Improving TCP Performance in Multipath Packet Forwarding Networks

Youngseok Lee, Ilkyu Park, and Yanghee Choi

Abstract: **This paper investigates schemes to improve TCP performance in multipath forwarding networks. In multipath routing, packets to 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 utilize 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 simple TCP modifications. At the sender, the fast retransmission threshold is adjusted taking the number of paths into consideration. At the receiver, the 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 diminished, 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 show by simulations that TCP performance is increased in proportion to the number of paths regardless of delay differences.**

Index Terms: **TCP, fast retransmission, delayed ACK, multipath.**

I. INTRODUCTION

A. Multipath Routing

Multipath routing is employed in order to increase the total network utilization and the end-to-end throughput. The inter-domain routing protocols such as Border Gateway Protocol (BGP) [1] in the Internet are usually based on policy, while most intra-domain routing protocols such as Open Shortest Path Forwarding (OSPF) [2] and Intermediate System to Intermediate System (IS-IS) [3] adopt the simple shortest path routing algorithm which basically uses a single path from a source to a destination. Although the shortest path algorithm is simple and can easily be implemented, 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, 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 efficiently utilizes the network by balancing the load among redundant paths.

The multipath routing model is defined as a routing model where routing algorithms provide potentially multiple paths between node pairs. 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 [2]. This is explicitly supported by several routing protocols such as OSPF and IS-IS. Some router implementations allow equal-cost multipath with Routing Information Protocol (RIP) or other routing protocols [4]. Recently, a multipath extension to the distance vector and link state routing protocols is added in [5], [6].

In the Multi-Protocol Label Switching (MPLS) [7] 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, Constraint-based Routed Label Distribution Protocol (CR-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[1]-level mode (Fig. 1).

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. In order to remedy possible end-to-end performance degradation, an additional buffer management scheme to fix the mis-ordering packets is required. 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 [8] or the NetFlow of the Cisco routers [9]. The flow detectors will

Manuscript received February 23, 2001; approved for publication by Danny Raz, Division III Editor, January 8, 2002.

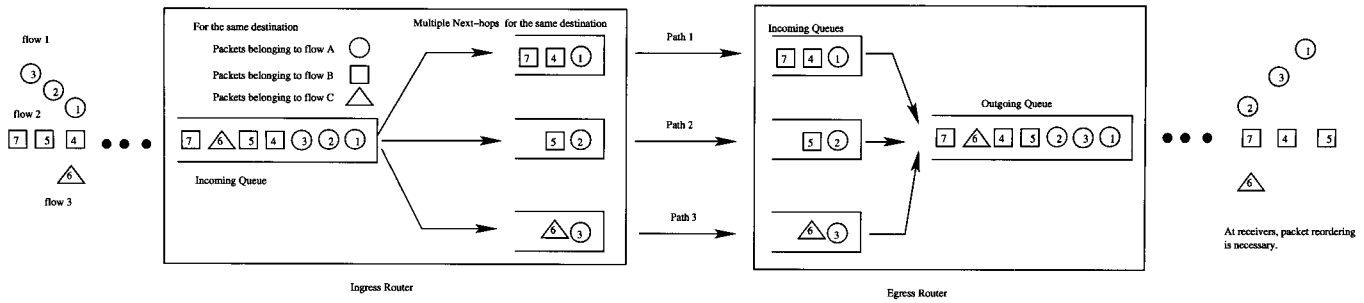
Y. Lee is with Multimedia and Computer Communication Laboratory, School of Computer Science and Engineering, Seoul National University, Seoul, Korea, e-mail: yslee@mmlab.snu.ac.kr.

I. Park is with Distributed Virtual Reality Research Team, Electronics and Telecommunications Research Institute (ETRI), Taejon, Korea, e-mail: xiao@etri.re.kr.

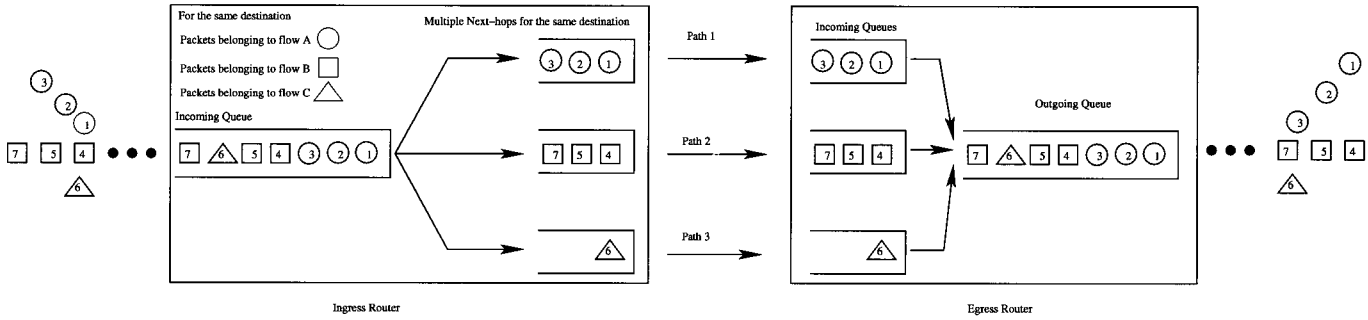
Y. Choi is with Multimedia and Computer Communication Laboratory, School of Computer Science and Engineering, Seoul National University, Seoul, Korea, e-mail: yhchoi@mmlab.snu.ac.kr.

