AFEC: A Method of Aggregating Forwarding Equivalence Classes Based on Overlapped Paths

Baobao Zhang, Jun Bi, Jianping Wu
Network Research Center, Tsinghua University; Department of Computer Science, Tsinghua University
Tsinghua National Laboratory for Information Science and Technology (TNList)
Email: zbb@netarchlab.tsinghua.edu.cn, junbi@tsinghua.edu.cn, jianping@cernet.edu.cn

***Index Terms*—Aggregation, Forwarding Table, MPLS**

I. Introduction

The traditional IP routing protocols transfer traffic only according to destination, which is not fine-grained enough to satisfy the significant demands of quality-of-service (QoS) and traffic engineering, so fine-grained flow control techniques, such as Multiprotocol Label Switching (MPLS) [1], emerge. MPLS is the most widely used technique to do fine-grained flow control.

MPLS can route traffic according to any filed of the IP packet header, and even incoming directions of packets. There are more than ten fields in the IP header. Thus, the forwarding table is essentially a space with more than ten dimensions, which is a very vast space. To reduce the size of the forwarding table, MPLS conducts a simple aggregation scheme: IP packets that are routed in the same path are aggregated into a forwarding equivalence class (FEC). Packets belonging to the same FEC share one path connection from the source to the destination. For the MPLS switch, the number of entries in the forwarding table is the number of FECs that go through the switch. The number of FECs is still very large, which restricts MPLS being used in a large scale.

In this work, we propose a scheme of aggregating FECs, which is named AFEC. The key insight of AFEC is that: several FECs that have an overlapped path can be aggregated into one on their overlapped path. Thus, the size of the forwarding table of MPLS switches that are located on the overlapped path can be reduced. In this work, we describe AFEC in the MPLS environment, but AFEC can also be used in other environments, such as OpenFlow [2], AFEC can also be used to reduce the flow table of OpenFlow. We will explore the detailed mechanism of applying AFEC to other environments in the future.

II. Approach

In this work, we propose a method of aggregating forwarding equivalence classes, which is named AFEC. The key insight of AFEC is that: several FECs that have an overlapped path can be aggregated into one on their overlapped path. Thus, the size of the forwarding table of MPLS switches that are located on the overlapped path can be reduced. The aggregation is done by establishing a tunnel on an overlapped path. FECs that go through the overlapped path can share the tunnel, which is equivalent to that these FECs are aggregated into one when going through the overlapped path. The tunnel is very similar with the virtual path in Asynchronous Transfer Mode (ATM). The switch at the beginning of the tunnel is called tunnel entrance switch, and the switch at the end of the tunnel is called tunnel exit switch. The middle switches of the tunnel are called tunnel inner switches. AFEC works as follows.

a) Tunnel entrance switch maintains an encapsulation table. The data structure of encapsulation table is as shown in Table 1. The two fields of incoming label and incoming port are used to match packets. The matched packets are encapsulated with the corresponding encapsulation label, and then forwarded to the corresponding outgoing port.

b) Tunnel inner switches transfer packets normally. Note: an encapsulated packet is forwarded through matching its encapsulation label instead of its original label.

c) Tunnel exit switch de-encapsulates packets. Encapsulated MPLS headers are striped here. And then packets are forwarded through using their original labels.

The structure of the forwarding table, which is as shown in Table 2, is similar with the one of the encapsulation table. The difference is that: the encapsulation label in the encapsulation table is enclosed in the MPLS header while the outgoing label in the forwarding table covers the original label. The encapsulation table and the forwarding table need the same storage space for one entry.

For the tunnel entrance switch, forwarding entries will be converted by AFEC to the same number of encapsulation entries. Thus, the storage space for the tunnel entrance switch will not be reduced. For the tunnel exit switch, packets are forwarded in a normal way, so the storage space for the tunnel exit switch will not be reduced by AFEC, either. For the tunnel inner switches, only one forwarding entry of establishing the tunnel is needed for FECs aggregated. Thus, the storage space for the tunnel inner switches can be reduced by AFEC aggregation.

Table 1: The data structure of the encapsulation table

Incoming Label	Incoming port	Encapsulation Label	Outgoing Port

Table 2: The data structure of the forwarding table

Incoming Label	Incoming port	Outgoing Label	Outgoing Port

978-1-4673-2447-2/12/$31.00 ©2012 IEEE

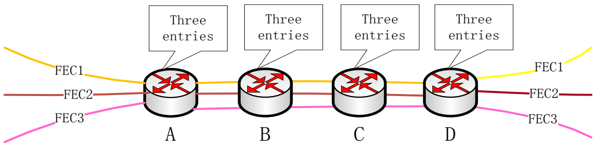

Fig.1. The situation without using AFEC

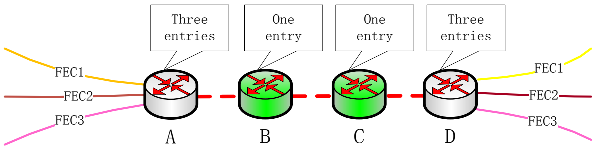

Fig.2. The situation of using AFEC

The work way of AFEC is as follows. The first step is to choose tunnel inner switches. If the size of the forwarding table of a switch is unacceptable, the switch is chose as a tunnel inner switch. Naturally, the switches that are neighbored with tunnel inner switches are tunnel entrance switches and tunnel exit switches, which are used to aggregate and decompose FECs. The second step is to establish a tunnel from the tunnel entrance switch to the tunnel exit switch. As described above, the forwarding table of the tunnel inner switch can be reduced.

