Routing-Protocol-Less Signalling Architecture for Connection Oriented Ethernet

Ko Kikuta[1], Eiji Oki[1, 2], Naoaki Yamanaka[1]
Contact: kikuta@yamanaka.ics.keio.ac.jp

[1]KEIO University
Graduate School of Science and Technology
3-14-1 Hiyoshi, Kohoku-ku,
Yokohama, 223-8522, JAPAN.

[2]The University of Electro-Communications
Graduate School of Informatics and Engineering
1-5-1 Chofugaoka, Chofu-shi,
Tokyo 182-858, JAPAN

Abstract

Connection Oriented Ethernet is most attractive network for next-generation Carrier-Grade network. However, to establish a connection, conventional source-routing architecture such as GMPLS requires complicated processing. It increases operational cost and equipment investment of Ethernet equipment, and then eliminates the advantage of Ethernet, the simplicity and cost-effectiveness. Thus, more simple architecture to establish a connection is required. In this paper, architecture for Connection Oriented Ethernet is proposed. This architecture performs to establish a connection on cost-shortest route without routing-protocol and path computation engine. With computer simulation, this architecture achieves sufficiently low blocking probability under infrequent signaling environment.

I. INTRODUCTION

Ethernet is the most promising technology for next-generation Carrier-Grade Layer-2 network replacing Synchronous Digital Hierarchy (SDH). The simple architecture of Ethernet has facilitated to achieve high-speed and transparent communication with cost-effective equipments. Ethernet is now achieving 100Gbps ultra high-speed interfaces for Carrier-Grade network. However, Carrier-Grade network differs from Local Area Network (LAN) in many ways. It requires not only high-speed but also reliability, QoS management, control the route, fault detection and OAM. To realize Carrier-Grade Ethernet, there are several researches and standardization activities related these issues [1].

To realize Carrier-Grade Ethernet with above features, the network has to be connection-oriented. Guarantee of QoS or bandwidth will be provided by a connection established between the source node and the destination node. However, Ethernet was developed as a connection-less protocol for LAN. Every node (i.e. Ethernet Switch) does not manage any network information and any connection will never be established in the original Ethernet. Therefore the Carrier-Grade Ethernet requires an extension to Connection Oriented Ethernet with the control plane, which enables connection establishment.

One of approaches to implement control plane on Carrier-Grade Ethernet is using Generalized Multi-Protocol Label Switching (GMPLS) [2]. GMPLS is an extension of MPLS applicable for any kind of network technology on any network layer. It realizes to set up and tear down a connection, called Label Switched Path (LSP) automatically. GMPLS controlled Ethernet Label Switching (GELS) is a scheme of Connection Oriented Ethernet which employs GMPLS control plane to establish a VLAN LSP. GELS are now being standardized by IETF [3]. On the other hand, PBB-TE [4] is an approved networking standard for hierarchical design of Carrier-Grade Ethernet. In PBB-TE, instead of the eliminated MAC learning system, a configuration of forwarding tables will establish a connection. Then, an application of GMPLS to establish a connection by building the forwarding table is proposed by an IETF document [5].

Connection establishment of GMPLS is based on typical source-routing architecture. This architecture allows the source node to determine a route. In this architecture, there are three processes to establish a connection; routing process, path computation process and signaling process. In routing process, every node exchanges the connectivity information with their neighbors using routing protocol such as OSPF-TE. Then, using this information, source node computes an applicable route to destination. As soon as a route is derived, the all nodes on the route are indicated to establish a connection by signaling protocol. Each process is working individually, thus source-routing architecture have advantages in extensibility and compatibility.

On another front, the all processes in source routing architecture for Connection Oriented Ethernet are more complicated than the original Ethernet. Due to two protocols and one computation engine, this architecture increases operational cost and equipment investment. Thus, the adoption of source-routing architecture will remove the advantage of simple and cost-effective communication of Ethernet. For Connection Oriented Ethernet requiring only supporting guaranteed communication, a more simple architecture is required to establish a connection.

In this paper, the simple architecture to establish a connection for Connection Oriented Ethernet is proposed. In this architecture, signaling message is flooded and delivered to the destination node. After one process of this enhanced signaling, a connection will be established along the cost-shortest route. Routing process and path computation

