Management of DiffServ-over-MPLS Transit Networks with BFD/OAM in ForCES Architecture[†]

Seung-Hun Yoon, Djakhongir Siradjev, Young-Tak Kim[‡]

Dept. of Information and Communication Engineering,
Graduate School, Yeungnam University
214-1, Dae-Dong, Kyungsan-Si, Kyungbook, 712-749, KOREA
bthuni@yumail.ac.kr, m0446086@chunma.yu.ac.kr, ytkim@yu.ac.kr

Abstract. This paper proposes a management of DiffServ-over-MPLS transit network with BFD(Bidirectional Forwarding Detection)/OAM (operation, administration and maintenance) in ForCES (Forwarding and Control Element Separation) architecture for QoS-guaranteed DiffServ-over-MPLS traffic engineering. The proposed BFD and ForCES functions are implemented on Intel 2400 network processor, where BFD/OAM packets for MPLS TE-LSP are exchanged every 5 ~ 10 ms interval for performance measurements and link failure detection. The operations of BFD/OAM-based fault detection and performance measurement are controlled via distributed control plane with ForCES (forwarding and control element separation) architecture for large scale IP/MPLS router using multiple network processors in each network interface card. We explain the implementation details of ForCES-based distributed control plane functions, hierarchical traffic grooming with label stacking, and BFD/OAM mechanisms. The link failure detection performance of BFD/OAM functions for MPLS TE-LSP is evaluated.

Keywords: DiffServ-over-MPLS, QoS, ForCES, BFD, OAM, Network Processor

1. Introduction

In next generation Internet, various QoS-guaranteed realtime broadband multimedia services, such as video telephony, multimedia teleconference, IP-TV and video-on-demand, should be provided based on IP/DiffServ-over-MPLS transit networks with efficient traffic engineering [1]. For end-to-end QoS-guaranteed multimedia service provisioning, the virtual overlay transit network for each DiffServ class-type must be continuously monitored for available bandwidth and edge-to-edge packet delivery performance, such as delay, jitter, and packet loss/error rate[1].

IETF BFD (Bidirectional Forwarding Detection) has been designed to detect faults in the bidirectional path between two forwarding entities with protocol independent to physical layer and path types [2-5]. BFD also provides the continuity checking functions of data link layer as OAM (operation, administration and maintenance), and

[†] This research was supported by the MIC, under the ITRC support program supervised by the IITA.

[‡] Corresponding author.

the link management protocol (LMP) of WDM optical link. The most important function of BFD is protocol-independent fast detection of data link failure on any kind of path between two nodes, including direct physical links, virtual circuits, tunnels, MPLS LSPs, multi-hop routed paths, and uni-directional links, so long as there are some return paths. Except SONET/SDH transmission systems, fast link failure detection and fault restoration are not mostly supported by physical layer. In order to provide link failure detection and fault restoration within 50 ms (as in the automatic protection switching of SONET/SDH), the BFD/OAM continuity check must be performing at 5 ~ 10 ms interval, and dedicated hardware functions of network processor are required.

IETF ForCES (forwarding and control element separation) standards [6-9] aim to define a framework and associated mechanisms for exchange of information between the logically separate functionality of the control plane (including routing protocols, admission control, and signaling) and the forwarding plane (including fast packet processing such as packet forwarding, queuing, and header editing). The standard separation mechanism of ForCES allows the control plane and forwarding plane to innovate in parallel while maintaining interoperability [5]. In distributed/parallel IP/MPLS packet switching architecture, the control plane functions and the data forwarding plane functions should be carefully distributed to increase the processing capacity by parallelism while minimizing the inter-module communication overhead.

In this paper, we design and implement the management functions of DiffServ-over-MPLS transit network with BFD/OAM in ForCES architecture for QoS-guaranteed broadband realtime multimedia service provisioning. The proposed BFD and ForCES functions are implemented with Intel 2400 network processor, where BFD/OAM packets for MPLS TE-LSP are exchanged every 5 ~ 10 ms for performance measurements and link failure detection. The operations of BFD/OAM-based fault detection and performance measurement are controlled via distributed control plane with ForCES architecture for large scale IP/MPLS router using multiple network processors in each network interface card (NIC).

