PAPER

New TCP Congestion Control Schemes for Multimodal Mobile Hosts

Kazuya TSUKAMOTO[†a)], ***Student Member***, **Yutaka FUKUDA**[††], ***Member***, **Yoshiaki HORI**[†††], ***Nonmember***, ***and*** **Yuji OIE**[†], ***Fellow***

SUMMARY Two congestion control schemes designed specifically to handle changes in the datalink interface of a mobile host are presented. The future mobile environment is expected to involve multimode connectivity to the Internet and dynamic switching of the connection mode depending on network conditions. The conventional Transmission Control Protocol (TCP), however, is unable to maintain stable and efficient throughput across such interface changes. The two main issues are the handling of the change in host Internet Protocol (IP) address, and the reliability and continuity of TCP flow when the datalink interface changes. Although existing architectures addressing the first issue have already been proposed, **the problem of congestion control remains**. In this paper, considering a large change in bandwidth when the datalink interface changes, two new schemes to address these issues are proposed. The first scheme, Immediate Expiration of Timeout Timer, detects interface changes and begins retransmission immediately without waiting for a retransmission timeout as in existing architectures. The second scheme, Bandwidth-Aware Slow Start Threshold, detects the interface change and estimates the new bandwidth so as to set an appropriate slow start threshold for retransmission. Through simulations, the proposed schemes are demonstrated to provide marked improvements in performance over existing architectures.

key words: *multimodal, vertical hand-off, TCP congestion control*

1. Introduction

A wide range of datalink technologies have been introduced in recent years to provide access to the Internet backbone. In particular, wireless datalink technologies have received much attention as a means of obtaining mobile access to the Internet. In the upcoming third-generation digital mobile radio standard, the Universal Mobile Telecommunication System (UMTS) will provide packet-switched bearer services with capacities of up to 2 Mb/s, available over a wide area. For mobile computing, IEEE802.11a/b Wireless Local Area Networks (WLANs) have become a widely implemented technology that can provide high-speed access to the Internet within a limited area.

Wireless LANs and cellular networks are complementary technologies. Wireless LANs provide relatively inexpensive, and broadband connectivity, yet offer only limited coverage. In contrast, cellular networks provide wide-area coverage, yet are currently capable of only narrowband connectivity at relatively high expense. However, while only cellular services are currently available over a wide area, it is expected that Wireless LAN hotspot services will be available at many public spaces, including coffee shops and waiting areas for public transport. Naturally, integrating Wireless LANs and cellular networks will provide service to users who need both high-speed wireless access as well as anytime-anywhere mobile connectivity [1].

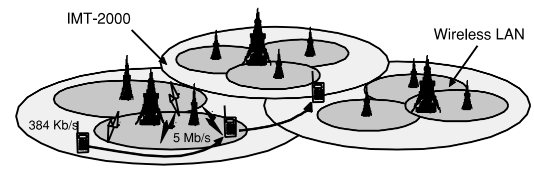


Fig. 1 Future mobile environment.

In order to achieve this level of integration, mobile hosts are likely to be designed to be capable of handling multiple datalink interfaces. It will therefore become necessary for the mobile host to automatically select the available datalink interface based on the transmission capacity, signal intensity, bit error condition, and other factors, as shown in Fig. 1. The mobile host will be able to convey information about the datalink interface state (such as link-up or link-down) from the link layer to the transport layer through a cross-layer mechanism [2] such as an interlayer signaling pipe (ISP) [3]. The change from one access network to another is commonly referred to as a **hand-off**. In particular, the type of hand-off between the different wireless network technologies is referred to as a "**vertical hand-off**" [4]. When the vertical hand-off occurs, the transmission capacity such as bandwidth and transmission delay can be changed drastically. In our paper, the vertical hand-off will be considered as hand-off from now on.

This introduces the concept of multimodal communication for mobile devices, but raises the concern of how to ensure efficient Transmission Control Protocol (TCP) flow across changing interfaces. The most important issues are potential changes in the Internet Protocol (IP) address of the mobile host when the interface mode is switched, and the inability for TCP congestion control to handle interface hand-offs. In the former issue, as a TCP flow is identified uniquely by the four parameters of source IP address, source

Manuscript received July 11, 2005.
Manuscript revised October 15, 2005.

[†]The authors are with the Dept. of Computer Science and Electronics, Kyushu Institute of Technology, Iizuka-shi, 820-8502 Japan.
[††]The author is with Information Science Center, Kyushu Institute of Technology, Kitakyushu-shi, 804-8550 Japan.
[†††]The author is with Dept. of Computer Science and Communication Engineering, Kyushu University, Fukuoka-shi, 812–8581 Japan.
a) E-mail: kazuya@infonet.cse.kyutech.ac.jp

DOI: 10.1093/ietcom/e89–b.6.1825

port, destination IP address, and destination port, disconnection will occur because the TCP flow will be recognized as a different flow if the host IP address is changed. In the latter issue, the time required to switch from one datalink interface to another may exceed the timeout for current TCP congestion control, which may also be unable to accommodate the potentially large change in available bandwidth. A number of existing architectures have already been proposed to resolve the problem associated with the change in IP address. In contrast, although some papers including [5]–[7] focus on the TCP congestion control in ongoing connection, **none**[†] of them focus on the TCP congestion control in order to address the drastic change in transmission capacity of a wireless link around the hand-off.

There is much active research on transport protocols at present, and one of the major developments has been the modification of window congestion control using the end-to-end measurement of the available bandwidth as feedback. Hoe [5] proposed that the initial slow start threshold (**ssthresh**) value should be set to the estimated available Bandwidth-Delay product to improve start-up behavior. In TCP-Westwood [6], [7], the window is controlled based on end-to-end rate estimation of the available bandwidth in order to prevent performance degradation on wireless networks with lossy links. In addition, several proposed transport protocols, including XCP [9], QuickStart [10], and HighSpeed TCP [11] have been specially designed for use in high-speed wide-area links, and all can utilize the large available bandwidth. TCP-Peach [12] has been specially designed for use in satellite networks to account for the inherently long propagation delays and relatively high bit error rates. TCP-Peach uses low-priority dummy packets to probe the available bandwidth, allowing the available bandwidth to be utilized more effectively.

This paper presents two new congestion control schemes that can accommodate potentially large changes in available bandwidth and that offer improved performance over conventional TCP when considering changes in mobile datalink interfaces.

2. Existing Technologies for Multimodal Mobile Hosts

In this section, we review two existing schemes required specially for effective and continuous communication in the future mobile environment: (I) Mobility Management scheme, and (II) Congestion Control and Bandwidth Estimation scheme.

