DSPF: Datalink Shortest Path First

Motoyuki OHMORI
Chikushi Jogakuen University
ohmori@chikushi-u.ac.jp

Takayoshi NOBUOKA
Trans New Technology, Inc.
taka@trans-nt.com

Hiroki NAKANO
Trans New Technology, Inc.
cas@trans-nt.com

Hironori IKURA
Trans New Technology, Inc.
hikura@trans-nt.com

ABSTRACT

In this paper, we will propose Datalink Shortest Path First (DSPF), which is a link state routing protocol for packet switching at data link layer, especially for Ethernet. We have designed DSPF based upon Open Shortest Path First (OSPF) and Intermediate System to Intermediate System (IS-IS) that are well-known as major link state routing protocols today. Based upon our experiences on protocol developments at data link or network layer, we have then refactored or simplified some functions of these protocols in order to enable to easily and/or incrementally deploy DSPF in existing networks. We have then implemented DSPF not only on PC based systems but also on Tilera embedded system.

1. INTRODUCTION

Ethernet has been widely deployed not only within home networks but also in larger networks such as campus, office, data center networks and even backbone networks of Internet Service Providers (ISPs) today. It has been then attracting network operators and researchers to effectively utilize link bandwidths of Ethernet based networks and improve their availability. However, traditional protocols of Ethernet such as Spanning Tree Protocol (STP)[5] are well-known to have flaws. For example, STP cannot maximize bandwidth utilizations between two nodes where these nodes have more than one multiple links between them. In this environment, all packets go through only one link between them while other links are *blocked*, i.e., no traffic goes through on those links. That is, STP does not support Equal Cost Multi Path (ECMP), and this is obviously inefficient. STP then may not be able to take the shortest path from one node to another when forwarding packets. This results from that STP builds a single spanning tree from one root node to other nodes, and shares the single spanning tree in order to forward all packets. Hence, STP may cause that packets detour, and concentrate traffic on a single link. These are obviously inefficient and causes a problem of a single point of failure. In addition, STP needs much longer time to converge on link failures, which may detract network availability.

In this paper, we will present Datalink Shortest Path First (DSPF), which is a link state routing protocol for packet switchings in Ethernet based networks. DSPF enables faster convergence on link failures, more efficient routes with shortest path tree and ECMP at a data link layer. We design DSPF based upon Intermediate System to Intermediate System (IS-IS)[8] and Open Shortest Path First (OSPF) [11], and then adopt their natures with simplified protocols.

The rest of this paper is organized as follows. In section 2, we present our protocol, DSPF. In section 3, we briefly present our implementation. In section 4, we refer to related works and differences between DSPF and them. In section 5, we refer to future works and conclude this paper.

2. DSPF

In this section, we firstly present an overview of DSPF, and then go into more detail of DSPF in following sections. We here borrow technical terminologies from OSPF and IS-IS, therefore, refer to their specification[8, 11] for more information about terminologies. Note that we do not consider the case where there are traditional nodes (i.e., DSPF non-capable nodes) between DSPF capable nodes. That is, we assume that there are two nodes on a link at maximum and a link is considered to be so called *point-to-point* link. The case where there are traditional nodes between DSPF is out of scope of this paper and our future work.

2.1 Overview

In DSPF, we call a network equipment a *node*. Each node is identified by a unique identifier, *node ID*, which is statically assigned and configured to each node by network operators or administrators. Each node then periodically exchanges *Hello* messages in order to detect appearances and disappearances of *neighbors*, other neighboring nodes. After detecting an appearance of a new neighbor by receiving a Hello message, a node makes sure that the neighbor also receives Hello message from the node in order to ensure *two-way* communications, i.e., packets can reach from the node to the neighbor and vice versa. The node then tries to form an *adjacency* with the neighbor, which ensures that both of the node and the neighbor hold same *Link State Advertisements (LSAs)*. An LSA describes links of each node, each LSA is identified by advertising node ID, and all nodes hold all LSAs in their own *Link State Database (LSDB)*. After exchanging LSAs, the node and the neighbor originate their own LSA. They then *flood* it to other neighbors in order to inform that their new link is available for packet forwarding. All nodes then execute SPF computation with new LSAs in order to forward packets toward new links if it is the shortest path.

2.2 DSPF control messages

DSPF employs fewer control messages, *Hello*, *Link State (LS) update* and *LS acknowledgment (Ack)*, than IS-IS or OSPF. We here briefly present these DSPF control messages.

2.2.1 Hello message

Hello message enables a node to detect neighbors. To this end, Hello messages include the node ID sending those messages. In order to decrease a delay to detect link failures, their sending intervals are in millisecond while one of general OSPF is in second. Hello messages also include the configured values such as a link metric, MTU and Hello sending interval in order to eliminate misconfiguration *in the wild*. This is similar not to IS-IS but to OSPF.

