A Multipath Mechanism Applied to Traffic Engineering In MPLS Network*

Sun Qiong Geng Yanhui Wang Hui Yu Nenghai
Information Processing Centre
University of Science and Technology of China
Hefei, Anhui, 230027
P.R. China
{joansun, yhgeng, hwang33}@mail.ustc.edu.cn; ynh@ustc.edu.cn

Abstract **– Recent work on differentiated services in the Internet has introduced new technique on aggregated traffic. But transmitting packets on the shortest path has not been able to take full use of source of networks, which wastes multiple paths that frequently exist in Internet service provider networks. As a result, the network may not operate efficiently, especially when the traffic patterns are dynamic. In this paper, we describe a novel multipath scheme applied to traffic engineering. The main goal of the scheme is to avoid network congestion by adaptively balancing the load among multiple paths based on the feedback, which is the measurement and analysis of path congestion. This paper presents the novel mechanism, and simulation results are provided to illustrate the efficacy of our scheme.**

I. INTRODUCTION

Today, with the increase of network request, Internet must provide real time applications, for example, video flow, which needs the guarantees of bandwidth, delay, packet loss and so on. So traffic engineering is used to resolve these problems, which can provide fast, reliable and differentiated services. According to the Internet engineering task force (IETF), Internet traffic engineering is broadly defined as that aspect of network engineering dealing with issue of performance evaluation and performance optimization of operational IP networks [1].It is more effective than QoS routing in the maximizing operational network while meeting certain constraints. It is because of QoS focusing on supporting varying service qualities for each individual end-to-end traffic flow, which is per-flow model. However, Internet service providers generally have not embraced the per-flow model, mostly due to the need to maintain state information for each flow at each router on its path.

Recently, the emergence of MPLS (multiprotocol label switching) has facilitated traffic engineering [2]. MPLS supports explicit routing, which allows a particular packet stream to follow a pre-determined path rather than a path computed by hop-by-hop destination-based routing such as OSPF. In MPLS a packet is forwarded along the LSP (label switched path) by swapping labels. Thus, support of explicit routing in MPLS does not entail additional packet header overhead, which can provide more effective routing.

In this paper, we propose a novel multipath mechanism, which can be used in MPLS to control congestion and achieve the aim of load balancing more effectively. This mechanism is an integrated state-dependent traffic engineering mechanism, which is different from the past work. Our mechanism assumes that multiple explicit LSPs between an ingress node and an egress node in an MPLS domain have been established. It uses normal routing protocol such as OSPF, or configured manually. Besides these, our mechanism is based on aggregated flow, which can avoid the gap between the growing need for service differentiation and the inability of the per-flow QoS model. The goal of the ingress node is to aggregate traffic which belongs to the same class and to assign traffic across the multiple LSPs. And the egress node reorders and transmits traffic to neighboring networks. Fig. 1 shows an example of a network where there is a pair of ingress node and egress node and three paths between the two nodes. And these paths are disjointed, which means they don not share the same node. Our mechanism runs on these paths between the ingress node and the egress node. The traffic is analyzed and diffused on three paths by the ingress node and then is aggregated at the egress node. There are three steps in our mechanism: (1) probing and finding congestion, (2) selection of new paths, and (3) allocation of traffic along the new paths.

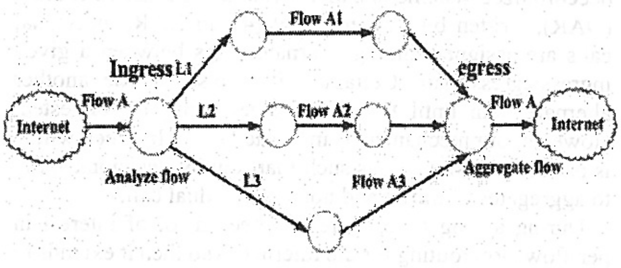

Fig. 1. Network under multipath mechanism

The paper is organized as follows. Section 2 describes related work. In Section 3, we provide the details of our multipath routing scheme in MPLS. Section 4 shows the simulation results that illustrate the ability of our scheme to reroute flows around congestion. In Section 5, we discuss our results and make brief conclusions.

II. RELATED WORK

Current internet transmissions normally rely on the shortest path between the two end points. Although such scheme has been shown to improve the efficiency of the network, it in turn becomes the bottleneck of the system:

- Due to high traffic
- Some technical failure of this path

Keeping in view the demands of the present internet services, the shortest path does not seem to be a viable solution. In addition, traditional shortest path suffer from some others limitations including:

- Can not achieve the aim of optimization, such as the aim of optimizing bandwidth, delay and packet loss.

