INVITED PAPER *Special Section on Networking Technologies for Overlay Networks*

Overlay Network Technologies for QoS Control

Tutomu MURASE[†a)], **Hideyuki SHIMONISHI**[†], ***Members***, ***and*** **Masayuki MURATA**[††], ***Fellow***

SUMMARY Overlay networks are expected to be a promising technology for the realization of QoS (Quality of Service) control. Overlay networks have recently attracted considerable attention due to the following advantages: a new service can be developed in a short duration and it can be started with a low cost. The definition and necessity of the overlay network is described, and the classification of various current and future overlay networks, particularly according to the QoS feature, is attempted. In order to realize QoS control, it is considered that routing overlay and session overlay are promising solutions. In particular, session and overlay networks are explained in detail since new TCP protocols for QoS instead of current TCP protocols that control congestion in the Internet can be used within overlay networks. However, many open issues such as scalability still need further research and development although overlay networks have many attractive features and possess the potential to become a platform for the deployment of new services.

key words: *overlay network, QoS, quality, routing, session, transport, TCP/IP, relay*

1. Introduction

Overlay networks are not a new technology; however, they have recently been widely noticed. Thus, we would like to understand why it attracts a considerable amount of attention. The two main features of this overlay network—development of a new service in a short duration and start with a small cost—are suitable to fulfill the present needs of users.

The following two backgrounds can explain overlay network deployment:

(1) Broadband network deployment
(2) Borderless users and infrastructure

First, broadband access lines such as FTTH (Fiber to the Home), ADSL, and IEEE 802.11 wireless LAN are rapidly being deployed. A broadband access network rarely experiences a bottleneck when a new service is provided out of the network, i.e., in end hosts. Thus, new applications such as the so-called P2P (Peer to Peer) and VoIP (Voice over IP) have recently been developed. Some of the new applications handling "rich contents" have individual and strong requirements for QoS in public networks.

Next, Internet usage is expanding among people who are not familiar with Internet technology, that is, people with low information literacy. This rapidly increases the social demand for technology to protect such people from virus/spyware threats and from becoming unintended attackers as hosts of botnets. There is, however, no effective countermeasure to prevent attacks that rapidly change (evolve) their protocols or exploit applications. Thus, a countermeasure always lags behind the threats. Therefore, the rapid deployment of novel and low cost countermeasures is necessary.

In order to meet the above network requirements, there are three countermeasures as follows:

(1) Adding new features to an existing network
(2) Constructing a new independent network
(3) Realizing new features on a virtual network on top of the existing network

The first approach, however, seems unrealistic because the addition of new features, which are efficient only for specific applications, cannot be justified from a viewpoint of quality and cost. For example, adding a new feature on routers often increases the number of processing steps in forwarding packets. For the applications that do not benefit from it feature, this feature merely increases their delay (decreases their quality) and imposes costs on them. Moreover, the first countermeasure is not practical since it requires a significant investment from the beginning; this is because the addition of a new feature to an existing network requires the modification of all the routers and switches that have been deployed. Most of the newly proposed features have not been realized because it is difficult to formulate a convincing business plan that promises good returns for a high initial investment.

Service providers and investors are usually caught in a dilemma. This is because service providers initially require finance from investors to start new services, whereas investors prefer to wait until these services actually start drawing revenues. This is one of the possible explanations why most of the proposed methods for adding a new feature or service to an existing network have not been realized.

The second countermeasure (construction of a new independent network) is more difficult than the previous one from a practical viewpoint, except for few promising services, e.g., the VoIP service.

The third countermeasure—an overlay network approach—will be discussed in this paper. The following are the advantages of the overlay network approach:

(1) Virtual or logical networks with new features can be re-

Manuscript received January 16, 2006.
Manuscript revised April 6, 2006.

[†]The authors are with NEC, Kawasaki-shi, 211-8666 Japan.
[††]The author is with Osaka University, Suita-shi, 565-0871 Japan.
a) E-mail: t-murase@ap.jp.nec.com

DOI: 10.1093/ietcom/e89–b.9.2280

alized using the existing network without the addition of new features

(2) A "small start" is possible
(3) Standardization is not necessary

These advantages enable us to construct a (virtual) network and start new services that are comparatively faster and cheaper than the existing approaches.

An overlay network provides a solution for the dilemma between investors and service providers. This network enables service providers to develop new features with small investments and provide experimental service to advanced users. Due to this, service providers can demonstrate the feasibility of new services to investors; this approach is expected to decrease the obstacle for the investments.

There are two categories to be considered while realizing overlay networks. P2P file sharing applications, such as, Napster [1], Gnutella [2], and Kazaa [3] are the most popular examples of overlay networks provided by end hosts. In a legacy network, since the network nodes have to be reconfigured, it takes a long time to provide a new type of content distribution or a new type of connectivity. On the other hand, the abovementioned applications do not require any network support and an overlay network can be instantly set up by installing these applications into user hosts. This feature of overlay networks should be one of the reasons for the rapid and wide deployment of these applications. In the future, overlay network technology seems inevitable for incorporating new Internet applications.

The other category of overlay networks is those comprising network nodes; this type of network is typically called PEP (performance enforcement proxy) or middle box. In this paper, this network is generally referred to as overlay nodes. These nodes are provided by network operators such as carriers or ISPs, and not by end users. Further, operators can attain high incomes from the new services realized by these nodes. This type of overlay network is also used for providing content distribution and is often referred to as CDNs (content delivery networks) or virtual connectivity that is commonly known as VPN (virtual private network).

In addition to these scenarios, QoS has attracted considerable attention. The rapid deployment of high speed Internet access environments has facilitated the emergence of diverse applications like IP telephony, IP broadcasting, online games, and so on; these have indeed increased the requirements for QoS mechanisms. These mechanisms have been extensively studied as those that strictly or statistically share network resources enforced at each network node, resulting in slow deployment. On the other hand, overlay network technology is a promising mechanism to provide QoS because of its flexible deployment. As shown in Fig. 1, with the recent growth of network capacities, and the potential ability of tuning transport behaviors between overlay nodes to meet specific application requirements, overlay networks are becoming suitable candidates for providing QoS.

In this paper, we overview the overlay network technology that realizes QoS, which has recently attracted considerable attention. In the first half of the paper, we describe the overlay network technology that is proposed in order to provide QoS control. In the second half of the paper, we focus on a session layer overlay network that uses session layer connections such as TCP and UDP in order to construct virtual or logical networks. Further, this network has been extensively studied to achieve QoS control. Currently, TCP protocol that is used in end hosts plays a major role in QoS control in the Internet. Thus, the authors expect that TCP can be successfully used for QoS control in the Internet.

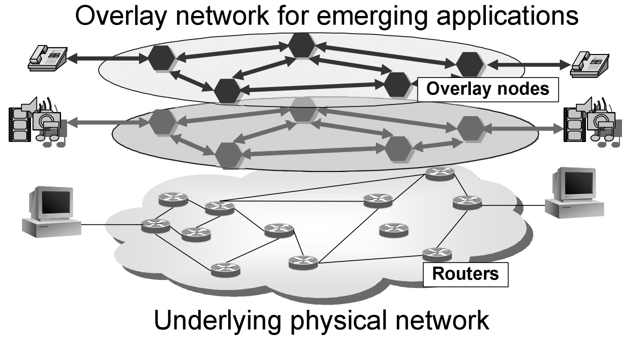

Overlay network for emerging applications
Overlay nodes
Routers
Underlying physical network

Fig. 1 Session overlay network for QoS control.

