

# Improving TCP Performance in Multipath Packet Forwarding Networks

Youngseok Lee, Ilkyu Park, and Yanghee Choi

E-mail: {yslee, xiao, yhchoi}@mmlab.snu.ac.kr

School of Computer Science and Engineering, Seoul National University, Korea

February 23, 2001

## Abstract

This paper investigates schemes to improve TCP performance in multipath forwarding networks. In multipath routing, packets for the same destination are sent to multiple next-hops in either packet-level or flow-level forwarding mode. Effective bandwidth is increased since we can use unused capacity of multiple paths to the destination. In packet-level multipath forwarding networks, TCP performance may not be enhanced due to frequent out-of-order segment arrivals at the receiver because of different delays among paths. To overcome this problem, we propose two TCP modifications. At the sender, the fast retransmission threshold is adjusted considering the number of paths. At the receiver, delayed acknowledgment scheme is modified such that an acknowledgment for an out-of-order segment arrival is delayed in the same way for the in-order one. The number of unnecessary retransmissions and congestion window reductions is reduced, which is verified by extensive simulations. In flow-level multipath forwarding networks, hashing is used at routers to select outgoing link of a packet. Here, we prove that TCP performance is increased in proportion to the number of paths regardless of delay differences.

**Keywords:**TCP, Fast Retransmission, Delayed ACK, Multipath

# 1 Introduction

## 1.1 Multipath Routing

Multipath routing increases the total network utilization and the end-to-end throughput. Most of the current routing protocols in the Internet are based on the simple shortest path routing algorithm which uses a single path from a source to a destination. Although the shortest path algorithm is simple and can be easily implemented, the network resources may not be efficiently utilized. For example, connections along a congested path will experience severe packet delays or losses while there exist alternate unloaded paths between the source and the destination. In this case, the congestion at the bottleneck link or path will be reduced through the increased bandwidth by multiple paths, which improves end-to-end performance. Besides, multipath routing makes the network efficiently utilized by load balancing among redundant paths.

The multipath routing model is defined as a routing model where the routing algorithms provide potentially multiple paths between node pairs. The essential components to make multipath networks viable are algorithms which can compute multiple paths, an efficient multipath forwarding method, and a multipath transport protocol.

Routers can provide multiple paths with recently developed or proposed routing protocols. The easiest extension to multipath routing is to use the equal-cost multiple shortest paths when calculating the shortest one, which is known as Equal-Cost Multi-Path(ECMP) routing. This is explicitly supported by several routing protocols such as Open Shortest Path First(OSPF)[1] and Intermediate System to Intermediate System(IS-IS)[2]. Some router implementations allow equal-cost multipath with Routing Information Protocol(RIP) and other routing protocols. Recently, a multipath extension to the distance vector and link state routing protocols is added in [4]. The shortest  $K$  algorithms, which can find multiple paths with different costs, compute  $K$  shortest paths minimizing the total costs of the path. In the shortest  $K$  algorithms, multiple paths between the source and the destination will have many overlapped links, which may not be effective. On the other hand, calculating link disjoint paths overcomes the disadvantage of the overlapped links. Algorithms for the maximum flow problem find multiple paths with the maximum throughput. However, the maximum flow paths do not provide the end-to-end paths and do not consider the path length.

In the Multi-Protocol Label Switching(MPLS) network, where IP datagrams are switched by looking up the fixed-size label, paths between an ingress router and an egress router can be explicitly set up by a signaling protocol, Explicitly Routed Label Distribution Protocol(ER-LDP) or the modified Resource ReSerVation Protocol(RSVP).

Therefore, multiple Label Switched Paths(LSPs) between an ingress router and an egress router allow the network to utilize the alternate non-shortest path for multipath routing.

A router employing multipath routing protocol distributes incoming packets with the same Forwarding Equivalent Class(FEC) to multiple outgoing links or next-hops. Routers forward packets either in packet-level or flow<sup>1</sup>-level mode(Fig. 1).

