IEICE TRANS. INF. & SYST., VOL.E89–D, NO.12 DECEMBER 2006

PAPER Special Section on New Technologies and their Applications of the Internet IV

AMS: An Adaptive TCP Bandwidth Aggregation Mechanism for Multi-homed Mobile Hosts

Shunsuke SAITO[†a)], **Yasuyuki TANAKA**[††b)], **Mitsunobu KUNISHI**[†c)], **Yoshifumi NISHIDA**[†††d)], ***Nonmembers***, ***and*** **Fumio TERAOKA**[†e)], ***Member***

SUMMARY Recently, the number of multi-homed hosts is getting large, which are equipped with multiple network interfaces to support multiple IP addresses. Although there are several proposals that aim at bandwidth aggregation for multi-homed hosts, few of them support mobility. This paper proposes a new framework called AMS: Aggregate-bandwidth Multi-homing Support. AMS provides functions of not only bandwidth aggregation but also mobility by responding to the changes of the number of connections during communication without the support of underlying infrastructure. To achieve efficient data transmission, AMS introduces a function called address pairs selection to select an optimal combination of addresses of the peer nodes. We implemented AMS in the kernel of NetBSD and evaluated it in our test network, in which dummynet was used to control bandwidth and delay. The measured results showed that AMS achieved ideal bandwidth aggregation in three TCP connections by selecting optimal address pairs.

key words: *TCP, multi-home, bandwidth aggregation, address pairs selection*

1. Introduction

Mobile nodes with multiple network interfaces are getting popular, for example, some note PCs equip Ethernet and Wireless LAN devices. These nodes can have multiple addresses and connect to various networks. Such configuration nodes are called *multi-homed mobile hosts*. Multi-homed hosts are not so common in the current Internet, however, they would be popular by development and spread of IPv6 in the near future. Multi-homed mobile hosts can aggregate network bandwidth on multiple paths, achieve traffic load balancing, and improve robustness against network failure by using one or more access lines simultaneously. The focus of our research is methods of bandwidth aggregation for TCP (Transmission Control Protocol) which is a connection-oriented end-to-end transport layer protocol. TCP provides congestion control, flow control and reliability. Since most Internet traffic (e.g., Web browsing and sending/receiving e-mail) is currently using TCP, many researchers have tackled optimization of TCP transmission.

Several mechanisms for TCP to aggregate bandwidth on multiple network paths have been proposed in recent years. There are a number of requirements for TCP bandwidth aggregation (e.g., simultaneous use of multiple TCP flows, retransmission, congestion control, flow control, adaptive data assignment for a variety of network paths, IP address changes, etc.). Existing proposals tackle important tasks such as retransmission for multiple flows and adaptive data assignment for a variety of network paths. However,they have some problems in common. First, these protocols cannot deal with increase and decrease of the number of connections during communication. Second, they do not consider methods of address pairs selection. Third, only a few of them evaluated their proposed methods based on implementation on actual operating systems.

The focus of our research is to design an end-to-end transport protocol for TCP bandwidth aggregation on multi-homed mobile hosts, to tackle the problems that the existing proposals have. In this paper, we propose an end-to-end transport layer protocol called *AMS* (Aggregate-bandwidth Multi-homing Support), which provides functions of not only bandwidth aggregation but also dynamic response to changes of the number of connections during communication and selection of address pairs used to communicate between hosts for efficient transmission. AMS have been implemented on TCP of a real operating system and evaluated in our test network.

The rest of the paper is organized as follows: Section 2 describes related work. Sections 3 and 4 describe the design and the implementation of AMS, respectively. Section 5 demonstrates the experimental results in our test network. Section 6 discusses the methods of address pairs selection in detail. Finally, Sect. 7 concludes.

2. Background and Related Work

Bandwidth aggregation schemes for multi-homed mobile hosts should be able to properly handle changes of the number of active interfaces and addresses during communication because the active interfaces and the number of active interfaces would be changed by location. This section describes several existing studies for bandwidth aggregation, and shows that the transport layer data striping protocol with

Manuscript received March 10, 2006.
Manuscript revised June 23, 2006.

[†]The authors are with Keio University, Yokohama-shi, 223–8522 Japan.
[††]The author is with Communication Plateform Laboratory, R&D Center, Toshiba Corporation, Kawasaki-shi, 212–8582 Japan.
[†††]The author is with Sony Computer Science Laboratories, Inc., Tokyo, 141–0022 Japan.
a) E-mail: shun@tera.ics.keio.ac.jp
b) E-mail: yatch@isl.rdc.toshiba.co.jp
c) E-mail: kunishi@tokoro-lab.org
d) E-mail: nishida@csl.sony.co.jp
e) E-mail: tera@ics.keio.ac.jp
DOI: 10.1093/ietisy/e89–d.12.2838

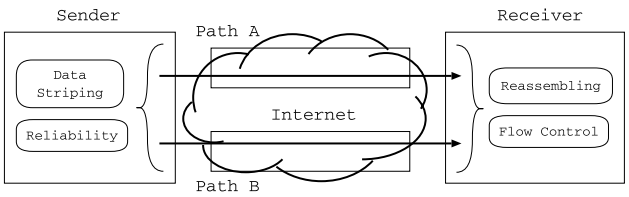


Fig. 1 Bandwidth aggregation.

the dynamic address configuration function is the most suitable system to support the bandwidth aggregation function on multi-homed mobile hosts. Finally, we introduce several transport layer schemes for TCP bandwidth aggregation and mention their problems.

2.1 Data Striping Schemes

Data striping schemes for bandwidth aggregation using multiple network interfaces in parallel have been studied in several layers (the link layer, the network layer, the transport layer, and the application layer). Figure 1 shows bandwidth aggregation on multi-homed hosts with two network paths and describes some fundamental functions to deal with multiple data streams. If Paths A and B have available bandwidth of 1 Mbps and 3 Mbps, respectively in Fig. 1, hosts can potentially transmit data with rate of 4 Mbps.

