# High-speed, Short-latency Multipath Ethernet Transport for Interconnections

Nobuyuki Enomoto, Hideyuki Shimonishi, Junichi Higuchi,
Takashi Yoshikawa, and Atsushi Iwata
*System Platforms Research Laboratories, NEC Corporation*
*n-enomoto@db.jp.nec.com*

## Abstract

*In this paper, we propose an Ethernet-based transmission-guaranteed, congestion-controlled network using a simplified multi-path aggregation scheme. Multi-path aggregation increases throughput by multiplying the bandwidth of a single path by the number of paths. There are several aggregation schemes, such as Link Aggregation and Multi-path TCP. However, Link Aggregation is unable to utilize multiple paths to increase throughput because it distributes a single flow only into a single path, The Multi-path TCP scheme requires managing two stages of sequence numbers (SEQ) (i.e., the SEQ assigned to a path and the SEQ assigned to the flow). This complexity requires software-based implementation, which means the method fails to provide high throughput and short latency. We therefore propose a single-stage sequence number scheme with a reorder-buffer-usage-based retransmission-activating algorithm. Hardware implementation of this scheme is very simple. In addition, packet-loss detection is rapid. Our simulation and experimental results showed that our scheme provided a high-throughput, short-latency, and transmission-guaranteed network over conventional, lossy, delay-fluctuated, and best-effort Ethernet.*

## 1. Introduction

The applications of Ethernet are expanding to extremely short-distance communications, such as CPU-CPU and CPU-I/O interconnections in data-center clusters. Compared to common interconnection technologies, such as InfiniBand or PCI-Express switches, Ethernet can provide scalable and cost-effective transport for computer systems that connect a large number of CPUs and I/Os.

Efforts [1] have been made to tunnel PCI-Express data packets into Ethernet. Since PCI-Express is a (packetized) computer-bus technology that provides a channel equivalent to a physical wire, the challenge of such tunneling technology is to achieve very short transmission latency, as well as high reliability, without any packet loss or congestion. We therefore propose an extended MAC layer scheme, called EFL (Ethernet with Flow Label), to provide end-to-end (MAC-to-MAC) packet retransmission and congestion control over an Ethernet path [2]. The scheme introduces a flow-label header into an Ethernet frame to identify its sequence number (SEQ) and time stamp, as well as an explicit acknowledgement (ACK) using a backward ACK packet. Based on these extensions, the scheme provides packet retransmission using duplicate ACKs and retransmission time-out, as well as delay-based congestion control.

The proposed scheme thus provides reliable Ethernet transport over an Ethernet path. However, in many cases, the capacity of a single Ethernet link is not large or efficient enough for tunneling interconnection packets into Ethernet. For example, PCI-Express x16 requires 32-Gbps throughput, while the maximum link capacity standardized for Ethernet so far is 10 Gbps. PCI-Express x2 requires 2-Gbps throughput, but the cost of two 1-Gbps links may be much less than that of one 10-Gbps link. Therefore, aggregation of multiple Ethernet paths could provide an effective means of interconnection transport.

Link Aggregation (Trunking) [3], which employs multiple Ethernet links in parallel, is widely used to distribute multiple MAC-to-MAC flows into multiple paths and increase link capacity. However, it cannot distribute a single MAC-to-MAC flow into multiple paths and cannot be applied to EFL because it uses a Source/Destination MAC address hash to distribute packets over multiple links. Multipath TCP schemes [4] [5] have been proposed to dispatch a TCP flow into multiple paths, so that the flow can utilize more than a single-path capacity. One scheme that has been proposed [5] dispatches packets into multiple paths in a

1550-4794/08 $25.00 © 2008 IEEE
DOI 10.1109/HOTI.2008.13

round-robin manner from a sender, and sorts packets received from the paths using reorder buffers at a receiver. The scheme can also be applied to EFL for aggregation; however, it requires two stages of SEQ (an SEQ assigned to the path and an SEQ assigned to the flow) management and is thus unsuitable for implementation in hardware. As a result, it is almost impossible to provide high throughput and short latency with this scheme.

To address this problem, in this paper we propose an extended Ethernet MAC scheme, which we call EFL-MP (Ethernet with Flow Label for Multi-Path). EFL-MP distributes the packets of a particular flow into multiple paths with a single-stage SEQ assigned to each flow. EFL-MP is as simple as Link Aggregation, but still provides the same high throughput and short latency as Multipath TCP in a local area environment. In addition, our algorithm improves throughput and latency in the event that packet loss occurs.

Below, we discuss multipath Ethernet transport schemes and describe our proposed simple, high-performance multipath Ethernet transport mechanism. We also present the results of a simulation of the scheme.

## 2. Multipath transport for data-center area communications

In this section, we evaluate three Ethernet aggregate schemes for CPU-CPU and CPU-I/O interconnections.

### 2.1. Multipath Ethernet schemes

Link Aggregation (Trunking) [3], which employs multiple Ethernet networks in parallel, is widely used to increase link capacity. In combination with congestion control and packet retransmission, Link Aggregation is applied to interconnections (Fig. 1). However, due to the fact that it uses a source/destination MAC address hash to dispatch packets over multiple links, Link Aggregation is unable to transport wide-bandwidth flows at a speed beyond the limit of any one link.