[1] 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.

1229-2370/02/$10.00



(a) Packet-level multipath forwarding



(b) Flow-level multipath forwarding

Fig. 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.

consider packet streams belonging to the same flow specification as a flow when X packets arrive within Y timeout. That is, if the number of packet arrivals for the given flow specification is less than X within Y timeout, the flow will not be detected or will be deleted from the flow table. However, these schemes require per-flow states in routers, thus not scalable. A hashing function can be used for stateless multipath forwarding [10], and load balancing is achieved by adjusting the hashing range. Yet, it is required that implementation of the hashing function is deployed to all the routers supporting multipath routing. With multiple paths at the network layer, the traditional transport protocol will not perform well. Instead of devising a new multipath transport protocol [11], we propose a simple modification of the existing transport protocol, namely TCP.

B. TCP Performance over Multiple Paths

Transmission Control Protocol (TCP) provides 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 [12] allows TCP receivers to send an ACK after a timer expiration (traditionally 200 ms[2]) 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 [13]. 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 [14].

TCP performance over 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 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 $\frac{1}{RTT_i}$ [15]. 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 end-to-end performance degradation in packet-level multipath forwarding. Then, it is verified that the hashing-based

[2]The delay should be less than 500 ms [12], and BSD-based TCP implementations use 200 ms usually.

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

The remainder of this paper is organized as follows. Related works are introduced in Section II. In Section III, we propose TCP modifications and verify the effectiveness of the hashing-based flow mapping scheme. The results of performance evaluation by simulation are discussed in Section IV, and Section V concludes this paper.

II. RELATED WORKS

In connection-oriented networks, Cidon *et al.* have analyzed the performance of multipath routing algorithms and have shown that the connection establishment time for multipath reservation is significantly lowered [16]. Park and Zarki have 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 [17]. Therefore, in connection-oriented networks, only connection or call-level routing and forwarding are considered. 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 [18]. The enhanced routing scheme for load balancing by separating long-lived and short-lived flows is proposed and it is shown that congestion can be greatly reduced [19]. It is shown that the quality of services can be enhanced by dividing the transport-level flows into UDP and TCP flows [20]. Yet, it does not consider the aggregated flows. In the MPLS network, a traffic engineering method using multiple multipoint-to-point LSPs is proposed [21]. As backup routes are used against failures, 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 [11]. 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. Allman has proposed a byte counting method to improve TCP performance degradation during/after slow start when using delayed ACKs [13]. In asymmetric networks, Balakrishnan *et al.* maintain that TCP performance depends on forward path as well as backward path, and propose dynamically varying delayed ACKs [14]. 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 [22].

III. 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.

A. *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, a receiver-side one, or an integrated scheme that both the sender and the receiver use TCP modification.

A.1 Sender-side Solution: Increased Fast Retransmission Threshold

The default fast retransmission threshold value (= 3) of the TCP senders may 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, under the given number of multiple paths, N. Without loss of generality, the number of practical multiple links or paths will be restricted to a small value (e.g., 2 - 8), because multiple next-hops for the same destination correspond to interfaces at a router. To take advantage of practical multiple paths, we classify multiple paths into the range of $[2^{N-1}, 2^N - 1]$, which may be thought as a small multipath set consisting of two or three paths, a middle-sized multipath set of 4 - 7 paths, a large one with 8 - 15 paths, and the huge set with over 16 paths. Then, for each multipath range, we increase the fast retransmission threshold by the number of duplicate ACKs, D, which is set to three for a single path by default, as follows.

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

Fig. 2 shows the fast retransmission threshold as the number of multiple paths increases. When two or three paths are used, TCP enters the fast retransmission mode after six duplicate ACK arrivals. Therefore, receivers have more time to wait for late packet arrivals through different paths. Determining the fast retransmission threshold is validated by simulation in the next section.

A.2 Receiver-side Solution: Modified 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

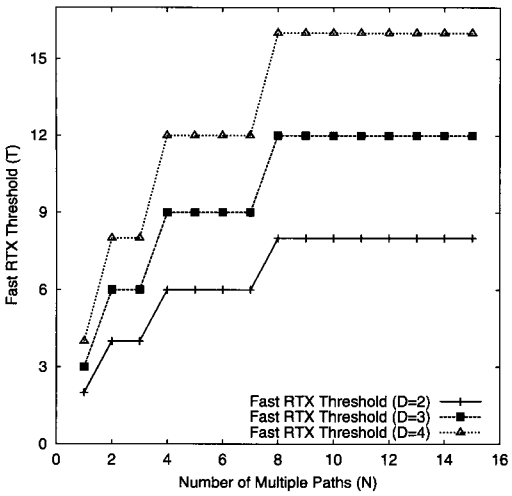


Fig. 2. Fast retransmission threshold according to the number of multiple paths.

modification of delayed ACKs for TCP receivers to reduce duplicate ACKs.