2.2.2 LS update message

LS update messages are exchanged between neighbors in order to exchange LSAs. Each LSA includes a sequence number, an age and link descriptions of advertising node, which is similar to router LSA in OSPF. Each link description in an LSA describes a metric and an interface of its link, which is used during SPF computations. Each LSA in DSPF does not include check summary anymore because of recent reliable data link layer while OSPF and IS-IS include. LS update messages are then exchanged when new neighbors appears or when newer LSA is originated. Once all LSAs are exchanged between nodes, they are not duplicatedly exchanged between same nodes until newer LSA is originated. DSPF does not re-confirm LSA existences with neighbors in order to reduce the number of exchanged control messages while IS-IS does.

2.2.3 LS Ack message

LS Ack messages are exchanged between neighbors in order to make sure if the neighbor properly receives LS update messages without losses. This is similar to OSPF.

3. IMPLEMENTATION

In this section, we briefly present our prototype implementations of DSPF. We have implemented DSPF not only on PC-based systems that are running with NetBSD and Linux but also on the Tilera card, TILEncore-Gx36^{TM}PCIe card[1], which equips with two 1 Gigabit Ethernet (GbE) and two 10 GbE interface, and multiple processor cores. Among these platforms, most of implementations are shared except for some points, especially in packet sending and receiving functions. We have employed Berkeley Packet Filter (BPF)[9] in order to send and receive Ethernet Frames on NetBSD while we have employed packet interface[3], so called *PF_PACKET*, on Linux. On the Tilera card, we have employed their own APIs. We have then confirmed that our implementations have successfully compute shortest path trees. Our implementations have not yet support ECMP but we will implement it in near future.

4. RELATED WORKS

J.M. McQuillan et al. had firstly invented a link state algorithm for ARPANET[10]. IS-IS[8] had been then standardized as a link state routing protocol for Connectionless Network Service (CLNS) of Open Systems Interconnection (OSI). IS-IS had been originally designed only for CLNS but well-designed to support extensions. After IS-IS, OSPF[11] has been proposed for TCP/IP, whose concept is similar to IS-IS and improves some limitations of IS-IS such as a maximum metric value and the large number of control message exchanges and so on.

Regarding a data link layer, STP[5], Rapid Spanning Tree Protocol[7] and Multiple Spanning Tree Protocol (MSTP)[6] have been proposed in order to build up spanning tree to forward packets. However, all of STP, RSTP and MSTP still employ a spanning tree. They are, therefore, inefficient as described in section 1.

In order to solve these issues, Transparent Interconnection of Lots of Links (TRILL)[2] or Shortest Path Bridging (SPB)[4] can be applied, which employ IS-IS as a routing protocol. They are very similar protocols and have been originally motivated in order to support network virtualizations among remote data center while DSPF focuses especially on edge nodes. We will compare benefits between DSPF and others in the future.

5. CONCLUSIONS AND FUTURE WORKS

In this paper, we have proposed DSPF, which is a link state routing protocol for packet switching for Ethernet. We have then implemented prototypes not only on PC-based systems but also on Tilera embedded system. We will improve and evaluate DSPF and our implementation in order to confirm DSPF benefits in future.

6. REFERENCES

[1] T. Corporation. Tilencore-gx36 development platform. http://www.tilera.com/sites/default/files/productbriefs/TILEncore-Gx36.pdf.

[2] D. Eastlake, A. Banerjee, D. Dutt, R. Perlman, and A. Ghanwani. Transparent Interconnection of Lots of Links (TRILL) Use of IS-IS. RFC 6326 (Proposed Standard), July 2011.

[3] P. T. Eugster. Kernel korner: Linux socket filter: sniffing bytes over the network. *Linux J.*, 2001(86):8, June 2001.

[4] IEEE Std. 802.1aq-2012. *Local and Metropolitan Area Networks: Virtual Bridged Local Area Networks - Amendment 8: Shortest Path Bridging*. 2012.

[5] IEEE Std. 802.1d-2004. *Local and Metropolitan Area Networks: Media Access Control (MAC) Bridge*. 2004.

[6] IEEE Std. 802.1s-2002. *Local and Metropolitan Area Networks: Multiple Spanning Trees*. 2002.

[7] IEEE Std. 802.1w-2001. *Local and Metropolitan Area Networks. Rapid Reconfiguration of Spanning Tree*. 2002.

[8] ISO/IEC 10589:2002. *Intermediate System to Intermediate System Intra-Domain Routing Information Exchange Protocol for use in Conjunction with the Protocol for Providing the Connectionless-mode Network Service (ISO 8473)*. Second edition, 2002.

[9] S. McCanne and V. Jacobson. The BSD packet filter: a new architecture for user-level packet capture. In *Proceedings of the USENIX Winter 1993 Conference Proceedings on USENIX Winter 1993 Conference Proceedings*, USENIX'93, pages 2–2, Berkeley, CA, USA, 1993. USENIX Association.

[10] J. M. McQuillan, I. Richer, and E. C. Rosen. The new routing algorithm for the ARPANET. *IEEE Trans. Communications*, 28(5):711–719, May 1980.

[11] J. Moy. OSPF Version 2. RFC 2328 (Standard), Apr. 1998. Updated by RFCs 5709, 6549.