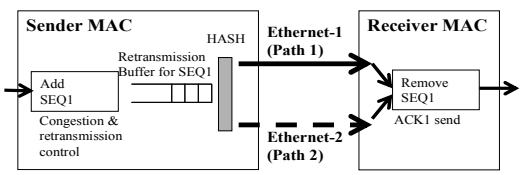


**Fig. 1. Link aggregation with congestion control, packet retransmission, and hash**

Multipath TCP [5] (Fig. 2) can transmit over-10-Gbps flows by dispatching packets over multiple paths using a round-robin technique and reorder buffers (buffers for data sorting at the receiver). However, when there are N paths between the endpoints, the scheme needs a pair of SEQs (one add SEQ and one remove SEQ) assigned to the aggregated path (SEQ1 in Fig. 2), N pairs of SEQs assigned to each path (SEQ2 in Fig. 2), N+1 retransmission buffers, and N reorder buffers. This two-stage SEQ management is not suitable for implementing in hardware because of the complexity caused by the management of the retransmission buffers. Thus, it fails to provide high throughput and short latency.

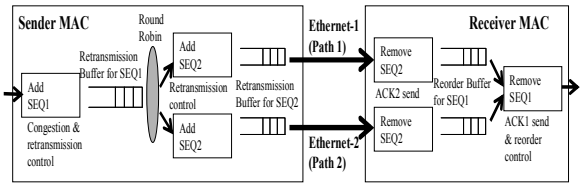


**Fig. 2. Multipath TCP**

To address this problem, we propose an extension of EFL for multi-path load-balancing (Fig. 3). EFL-MP employs only single-stage (aggregated-path level) SEQ management, and transmits over-10-Gbps flows using a round-robin technique and reorder buffers. The scheme needs an SEQ add/remove pair assigned only to the aggregated path (SEQ1 in Fig. 3), a single retransmission buffer, and N reorder buffers.

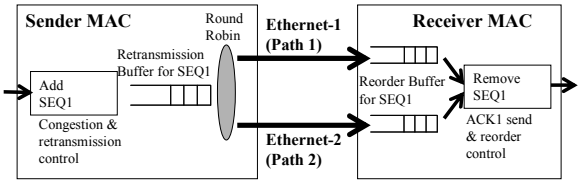


**Fig. 3. EFL-MP**

**Table 1. Complexity of various schemes**

| | Number of SEQ add/remove pairs | Number of retransmission buffers | Number of reorder buffers |
|---|---|---|---|
| Link Aggregation [3] with Congestion Control | O(1) | O(1) | O(0) |
| Multipath TCP [5] | O(N) | O(N) | O(N) |
| EFL-MP (proposed) | O(1) | O(1) | O(N) |

## 2.2. Model of multipath data-center environment

For the sake of building a disjoint network simply, we assume that tunneling of interconnection packets (EFL) is generally used inside local area environments such as data centers, and the networks are not shared with other applications such as FTP or WWW but are intended only for EFL. We also assume that the network topology is symmetric, i.e., each endpoint is connected to both Ethernet-1 and Ethernet-2, and both Ethernets are disjointed, as shown in Fig. 4.

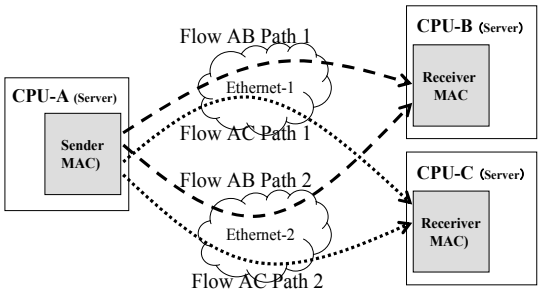


**Fig. 4. Model of multipath data-center environment**

## 2.3. Best multipath Ethernet scheme for a data-center

In the environment mentioned in 2.2 above, the delay and the throughput between two paths are almost the same, and there is little delay and little bandwidth gap between the paths. Therefore, in most cases, there is no advantage over Multipath TCP, which adapts the transmission speed to each path. The performance of EFL-MP, which has a single-stage SEQ, and the performance of Multipath TCP, which has a double-stage SEQ, are approximately the same in this situation. However, Multipath TCP has a serious disadvantage because the scheme is not suitable for hardware implementation.

To summarize the points made in 2.1 and 2.2 above, the proposed scheme, EFL-MP, is as simple as Link Aggregation, but still provides the same high throughput and short latency as Multipath TCP in the assumed environment.

As already mentioned, Multipath TCP has a two-stage (path-level and aggregated path-level) SEQ. Thus, in each path, retransmission is immediately activated (i.e., a duplicate ACK will be sent) at the Receiver MAC if there are discontinuous path levels of SEQs, and fast retransmission [6] will be invoked at the Sender MAC. In contrast, EFL-MP, which has only a single-stage (aggregated path-level) SEQ, cannot send a duplicate ACK because it does not have a path-level SEQ and cannot distinguish whether the reason for a discontinuous aggregated path-level SEQ is due to a reordering problem (merely a delay in packet arrival) or to packet loss. As a result, the EFL-MP receiver (Receiver MAC) cannot activate retransmission, and this forces the retransmission to be activated by the retransmission timer in the EFL-MP sender (Sender MAC). This causes a delay in activating retransmission, resulting in a decrease in throughput and longer latency.