The rest of this paper is organized as follows. Section 2 describes the related work on BFD/OAM, ForCES, and distributed OSPF function. In Section 3, we explain the implementation of BFD/OAM functions for DiffServ-over-MPLS transit networks with hierarchical TE-Links. Section 4 analyzes the overall performance of the proposed BFD/OAM functions for DiffServ-over-MPLS transit networks, and finally we conclude this paper in section 5.

2. Background

2.1 Bidirectional Forwarding Detection

BFD is a protocol intended to detect faults in the bidirectional path between two forwarding engines, including physical interfaces, subinterfaces, data link(s), and to the extent possible forwarding engines themselves, with potentially very low latency[2-5]. It operates independent of transmission media, data protocols, and routing protocols. An additional goal is to provide a single mechanism that can be used for continuity checking and QoS measurement (including delay, jitter, and packet error/loss) over any transmission media, at any protocol layer, with a wide range of detection times and overhead, to avoid a proliferation of different methods.

BFD includes following important characteristics [2]: i) It must be simple, fixed-field encoding to facilitate implementations in hardware, ii) It should be independent of the data protocol being forwarded between two systems; BFD packets are carried as the payload of whatever encapsulating protocol is appropriate for the medium and network, iii) BFD must be path-independent. BFD can provide failure detection on any kind of path between systems, including direct physical links, virtual circuits, tunnels, MPLS LSPs, multi-hop routed paths, and unidirectional links, so long as there is some return paths. BFD also provides the continuity checking functions of data link layer as OAM, and the link management protocol (LMP) of WDM optical link.

2.2 ForCES (Forwarding and control element separation)

IETF ForCES aims to define a framework and associated mechanisms for standardizing the exchange of information between the logically separated functionality of the control plane (including routing protocols, traffic engineering link maintenance, admission control, and signaling) and the forwarding plane (including per-packet processing, packet forwarding, queuing, and protocol data unit (PDU) header editing). Fig. 1 shows examples of control elements (CEs), forwarding elements (FEs), and their interactions using ForCES protocol. Having standard mechanisms allows CEs and FEs to be developed by different vendors and interoperate with each other [6-9]. ForCES will enable rapid innovation in both control and forwarding planes while maintaining interoperability. Scalability is also easily provided by ForCES architecture where additional forwarding or control capacity can be added to existing network elements without the needs of big change in system architecture.

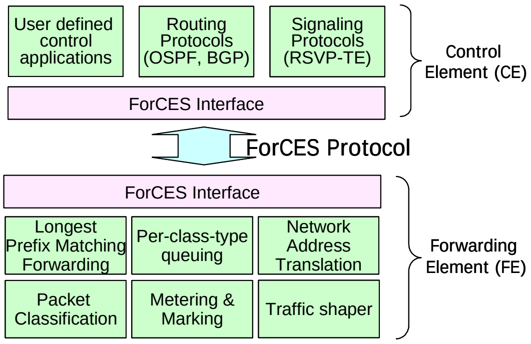


Fig. 1. Examples of control elements, forwarding elements and interactions with ForCES protocol

In ForCES architecture the physical forwarding elements may be implemented by using multiple network processors, ASICs, general purpose processors, installed on line cards, daughter boards, mezzanine, or stand-alone boxes. The control element and the forwarding element may be in close proximity (same room or small number of hops) or in very short distance (same box or single hop). In real implementations, the control elements may also be distributed on several functional modules for better performance. For example, the link status monitoring and update function in OSPF for multiple high-speed links in a large scale IP/MPLS router requires time consuming processing, and can be distributed to multiple network processors that control and manage multiple physical ports individually. Also, with distributed link status monitoring module, we can implement fast fault discovery and fast link restoration.

2.3 Distributed Control Plane with distributed OSPF link status monitoring supported by Bidirectional Forwarding Detection (BFD)

The control plane functions, such as IP routing protocols (OSPF, IS-IS and BGP, and MPLS signaling (RSVP-TE), are generally implemented on a centralized controller for the whole IP/MPLS router/switch. For better scalability and functionality, some part of control functions may be carefully distributed to multiple functional modules which utilize high-speed multiprocessing with network processor. As an example, the periodic link status monitoring for each data link of OSPF can be distributed to each network interface module that can exchange BFD/OAM message periodically with its neighbor network interface module.