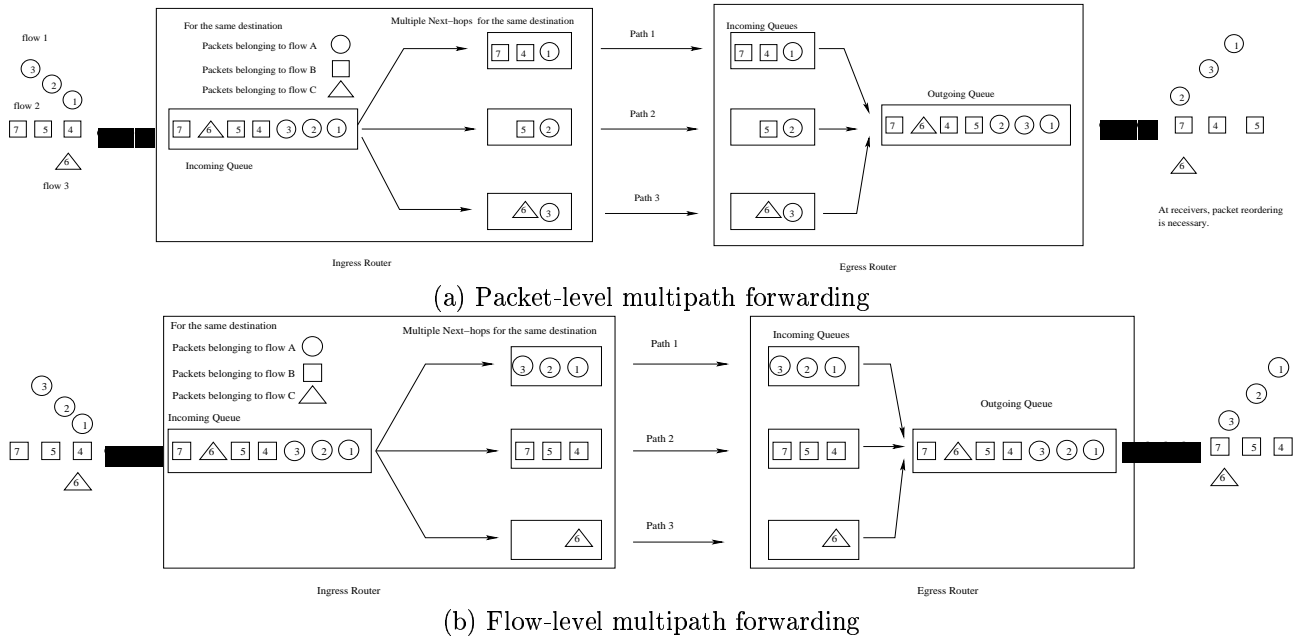


Figure 1: Multipath Packet Forwarding Schemes in Routers: In packet-level forwarding, packet arrivals may be out-of-order, while packet sequence is kept in flow-level forwarding.

In the packet-level forwarding method, packets for the same FEC are delivered to one of multiple next-hops in round-robin or random manner. Therefore, packets belonging to the same flow are forwarded to different next-hops. Although this mechanism can be easily implemented, it will cause many out-of-order packet arrivals at the destination since multiple paths may have different delays. This will degrade end-to-end performance while the buffer requirement is increased. On the contrary, flow-level forwarding routers deliver packets belonging to a flow along the same next-hop. Routers classify packet streams into flows using flow detectors such as the X/Y(X: the number of packets, Y: timeout) flow classifier of the IP switch[3] or the NetFlow of the Cisco routers[18]. However, these schemes require per-flow states in routers, thus not scalable. A hashing function can be used for stateless multipath forwarding[14], and load balancing is achieved by adjusting the hashing range. Yet, it is required that implementation of the hashing function is deployed to all routers. With multiple paths at the network layer, the

<sup>1</sup>In this paper, a flow is defined by packet stream belonging to the same flow specification consisting of the end-point addresses(e.g., source/destination IP addresses and port number) and a timeout between consecutive packet arrivals.

traditional transport protocol will not work. Instead of devising a new multipath transport protocol, we modify the existing transport protocol, namely TCP.

## 1.2 TCP Performance over Multiple Paths

Transmission Control Protocol(TCP) provides the reliable data transfer between node pairs. TCP receivers send an immediate acknowledgment(ACK) when a segment arrives. These ACKs are cumulative and acknowledge all in-order segment arrivals. TCP senders detect a packet loss by a timer expiration or duplicate ACKs. In order to recover the lost packet fast, TCP senders retransmit the lost packet when three duplicate ACKs arrive(*the fast retransmit* algorithm). After the fast retransmission, the congestion avoidance is performed(*the fast recovery* algorithm). An optional *delayed acknowledgment* mechanism[15] allows TCP receivers to send an ACK after a timer expiration(traditionally 200 ms)<sup>2</sup> or at least at every second full-sized segment arrivals. However, when out-of-order segments arrive at a receiver, receivers send duplicate ACKs for each packet to recover the lost packet. Delayed ACKs decrease the amount of data which senders can send because the number of ACKs is reduced, and they are harmful during slow start[11]. However, during the congestion avoidance period, delayed ACKs will conserve network bandwidth and host computation resources by sending fewer ACKs without drastically reducing performance. Delayed ACKs are useful for bulk transfers and asymmetric links[12].