This problem, however, can be resolved by using the following retransmission-activating algorithm. To detect packet losses and activate retransmission on the receiver side while using a single-stage SEQ, we propose a novel reorder-buffer-usage-based retransmission-activating algorithm that sends a duplicate ACK when it detects one or more queued packets in all of the reorder buffers. By using this algorithm, receivers are able to detect packet losses and send duplicate ACKs, while employing a single SEQ scheme.

# 3. Simple, high-performance multipath Ethernet transport mechanisms (EFL-MP)

## 3.1. Single-stage sequence number scheme

EFL-MP extends MAC with packet retransmission, congestion control, and load-distribution (multipath forwarding) mechanisms. These mechanisms are controlled by a single-stage SEQ add/remove mechanism that is assigned to each flow (aggregated path level). The packet retransmission mechanism utilizes it to check whether the packet has arrived properly or not. The congestion-control mechanism utilizes it to calculate the round-trip time and transmission rate, and the load-distribution mechanism utilizes it to reorder packets.

Although the above-mentioned single-stage sequence number scheme cannot detect packet losses and activate retransmission, our reorder-buffer-usage-based retransmission-activating algorithm improves throughput and latency when packet loss occurs.

## 3.2. Packet retransmission mechanism

To provide reliability, we used the Go-back-N ARQ (Automatic Repeat reQuest) mechanism for packet retransmission. To activate retransmission due to packet losses, we employed both a fast retransmission mechanism triggered by duplicate ACKs and retransmission timeout.

### 3.3. Congestion-control mechanism

The congestion-control algorithm employs a delay-based congestion-control algorithm along with bandwidth probing because delay-based algorithms have the advantage of maintaining shorter queuing delays.

### 3.4. Load-distribution mechanism

To maximize the throughput of each path and transfer flows with a throughput exceeding the bandwidth of each Ethernet link, EFL-MP uses a round-robin technique to distribute packets, as has already been proposed [5]. For example, if there are four paths of 10-Gbps Ethernets and a pair of EFL-MP MACs connected by the four paths, the maximum throughput will be 40 Gbps because the flow is distributed into the four paths.

When using a round-robin technique, packet reorders may occur if there is a delay gap or bandwidth gap between the paths. To guarantee the sequence of the flow, EFL-MP employs a FIFO-based reorder buffer in each path in the Receiver MAC to reorder packets.

### 3.5. Retransmission-activating algorithm for maximizing throughput and minimizing latency

As discussed in Section 2, since EFL-MP has only a single-stage SEQ, it is not able to detect packet loss and activate retransmission because it does not have a path-level SEQ and thus cannot distinguish the reason for a discontinuous SEQ, i.e., whether it is due to a reordering problem (merely a delayed packet arrival) or packet loss. Therefore, the EFL-MP receiver cannot activate retransmission and consequently the throughput decreases and the latency becomes longer when packet losses do occur. Here, we demonstrate the use of a retransmission-activating algorithm to solve this problem.

A reorder buffer is installed to guarantee the sequence of flow despite the delay and jitter gap between the paths. Also, according to the specification [3], there is no reorder in Ethernet. Therefore, in Fig. 5, if the delay for Path-1 is shorter than that for Path-2, packets sent into the former will arrive earlier than packets sent into the latter. Packets arriving from Path-1 will be queued at the reorder buffer and held until the packets from Path-2 arrive. If no packets are dropped, the packets arriving from Path-2 will never be queued at the reorder buffer.

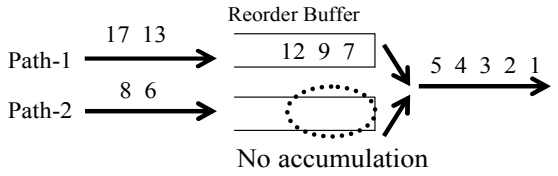


**Fig. 5. Packet accumulation with delay skew**

If packet loss occurs, as shown in Fig. 6, packets arriving from both Path-1 and Path-2 will be queued at the reorder buffer. Because there is no requested packet at the head of the queue, and no packet is picked from the queue, the arriving packets are continuously queued at the reorder buffer.

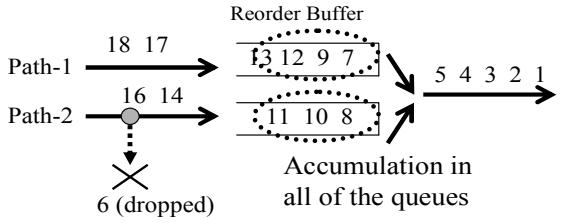


**Fig. 6. Packet accumulation with packet loss**

```
RecvPacketMP() {
  proceed = NO;
  while (CheckSeq(QueueHead, ether path in PATHS) == nextSEQ) {
    PickPacket(QueueHead, path);
    nextSEQ++;
    proceed = YES;
  }
  if (proceed) SendAckDelayed(nextSEQ-1);
  else if (QueueLength>0 for all path) SendAckImmidiate(nextSEQ-1);
}
```