1. **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.
2. **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 (200 ms), 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.

With 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) [15] 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.

A.3 Integrated Solution: Increased Fast Retransmission Threshold and Modified Delayed ACKs

When both the sender and the receiver use the modified TCP, the probability that the mis-ordered TCP segments will be recovered at the receiver increases in multipath routing. In the single shortest path-based routing, however, TCP throughput will be degraded, because it takes longer for the sender to detect the real segment loss while the receiver delays ACKs and the sender waits for additional duplicate ACKs.

B. 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 [10] 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.

IV. PERFORMANCE EVALUATION

A. Effects of Increased Fast Retransmission Threshold

We used the ns network simulator [23] for our experiments. The network topology is shown in Fig. 3, where up to five paths can be used between a source and a destination.

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 (10 ms) with identical bandwidth (100 Mbps). 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, and 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.

For the comparison of TCP performance, we use the relative TCP goodput which is computed as the total received TCP segments except retransmission of the proposed scheme over that of the single path case of the standard TCP sender and receiver. It is shown in Fig. 4 that, when the link delays are different (100 %) and link loss is 0.001, TCP performance is generally enhanced as the fast retransmission threshold increases in multipath routing; however, TCP performance in single path routing begins to degrade when the fast retransmission threshold is greater than six. Under two or three multiple paths, it is shown that TCP performance is not much enhanced after six or more duplicate ACKs. In case of four or five multiple paths, increasing the

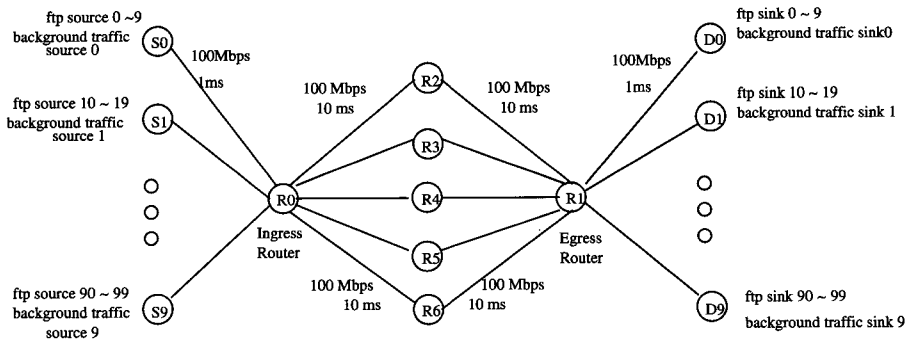


Fig. 3. Network topology for simulation.

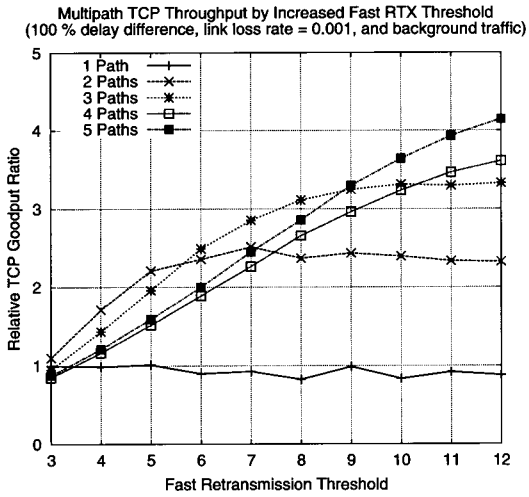


Fig. 4. The effects of changing the fast retransmission threshold.

fast retransmission threshold continuously enhances TCP performance. Yet, changing the threshold affects TCP performance differently according to the number of multiple paths. For instance, with the threshold of six, TCP performance of two or three paths shows better than that of four or five paths, while with the threshold of nine, TCP performance of four or five paths is better than that of two or three paths. Therefore, we use the heuristic for determining the fast retransmission threshold for the number of multiple paths in Eq.(1).

B. *Effects of the Number of Multiple Paths with Different Delays and Loss Rates*

In the same simulation environment (Fig. 3), we investigate the effects of the number of paths with different delays and link loss rates.

Fig. 5 shows the TCP throughput as the fast retransmission threshold is increased under the different path delays and link loss rate. 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 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. Still, under lossy links, TCP throughput increases as the number of paths by increasing the fast retransmission threshold.

