Reliable Services in MPLS

Thomas M. Chen and Tae H. Oh, Southern Methodist University

ABSTRACT MPLS is a convergence of various implementations of IP switching using ATM-like "label swapping" to speed up packet forwarding without changes to existing IP routing protocols. An important practical issue is the capability to recover quickly from faults. We examine distributed methods for fast fault recovery using modified Label Distribution Protocol messages. To maintain and verify service continuity, methods are proposed for traffic and performance monitoring.

Since the Internet was opened to commercial traffic in 1992, it has grown rapidly from an experimental research network to an extensive public data network. Demand is pushing the capabilities of today's Internet in several dimensions: transmission bandwidth, number of hosts, geographic size, and traffic volume. At the same time, the Internet is evolving from best-effort service toward an integrated or differentiated services framework with quality of service (QoS) assurances which will be necessary for many new applications such as voice over IP, videoconferencing, and multimedia.

In recent years the industry has been searching for an approach to combine the best features of IP and asynchronous transfer mode (ATM), for example, IP routing with the performance and throughput of ATM switching. The Internet Engineering Task Force's (IETF's) classical-IP-over-ATM model treated IP as an overlay above ATM and defined logical IP subnets (LISs) over an ATM network [1]. This simple overlay approach allowed IP and ATM to work together without changes to either protocol, but did not take advantage of the strengths of ATM. Also, the approach was difficult to scale to many routers and was inefficient in certain aspects. The ATM Forum pursued an overlay approach with LAN emulation (LANE) and later multiprotocol over ATM (MPOA). The approaches used servers for address mapping and routing, and did not take advantage of QoS capabilities in ATM.

The recent multiprotocol label switching (MPLS) approach is a convergence of various implementations of "IP switching" that use ATM-like *label swapping* to speed up IP packet forwarding without changes to existing IP routing protocols [2, 3]. Toshiba's Cell Switch Router (CSR) proposal in 1994 was perhaps the first proposal for an ATM switch that could be controlled by IP protocols rather than ATM signaling protocols. A CSR appears to be an IP router, but can select a flow for cut-through switching at the ATM layer to the next CSR. Ipsilon's proprietary IP Switch was essentially an ATM switch fabric controlled by an external switch controller running IP protocols. This was an example of the *data-driven* approach where persistent flows are automatically redirected through the ATM fabric. Cisco Systems' Tag Switching, an example of a *control-driven approach*, added a few innovations such as forwarding equivalence classes (FECs), a tag distribution protocol, and stacked tags. IBM's Aggregate Route-Based IP Switching (ARIS) was similar to Tag Switching as a control-driven approach, but more specifically designed for ATM switching. It used the approach of initiating a label-switched path by the egress router, and propagating label binding information in the backward direction.

These different implementation approaches led to the formation of the IETF's MPLS working group in 1997 to establish common agreements on the base technology for label-switched IP routing. Although better scalability and faster packet forwarding performance are the most obvious motivations behind MPLS, attention is also focusing on traffic engineering and new routing functionalities not possible with conventional IP routing (see the other articles in this Feature Topic). This article examines issues related to providing reliable services and proposes the use of the Label Distribution Protocol (LDP) for fast fault recovery and network monitoring. First, MPLS and LDP are reviewed. A method for fault recovery using modified LDP messages is proposed. Finally, methods for traffic monitoring and performance monitoring are described.