**Fig. 7. Proposed retransmission-activating algorithm**

```
RecvPacket() {
  proceed = NO;
  while (CheckSeq(QueueHead) == nextSEQ) {
    PickPacket(QueueHead);
    nextSEQ++;
    proceed = YES;
  }
  if (proceed) SendAckDelayed(nextSEQ-1);
  else SendDupImmidiate(nextSEQ-1);
}
```

**Fig. 8. TCP retransmission-activating algorithm**

Our proposed algorithm (Fig. 7) makes use of the behavior described in Fig. 6 and activates retransmission due to packet loss. In our algorithm, when a packet arrives at the Receiver MAC, it is stored on the reorder buffer. If the SEQ of the packet at the head of the reorder buffer is equal to the requested SEQ, the packet at the head of the buffer is picked and a delayed ACK is sent (this delayed ACK is regarded as an ordinary acknowledgement at the Sender MAC). Otherwise, when there is more than one packet in each

queue or any of the queue becomes full, an ACK is sent immediately because the condition is regarded as a packet loss having occurred. This ACK is regarded as a retransmission request (i.e., a duplicate ACK) when it arrives at the Sender MAC and invokes retransmission. By using this algorithm, receivers are able to activate retransmission when employing a single-stage SEQ scheme.

The proposed algorithm is regarded as an N-path extension of the TCP ACK returning algorithm shown in Fig. 8. If the number of paths is 1, the proposed algorithm (Fig. 7) will be equivalent to the TCP ACK returning algorithm (Fig. 8).

## 4. Simulation of performance

In this section, we present NS2 [7] simulation results showing that, despite its simple mechanism, EFL-MP using the proposed retransmission-activating algorithm provides the same high throughput and short latency as Multipath TCP, and that its throughput and latency satisfy interconnection requirements (In the NS2 simulation, processing performance is infinite, thus the Multipath TCP nearly reaches the ideal performance despite its complexity). In addition, we demonstrate that the proposed retransmission-activating algorithm activates retransmission faster than the conventional timer-based retransmission-activating algorithm.

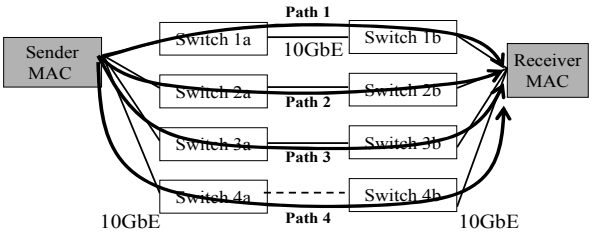


**Fig. 9. Evaluation model**

Figure 9 shows a model of the network used in the evaluation (Because of limited space, more complex simulations will be left as a future subject of research). To evaluate whether the schemes can or cannot transport PCI-Express x16 traffic (32 Gbps), there were four networks (paths 1 - 4) between the sender and receiver. All links had a bandwidth of 10 Gbps, and the round-trip propagation delay between the sender and receiver was 30 us. (The round-trip delay of a link is usually 10 us; because there are three links on a path, it is set at 30 us. Delays of switches are small enough to neglect.) Switches employ tail-drop buffers with 1-MB capacity. Assuming that the size of a PCI-Express packet is 128 bytes, together with MAC headers, the packet size is 192 bytes in total. In 4.2 below, we varied the delay between Switch 4a and Switch 4b between 1 and 10. Under this condition, a maximum of 32 packets will be queued in the reorder buffer. To add some margin to this, we set the size of the reorder buffer at 40 packets.

To ascertain whether EFL-MP and the proposed retransmission-activating algorithm resulted in an improved performance or not, we compared the following five multipath Ethernet transport schemes together with ideal values.

(a) LAH (Link Aggregation with Congestion Control, Retransmission and Hash): To apply it to interconnection, Link Aggregation was combined with congestion control, packet retransmission, and a hash. This model uses a hash and thus cannot transfer any wide-bandwidth flow with a speed beyond that of a single cable or port (see Fig. 1).
(b) LARR (Link Aggregation with Congestion Control, Retransmission and Round Robin): Usually, Link Aggregation adopts a hash to distribute packets, but in this case, we used a round-robin technique to transport a wide-bandwidth flow.
(c) MTCP (Multipath TCP): Multipath TCP (Fig. 2) can transmit over-10-Gbps flows by dispatching packets over multiple paths using a round-robin technique and reorder buffers. However, since it is not suitable for hardware implementation, it fails to provide high throughput and short latency. (In simulations, processing power is infinite; thus, this disadvantage does not appear in the simulation results.)
(d) EFLMP-conventional: EFL-MP (Fig. 3) using a "conventional" retransmission-timer-based retransmission-activating algorithm.
(e) EFLMP-proposed: EFL-MP (Fig. 3) using the "proposed" reorder-buffer-based retransmission-activating algorithm
(f) IDEAL: Ideal throughput/latency, which is the aggregate throughput of links 1-4, or the shortest latency of links 1-4, and is not affected by any overhead.

### 4.1. Robustness against link bandwidth differentiation

Since EFL-MP is used in a symmetric environment, the bandwidth and delay gap between the paths is basically very small. However, because of the processing time difference between the Ethernet switches along the paths or other factors, small fluctuations in bandwidth and delay may occur. Here, we show how much fluctuation is acceptable for multipath Ethernet transport schemes.