TCP performance over the multiple paths is supposed to be enhanced, because the bottleneck bandwidth increases. However, TCP senders will regard triple duplicate ACKs, resulting from out-of-order packet arrivals at the receiver due to the different delays of the multiple paths, as an indicator of a packet loss, and will retransmit the lost packet(Fast Retransmission). Not only this wastes extra bandwidth, but also it reduces the congestion window(*cwnd*) of the TCP sender(Fast Recovery). Moreover, if ACKs arrive at the TCP sender out-of-order, the late segment with the lower sequence number will be ignored and cannot serve for increasing the *cwnd*, since TCP senders accept only the in-order ACKs. Besides, the variable path Maximum Transfer Unit(MTU) size for each packet will make path MTU discovery useless. The loss rate( $p_i$ ) and the Round-Trip Time( $RTT_i$ ) over  $N$  multiple paths will degrade TCP performance by  $\frac{1}{\sqrt{\sum_i^N p_i}}$  and  $\frac{1}{RTT_i}$ [16]. Hence, TCP performance in packet-level multipath forwarding may not be increased under the conventional environment.

This paper proposes simple TCP modification schemes to prevent the end-to-end performance degradation in packet-level multipath forwarding. Then, it is verified that the hashing-based flow-level multipath forwarding

---

<sup>2</sup>The delay should be less than 500 ms[15], and BSD-based TCP implementations use 200 ms usually.

method linearly increases linearly TCP performance with no need of per-flow states.

The remainder of this paper is organized as follows. The related works are introduced in Section 2. In Section 3 we propose TCP modifications and verify the effectiveness of the hashing-based flow mapping scheme. The results of the performance evaluation by simulation are discussed in Section 4, and Section 5 concludes this paper.

## 2 Related Work

In connection-oriented networks, [5] has analyzed the performance of multipath routing algorithms and has shown that the connection establishment time for multipath reservation is significantly lowered. [6] has proposed a dynamic multipath routing algorithm in connection-oriented networks, where the shortest path is used under light traffic condition and multiple paths are utilized as the shortest path becomes congested. Therefore, in connection-oriented networks, only connection or call-level routing and forwarding are considered. In [9], Quality-of-Service(QoS) routing via multiple paths under time constraint is proposed when the bandwidth can be reserved, assuming all the reordered packets are recovered by the optimal buffer at the receiver, with much overhead of the dynamic buffer adjustment at the receiver. The enhanced routing scheme for load balancing by separating long-lived and short-lived flows is proposed in [7], and it is shown that congestion can be greatly reduced. In [8], it is shown that the quality of services can be enhanced by dividing the transport-level flows into UDP and TCP flows. Yet, it does not consider the aggregated flows. In the MPLS network, a traffic engineering method using multiple multipoint-to-point LSPs is proposed in [10], where backup routes are used against failures. Hence, the alternate paths are used only when primary routes do not work.

Although there have been researches on multipath routing algorithms, the effects on the transport or application layers are largely ignored. Recently, a reliable bit stream multipath transport protocol, MPTCP, is proposed for multipath forwarding networks[4]. Although the MPTCP improves the TCP throughput over multiple paths, it is required that multiple independent TCP connections are established. Therefore, in MPTCP, a sender's data stream is fragmented and attached with two-level(MPTCP and TCP) sequence numbers, and the original data stream is reconstructed at the receiver. Although TCP designed for a single path is naturally extended for the multiple paths, the implementation overhead of the end-host is high.

For the improvement of TCP performance, several modified TCP acknowledgments are proposed. [11] has proposed a byte counting method to improve TCP performance degradation during/after slow start when using delayed ACKs. In asymmetric networks, [12] maintains that TCP performance depends on forward path as well as backward

path, and proposes dynamically varying delayed ACKs. In [13], the effects of extended acknowledgment interval on TCP performance are analyzed, and it is shown that extending ACK interval increases the TCP throughput when two machines are on relatively close networks.

### 3 Two Approaches to Improve TCP Performance over Multiple Paths

This section explains two approaches to enhance TCP performance over the multiple paths according to multipath forwarding capabilities of routers.

#### 3.1 End-host Modifications in Packet-level Multipath Forwarding Networks

When routers support only packet-level multipath forwarding, minor TCP modifications are enough to enhance end-to-end performance. The proposed schemes are either a sender-side solution or a receiver-side one.

##### 3.1.1 Sender-side Solution: Increasing the Fast Retransmission Threshold

The default fast retransmission threshold value(=3) of the TCP senders is not be suitable where in packet-level multipath forwarding networks many out-of-order packet arrivals occur due to delay mismatches among path. At the sender, duplicate ACKs are considered as an indication of packet loss, and the lost segment is retransmitted after triple duplicate ACK arrivals. By increasing the fast retransmission threshold unnecessary retransmissions resulting from out-of-order segment arrivals can be avoided. When long-lived flows are multicast from a sender(or server) to a lot of receivers(or clients) along multiple paths usual in multimedia applications, this solution is very effective because no change is needed at the receivers. We propose a simple heuristic for determining the fast retransmission threshold,  $T$ , as follows.

$$T = 3 \cdot (1 + \lfloor \log_2 N \rfloor) \tag{1}$$

$N$  = number of paths

When two or three paths are used, TCP enters the fast retransmission mode after six duplicate ACKs arrivals. Therefore, receivers have more time to wait for late packet arrivals through different paths.