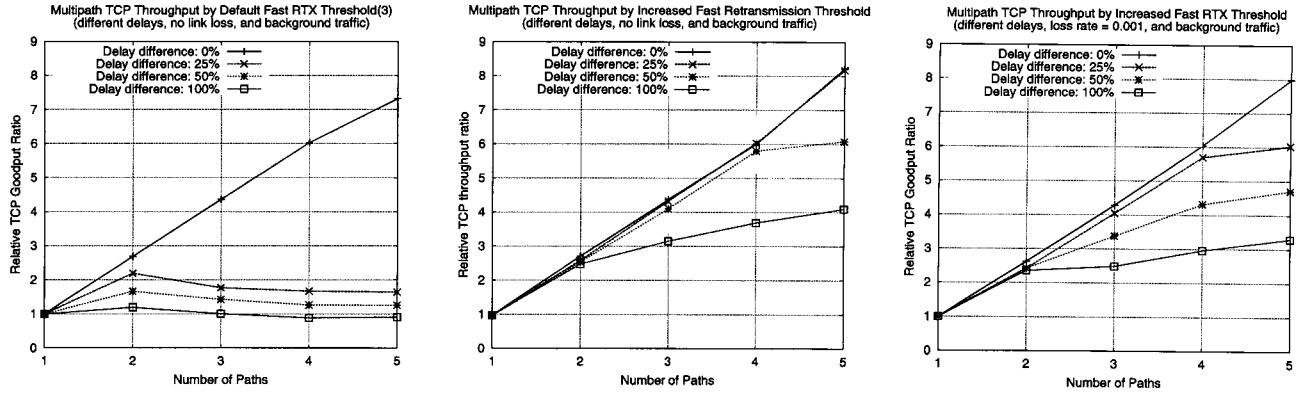
Fig. 6 depicts the relative TCP throughput ratio by proposed delayed ACKs. As the path delay difference becomes large, TCP performance by the non-delayed or the standard delayed ACKs is severely degraded, even lower than that of a single path, because many out-of-order TCP segment arrivals are considered as packet losses. When receivers adopt modified delayed ACKs (Fig. 6-(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. Fig. 7 shows the relative TCP throughput ratio when link loss rates are added. Under link loss rate of 0.001, TCP throughput by modified delayed ACK is enhanced. Therefore, we conclude that when routers support only packet-level multipath forwarding, two or three paths are enough for modified delayed ACKs.

In Fig. 8, TCP performance is shown when both the sender and the receiver use the modified TCP. The integrated scheme enhances TCP performance more than the sender-only or receiver-only method, because the packet recovery probability increases.

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. 9, 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 and link loss, because packet sequences are maintained for every TCP flow along each path.

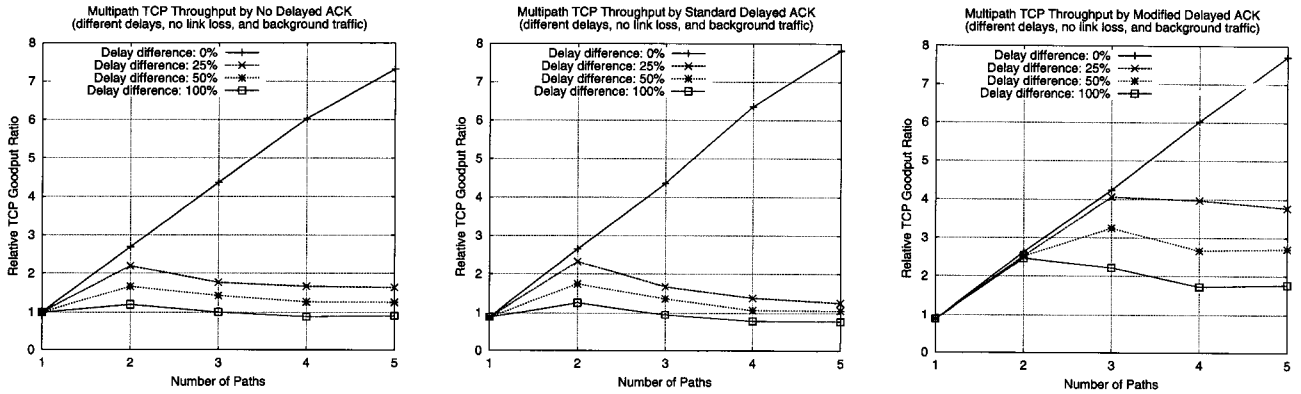
C. *Effects of Partial and Explicit Multiple Paths*

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



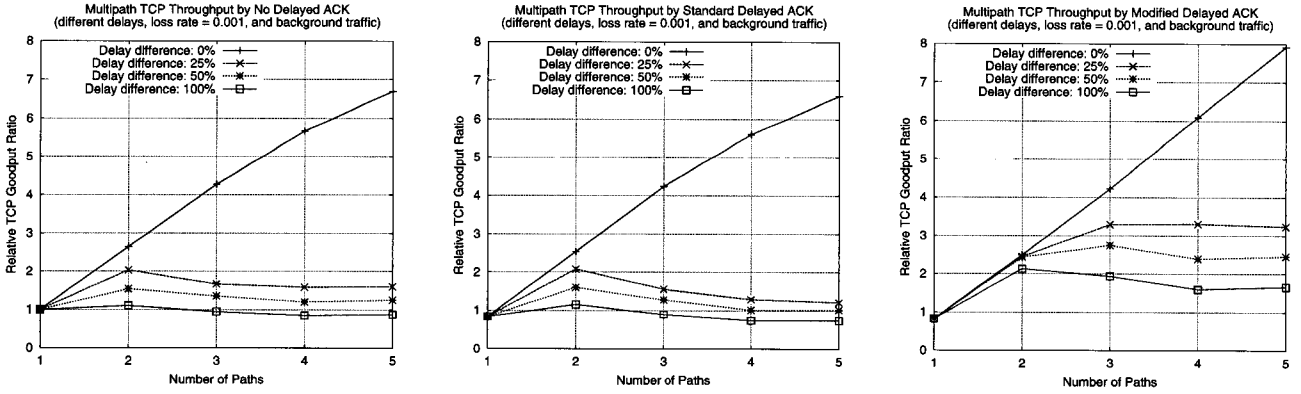
(a) Default threshold (b) Proposed threshold (c) Proposed threshold (link loss)