Now we give an example. As shown in Fig. 1, there are three FECs: FEC1, FEC2 and FEC3. The three FECs have an overlapped path: A-B-C-D. We choose Switch B and C as tunnel inner switches. Naturally, Switch A is the tunnel entrance switch, and Switch D is the tunnel exit switch. If not using AFEC, each of the four switches needs three forwarding entries. If AFEC is used, as shown in Fig. 2, for inner switch B and C, each of them only needs one forwarding entry. Entrance switch A needs three encapsulation entries, and exit switch D still needs three forwarding entries.

III. Evaluation

From rocketfuel project [3], we connected nine router-level topologies, which are located in different autonomous systems (AS). For simplicity, we make some assumptions as follows: a) each router has one local prefix; b) we only use two fields (source address and destination address) to construct FECs; c) the routing algorithm is the shortest path algorithm; d) we consider all the routers as MPLS switches; e) we consider both encapsulation table and forwarding table as forwarding table uniformly due to the same storage space for one entry.

We observe the number of entries in the forwarding table for each switch before and after using AFEC aggregation. We take the topology of AS1221 for example. There are 2515 switches and 6078 links in the topology. If the AFEC aggregation is not used, the number of forwarding entries for each switch is actually the number of FECs that go through the switch, as shown in Fig. 3. Each point denotes one switch. The horizontal coordinate denotes the ID of the switch. The vertical coordinate denotes the number of entries in the forwarding table. We number switches with decreasing order according to the number of entries in the forwarding table without using AFEC. We find that if the AFEC aggregation is not used, the maximum number of forwarding entries is high to 2,832,523.

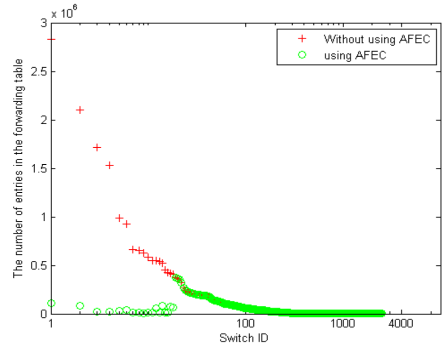

Fig.3. The number of entries in the forwarding table

The BGP routing table has 350K routing entries now. If the maximum number of forwarding entries for switches is limited to be less than 400K, it may be reasonable. Thus, here we assume that the maximally acceptable number of entries is 400K. If we want to limit the maximum number of entries in the forwarding table to be below 400K, it can be done as follows. There are 18 switches whose entries in the forwarding table are more than 400K. We can consider the 18 switches as tunnel inner switches. And switches that are neighbored with the 18 tunnel inner switches are considered as tunnel entrance or exit switches, there are 478 such switches. For FECs that go through the tunnel inner switches, we can establish tunnels for them using AFEC aggregation. Then the number of forwarding entries for the 18 tunnel inner switches can be decreased to 110K. As shown in Fig.3. Thus, after the AFEC aggregation is used, the maximum number of forwarding entries is less than 400K, achieving our goal of limiting the maximum forwarding entries. Of course, we can limit the maximum number of forwarding entries to be smaller. We will study how to achieve the smallest value of the maximum number of forwarding entries based on overlapped path tunnel.

IV. Conclusion and Future Work

In this work, we devise an aggregation method based on overlapped path tunnel. This aggregation method can be used to adjust the maximum number of forwarding entries to an acceptable value. In this work, we show the effectiveness of the aggregation method using a manual way. In the future, we will study the automatic way of establishing tunnels to limit the maximum number of entries to an acceptable value.

Acknowledgement

This research is supported by National Science Foundation of China under Grant 61073172, Program for New Century Excellent Talents in University, National Basic Research Program ("973" Program) of China under Grant 2009CB320501, and China Next Generation Internet.

References

[1] Eric C. Rosen, Arun Viswanathan, Ross Callon, "Multiprotocol Label Switching Architecture", RFC 3031, January 2001

[2] Nick McKeown, Tom Anderson, Hari Balakrishnan, Guru Parulkar, Larry Peterson, Jennifer Rexford,Scott Shenker, Jonathan Turner, "OpenFlow: Enabling Innovation in Campus Networks", White Paper, March 14, 2008

[3] rocketfuel: http://www.cs.washington.edu/research/networking/ rocketfuel