The link layer data striping approaches [4], [5] transparently supply the function to aggregate the bandwidth across multiple channels to upper layers of the protocol stack. However, they cannot properly handle the various network characteristics in multi-home environment, because multi-homed hosts potentially connect to different networks that have different bandwidth, propagation delay and quality of links. In addition, when hosts use the transport protocols that operate congestion control, they suffer from throughput degradation because all the packet losses in the multiple subflows are processed with a single congestion window and the transport protocols misinterpret that the network path is in a heavy congestion state.

The network layer is the one upper layer of the link layer in the protocol stack, and conceals a wide variety of link layer technologies. The network layer data striping approaches have been described in [6]. In the network layer approach, the process of data striping is prior to routing each packets. Striping data is encapsulated in an IP datagram, and delivered across different paths. In the receiver side, the received data from multiple paths is reconstructed to a single data stream after the decapsulation of the packets. Encapsulation of packets leads packet header overhead and might cause fragmentation of packets. Furthermore, in the case that a transport layer protocol having congestion control is used, deterioration of TCP performance occurs because congestion control is not executed on individual network path independently.

Recently, SCTP (Stream Control Transmission Protocol) [1], a new transport layer protocol, has been proposed. SCTP also provides the functions of congestion control, flow control and reliability to applications as the same as TCP. The main features of SCTP are to support multi-streaming and multi-homing. Multi-streaming is the function that the data stream from the application can be divided into one or more data streams in a single SCTP connection. These divided streams are handled as individual data flows and reassembling processing is done independently. The multi-homing support function of SCTP is mainly used for fault tolerant purpose. Thus, SCTP can realize the function of bandwidth aggregation by some additional modification. We research bandwidth aggregation based on TCP for optimization of most of current Internet traffic, while our researches is able to divert to SCTP.

R-MTP [7], pTCP [9], DMTCP [11] and mTCP [13] are transport layer striping schemes which use multiple TCP connections simultaneously. Since these protocols individually support congestion control on each TCP flows, performance degradation problem that the link layer and the network layer approaches encounter is avoided. We discuss these protocols in detail in Sect. 2.2.

Several approaches have been proposed to use multiple TCP sockets at the application layer [14], [15]. While the application layer approaches can improve application throughput by bundling multiple TCP flows without making any modifications to the operating systems, they need the functions of flow control, reliability and in-sequencing to aggregate multiple flows at the application layer. Therefore, they force application developers to implement these functions. In addition, when TCP is used as a transport protocol, it is obviously that implementation of these functions is redundant.

As stated above, several approaches in several layers have been proposed, the transport layer data striping schemes are the most efficient for multi-homed mobile hosts to provide bandwidth aggregation to the application layer in terms of independent congestion control of each path and avoiding the functions of flow control and reliability to be redundantly implemented. In addition, transport layer striping protocols with a function of dynamic IP address configuration between hosts can achieves an end-to-end support for handovers without relying on the underlying infrastructure support [16].

2.2 Transport Layer Bandwidth Aggregation Schemes for TCP

Several transport layer schemes for TCP bandwidth aggregation have been proposed. pTCP, DMTCP and mTCP stripe data across multiple connections based on the congestion window size and/or the window space while R-MTP stripes data based on estimated available bandwidth. As mentioned in [9], rate-based data striping approach such as R-MTP suffers from occurrence of bandwidth fluctuation on a smaller time scale than the interval time of probe transmission.

R-MTP supports host mobility using LLM (Link Layer Manager) [8]. Since the transport layer needs information

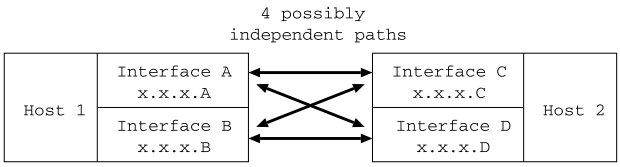


Fig. 2 4 available paths between hosts with two IP addresses.

about availability of interfaces at the link layer to support handovers, LLM intermediates the communication between layers. pTCP uses call-back functions from the network layer to know the current active interfaces [10]. R-MTP and pTCP allows maintenance of connectivity with support from underlying mechanisms at the lower layers even if the host moves. They use the information of the link status. However, it is not necessarily the case that an active link (interface) has a valid IP address (e.g., due to client authentication, etc.). Moreover, although an active link in IPv6 may have multiple IP addresses, R-MTP and pTCP have no way to know these information. DMTCP counts the number of consecutive retransmission times in each TCP connections and it detects link failure on the path that exceeds some threshold value to deal with decrease of the number of available connections. However, it cannot handle addition of new IP address. While mTCP can deal with increase and decrease of the end-to-end paths that response to changes of the number of participation nodes to the overlay network, transmission gets disconnected when IP address of end-hosts changes.

The major goal of these existing research work is to provide architectures that aggregate multiple TCP connections. Hence, providing a selection method of appropriate address pairs for TCP connections has not been considered, while this is another important point to improve communication performance under multi-home environment. Figure 2 illustrates the importance of the address selection. In Fig. 2, two communication hosts have two interfaces allocated different IP addresses, respectively. In this case, it is possible to establish 4 (2×2) independent TCP connections between the hosts. Duplicate use of any IP address in a single application flow leads unfair share with the existing flows as with the data striping schemes at the application layer. Therefore, the following sets of address pairs can be selected in transmission for bandwidth aggregation in Fig. 2.

1. (Host 1: x.x.x.A ↔ Host 2: x.x.x.C) +
   (Host 1: x.x.x.B ↔ Host 2: x.x.x.D)
2. (Host 1: x.x.x.A ↔ Host 2: x.x.x.D) +
   (Host 1: x.x.x.B ↔ Host 2: x.x.x.C)

An end-to-end path in a TCP connection is dependent on the IP address pair, hence selection of the address pairs affects the total available bandwidth obtained in parallel TCP flows. Some address pairs may form the paths that include the link with heavy congestion. To maximize TCP aggregation throughput, it is necessary that the address pairs are selected appropriately.