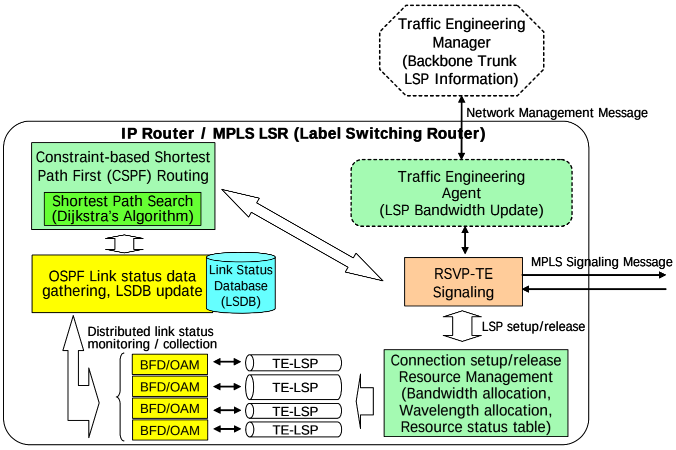

Traffic Engineering Manager (Backbone Trunk LSP Information)
Network Management Message
IP Router / MPLS LSR (Label Switching Router)
Constraint-based Shortest Path First (CSPF) Routing
Shortest Path Search (Dijkstra's Algorithm)
Traffic Engineering Agent (LSP Bandwidth Update)
OSPF Link status data gathering, LSDB update
Link Status Database (LSDB)
RSVP-TE Signaling
MPLS Signaling Message
LSP setup/release
Distributed link status monitoring / collection
BFD/OAM
BFD/OAM
BFD/OAM
BFD/OAM
TE-LSP
TE-LSP
TE-LSP
TE-LSP
Connection setup/release Resource Management (Bandwidth allocation, Wavelength allocation, Resource status table)

Fig. 2. Distributed Control Plane Architecture of IP/MPLS Router

Fig. 2 shows the functional architecture of distributed control plane, where the OSPF link status data gathering and LSDB (Link Status Database) update in a large scale IP/MPLS are distributed to multiple network interface modules that utilize high-speed packet processing and parallel processing with network processor. BFD/OAM function is implemented for each physical link or TE-LSP that is used for traffic engineering trunk. The connection setup and release function can also be partially distributed to network interface module for increased performance of control plane.

When physical layer protocol supports a well defined OAM function, such as SONET/SDH transmission system, the link status monitoring can utilize the performance measurement and fault monitoring function of the physical layer. If the physical layer protocol does not support well defined OAM functions as in Gigabit Ethernet link, however, the IP/MPLS layer protocol must implement BFD (bidirectional forwarding detection) function to detect the connectivity failure of each data link. Within the network interface module where multiple network processors might be used, the embedded processor (i.e., Xscale embedded processor in IXP2400/2800) in each network processor can execute these partial control element functions.

For fast detection of any connectivity failure in physical and logical data link protocol layer between two network interface modules, the BFD should be used. For example, in order to achieve the fault detection and recovery performance of the SONET physical layer which is limited to 50 ms, the BFD message must be periodically exchanged within every 5 ~ 10 ms, to enable the network interface module to decide logical path/link failure based on 3 consecutive BFD message losses.