Figure 10 compares the application-level throughput at the Receiver MAC when the link bandwidth between Switch 4a and Switch 4b (Link 4: dashed-line link in Fig. 9) was varied between 1 and 10 Gbps. All of the other link bandwidths were fixed at 10 Gbps. Each point of the graph is the average throughput during a 50-msec data transfer.

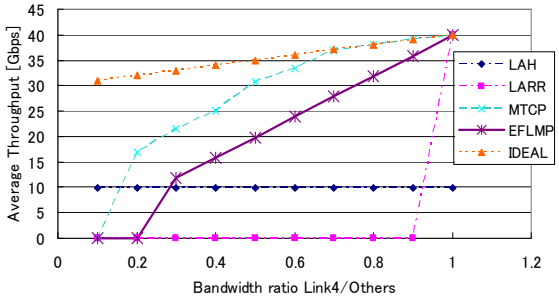


**Fig. 10. Robustness for link bandwidth differentiation**

The results showed that the EFL-MP (EFLMP) and the Multipath TCP (MTCP) utilized the aggregate bandwidth of links 1-4. MTCP utilized the bandwidth especially well because it can control the transmission rate at a fine granularity with two stages of SEQ, and calculates the transmission rate per path as well as per flow.

Our EFLMP utilized the aggregate bandwidth, but less efficiently than the MTCP. To simplify the scheme, the EFLMP employed the single-stage SEQ scheme and the transmission rate was calculated in rough granularity (per flow). Therefore, the throughput of each path was limited to the bandwidth of the narrowest path (bottlenecked link). Thus, the throughput between the Sender MAC and Receiver MAC was limited to roughly four times the bandwidth of the bottlenecked link between Switch 4a and Switch 4b in Fig. 9. In this evaluation, there was no packet loss in all links. Thus, the retransmission-activating algorithm proposed in Section 3 did not affect the results, and the results of EFLMP-proposed and EFLMP-conventional were the same.

The LAH throughput was limited to 10 Gbps, the bandwidth of each link. The LAH uses a Source/Destination MAC address hash to spread packets, and the hash limit is that only one path can be used to transfer a single flow. Thus, the LAH is not affected by the bandwidth gap, but is unable to transport a wide-bandwidth flow with a speed beyond the limit of any one cable or port.

The LARR cannot be used when there is a bandwidth gap between the paths. Since the LARR does not have a reorder buffer in the Receiver MAC, it cannot function under bandwidth differentiation.

Even if the bandwidth ratio is 0.8, the average throughput of the EFLMP is still about 32.0 Gbps. In a symmetric environment, the bandwidth gap is small and this throughput is thus sufficient to transport PCI-Express x16 signals. Therefore, the proposed scheme satisfies interconnection requirements.

## 4.2. Robustness against link delay differentiation

Here, we show how much delay fluctuation is acceptable for multipath Ethernet transport schemes.

Figure 11 compares the throughput at the Receiver MAC when the link delay between Switch 4a and Switch 4b (Link 4: dashed-path link in Fig. 9) was varied between 1 and 10 usec. All of the other link delays were fixed at 5 usec (i.e., RTT (round-trip time) of Path 4 was varied between 22 and 40 usec and RTTs of all the other paths were fixed at 30 usec). Each point of the graph is the average throughput during a 50-msec data transfer.

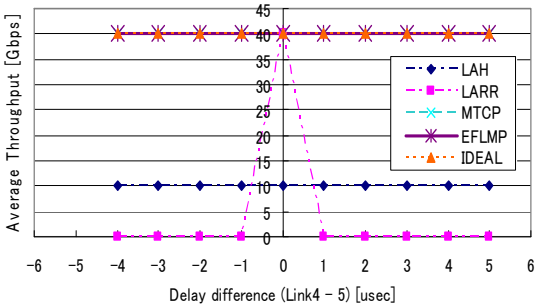


**Fig. 11. Robustness for link-delay differentiation**

The results showed that both the EFLMP and MTCP utilized the entire aggregate bandwidth (plots of the MTCP and EFLMP are hidden under that of Ideal in Fig. 11). They both calculate the RTT at a fine granularity (per path) to detect congestion. Thus, they can function under delay differentiation. In this evaluation, there was no loss in any links. The retransmission-activating algorithm proposed in Section 3 therefore did not affect the results, and the results of EFLMP-proposed and EFLMP-conventional were the same.

The LAH uses only one path to transfer a single flow. Thus, the LAH throughput was limited to 10 Gbps, and was not affected by the delay gap.

The LARR cannot be used when there is a delay gap between the paths. Since LARR does not have a

reorder buffer in the Receiver MAC, it cannot function under delay differentiation.

Throughout the delay-difference range, EFLMP maintains 40 Gbps. This throughput is sufficient to transport PCI-Express x16 signals. Therefore, the proposed scheme satisfies interconnection requirements.

### 4.3. Robustness against packet loss (throughput)

Since the EFL-MP is used in a local area environment, basically packet loss rate is estimated to be very low. However, a small amount of packet loss may occur. From the throughput viewpoint, this section evaluates what level of packet loss is acceptable for multipath Ethernet transport schemes.