MPLS

PACKET FORWARDING

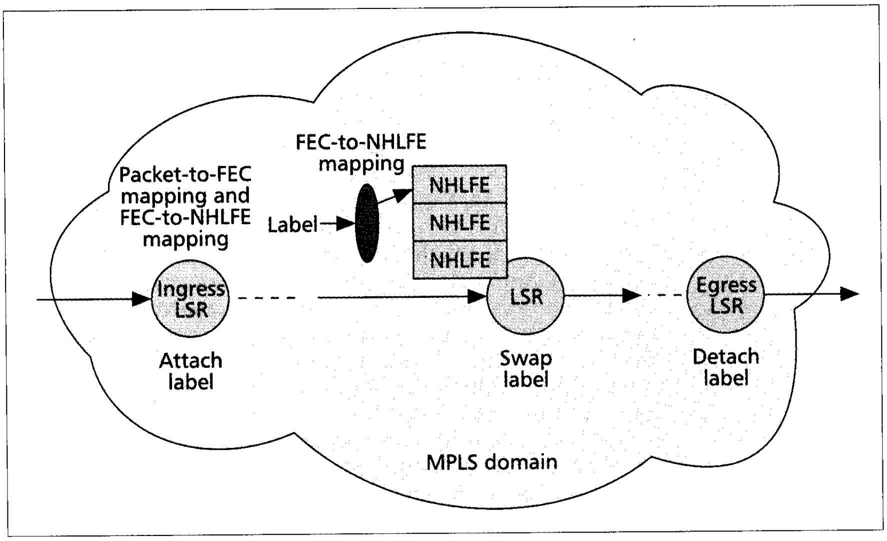
A key concept in MPLS is the separation of an IP router's functions into two parts: forwarding and control. The forwarding part, responsible for how data packets are relayed between IP routers, uses label swapping similar to ATM switching using virtual path/virtual channel identifiers, but actually the concept of FECs is more general than virtual paths/channels. A label is a short fixed-length number independent of the network layer (e.g., a label does not include any network layer addresses) [4, 5]. The label swapping technique essentially involves a table lookup of a packet's label to determine its route and new label value. Label swapping is considerably simpler than normal datagram processing involving longest prefix matching, and thus improves price/performance and scalability. A router capable of MPLS is a *label switching router* (LSR), and a set of LSRs traversed by a packet is called a *label-switched path* (LSP). A contiguous set of LSRs under a single administration constitutes an MPLS domain. A packet is forwarded across an MPLS domain based only on its label.

A specific label format is not mandated by MPLS specifications because MPLS is intended to work over any layer 2 protocol. Instead, label encoding is based strictly on mutual agreement between two neighboring MPLS-enabled routers and has meaning only on the particular link between them. The label can use an existing layer 2 header field (e.g., the VPI/VCI field in the ATM cell header) or be inserted between the layer 2 and IP headers as a small *shim* label. A shim label might consist of a 20-bit label value, 3-bit class of service, 1-bit bottom of stack indication, and 8-bit time-to-live (TTL) to prevent accidental looping [2]. In any case, MPLS may be viewed as a protocol layer between the data link and network layers.

MPLS allows hierarchical labels supported as a last in first out (LIFO) *label stack* [4]. A packet is always processed based on the top label regardless of other labels that may be below it. In a label stack, the label at the bottom of the stack is called the *level 1 label*, and labels above it are numbered consecutively up to the level *n* label (then the label stack has depth *n*). After the top label is processed, a router may pop and/or push the label stack.

As mentioned earlier, FECs are a more general concept than virtual connections. All packets can be divided into subsets called FECs based on IP source address, destination address, IP protocol, TCP/UDP source/destination ports, TTL, or type of service (TOS) fields. Because the mapping of

0163-6804/99/$10.00 © 1999 IEEE



■ **Figure 1.** *Packet forwarding in an MPLS domain.*

packets to FECs may be general, FECs allow a wide range of different granularities for packet forwarding. For example, a coarse-grain FEC may be chosen to consist of all packets with the same destination address. A fine-grain FEC might be packets belonging to a particular application running between two hosts. Coarse-grain FECs allow the overall system to be scalable to large networks, where it is useful to handle large bundles of flows as a single class of traffic. Coarse-grain FECs are also useful for reliability, allowing bundles of flows to be rerouted as a single bundle around a fault. On the other hand, FECs also allow finer differentiation of traffic so that individual flows may be treated differently, say, for different QoS handling or routing of traffic flows.

The mapping of packets to an FEC is performed only once when packets enter an MPLS domain. Subsequently, packets are processed and forwarded strictly according to their labels, and there is no need to reexamine the network layer packet header. The label is removed by the egress LSR. A label essentially serves as an index into a LSR's forwarding table, as shown in Fig. 1. The next-hop label forwarding entries (NHLFEs) in the forwarding table contain information needed for handling a packet, including:

- Outgoing interface (next hop)
- Outgoing label (or push/pop the label stack)
- Data link encapsulation (optional)
- Information about resources (optional)
- Packet handling policies (optional)

An LSR maintains an FEC-to-NHLFE mapping which will associate an incoming labeled packet with an NHLFE in the forwarding table [4]. A mapping is needed because multiple NHLFEs might exist for an FEC in the forwarding table. For a particular packet the FEC-to-NHLFE mapping will select only one NHLFE, but the mapping may be changed for various reasons, such as load balancing over multiple paths or rerouting from a failed path to an alternate path. .