Fig. 5. 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.



(a) No Delayed ACK (b) Standard Delayed ACK (c) Modified Delayed ACK

Fig. 6. TCP performance according to the types of TCP receivers in a packet-level forwarding network.

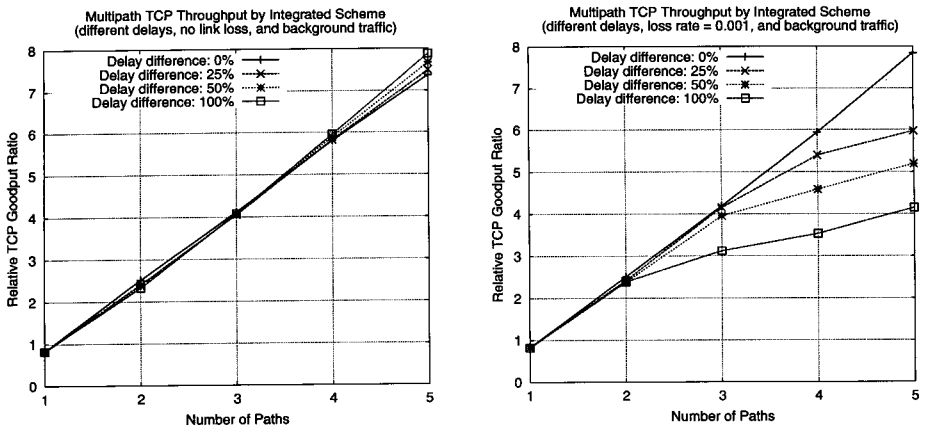


(a) No Delayed ACK (b) Standard Delayed ACK (c) Modified Delayed ACK

Fig. 7. TCP performance according to the types of TCP receivers in a packet-level forwarding network with lossy links.

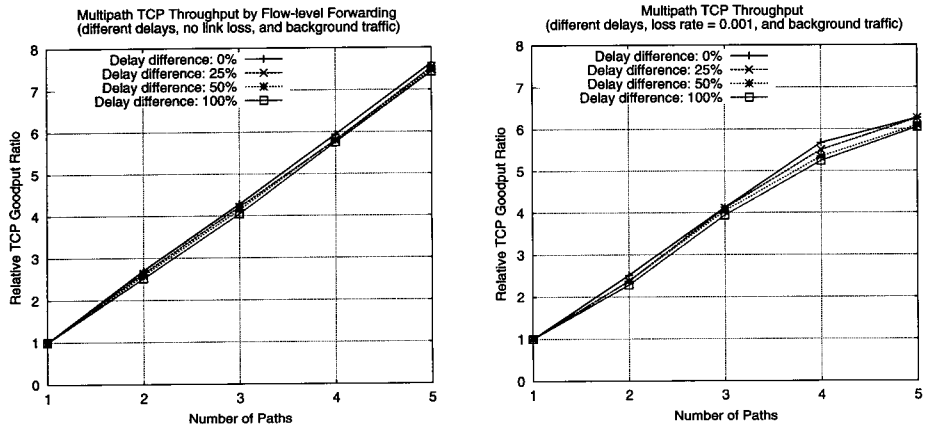
(LSPs), the non-shortest paths can be efficiently utilized for multipath routing. Therefore, when flow-level forwarding routers provide explicit multiple paths instead of equal cost multipath, both the network resource utilization and the end-to-end throughput can be maximized.

We examine a more complex network (Fig. 10) to see the effects of partial and explicit multiple paths. In this network topology (Fig. 10-(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



(a) Without link loss (b) With link loss

Fig. 8. TCP performance by the integrated scheme that both the sender and the receiver use the modified TCP.



(a) Without link loss (b) With link loss

Fig. 9. TCP performance in a flow-level forwarding network.