At the current moment, R-MTP and pTCP are implemented in the network simulator, ns2 [18]. pTCP is also implemented in the kernel of the Linux operating system. DMTCP is a design phase [12]. mTCP is implemented on PULTI (Portable User-Level TCP/IP stack) and includes an overlay router modified from RON [2]. To verify practical effectiveness of a proposed protocol, it is essential that performance of an implementation on real operating systems is evaluated. However, only a few of previous researches evaluated based on implementations.

Thus, bandwidth aggregation protocol for multi-homed mobile hosts should have the following features:

1. detection of link status,
2. way to deal with IP address change,
3. selection of IP address pair, and
4. evaluation based on implementation.

Table 1 Tackling the issues in this paper.

Tackling Issues	R-MTP	pTCP	DMTCP	mTCP
(1) Link Status	○	○	△	△
(2) Addr Changes	△	△	×	×
(3) Address-Pair	×	×	×	×
(4) Implementation	×	○	×	△

Table 1 shows the required features shown above and comparisons of existing researches. As the table shows, there is no protocol that meets the required features.

3. AMS Design

AMS is a new transport layer protocol to stripe application data across multiple paths to provide higher throughput to multi-homed mobile hosts. To easily deploy AMS, AMS does not require additional modifications to any other protocols except TCP. AMS provides the functions of not only bandwidth aggregation but also dynamic response to the changes in the number of connections during communications without the support of underlying infrastructure. Additionally, AMS has the function that the hosts properly select address pairs for efficient bandwidth aggregation. One of the contributions of this paper is to demonstrate the importance of address pairs selection and exhibit a mechanism that provides appropriate address selection for TCP connections.

3.1 Design Overview

Figure 3 shows the architecture of AMS. AMS mediates the communications between TCP and applications. AMS introduces a component called *the AMS aggregation control entity* to use multiple TCP connections to achieve high performance. Congestion control and retransmission are executed in each TCP connection independently, while flow control and reliability control are executed on the aggregated flow by the AMS aggregation control entity. The AMS aggregation control entity has the sending list of all the sending and unacknowledged data segments to manage the send

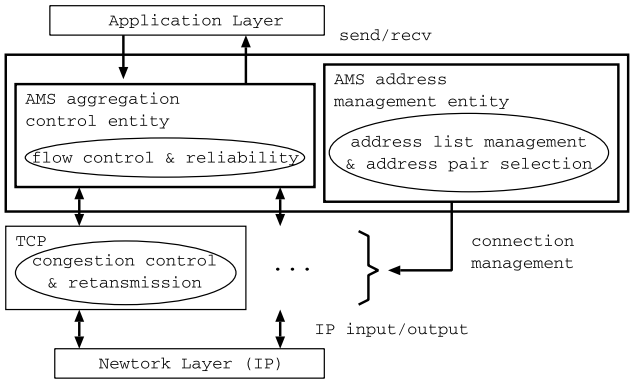


Fig. 3 Overview of the AMS architecture.

buffer of the socket. The entry of the sending list has information of the local sequence number for each flow, the aggregate sequence number for the whole flow, and the segment length. Additionally, the AMS aggregation control entity has the aggregate reassembly queue to deal with out-of-order segments in the aggregated flow.

To meet the required features (1) to (3) (detection of link status, address changes during communication, and address pairs selection), AMS introduces *the AMS address management entity*. It has the address list which includes the information about IP addresses of the sender and the receiver, respectively. This means that AMS manages the information of the network layer (i.e. IP addresses) in the AMS architecture, and allows the AMS hosts to detect active links/interfaces and IP addresses, and multiple IPv6 addresses for one physical interface. Moreover, the AMS address management entity manages information to select address pairs to use for data transmission and deal with dynamic TCP connection configuration (e.g., establishment of additional TCP connection and tear down of existing TCP connections).

Note that AMS is designed to avoid duplicate use of an IP address in a single AMS flow because duplicate use of an address leads unfair bandwidth share with existing flows. Therefore, to aggregate available bandwidth by AMS, both of the nodes must be multi-homed. If one of them has multiple addresses, AMS is effective in terms of address pair selection, dynamic address configuration, and detection of path failure without bandwidth aggregation. In the case that both nodes are single-homed hosts which has single address, they operate as with existing TCP flows using single TCP connection for fair bandwidth share.

3.2 Communication Procedure

To support bandwidth aggregation in TCP, AMS introduces three new TCP options[†]. *The AMS Permitted Option* are used to check whether the peer hosts support the AMS protocol each other. This option holds backward compatibility to the existing TCP flows. The host can inform the other side of information about own IP addresses by using *the AMS Control Option*. *The AMS Common Option* includes the total ordering sequence number (the aggregate sequence number) which is added to all the segments. The receiver side reassembles segments across multiple paths into a single data flow according to the aggregate sequence number included in the AMS Common Option.

The communication procedure of the AMS protocol can be explained by client-server model as an example. The client initially sends the SYN segment with the AMS Permitted Option to the server. If the server supports the AMS protocol, it sends back the SYN/ACK segment with the AMS Permitted Option. After 3-way handshake, the first TCP connection is established between the hosts. It is called the "primary" connection for the sake of convenience. The client and the server exchange their own information about IP addresses each other by the AMS Control Option on the primary connection. The client selects address pairs to use, and establishes new TCP connections with the server. To reassemble data of multiple flows, The AMS Common Option is appended to all the segments.

3.2.1 Permitted Option

The AMS Permitted Option format is shown in Fig. 4. It includes the Type field in addition to the Kind field and the Length field that are the essential fields in TCP. "Normal" and "Extended" are defined in the Type field. The Normal type has length value of 3. It is set for establishment of the primary connection. The Extended type is set for establishment of the other additional connections. It has length value of 7, and includes the value of the 32-bit ISS (Initial Send Sequence Number) in the communication initiator side in the primary connection.

The ISS and the IRS (Initial Receive Sequence Number) values are set in a TCP connection. The host decides the ISS value for a TCP connection with sufficient randomness. The hosts inform each other of this value in 3-way handshake. When the host receives the SYN segment includes the AMS Permitted Option of the Extended type, it specifies which primary connection and socket relates this SYN segment by using the included ISS value and the source address of the incoming segment. Additionally, the hosts always exchange own address information with each other before establishment of new additional TCP connection(s). Therefore, the host which receives the SYN segment(s) including the AMS Permitted Option with the Extended type can check whether incoming segment(s) is(are) sent by the peer host. These operations enable to identify the valid segment with the AMS Permitted Option and prevent memory of nodes from exhausting due to SYN flooding DoS attack.