QoS Routing

The control part of MPLS consists of network layer routing protocols to distribute routing information between LSRs, and label binding procedures for converting this routing information into the forwarding tables needed for label switching. MPLS is designed to work with the existing Internet routing protocols such as Open Shortest Path First (OSPF) or Border Gateway Protocol (BGP). Because MPLS allows traffic engineering and explicit routing, there is keen interest in QoS routing which allows selection of routes subject to QoS requirements (and possibly additional constraints such as policy) instead of simply the least cost or shortest route found by traditional routing protocols [6]. In addition to topological information, QoS routing requires additional information about the availability of resources in the network and QoS requirements of each flow. At the minimum, routers will need to advertise the available link bandwidth and possibly additional information such as packet delay, delay jitter, and packet loss ratio (or reliability). QoS routing may be implemented by extensions of traditional routing protocols such as OSPF [7].

A QoS routing protocol for identifying the best route works in combination with a signaling protocol for reserving needed resources along a selected route. Two alternative methods have been identified (both currently allowed in MPLS): Resource Reservation Protocol (RSVP)-MPLS and a label distribution protocol [8, 9]. RSVP has become accepted as a signaling protocol in the IETF's integrated services framework and can be extended to establish LSPs. In RSVP-MPLS, the sender first transmits a Path message to the receiver with a description of traffic characteristics. In response, the receiver returns a Resv message to request resources for the flow. Each node along the route has an opportunity to accept or reject the Resv message. If the request is rejected, the node will send an error message to the receiver to terminate the signaling process. If the request is accepted, the relevant flow state information will be installed to reserve resources at each node. To work with MPLS, the first LSR inserts a Label_Request object into the Path message to request a label binding [8, 10]. If an Explicit_Route object is added to the Path message, the Path message will be forwarded along a specific route. The last LSR will return a Resv message including a Label object. As the Resv message travels upstream (i.e., upstream relative to the direction of data in the LSP being established), each LSR will receive a label and record it in the forwarding table for the new LSP, and forward a chosen label to the next upstream LSR. By piggybacking label binding information on RSVP messages, resources can be reserved for an LSP at the same time as label assignments.

Label Distribution Protocol

It is feasible and perhaps advantageous to piggyback label binding information on an existing protocol such as RSVP or BGP. An alternative is a separate LDP or an LDP with constraint-based routing (CR-LDP) designed specifically for LSRs to exchange label binding information [9, 11]. Here we focus on the LDP because later discussions will propose extensions for fault recovery and network monitoring.

Four classes of LDP messages serve different purposes:

- Discovery messages advertise the presence of LSRs.
- Session messages establish and maintain LDP sessions.
- Advertisement messages create, change, and delete label mappings for FECs.
- Notification messages carry advisory and error information.

All LDP messages have the format shown in Fig. 2. If an LSR does not recognize a message, the U (unknown message) bit tells the LSR whether to notify the sender. The 15-bit message type field identifies an LDP as one of 10 defined types:

- Hello message for LDP discovery
- Initialization message for LDP session establishment
- Keep Alive message to maintain the continuity of an

Bits:	1	15	16	32	Variable	Variable
	U	Message type	Message length	Message ID	Mandatory parameters	Optional parameters

■ **Figure 2.** *The format of LDP messages.*

Bits:	1	1	14	16	Variable
	U	F	Type	Length	Value

■ **Figure 3.** *TLV encoding.*

LDP session in absence of other messages

- Address message to advertise interface addresses
- Address Withdraw message to withdraw previously advertised interface addresses
- Label Mapping message to advertise label bindings
- Label Request message to request a label binding for an FEC
- Label Withdraw message to break a previously established FEC label mapping
- Label Release message to free an FEC label mapping
- Notification message to give advisory or error information about various events

The 16-bit message length field is the total length of the message in bytes. The 32-bit message ID is a number that uniquely identifies the particular message (e.g., for reference in a Notification message). The mandatory parameters are an ordered set of required fields that depend on the particular message type.