Figure 12 compares robustness against packet loss. The packet-loss rate of the link between Switch 4a and Switch 4b (dashed line in Fig. 9) was set between 0.0001% and 1%.

The results show that LARR achieved the highest throughput. However, since LARR is not effective when there is a bandwidth/delay gap between the paths, it cannot be used for interconnections.

The MTCP achieved the second-highest throughput, but it was only slightly better than the EFLMP-proposed throughput.

The EFLMP-proposed throughput was much higher than the EFLMP-conventional throughput. With the proposed algorithm, the throughput is higher than that of the conventional algorithm for all packet loss rates. It is especially notable that the throughput of our proposed algorithm was eight times higher than that of the conventional algorithm at a loss rate of 0.1%. This clearly shows that the proposed algorithm enhances throughput.

Even if the loss rate is 0.01% (although in a local area environment, it is estimated to be much lower than 0.01%), the average throughput of EFLMP-proposed is still about 24.0 Gbps. This throughput may degrade the system performance, but it enables the system to continue running. Therefore, the proposed scheme and algorithm satisfy interconnection requirements.

### 4.4. Robustness against packet loss (latency)

From the latency viewpoint, this section evaluates how much loss is acceptable for multipath Ethernet transport schemes.

Figure 13 compares the end-to-end latency between the Sender MAC and Receiver MAC against packet loss. The packet loss rate of the link between Switch 4a and Switch 4b (dashed line in Fig. 9) was set between 0.0001% and 1%.

The results show that LARR achieved the shortest latency. However, LARR does not function when there is a bandwidth/delay gap between the paths and thus cannot be used for interconnections.

MTCP achieved the second shortest latency, but it was only slightly better than that of our EFLMP-proposed. In this environment, however, MTCP has a serious disadvantage in that the scheme is not suitable for hardware implementation.

The latency of EFLMP-proposed was much shorter than that of EFLMP-conventional. Compared to using the conventional algorithm, use of the proposed algorithm produced a shorter latency under all loss rates. It is especially notable that the latency of our proposed algorithm was 100 or more times shorter than that of the conventional algorithm at a loss rate of 0.01% or more. We thus demonstrated that the proposed algorithm shortens the latency period.

Even if the loss rate is 1% (although in a local area environment, the loss rate is estimated to be much lower than 1%), the average latency of our proposed EFLMP is still 200 usec. However, the PCI-Express specification [8] requires that the latency must be 10 msec or less. Thus, the proposed scheme and algorithm satisfy interconnection requirements.

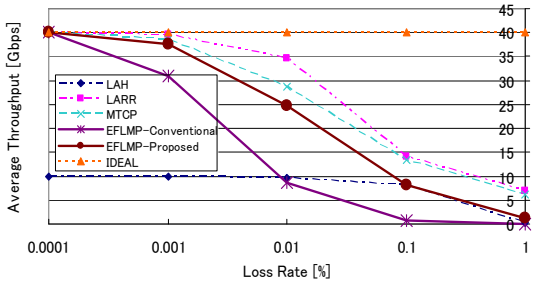


**Fig. 12. Robustness for loss rate differentiation (throughput)**

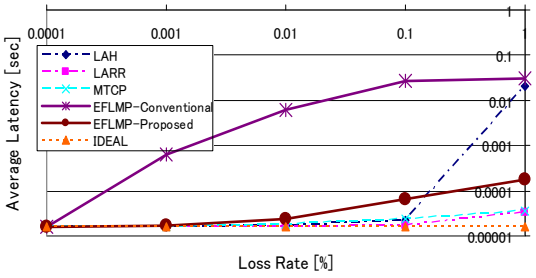


**Fig. 13. Robustness for loss rate differentiation (latency)**

The above evaluations, 4.1 – 4.4, demonstrated that, in the assumed environment described in Section 2.2), EFLMP with the proposed retransmission-activating algorithm provided the same high throughput and short latency as the Multipath TCP, despite the EFLMP's simple mechanism.

## 5. Experimental evaluation

In this section, we present experimental results showing that the EFL-MP and the proposed retransmission-activating algorithm can be implemented in hardware and that they work properly in a real environment. In addition, some EFL-MP Senders/Receivers share the same network in the real environment (as shown in Fig. 4). Therefore, congestion control must lower the transmission rate quickly when other traffic starts to use the shared bandwidth, and must increase the transmission rate when other traffic stops using the shared bandwidth. This section evaluates how well the EFL-MP can adjust its transmission rate to the residual bandwidth.

### 5.1. Hardware implementation

We implemented both the Sender and Receiver MAC into an FPGA evaluation board (Fig. 14). The features of the evaluation board are listed in Table 2 below.

**Fig. 14. FPGA evaluation board**

The evaluation board has one copper gigabit Ethernet port and two optical gigabit Ethernet ports. We utilized the copper one for the input/output of Ethernet frames (P1a/P1b in Fig. 15) and the optical ones for Ethernet frames with a flow label (P2a/P3a/P2b/P3b in Fig. 15). When an Ethernet frame (PCI-express over Ether frame (A) in Fig. 15) arrives at P1a in the Sender MAC, the Sender MAC adds a flow label (i.e. EFL Header in Table 3) to the frame, and sends it (Frame (B) in Fig. 15) from P2a or P3a. When the frame arrives at P2b or P3b in the Receiver MAC, the Receiver MAC removes the flow label, and sends the Ethernet frame to P1b.