2. Overlay Network

2.1 Overlay Network and Its Services

First, we define an overlay network for the following discussions. An overlay network is a virtual or logical network that is constructed using the same or higher layer technologies rather than using underlay or legacy networks (see Fig. 1).

A virtual network can be constructed on the basis of two approaches:

(1) Higher-layer-protocol-based network (Here, we define this virtual network as an overlay network.),
(2) Lower-layer-protocol-based network.

This means that a virtual network is not directly equivalent to an overlay network since virtual networks can also be realized by lower layer protocols. The wavelength path network in WDM can be called a virtual network. The traffic engineering service of MPLS (Multi Protocol Label Switching) establishes a tunnel on an IP network in order to achieve QoS control and bandwidth management control in a manner similar to ATM. Such lower-layer-protocol-based networks have a disadvantage in terms of scalability and deployment.

Active network is another virtual network technology. It is similar to an overlay network. An active network is defined as a network with programmable nodes throughout the network. In comparison with an active network, an overlay network has functions only on the edge nodes or terminals. An overlay network can thus overcome limitations that are too difficult to be programmed into each core node on an active network. An overlay network does not attempt to make any changes in the underlay network. In other words, it is based on KISS—keep it simple, stupid [4].

Some applications and services, which are examples of overlay networks, are as follows: Internet VPN [5], P2P file sharing system [6], Application level multicast [7], SSL VPN [8].

The following overlay networks have attracted considerable attention because they are expected to realize the QoS functions in practice.

- Routing overlay network: This network provides better routing and bandwidth management than underlay networks,
- Session overlay network: This network provides QoS control by using session layer protocol relay.

It should be noted that not only these overlay network technologies but also many new overlay network technologies have been evaluated on network test beds such as PlanetLab [9]. This test bed is constructed on the public Internet and facilitates the testing of new technologies

2.2 Conventional Approaches and Problems in QoS Control

In order to realize QoS, many conventional approaches, including DiffServ and IntServ, have been slow in their deployments in public networks because they need special router functions to underlay networks. That is why a paradigm known as hourglass paradigm has been paid much attention when discussing whether underlay networks such as IP networks are the ones providing new such functions as QoS. Based on the paradigm, it is a source of a potential disruptive innovation to keep underlay networks as simple as possible. It also helps the network be scalable and robust infrastructure.

With regard to this paradigm, common functions such as IP network functions should be minimized. Special functions such as QoS mechanisms should be employed on network edges or end hosts. This is a central principle of overlay network

On the other hand, if the functions of an IP layer are enhanced by adding a new function, an architecture failure is possible due to the complexity caused by various related functions. Traditional QoS controls such as Intserv and Diffserv have not yet been able to solve the following problems:

(1) The chicken and egg problem in business is as discussed in Sect. 1
(2) Traffic engineering for QoS;
There is no single way of traffic engineering for satisfying widely diversified QoS requirements of every applications or services. Moreover, once a network is optimized for a certain set of applications, it can hardly be optimized again for new applications.
(3) Other network parameters such as reliability, flexibility, and fairness;

In order to provide QoS for end users, there are a bunch of unsolved problems, as well as how to realize QoS mechanisms. For example, network operators have to consider business issues like billing and authentification, as well as operation and management issues.

It should be noted that it is necessary to continue to make efforts on IP network enhancement. If the cost advantage is greater than the disadvantage of modifying the existing IP network, the modification can be carried out. However, we have to consider the deployment cost in terms of time because of the extensiveness of the existing IP network.

3. Characteristics of Overlay Network

3.1 Characteristics and Categories of Overlay Networks

Various categories of overlay networks have already been proposed in terms of their objectives and functions. For better understanding, it is useful to categorize overlay networks based on their characteristics. The followings categories are hereby listed (see Fig. 2):

(1) Objective,
(2) Control using underlay network QoS information,
(3) Processing on an overlay node, and
(4) Protocol layer on which an overlay network is constructed.

(1) Objective
Objectives of overlay networks are divided into two categories; QoS and the others. It is most difficult to realize QoS functions on the Internet. QRON [10], SON [11], session overlay [12], OverQoS [13], and so on use overlay networks in order to overcome QoS drawbacks such as delay and loss of packet level. RON [14] focuses on the QoS of the connectivity level.
(2) Control using underlay network QoS information
It is very useful if the overlay networks which try to improve end-to-end QoS can get accurate QoS information from the underlay networks. It is, however, difficult with today's real network. Typically, in one case, network operators may have overlay nodes in their networks for providing their own services. Or end users may provide their own hosts to organize overlay networks independent of the network operators.

It is assumed that the network operator installs the overlay network. In this case, the overlay network is only expanded within the range of the operated network. Because the current network is divided into the access network, metro network, core network, etc. and is separately operated, the effect of end-to-end control cannot be expected. Therefore, it is appropriate to construct the overlay network within the network in order to improve the QoS when SLA (service level agreement) should be achieved for an individual network. Further, solving the problem of interference between the underlay network and overlay networks [15], [16] appears to be easy because the network operator can serve as a single resource manager among coexisting overlay networks, or between overlay networks and the underlay network. A typical case is an overlay network where IP broadcasting is performed by using the multicast node of an

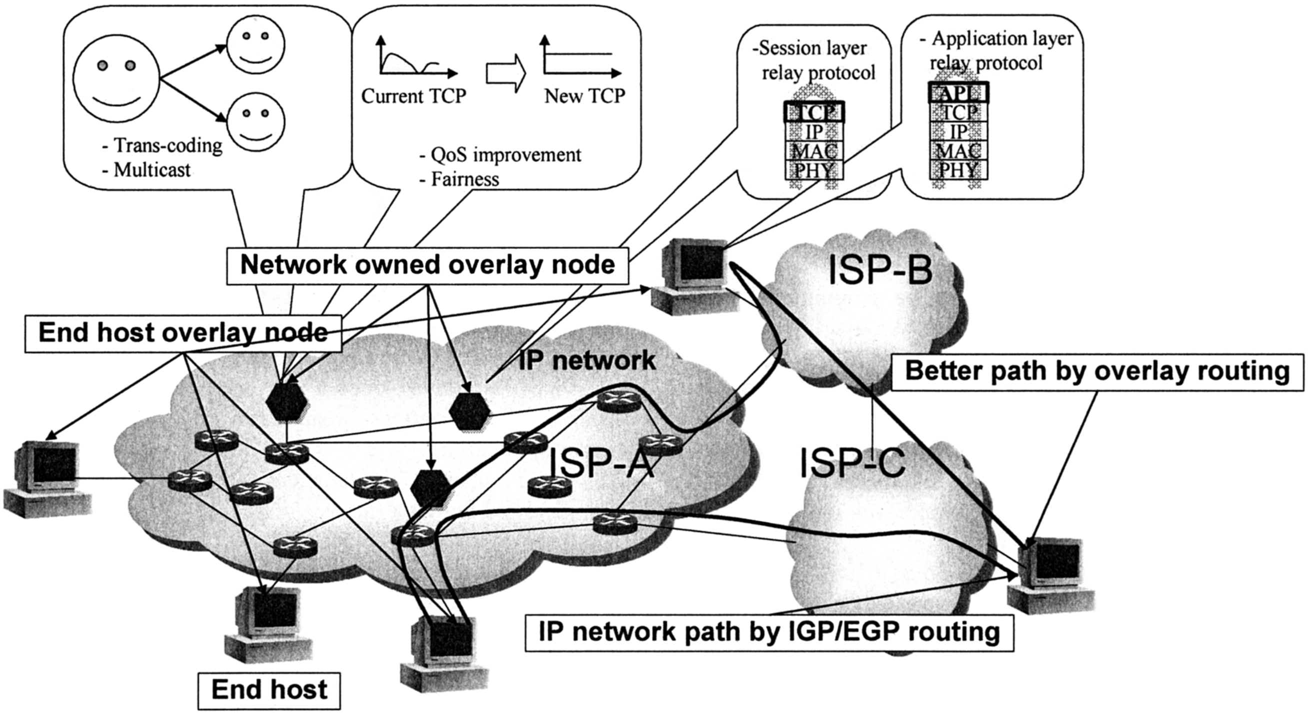

- Trans-coding
- Multicast
Current TCP
New TCP
- QoS improvement
- Fairness
-Session layer relay protocol
TCP
IP
MAC
PHY
- Application layer relay protocol
APL
TCP
IP
MAC
PHY
Network owned overlay node
End host overlay node
ISP-B
IP network
Better path by overlay routing
ISP-A
ISP-C
IP network path by IGP/EGP routing
End host

Fig. 2 Overlay network categories.

application level multicast.