Mandatory and optional parameters use so-called type-length-value (TLV) encoding with the format shown in Fig. 3. If an LSR does not recognize the TLV, the U (unknown TLV) bit tells the LSR whether to notify the sender and ignore the entire message, or ignore the TLV and process the remainder of the message. If an LSR does not recognize the TLV and the message is to be forwarded, the F (forward unknown TLV) bit tells the LSR whether to forward the unknown TLV. The 14-bit type field indicates one of seven defined TLV types: FEC, Label, Address List, COS (class of service), Hop Count, Path Vector, or Status. The 16-bit length field is the length of the value field in bytes. The variable-length value field is a number interpreted according to the TLV type.

Two LSRs must begin a bidirectional LDP session to exchange label information as LDP peers. LSRs learn about the presence of direct neighbors through a basic discovery procedure [11]. LSRs periodically send Hello messages carrying the LDP identifier the LSR intends to use for the interface. Discovery of a neighbor will initiate an LDP session. First, the two LSRs will open a TCP connection (all LDP messages except discovery messages use TCP for reliability). Next, they will exchange Initialization messages to negotiate session parameters such as LDP protocol version (currently 1), label distribution method, timer values, and VPI/VCI ranges (if layer 2 is ATM). An LDP session is finally established by a Keep Alive message. Hello messages must be exchanged periodically to maintain a label space and peer relationship. The absence of a Hello indicates that a peer LSR wishes to terminate using the agreed upon label space or that the peer has failed. In either event, the LDP session is terminated. Also, in the absence of other messages, Keep Alive messages should be sent regularly to maintain a session.

During an LDP session label bindings are assigned by the downstream LSR, and label assignments are distributed from the downstream LSR to the upstream LSR. That is, label assignment information flows in the opposite direction to the data packets. This can be done automatically by a downstream LSR advertising to its neighbors via a Label Mapping message without a label request (called *downstream unsolicited distribution*), or on demand when an upstream LSR requests a label assignment from a downstream LSR via a Label Request message (called *downstream-on-demand label distribution*). Both modes of distribution can be used in the same MPLS domain, but adjacent LSRs must agree on one mode between them.

Also, distribution control can be either independent or ordered. In independent control, each LSR may advertise label mappings to its neighbors at any time. Each LSR makes an independent decision to bind a label to an FEC and distribute that binding to LDP peers. An upstream label can be advertised before a downstream label is received. In ordered control, label distribution for a flow is initiated by the egress LSR. An LSR must wait until a label is received from a downstream LSR (unless it is the egress LSR for that FEC). Ordered control must be used if the traffic for a particular FEC is to follow a path with some specified set of properties. In independent control some LSRs may begin label switching before the LSP is completely set up, and the path may not have specified a set of properties.

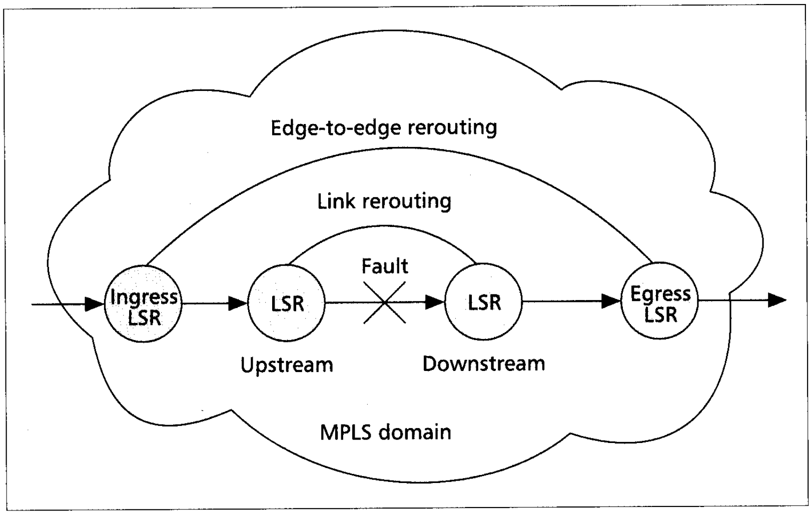
LPD may be modified to reserve resources along an explicit path as well as distribute label information [9]. A Label Request message is sent from the ingress LSR through a specified sequence of LSRs to the egress LSR. In addition to a CoS request, the message may contain a TLV for traffic parameters such as peak rate, peak burst size, and committed rate. If an LSR can accomodate the new connection, it will reserve the corresponding resources and forward the Label Request message to the next LSR. Upon receiving the message, the egress LSR will return a Label Mapping message in the upstream direction to the ingress LSR.