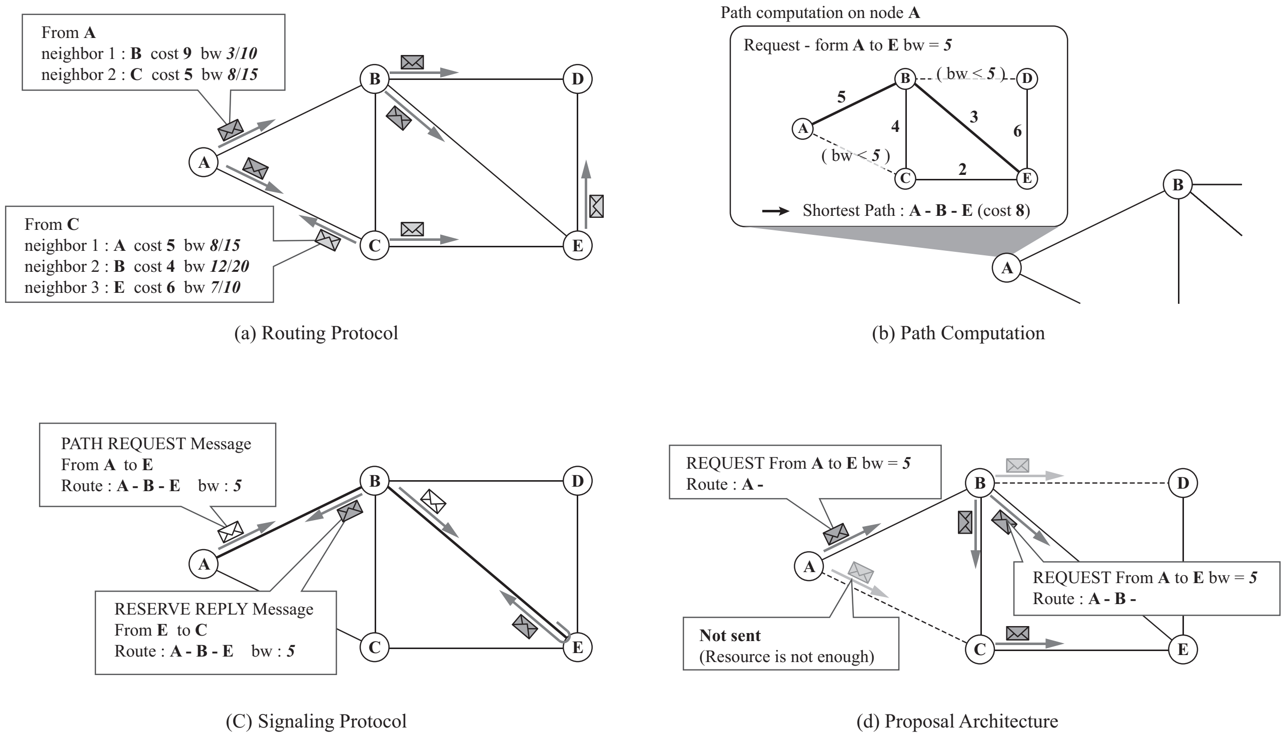

Fig. 1. Overview of source-routing architecture (a) - (c), and proposed architecture (d)

engine are not required in this architecture. This architecture works properly and achieves an equivalent performance to source-routing architecture in infrequent signaling situation.

II. Source-Routing Architecture

As described in section 1, GMPLS requires three processes to establish a new connection. In this section, these processes are described with the Figure 1.

The purpose of routing process is to give the whole topology information to all potential source nodes. After network was constructed or restructured, the concerned nodes advertise this change to the all nodes by using routing protocol as shown in Fig.1 (a). Each node makes an advertisement message that includes not only neighbor information (node ID or IP address) but also link costs and resources (bandwidth) information. Also they build a Traffic Engineering Database (TED) using this information and keep them up-to-date. To avoid the message loop and ineffectual message, routing protocol equips sophisticated features such as LSA timers running on the node.

Second process, path computation is launched when a new connection setup is requested. This process is illustrated in Fig.1 (b). A source node, which receives the request, has to find an applicable route to destination. This route satisfies the request with available resources of each link. Therefore, Constrained Shortest Path First (CSPF) algorithm is performed on the source node.

As soon as a route is computed, the signaling process is launched. A signaling message is generated and sent to the destination node as shown in Fig.1 (c). This signaling message includes the identification of signaling, route information and required bandwidth information. This message is forwarded to the next node toward the destination along the route. When the message is received, each node checks own resources again, and send it to the next node. The destination node receives this message, processes it and sends a reply message to the source node. This reply message reserves their resources. Then connection is established. The signaling process handles not only connection establishment, but also teardown of connection or other notification.

As mentioned above, source-routing architecture re- quires three individual processes. These processes cause Ethernet equipment more complicated. To achieve Connection Oriented Ethernet with characteristic cost-effectiveness of Ethernet, another simple architecture is required. In the next section, our proposal of the new architecture for connection-oriented Ethernet is introduced.