When an overlay network is installed outside the network, the probability of realizing the end-to-end function is high; however, interference from the underlay network can pose a problem. In particular, an underlay network may enforce a traffic control policy that is not desirable for overlay networks, for example, when a P2P file sharing program generates an elephant flow [17], the flow is restricted within the underlay network. The typical applications that fall into this category are CDN, Internet VPN, GRID and softether [18], [19], the file search of P2P file sharing, and the data transfer of Freenet-type P2P file sharing [20].

(3) Processing on overlay node

Only some overlay network applications process relayed data. Payload data is generally not modified although the protocol header used by the relaying protocol is generally modified. In application multicast and trans-code [21], [22] services, payload data is processed on the overlay node. Node-by-node encryption data transfer and FEC (forward error correction) or redundant transport technologies process payload data in a similar manner.

Store-and-forward is another processing technique. However, this is not a payload processing technique and is time consuming. Freenet-type P2P application and Caching overlay network [74] are introduced. In the Caching overlay network, cached data is stored or forwarded when the forwarding link is busy or idle, respectively.

(4) Protocol layer on which an overlay network is constructed

An overlay network is constructed on either of protocol layers including, network layer (IP), transport layer (TCP/UDP), or application layer. Which layer should be used?

Assume an IP network as an underlay network. "IP in IP" is an overlay network and is termed IP tunneling as in IP VPN. A relay on the IP layer has a light processing overhead. However, it is difficult to analyze the upper layer protocol. This implies a coarse-grain QoS control such that it is difficult to meet individual QoS requirements. X-bone [27] gives an application developer a readily available IP in IP overlay networks.

An overlay network in the session layer is referred to as a session overlay network in this paper. One of the session overlay networks [12] is dedicated to QoS control. It relays session protocols and instead of only standard protocols, namely, TCP/UDP, it can also use various session protocols. Although TCP/UDP port numbers can be used as a clue to application discrimination for applying QoS policy, there emerges an exceptional case where the well-known port 80 is also used in tunneling protocols as well as in web-based applications.

It is possible to meet an application requirement when an application overlay network is used. However, we have to pay much cost for protocol processing overhead for handling application. In addition, overlay nodes have to have all application protocols that should be treated and relayed.

3.2 Routing Control and Session Control

Applications and services on an overlay network function satisfactorily without any problems; this is based on the assumption that an IP network is sufficiently fast, wideband, and stable with respect to an overlay network. However, this type of network cannot be fitted into the current public network. Thus, it is becoming increasingly important to know the physical network (IP network) state and to control an overlay network by using the state information. In particular, this is essential if specialized QoS control is desired.

Most of all QoS control methods that have been implemented and proposed are categorized into the space-domain or the time-domain control. For example, MPLS traffic engineering is fallen in space-domain control category, and buffer priority control and connection admission control are considered to be time-domain control methods. Since in overlay networks, the same approach can be employed, there have been many attempts to realize QoS control. The two most promising approaches, routing overlay and session overlay, are introduced in the following section.

The routing overlay and session overlay hereby mentioned are considered to correspond to space-domain control and time-domain control, respectively. The two approaches can be implemented together in the same overlay networks or be separately implemented depending on a specific QoS requirement.

3.2.1 Routing Overlay

Unstable IP route is a well-known problem encountered in many applications; this problem needs to be resolved immediately. In order to obtain stable IP routes, overlay routing as an integral part of the fundamental functions is proposed [23]. The objectives of overlay routing are as follows:

(1) Reliability routing
(2) QoS routing.

(1) Reliability routing
The objective is to get a smaller cost for routing, failure discovery and recovery than current IP routing such as given by BGP. The cost of IP routing is higher cost than overlay routing because of the large number of IP nodes.

IP reachability is the most important and basic function of the Internet. RON [14] showed that an unreachable state occurs due to route failure. For an application, since an unreachable state means infinite delay, a better path selection ensures reachability.

RON describes the experiment for ensuring reachability in the Internet. RON exhibited path outages of tens of minutes or more. Twelve RON nodes can avoid outage paths of more than 30 minutes 32 times in a 64-hour experiment. It also shows that even one more hop in addition to a direct hop can significantly improve reachability. By improving delay and loss performance, the TCP throughput is also improved. Based on RON, it is concluded that overlay routing is effective and is useful for shifting routing functions to hosts.

(2) QoS routing
Other experimental results in [24]–[26] show that overlay routing can improve packet level performance by avoiding congestion even if the underlay network is stable and reliable. In practice, however, we have to consider costs such as scalability and control overhead. A study in [25] has indicated that the performance metrics of overlay routing are (1) the number of overlay nodes, (2) the number of nodes that exchange routing information, (3) the frequency at which routing information is updates, and (4) the maximum number of hops.

SON [11] uses a different approach whereby routing is decided after confirming the availability of the required bandwidth on a selected path by using a bandwidth broker.

These studies suggest that QoS improvement is possible without changing the routing in an underlay network. Although the effectiveness should be discussed as a tradeoff between improvement and routing overhead, overlay routing works satisfactorily under various conditions. This is because routing in the underlay network is not expected to be optimum, partly due to political selections of a route, as in BGP routing among ISPs. It should be noted that schemes for routing information exchange and for the notification of congestion information in overlay networks are still under study.

3.2.2 Session Overlay

Session overlay has attracted considerable attention with regard to TCP overlay and application overlay for QoS control. Current high speed networks reveal throughput bottlenecks caused by the TCP acknowledgement mechanism. Some network equipment vendors announced a TCP throughput improvement mechanism or a new performance-enhanced TCP implemented in their products. The authors also confirmed that the experiments performed in the Internet revealed a throughput improvement that is three to ten times higher than ordinary TCP, which only gets a maximum of 10% of the available bandwidth [42].

The TCP acknowledgement mechanism piggybacks congestion control information in order to manipulate this information on an overlay node; this is attempted in order to attain fairness and QoS differentiation. Session overlay in detail will be described in Sect. 4.

3.3 Experimental Overlay Network

Some experimental networks for overlay network research are available in a dedicated network or in a virtual network over the public Internet. There are two types of experimental networks—one is expanded on the public Internet and the other is a dedicated network for experimental purposes only.

