A Novel Approach to Improve the Performance of MPLS-based VPN*

Geng Yanhui Sun Qiong Chen Yuzhong Yu Nenghai
Information Processing Center
University of Science and Technology of China
Hefei, Anhui, 230027
P.R. China
{yhgeng, joansun, yzchen}@mail.ustc.edu.cn; ynh@ustc.edu.cn

Abstract –**Internet has experienced an exponential growth in recent years, various kinds of application is raising a critical challenge to current IP network. The IP-based virtual private network (VPN) technology is now becoming a good solution for the delivery of future Internet services, and Multiprotocol Label Switching (MPLS) technology is being adopted by some largest Internet service providers to offer VPNs and some value-added applications on top of their VPN transport networks. In this paper, we proposed a novel approach based on using label stack properly to reduce the label size efficiently. In this way, the performance of MPLS-based VPN can be improved. And our simulation results validated our approach, demonstrating that MPLS-based VPN established by our approach has a better performance.**

I. INTRODUCTION

A number of companies have geographically dispersed operations with local networks, supporting the information processing requests at every endpoint. Typically, these sites are connected with point-to-point dedicated communication lines provided by the service provider reliably and in security. While the cost for it is fairly expansive. Alternatively, virtual private network (VPN) [7], [11] technology is a good solution; it can provide virtual dedicated lines over Internet. This solution brings us substantial cost saving since it obviates the use of dedicated lines by using resources from the public infrastructure [5], [8]. VPNs provide enterprise-scale connectivity between sites across a shared infrastructure in a secure manner with the same policies as a private network.

MPLS [1] is a technology proposed by the IETF [4]. It is developed for the purpose of improving forwarding performance. The basic idea of MPLS is to forward the incoming packets based on a short, fixed-format label. When a packet enters the ingress node of a MPLS domain, a label is inserted into the packet header. Then the packet is forwarded along a connection-oriented label switch path (LSP) by performing label swapping, instead of looking for the longest address match at each hop. The obvious advantage of MPLS technology is that the packet header analysis process needn't be done at every hop. We just analyze the header and assign the labels to the packet when it has just entered the network. The ingress router may use some additional information about the packet such as its source and its data type to assign the packets different route to meet different QoS requirements. For example, multimedia application data may request low delay and low delay jitter, then we can encode this special information in the packets' header analyzed by the ingress router to satisfy the delivery. And by using the explicit route in MPLS network, traffic engineering is easier to be realized in contrast to the conventional IP network. All these have made MPLS an advanced and attractive technology applied in backbones. The high performance MPLS backbone makes MPLS-based VPN powerful and efficient.

But to achieve its optimal performance; in this paper we presented a novel approach based on using label stack properly to reduce the label size efficiently. In this way, the performance of MPLS-based VPN can be improved. The rest of the paper is organized as follows. In section II, we bring forward the problem existing in conventional MPLS-based VPN and have an analysis of it. In section III, we present our novel approach and scheme. Section IV reports simulation results and performance analyses of the proposed approach. Conclusion is drawn in Section V.

II. PROBLEM ANALYSIS

Fig. 1 shows the architecture of MPLS-based VPN. MPLS VPN is composed of Customer Edge router (CE), Provider Edge router (PE), Provider backbone router (LSR). The CE can use IP protocol instead of MPLS to transmit packet. Services are implemented at the edge of the network. In fact, the VPNs only exist at the edge of the service provider's network. The core routers do not participate in the actual VPNs, they just continue to forward packets over various LSPs. The customer's routers also do not participate in the VPNs, since they simply continue to route IP packets in according to the customer's established addressing and routing [3] schemes.