III. Proposal Architecture

A. Overview

In this section, the detail of proposed architecture is de- scribed. Because of no TED, this architecture does not require any routing protocol to keep TEDs up-to-date. Also path computation engine can be removed on any node. Although this architecture has only enhanced signaling process, the signaling message is able to find the cost-shortest route and establishes a connection along it.

The overview of proposal architecture is illustrated in Fig. 1 (d). The signaling messages will be flooded over the network from the source node. This message has the signaling identification and sum of link costs on the all links it passed. Using these values, message loop is avoided and shortest path is found. However, the role of each node in the network is very simple. Every node only records these values and incoming port number of this node on its forwarding table, then forwards the only one message which have shortest cost toward the destination. The destination node will receive the message that is routed through the shortest route. After the destination node replies this message to source node, the message is forwarded to same way, then each resource is reserved and the new connection is established.

Although this architecture allows every node to simple process, replace the complexity of source-routing architecture, a connection will be established. However, this architecture can avoid any message loop and find a cost-shortest route. Also signaling collision is resolved. These mechanisms are described in this section.

B. Building Extended Forwarding Table

In proposal architecture, signaling message has not only common information of signaling such as destination node ID and connection bandwidth, but also network-unique signaling ID and sum of link costs on the route it passed. Every node receives this message, records above information and the incoming port number into the forwarding table. The incoming port number represents the direction of the source node with cost-shortest way. This information will be used when the reply message is sent to the source node. As well as the forwarding table of the original Ethernet, each entry will be deleted when the aging time elapsed.

If there is an entry of the same signaling ID on the table, the sum link costs in both received message and the entry is compared. Then one with smaller cost is recorded and also incoming port is updated. Both of these messages are also compared, and only one with smaller cost is forwarded to the next nodes. Thus, all messages arriving each node should be compared, and then only one will be sent to the next nodes. In this context, the next nodes are determined as unreceived ports of same signaling messages. This mechanism conducts the signaling message to destination node along the shortest route. Also all messages are not forwarded toward source node, and then message loop is avoided.

C. Message waiting time on Every Node

Since all arrived messages should to be compared on every node, the messages have to wait on the node. This waiting time is given as a constant value and is the most important parameter in this proposal architecture.

When the message is waiting on the node, and another message of same signaling is arrived, both route costs are compared. Only a message, which has smaller cost is efficient. This message updates the forwarding table and is held on the node. After the waiting time elapsed, this message is duplicated and forwarded from designated output ports. The output ports are the all ports of this node excluding the same signaling message received. After this moment, any received same signaling message with smaller cost is forwarded immediately.

Waiting time should be set very carefully. With smaller waiting time, effectual shortest route is not found. Otherwise, larger waiting time increases the connection setup time.

D. Resolution of Collision with Another Signaling

Previously the mechanism of the enhanced signaling for one signaling is described. At the end of this section, additional mechanism for resolving any collision of two or more different signaling is introduced here.

If two or more connection setups are required in same time and begin all together, they may disturb each other. This architecture allows all signaling messages to meet another message of same signaling on every node to derive cost-shortest route. Therefore, the delay of message processing time influences their route critically. When the message is delayed on a node, this message may miss to meet another message on the next node. As a result, a wrong cost-shortest route may be derived.

To avoid this case, a connection setup should be done network-dominantly. The proposed architecture allows each signaling to occupy the all arrival nodes for constant time, called occupation time. Introduction of parameter increases connection setup time occasionally. However, this mechanism also degreases blocking probability effectively. Since every signaling message is flooded in proposal architecture, all nodes can detect another current signaling and avoid the collision.

IV. PERFORMANCE EVALUATION

Up to previous section, the detail of proposal architecture is described. This architecture allows every node to execute only simple process of enhanced signaling. This process achieves a connection establishment on the cost-shortest route without routing protocol and path computation engine.

In this section, the performance of this architecture is evaluated with computer simulation.

A. Simulation Environment

The simulator is implemented with C++ as event driven program. It operates the routing protocol, path computation engine and normal signaling protocol for conventional source-routing architecture, and enhanced signaling protocol for proposed architecture. A topology including 100 nodes and

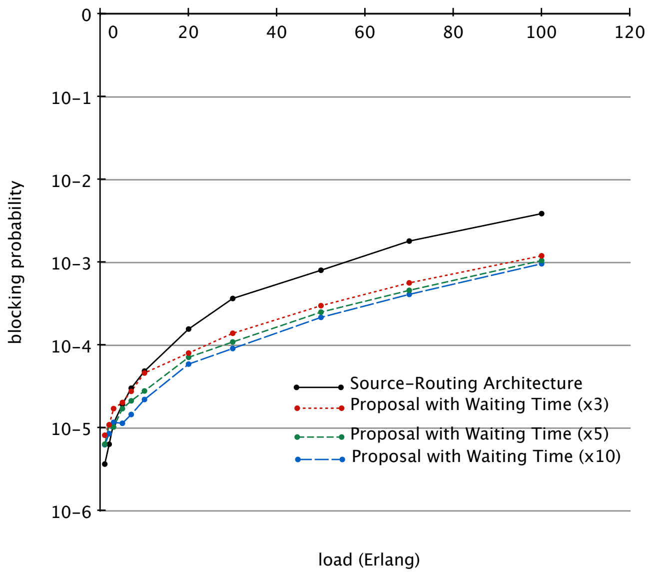

Fig. 2. Blocking Probability.