If an overlay node consists of only end hosts, it is not very difficult to establish an overlay network. The X-bone [27] research team offers freeware for constructing a user-defined overlay network. PlanetLab [9], [28] is an experimental network, which is constructed is the public Internet, and offers several programmable overlay nodes. Members who provide their hosts as an overlay node to Planet

Lab can use the virtual machines implemented on the provided nodes. Over 600 of PlanetLab's virtual machines, i.e., overlay nodes, are currently being used and over 500 experiments are currently being performed concurrently as of January 2005. If the public Internet is the underlay network, it is difficult to identify network states that change rapidly with time. Further, it is difficult to identify the reasons for the difference between the experimental results and the expected results. The reasons may come from a traffic enforcement by the underlay network. On the other hand, PlanetLab-like experimental networks are very beneficial for users who plan to start a service on the public Internet in the near future because they can understand how the underlay network reacts or how the underlay traffic affects the service.

Examples of a dedicated network are Internet2 or JGN II [75]. In this type of network, experiments are managed or preorganized and users determine the characteristics of underlay traffic characteristics. Traffic control policy in underlay networks is already known or does not exist.

4. Session Overlay Network

4.1 Definition of Session Overlay Network

In this section, we describe a certain type of overlay network technology that we refer to as session overlay network. "Sessions" are established between a sender and a receiver in order to provide transport functions; this is typically performed by using TCP/UDP. A session overlay network is defined as an overlay network that relays multiple sessions between a sender and a receiver. An overlay node terminates user initiated sessions and relays these sessions to connecting sessions at the next overlay node or final destination.

4.2 Objectives and Advantages of Session Overlay Network

Although there are many objectives of session overlay networks, in this paper, we will focus on QoS control. QoS mechanisms have been extensively studied as a mechanism that strictly or statistically shares network resources enforced at each network node, thereby resulting in slow deployment. On the other hand, the overlay network technology is another promising mechanism to provide QoS because of its flexible deployment.

Since transport protocols play an important role in traffic control, new variants of transport protocols, which have differentiated traffic control functions based on application requirements, have been extensively studied. Based on this, a session overlay network, which provides a variety of session controls for QoS within transport traffic, should be a worthwhile objective of overlay networks.

An advantage of providing QoS controls over session overlay networks is that the former can be provided without modifying the existing routers or end hosts. More specifically:

(1) It is not necessary to modify application software or the operation systems of end hosts or servers. However, new variants of transport protocols can be introduced in end hosts and servers in order to achieve QoS control. However, the modification of a large number of end hosts in corporate networks or those in residential networks will be difficult; this would lead to substantial delays in the deployment of new services.

(2) It is not necessary to modify existing routers and switches in the underlay physical networks. QoS mechanisms—Intserv and Diffserv—executed at routers have been studied extensively; however, their deployment has been slow due to their high cost and diverse architectures and functionalities. In addition, the provision of new services would be prolonged with this approach.

4.3 Standardization

There are no specific discussions with regard to the standardization of session overlay networks; however, some related technologies have been discussed at IETF (Internet Engineering Task Force).

In the PILC working group [29] and the WAP Forum [30], split TCP technologies have been discussed for wired-wireless-combined networks. Originally TCP was designed for a wired environment in the Internet, where congestion accounts for most of the packet losses. It is well known that the TCP throughput deteriorates in a wireless environment, where link errors cause non-congestion random losses. Thus, in split TCP technologies [21], [22], [33], [34], PEP [35]–[38] is used to connect split TCP sessions; one technique uses the standard TCP version for wired networks and the other uses new TCP variants, such as Wireless-profiled TCP [39], [40] for wireless networks.

PEPs are also used for other environments. For example, PEPs are used to enhance the TCP throughput in fast and long distance environments, such as satellite links and transcontinental links.

4.4 Relaying TCP Sessions

One of the fundamental technologies of session overlay networks is the relay of data streams between two (and possibly more) TCP sessions. They include the following:

(1) Session setup,
(2) Connecting two split TCP sessions,
(3) Connecting heterogeneous transports, and
(4) Reliability of PEPs.

(1) In order to set up and maintain a series of TCP sessions between an original sender and a final destination, several technologies, including (a) a decision whether to split the TCP session, (b) splicing split TCP sessions, and (c) maintenance of the session status, are involved.

(a) In order to decide whether to split a TCP session or maintain an end-to-end TCP session, several factors have to be considered. For example, when the CPU load of PEPs or the number of maintained TCP sessions exceeds a predetermined threshold, newly arriving sessions will not split and cannot be handled at the PEP. On the other hand, sessions of specific applications are split, while those of the others are not split. In other words, the sessions from specific servers or to specific hosts are split while those of the others are not split. In addition, once the split of a session is confirmed, the relay is also determined, i.e., using high-speed TCP versions, or low-priority TCP versions, etc.

(b) There exist several methods to split and splice TCP sessions. A typical method is to first set up a session between a sender and a PEP, followed by setting up another session between the PEP and the next PEP or a receiver. When the PEP snoops a TCP-SYN packet from the sender, it terminates the session setup and instead of sending a SYN-ACK packet the receiver, it sends this packet back to the sender. The PEP then sends out a new TCP-SYN packet to the receiver. This method seems simple and easy; however, the maintenance of a series of TCP sessions is a drawback. For example, there arises a problem when PEP finds malfunction of the succeeding PEPs and can not set up a session downwards. In this case, a session is already set up between a sender and the PEP, while successive sessions are not; thus, the sender and receiver become inconsistent in their session status.

Another method is to hold the session setup until a SYN-ACK packet is received from successive PEPs. In this case, as shown in Fig. 3, the PEP does not terminate the TCP-SYN packet, but passes it downwards [41]. At the same time, the PEP prepares and holds the session setup. Then, if it receives a SYN-ACK packet from its successor PEPs, the session setup is confirmed, otherwise, the PEP abandons the session setup.

(c) In order to maintain a series of split TCP sessions, the PEPs involved in these sessions have to consistently maintain the status of these sessions. For example, when a PEP has to abandon a session due to scarcity of resources or any other reason, it has to ensure that the other PEPs along with the sender and the receiver should also abandon the session. Further, when the sender or the receiver aborts the session without a formal teardown procedure, the PEPs involved in the sessions have to identify the abortion and abandon their session.

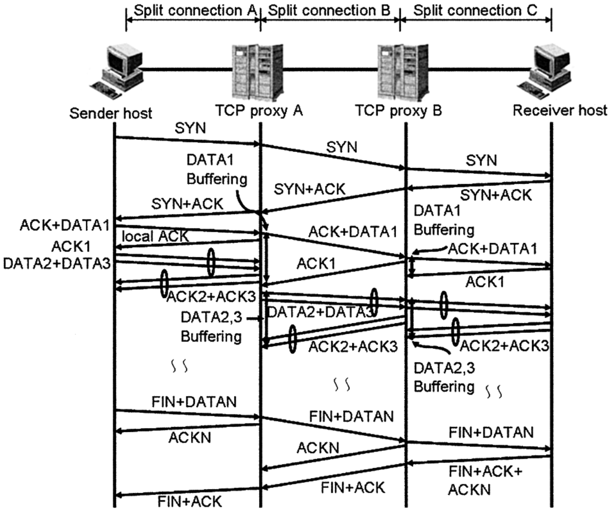


Fig. 3 Communication diagram between hosts and PEPs.

(2) In order to connect two split TCP sessions, several technologies including (a) conveyance of congestion information among split TCP sessions, (b) store-and-forward relay, and (c) high-speed protocol processing, are involved.

(a) PEPs must convey congestion information among the split TCP sessions, i.e., when a PEP detects congestion in a downward session, it has to slow down the upward session in order to balance the throughputs between them. When a downward session slows down due to congestion, the data stream queues at the PEP buffer. Further, when the buffer is full, the zero window size is advertised upward; this halts the transmission from the sender. In some cases, this behavior causes serious throughput degradation and several countermeasures have been proposed [41], [43]

(b) At the PEPs, the received segments may be transmitted as soon as they are received, or they might wait at the PEP till suitable transmission timing. For example, segments have to wait until unused bandwidth to the next PEP or end host is obtained in order to provide nonintrusive background transfers [44], [45].