### 3.1.2 Receiver-side Solution: Modifying Delayed ACKs

When it is difficult to change TCP senders, TCP receivers can employ modified delayed ACKs to achieve TCP performance improvement as follows. Therefore, we propose the following modification of delayed ACKs for TCP receivers to reduce duplicate ACKs.

- *Delaying ACKs for out-of-order segments:* TCP receivers do not send an ACK immediately for each out-of-order packet. Instead, an ACK is generated at every two segment arrivals or after a delayed ACK timer expiration regardless of packet sequence orders.
- *Sending immediately ACKs for retransmitted segments:* If ACKs for out-of-order segments are delayed, it is possible that two or more retransmission events occur. For instance, under the round trip time less than the traditional delayed ACK timer(200ms), although the sender retransmits the lost packet, TCP receivers employing the proposed delayed ACK may wait until two segments arrive or the timer expires, which causes a retransmission timeout. Therefore, receivers should send an ACK immediately for a retransmitted segment during the duplicate ACK period.

By the proposed delayed ACK method, the probability that the window reduction resulting from triple duplicate ACKs by out-of-order packet arrivals will be decreased, thereby avoiding TCP performance degradation.

In general, delayed ACKs may decrease the TCP throughput by  $\frac{1}{\sqrt{b}}$  ( $b$  = number of packets per ACK)[16] in a single path. In addition, receivers employing modified delayed ACKs respond slowly to a packet loss because ACKs for out-of-order packets are delayed. However, when packet arrivals are frequently out-of-order in packet-level multipath forwarding networks, TCP performance will be enhanced with modified delayed ACKs. Furthermore, by reducing the number of ACKs, the receiver-side solution is useful in the asymmetric network where up link bandwidth is scarce.

### 3.2 Router-based Approach: Flow-level Multipath Forwarding

When forwarding is done in flow by flow, routers send packets belonging to the same TCP flow to the same next-hop. Therefore, out-of-order packet arrivals will be prohibited in advance. End hosts are not changed and at routers we use the hashing-based flow mapping scheme[14] that does not need flow states. Flow-aware routers generally keep per-flow states in memory to detect a flow and map it into an appropriate path. However, the hashing-based method removes the scalability problem with the stateless operation, which becomes suitable in high-speed

networks. Although it is required that the additional implementation of the hashing function should be deployed to all the multipath-capable routers, end-to-end performance as well as the network resource utilization will be increased in proportion to the number of paths without the overhead of the end host modification.

## 4 Performance Evaluation

### 4.1 Effects of the Number of Multiple Paths with Different Delays

We used the ns[17] network simulator for our experiments. The network topology is shown in Fig. 2, where upto five paths can be used between a source and a destination.

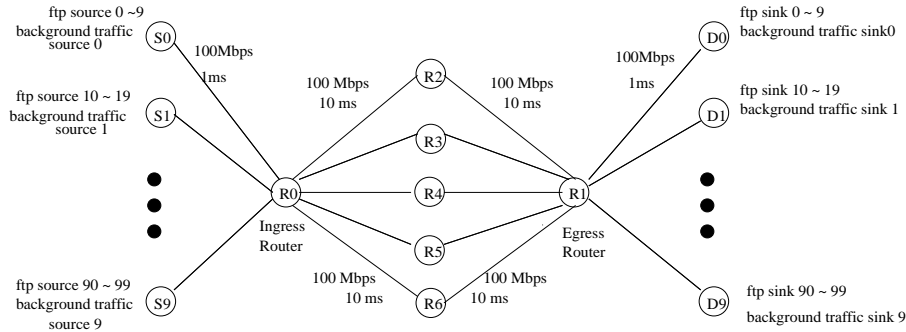


Figure 2: Network Topology

All the access links( $S_i - R_0, R_1 - D_i$ ) have bandwidths of 100 Mbps and link delays of 1 ms, while  $R_i - R_j$  links in the core network have longer delays(10ms) with the same bandwidths(100Mbps). For each node pair ( $S_i, D_i$ ), ten FTP/TCP(NewReno) connections are opened. Therefore, in total, one hundred FTP connections are run. The TCP senders use the segment size of 512 bytes, and the advertised window size is 100. For the background traffic, ten Pareto connections(one for each node) compete for the bottleneck links between  $R_0$  and  $R_1$ . The Pareto traffic sources are configured with the packet size of 500 byte, burst time of 500 ms, idle time of 500 ms, shape value of 1.5. The peak rate of each Pareto source is 10 Mbps, which will cause congestion in case of a single path between  $R_0$  and  $R_1$ . To examine the effects of multipath forwarding, up to five paths can be used between  $R_0$  and  $R_1$ . The router adopts the drop-tail queue discipline for simplicity.