3. Design of BFD/OAM for Management of DiffServ-over-MPLS Transit Network

3.1 BFD/OAM for QoS-guaranteed DiffServ-over-MPLS Service Provisioning with hierarchical traffic grooming

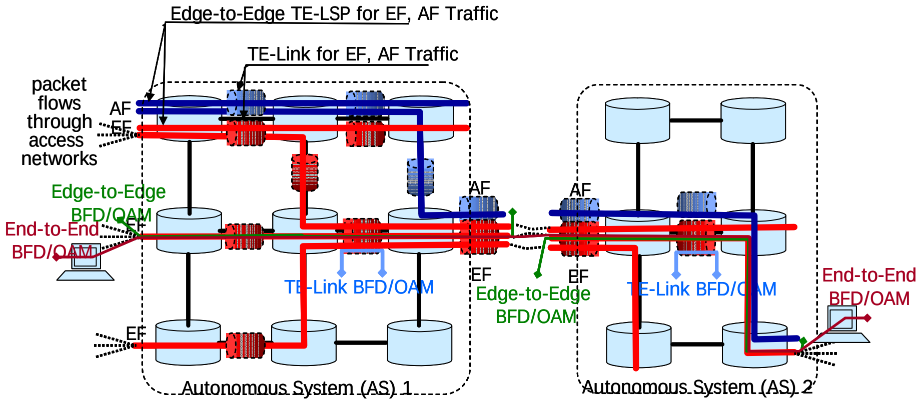


Fig. 3. Traffic grooming and associated OAM/BFD function

In order to guarantee the pre-configured QoS for DiffServ-over-MPLS, the virtual overlay network for the class-type must be continuously monitored, and the available bandwidth and edge-to-edge packet transfer delay must be continuously measured and analyzed. Fig. 3 shows the traffic grooming with hierarchical MPLS TE-LSP label stacking in DiffServ-over-MPLS virtual overlay networks, and their associated OAM functions.

Each network interface module can implement the BFD function for edge-to-edge TE-LSP for each class-type, and for the aggregated TE-LSPs of a class-type between adjacent IP/MPLS routers. BFD for each TE-LSP and TE-Link will send periodic monitoring packet with time stamp to measure the packet transfer delay, jitter (delay variation), packet error rate, and packet loss rate. The distributed control function will update the link status periodically, and allow the centralized OSPF daemon to retrieve the most up-to-date link information. If there is any abnormal condition on any TE-Link or TE-LSP, the distributed control element on network processor should inform the fault to the centralized OSPF daemon immediately.

3.2 Design of ForCES based distributed control plane

Fig. 4 depicts an example of highspeed packet switch architecture with multiple control elements and forwarding elements distributed in multiple functional modules. In this architecture a centralized controller with signaling functions will control the overall routing and switching of user packet flows. The centralized controller is usually implemented as a special control module in the router/switch node, or can be implemented as a remote control node system. Partial control functions, such as link monitoring of OSPF by BFD/OAM, may be distributed at each network interface module where forwarding elements are collocated. Multiple forwarding engines are used to support multiple physical link interfaces for 1 ~ 40 Gbps rate.

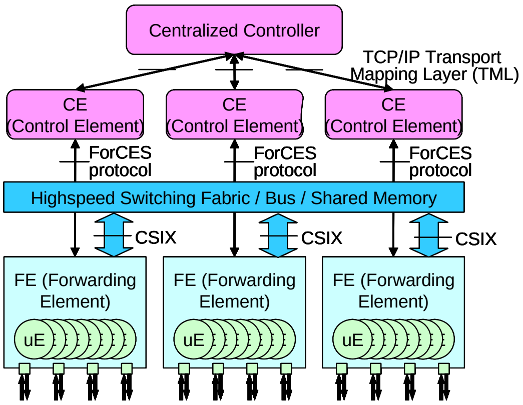


Fig. 4. Distributed control elements and forwarding elements with multiple network interface modules

One forwarding element may contain 1 ~ 2 network processors to support multiple link/port interfaces, and each forwarding element is connected to a high-speed switching fabric/bus or shared memory via CSIX (common switch interface) to support packet switching among different forwarding element modules.

The ForCES protocol provides the communication functions among CE-FE for resource discovery, establishment of associations, configuration, query and response, event notification, redirection of IP packets, and heartbeat messaging. When the partial control element is collocated with forwarding elements on a network interface module where multiple network processors are used, the communication between the partial control elements and the forwarding elements may be implemented within the same network interface module. So, the ForCES communication can be much simpler than the communication among remote systems. The communication between the centralized controller and the partially distributed control elements can be implemented with TCP/IP transport mapping layer (TML) [10].

3.3 BFD/OAM with ForCES functional blocks on IXDP2400

Fig. 5 depicts the BFD/OAM component block diagram on Intel IXDP2400 platform. The BFD/OAM configuration component provides interface functions, such as session creation/deletion, BFD/OAM activate/deactivate, performance analysis, fault detection and notification, and clock synchronization. BFD/OAM core component is using BFD/OAM session table that contains the detailed information of the BFD session for each TE-LSP. For each TE-LSP creation, the BFD session entry is created and the BFD/OAM activity is initialized as default *deactivated* state. In this state BFD/OAM does not transmit polling packets, but does respond to the polling packets transmitted from remote side. When the control plane activates the BFD/OAM function through ForCES protocol, the BFD/OAM core component starts periodically transmitting polling packets and receiving BFD/OAM respond packets.