(c) There are two methods of improving the protocol-processing performance. One method is to tune protocol stacks for splicing TCP sessions. The optimization of buffer structure to produce non-negligible performance improvement is discussed in [46]. The other method is the employment of hardwired TCP engines implemented in LSI macros [47], [70], TOEs (TCP offload engine) [28], [71], [72], and hardware PEP [49], [73]. In spite of the high performance of the hardwired engines, the flexibility of software implementations is very advantageous particularly since new variations of TCP algorithms are continuously added to new applications or new environments.

(3) Besides TCP, any arbitrary transport protocol can be relayed by TCP sessions. For example, any UDP session can be relayed by HTTP over a TCP session in order to traverse firewalls, which prohibits any direct UDP communication in default.

When end users begin using their own transport protocols that are not compatible with legacy protocols and could deteriorate other communications, any such protocol is relayed by the standard TCP at the edge of the network in order to protect the network from selfish resource occupation.

(4) The reliability of PEPs is another crucial issue. Since PEPs store "on-the-fly" segments, which are acknowledged by the senders, but not by the receivers, PEP failures cause unrecoverable segment loss that goes unnoticed by the senders. Thus, the receivers wait for these lost segments that are not retransmitted by the senders. In this situation, some applications may terminate the session and abandon the data received thus far; this causes inconvenience for users since

they have to wait for long periods for timeouts. In the worst case, applications may not notice the transport layer failure and this inconsistent status may cause false behaviors.

In order to avoid these situations, either (a) the recovery of lost segments or (b) the auto-termination of failed sessions, is required.

(a) The recovery of lost segments can be accomplished by the redundant operation of a PEP. A pair of PEPs—an active and a standby PEP—is operated redundantly. When the active PEP fails, the on-the-fly segments and session status are transferred to the standby PEP that takes over the operation. Since this redundant operation requires packet-by-packet synchronization of active and stand-by systems [48], realizing efficient synchronization mechanism is a significant challenge.

(b) The auto termination of failed sessions will be performed by the active PEPs around the failed PEPs. When a PEP identifies the failure of a neighboring PEP, it cannot recover the lost segments but it can at least reset the session in order to indicate the failure to the applications. This is performed by generating a TCP reset packet on behalf of the failed PEP [50].

4.5 Overlay Traffic Control Using New TCP Variants

Various overlay services can be provided by applying various traffic control algorithms between TCP PEPs. For example, PEPs connected to a very fast and long-distance link would apply high-speed versions of TCP to provide high-speed communication services without modifying end hosts. Also, low-priority versions of TCP would be used for background transfer services without modifying underlying physical networks. In the same way, new network services can be realized by relaying end-to-end sessions using enhanced transport protocols having specific behaviors.

Since such overlay services rely on appropriate enhancements to TCPs at PEPs, traffic control algorithms play a key role to determine service menus and service performances. Therefore, in this section, to explore potential overlay services, we focus on new TCP variants potentially available for overlay traffic control. There variants are mainly used for the following objectives; (1) higher efficiency, and (2) QoS control.

With regard to coexistence, some of these variants are compatible with the legacy versions of TCP, namely, TCP-Reno and its variations, while the others are not. A friendly version of TCP does not deteriorate coexisting legacy TCP sessions more than the one that uses legacy TCP versions. Therefore, one can use friendly TCP versions without considering the interference; and thus, overlay networks can be built independently of the underlay physical networks. On the other hand, overlay networks using incompatible variants should be managed by the managers of physical networks so that applications on both the overlay and underlay networks can appropriately share the network resources.

(1) There are a number of TCP-related proposals for increasing the efficiency in the utilization of path bandwidth; they are as follows: (a) Some of them are proposed for fast and long distance networks in which TCP sessions have to maintain very large congestion window size, (b) Some are proposed for wireless networks with non-negligible random packet losses; these losses should not be considered as congestion signals. (c) Some are proposed to prevent throughput degradation caused by the burstiness of sent packets.

(a) Because of the window-based flow control nature of TCP, the maximum throughput of a TCP session is limited by its maximum window size or its congestion window size. For example, when the maximum window size is limited to 64 KB, which is the default value for major operation systems, the TCP throughput is limited to 25.6 Mbps on a network path with a 20 ms round-trip propagation delay. In order to avoid time constraints, either an increase in the socket buffer sizes of both the sender and receiver or a reduction in the round propagation delay is required. A TCP PEP in the middle of a network path is useful for both these requirements.

To increase the efficiency of TCP sessions with a limited throughput due to the by congestion window size, a number of TCP variants with improved congestion window controls, have been proposed to maintain a large congestion window size. High speed TCP [51] and FAST TCP [52] are major examples of these TCP variants; however, it has been demonstrated that they are not compatible with TCP-Reno flows. In other words, high speed TCP deteriorates the throughputs of coexisting TCP-Reno sessions and FAST TCP suffers from low throughput. Therefore, a TCP-Reno-friendly version of these protocols have recently been proposed [53], [54]. TCP-AdaptiveReno (TCP-AR) achieves this goal by dynamically adjusting congestion window parameters based on the congestion measurement, whereas high speed TCP adjusts these parameters based on the congestion window size. This behavior ensures that TCP-AR flows fairly share the bandwidth with TCP-Reno flows when a network is already utilized to its fullest capacity. However, the flows obtain higher throughput when the network is underutilized as shown in Fig. 4.

(b) A number of TCP variants are proposed for wireless networks with non-negligible random packet losses. Since, in wireless network, packet losses that should not be counted as a congestion signal occur due to link errors, TCP must recognize whether the loss indicates congestion or not. Some of the variants utilize link layer information provided by wireless network equipment, while the others utilize their own estimation of packet losses [55].

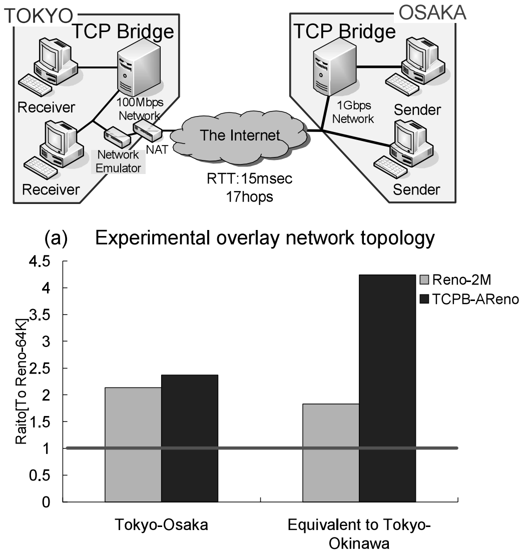
(c) In order to reduce the burstiness of sending packets, paced TCPs are proposed [56], [57]. In the slow start of congestion window, in particular, packets are often sent in bulk; this will cause multiple packet losses at a router buffer. As a result, TCP sessions suffer from retransmission timeouts and correspondingly severe throughput degradation. In order to space the sent packets, several approaches are proposed; these include the ones that use OS timers, special NIC hardware, and dummy ether frames [58].

(2) For realizing QoS control on session overlay net-

works, variants of TCPs with differentiated congestion controls are used. For this purpose, low-priority TCPs such as TCP-LP [59], TCP-Nice [60], and TCP-Westwood Low-Priority [61] have been proposed.