[†]TCP has provision for optional header fields identified by an option kind field. These numbers have be assigned and registered by the IANA registries. In this paper, we temporarily used three TCP option kind numbers, 10, 11, and 12 for only this experiment, which are already assigned by the IANA registries for other proposals.

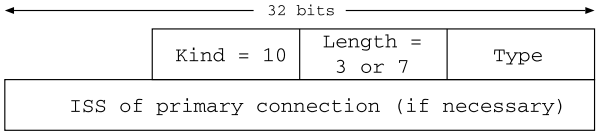


Fig. 4 AMS permitted option format.

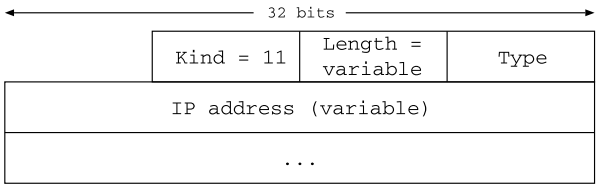


Fig. 5 AMS control option format.

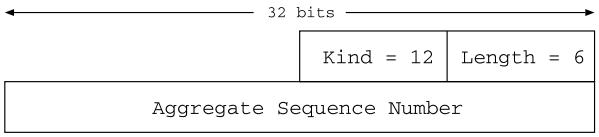


Fig. 6 AMS common option format.

3.2.2 Control Option

Figure 5 shows the AMS Control Option format. It includes the Kind, the Length and the Type fields as the same as the AMS Permitted Option. "Add-IP" and "Delete-IP" are defined in the Type field and are used to report the information of addition and deletion of IP addresses to the other host, respectively. The Option length takes variable value depending on the size of the IP address and the number of the IP address to inform.

3.2.3 Common Option

The AMS Common Option format is shown in Fig. 6. It includes the 32-bit aggregate sequence number to reassemble data at the receiver. The aggregate sequence number is independent of the local sequence number in each TCP flow. pTCP and DMTCP need to add the information about the total acknowledgment number in addition to the total sequence number. In contrast, AMS do not need the information of the total acknowledgment number because the AMS sender manages sending data and is able to specify which data is acknowledged by checking the local acknowledgment number. Therefore, AMS reduces header overhead of 4-bytes compared with pTCP and DMTCP.

3.3 Change in Number of Connections

The AMS hosts can exchange the information about change of available IP addresses during transmission. For instance, when the multi-homed mobile host moves out of the coverage area of wireless LAN or away from connection point of Ethernet, the host can handle decrease of IP addresses and continue the aggregated connection by decreasing the number of connection. On the contrary, when the AMS host moves into the overlapping area to access the Internet, it potentially can increase the number of connection and obtain higher throughput.

By sharing the information about the number of available IP addresses between the hosts, transmission can be continuous while one or more connections are constantly active. AMS ensures continuity of transmission wherever possible. As a result, this leads to support transport layer handovers. It is a part of the functions of host mobility support. Although complete mobility solution also needs to provides location management system, this function should be supported by network layer protocols for mobility support (e.g., Mobile IP, Mobile IPv6, and LIN6). The AMS hosts with mobility support protocols at the network layer obtain an advantage in both bandwidth aggregation and host mobility.

3.4 Address Pairs Selection

As mentioned above, it is quite important that the hosts properly select the address pairs to use for efficient bandwidth aggregation. AMS introduces a function called *address pairs selection* to select an optimal combination of addresses of the nodes. The AMS hosts use address pairs selection schemes to decide the address pairs. While AMS is able to equip various kind of the functions of address pairs selection, we introduce the method called *the SYN flood challenge* in this paper. The SYN flood challenge method uses the SYN segment to select optimal address pairs. After the exchange of the IP address information on the primary connection, the communication initiator (the client in client-server model) sends the SYN segments for all the available address pairs except the address pair of the primary connection to measure RTTs in all the paths. The RTT on the primary connection is measured when it is established.

Suppose that there are two hosts, initiator host A with addresses a1, a2, a3 and responder host B with addresses b1, b2. In AMS protocol, duplicate use of an address is avoided to prevent unfair bandwidth share with existing TCP flows. Since the minimum number of addresses for each hosts is two, AMS hosts can use up to two connections. Addresses b1 and b2 for host B are certainly used to establish two connections. In the point of view that initiator host A choose local address pair (x, y) to use for address pair (b1, b2), which x is a address to establish a connection with address b1 and y is one for b2, host A needs to select one of the address combinations, (a1, a2), (a1, a3), (a2, a1), (a2, a3), (a3, a1), and (a3, a2). Host A calculates sum of RTT for each address combination based on RTT measured by using the SYN segments. (a1, a2) means the combination of a1↔b1 address pair and a2↔b2 address pair. RTT for a1↔b1 plus RTT for a2↔b2 makes sum of RTT for this address combination. Host A selects the combination with the smallest sum of RTT from all address combinations. If retransmission timeouts occurs for reasons of path failure or heavily

congestion, AMS host selects the address combination without this path as RTT value of this one is infinity.

Since the SYN segments are required to establish the TCP connection, the address pairs selection and the connection establishment can be processed concurrently in the SYN flood challenge. Since the SYN segment does not contain the application data, traffic load to the network paths is a little. In addition, the AMS hosts can know which paths are bi-directionally reachable by sending the SYN segments to all the possible paths. To detect path failures, AMS uses TCP retransmission timeout. AMS host manages the number of consecutive timeouts on each connections and is able to detect path failure by detecting consecutive retransmission timeouts on the connection.

4. Implementation

AMS has been implemented on TCP of the NetBSD-1.6.2-stable. Since the version of TCP in the NetBSD-1.6.2-stable is Reno, the implementation in this paper provides the congestion control based on Reno.