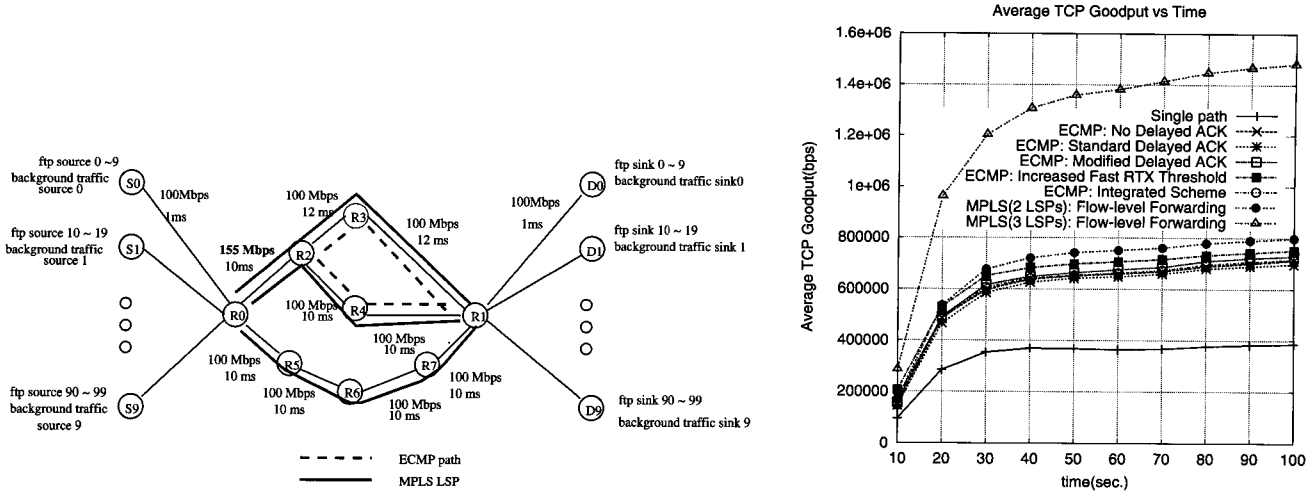
to 155 Mbps of the $R_0 - R_2$ link, and the background traffic will be divided into two paths by half. In the MPLS network, however, three explicit paths (the above two, and non-shortest $R_0 - R_5 - R_6 - R_7 - R_1$) can be set up and used for multipath forwarding, thereby increasing the bottleneck bandwidth from 155 Mbps to 255 Mbps.

Under the same FTP/TCP configuration to the previous simulation, the average TCP goodput of one hundred FTP flows is shown in Fig. 10-(b). In multipath routing with ECMP support, modified delayed ACKs and integrated scheme 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.

To take advantage of all the available multiple paths between R_0 and R_1 in ECMP routing in the same way as MPLS networks, we adjust the metric of $R_0 - R_2$ link to half of the default link metric. Therefore, three multiple paths available in ECMP routing are same as explicit multiple LSPs in the MPLS network. In this network configuration, Fig. 11 shows TCP performance by various types of senders and receivers. In Fig. 11-(a), we can see that all the proposed TCP modification schemes outperform the standard TCP method and the two explicit LSPs in the MPLS network. However, three explicit LSPs in the MPLS network utilize the maximum TCP throughput of the available bandwidth along multiple paths between R_0 and R_1. When the link loss rate of 0.001 is added (Fig. 11-(b)), fast retransmission based TCP modification methods still enhance TCP performance more than modified delayed ACK. Yet, flow-level multipath packet forwarding with three explicit multiple paths in the MPLS network achieves the maximum TCP throughput.

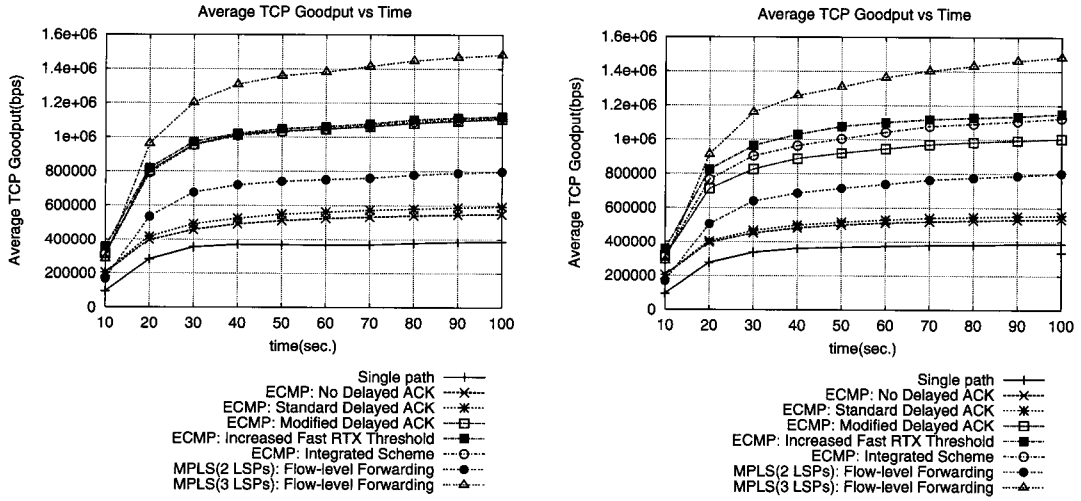
D. *Effects of Different Background Traffic*

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. 12) [24]. 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



(a) Topology (b) TCP performance

Fig. 10. TCP performance by explicit multiple paths in an MPLS network vs ECMP-based network.