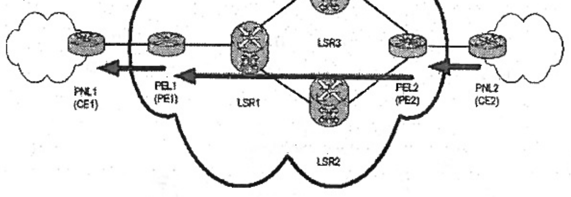

Fig. 1. Forwarding VPN information using MPLS

Service providers are anxious to offer VPN services to their corporate customers. Here we don't focus on the advantages of MPLS-based VPN. The emphasis is put on the problems exist in the conventional MPLS-based VPN. VPNs create a high degree of complexity that offsets some of the simplicity of a raw MPLS network. The uppermost problem is that with some large service providers providing VPN services in MPLS network, it's more and more difficult and complex to providing these MPLS-based VPN

*Supported by the *Chinese 863 National High Technology Program*, under grant No. 2002AA783043

services to thousands of customers since that will require the service provider to set up and manage thousands of MPLS LSPs connecting the VPN endpoints. This is the method adopted by Layer 2 MPLS VPNs and VPN services based on the overlay model which are widely used today [2]. It turns to be a puzzle business since only 20 bits of each 32 bits label stack entry are allowed to encode the label and then the number of label is limited within 220. So we must use the label frugally consider the extendibility possible. And with the number of labels used decreasing we can make the size of the forwarding table smaller which is used by each label switching router (LSR) to make swap or forwarding decisions for label of every incoming packet. The routers can then switch packets faster with the smaller forwarding table.

In the following section we proposed our novel approach based on using label stack to reduce the number of label needed and achieve the goal of improved performance of MPLS-based VPN.

III. PROPOSED APPROACH BASED ON USING LABEL STACK

From the analysis in section 2, we know that it's important to have a smaller label size for achieving better performance of MPLS-based VPN. However, as we will show in the following by an example, we can't have a smaller label size and a lower label stack depth at the same time. Smaller label size is obtained with the cost of deeper stack. While deep stack is not desirable since longer stacks will employ more space in IP headers of packet. So we must have a balance or trade-off [6] between smaller label size and lower label stack depth to achieve the goal of improved performance of MPLS-based VPN.

Now let's illustrate the trade-off between label size and label stack depth briefly by a concrete example. Consider the network shown in Fig. 2. A, B, C and D are four VPN endpoints, and each has equal input bandwidth and output bandwidth. We adopted the VPN model provided in the Hose Model (point to multi-point model) [9], [12]. In this model input bandwidth and output bandwidth must be specified by every VPN endpoint specifies. They are the maximum amount of traffic allowed to be inputted to the VPN endpoint and the maximum amount of traffic allowed to be outputted to the VPN endpoint. We can use a tree structure [10] as shown below in Fig. 3 to connect VPN endpoints to utilize the network bandwidth efficiently.

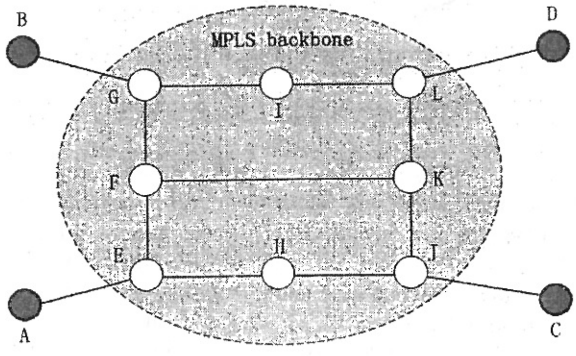

Fig. 2. Network topology

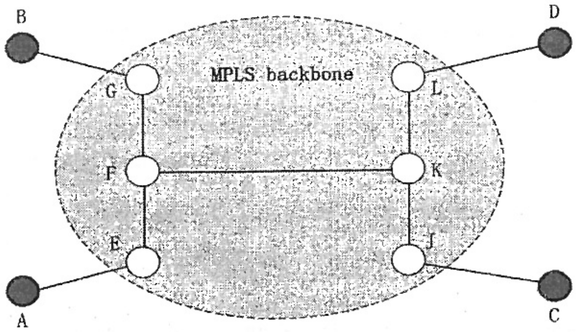

Fig. 3. Optimized network topology

First we adopted the scenario in which the label stack depth is 1. Then 4 labels are needed since we must distinguish the 4 VPN endpoints by assigning each of them a single label. Every node in network just forwards every incoming packet with the label assigned to the special VPN endpoint along the outgoing interface in the direction of the endpoint and needn't pop the label. The TABLE I below gives out the outgoing interface and action for every incoming packet of node F and node K.