In the current NetBSD implementation, a single TCP connection requires an Internet PCB (Protocol Control Block), which is commonly allocated to all the transport layer protocol, and a TCP Control Block, which is allocated to only TCP. Internet PCB includes source/destination port numbers, a pointer to per-protocol PCB, a back pointer to the socket, and so on. The TCP Control Block maintains the TCP-specific information such as the size of the congestion window, the advertised window, the slow start threshold, etc.

4.1 SYN Cache

AMS uses the SYN cache [3] to establish the primary and the additional TCP connections. 3-way handshake is completed with the same procedure of the existing TCP using SYN cache when a primary connection is established. The AMS client host sends the SYN segment including the AMS Permitted Option with the ISS value of the primary connection for the additional TCP connections. When the server in AMS receives such SYN segment, it saves the ISS values included in the AMS Permitted Option to the SYN cache memory, and returns a SYN/ACK segment to the client host. When the server receives the valid ACK segment corresponding to the SYN cache entry, it compares the ISS values recorded in the SYN cache with the IRS values of the TCP connections currently established, and identifies the primary connection associated with the SYN segment with the AMS Extended Permitted Option.

4.2 Memory Allocation

When a new connection is added in AMS, it is necessary to allocate memory resources (the Internet PCB and the TCP Control Block) for the TCP connections other than the space of the socket. Figure 7 shows the main memory resources to allocate for socket API and TCP, and the reference pointers when three TCP connections including the primary connection are established. The Internet PCB of the newly added connection has the one-way pointer to the socket, and the two-way pointer with the own TCP Control Block. *The Flow list* that includes the TCP Control Blocks for the TCP flows currently used is maintained by the Internet PCB of the primary connection. It is used to management data striping across multiple TCP connections.

Table 2 Specification of the machines.

No.	CPU	CPU Clock	Memory
1	AMD ElanSC520	133 MHz	64 Mbyte
2	VIA C3	800 MHz	512 Mbyte
3	XEON	2.4 GHz	512 Mbyte × 2

4.3 Small Traffic Adjustment

The traffic on the Internet consist of a number of various applications, for example, some application protocols generate large traffic (e.g., FTP, iSCSI, etc.) and some protocols often generate small traffic (e.g., Telnet, SMTP, etc.). Transport Layer Bandwidth Aggregation Schemes for TCP are proposed for traffic optimization of the former applications while the latter applications do not need these aggregation protocols except the case of path failure, and so on. Therefore, AMS hosts creates multi-connection(s) for fault tolerance, however, they do not use additional connection(s) for data transfer as long as their send buffer keeps one MSS (Maximum Segment Size) data or more after AMS host sends one segment in the primary connection. This leads suppression of unnecessary network resource consumption and overhead of multi-flows bundling.

5. Evaluation

This section describes the performance evaluations of AMS implementation on NetBSD-1.6.2-stable. The specifications of the machines used in our test network are shown in Table 2. In Table 2, machine No.1 is Soekris Engineering net4501, No.2 is EBS 1563P-563I, and No.3 is XSB-2400-5. In this experiment, the sender and the receiver are running NetBSD-1.6.2-Stable and the routers are running FreeBSD-5.3-Release. The maximum segment size and the maximum window size are set to 1460 and 65536 bytes, respectively.

To check the behavior and evaluate the performance of AMS on various environments, a wired test network was built, in which bandwidth and delay are controlled by the routers. *dummynet* [24], which is a network management tool built in FreeBSD, was used in order to emulate a wide variety of wired and wireless environment with the routers.

5.1 Performance

We evaluated the performance of TCP bandwidth aggregation using the AMS options. Various factors affect the performance of parallel TCP connections. Only the performance of the number of connections was evaluated in this

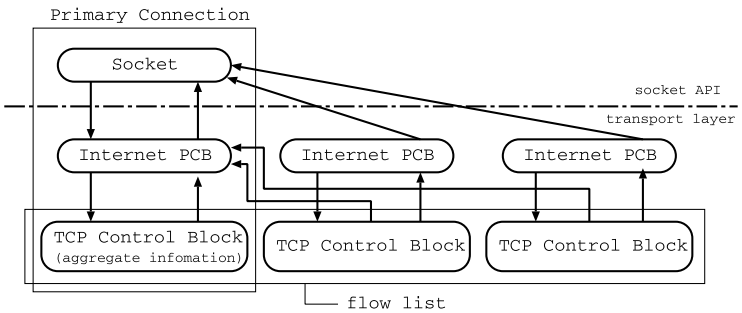


Fig. 7 Reference to memory resources in AMS implementation for NetBSD.

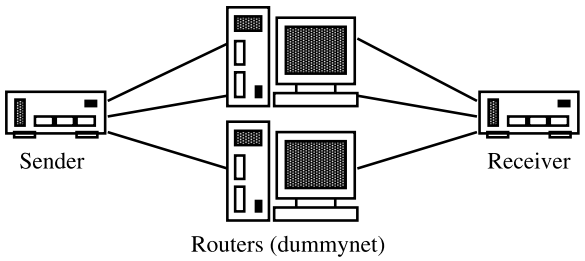


Fig. 8 Experimental topology 1.

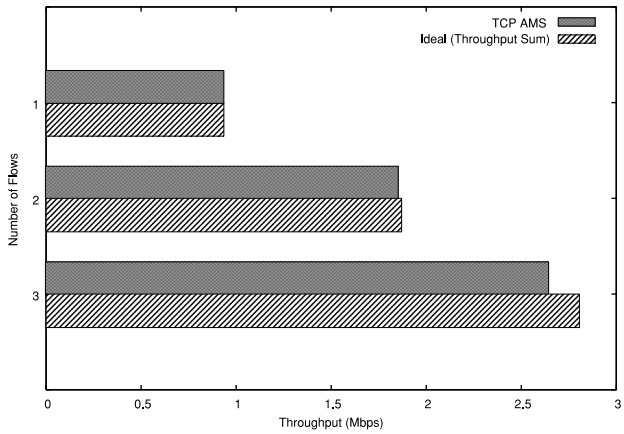


Fig. 9 Effects of number of connections.

paper due to space constraints. In this paper, AMS hosts use the proportion of the outstanding data to the window size on each connection to stripe data segments across multiple connections. The transmission processing is executed on the connection with the smallest ratio in the current active connections. We choose the ratio of the outstanding data to the window, not the window size because of efficient use of multiple flows to avoid biased use of the connection which has a large window size. Evaluations of data striping algorithms are out of scope of this paper. The details of data striping algorithms are discussed in existing researches [9]–[13].