**Table 2. Features of the evaluation board**

| | | |
|---|---|---|
| Type of board | ALTERA PCIE Development Board StratixII GX Edition DK-PCIE-2SGX90N | |
| Interface | RJ45 Interface (copper gigabit Ethernet) | 1 ch |
| | SFP Interface (optical gigabit Ethernet) | 2 ch |
| | PCI-Express | Not in use |
| Clock | 156.25 MHz , 128 bit (Support 20Gbps) | For core block |
| | 125 MHz , 8bit | For 1G MAC |
| FPGA | ALTERA StratixII GX EP2SGX130GF1508C5N | 65% logic utilization<br>64% memory utilization |

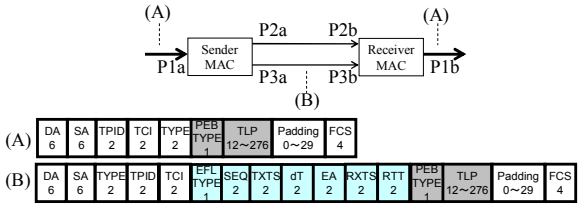


**Fig. 15. Port assignment**

**Table 3. Header fields**

| | | |
|---|---|---|
| Ethernet header | DA | Destination Address |
| | SA | Source Address |
| | TYPE | Ether Type |
| | TPID | Tag Protocol Identifier |
| | TCI | User Priority, CFI, VLAN ID |
| PCIe over Ether header | PEB TYPE | Sub Type for PCIe over Ether |
| | TLP | PCI-express TLP (Transaction Layer Packet) |
| EFL header | EFL TYPE | Sub Type for Ethernet with Flow Label |
| | SEQ | Sequence Number |
| | TXTS | Timestamp of Data Transmit |
| | dT | Receiver Processing Time |
| | EA | Entry Address |
| | RXTS | Timestamp of ACK Receive |
| | RTT | Latency Calculated from RXTS, dT, and TXTS |
| Reserved | Padding | |
| | FCS | Bits for CRC Check |

Figure 16 shows a block diagram of the FPGA. We implemented both the Sender MAC and Receiver MAC in one FPGA. Our functions consumed 86000 Logic Elements and 4.3Mbits RAM on the FPGA. If we implemented Multipath TCP, which is more complicated than EFL-MP, it would probably exceed

the capacity of the FPGA. Thus our functions are simple enough for hardware implementation.

The Sender MAC is the sender part of EFL-MP and includes Retransmission Control, Congestion Control, Round Robin, Control Frame Generation, Frame Analyze, Failure Detection, and MACs. The Receiver MAC is the receiver part of EFL-MP and includes Reorder Control, Frame Analyze, Switch, and MACs. The Common Function manages flows and registers for the Sender/Receiver MACs. The details of these functions are explained in Section 3.

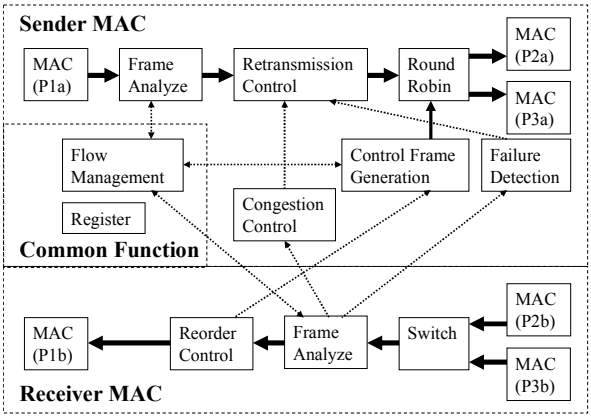


**Fig. 16. Block diagram of FPGA**

## 5.2. Experimental setup

Figure 17 shows the experimental setup. There were two networks (Paths 1-2) between the sender and receiver. All links had a bandwidth of 1 Gbps. Assuming that the maximum size of a PCI-Express TLP (Transaction Layer Packet) was 276 bytes, together with Ethernet headers and a flow label (EFL header in Table 3), the packet size was 312 bytes in total.

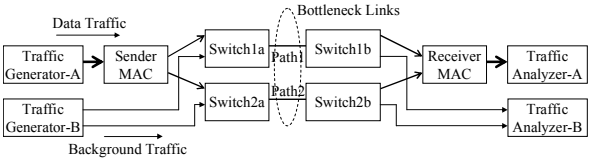


**Fig. 17. Experimental setup**

There were two traffic generators and two traffic analyzers. Traffic generator-A sent continuous data traffic, which was analyzed by Traffic analyzer-A. The transmission rate of the data traffic was controlled by the congestion control in the Sender MAC. Traffic generator-B sent continuous background traffic, which was analyzed by Traffic analyzer-B. The background traffic did not have congestion control; it used the bandwidth prior to the traffic sent by Traffic generator-A. Therefore, the residual bandwidth of the bottleneck link was controlled by the background traffic.