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

- Traffic of different QoS can not be classified because of following the same path.
- Can not balance load
- The statistic character of traffic should be known in advance.

Applying multipath to traffic engineering in MPLS is the objective of this paper to address these limitations.

However, the state dependent mechanism is adapted to unpredictable traffic variations. So it is more practical at any time. Generally, the state is based on one metric, such as buffer occupancy, packet delay and packet loss. In our paper, the state is an integrated metric including the three metrics.

In a time dependent mechanism, the traffic assignment must be based on the historical information from seasonal variations in traffic. If customers want to use this mechanism, they have to know much about changes of traffic on links during a relatively long time scale. Time-dependent mechanism do not adapt to unpredictable traffic variations or changing network conditions.

When we can not predict the traffic in the changing networks using historical information, the time-dependent may not be able to control congestion. So we use state-dependent mechanism to deal with the traffic engineering. The state is based on buffer occupancy, packet delay and packet loss. In this paper, the state is the integration of the three metrics.

The basic idea in alternate routing roots from the dynamic and alternate routing algorithms developed for circuit switched networks in the 1980s and 1990s [3, 4]. The decentralized scheme is called dynamic alternative routing (DAR), written by Gibbens et al [4]. In DAR, individual calls are assigned among alternate paths between a given ingress/egress pair. Certainly, it is also to use another alternate path until the paths under using is congested. However, our mechanism is interested in the Internet, which is packet switched and is such that routing decisions apply to aggregates of traffic and not to individual call.

During a long time, there has been a lot of interest in per-flow QoS routing for the Internet. And then it extends to OSPF [5, 6, and 7]. In Ref. [8], they propose a 'localized' QoS routing scheme, where ingress nodes use locally available information in selecting paths for individual QoS flows. This scheme is similar to ours in rerouting flows on the basis of locally collected information. Zappala [9], who describes an alternative path routing mechanism similar to ours for multicast traffic, focuses on issues of path computation and installation.

But routing for per-flow QoS will cause high routing overheads, which have to maintain all per-flow state that is based on < source IP address, source port, destination IP address, destination port, IP protocol>. It is not good to improving the efficiency of the network. In studying the literature, we have found Stoica and Zhang's work on LIRA [10] considers economic mechanisms for traffic conditioning and routing without appealing to a per-flow QoS model. LIRA is a relatively complicated scheme for aggregate QoS since source routing is used to assign packets to a given path. In comparison, our mechanism is not so complicated, which does not require any interaction with the underlying routing protocol. Although routing for aggregated flow causes the problem of reordering packets, we can use the approach prompted in [11] to reduce the time of reordering.

III. NOVEL MULTIPATH MECHANISM APPLIED TO TRAFFIC ENGINEERING

The basic idea of our mechanism is not complex, which is accomplished by the ingress/egress nodes pair. In general, our mechanism has three main components: (1) monitoring and finding congestion, (2) selecting candidate alternate paths, and (3) load balancing by assigning traffic to alternate paths. We describe these mechanisms in detail in Subsections A-C, respectively.

A. Monitoring and Finding Congestion

Since the goal of alternate routing is to reroute traffic around congestion, it is essential to have a mechanism in place for discovering congestion at least along direct path. In this paper, we propose an integrated metric congestion discovery method. Specially, congestion discovery is based on probe packet that provides feedback indicating the existence of congestion and condition of assigning traffic along the given paths. Here, congestion is defined in terms of buffer occupancy, delay and loss rate, which is an integrated of the three measurements. See (1), we assume three weights respectively to the three measurements. The weights can help find the congestion more betimes. Its application here is somewhat different in that we are only interested in one measurement.

We now discuss the progress of congestion discovery. The mechanism is executed during a congestion discovery period, which is a binary congestion feedback scheme. During the probe period, the ingress node sends probe packets to the egress node among every LSP. On the way to the egress node, the probe packets record the buffer occupancy, delay and loss rate on every LSP, which will be sent back by egress. And then the ingress use (1) to calculate the integrate measurement and estimate the condition of congestion on the LSPs. B is buffer occupancy, D describes the delay and L is used to show packet loss. C is final congestion measurement. k1, k2, and k3 is weights.

$$C = k1 * B + k2 * D + k3 * L \qquad (1)$$

We will introduce the three measurements as follows.

Buffer occupancy probe packet will compare buffer occupancy of every node on the same LSP and then record the max buffer occupancy, which will be reflected to the ingress by the egress.