Fig. 3 shows the TCP throughput as the fast retransmission threshold is increased under the different path delays. The relative TCP goodput ratio is the TCP goodput of multiple paths over TCP goodput of a single path. TCP performance increases linearly. TCP senders with the default fast retransmission threshold(3) do not benefit from the increased bandwidth, and TCP performance is even lower than that of the single path. On the other



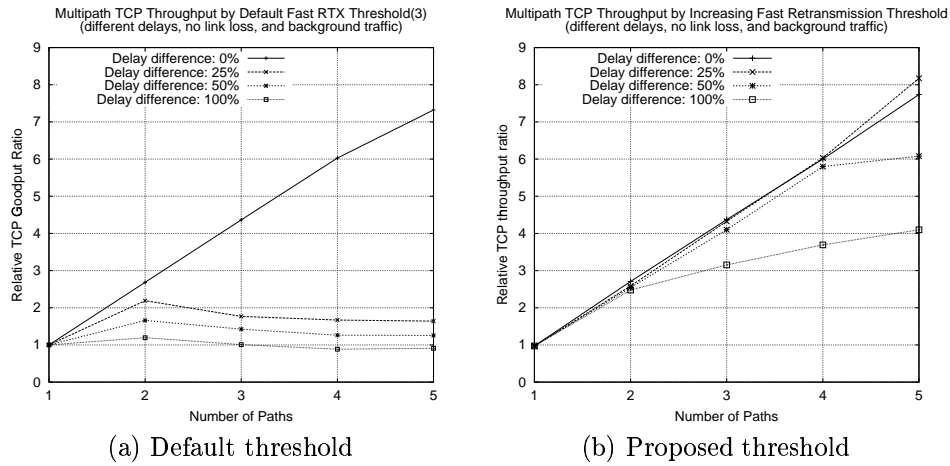


Figure 3: TCP performance according to the fast retransmission threshold in a packet-level forwarding network: the fast retransmission threshold is 6 and 9 for 2-3 and 4-5 multiple paths, respectively

hand, TCP performance of the proposed fast retransmission threshold method is drastically improved regardless of different path delays.

Especially, in case of five paths, TCP performance of 25 % path delay difference outperforms that of the same path delays, because most out-of-order packet arrivals are recovered and packet drops are reduced by the increased *delay-bandwidth* product. However, as the difference in path delays becomes large, many out-of-order packet arrivals degrade TCP performance.

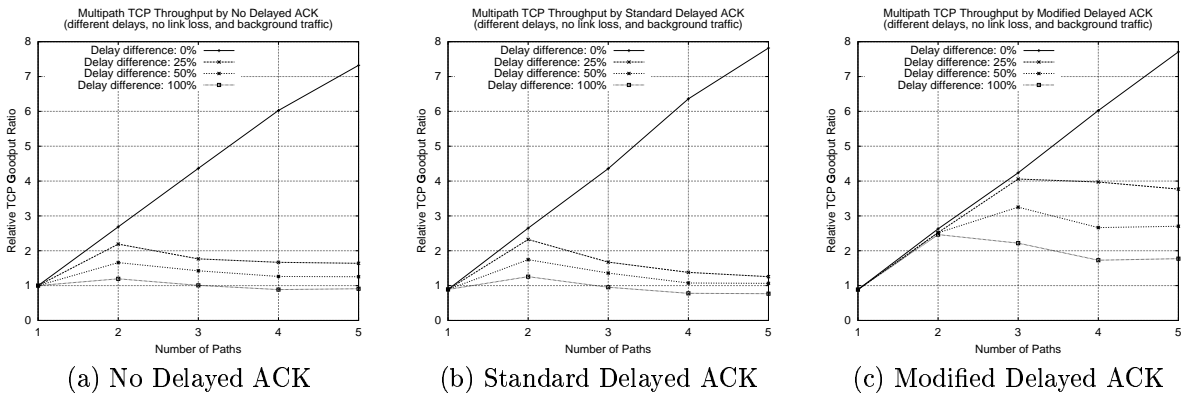


Figure 4: TCP performance according to the types of TCP receivers in a packet-level forwarding network

Fig. 4 depicts the relative TCP throughput ratio by proposed delayed ACKs. As the path delay difference becomes large, TCP performance by the non-delayed or the standard delayed ACKs is severely degraded, even lower than that of a single path, because many out-of-order TCP segment arrivals are considered as packet losses. When receivers adopt modified delayed ACKs(Fig. 4-(c)), TCP performance is improved. In case of two or three

paths, the modified delayed ACK method increases the TCP throughput more than two times in spite of the different path delays. However, when the number of paths is four or five, TCP performance is not any more improved by the modified delayed ACK. Therefore, we conclude that when routers support only packet-level multipath forwarding, two or three paths are enough for modified delayed ACKs.

