its upstream neighbor (bundling together the notification of all of the failed bearer channels) and the node will put the ports associated with the failed bearer channels into the standby state. An upstream node that receives the Channel_Fail message will correlate the failure to see if there is a failure on the corresponding input and output ports for the optical trail(s). If there is also a failure on the input channel(s) of the upstream node, the node will return a Channel_Fail_Ack message to the

downstream node (bundling together the notification of all the channels), indicating that it too has detected a failure. If, however, the fault is CLEAR in the upstream node (i.e., there is no LOL on the corresponding input channels), then the upstream node will have localized the failure and will return a Channel_Fail_Nack message to the downstream node, and initiate protection/restoration procedures.

As part of the Channel_Fail_Nack message, a Notify object may be included when M:N span protection is provided. The Notify object will be used to coordinate channel switchover and will include one or more sub-objects depending on the number of channels that need to be switched. Each sub-object will include a Label pair where the first Label corresponds to the failed bearer channel and the second Label corresponds to the protection bearer channel to be switched to. The protection channels may be preconfigured (using the verify link procedure of Section 5) or they may be dynamically selected by the OXC on the transmit side.

6.3. Examples of fault localization

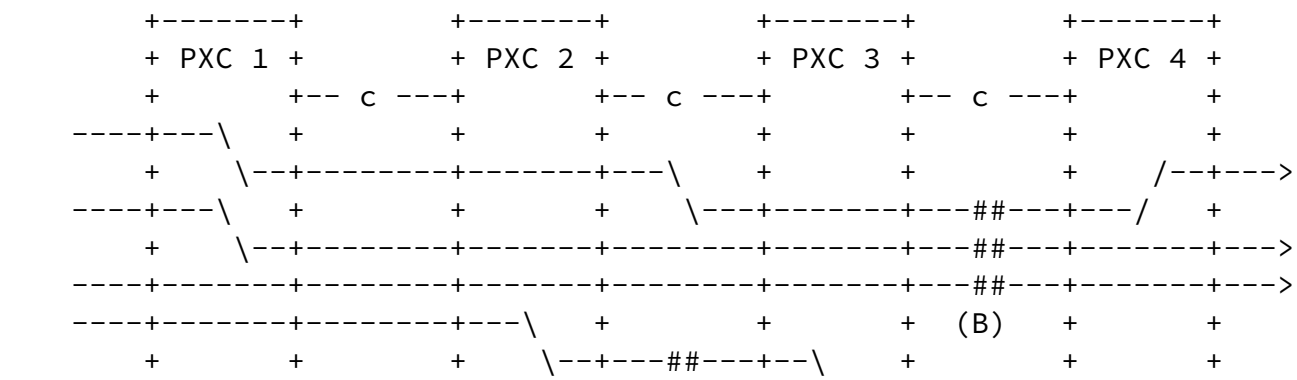
In Fig. 2, a sample network is shown where four OXCs are connected in a linear array configuration. The control channels are bi-directional and are labeled with a "c". All optical trails are uni-directional going left to right.

In the first example [see Fig. 2(A)], there is a failure on a single bearer channel between PXC2 and PXC3. Both PXC3 and PXC4 will detect the failure and each node will send a Channel_Fail message to the corresponding upstream node (PXC3 will send a message to PXC2 and PXC4 will send a message to PXC3). When PXC3 receives the Channel_Fail message from PXC4, it will correlate the failure and return a Channel_Fail_Ack message back to PXC4. Upon receipt of the Channel_Fail_Ack message, PXC4 will move the associated ports into a standby state. When PXC2 receives the Channel_Fail message from PXC3, it will correlate the failure, verify that it is CLEAR, localize the failure to the bearer channel between PXC2 and PXC3, and send a Channel_Fail_Nack message back to PXC3.

In the second example [see Fig. 2(B)], there is a failure on three

bearer channels between PXC3 and PXC4. In this example, PXC4 has correlated the failures and will send a bundled Channel_Fail message for the three failures to PXC3. PXC3 will correlate the failures, localize them to the channels between PXC3 and PXC4, and return a bundled Channel_Fail_Nack message back to PXC4.

In the last example [see Fig. 2(C)], there is a failure on the tributary channel of the ingress node (PXC1) to the network. Each downstream node will detect the failure on the corresponding input ports and send a Channel_Fail message to the upstream neighboring node. When PXC2 receives the message from PXC3, it will correlate the Channel_Fail message and return a Channel_Fail_ACK message to PXC3 (PXCs3 and 4 will also act accordingly). Since PXC1 is the ingress node to the optical network, it will correlate the failure and localize the failure to the bearer channel between itself and the network element outside the optical network.