2.1 Mobility Management Scheme

As stated in the introduction, a number of technologies have been proposed to handle host IP address changes. One example is Mobile IP (v4 and v6) [13], [14], which has matured rapidly to a stage where it is being proposed as a standard by the Internet Engineering Task Force (IETF) to support mobility on the Internet. Furthermore, some enhanced protocols for Mobile IP such as Fast Handovers for Mobile IPv6 (FMIP) [15] and Hierarchical Mobile IPv6 (HMIP) [16] have been proposed to improve the performance around the hand-off. This Mobile IP technology provides transparent support for host mobility by concealing changes in mobile host IP addresses from the upper layer (higher than the transport layer). Mobile IP deploys a home agent (HA) that intercepts packets destined for a mobile host and delivers the packets to the mobile host via a foreign agent (FA) in the foreign network. The mobile host informs the HA and the corresponding host (only in Mobile IPv6) of its new IP address by sending Binding Update packet, allowing continuous delivery of datagrams even when the IP address of the mobile host changes.

Another proposed technology is a Migrate TCP [17], which eliminates the dependence of TCP connection upon the host IP address, thereby preserving the TCP connection in the event that the IP address of the mobile hosts changes. This approach, proposed by Snoeren et al., adds two new TCP options; a migrate option, and a migrate-permitted option, to the current TCP. This architecture preserves a TCP connection by sending packets that contain these two TCP options from the mobile host to the corresponding host.

Both TCP over Mobile IP and Migrate TCP enable continuous communication thanks to mobility management described above even if the host IP address changes. These TCPs with conventional congestion control will be referred to as TCP with mobility management or in short TCP/mm.

2.2 Congestion Control and Bandwidth Estimation Scheme

TCP/mm uses the conventional congestion control, thereby raising the following two issues : **(I) Communication interruption due to retransmission timer timeout.** The corresponding host waits for ACK signals for segments lost during hand-off. Therefore, the host can not begin transmitting new packets until the retransmission timeout timer expires. **(II) Significant packet loss after the TCP communication restarts.** Current TCP congestion control can not adapt quickly to the new network condition after reconnection, resulting in multiple packet losses after the hand-off. Some enhanced protocols for Mobile IP including FMIP [15] and HMIP [16] described in the previous subsection have been proposed to achieve the seamless hand-off. More specifically, these technologies can avoid the occurrence of loss of packets during the hand-off, i.e., the first issue (communication interruption) can be solved. However, they use the conventional congestion control like TCP/mm. Therefore, the second issue still remains, that is, significant packet losses may occur after the hand-off even when the enhanced

[†]There were no existing papers which focus on the TCP congestion control considering the change in bandwidth and transmission delay after the hand-off (vertical hand-off) until we presented the preliminary version of our paper (approximately 5 pages) at IEEE VTC-spring in 2002. At present, some papers including [8] have been proposed the TCP congestion control method considering the drastic change of a transmission capacity around hand-off.

protocols are employed.

In contrast, Freeze TCP [18] entails a new congestion (flow) control scheme that considers mobility. In this scheme, whenever the receiver detects a disconnection, an **ACK signal** is sent back to the sender with a zero advertised window size (*awnd*). Upon seeing an *awnd* of zero, the sender freezes all retransmission timers and interrupts transmission until the sender receives an ACK signal with a non-zero *awnd*. After the interruption, the sender can restart communication using a window size equal to the value of *cwnd* before disconnection. This provides efficient communication even when there is a considerable interruption before reconnection. However, although this scheme achieves the desired result by preventing the retransmission timeout timer from expiring where a hand-off occurs between two networks employing the same technology (**intrasystem hand-off**), it does not consider the possibility that the bandwidth/latency changes after reconnection (**intersystem hand-off**). Therefore, significant packet loss may occur after the TCP communication restarts.

When multiple packets are lost, TCP cannot restart the transmission until the retransmission of all the packet losses is fully completed. In particular, NewReno TCP can retransmit only one packet per one RTT. Therefore, in such a case, the communication will cease for quite long period, thereby resulting in the significant degradation of TCP performance. That is, the TCP performance can be affected drastically by the multiple packet losses. So, in our paper, avoiding performance degradation due to multiple packet losses around hand-off becomes a critical issue.

To achieve efficient and continuous communication during or just after the hand-off period, setting of the transmission window based on an estimation of the new bandwidth, in addition to the mobility management scheme, is essential.

Several bandwidth estimation schemes for TCP congestion control have been proposed to utilize the available bandwidth more effectively and fairly. Here, we review some of them, pointing out their characteristics.

Packet pair scheme [19] measures the time spacing between received ACKs at the sender. A sample of the ACK bandwidth, i.e., the bandwidth promised by the ACK to the sender, is obtained by dividing the amount of acknowledged bytes by the interarrival time between consecutive ACKs. The method proposed by Hoe [5] uses three samples of the ACK bandwidth, and sets **quickly** the first value of *ssthresh* in the start-up period.

In TCP-Westwood [6], [7] and TIBET [20], some filtering techniques using the previous estimated value can be added to the sample sequence to smooth fast variations, and to reduce the impact of random losses. However, although these technologies achieve the desired performance on wireless link, the most important issues are potential large changes in available bandwidth when the interface is switched. Therefore, after the hand-off, these schemes based on some value obtained through **preceding estimation** can lead to the **inaccurate estimation** and their estimation takes **more than a few second** to converge to the actual available bandwidth.

Considering the above discussion, the bandwidth estimation method requires the **quickness** rather than the **accuracy**, whenever the interface is changed. In this paper, we focus on the **packet pair** scheme [19] as the bandwidth estimation scheme.

3. Proposed Technologies

In this paper, two new schemes are proposed to address two issues related to mobile TCP/IP connectivity as described in the previous section. The "Immediate Expiration of Timeout Timer" scheme detects the change in interface as a trigger to restart transmission, while the "Bandwidth-Aware Slow Start Threshold" scheme approaches the problem by detecting the interface change as a trigger for adapting its congestion control to the new network conditions, setting the *ssthresh* value based on an estimation of the new bandwidth.

The two proposed schemes are implemented assuming TCP/mm technologies such as Mobile IP and Migrate TCP for datalink interface changes, thereby preserving the TCP connection. These TCP/mm technologies, however, do not consider the change in bandwidth and delay after the hand-off. Therefore, in order to adapt the TCP congestion control to the network conditions after hand-off, these proposed schemes require an architecture (like the ISP [3] as described in the Introduction) that allows the event signal in the **network layer**, such as the change in IP address, to be accessible by the **transport layer** through a cross-layer mechanism [2]. In this paper, we suppose that the benefit of introducing of cross-layer approach [2] is greater than its cost paid for benefits [21], and thus employ the cross-layer approach to achieve the interaction between these layers.

3.1 Immediate Expiration of Timeout Timer

On detecting a change in the datalink interface, the sender restarts data transmission immediately using a slow start instead of waiting for the TCP retransmission timeout timer to expire. To achieve this quick restart, this scheme forces the timer to expire immediately, thereby avoiding unnecessary communication interruption like FMIP [15], HMIP [16], and Freeze TCP [18]. The existing TCP congestion control is then applied for data transmission after restart. This scheme is proposed to clarify the impact of the communication interruption after the hand-off, and is referred to in short as the Immediate Expiration scheme.