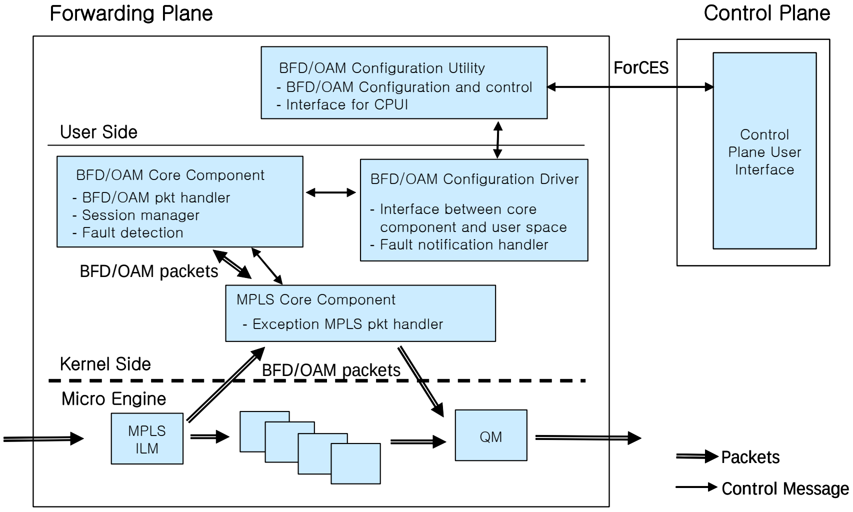


Fig. 5. BFD/OAM related functional blocks

BFD/OAM Core Component (CC) functional block diagram is shown in Fig. 6. BFD/OAM CC maintains two basic data structures: BFD/OAM session table and BFD/OAM active session list. BFD/OAM session table stores BFD related information about bi-directional link. BFD/OAM active sessions list store sessions that are currently checking their link status. BFD/OAM active sessions list entries and BFD/OAM sessions table entries are cross-linked, to avoid search of entries and allow faster processing. MPLS CC provides couple validation message handling to allow BFD/OAM CC to know whether LSPs in the couple are valid or not. Also if any LSP participating in BFD/OAM session does not exist anymore by some reason (e.g., removed), BFD/OAM CC is informed that LSP couple is not valid anymore. BFD/OAM configuration driver registers fault notification handler in BFD/OAM CC and it is called once any link changes its status.

MPLS microblock forwards all packets containing Router Alert label to the MPLS CC as exception, which forwards them to BFD/OAM CC. BFD/OAM CC packet handler processes the packet according to usual BFD processing. If received packet is polling, response is sent, if received packet is response, timestamp of last received packet and average Round-Trip Time (RTT) is updated. Also, if the link is *DOWN*, when the response is received, its status is changed to *UP* and fault notification handler is invoked to inform control plane about link status change.

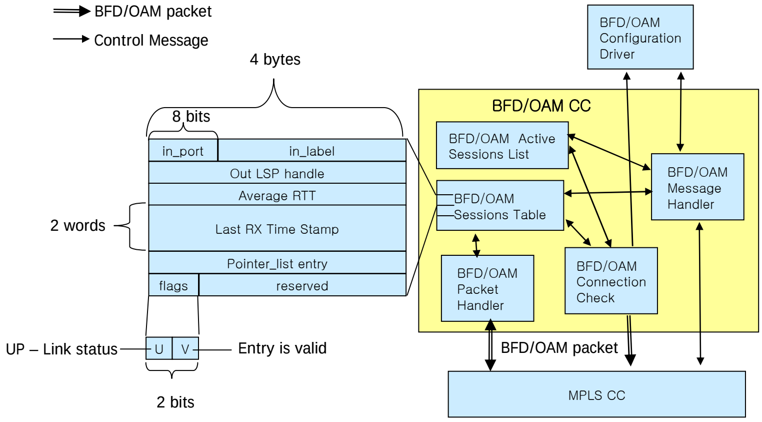


Fig. 6. BFD/OAM Core Component Functional Block Diagram

BFD/OAM core component creates a separate thread for active sessions list traversal. When the traversal starts, time is saved, and next traversal is scheduled after 5 ~ 10 ms. If the traversal takes more than 5 ~ 10 ms, next traversal is scheduled immediately after the previous is finished. Linked list entries and timestamps are accessed by one thread at the same time to avoid data corruption. During traversal BFD/OAM packets are transmitted for each active session, and also time difference between current time and last received packet timestamp is calculated. If this difference exceeds the pre-defined limit (i.e., 15 ms for BFD/OAM interval of 5 ms), connection is marked as *DOWN*, and fault notification handler is executed.