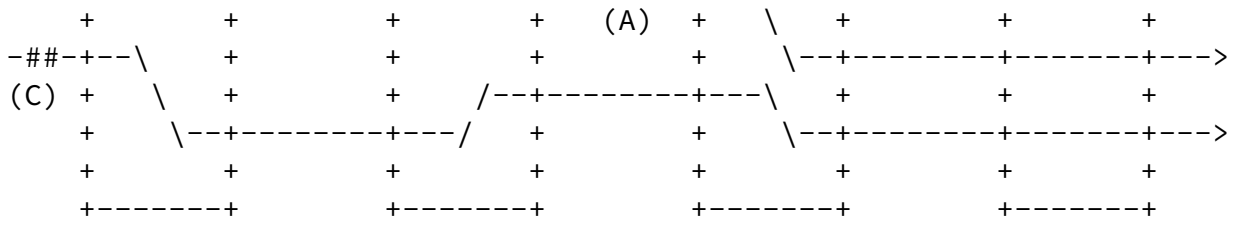


Figure 2: We show three types of bearer channel failures (indicated by ## in the figure): (A) a single bearer channel fails between two PXCs, (B) three bearer channels fail between two PXCs, and (C) a single bearer channel fails on the tributary input of PXC 1. The control channel connecting two PXCs is indicated with a "c".

7. Security Considerations

Security considerations are for future study.

8. References

[1] Bradner, S., "The Internet Standards Process -- Revision 3," BCP 9, RFC 2026, October 1996.

[2] Basak, D., Awduche, D. O., Drake, J., Rekhter, Y., "Multi-protocol Lambda Switching: Issues in Combining MPLS Traffic Engineering Control with Optical Cross-connects," Internet Draft, draft-basak-mpls-oxc-issues-01.txt, February 2000.

[3] Awduche, D. O., Rekhter, Y., Drake, J., Coltun, R., "Multi-Protocol Lambda Switching: Combining MPLS Traffic Engineering Control with Optical Crossconnects," Internet Draft, draft-awduche-mpls-te-optical-00.txt, October 1999.

[4] Ceuppens, L., Blumenthal, D., Drake, J., Chrostowski, J., Edwards, W. L., "Performance Monitoring in Photonic Networks," Internet Draft, March 2000.

[5] Awduche, D. O., Berger, L., Gan, D.-H., Li, T., Swallow, G., Srinivasan, V., "Extensions to RSVP for LSP Tunnels," Internet Draft, draft-ietf-mpls-rsvp-lsp-tunnel-04.txt, September 1999.

[6] Jamoussi, B., et al, "Constraint-Based LSP Setup using LDP," Internet Draft, draft-ietf-mpls-cr-ldp-03.txt, September 1999.

[7] Katz, D., Yeung, D., "Traffic Engineering Extensions to OSPF," Internet Draft, 1999.

[8] Smit, H. and Li, T., "IS-IS extensions for Traffic Engineering," Internet Draft, 1999.

[9] Kompella, K., Rekhter, Y., Awduche, D. O., et al, "Extensions to IS-IS/OSPF and RSVP in support of MPL(ambda)S," Internet Draft, draft-kompella-mpls-optical-00.txt, February 2000.

9. Acknowledgments

The authors would like to thank Vishal Sharma and Stephen Shew for their comments on early versions of the draft.

10. Author's Addresses

Jonathan Lang
Chromisys, Inc.
421 Pine Avenue
Santa Barbara, CA 93117
Email: jplang@Chromisys.com

Krishna Mitra
Chromisys, Inc.
1012 Stewart Drive
Sunnyvale, CA 94086
email: krishna@Chromisys.com

John Drake
Chromisys, Inc.
1012 Stewart Drive
Sunnyvale, CA 94086
email: jdrake@Chromisys.com

Kireeti Kompella
Juniper Networks, Inc.
385 Ravendale Drive
Mountain View, CA 94043
email: kireeti@juniper.net

Yakov Rekhter
Cisco Systems
170 W. Tasman Dr.
San Jose, CA 95134
email: yakov@cisco.com

Debanjan Saha
Tellium Optical Systems
2 Crescent Place
Oceanport, NJ 07757-0901
email: dsaha@tellium.com

Lou Berger
LabN Consulting, LLC
email: lberger@labn.net

Debashis Basak
Marconi
1000 Fore Drive
Warrendale, PA 15086-7502
email: dbasak@fore.com