3.2 Bandwidth-Aware Slow Start Threshold

This scheme restarts data transmission using a slow start similar to the case for the Immediate Expiration scheme whenever the interface is changed. Afterward, this scheme estimates the new bandwidth and updates *ssthresh* accordingly, allowing *cwnd* to rapidly expand up to the available bandwidth without becoming so large that packets will be lost upon reconnection. After updating of the *ssthresh*, the TCP congestion control of our proposed scheme is same as the conventinal TCP. This scheme is also referred to as the Bandwidth Aware scheme.

A method proposed by Aron et al. [22] is also applied to update *ssthresh* dynamically. In Aron's method, *ssthresh* is estimated and modified dynamically for a given interval time whenever some predetermined conditions are satisfied. However, in the Bandwidth Aware scheme, the *ssthresh* is set according to our proposed method, which simplifies the Hoe's proposal [5] based on the packet pair scheme [19], whenever the interface is changed **(hand-off)**. The initial *ssthresh* is adjusted to the optimized value given by the estimated Bandwidth-Delay product **(BW-DL)** using only first **three data** packets in the start-up period (slow start) of a TCP connection, as given by the following relation.

$$BW - DL = RTT \times \frac{Packet\,Size}{Lag\,of\,\mathbf{ACK}}. \tag{1}$$

where $\frac{PacketSize}{Lagof\mathbf{ACK}}$ represents the bandwidth. Data packets, which are sent closely spaced, arrive at the receiver at the rate of the bottleneck link bandwidth. The round-trip time (RTT) can be approximated by timing the **first** transmitted segment. When the **second and third** ACKs arrive at the sender with approximately the same spacing due to the congestion control mechanism of slow start phase, the approximate bandwidth can be calculated from the respective arrival times. In our proposed scheme, the difference of these arrival times is defined as "*Lag of* **ACK**." This gives a calculation of the BW-DL, and *ssthresh* is set according to this product. This method is applied just only after the hand-off, and congestion control of conventinal TCP is subsequently applied.

Note that the aim of our scheme is to avoid multiple packet loss causing severe performance degradation. Thus, we try to appropriately update ssthresh based upon the estimation of bandwidth by Eq. (1). For this purpose, as described in Sect. 2.2, the estimation should be performed quickly just after the restart and before the window size increases too much. This results in rough estimation of the bandwidth. Thus, our scheme does not precisely estimate the available bandwidth, but roughly estimate the physical one.

The pseudocode for the algorithm is as follows:

```
if ( the interface is changed )
        ssthresh = (BW-DL)/packet_size;
        if (ssthresh < 2)
               ssthresh = 2;
        endif
endif.
```

Note that the packet means the data packet as mentioned in the previous paragraph. Thus, the *packet_size* can be different for each data packet. In addition, the least number of *ssthresh* is defined as 2 in the conventional TCP congestion control.

As mentioned above, we employ the quite simplified BW-DL estimation method, that is, quickness of the bandwidth estimation is prior to accuracy of it. More specifically, in [5], it will take 3RTT to calculate the BW-DL: According to the "slow start" mode, the first packet is transmitted to calculate the RTT (1RTT). Subsequently, three packet pairs are transmitted to calculate the Bandwidth (2RTT). Just after the calculation of BW-DL and the setting of the *ssthresh* based on the BW-DL, the value of *cwnd* is boosted to 8, which indicates that 8 packets are transmitted in the network. However, a mobile host may access the Internet via a wireless network with low transmission capacity such as IMT-2000 under future mobile environment as shown in Fig. 1. In such a case, the optimal value of ssthresh has a potential to be less than 8 packets. For instance, when the mobile host is connected with IMT-2000 (bandwidth: 384 Kb/s, delay: 70 ms), BW-DL is set to 53760 bit (384*1000*0.14). When the size of a packet is set to 1500 bytes (1000 bytes), the BW-DL is equal to 4.48 (6.72) packets. Consequently, multiple packet losses may occur due to the excess transmission. In such a case, *cwnd* halves according to the TCP congestion control, and then never increases until the retransmission of these lost packets is completed, thereby causing performance degradation after the hand-off. Therefore, we employed the simplified method, which calculates the bandwidth using only one packet pair, to avoid the performance degradation.

Furthermore, in consideration of the accuracy of the Bandwidth Aware scheme, the computation of a BW-DL for *ssthresh* is known to be a difficult problem. The data packets transmitted back-to-back (second and third data packets) for the bandwidth estimation are supposed neither to be interrupted by cross-traffic nor to be lost at intermediate routers. As such interruption and packet loss are quite likely to occur in practice, the estimation of the bandwidth may become inaccurate. However, in Bandwidth Aware scheme, there is a low probability of occurrence of interruption and packet loss, because the bandwidth estimation is completed by the exchange of just only one packet pair (back-to-back packets). Furthermore, even if packets are interrupted or get lost, these interferences will not cause the severe performance degradation. More specifically, in such a case, the bandwidth will be underestimated, and the underestimation can lead to quick transition from slow start mode to congestion avoidance mode. This would cause performance degradation. However, it should be noted that the transmission rate can be increased steadily even in this case, whereas the transmission will cease for quite long time in the case of multiple packet loss. That is, the underestimated estimation can prevent an excessive increase in *cwnd* after the hand-off.

Actually, we have evaluated the accuracy of Bandwidth Aware scheme when the cross-traffic exists. As shown in Fig. 2, we deal with two cases: one mobile host (MH1) is changed from Wireless LAN to IMT-2000 (IMT-2000 case), and MH1 was changed from IMT-2000 to Wireless LAN (WLAN case). Note that Wireless LAN (20 Mb/s) is not a bottleneck link of end-to-end path between FH1 and MH1; meanwhile, IMT-2000 (384 Kb/s) is a bottleneck

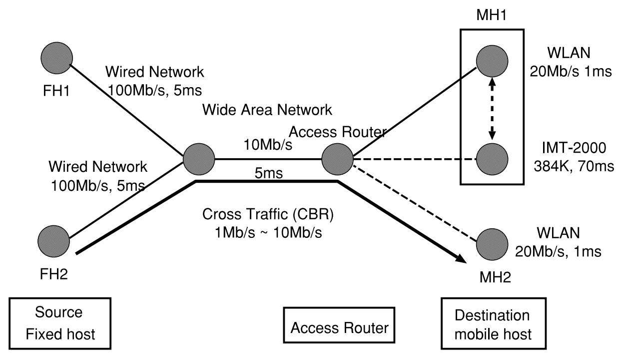


Fig. 2 Simulation model (Accuracy of bandwidth estimation).

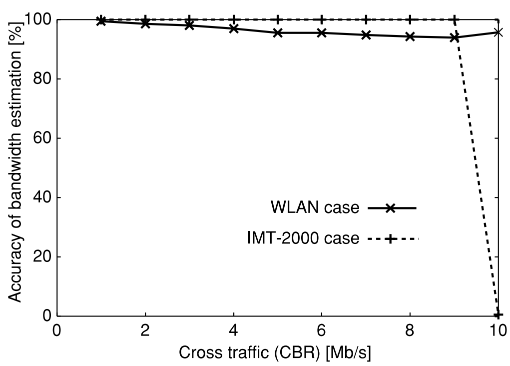


Fig. 3 Accuracy of bandwidth estimation.

link. In WLAN case, the bandwidth of bottleneck link is 10 Mb/s. Another mobile host (MH2) communicates with a fixed node (FH2) via Wireless LAN under cross-traffic. In our simulation, we treat the CBR traffic as a cross traffic, and change its traffic rate from 1 Mb/s to 10 Mb/s.