Reliability

The current Internet inherently has a degree of survivability due to the connectionless IP protocol. Dynamic routing protocols are designed to react to faults by changing routes when routers learn about topology changes via routing information updates (e.g., link status advertisements). Loss of QoS has not been an issue because current Internet traffic is best-effort. In contrast, the MPLS approach is connection-oriented, which implies greater potential vulnerability to faults. At the same time, MPLS will support integrated services, which are more sensitive to loss of service. Reliability is becoming more important as more users depend on the Internet for critical communication services and expect a higher level of performance.

In practice, fault restoration capabilities are implemented in multiple protocol layers, such as automatic protection switching in the physical transmission layer, self-healing in the ATM virtual path layer, and fast rerouting in MPLS. Usually, fault recovery is attempted first at the lowest layer, and escalated to the next layer if recovery was unsuccessful or not possible. Since MPLS resides between layers 2 and 3, it may be assumed that fault recovery at layer 2 will be given a chance before MPLS fault recovery. Fault recovery capabilities in the MPLS layer are needed as well to decouple MPLS from dependence on physical layer fault recovery mechanisms which may differ between networks.

Traditionally, faults trigger alarms to a centralized network manager who reconfigures traffic around the fault. Clearly,



■ **Figure 4.** *Link rerouting and edge-to-edge rerouting.*

manual reconfiguration may be too slow for critical traffic, and centralized control may be less dependable than distributed control. For distributed control, automated fault recovery functions must be delegated to the LSRs in an MPLS domain. After a fault is detected, the LSRs will automatically carry out procedures for:

- Fault notification to all affected LSRs
- Search for an alternate path for the affected traffic
- Rerouting to the alternate path
- (Optional) redistribution of the network traffic to ensure that capacity will be available to recover from subsequent faults

It is possible to use modified RSVP for rerouting [10] or LDP/CR-LDP for fault recovery, but only LDP/CR-LDP is discussed here.

Fault notification can be done by LDP Notification messages, which are intended to signal important events. In particular, a Notification message carries a Status TLV that, in general, indicates the type of event, whether it is a fatal error or not, whether to forward it to the next hop, and status of the event. A new Status TLV for fault notification could identify the failed link, cause of the failure (if known), and the effected FEC.

The Notification message is generated by the LSR on the downstream end from where the fault is detected (or LSRs on both sides of the fault if the link is bidirectional). The recipient of the Notification message depends on whether the fault recovery is being done by link rerouting or edge-to-edge rerouting. In link rerouting, an alternate path is found between the two LSRs on the ends of a failed link, as shown in Fig. 4. This approach has the advantages of relative simplicity and speed because the downstream LSR must only notify the upstream LSR. For fast recovery, the alternate path may be pre-established based on the most recent routing information, in which case the upstream LSR already has an NHLFE for the alternate path in its forwarding table. Rerouting is accomplished by a simple change in the upstream LSR's FEC-to-NHLFE mapping. Resources may be reserved along the alternate path for reliable recovery (perhaps only for guaranteed traffic), but might not be reserved for more efficient resource utilization. If not reserved, there is no guarantee that the alternate path will be available or capable of sustaining the desired QoS at the time it is needed. The Notification message should check the availability of resources along the pre-established alternate path as it travels to the upstream LSR. To maximize the probability of success, alternate paths may be recomputed regularly in the background so that the best alternate path is always selected and up to date.