In Fig. 5, an experimental result is shown for an overlay network using TCP-Westwood Low-Priority on a PEP. This figure shows that a background traffic using TCPW-LP fully utilize the link capacity, whereas it defers to foreground traffic using TCP-Reno when they coexists. Thus, this example indicates that prioritized QoS can be provided on the session overlay network without requiring specific router supports.

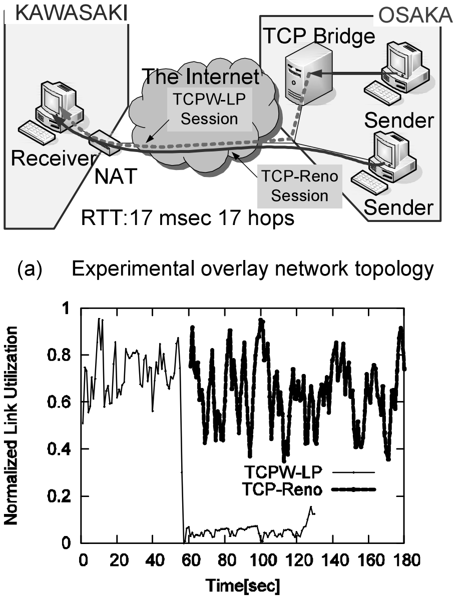
Mul-TCP [62] is also proposed to differentiate the TCP throughput. A Mul-TCP session has specialized congestion control parameters such that it achieves a throughput N times larger than those of normal TCP-Reno flows.



(a) Experimental overlay network topology

(a) Throughput of foreground/background flows

Fig. 4 Overlay network for high-speed communication.



(a) Experimental overlay network topology

(a) Throughput of foreground/background flows

Fig. 5 Overlay network for QoS.

TCP-BC [63] is proposed in order to guarantee minimum bandwidth. A TCP-BC session attempts to maintain a congestion window size that is large enough to maintain the specified minimum guaranteed bandwidth. It also tries to avoid severe congestion by monitoring the congestion level via RTT measurement.

4.6 Multi-Path Communication

Two or more simultaneous sessions, possible on different physical paths, may be set up between a sender and receiver, or between neighboring PEPs in order to increase efficiency and reliability.

GridFTP [64] is proposed for high-speed file transfer. It divides a file into several fragments and sets up multiple TCP sessions to transmit these fragments simultaneously; thus, theoretically, Gri pFTP can obtain a throughput N times larger than the total throughput using N multiple sessions.

A similar approach is proposed for general TCP transfers and not only file transfers, by dispatching TCP segments rather than file fragments to PEPs [65], [66]. In order to achieve high speed and reliable transmission using multiple, and possibly unstable paths, this scheme optimizes the segment distribution so that the sorting of segments at the merging point will not halt transmission due to buffer overflows.

5. Current and Future Overlay Network Issues for QoS Control

Finally, we address overlay network issues that may pose problems in the future in order to solve these issues. The issues listed below are not limited only in QoS control but also are expandable to general overlay networks.

(1) Platform for various types of overlay networks
In the absence of unified platform, interests of overlay networks coexisting over an underlay network may conflict and the overlay network acts or is operated as if it exists alone or try to selfish to form optimally for it. This may result in suboptimal performance or a collapse of all the networks. Thus, we have to investigate the interference between mutual overlay networks (horizontal interaction) as well as that between overlay and underlay networks (vertical interaction) [15], [16], [67]–[69]. The important functions of a platform are stability, reliability, robustness, etc.

Since several selfish overlay routings work without serious problems [67], we need to further discuss the necessity of a platform.

(2) Deployment scenario
A minimum cost of the overlay network depends on the target service. Management complexity might increase the cost. A tradeoff between increased cost and the advantages of management is not evident. Thus, a new network the-

ory for overlay networks with a new performance metric is needed.

(3) Scalability
As typically observed in the flooding problem of the P2P search, unless an overlay network has sufficient scalability, which is the most important and attractive feature, such overlay network may disappear before being widely deployed.

(4) Function
Does an overlay network provide functions that are originally provided by an underlay network? When overlay networks provide routing functions and logical address assignment functions, current ISPs, which provide underlay networks, will exist only for bit transfer.

On the other hand, the routing or session QoS control of overlay networks generally exhibits lesser accuracy than underlay network. A comparison of their accuracies or the quantitative accuracy evaluation of the QoS control of overlay networks should be investigated in order to meet the service requirements.

(5) Dependability
Can overlay networks be dependable? IP networks have evolved in terms of features such as diagnosis, reliability, failure recovery, and dimensioning. The application of such technologies to overlay networks needs to be discussed.

(6) Efficiency
Some overlay networks are interested in QoS improvement and try to efficiently use the underlay network. Others, however, is not interested in efficiency and may waist network resources.

There is a contradiction between network operators and end users for a tariff for bit transfer. The network operators pay based on a distance and on a traffic volume. It, however, is common for the end users that charge is not based on a distance. Moreover, flat rate for traffic volume is not extraordinary. A routing overlay network which end users establish could improve QoS without increasing traffic itself. This results in higher efficiency in terms of network resource consumption. On the other hand, an overlay network formed by end users may not be interested in efficiency, if a P2P file sharing application or redundant transfer application such technologies are running in the network. Thus, such application seems selfish from network operation point of view.

6. Conclusion

We have described overlay network that have recently attracted considerable attention; they have the following advantages: development of a new service in a short duration and start with a low cost. The necessity of overlay networks as a disruptive innovation platform was described, and the classification of various overlay networks, particularly according to the QoS feature, was attempted. In order to realize QoS control, it is considered that routing overlay and session overlay are promising solutions. In particular, session and overlay networks are explained in detail since new TCP protocols instead of the current TCP protocols that control congestion in the Internet can be used for QoS within overlay networks. Finally, several issues that need to be solved with regard to overlay networks are listed. We conclude that many issues such as scalability still need further research and development although overlay networks have many attractive features and possess the potential to become a platform for the deployment of new services.

Acknowledgements

For invaluable discussions about overall technologies, the authors deeply appreciate each member of overlay network research team both in NEC Corporation and Osaka University. This work was supported by the Strategic Information and Communications R&D Promotion Programme (SCOPE) of the Ministry of Internal Affairs and Communications of Japan.