Figure 3 shows how our proposed scheme can estimate precisely the bandwidth even when the cross-traffic exists. Each of the results was obtained through 1000 simulations. In WLAN case, the estimation of the bandwidth slightly becomes inaccurate with the increase of the cross-traffic. On the other hand, in IMT-2000 case, Bandwidth Aware scheme can precisely estimate the bandwidth until the rate of cross-traffic reaches the bottleneck bandwidth (10 Mb/s). From these results, we can remark that Bandwidth Aware scheme can provide highly accurate estimation over a wide range of the amount of cross-traffic for both types of datalink interfaces (WLAN and IMT-2000), while the estimation of bandwidth becomes inaccurate for the heavy cross-traffic in both the types. In WLAN case, if the back-to-back packets are interrupted by one packet, the bandwidth will be incorrectly estimated as 5 Mb/s. Similarly, in IMT-2000 case, if one of these packets is lost, the estimated bandwidth becomes 47.6 Kb/s. In the simulations for both of the cases, the bandwidth is almost always correctly estimated even for very high cross-traffic, as shown in Fig. 3, while the bandwidth is incorrectly estimated as 5 Mb/s with high probability of 0.98 in WLAN case and as 47.6 Kb/s with 0.97 in IMT-2000 case if the estimation is incorrect. Furthermore, it should be noted that even if packets are interrupted or get lost by a cross-traffic, the bandwidth will be underestimated, thereby preventing an excess increase in *cwnd*. That is, Bandwidth Aware scheme can avoid the performance degradation due to the multiple packet loss. Thus, in this paper, cross-traffic will not be further considered.

As well as the congestion caused by the cross-traffic in the wired network, the transmission error in a wireless network can also cause the packet loss, and then the BW-DL can be underestimated. However, because the TCP congestion control of our proposed scheme is same as the conventional TCP, it is difficult for our proposed scheme to distinguish between the packet loss due to the transmission error in wireless network and the packet loss due to the congestion in wired network. If a packet gets lost due to the transmission error in the wireless network, the *cwnd* halves according to the TCP congestion control, while the *ssthresh* is limited to rather small value due to underestimated BW-DL. Although the transmission rate just after the hand-off is not enough high to efficiently use an available bandwidth due to the small *ssthresh* and *cwnd*, it never causes an excessive increase of transmission rate like in the cross-traffic case. As mentioned in the previous paragraph, the decrease of transmission rate cannot cause the significant performance degradation, while packet losses after the hand-off can cause it. Therefore, in our paper, the transmission error in a wireless network is not considered.

Note that, as described before, the proposed scheme is applied just only after the hand-off. Therefore, that can estimate the BW-DL of the network B by utilizing the first 3 packets transmitted to network B, even if an MH moves across between two wireless networks (from network A to network B) with different packet loss rate. That is, the transmission error in the previous network (network A) does not have an impact on the accuracy of the estimation.

4. Simulation Model

The proposed schemes are compared with existing schemes using the Virtual InterNetwork Testbed (VINT) network simulator (NS) version 2 [23]. In the simulations, the bandwidth and propagation delay of the link between the access router (AR) and the destination vary dynamically according to the channel conditions, as shown in Fig. 4. Several cases were investigated extensively, including cases in which the datalink interface was changed from one wireless interface to another wireless interface. In other cases, the mobile host was changed from IMT-2000 to wired LAN. Cases involving more than one change and the long associated communication interruptions were also simulated.

The simulation parameters are listed in Table 1. Simulations were conducted for a period of 60 s, in which the datalink interface was changed at the 10 s mark. The duration of interruption associated with switching datalink interfaces was 0.5 or 3.5 s. The theoretical hand-off time in [24], [25] was employed in this simulation. Up to two TCP flows are considered, with a TCP packet size of 1000 bytes. The

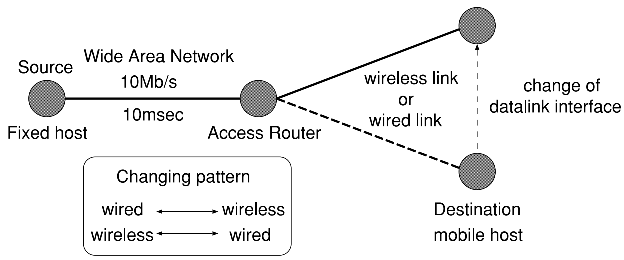


Fig. 4 Simulation model.

Table 1 Simulation parameters.

Simulation time	60 s
Time when the interface starts changing	10 s
Interruption time	0.5 or 3.5 s
Packet size	1000 bytes
TCP	NewReno
IMT-2000	384 Kb/s, 70 ms
Wireless LAN (IEEE802.11a)	20 Mb/s, 1 ms
Wired LAN	100 Mb/s, 5 ms

TCP variant employed here is NewReno [26], and the TCP traffic considered is greedy file transfer such as ftp-data. The propagation delay and bandwidth change drastically upon a change in datalink interface. In these simulations, the bandwidth and delay of each datalink interface were set at fixed values as shown in Table 1, and wireless characteristics such as bit error and fading were not considered. In our simulation model, a propagation delay includes processing delay in the AR and the mobile host. A link delay of IMT-2000 network is set to large value (70 ms) because this propagation delay is typically larger than that of Wireless LAN netwrok and a Transmission Time Interval (TTI) [27] is added to this delay.

In this paper, the characteristics of TCP *cwnd*, TCP throughput, and transient TCP performance in the periods of interest are examined in detail to clarify the impact of the proposed scheme.

5. Results and Discussion

This section presents the simulation results for four cases. The first three cases consider a single TCP flow, and the last case considers two TCP flows.

5.1 Case 1: Effect of Bandwidth Change

Two simulations were performed to examine the effect of bandwidth change. The first considered a mobile datalink interface change from IEEE802.11a Wireless LAN (20 Mb/s) to IMT-2000 (384 Kb/s), with a communication interruption time of 0.5 s and a propagation of 1 ms (Wireless LAN) or 70 ms (IMT-2000).

Figure 5 shows the change in *cwnd* over this period, and Fig. 6 shows how the sequence number increases after the datalink interface is changed. Communication under TCP/mm stops at 11 s due to retransmission timeout (I),

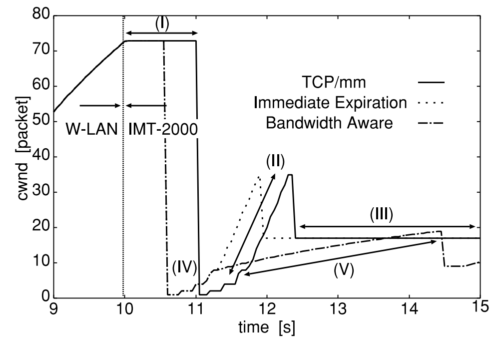


Fig. 5 *cwnd* (20 Mb/s → 384 Kb/s, 0.5 s).