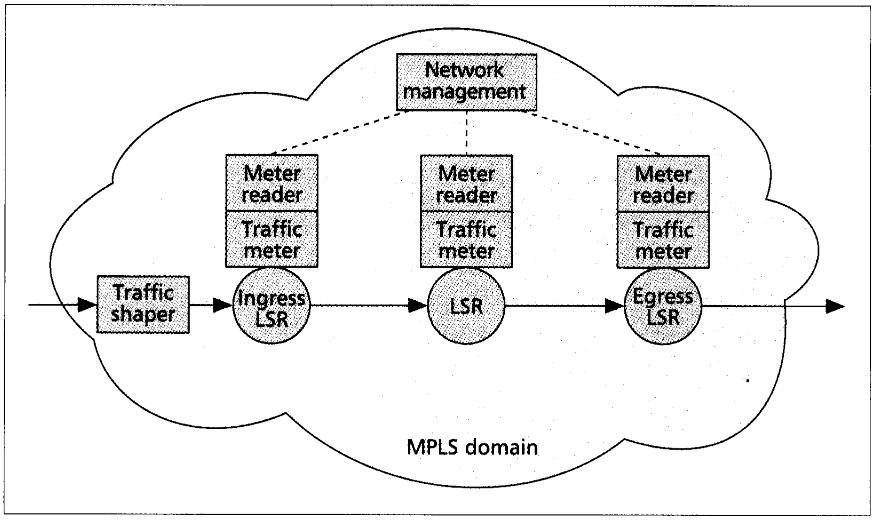
Alternatively, an alternate path may be sought dynamically after fault notification. If the downstream LSR has QoS routing information, it may select a feasible alternate path and send a modified Label Mapping message or signaling message to create label bindings and reserve resources along the selected alternate path. In the unlikely case that no routing information is available, the downstream LSR might search for an alternate path by flooding Notification messages to the upstream LSR, which can choose among the paths found by the messages that reach it successfully. Generally, flooding is not preferred due to the additional overhead and delay incurred by flooding messages, but it will work in the absence of routing information.

In any case, link rerouting has the disadvantage of more difficulty in handling node failures or multiple link failures. Edge-to-edge rerouting is a more complex approach which finds an alternate path between the ingress and egress LSRs that is completely disjoint from the failed path. It has the advantage of being more capable of handling node failures or multiple link failures. Unfortunately, each effected FEC may involve different ingress/egress LSR pairs and hence must be rerouted individually, which argues for the use of coarse-grain FECs or LSP merging to minimize the number of FECs to reroute. The downstream LSR can notify the egress LSR with an LDP Notification message carrying a Status TLV indicating the failed link, cause of failure (if known), and the effected FEC to reroute.

Again, the alternate path may be pre-established for efficiency and minimal recovery time. This approach is similar to ATM self-healing. For edge-to-edge rerouting, pre-established alternate paths may be the preferred approach over flooding because the overhead and delay incurred by flooding may be greater disadvantages when longer alternate paths are sought. Assuming that sufficient capacity has been planned for survivability, QoS routing should identify multiple feasible paths, and the working path is established first to avoid possible conflicts with simultaneous establishment of the alternate path. After the working path has been established, the alternate path may be selected and established by creating label bindings and NHLFEs at all LSRs along the alternate path. Resources may be reserved along the alternate path if reliable recovery is desired (perhaps only for guaranteed traffic). If resources are not reserved, there is no guarantee that the alternate path will be available or capable of sustaining the desired QoS at the time it is needed. The probability of success should be maximized by regularly checking candidate alternate paths and recomputing the best alternate path in the background.

MONITORING

Network monitoring is useful for several reasons: to verify QoS, verify connectivity, check the status of candidate alternate paths, and seek early indications of imminent faults or performance troubles. A rudimentary means of network monitoring has long been available as part of network management using Simple Network Management Protocol (SNMP) [12]. With SNMP, network managers can essentially poll nodes for status or operational parameters, or receive alarms for pre-specified trouble conditions. For traffic monitoring, the IETF has extended the SNMP paradigm to include real-time flow measurement (RTFM) *traffic meters*, as shown in Fig. 5 [13]. Traffic meters are situated around the network, capable of observing flows of packets that pass through them. A traffic meter can be configured to selectively observe a specific packet flow and its various attributes defined by rules provided by a network manager. The specified attributes of the flow (e.g.,



■ **Figure 5.** *Traffic monitoring in an MPLS domain.*

number of packets or bytes observed) are recorded in a database which can be retrieved by *meter readers*. In turn, applications can fetch data from meter readers through regular FTP or SNMP protocols.

In MPLS, FECs provide a natural definition of a flow, and packets belonging to the same FEC can easily be identified by their label values. Traffic meters can simply examine the MPLS labels to classify packets in an MPLS domain. The RTFM architecture may provide a convenient and useful means to monitor the traffic along an LSP. For example, traffic meters at each LSR in an LSP can record the observed throughput of an FEC. By comparing the throughput at each LSR, network managers can determine whether packets are experiencing any congestion or packet loss, and identify the location of congestion. Unfortunately, traffic monitoring cannot provide complete measurements about QoS, such as edge-to-edge delay. Delay measurements may be made by the layer 2 protocol (e.g., operation and maintenance, OAM, cells in ATM). Traffic monitoring may use traffic shapers (e.g., token leaky buckets) at the ingress to an MPLS domain to monitor the incoming traffic flow rates and possibly enforce conformance of traffic flows to their specified characteristics by smoothing out traffic bursts. Traffic shaping can provide monitoring and rate enforcement without changes to existing protocols.