TABLE I

FORWARDING TABLE WITH LABEL STACK DEPTH 1

Node	Incoming interface	Label of packet	Outgoing interface	Node action
F	K-F, G-F	1	F-E	None
	E-F, K-F	2	F-G	None
	E-F, G-F	3, 4	F-K	None
K	F-K, L-K	3	K-J	None
	F-K, J-K	4	K-L	None
	J-K, L-K	1, 2	K-F	None

From the table above we can see clearly that one endpoint can send packets to any of the others by pushing a special label bound with the object endpoint onto the label stack. For example, when node F encounters a packet from interface E-F or G-F with a label 3, it will forward the packet to interface F-K without pushing or popping the label, and then when node K encounters the packet from node F with label 3, it will forward the packet to interface K-J, and finally node J will forward the packet to endpoint C.

It means that 4 labels are needed in this scenario (one label for one endpoint) with a label stack depth 1.

Now we will turn to the scenario in which the label stack depth is 2. In this case we need only 2 labels (1 and 2) to make it possible for every endpoint to communicate with any of the others. The forwarding table of node E, F, G, J, K and L is shown in TABLE II (* denote 1 or 2). For instance, if endpoint A want to communicate with endpoint B, it just push label 1 onto the label stack to realize the communication objective. More complex, to send some packets to endpoint C, endpoint A simply need to push label 1 and label 2 onto the label stack orderly. When node E encounters the packets from interface A-E with label (1, 2), it will forward the packets to interface E-F without any action according to the forwarding table. And then node F will forward the packets to interface F-K and pop the label 2 on the top of the label stack. By this time there is only 1 label (1) in the label stack. Then node K will forward the packets from node F to interface K-J since the top of the label stack is label 1. And finally node J will forward the packets to endpoint C.

In this concrete example, the number of label needed decrease from 4 to 2 with the label stack depth increases from 1 to 2.

TABLE II

FORWARDING TABLE WITH LABEL STACK DEPTH 2

Node	Incoming interface	Label of packet	Outgoing interface	Node action
E	A-E	*	E-F	None
	F-E	*	E-A	None
F	E-F	1	F-G	None
	G-F	1	F-E	None
	E-F, G-F	2	F-K	Pop
	K-F	1	F-E	None
	K-F	2	F-G	None
G	F-G	*	G-B	None
	B-G	*	G-F	None
J	J-K	*	C-J	None
	C-J	*	J-K	None
K	J-K	1	K-L	None
	L-K	1	K-J	None
	J-K, L-K	2	K-F	Pop
	F-K	1	K-J	None
	F-K	2	K-L	None
L	K-L	*	L-D	None
	D-L	*	L-K	None

As we can see from the discussion above, the approach based on using label stack can reduce the number of label needed efficiently. Then the space to encode label become smaller correspondingly. While what's the measurable relation between the label size and the label stack? Here we list some conclusions with the help of information theory.

1. We need at least n labels in a network with n nodes if the label stack depth is 1, or else we won't even distinguish 'between the objective nodes in the network.
2. If the label stack depth is s, at least $L \geq n^{1/s}$ labels are needed where n is the number of distinct destination in the network.
3. Concluded from the entry above, we must have the label stack depth $s \geq \log n / \log L$ if the number of label L is fixed.