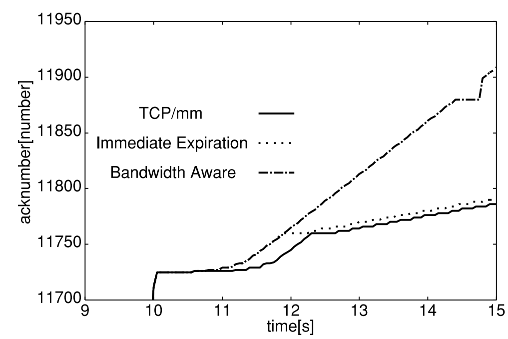


Fig. 6 Sequence number (20 Mb/s → 384 Kb/s, 0.5 s).

even though the new datalink interface becomes available at 10.5 s. The TCP sequence number remains static in this interval (between 10.5 and 11 s) while the sender waits for ACK signals for segments lost during hand-off. Transmission is restarted using slow start, by which *cwnd* is increased exponentially (II). However, this rapid expansion of *cwnd* results in multiple packet loss if the bandwidth is reduced as in this case. This can be seen from the constant value of *cwnd* after 12.5 s (III), which represents retransmission of lost packets under the fast recovery algorithm of NewReno, with a corresponding very slow increase in packet sequence number.

Under the Immediate Expiration scheme begins transmitting packets using slow start immediately after the new datalink interface becomes available after hand-off (IV). However, *cwnd* becomes constant for a long time shortly after restart, similar to the case for the TCP/mm scheme. This is because Immediate Expiration also uses the existing TCP congestion control for data transmission.

The Bandwidth Aware scheme also begins transmitting packets using slow start immediately after hand-off. However, this scheme also estimates the available bandwidth on the new datalink interface, and sets *ssthresh* to the most appropriate value. This successfully prevents excess packet loss after hand-off, allowing *cwnd* to be expanded effectively during slow start. As a result, the throughput of the Bandwidth Aware scheme is far greater than for the Immediate Expiration or TCP/mm.

Table 2 shows the average TCP throughput for the period between 10 and 15 s, normalized against the bottleneck

Table 2 Throughput (20 Mb/s → 384 Kb/s, 0.5 s).

Datalink interface change pattern 20 Mb/s → 384 Kb/s	Throughput Mb/s(%)
Ideal throughput	0.384
TCP/mm	0.117 (30.5)
Immediate Expiration	0.122 (31.8)
Bandwidth Aware	0.291 (75.8)

Table 3 Throughput (384 Kb/s → 100 Mb/s, 0.5 s).

Datalink interface change pattern 384 Kb/s → 100 Mb/s	Throughput Mb/s(%)
Ideal throughput	10.000
TCP/mm	6.442 (64.2)
Immediate Expiration	7.372 (73.7)
Bandwidth Aware	8.218 (82.2)

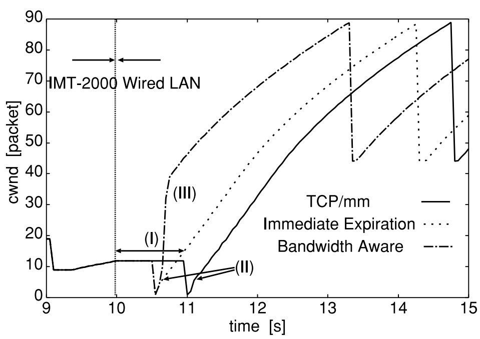


Fig. 7 *cwnd* (384 Kb/s → 100 Mb/s, 0.5 s).

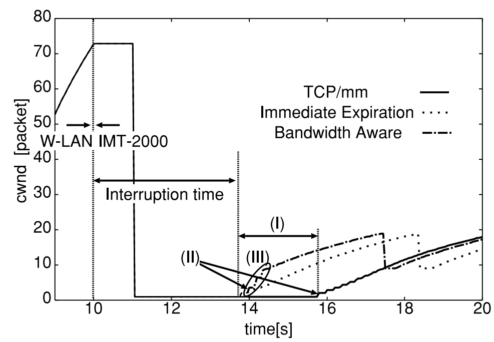


Fig. 8 *cwnd* (20 Mb/s → 384 Kb/s, 3.5 s).

link bandwidth (384 Kb/s in this case) given in parentheses. The Bandwidth Aware scheme achieves a throughput of 0.291 Mb/s, twice that of TCP/mm (0.117 Mb/s), while the improvement offered by Immediate Expiration is not significant (0.122 Mb/s). These results suggest that the Bandwidth Aware scheme greatly contributes to throughput performance after hand-off from one datalink interface to another in this context.

In the second simulation, the interface was switched from IMT-2000 (384 Kb/s) to wired LAN (100 Mb/s), with a communication interruption time of 0.5 s and a propagation delay of 70 ms (IMT-2000) or 5 ms (wired LAN).

Figure 7 shows how *cwnd* changes after the datalink interface is switched. In this case, TCP/mm is forced to wait until 11 s (I) for the expiration of the retransmission timeout timer. After that, packets are first transmitted in the slow start phase, which is however immediately changed to a congestion avoidance phase (II). This prevents *cwnd* from increasing efficiently, even though the bandwidth was expanded to 100 Mb/s. The inability of the existing TCP congestion control to reset *ssthresh* to an appropriate value when the datalink interface is changed leads to inefficient expansion of *cwnd* after hand-off. The Immediate Expiration scheme also suffers from the same problem of congestion avoidance after restart, resulting in similar performance to TCP/mm.

In contrast, the Bandwidth Aware scheme sets *ssthresh* to an appropriate value based on the estimated bandwidth when restarting by slow start immediately after hand-off. This ensures that congestion avoidance is not triggered shortly after the restart, allowing *cwnd* to expand optimally in this context (III). As a result, the throughput of the Bandwidth Aware scheme outperforms the other two schemes by a significant margin.

The average throughput for the 10–15 s interval for this scenario is shown in Table 3, with values normalized to a bottleneck bandwidth (10 Mb/s in this case) given in parentheses. The Bandwidth Aware scheme achieves a throughput of 8.218 Mb/s, about 1.3 times that of TCP/mm (6.442 Mb/s). Again, the improvement offered by the Immediate Expiration scheme is not significant (7.372 Mb/s). The Bandwidth Aware scheme therefore offers significant performance increases in this scenario as well.

These results show that updating *ssthresh* based on the estimated bandwidth has a significant impact on the throughput of TCP in situations where the bandwidth changes markedly when the datalink interface is switched.

5.2 Case 2: Effect of Interruption Time

In this case, the switch from 20 Mb/s Wireless LAN to 384 Kb/s IMT-2000 was simulated assuming a 3.5 s interruption during hand-over. The propagation delay was 1 ms (Wireless LAN) or 70 ms (IMT-2000). This interruption of 3.5 s simulates typical behavior for the current IMT-2000 architecture, which requires a Point-to-Point Protocol (PPP) connection to be established with the gateway connecting the radio access network (RAN) and the Internet in advance.