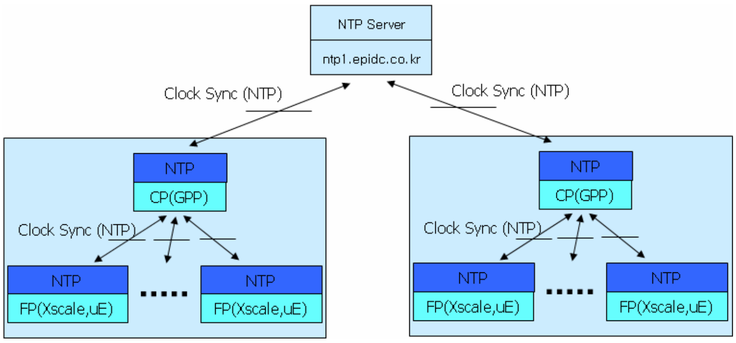


Fig. 7. NTP based clock synchronization of BFD/OAM modules

3.4 Clock Synchronization among distributed BFD/OAM modules

Clock synchronization among network processors is another important issue for correct analysis of the packet delivery delay of the link or tunnel. In order to increase the clock precision for link failure detection, the time clock of BFD/OAM transmitter and receiver must be synchronized in micro-second order.

We enhanced the network time protocol (NTP) version 4 [18] implemented in Monta Vista Embedded Linux system on IXPD2400, to synchronize the network processors. Fig. 7 shows the NTP based clock synchronization for BFD/OAM. The system clock in each network processor is synchronized with higher precision (less than 11 us) with enhanced NTP protocol with time stamp in micro-second order.

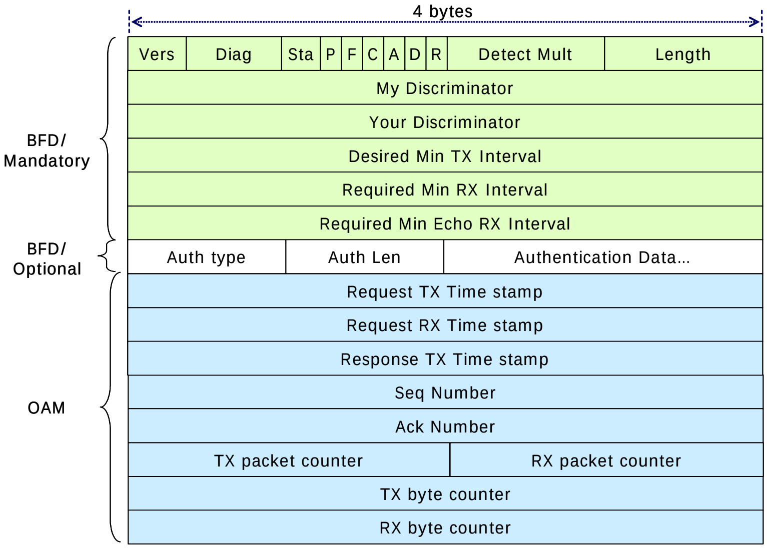
4. Implementation and Performance Analysis of Distributed Control Plane on Intel IXDP2400 Platform

4.1 Implementation of BFD/OAM on Intel IXP2400 Network Processor

In real implementation of large scale IP/MPLS router/switches, each network interface module will include 2 ~ 10 optical ports, multiple network processors, shared memory, and optional switching fabric block with CSIX (common switch interface). The network interface modules and backbone switching module will comprise the forwarding element (FE) function. The control element (CE) will be mostly implemented on the system controller board that contains signaling protocols (RSVP-TE), routing protocols (OSPF or ISIS, BGP), and open service architecture (OSA) interface. Some part of the control element function should be distributed on each network interface module to increase the scalability and fast processing. For the scalability of control plane and packet forwarding plane, the overall system must be optimized in parallelism while minimizing the inter-module communication overhead.