We used Iperf [23] to check the AMS behavior and measured throughput between hosts, which is a tool for measuring TCP and UDP bandwidth performance. The number of connections in parallel use were changed from one to three, and the performance of the existing TCP and the AMS protocol were measured. In this experiment with this network topology, Fig. 8 shows the topology of the test network. No.2 and No.3 machines in Table 2 were used to the sender/receiver and the routers, respectively. In Fig. 8, the links between the sender and the routers are assumed to be wired links while the links between the routers and the receiver are assumed to be wireless links. By using dummynet, bandwidth/delay of the links between the sender and the routers, and between the routers and the receiver were set to 10 Mbps/10 ms and 1 Mbps/0.1 ms, respectively.

Figure 9 shows the performance of TCP AMS and Ideal, which is the total throughput of a single flow on one path multiplied by the number of connections. The x-axis and y-axis in Fig. 9 denote throughput (Mbit/s) and the number of TCP flows, respectively. Each experiment ran for 60 seconds and the results are given by averaging three runs. The result in Fig. 9 shows that AMS can achieve the ideal bandwidth aggregation when the number of connections to aggregate is two. When AMS aggregates three TCP connections, its performance is slightly inferior to Ideal's one in TCP throughput. This is due to the effects of header overhead, management processing of the sending list or a large number of total out-of-order segments between each flow that dummynet introduces.

5.2 Dynamic Address/Connection Configuration

To verify the operation of dynamic addition of a new IP address, establish a new TCP connection, and close the first TCP connection during transmission, the test network that consists of one sender, one receiver and two routers was built. Data transmission from the sender to the receiver using Iperf ran for 90 seconds with the following scenario.

1. The sender has two IP addresses and the receiver has one IP address. The AMS hosts communicate by using a single TCP flow.
2. After 30 seconds, the receiver gets a new IP address.
3. The receiver informs a new IP address to the sender, and establishes a new TCP connection to improve the performance.
4. After 30 seconds, the path for the first TCP connection (the primary connection) fails.
5. The sender detects the path failure by occurrence of consecutive retransmission timeout on that connection,

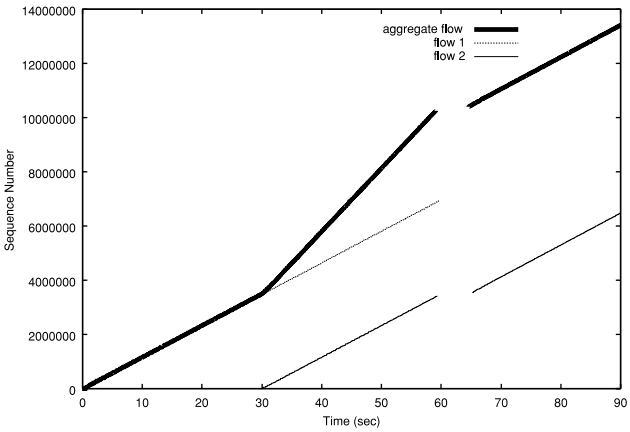


Fig. 10 Dynamic address configuration and continuity of TCP connection transmission.

and the sender retransmits the un-ACKed segments which sent on the first TCP connection, on the second TCP connection.

In this experiment, No.1 and No.3 machines in Table 2 were used to the sender/receiver and the routers, respectively.

The changes of the local sequence number and the aggregate sequence number are shown in Fig. 10. The x-axis and y-axis in Fig. 10 denote time (sec) and sequence number. In Fig. 10, flows 1 and 2 represent the flows of the primary connection and the additional one, respectively. As shown in Fig. 10, AMS can continuously increase the number of TCP connection during transmission, aggregate network bandwidth on two paths and achieve higher throughput to application layer. Moreover, AMS is able to detect path failure by detecting consecutive retransmission timeout on the connection. Figure 10 shows the sender can continue transmission on the additional connection even if the path of the primary connection is down. This operation implies that AMS can support handovers from one access network to another without the underlying infrastructure support when the mobile host has two or more IP addresses during handovers.

5.3 Overhead of SYN Flood Challenge

The memory resources to send the SYN segments and store the measured RTT samples are required in the Internet PCB and the TCP Control Block at the client side, and the SYN cache entry is required at the server side. Since our implementation on NetBSD-1.6.2 allocates only minimal memory resource for a TCP connection before connection establishment, the overhead of the resources should be minor. We also measured the time required to send one SYN segment. It takes about 146 microseconds on No.2 machine as shown in Table 2. This value is so smaller than propagation delay that is commonly the millisecond time scale. The overhead of time required to send a SYN segment may be acceptable.

Next, we examined the method for the address pairs selection using the SYN segments to show advantages of the data striping scheme with the address pairs selection

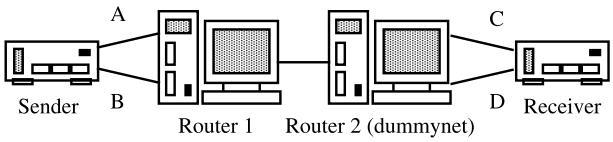


Fig. 11 Experimental topology 2.

Table 3 Link parameters for experimental topology 2.