Figure 8 shows the change in *cwnd* after the datalink interface is changed to IMT-2000 with a 3.5 s interruption. Communication under TCP/mm is stopped for a further 2.5 s (I) after the interruption due to retransmission timeout. The longer delay in this case is due to the way in which the retransmission timeout timer is handled under TCP: the timer is increased exponentially each time the timer expires, which would occur a number of times in a lengthy interruption. On such occasions, *ssthresh* is reset to the minimum value as a result of multiple timeouts, resulting in congestion avoidance soon after slow start (II). This results in a very inefficient increase in *cwnd* and poor network performance.

Table 4 Throughput (20 Mb/s → 384 Kb/s, 3.5 s).

Data link interface change pattern 20 Mb/s → 384 Kb/s	Throughput Mb/s(%)
Ideal throughput	0.384
TCP/mm	0.124 (32.3)
Immediate Expiration	0.199 (51.8)
Bandwidth Aware	0.213 (55.5)

Under the Immediate Expiration scheme, retransmission begins immediately after the new datalink interface becomes available at 13.5 s, but congestion avoidance is again triggered shortly after in this case. Although performing better than TCP/mm with the longer delay mentioned above, Immediate Expiration is still unable to increase *cwnd* efficiently after a hand-over.

In contrast, the Bandwidth Aware scheme starts retransmission by slow start immediately after hand-off with an appropriate *ssthresh* value for the new available bandwidth. This promotes efficient expansion of *cwnd* after the new interface becomes available in this context, outperforming the other schemes.

The Bandwidth Aware scheme achieves an average throughput of 0.213 Mb/s, about 1.7 times that of TCP/mm (0.124 Mb/s), as shown in Table 4. However, the improvement offered by the Immediate Expiration scheme is also quite good in this case (0.199 Mb/s).

These results demonstrate that restarting data transmission immediately by slow start instead of waiting for the retransmission timeout has a significant impact on the TCP throughput when the interruption associated with the change of the datalink interface is relatively long.

5.3 Case 3: Effect of Multiple Changes

Future mobile environment is expected to turn into heterogeneous networks, with the potential for frequent hand-offs to different types of networks at short intervals. The simulation presented here represents such a case of multiple network changes.

The effect of multiple changes of datalink interfaces was examined by switching the interface between Wireless LAN (20 Mb/s) and IMT-2000 (384 Kb/s) at 10, 13, 16, and 19 s, with a communication interruption time of 0.5 s and a propagation delay of 1 ms (Wireless LAN) or 70 ms (IMT-2000).

Figure 9 shows how *cwnd* changes after the datalink interface is changed multiple times. Under TCP/mm, communication stops for about 1 s due to retransmission timeout, even though the new datalink interface becomes available after the first hand-off. Transmission is then restarted by slow start, and *cwnd* increases exponentially. However, because of the reduced bandwidth, the scheme suffers multiple packet loss, entering fast recovery under NewReno and retransmitting lost packets until 15.5 s, as indicated by the constant value of *cwnd*. Retransmission packets under fast recovery will be lost due to the second hand-off at 13 s, and will never be recovered until the retransmission timeout timer expires [28]. After the timer expires, the data transmission including retransmissions can start by slow start, whereas *cwnd* is prevented from increasing efficiently. Furthermore, as existing TCP congestion control schemes set *ssthresh* to half the old *cwnd* value after expiration of the retransmission timeout, *ssthresh* will become very small and TCP congestion control will enter congestion avoidance immediately.

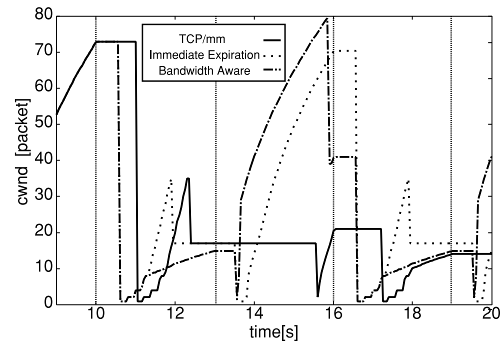


Fig. 9 *cwnd* (20 Mb/s ⇔ 384 Kb/s, 0.5 s).

Table 5 Throughput (20 Mb/s ⇔ 384 Kb/s, 0.5 s).

Datalink interface change pattern 20 Mb/s ⇔ 384 Kb/s	Throughput Mb/s(%)
Ideal throughput	3.192
TCP/mm	0.253 (7.8)
Immediate Expiration	2.047 (64.5)
Bandwidth Aware	2.670 (83.6)

The Immediate Expiration scheme starts retransmitting packets by slow start immediately after the new datalink interface becomes available after the first hand-off (10.5 s), and then *cwnd* increases exponentially. This results in an improvement in throughput performance as shown in Table 5.

The Bandwidth Aware scheme starts transmitting packets by slow start immediately after hand-off with a *ssthresh* value set optimally for the new available bandwidth. As shown in the figure, this results in much higher performance when the interface is switched consecutively.

The average throughput clearly shows the improvement offered by the Bandwidth Aware scheme (Table 5). For the period between 10 and 20 s, the ideal throughput is 3.192 Mb/s, as calculated by

$$Ideal\,throughput = \frac{sum\,of\,(BW \times duration)}{simulation\,time}. \qquad (2)$$

The Bandwidth Aware scheme achieves a throughput of 2.670 Mb/s, about 10 times that of TCP/mm (0.253 Mb/s) and greater than the Immediate Expiration scheme (2.047 Mb/s).

These results clearly demonstrate that the Bandwidth Aware scheme greatly improves throughput performance when the datalink interface is switched consecutively at short intervals.

5.4 Case 4: Effect of Multiple TCP Flows

The last case considered here is the switching of multiple TCP flows due to hand-off. In the future mobile Internet environment, many users will often have the opportunity to establish multiple TCP flows at the same time. For example, email attachments, which may contain small text files or bulky image data, may be transmitted over a pretty long time, resulting in simultaneous activity using the most suitable network.

In these simulations, two TCP flows are considered. The first simulation considers a change of the mobile datalink interface from 20 Mb/s Wireless LAN to 384 Kb/s IMT-2000, with a communication interruption time of 0.5 s and a propagation of 1 ms (Wireless LAN) or 70 ms (IMT-2000).

Figure 10 shows the change in *cwnd* after the datalink interface is changed for the TCP/mm, Immediate Expiration and Bandwidth Aware schemes. As shown in Fig. 10(a), all TCP communications under TCP/mm stop at 11 s due to retransmission timeout, even though the new datalink interface becomes available at 10.5 s. These transmissions are restarted using slow start, by which *cwnd* is increased exponentially. However, because of the reduced bandwidth, these flows suffer multiple packet loss. This leads to retransmission of lost packets under the fast recovery algorithm of NewReno, as indicated by the constant value of *cwnd* after 12.5 s (I).