## 5.3. Robustness against residual bandwidth

Here we varied the residual bandwidth of the bottleneck links by adjusting the transmission rate of the background traffic, and evaluated the transmission rate of the data traffic. Figure 18 shows the total transmission rate, and average latency, for Paths 1-2, against the total residual bandwidth of Paths 1-2.

In Fig. 18, the total transmission rate varies in proportion to the total residual bandwidth. This indicates that the congestion control of EFL-MP utilized the residual bandwidth.

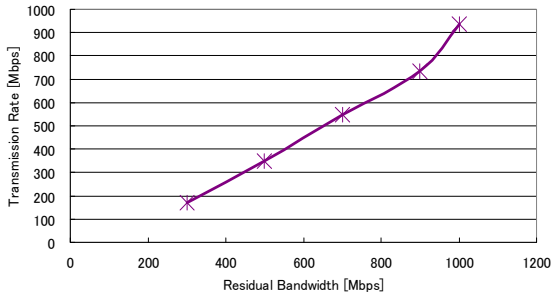


**Fig. 18. Transmission rate against residual bandwidth**

## 5.4. Stability against bandwidth change

Here we varied the residual bandwidth of the bottleneck links by adjusting the transmission rate of the background traffic, and evaluated the transitional behavior of the data-traffic transmission rate. Figure 19 shows the transmission rate for Path 1 (note that this is not the total transmission rate for Paths 1-2.). We sent background traffic consecutively at 1-sec intervals, and each time we increased the background traffic by adding 100 Mbps to the previous rate.

As Fig. 19 shows, the transmission rate changed quickly when the background traffic increased. When the background traffic stopped, the EFL-MP quickly utilized the residual bandwidth. The time taken to stabilize the transmission rate became longer in proportion to the transmission rate of the background traffic. This is because we utilized the background traffic to create bottlenecks; thus, the queuing delay for the switches shown in Fig. 17 changed more dynamically when the background traffic increased. This increased the time needed for congestion control to achieve stability.

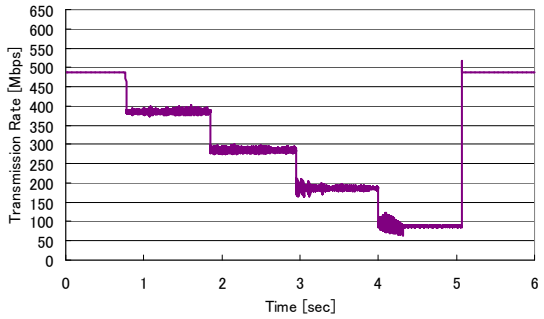


**Fig. 19. Stability against bandwidth change**

## 6. Conclusion

In this paper, we proposed a simple scheme for multi-path aggregation of Ethernet paths. To maximize data-transfer throughput and minimize latency, we proposed a new single-stage sequence number scheme "EFL-MP" with a reorder-buffer-usage-based retransmission-activating algorithm. Using the single-stage sequence number scheme mitigates the hardware implementation complexity caused by the management of retransmission buffers, which increases in proportion to the number of paths. The reorder-buffer-usage-based retransmission-activating algorithm activates retransmission earlier than the retransmission-timer-based retransmission-activating algorithm and thus achieves high throughput and short latency. Simulation results showed that, despite its simple mechanism, the EFL-MP scheme provided the same high throughput and short latency as the Multipath TCP, and that its throughput and latency satisfy interconnection requirements. The results also showed that the proposed retransmission-activating algorithm activated retransmission faster than a conventional timer-based retransmission activating algorithm. Our algorithm achieved roughly eight times higher data-transfer throughput and 100 times shorter data-transfer delay compared to the conventional retransmission-activating algorithm. Experimental results showed that the EFL-MP and the proposed retransmission-activating algorithm can be implemented in hardware and that they work properly in a real environment. The EFL-MP can quickly adjust its transmission rate to the residual bandwidth, and the rate will become stable in a few milliseconds. This behavior satisfies interconnection requirements.

## 7. Acknowledgement

We are indebted to Mr. T. Kosawa for assistance in the collection of experimental data.

## 8. References

[1] J. Suzuki et al., "ExpressEther - Ethernet-Based Virtualization Technology for Reconfigurable Hardware Platform," HOT Interconnects 14, 2006.

[2] H. Shimonishi et al., "A Congestion Control Algorithm for Data Center Area Communications," to be presented at IEEE CQR Workshop, Apr. 2008.

[3] IEEE Standard 802.3AD-2000, IEEE, 2000.

[4] M. Zhang, A. Krishnamurthy, L. Peterson, and R. Wang, "A Transport Layer Approach for Improving End-to-End Performance and Robustness Using Redundant Paths," USENIX 2004, June 2004.

[5] K. Rojviboonchai and H. Aida, "An Evaluation of Multipath Transmission Control Protocol (M/TCP) with Robust Acknowledgement Schemes," Internet Conference 2002, Oct. 2002.

[6] W. Stevens et al., "TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms," RFC 2001, IETF, 1997.

[7] The network simulator ns-2, http://www.isi.edu/nsnam/ns/

[8] PCI Express Base Specification Rev. 2.0, PCI-SIG, Dec. 2006.