Delay packet delay is another metric that can be reliably measured. The probe packet is time-stamped at the ingress with time Ta and recorded at the egress node at time Tb. If the ingress clock is not synchronous with the egress clock, and the ingress is faster than the egress by Td. The total packet delay is Ta-Tb+Td.

Loss rate packet loss is the third metric that can be estimated by probe packets. The ingress encodes a sequence number in the probe packet to notify the egress node how many probe packets have been transmitted by the ingress node, and another field in the probe packet to indicate how many probe packets have been received by the egress node. When the ingress checks these fields of probe packets, it is able to estimate the one-way packet loss.

When we use (1) to calculate congestion measurement, we can specify the weights of the three metrics. For example, if it is thought that the loss is most important metric, the k3 can be assumed more than others. The advantage of this approach is that weights can be assumed based on practical needs.

B. Selecting Candidate Alternate Paths

The goal of probing congestion is to allocate traffic effectively. So except the direct path, we also need more alternate to share additional traffic. Here we describe the algorithm for selecting candidate alternate paths for aggregate flows. Decisions to utilize alternate paths occur at the same time as the congestion discovery process described above. The scheme is very simple: when we find the direct path congested, the scheme attempts to find other alternate paths to share traffic. We firstly use the shortest path, and use the second short path, and then the third. The longest of all of candidate alternate paths finally is to be used. It is a better approach to use all alternate paths following the certain order, which always guarantees that the best path is used at any time firstly. It means that we make every effort to improve efficiency of the network.

The principle of finding alternate paths can be described in principle 1 below.

Principle 1. *Finding alternate paths*

First, every node x in graph G which is abstracted from the topology of network owns an empty path set *Px*, which is used to store the disjoint paths from node x to node z. An empty path set *Cx* of node x stores the alternate paths from node x to node z which may be not disjoint from the paths of *Px*. *Px* and *Cx* have the same structure {metric, path}, in which metric is defined as: metric=m (s, cost, node#) where s is the security modulus of the link, and node# is the number of nodes. Paths in *Px* and *Cx* are stored in the order of size of metric, and the order must be kept whenever a path is added or deleted.

The Principle 1 is realized in two steps

Step 1: Finding the shortest path and the second shortest disjoint paths.

Step 2: Finding more disjoint paths by exchanging messages.

C. Load Balancing by Assigning Traffic to Alternate Paths

Once alternate paths have been defined, the remaining task is to adjust the amounts of alternately routed traffic according to congestion feedback information. The step is not difficult too, but minimizing the need for packet reordering should be paid attention to in the actual implementation.

At the same time of allocating traffic on the second path after the shortest one congested, we keep sending probe packets on the two paths to know the condition of allocating traffic. If the second path is competent for sharing additional traffic the shortest path can not afford, it will not need new paths. Otherwise, the third short path will be added to share traffic. The mechanism will not stop until load on all paths being used is balanced. The mechanism makes full use of resource of network and controls congestion more effectively.

The process of allocating alternate flow is specified in Algorithm 1 below.

Algorithm 1. *Allocating alternate flow*

Step 1: Given the node pair *(A, B)*; we get the path set *Px* with the help of *principle 1*.

Step 2: Paths in set *Px* are sorted by the size of their *cost*. Then we get the set *{p1, p2, p3…}*, in which $cost_{p1} < cost_{p2} < cost_{p3} \ldots$

Step 3: We adjust the amount of flow following an additive increase/additive decrease rule. The change to the fraction of flow is a constant amount *k*. By sending probe packet periodic if *p1* is found congested, *k* of the flow is shifted to path *p2*. If the congestion continues when U_{p2} (bandwidth occupancy ratio) increases to the optimal extent of (U_{min}, U_{max}), *k* of the flow is shifted to path *p3*.

Step 4: Taking *step 3* repeatedly until *p1* is uncongested.

IV. SIMULATIONS

In order to prove the effectiveness of the proposed mechanism, we employ network simulation to compare the performance of our mechanism with that of ordinary MPLS (single path) mechanism.

A network model shown in Fig. 2 below is the representation of MPLS network

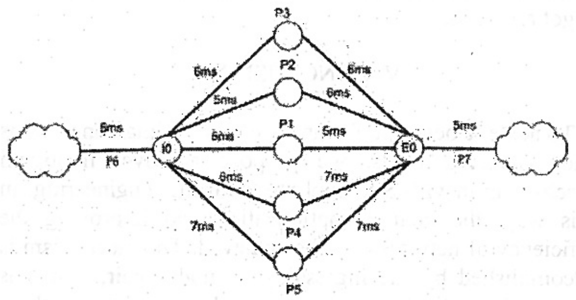

Fig. 2. Topology of simulation network

There is a single IE pair (Ingress node and Egress node) connected by multiple LSPs in the network. Bandwidth of Links except p6 and p7 (4Mbps) is 1Mbps. The traffic rate is 3.5Mbps. The traffic flow will only take the shortest path p1 despite of the overflow under the ordinary MPLS mechanism, while under our proposed mechanism the traffic can be shifted to other paths step by step. And finally the goal of load balance is achieved. The simulation results are shown in TABLE I and Fig. 3 below.