Under the Immediate Expiration scheme, packet transmission begins using slow start immediately after the new datalink interface becomes available at 10.5 s, as shown in Fig. 10(b). However, *cwnd* becomes constant for a long time shortly after restart (II), similar to the case for TCP/mm.

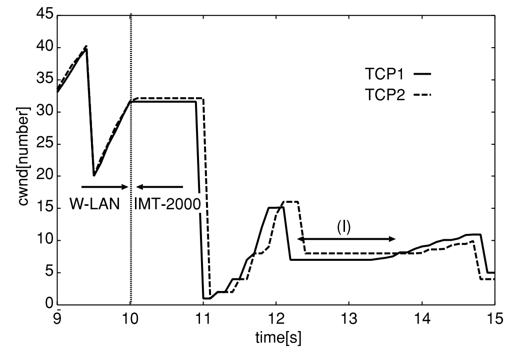
The two TCP flows under the Bandwidth Aware scheme begin transmitting packets immediately after hand-off using slow start with the appropriate *ssthresh* value (III). As shown in this figure, *cwnd* value increases effectively without excess packet loss after hand-off, resulting in much higher performance than either the Immediate Expiration or TCP/mm scheme.

Table 6 shows the average TCP throughput for the 10-15 s interval in this scenario. The average throughput in this case represents the sum of the two TCP flows. The Bandwidth Aware scheme achieves an average throughput of 0.303 Mb/s, about 80% of the maximum throughput offered by the new available bandwidth (384 Kb/s in this case) and significantly outperforming the other schemes.

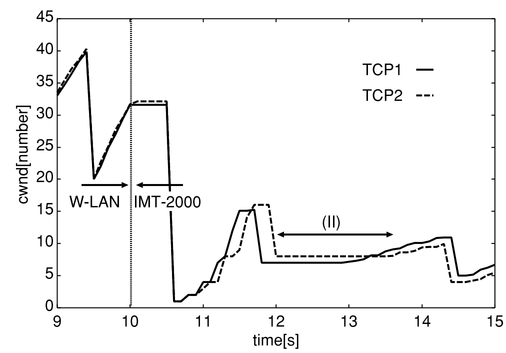
Several cases of two TCP flows were examined. The first case is a change from 384 Kb/s IMT-2000 to 20 Mb/s Wireless LAN, and the other cases are switching from 20 Mb/s Wireless LAN to 384 Kb/s IMT-2000 with a 3.5 s interruption. The results of these simulations also show that the Bandwidth Aware scheme provides a significant increase in throughput performance after hand-off.

In the second simulation, the effect of multiple changes in datalink interfaces was examined by switching the interface between Wireless LAN (20 Mb/s) and IMT-2000 (384 Kb/s) at 10, 13, 16, and 19 s, with a communication interruption time of 0.5 s, and a propagation delay of 1 ms (Wireless LAN) or 70 ms (IMT-2000) as in case 3.

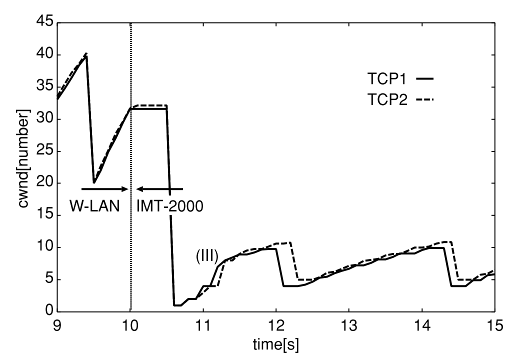
Figure 11 shows the change in *cwnd* after the hand-off for the TCP/mm, Immediate Expiration and Bandwidth Aware schemes. As shown in Fig. 11(a), TCP communication under TCP/mm stops at 11 s due to retransmission timeout, and transmissions restart by slow start, increasing



(a) TCP/mm



(b) Immediate Expiartion

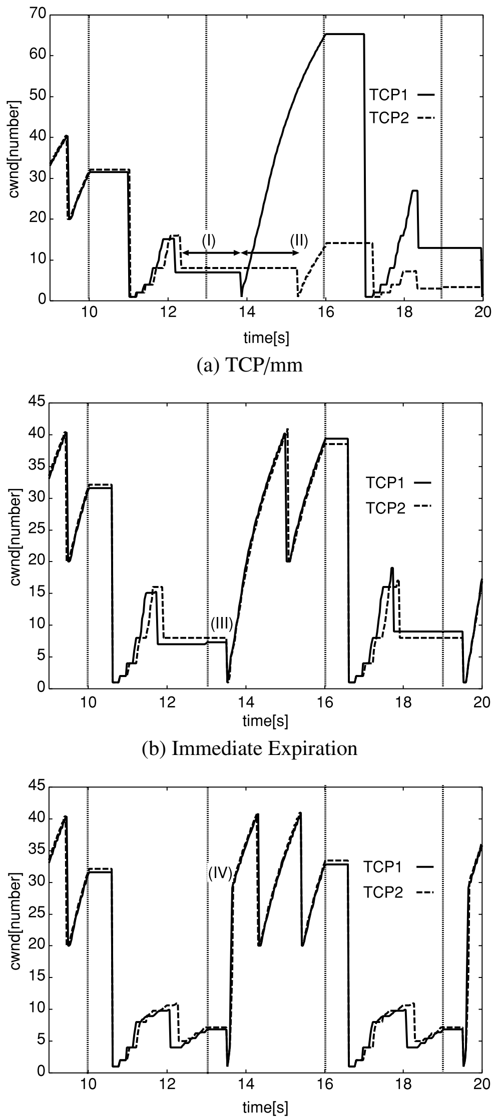


(c) Bandwidth Aware

Fig. 10 *cwnd* (20 Mb/s ⇔ 384 Kb/s, 0.5 s, 2flow).

Table 6 Throughput (20 Mb/s ⇒ 384 Kb/s, 0.5 s).

Datalink interface change pattern 20 Mb/s ⇒ 384 Kb/s	Throughput [Mb/s] (%)
Ideal throughput	0.384
TCP/mm	0.220 (57.3)
Immediate Expiration	0.250 (65.1)
Bandwidth Aware	0.303 (78.9)



(a) TCP/mm
(b) Immediate Expiration
(c) Bandwidth Aware

Fig. 11 *cwnd* (20 Mb/s ⇔ 384 Kb/s, 0.5 s, 2flow).

cwnd exponentially as in the previous simulation. However, this rapid expansion of *cwnd* leads to multiple packet loss when a change in datalink interface occurs, as shown by the constant *cwnd* after 12.5 s (I), which represents the retransmission of lost packets under the fast recovery algorithm. However, retransmission packets of all TCP communication will be lost when the second hand-off occurs at 13 s. TCP1 flow then starts by slow start after the TCP1 retransmission timeout at 13.5 s. Similarly, transmission of TCP2 flow starts after the TCP2 retransmission timeout at 15.5 s (II). As shown in the figure, this results in imbalance between the TCP flows.