Path	Bandwidth	Propagation Delay
A-C / B-D	250 Kbps	40 ms
A-D / B-C	1 Mbps	10 ms

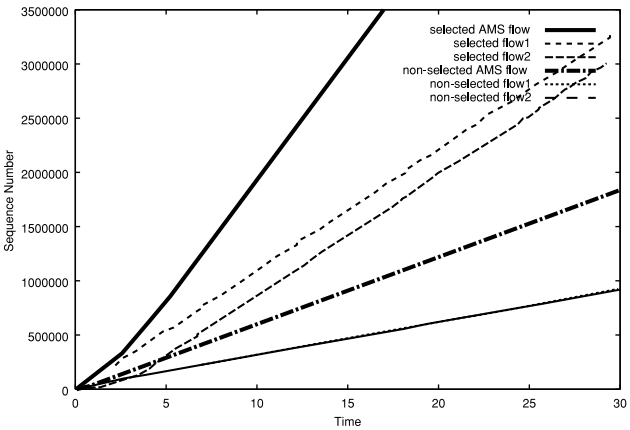


Fig. 12 Selected/non-selected address pairs.

method compared with existing one. This scenario gives validity of the address pairs selection of the AMS protocol. As the same as the example in Fig. 2, two communicating hosts have two IP addresses and there are four independent paths between the hosts. We evaluated AMS and AMS without the function of the address pairs selection instead of other existing bandwidth aggregation schemes. Figure 11 shows the topology of the test network for this experiment. No.2 and No.3 machines in Table 2 were used to the sender/receiver and the routers, respectively. In Fig. 11, A to D represent IP addresses that receiver and sender have. Router 2 with dummynet in Fig. 11 emulated bandwidth and propagation delay as shown in Table 3.

We examined the operation of the AMS address pairs selection when the sender and the receiver established the primary TCP connection from A to C. The Changes in the local sequence number are shown in Fig. 12. The x-axis and y-axis in Fig. 12 denote time (sec) and sequence number. In Fig. 12, non-selected flows 1 and 2 represent the A↔C connection (this is the primary connection) and the B↔D one, selected flows 1 and 2 represent the flows of A↔D and B↔C, respectively. As shown in Fig. 12, while non-selected AMS aggregate flow (non-selected flows 1 and 2) suffers from slower paths, AMS with SYN flood challenge can properly select two address pairs (A↔D and B↔C) from four available one by measuring RTTs of all the possible paths using SYN flood challenge even if the primary

connection includes slower link. This leads higher throughput than any other protocols for TCP bandwidth aggregation in a number of situations.

6. Discussion

It is important for efficient data transmission to select IP addresses to establish each TCP connection on an aggregate TCP session because an end-to-end address pair decides the end-to-end path and available bandwidth that the hosts can obtain. In this section, we discuss the methods for selection of address pairs when the hosts have two or more IP addresses.

mTCP uses `traceroute` command to select some paths on overlay network to use in the communication of TCP bandwidth aggregation. The hosts can detect physical overlapping links with large queuing delay by using `traceroute`. However, it is undesirable that the transport protocol employs the function of ICMP at network layer from the viewpoint of the Internet layering model. Additionally, ICMP messages may be dropped by packet filters of some ISPs due to security reasons.

One way that the hosts select address pairs is to use the network address information. IPv6 has been designed from its foundation to support efficient, hierarchical addressing and routing. IPv6 Global Unicast Address Format [17] has been defined to allocate address blocks to local Internet registries and the ISPs in a planned manner. Therefore, it is expected that a segment passes across a physically nearer path when the address block between the IP addresses is logically near. Thus, the hosts can select preferable address pairs by exchanging netmask in addition to IP address information. However, unlike IPv6, some of the current IPv4 addresses have been allocated without plan, the logical closeness between two IP addresses does not necessarily correspond to the geographical distance between the two hosts. Additionally, this method does not consider existence of the path with congested link.

In this paer, we adopt the SYN flooding challenge approach to decide appropriate address pairs. This approach can be easily implemented and can avoid to select congested paths or paths with long RTT without wasting much bandwidth. However, since this approach depends on measuring RTT which is not always accurate to estimate the available bandwidth in the path, it might select inappropriate paths in some cases. Some methods that estimate available bandwidth of the communication path have been studied [19]–[22]. The path selection function in AMS can be improved by combining these techniques.

7. Conclusion

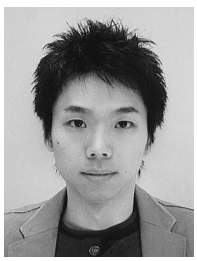
In this paper, we discussed the problems related to bandwidth aggregation in TCP for multi-homed environments. We proposed a new framework for TCP called AMS which supports simultaneous use of multiple TCP connection to improve communication performance. In addition to bandwidth aggregation, it also supports dynamic changes of address and connection configuration during data transfer. Furthermore, AMS provides the function to select appropriate address pairs for TCP connections so that it can fully utilize multi-homed environments. The new TCP options (AMS Permitted, AMS Control, AMS Common) were defined to realize TCP bandwidth aggregation suitable for multi-homed mobile hosts. Since all the AMS operations are actualized without additional modification to socket API and IP, it is not necessary that application and network infrastructure are modified.

We implemented AMS on NetBSD and evaluated it in our test network, in which dummynet was used to emulate bandwidth and delay. The measured results showed that AMS can achieve nearly idealized bandwidth aggregation by simultaneously using three TCP connections. AMS could continuously increase the number of TCP connections during transmission. Aggregated network bandwidth on two paths provided higher throughput to the application layer. Additionally, AMS could continue TCP transmission even if the first TCP connection is down. Appropriate behavior of address pairs selection using the SYN flood challenge was also shown.