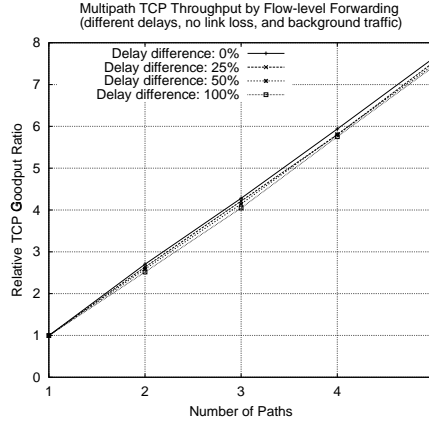


Figure 5: TCP performance in a flow-level forwarding network

Next, we replace packet-level forwarding routers with flow-level packet forwarding ones. Each TCP flow is mapped to one of multiple next-hops by hashing the source address. As shown in Fig. 5, the TCP throughput of multiple paths in a flow-level forwarding network increases in proportion to the number of paths irrespective of the different delays, because packet sequences are maintained for every TCP flow along each path.

## 4.2 Effects of Explicit Multiple Paths

The explicit path setup of the MPLS network provides increased TCP performance when combined with flow-level multipath forwarding. Through the explicit Label Switched Paths(LSPs), the non-shortest paths can be efficiently utilized for multipath routing. Therefore, when flow-level forwarding routers provide explicit multiple paths instead of equal cost multipath, both the network resource utilization and the end-to-end throughput can be maximized.

We examine a more complete network(Fig. 6) to see the effects of explicit multiple paths. In this network topology(Fig. 6-(a)), when a single shortest path is used between  $R_0$  and  $R_1$ , the bottleneck bandwidth between senders and receivers would be 100 Mbps of the  $R_2 - R_4 - R_1$  path. If ECMP is supported, two equal-cost multiple paths( $R_0 - R_2 - R_3 - R_1$ , and  $R_0 - R_2 - R_4 - R_1$ ) increase the bottleneck bandwidth to 155 Mbps of the  $R_0 - R_2$  link, and the background traffic will be divided into two paths by half. In the MPLS network, however, three explicit paths(the above two, and non-shortest  $R_0 - R_5 - R_6 - R_7 - R_1$ ) can be set up and used for multipath

forwarding, thereby increasing the bottleneck bandwidth from 155 Mbps to 255 Mbps.

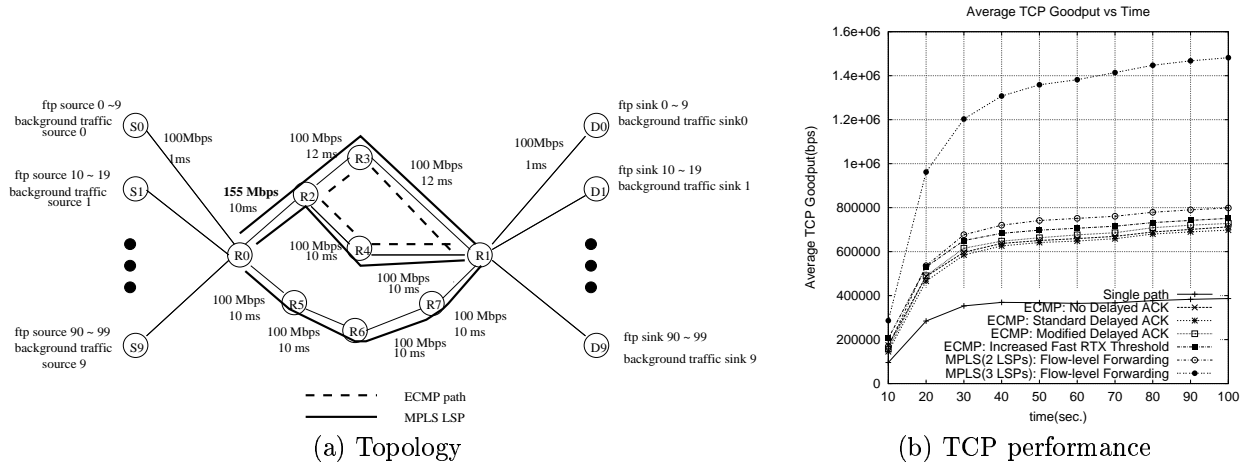


Figure 6: TCP performance by explicit multiple paths in an MPLS network vs ECMP-based network.