References

[1] http://www.napster.com/
[2] http://www.gnutella.com/
[3] http://www.kazaa.com/
[4] D.S. Isenberg, "The rise of the stupid network," Computer Telephony, pp.16–26, Aug. 1997.
[5] R. Friend, "Making the gigabit IPsecVPN architecture secure," Computer, vol.37, no.6, pp.54–60, June 2004.
[6] S. Sen and J. Wang, "Analyzing peer-to-peer traffic across large networks," IEEE/ACM Trans. Netw., vol.12, no.2, pp.219–232, April 2004.
[7] H. Erikson, "MBONE: The multicast backbone," Commun. ACM, vol.37, no.8, pp.54–60, Aug. 1994.
[8] W. Chou, "Inside SSL: The secure sockets layer protocol," IT Professional, vol.4, no.4, pp.47–52, July/Aug. 2002.
[9] http://www.planet-lab.org/
[10] Z. Li and P. Mohapatra, "QRON: QoS-aware routing in overlay networks," IEEE J. Sel. Areas Commun., vol.22, no.1, pp.29–40, Jan. 2004.
[11] Z. Duan, Z.L. Zhang, and Y.T. Hou, "Service overlay networks: SLAs, QoS, and bandwidth provisioning," IEEE/ACM Trans. Netw., vol.11, no.6, pp.870–883, Dec. 2003.
[12] T. Murase, H. Shimonishi, and Y. Hasegawa, "TCP overlay network architecture," Proc. Commun. Conf. IEICE'02, B-7-49, Sept. 2002.
[13] L. Subramanian, I. Stoica, H. Balakrishnan, and R. Katz, "OverQoS: Offering internet QoS using overlays," HotNets-I, 2002.
[14] D. Andersen, H. Balakrishnan, F. Kaashoek, and R. Morris, "Resilient overlay networks," SOSP2001, pp.131–145, Oct. 2001.
[15] M. Kwon and S. Fahmy, "Toward coopertive inter-overlay networking," IEEE ICNP, Nov. 2003.
[16] A. Nakao, L. Peterson, and A. Bavier, "A routing underlay for overlay networks," ACM SIGCOMM, pp.11–18, Aug. 2003.
[17] N. Brownlee and K.C. Claffy, "Understanding Internet traffic streams: Dragonflies and tortoises," IEEE Commun. Mag., vol.40, no.10, pp.110–117, Oct. 2002.
[18] N. Enomoto, "A secure and easy remote access technology," APSITT 2005, pp.364–368, Nov. 2005.
[19] B.P. Lee, L. Jacob, W.K.G. Seah, and A.L. Ababda, "Avoiding congestion collapse on the Internet using TCP tunnels," Comput. Netw., vol.39, no.2, pp.207–219, Dec. 2002.
[20] I. Clarke, O. Sandberg, B. Wiley, and T.W. Hong, "Freenet: A distributed anonymous information storage and retrieval system," Workshop on Design Issues in Anonymity and Unobservability, pp.311–320, July 2000.

[21] A. Vetro, C. Christopoulos, and S. Huifang, "Video transcoding architectures and techniques: An overview," IEEE Signal Process. Mag., vol.20, no.2, pp.18–29, March 2003.

[22] Y. Zhu, B. Li, and J Guo, "Multicast with network coding in application-layer overlay networks," IEEE J. Sel. Areas Commun., vol.22, no.1, pp.107–120, Jan. 2004. Digital Object Identifier 10.1109/JSAC.2003.818801.

[23] N. Feamster, D.G. Andersen, H. Balakrishnan, and F. Kaashoek, "Measuring the effects of Internet path faults on reactive routing," ACM SIGMETRICS, vol.31, no.1, pp.126–137, June 2003.

[24] S. Banerjee, T.G. Griffin, and M. Pias, "The interdomain connectivity of PlanetLab nodes," 5th Anuual Passive & Active Measurement Workshop PAM2004, vol.3015, pp.73–82, April 2004.

[25] S. Rewaskar and J. Kaur, "Testing the scalability of overlay routing infrastructures," 5th Anuual Passive & Active Measurement Workshop PAM2004, vol.3015, pp.33–42, April 2004.

[26] M. Uchida, S. Kamei, and R. Kawahara, "Evaluation of QoS-aware routing in overlay network," IEICE Technical Report, IN2005-31, July 2005.

[27] J. Touch and S. Hotz, "The X-bone," Proc. Third Global Internet Mini-Conference at Globecom'98, pp.59–68, Nov. 1998.

[28] B. Chun, D. Culler, T. Roscoe, A. Bavier, L. Peterson, M. Wawrzoniak, and M. Bowman, "PlanetLab: An overlay testbed for broad-coverage services," ACM Comput. Commun. Rev., vol.33, no.3, pp.3–12, July 2003.

[29] The Internet Engineering Task Force, http://www.ietf.org/

[30] WAP Forum, http://www.wapforum.org/

[31] H. Balakrishnan, V.N. Padmanabhan, and R. Katz, "Improving reliable transport and handoff performance in cellular wireless networks," ACM Wirel. Netw., vol.1, no.4, pp.469–481, Dec. 1995.

[32] H. Balakrishnan, V.N. Padmanabhan, S. Seshan, and R. Katz, "Comparison of mechanisms for improving TCP performance over wireless links," ACM SIGCOMM'96, pp.256–269, Palo Alto, CA, Aug. 1996.

[33] A. Bakre and B.R. Badrinath, "I-TCP: Indirect TCP for mobile hosts," DCS-TR-314, Rutgers University, Oct. 1994.

[34] R. Yavatkar and N. Bhagawat, "Improving end-to-end performance of tcp over mobile internetworks," IEEE Workshop on Mobile Computing Systems and Applications, pp.146–152, Dec. 1994.

[35] S. Dawkins, G. Montenegro, M. Kojo, V. Magret, and N. Vaidya, "End-to-end performance implications of links with errors," IETF, Internet Draft, work in progress Nov. 2000.

[36] J. Border, M. Kojo, J. Griner, G. Montenegro, and Z. Shelby, "Performance enhancing proxies," IETF, Internet Draft, work in progress Nov. 2000.

[37] G. Montenegro, S. Dawkins, M. Kojo, V. Magret, and N. Vaidya, "Long thin networks," IETF, RFC2757, Jan. 2000.

[38] J. Border, M. Kojo, J. Griner, G. Montenegro, and Z. Shelby, "Performance enhancing proxies intended to mitigate link-related degradations," IETF, RFC3135.

[39] WAP Forum, Wireless Profiled TCP, Version 31, March 2001.

[40] H. Inamura, G. Montenegro, R. Ludwig, A. Gurtov, and F. Khafizov, "TCP over second (2.5G) and third (3G) generation wireless networks," IETF, RFC 3481, Feb. 2003.

[41] I. Maki, G. Hasegawa, M. Murata, and T. Murase, "Performance analysis and improvement of TCP proxy mechanism in TCP overlay networks," IEEE Int. Conf. Commun., vol.1, pp.184–190, 2005.

[42] K. Yamanegi, T. Hama, G. Hasegawa, M. Murata, H. Shimonishi, and T. Murase, "Implementation experiments of TCP proxy mechanism," Information and Telecommunication Technologies 2005, Proceeding 6th Asia-Pacific Symposium, pp.17–22, 2005.

[43] Y. Yamasaki, T. Murase, G. Hasegawa, and M. Murata, "Congestion prevention buffer management in TCP proxy," IEICE Technical Report, IN2003-136, Dec. 2003.

[44] H. Shimonishi, T. Hama, and T. Murase, "Improving efficiency-friendliness tradeoffs of TCP congestion control algorithm," IEICE Technical Report, IN2004-266, March 2005.

[45] T. Hama, K. Yamanegi, H. Shimonishi, T. Murase, G. Hasegawa, and M. Murata, "Experimental study of a TCP-adaptive Reno for high speed networks," IEICE Technical Report, IN2004-267, March 2005.

[46] Y. Hasegawa and T. Murase, "High speed protocol processing methods for TCP proxy and performance evaluations," Proc. Conf. IEICE '03, B-7-5, Sept. 2003.

[47] K. Murata, T. Takeoka, and K. Abe, "Implementation of a TCP/IPv6 protocol stack on FPGA and its evaluation," Proc. 10th FPGA/PLD Design Conference, pp.171–176, Jan. 2003.

[48] M. Marwah, S. Mishra, and C. Fetzer, "TCP server fault tolerance using connection migration to a backup server," Proc. 2003 Int. Conf. Dependable Syst. Netw. (DSN'03), pp.373–382, 2003.

[49] M. Nishihara, S. Kamiya, T. Hayashi, H. Ueno, T. Domeki, and T. Kanoh, "Broadband service gateway platform for readily available and reliable business applications and services," NEC J. Adv. Technol., vol.1, no.2, pp.154–160, Spring 2004.