We implemented the proposed distributed control plane on IXDP 2400 network processor development platform and Linux host machine. Centralized control plane function of OSPF daemon is implemented on a remote Linux host machine, and BFD and OAM functional modules are implemented on the embedded Xscale processor in IXP2400. The communication between OSPF daemon and the BFD/OAM module is implemented with TCP/IP socket, and one IXP2400/2800 network processor controls 4 ports in IXDP2400 (10 ports in IXDP2800) of 1 Gbps Gigabit Ethernet interface. For each port, a dedicated thread of embedded Linux is created which periodically sends / receives BFD/OAM packet to/from its neighbor. BFD standard defines uni-directional link status monitoring with return path. In our implementation, for efficient processing of bidirectional link status monitoring and analysis, we use piggyback mechanism in BFD/OAM packet exchange.

Fig. 8 shows the BFD/OAM packet which includes fields of discriminators, Tx and Rx intervals, minimum response Rx interval, sequence number, time stamps for delay measurements, total transmitted packet count and size for packet loss/error analysis. In current implementation, the BFD/OAM packet is sent every 5 ~ 10 msec, containing the time stamps and packet transmission statistics data. The receiving thread records the arrival time of the BFD/OAM packet, checks the packet reception statistics data from the micro engine that handles the input port, compares the transmission statistics data from the BFD/OAM packet, and replies a BFD/OAM packet with piggybacked response data.



	4 bytes										
BFD/ Mandatory	Vers	Diag	Sta	P	F	C	A	D	R	Detect Mult	Length
	My Discriminator										
	Your Discriminator										
	Desired Min TX Interval										
	Required Min RX Interval										
	Required Min Echo RX Interval										
BFD/ Optional	Auth type	Auth Len	Authentication Data...								
OAM	Request TX Time stamp										
	Request RX Time stamp										
	Response TX Time stamp										
	Seq Number										
	Ack Number										
	TX packet counter	RX packet counter									
	TX byte counter										
	RX byte counter										

Fig. 8. BFD/OAM packet format

4.2 Analysis of failure detection performance with BFD/OAM

Fig. 9 shows the interaction between the control element and forwarding element with BFD/OAM function. The control element configures the operation mode of BFD/OAM function, specifying the interval of BFD/OAM packet delivery and event notification condition. The fault restoration procedure should be implemented as an additional network management function. For faster link failure detection, the BFD/OAM packet delivery interval should be shortened.

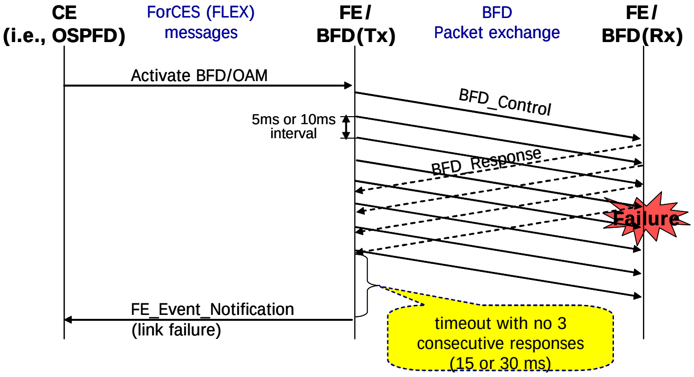


Fig. 9. Procedure of BFD packet exchange

In order to provide 50 ms link-failure restoration performance, as in SONET transmission system, we configure periodic BFD/OAM packet exchange, and if 3 consecutive BFD/OAM response packets do not arrive in expected time (i.e., 15 or 30 ms), it determines that a link failure occurred, and sends a link failure notification.

Table 1 shows the link failure detection time with BFD/OAM that has been implemented on Intel IXP2400 network processor. As shown in Fig. 5, the BFD/OAM core component periodically generate BFD/OAM packet periodically through MPLS core component that delivers the BFD/OAM packets through TE-LSP for link fault & performance management. When 3 consecutive BFD/OAM packet losses are used to indicate link failure detection, around 8 ms was taken to determine the link failure occurrence. As BFD/OAM period is reduced from 10 ms to 5 ms, the link failure detection time can be shortened from 38.029 ms to 23.095 ms.

Table 2 shows the overhead of BFD/OAM for TE-LSP. As the BFD/OAM interval is shortened, the transmission rate of BFD/OAM increases, and thus the overhead for the TE-LSP. When the TE-LSP transmission rate is more than 10 Mbps, however, the overhead of BFD/OAM with 5 ms interval is less than 1 %. Another consideration is the processing time for BFD/OAM by the network processor. In Intel IXP2400 network processor, the maximum number of active sessions is limited by 77 and 38 because of the processing speed limit of Xscale processor at the BFD/OAM interval of 10 ms and 5 ms, respectively.