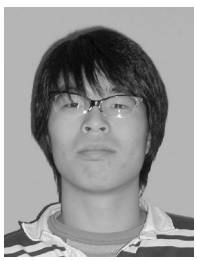
References

[1] R. Stewart, Q. Xie, K. Morneault, C. Sharp, H. Schwarzbauer, T. Taylor, I. Rytina, M. Kalla, L. Zhang, and V. Paxson, "Stream control transmission protocol," RFC 2960, Oct. 2000.
[2] D. Andersen, H. Balakrishnan, F. Kaashoek, and R. Morris, "Resilient overlay networks," Proc. ACM SIGOPS Operating Systems Review, vol.35, no.5, pp.131–145, 2001.
[3] J. Lemon (FreeBSD Project), "Resisting SYN flood DoS attacks with a SYN cache," Proc. BSDCon, pp.89–97, 2002.
[4] J. Duncanson, "Inverse multiplexing," IEEE Commun. Mag., vol.32, pp.34–41, April 1994.
[5] A.C. Snoeren, "Adaptive inverse multiplexing for wide-area wireless networks," Proc. IEEE GLOBECOM'99, pp.1665–1672, Dec. 1999.
[6] D.S. Phatak and T. Goff, "A novel mechanism for data streaming across multiple IP links for improving throughput and reliability in mobile environments," Proc. IEEE INFOCOM 2002, vol.2, pp.773–781, June 2002.
[7] L. Magalhaes and R. Kravets, "Transport level mechanisms for bandwidth aggregation on mobile hosts," Proc. IEEE ICNP 2001, pp.165–171, Nov. 2001.
[8] L. Magalhaes and R. Kravets, "A transport layer approach to host mobility," Proc. ACM MobiCom Student Poster Session, Sept. 2002.
[9] H.-Y. Hsieh and R. Sivakumar, "A transport layer approach for achieving aggregate bandwidths on multihomed mobile hosts," Proc. ACM MobiCom 2002, pp.35–46, Sept. 2002.
[10] H.-Y. Hsieh, K.-H. Kim, and R. Sivakumar, "An end-to-end approach for transparent mobility across heterogeneous wireless networks," ACM/Kluwer Mobile Networks and Applications Journal (MONET), Special Issue of Integration of Heterogeneous Wireless Technologies, vol.9, pp.363–378, Aug. 2004.
[11] M. Tonouchi, U. Tei, H. Mineno, S. Ishihara, O. Takahashi, and T. Mizuno, "Design of dynamic multi link TCP for mobile communication," DICOMO, pp.13–16, 2003.
[12] M. Tonouchi, H. Mineno, S. Ishihara, O. Takahashi, and T. Mizuno, "A study on retransmission control of multipath-extended TCP," IPSJ SIG-MBL: Mobile Computing, vol.28, no.27, March 2004.
[13] M. Zhang, J. Lai, A. Krishnamurthy, L. Peterson, and R. Wang, "A transport layer approach for improving end-to-end performance and

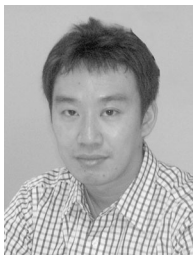
robustness using redundant paths," Proc. USENIX Annual Technical Conference 2004, pp.99–112, June 2004.
[14] M. Allman, H. Kruse, and S. Ostermann, "An application-level solution to TCP's satellite inefficiencies," Proc. WOSBIS, pp.100–107, Nov. 1996.
[15] T. Hacker and B. Athey, "The end-to-end performance effects of parallel TCP sockets on a lossy wide-area network," Proc. IEEE IPDPS, pp.434–443, April 2002.
[16] T. Goff and D.-S. Phatak, "Unified transport layer support for data striping and host mobility," IEEE J. Sel. Areas Commun., vol.22, no.4, pp.737–746, May 2004.
[17] R. Hinden, S. Deering, and E. Nordmark, "IPv6 global unicast address format," RFC 3587, Aug. 2003.
[18] http://www.isi.edu/nsnam/ns/
[19] S. Keshav, "A control-theoretic approach to flow control," Proc. ACM SIGCOMM'91, pp.3–15, 1991.
[20] S. Mascolo, C. Casetti, M. Gerla, S.S. Lee, and M. Sanadini, "TCP Westwood: Bandwidth estimation for enhanced transport over wireless links," Proc. ACM MobiCom 2001, pp.287–297, July 2001.
[21] M. Jain and C. Dovrolis, "End-to-end available bandwidth: Measurement methodology, dynamics, and relation with tcp throughput," Proc. ACM SIGCOMM, pp.295–308, Aug. 2002.
[22] C. Dovrolis, P. Ramanathan, and D. Moore, "Packet-dispersion techniques and a capacity-estimation methodology," Proc. IEEE/ACM Transactions on Networking (TON), pp.963–977, Dec. 2004.
[23] http://dast.nlanr.net/Projects/Iperf/
[24] http://info.iet.unipi.it/˜luigi/ip_dummynet/



Shunsuke Saito received the B.E. and the M.E. in computer science from Keio University, Yokohama, Japan, in 2003 and 2005, respectively. He was enrolled in Teraoka Laboratory, Graduate School of Science and Technology, Keio University, Japan. His research interests are TCP performance enhancement in wireless mobile and multi-home environments.



Yasuyuki Tanaka is a researcher at the Corporate Research and Development Center of Toshiba Corporation in Kawasaki, Japan, which he joined in 2005. He received the B.E. and the M.E. in computer science from Keio University, Yokohama, Japan, in 2003 and 2005, respectively. He was enrolled in Teraoka Laboratory, Graduate School of Science and Technology, Keio University, Japan.



Mitsunobu Kunishi received the B.E., M.E. and Ph.D. degrees in computer science from Keio University, Japan, in 1999, 2001 and 2004, respectively. He works for Fixstars Corporation and Keio University. His research interests are IP mobility, inter-layer network architecture, TCP performance enhancement and All IP computer architecture.

Yoshifumi Nishida received a Master's degree and a Ph.D. in Media and Governance from Keio University in 1996 and 1999, respectively. He joined Sony CSL in 1999. His research interests include congestion control, traffic analysis, transport protocols and wireless communication.

Fumio Teraoka received a master degree in electrical engineering and a Ph.D. in computer science from Keio University in 1984 and 1993, respectively. He joined Canon Inc. in 1984 and then moved to Sony Computer Science Labs., Inc. (Sony CSL) in 1988. Since April 2001, he is a professor of Faculty of Science and Technology, Keio University. He received the Takahashi Award of JSSST (Japan Society for Software Science and Technology) and the Motooka Award in 1991 and 1993, respectively. He also received the Best Paper Award in 2000 from IPSJ (Information Processing Society Japan). His research interest covers computer network, operating system, distributed system. He contributed to the activity of the Mobile working group of IETF by developing Virtual IP (VIP). He was a board member of IPSJ from 2000 to 2002. He is a member of JSSST from 2005. He is a member of ACM, IEEE, Internet Society, JSSST, and IPSJ.