[50] I. Yamaguchi, H. Shimonishi, and T. Murase, "A study for a recovery from TCP proxy failures," IEICE Technical Report, NS2004-257, March 2005.

[51] S. Floyd, "HighSpeed TCP for large congestion windows," RFC 3649, IETF, Dec. 2003.

[52] C. Jin, D. Wei, and S. Low, "FAST TCP: Motivation, architecture, algorithms, performance," Proc. IEEE INFOCOM, vol.4, pp.2490–2501, March 2004.

[53] Z. Zhang, G. Hasegawa, and M. Murata, "Performance analysis and improvement of HighSpeed TCP with TailDrop/RED routers," International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunications Systems (MASCOTS) 2004, pp.505–512, Oct. 2004.

[54] H. Simonishi and T. Murase, "Improving efficiency-friendliness tradeoffs of TCP congestion control algorithm," Global Telecommunications Conference, 2005. GLOBECOM apos;05. IEEE, vol.1, no.28, p.5, Nov./Dec. 2005.

[55] C. Casetti, M. Gerla, S. Mascolo, M.Y. Sanadidi, and R. Wang, "TCP Westwood: Congestion window control using bandwidth estimation," Conf. Rec. IEEE GLOBECOM 2001, vol.3, pp.1698–1702, 2001.

[56] J. Kulik, R. Coulter, D. Rockwell, and C. Partridge, "Paced TCP for high delay-bandwidth networks," IEEE Workshop on Satellite Based Information Systems, Rio de Janeiro, Dec. 1999.

[57] http://www.cs.washington.edu/homes/tom/pubs/pacing.pdf

[58] R. Takano, T. Kudoh, Y. Kodama, M. Matsuda, H. Tezuka, and Y. Ishikawa, "Design and evaluation of precise software pacing mechanisms for fast long-distance networks," PFLDnet05, 2005.

[59] A. Venkataramani, R. Kokku, and M. Dahlin, "TCP-nice: A mechanism for background transfers," ACM SIGOPS Operating Systems Review archive, vol.36, Issue SI (Winter 2002) table of contents, OSDI'02: Proc. 5th Symposium on Operating Systems design and implementation, pp.329–344, Dec. 2002.

[60] A. Kuzmanovic and E. Knightly, "TCP-LP: A distributed algorithm for low priority data transfer," Proc. IEEE INFOCOM 2003, vol.3, pp.1691–1701, 2003.

[61] H. Shimonishi, M.Y. Sanadidi, and M. Gerla, "Service differentiation at transport layer via TCP Westwood Low-Priority (TCPW-LP)," Proc. Computers and Communications, ISCC 2004, vol.2, pp.804–809, June 2004.

[62] J. Crowcroft and P. Oechslin, "Differentiated end-to-end Internet services using a weighted proportional fair sharing TCP," Comput. Commun. Rev., vol.28, no.3, pp.53–67, 1998.

[63] H. Shimonishi and T. Murase, "A TCP congestion control algorithm for bandwidth control," Proc. Commun. Conf. IEICE'05, B-7-36, Sept. 2005.

[64] B. Allcock, J. Bester, J. Bresnahan, A.L. Chervenak, I. Foster, C. Kesselman, S. Meder, V. Nefedova, D. Quesnet, and S. Tuecke, "Secure, efficient data transport and replica management for high-performance data-intensive computing," Proc. 18th IEEE Symp.

Mass Storage Syst. Technol., p.13, 2001.

[65] K. Rojviboonchai and H. Aida, "An evaluation of multi-path transmission control protocol (M/TCP) with robust acknowledgement schemes," Internet Conference 2002, Oct. 2002.

[66] Y. Hasegawa, I. Yamaguchi, T. Hama, H. Shimonishi, and T. Murase, "Improved data distribution for multipath TCP communication," Global Telecommunications Conference 2005, GLOBECOM'05 IEEE, vol.1, p.5, Nov./Dec. 2005.

[67] L. Qiu, Y.R. Yang, Y. Zhang, and S. Shenker, "On selfish routing in Internet-like environments," Proc. ACM SIGCOMM, pp.151–162, Aug. 2003.

[68] M. Seahadri and R.H. Katz, "Dynamics of simultaneous overlay network routing," UCB EECS Report UCB//CSD-03-1291, Nov. 2003.

[69] N. Wakamiya and M. Murata, "Toward overlay network symbiosis," 5th IEEE International Peer-to-Peer Computing, pp.154–155, Aug. 2005.

[70] "Treck TCP/IP," http://www.treck.com/pdf/TCP.pdf

[71] Y. Hasegawa, H. Shimonishi, and T. Murase, "Performance evaluation of hardware TCP offload engines and analysis of their TCP's behavior," ITRC-NGN, JSPS 163rd Committee on Internet Technology, May 2003.

[72] P. Balaji, W. Feng, Q. Gao, R. Noronha, W. Yu, and D.K. Panda, "Head-to-TOE evaluation of high-performance sockets over protocol offload engines," IEEE Cluster 2005, Sept. 2005.

[73] M. Yasuda and K. Yamada, "Development and performance evaluation of high performance hardware TCP engine," Proc. IEICE Gen. Conf.'05, B-7-57, March 2005.

[74] T. Murase, Traffic control and architecture for high-quality and high-speed Internet, Ph.D. Thesis, Graduate School of Information Science and Technology, Osaka University, March 2004.

[75] http://www.jgn.nict.go.jp/

Tutomu Murase was born in Kyoto, Japan in 1961. He received his M.E. degree from Graduate School of Engineering Science, Osaka University, Japan, in 1986. He also received his Ph.D. degree from Graduate School of Information Science and Technology, Osaka University in 2004. He joined NEC Corporation in 1986 and has been engaged in research on traffic management for high-quality and high-speed internet. His current interests include transport and session layer traffic control, network traffic traceability and network security. He was a secretary and has been a member of steering committee of Communication Quality Technical Group in IEICE. He is also a member of steering committee of Information Network Technical Group in IEICE. He is a vice chair person of Next Generation Network working group in JSPS 163rd Committee on Internet Technology (ITRC).

Hideyuki Shimonishi received his M.E. and Ph.D. degrees from Graduate School of Engineering Science, Osaka University, Osaka, Japan, in 1996 and 2002, respectively. He joined NEC Corporation in 1996 and has been engaged in research on traffic management in high-speed networks, switch and router architectures including cell/packet scheduling algorithms and buffer management mechanisms, and traffic control protocols. He was a visiting scholar at Computer Science Department, University of California at Los Angeles, to study next generation transport protocols. Dr. Shimonishi is a member of ACM.

Masayuki Murata received the M.E. and D.E. degrees in Information and Computer Sciences from Osaka University, Japan, in 1984 and 1988, respectively. In April, 1984, he joined Tokyo Research Laboratory, IBM Japan, as a Researcher. From 14 September 1987 to January 1989, he was an Assistant Professor with Computation Center, Osaka University. In February 1989, he moved to the Department of Information and Computer Sciences, Faculty of Engineering Science, Osaka University. From 1992 to 1999, he was an Associate Professor in the Graduate School of Engineering Science, Osaka University, and from April 1999, he has been a Professor of Osaka University. He moved to Advanced Networked Environment Division, Cyber-media Center, Osaka University in April 2000. In March 2004, he moved to Graduate School of Information Science and Technology, Osaka University. He has more than four hundred papers of international and domestic journals and conferences. His research interests include computer communication networks, performance modeling and evaluation. He is a member of IEEE, ACM, The Internet Society, and IPSJ.