These are some bound of label size and label stack, but when there are several pairs of label size and label stack which meet the bound above, which pair is the optimal? We can employ network simulation to evaluate the performance of MPLS-based VPN under different pairs of label size and label stack depth.

IV. SIMULATIONS

We refer to network simulation with the help of simulation platform NS2 (Network Simulator 2) to evaluate the optimal pair of label size and label stack depth in MPLS-based VPN network on various scale.

A. Simulation on a Small Scale

A network model shown in Fig. 4 below is the representation of MPLS-based VPN network on a small scale.

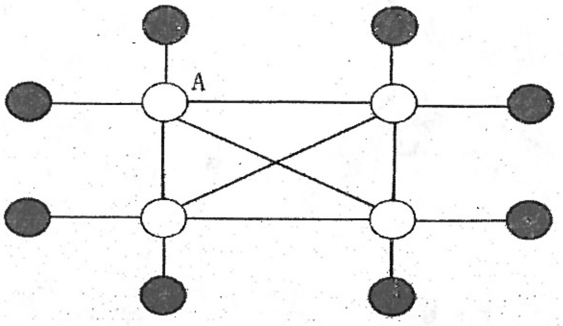

Fig. 4. Topology of network on a small scale

In the figure above, 8 black nodes are VPN endpoints, and the other 4 white nodes constitute the MPLS core network. If we adopt stack depth 1, the number of labels needed is 8 (3 bits space to encode 8 labels). When we increase the label stack depth to 2, 3 labels (2 bits space to encode 3 labels) are needed. But when we monitor the throughput of node A, we noted that there isn't nearly any difference with the increase of the label stack depth since the saving of 1 bit is too insignificant.

B. Simulation on a Middle Scale

In this scenario we build a MPLS-based VPN network with 1000 endpoints. We find that when increasing the label stack depth from 1 to 3, the number of labels needed decreases from 1000 (10 bits space to encode 1000 labels) to 70 (7 bits space to encode 70 labels). It means that we have saved more than 90% of the labels in contrast to the case in which the label stack depth is 1. The saving of labels is prominent while the performance of the routers hasn't been improved significantly. Anyway, a router can easily maintain 1000 labels and the space saving of 3 bits is not distinct anyway, while considers the scenario in which several hundred VPN clusters are provided over the same network, the savings will accumulate to significant savings.

C. Simulation on a Large Scale

In this case we create 100 VPN clusters (each of them has 1000 endpoints) and make them connected in one backbone. In such a large network with 100000 endpoints, the advantage and validity of our approach is shown clearly. We need 100000 labels when the label stack depth is 1. While only 7000 labels is needed when the label stack depth is increased to 3. The saving of labels is significant and this lightens the burden of the routers greatly. So the routers will forward packets faster and more efficiently. The throughput and delay of one router monitored by us is as in the figures below.