Under the same FTP/TCP configuration to the previous simulation, the average TCP goodput of one hundred FTP flows is shown in Fig. 6-(b). In multipath routing with ECMP support, modified delayed ACKs enhance TCP performance by 80 %, when compared to single path case. Also, TCP performance of the fast retransmission threshold of six is improved by 94 %. In the flow-level forwarding MPLS network with two(three) LSPs, TCP performance is 2.1(3.8) times more than that of single path.

### 4.3 Effects of Different Background Traffic

In order to investigate the effect of background traffic on TCP performance, our simulation is performed on the very high performance Backbone Network Service(vBNS)-like topology(Fig. 7)[19]. The backbone links are configured as OC-12(622 Mbps) and link delays are set in proportion to the distance between cities, while access links have 100 Mbps of bandwidth with 1 ms of delay. One hundred FTP connections(ten FTP senders/receivers per each leaf node) are concurrently tested along the path between San Francisco and Washington D.C with the Pareto-distributed background traffic(one Pareto source/sink per each leaf node). We changed the peak rate of each Pareto source for the different traffic load; lightly-loaded(30 Mbps), highly loaded(60 Mbps), and congested(70 Mbps).

In Fig. 8, it is shown that TCP performance over multiple paths is improved by the fast retransmission threshold heuristic, the modified delayed ACK mechanism, and the flow-level forwarding MPLS network under different background traffic loads. With proposed delayed ACKs, TCP performance increases by 43, 82, and 94% for the lightly-loaded, heavily-loaded, and congested path, compared with the single path case. Also, the adjusted

fast retransmission threshold improves TCP goodput by 56, 82, and 92 %. With flow-level forwarding in MPLS networks, TCP goodput is increased by 61, 111, and 122 %. However, when the traditional TCP version of non-delayed or standard delayed ACKs is used in the ECMP routing, TCP performance is degraded for various background traffic loads, even lower than that of a single path in the lightly loaded environment. Although the path is lightly loaded, many out-of-order packet arrivals, which are considered as losses, still occur, and they reduce the sender's window by the fast retransmission/fast recovery.