(a) Without link loss (b) With link loss

Fig. 11. TCP performance by explicit multiple paths in an MPLS network vs ECMP-based network after the link ($R_0 - R_2$) metric is adjusted.

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. 13, 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 %. When the integrated scheme is applied, TCP performance increases are 57, 84, and 95 %. 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.

On the other hand, if the short-lived TCP flows, found in WWW applications, use multipath routing, TCP performance enhancement is not so significant as in the case of ftp. However, as shown in the simulation results, the short-lived flows still benefit from multipath routing. In addition, if we consider that web sessions now include more long-lived and heavy file retrievals for multimedia applications, multipath routing will be-

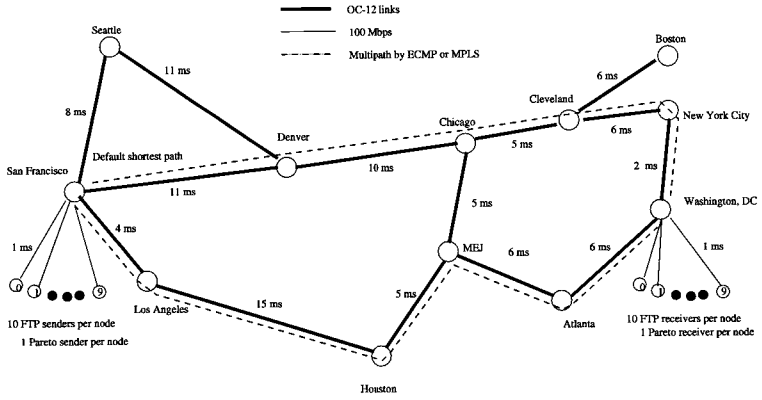
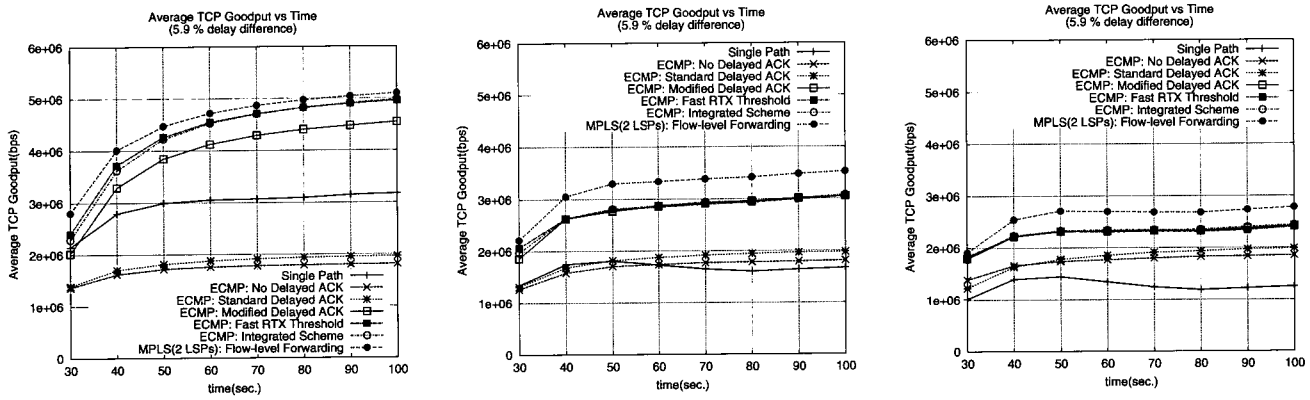


Fig. 12. vBNS-like network topology.



(a) Lightly loaded (b) Heavily loaded (c) Congested

Fig. 13. TCP performance by different background traffic.

come more useful.

V. CONCLUSION

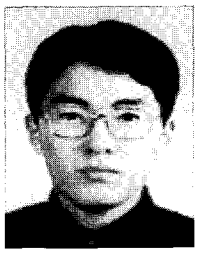
In this paper, we propose TCP modifications for multipath forwarding networks, and show that TCP performance is greatly improved even in a simple packet-level multipath forwarding networks. Increasing the fast retransmission threshold makes TCP senders wait for more than triple duplicate ACKs, which 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. The integrated scheme, which both senders and receivers use TCP modification, also improves TCP performance. 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