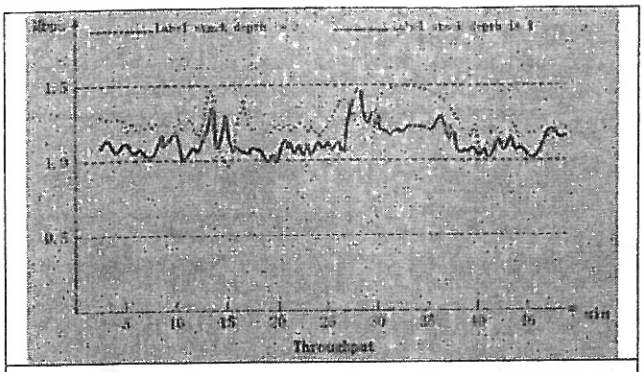

Fig. 5. Throughput of one router

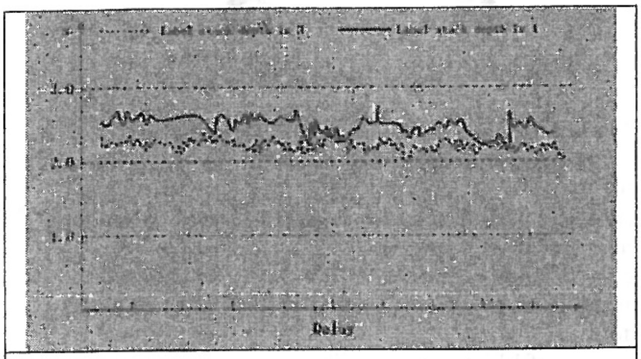

Fig. 6. Delay of one router

In the figures above, we note that the throughput of the router which is in the network with the label stack depth 3 is higher that with the label stack 1. While the delay of the router is lower correspondingly. It means that the routers adopting our approach can forward packets faster and then enable the packets' lower delay. Therefore the holistic performance of MPLS-based VPN is improved.

V. CONCLUSIONS

To achieve better performance of MPLS-based VPN, this paper develops a novel approach based on using label stack in the proper scenario. In this way, the number of labels needed in the network can be reduced efficiently, and then help reduce the size of the forwarding table hold by every router. A smaller forwarding table will lighten the burden of the routers and make them forward packets faster and more efficient. The holistic performance of MPLS-based VPN can be improved in this way. Besides the analysis and research in theory, we had done some simulation experiments, and the results also validated the advantage and validity of our approach.

VI. ACKNOWLEDGMENT

The authors gratefully acknowledge the contributions of Chen Zheng, Ren Rong, Wang Hui, and Shan Jianfeng.

VII. REFERENCES

[1] Eric C, Rosen, Arun Viswanathan, and Ross Callon, "MultiProtocol Label Switching architecture," (RFC 3031) http://www.ietf.org/rfc/rfc3031.txt, January. 2001.

[2] Bruce Davie and Yakov Rekhter, *MPLS: Technology and Applications*, Morgan Kaufmann Publishers, 2000.

[3] Greg N, Frederickson and Ravi Janardan, "Designing networks with compact routing tables," *Algorithmica*, vol. 3, 1988, pp. 171-190.

[4] Eric C, Rosen, Dan Tappan, Yakov Rekhter, Guy Federkow, Dino Farinacci, Tony Li, and Alex Conta, "MPLS label stack encoding," (RFC 3032). http://www.ietf.org/rfc/rfc3032.txt, January. 2001.

[5] N. G. Duffield, P. Goyal, A. Greenberg, P. Mishra, K. K. Ramakrishnan, and J. E. van der Merwe, "A Flexible Model for Resource Management in Virtual Private Networks," In *Proceedings ACM SIGCOMM*, 1998.

[6] David Peleg and Eli Upfal, "A trade-off between space and efficiency for routing tables," *J.Assoc. Comput. Mach*, vol. 36, no. 3, 1989, pp. 510-530.

[7] P. Ferguson and G. Huston, "What is VPN," *The Internet Protocol Journal*, 1999.

[8] Palmieri F, "VPN scalability over high performance backbones," In *Proceedings ISCC 2003*, 2003, pp. 975-981.

[9] A. Kumar, R. Rastogi, A. Silberschatz and B. Yener, "Algorithms for provisioning virtual private networks in the hose model," In *Proceedings ACM SIGCOMM*, 2001.

[10] Anupam Gupta, Amit Kumar, and Rajeev Rastogi, "Routing issues in MPLS," In *Proceedings of the 42nd Annual IEEE Symposium on Foundations of Computer Science*, 2001, pp. 148-157.

[11] Venkateswaran R, "Virtual private networks," *Potentials, IEEE*, vol. 20, no. 1, Feb-March. 2001, pp. 11-15.

[12] A. Gupta, J. Kleinberg, A. Kumar, R. Rastogi and B. Yener, "Provisioning a virtual private network: A network design problem for multicommodity flow," In *Proceedings of the 33rd ACM Symposium on Theory of Computing*, 2001.