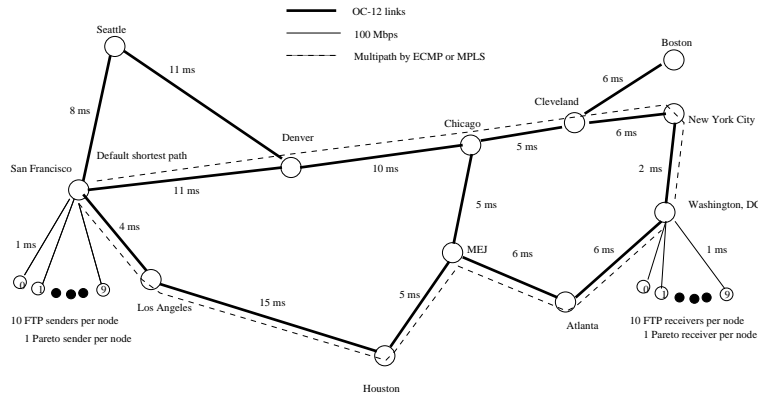


Figure 7: vBNS-like Network Topology

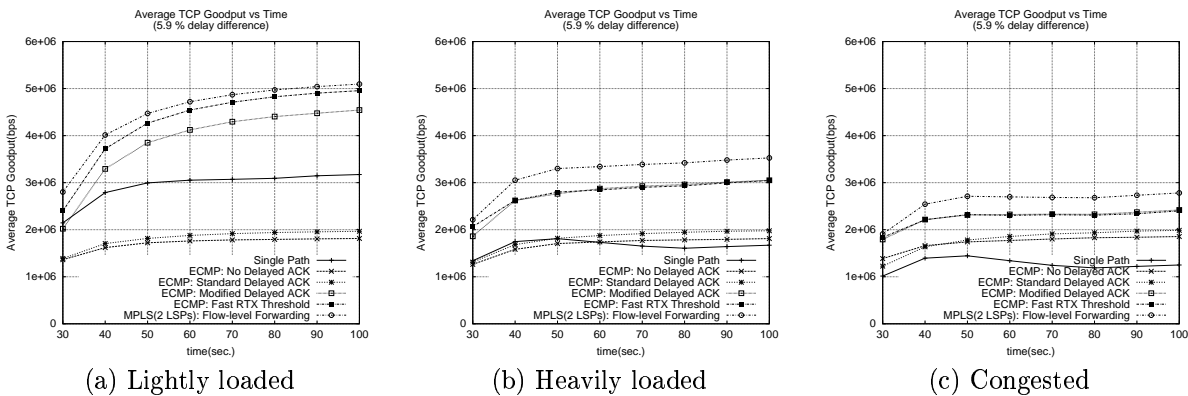


Figure 8: TCP performance by different background traffic

## 5 Conclusion

In this paper, we propose two TCP modifications for multipath forwarding networks, and show that TCP performance is greatly improved even in a simple packet-level multipath forwarding networks. Increasing the fast retransmission threshold makes TCP senders wait for more than triple duplicate ACKs, which reduces the number of the fast retransmission and the fast recovery events. This method is suitable when a few TCP senders push

long-lived flows to many receivers. The proposed delayed ACK method enhances TCP performance by generating less ACKs at TCP receivers, because out-of-order packet arrivals are recovered while ACKs are being delayed. This method achieves performance improvement similar to the sender-side solution, and is useful when sender modification is impossible. By the simulation results, it is seen that when the TCP modifications are employed, most out-of-order packet arrivals from different path delays are recovered and TCP performance is enhanced nearly by double especially for two paths. Therefore, TCP modification methods will be practically useful for the conventional packet-level multipath network. On the other hand, it is verified that hashing-based flow-level multipath forwarding routers increase TCP performance in proportion to the number of paths in spite of the differences in the path delays. Moreover, it has been shown that explicit multiple paths in the MPLS network, when combined with flow-level multipath forwarding, improves both the end-to-end throughput and the network utilization by using the non-shortest paths. This feature can be used for the efficient Internet traffic engineering.

## References

- [1] J. Moy, "OSPF Version 2," Internet RFC2328, 1998
- [2] R. Callon, "Use of OSI IS-IS for Routing in TCP/IP and Dual Environments," Internet RFC1195, 1990
- [3] P. Newman, T. Lyon, and G. Minshall, "Flow Labeled IP: A Connectionless Approach to ATM," Infocom'96, 1996
- [4] J. Chen, P. Druschel, and D. Subramanian, "An Efficient Multipath Forwarding Method," Infocom'98, 1998
- [5] I. Cidon, R. Rom, and Y. Shavitt, "Analysis of Multi-Path Routing," IEEE/ACM Transaction on Networking, vol. 7, no. 6, pp. 885 - 896, Dec. 1999
- [6] S. Bahk, and M. Zarki, "Dynamic Multi-path Routing and How it Compares with other Dynamic Routing Algorithms for High Speed Wide Area Networks," ACM Computer Communications Review, vol. 22, no. 4, pp. 54-64, Oct. 1992
- [7] A. Shaikh, J. Rexford, and K. G. Shin, "Load-Sensitive Routing of Long-Lived IP Flows," SIGCOMM'99, 1999
- [8] P. Bhaniramka, W. Sun, and R. Jain, "Quality of Service using Traffic Engineering over MPLS: An Analysis," Globecom'99, 1999

- [9] N. S. V. Rao, and S. G. Batsell, "QoS Routing Via Multiple Paths Using Bandwidth Reservation," Infocom'98, 1998
- [10] H. Saito, Y. Miyao, and M. Yoshida, "Traffic Engineering using Multiple Multipoint-to-Point LSPs," Infocom'2000, 2000
- [11] M. Allman, "On the Generation and Use of TCP Acknowledgements," ACM Computer Communication Review, vol. 28 no. 5, pp. 4-21, Oct. 1998
- [12] H. Balakrishnan, V. N. Padmanabhan, and R. H. Katz, "The Effects of Asymmetry on TCP Performance," ACM MobiCom, Sept. 1997
- [13] S. Johnson, "Increasing TCP Throughput by Using an Extended Acknowledgement Interval," Master's Thesis, Ohio University, June 1995
- [14] Z. Cao, Z. Wang, and E. Zegura, "Performance of Hashing-Based Schemes for Internet Load Balancing," Infocom'2000, 2000
- [15] R. Braden, "Requirements for Internet Hosts – Communication Layers," RFC 1122, Oct. 1989
- [16] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, "Modeling TCP Throughput: A Simple Model and its Empirical Validation," SIGCOMM'98, 1998
- [17] Network Simulator - ns(version 2), <http://www-mash.cs.berkeley.edu/ns>
- [18] Cisco NetFlow, [http://www.cisco.com/warp/public/cc/pd/iosw/ioft/neftct/tech/napps\\_wp.htm](http://www.cisco.com/warp/public/cc/pd/iosw/ioft/neftct/tech/napps_wp.htm)
- [19] very high performance Backbone Network Service(vBNS), <http://www.vbns.net>