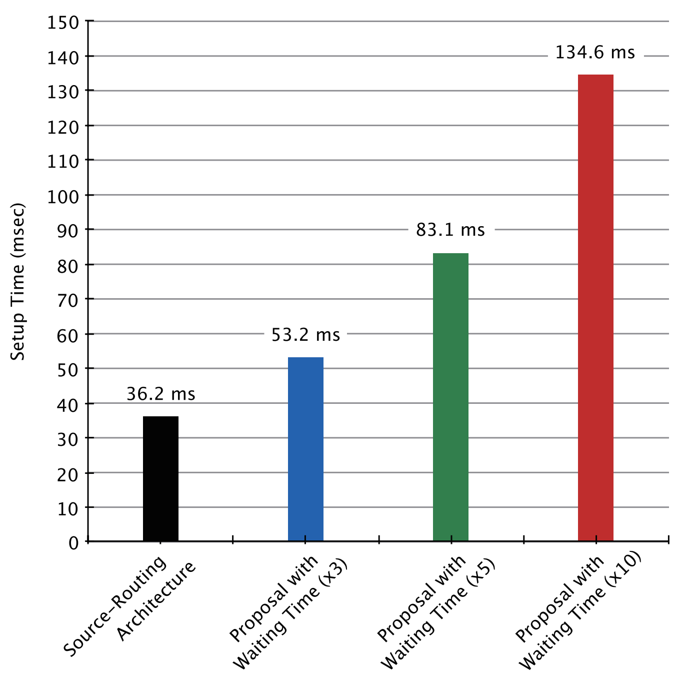

Fig. 3. Setup Time.

200 links is generated with Waxman topology generator. In this topology, each link cost is given from its distance. Every link has 100Mbps bandwidth capacity, and every connection requires 10Mbps on average. When connection is requested, a signaling message is generated and forwarded to destination node. Every received message is processed in three milli-seconds, or buffered when another message is being processed. To simplify the simulation, retrial of signaling is not implemented. When the connection is established, this connection is held for three hours.

With this simulation, Setup time and signaling blocking probability is evaluated. For proposal architecture, the parameters, waiting time is set to 2, 5, 8 times the amount of message processing time, and the occupation time is set to 10 times the amount of message processing time.

B. Simulation Result

The simulation results are shown in Figure 2 and 3. In this simulation, both architectures are evaluated under infrequent signaling environment. Since connection-holding time is large, there are little number of signaling processed at a time when blocking occurs. In proposal architecture, each source node can detects another signaling is in progress. Then a new signaling is held on the source node for constant time to avoid a collision. This mechanism reduces the blocking probability as shown in Fig. 2. On the other hands, it increases connection setup time as shown Fig. 3. However this is not so large amount. It may be permissible for every connection request.

V. Conclusion

In this paper, a new architecture for Connection Oriented Ethernet is proposed. This architecture enables connection establishment for guaranteed communication with only signaling process, without the routing process and path computation process. It is shown that the blocking probability on this architecture is sufficiently low under infrequent signaling environment.

References

[1] Fouli K. Maier M. “The road to carrier-grade ethernet,” IEEE Communication Magazine, vol.47, no.3, pp.S30-S38, March 2009.

[2] E. Mannie. (Editor), “Generalized Multi-Protocol Label Switching (GMPLS) Architecture,” IETF RFC 3945, Oct. 2004.

[3] D. Papadimitriou (Editor), “Ethernet Traffic Parameters,” draft-ietf-ccamp-ethernet-traffic-parameters-10.txt, Internet Draft, work in progress, January 2010.

[4] IEEE Std 802.1Qay, “IEEE Standard for Local and metropolitan area networks-Virtual Bridged Local Area Net- works Amendment 10: Provider Backbone Bridge Traffic Engineering,” Aug. 2009.

[5] Don Fedyk (Editor), “Generalized Multiprotocol Label Switching (GMPLS) control of Ethernet PBB-TE,” draft-ietf-ccamp-gmpls-ethernet-pbb-te-05.txt, Internet Draft, work in progress, August 2010.