Conclusions

We believe that MPLS is a promising approach to IP switching, but reliability is a practical issue that may slow the wide adoption of MPLS in the Internet. For fault recovery, we suggest fast rerouting techniques using modified LDP messages. Also, we propose methods for traffic monitoring to collect feedback information about network conditions. With more monitoring and accurate knowledge of traffic, traffic engineering should be more effective and efficient.

References

[1] M. Laubach, "Classical IP and ARP over ATM," Internet RFC 1577, Jan. 1994.

[2] A. Viswanathan *et al.*, "Evolution of Multiprotocol Label Switching," *IEEE Commun. Mag.*, vol. 36, May 1998, pp. 165–73.

[3] B. Davie, P. Doolan, and Y. Rekhter, *Switching in IP Networks*, Morgan Kaufmann, 1998.

[4] E. Rosen, A. Viswanathan, and R. Callon, "Multiprotocol Label Switching Architecture," Internet draft draft-ietf-mpls-arch-06, Aug. 1999.

[5] R. Callon *et al.*, "A Framework for Multiprotocol Label Switching," Internet draft draft-ietf-mpls-framework-05, Sept. 1999.

[6] N. Feldman *et al.*, "MPLS Constraint-Based Routing," this issue.

[7] R. Guerin, A. Orda, and D. Williams, "QoS Routing Mechanisms and OSPF Extensions," *GLOBECOM '97*, 3–8 Nov., 1997, pp. 1903–8.

[8] D. Awduche *et al.*, "Extension to RSVP for Traffic Engineering," Internet draft draft-swallow-mpls-RSVP-trafeng-00, Aug. 1998.

[9] B. Jamoussi, Ed., "Constraint-Based LSP Setup Using LDP," Internet draft draft-ietf-mpls-cr-ldp-02, Aug. 1999.

[10] D. Awduche *et al.*, "Extensions to RSVP for LSP Tunnels," Internet draft draft-ietf-mpls-rsvp-lsp-tunnel-03, Sept. 1999.

[11] L. Andersson *et al.*, "LDP Specification," Internet draft draft-ietf-mpls-ldp-05, June 1999.

[12] M. Schoffstall *et al.*, "A Simple Network Management Protocol (SNMP)," Internet RFC 1157, May 1990.

[13] N. Brownlee, C. Mills, and G. Ruth, "Traffic Flow Measurement: Architecture," Internet RFC 2063, Jan. 1997.

Additional Reading

[1] R. Kawamura, K.-I. Sato, and K. Tokizawa, "Self-Healing ATM Networks Based on Virtual Path Concept," *IEEE JSAC*, vol. 12, Jan. 1994, pp. 120–27.

Biographies

Thomas M. Chen (tchen@seas.smu.edu) is an associate professor in the Department of Electrical Engineering at Southern Methodist University, Dallas, Texas. He received B.S. and M.S. degrees in electrical engineering from MIT, and a Ph.D. degree in electrical engineering from the University of California, Berkeley. From 1989 to 1997 he worked on ATM research at GTE Laboratories, Inc., Waltham, Massachusetts. He is a senior technical editor for *IEEE Network*, a technical editor for *IEEE Communications Magazine*, and editor of *ComSoc e-News*. He received the IEEE Communication Society's Fred W. Ellersick best paper award in 1996. He is co-author of the monograph *ATM Switching Systems* (Artech House, 1995) and co-contributed a chapter to the upcoming *ATM Handbook* (McGraw-Hill).

Tae H. Oh received a B.S.E.E. degree from Texas Tech University in 1990 and an M.S.E.E. degree from Southern Methodisty University in 1994. He is currently pursuing his Ph.D. at SMU. From 1990 to 1994 he was with the Electrospace System Inc., where he was involved in the design of RF communication systems for military vehicles. From 1994 to 1998 he worked at Nortel Networks, where he developed call processing software. His current research interests include MPLS, constraint-based routing, IP, and diffserv.