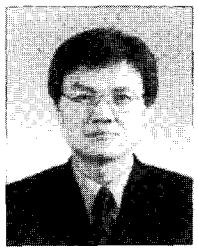
[1] Y. Rekhter and T. Li, "A border gateway protocol," IETF RFC 1771, 1995.
[2] J. Moy, "OSPF version 2," IETF RFC 2328, 1998.
[3] R. Callon, "Use of OSI IS-IS for routing in TCP/IP and dual environments," IETF RFC 1195, 1990.
[4] D. Thaler and C. Hopps, "Multipath issues in unicast and multicast nexthop selection," IETF RFC 2991, 2000.
[5] J. Chen, P. Druschel, and D. Subramanian, "An efficient multipath forwarding method," in *Proc. IEEE INFOCOM'98*, 1998, pp. 1418–1425.
[6] S. Vutukury and J. J. Garcia-Luna-Aceves, "MDVA: A distance-vector multipath routing protocol, " in *Proc. IEEE INFOCOM'2001*, 2001, pp. 557–564.
[7] E. Rosen, A. Viswanathan, and R. Callon, "Multiprotocol label switching architecture," IETF RFC 3031, 2001.
[8] P. Newman, T. Lyon, and G. Minshall, "Flow labeled IP: A connectionless approach to ATM," in *Proc. IEEE INFOCOM'96*, 1996, pp. 1251–1260.
[9] Cisco NetFlow, Available at http://www.cisco.com/warp/public/cc/pd/iosw/ioft/neflct.
[10] Z. Cao, Z. Wang, and E. Zegura, "Performance of hashing-based schemes

for internet load balancing," in *Proc. IEEE INFOCOM'2000*, 2000, pp. 332–341.
[11] J. Chen, "Multipath routing for large-scale networks," Ph.D. Dissertation, Rice University, 1999.
[12] R. Braden, "Requirements for internet hosts – communication layers," IETF RFC 1122, 1989.
[13] M. Allman, "On the generation and use of TCP acknowledgements," *ACM Computer Commun. Review*, vol. 28, no. 5, pp. 4–21, Oct. 1998.
[14] H. Balakrishnan, V. N. Padmanabhan, and R. H. Katz, "The effects of asymmetry on TCP performance," in *Proc. ACM/IEEE MobiCom'97*, Sept. 1997, pp. 77–89.
[15] J. Padhye *et al.*, "Modeling TCP throughput: A simple model and its empirical validation," in *Proc. ACM SIGCOMM'98*, 1998, pp. 303–314.
[16] I. Cidon, R. Rom, and Y. Shavitt, "Analysis of multi-path routing," *IEEE/ACM Trans. Networking*, vol. 7, no. 6, pp. 885–896, Dec. 1999.
[17] 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 Commun. Review*, vol. 22, no. 4, pp. 54–64, Oct. 1992.
[18] N. S. V. Rao and S. G. Batsell, "QoS routing via multiple paths using bandwidth reservation," in *Proc. IEEE INFOCOM'98*, 1998, pp. 11–18.
[19] A. Shaikh, J. Rexford, and K. G. Shin, "Load-sensitive routing of long-lived IP flows," in *Proc. ACM SIGCOMM'99*, 1999, pp. 215–226.
[20] P. Bhaniramka, W. Sun, and R. Jain, "Quality of service using traffic engineering over MPLS: An analysis," in *Proc. IEEE LCN'2000*, 2000, pp. 238–241.
[21] H. Saito, Y. Miyao, and M. Yoshida, "Traffic engineering using multiple multipoint-to-point LSPs," in *Proc. IEEE INFOCOM'2000*, 2000, pp. 894–901.
[22] S. Johnson, "Increasing TCP throughput by using an extended acknowledgement interval," Master's Thesis, Ohio University, June 1995.
[23] Network simulator - ns(version 2), Available at http://www-mash.cs.berkeley.edu/ns.
[24] very high performance Backbone Network Service (vBNS), Available at http://www.vbns.net.

Youngseok Lee received B.S and M.S. degrees in computer engineering from the Seoul National University, Seoul, Korea, in 1995 and 1997, respectively. Currently, he is studying toward the Ph. D degree at the School of Computer Science and Engineering, Seoul National University. He is mainly engaged in research on Internet traffic engineering and network planning.



Ilkyu Park received received B.S and M.S. degrees in computer engineering from the Seoul National University, Seoul, Korea, in 1999 and 2001, respectively. In 2001, he joined as a researcher the Distributed Virtual Reality Research Team, Electronics and Telecommunications Research Institute (ETRI), Taejon, Korea. His research interests include massive networked games, distributed system and high-speed network.



Yanghee Choi received B.S. in electronics engineering from Seoul National University, M.S. in electrical engineering from Korea Advanced Institute of Science, and Doctor of Engineering in Computer Science from Ecole Nationale Superieure des Telecommunications (ENST) in Paris, in 1975, 1977, and 1984 respectively. Before joining the School of Computer Engineering, Seoul National University in 1991, he has been with Electronics and Telecommunications Research Institute (ETRI) during 1977-1991, where he served as director of Data Communication Section, and Protocol Engineering Center. He was research student at Centre National d'Etude des Telecommunications (CNET), Issy-les-Moulineaux, during 1981-1984. He was also Visiting Scientist to IBM T. J. Watson Research Center for the year 1988-1989. He is now leading the Multimedia Communications Laboratory in Seoul National University. He is also director of Computer Network Research Center in Research Institute of Advanced Computer Technology (RIACT). He was editor-in-chief of Korea Information Science Society journals. He was chairman of the Special Interest Group on Information Networking. He has been associate dean of research affairs at Seoul National University. He is now president of Open Systems and Internet Association of Korea. His research interest lies in the field of multimedia systems and high-speed networking.