Table 1. Link failure detection time with BFD/OAM

BFD/OAM period	Link failure detection time	Remark
10 ms	38.029 ms	Excluding propagation delay
5 ms	23.095 ms	Excluding propagation delay

Table 2. BFD/OAM overhead

TE-LSP Transmission Rate	Polling interval	
	10 ms	5 ms
1 Mbps	5.4400%	10.8800%
10 Mbps	0.5440%	1.0880%
100 Mbps	0.0544%	0.1088%
622 Mbps	0.0087%	0.0175%
1 Gbps	0.0054%	0.0109%
2.4 Gbps	0.0023%	0.0045%
10 Gbps	0.0005%	0.0011%
Maximum number of active sessions (IXP2400)	77	38

5. Conclusion

In this paper, we designed and implemented the management functions of DiffServ-over-MPLS transit network with BFD/OAM in ForCES architecture for QoS-guaranteed broadband realtime multimedia service provisioning. The proposed BFD and ForCES functions are implemented with Intel IXP 2400 network processor, where BFD/OAM packets for MPLS TE-LSP are exchanged every 5 ms or 10 ms for performance measurements and link failure detection. The operations of BFD/OAM-based link failure detection and performance measurement are controlled via distributed control plane with ForCES architecture for large scale IP/MPLS router using multiple network processors in each network interface card (NIC).

We analyzed the processing overhead and maximum number of active BFD/OAM session that can be configured on IXP2400 network processor. With the BFD/OAM functions with less than 5 ms interval, we could implement protocol-independent fast link failure detection within 23 ms (excluding the propagation delay) without sophisticated link failure detection in physical layer. The proposed BFD/OAM function also provides performance measurement of delay, jitter, packet loss/error for TE-LSPs in DiffServ-over-MPLS virtual overlay transit networks. The measure QoS parameters of TE-LSPs are used in the constraint-based shortest path first routing for QoS-guaranteed multimedia service provisioning across multiple domain networks.

References

1. Young-Tak Kim, Hae-Sun Kim, and Hyun-Ho Shin.: Session and Connection Management for QoS-guaranteed Multimedia Service Provisioning on IP/MPLS Networks. Proceedings of ICCSA2005 (LNCS 3481). (2005) 157 ~ 168
2. IETF Bidirectional Forwarding Detection (bfd) working group.: http://www.ietf.org/html.charters/bfd-charter. html
3. D. Katz, et. al.: Bidirectional Forwarding Detection. IETF Internet Draft (2005)
4. D. Katz, et. al.: BFD for IPv4 and IPv6 (Single Hop). IETF Internet Draft (2005)
5. D. Katz, et. al.: BFD for Multihop Paths. IETF Internet Draft (2005)
6. L. Yang, et. al.: ForCES Architecture Framework. IETF RFC 3746 (2004)
7. A. Doria.: ForCES Protocol Specification. IETF Draft, draft-ietf-forces-protocol-01.txt (2004)
8. A. Audu, et. al.: Forwarding and Control Element Separation IP Transport Mapping Layer. IETF Draft, draft-audu-forces-iptml-00 (2004)
9. Furquan Ansari, et. al.: ForCES Intra-NE Topology Discovery. IETF Draft, draft-ansari-forces-discovery-01.txt (2004)
10. Hormuzd Khosravi, et. al.: TCP/IP based TML (Transport Mapping Layer) for ForCES protocol. IETF Draft (2004)
11. Raul Aggarwal, et. al.: BFD for MPLS LSPs. IETF Internet Draft (2005)
12. Douglas E. Comer.: Network Systems Design using Network Processors. Prentice Hall (2004)
13. Bill Carlson.: Intel Internet Exchange Architecture and Applications. Intel Press (2003)
14. Erik J. Johnson and Aaron R. Kunze.: IXP2400/2800 Programming. Intel Press (2003)
15. Intel IXP2400/IXP2800 Network Processors – Microengine C Language Support Reference Manual (2003)
16. Intel Internet Exchange Architecture Software Development Kit – Software Framework Installation Guide, Intel (2004)
17. Intel Control Plane – Platform Development Kit, Intel (2004)
18. Network Time Protocol (NTP) Distribution (2005)