TABLE I

LOSS RATIO AND AVERAGE DELAY OF 2 MECHANISMS

Mechanism	Time (s)	Loss ratio on p1 (%)	Average delay (ms)
Ours	0~15	71.38	44.56
	15~30	66.62	37.85
	30~45	59.94	34.87
	45~60	49.93	32.89
	60~75	33.24	32.06
	75~90	0	25.54
	90~130	0	25.59
Single path	0~130	71.42	63.439

As we can see from TABLE I and, under our proposed mechanism the loss ratio on p1 decreased effectively since the traffic has been shared by all the other paths. And the average delay of packets is also reduced greatly since the congestion has been solved.

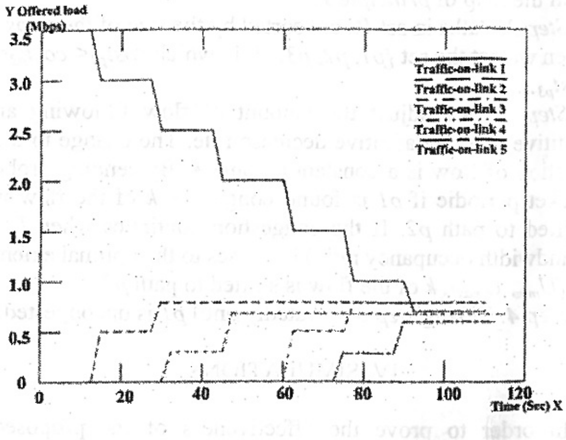

Fig. 3. Offered load under our mechanism

Fig. 3 shows the process of offering load under our mechanism. As we have described above, the process of shifting traffic to other paths follows an additive increase/additive decrease rule. A good state of load balance is got after 90s.

V. CONCLUSIONS

To achieve better performance of controlling congestions and balancing loads, we propose a novel multipath mechanism in MPLS to apply to the traffic engineering. In this way, the goal of optimization and improving the efficiency of networks can be achieved. The mechanism is accomplished by the ingress/egress nodes pair, which is easy and practical. Besides the analysis and research in theory, we had also done some simulations, 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, Ma Ke, and Shan Jianfeng.

VII. REFERENCES

[1] D. Awduche, A. Chui, A. Elwalid, I. Wodjaja, X and Xiao, "Overview and Principles of Internet Traffic Engineering," Available from Internet draft < Draft-ietf-tewg-principles-02.txt >, May 2001.

[2] E. C Rosen, A. Viswanathanm and R. Callon, "Multiprotocol label switching architecture," IETF RFC 3031, January. 2001.

[3] G. R. Ash and B. D. Huang, "An analytical model for adaptive routing networks," *IEEE Trans. Commun,* vol 41, no. 11, 1993, pp. 1748-1759.

[4] R.J. Gibbens, F.P. Kelly and P.B. Key, "Dynamic alternative routing – modeling and behaviour," in: *Teletraffic Science for New Cost-Effective Systems, Networks and Services,* ITC-12, Elsevier, Amsterdam, 1988, pp. 1019-1025.

[5] G. Apostopoulos, et al., "QoS routing mechanism and OSPF extensions," Internet RFC 1633, August. 1999.

[6] E. Crawley, et al., "A framework for QoS-based routing in the Internet," Internet RFC 2386, August. 1998.

[7] Z.-L. Zhang, et al., "Quality of service extension to OSPF," IETF Internet Draft < draft-zhang-qos-ospf-01. txt >, September. 1997.

[8] S. Nelakuditi, Z.-L. Zhang and R.P. Tsang, "Adaptive proportional routing: a localized QoS routing approach," *IEEE Infocom,* 2000, pp. 1566-1575.

[9] D. Zappala, "Alternate path routing for multicast," *IEEE Infocom,* 2000, pp. 1576-1585.

[10] I. stoica, H. Zhang, LIRA: "A model for service differentiatin in the Internet," *NOSSDAV'98,* 1998.

[11] E Gustafsson, G Katsson, "A literature survey on traffic dispersion," *IEEE Network,* vol 11, no. 2, March/April. 1997, pp. 28-36.