Under the Immediate Expiration scheme, packets are transmitted by slow start immediately after hand-off, and *cwnd* is increased exponentially in the slow start phase. Therefore, these TCP flows suffer multiple packet loss when the bandwidth is reduced dynamically. However, these TCP flows begin again by slow start immediately after the second hand-off (III), leading to balance between the TCP flows and an improvement in throughput performance, as shown in Table 7.

Table 7 Throughput (20 Mb/s ⇔ 384 Kb/s, 0.5 s).

Datalink interface change pattern 20 Mb/s ⇔ 384 Kb/s)	Throughput [Mb/s] (%)
Ideal throughput	3.192
TCP/mm	1.590 (49.8)
Immediate Expiration	2.571 (80.5)
Bandwidth Aware	2.825 (88.5)

The Bandwidth Aware scheme starts retransmitting packets by slow start immediately after every hand-off, each time with an appropriate *ssthresh* value for the new available bandwidth. This ensures that congestion avoidance is not triggered shortly after the restart, allowing *cwnd* to expand optimally. After the second hand-off, *cwnd* is increased to a higher value quicker than under the Immediate Expiration scheme (IV). As shown in this figure, this results in balance between the two TCP flows, as in the Immediate Expiration scheme and unlike TCP/mm.

The average throughput clearly shows the advantage of the Bandwidth Aware scheme (Table 7). For the period between 10 and 20 s, the calculated ideal throughput is 3.192 Mb/s as in Case 3. The Bandwidth Aware scheme achieves a throughput of 2.825 Mb/s, corresponding to a utilization of about 90% of the ideal throughput and outperforming the other schemes significantly (TCP/mm: 1.590 Mb/s, Immediate Expiration: 2.571 Mb/s).

These results clearly demonstrate that the Bandwidth Aware scheme greatly improves throughput performance when the datalink changes consecutively at short intervals with multiple TCP flows.

6. Conclusion and Future Work

Two new congestion control schemes were presented to facilitate the move to future mobile environments in which the datalink interface may change dynamically to achieve effective use of available network resources. In such an environment, ensuring stable and efficient TCP flow requires the scheme to be able to handle host IP address changes associated with the change in datalink interface, as well as significant changes in bandwidth.

The schemes were evaluated through extensive simulations of different scenarios. The first control scheme, Immediate Expiration of Timeout Timer, offers only moderate performance improvements over existing TCP/mm technologies in that the waiting time until start of retransmission is shortened, but subsequent congestion control is adversely affected by changes in bandwidth in a similar manner to TCP/mm. Under the second scheme, Bandwidth-Aware Slow Start Threshold, retransmission picks up promptly as in the Immediate Expiration scheme, but subsequent flow in this case is controlled by slow start with an optimized value of *ssthresh* based on estimation of the new bandwidth.

The simulation results demonstrated that the network performance offered by the Bandwidth Aware scheme is considerably higher than either TCP/mm or Immediate Expiration, even when considering more than one TCP flow. The Bandwidth Aware scheme is capable of 80% utilization of the available bandwidth in all simulation cases, and up to 90% when changes occur consecutively at short intervals. The Immediate Expiration scheme also performs quite well when the communication interruption associated with the interface change is long.

In our paper, our proposed scheme was evaluated through simulation experiments in detail. It is also of practical importance to implement it as well. Therefore, we are carrying out the implementation of our proposed scheme in the Linux Kernel. After the completion of the implementation, we plan to report the experiment results.

Acknowledgments

This work was supported in part by the Japan Society for the Promotion of Science, Grant-in-Aid for Scientific Research (s)(18100001) and JSPS Fellows (17-6551), and in part by the Ministry of Public Management, Home Affairs, Posts and Telecommunications, Japan.

References

[1] H. Luo, Z. Jiang, Z.K. Shankaranarayanan, and P. Henry, "Integrating wireless LAN and cellular data for the enterprise," IEEE Internet Computing Magazine, pp.25–33, March/April 2003.
[2] S. Shakkottai, T.S. Rappaport, and P.C. Karlsson, "Cross-layer design for wireless networks," IEEE Commun. Mag., vol.41, no.10, pp.74–80, Oct. 2003.
[3] G. Wu, Y. Bai, J. Lai, and A. Ogielski, "Interactions between TCP and RLP in wireless Internet," Proc. IEEE GlobeCom'99, pp.661–666, Dec. 1999.
[4] M. Stemm, "Vertical handoffs in wireless overlay networks," Mobile Netw. Appl., vol.3, no.4, pp.335–350, 1998.
[5] J.C. Hoe, "Improving the start-up behavior of a congestion control scheme for TCP," Proc. ACM SIGCOMM'96, pp.270–280, Aug. 1996.
[6] 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, Rome, Italy, July 2001.
[7] S. Mascolo, C. Casetti, M. Gerla, S.S. Lee, and M. Sanadini, "TCP-Westwood: Congestion window control using bandwidth estimation," Proc. Globecom 2001, San Antonio, TX, Nov. 2001.
[8] Y. Matsushita, T. Matsuda, and M. Yamamoto, "TCP congestion control with ACK-pacing for vertical handover," Proc. IEEE WCNC2005, New Orleans, LA, March 2005.
[9] D. Katabi, M. Handley, and C. Rhors, "Congestion control for high bandwidth-delay product networks," Proc. ACM SIGCOMM 2002, Pittsburgh, PA, Aug. 2002.
[10] A. Jain, "Quick-start for TCP and IP," Internet draft draft-amit-quick-start-02.txt, work in progress, Oct. 2002.
[11] S. Floyd, "Highspeed TCP for large congestion windows," Internet draft draft-floyd-tcp-highspeed-02.txt, work in progress, Feb. 2003.
[12] I.F. Akyildiz, G. Morabito, and S. Palazzo, "TCP-Peach: A new congestion control scheme for satellite networks," IEEE/ACM Trans. Netw., vol.9, no.3, pp.307–321, June 2001.
[13] C. Perkins, "IP mobility support for IPv4," Networking Group, RFC3220, Jan. 2002.
[14] D. Johnson, "Mobility support in IPv6," IETF Internet-Draft, draft-ietf-mobileip-ipv6-24.txt, June 2003.
[15] R. Koodli, "Fast handovers for Mobile IPv6," RFC4068, Internet Engineering Task Force (IETF), Oct. 2005.
[16] H. Soliman, C. Castelluccia, K. El Malki, and L. Bellier, "Hierarchical Mobile IPv6 mobility management (HMIPv6)," RFC4140, Internet Engineering Task Force (IETF), Aug. 2005.
[17] A.C. Snoeren and H. Balakrishnan, "An end-to-end approach to host mobility," ACM/IEEE MobiCom'00, no.6, pp.155–166, Boston, MA, Aug. 2000.
[18] T. Goff, J. Moronski, D.S. Phatak, and V. Gupta, "Freeze-TCP: A true end-to-end TCP enhancement mechanism for mobile environments," Proc. IEEE INFOCOMM'00, Tel Aviv, Israel, March 2000.
[19] S. Keshav, "A control-theoretic approach to flow control," Proc. ACM SIGCOMM, pp.3–15, Sept. 1991.
[20] A. Capone, L. Fratta, and F. Martignon, "Bandwidth estimation schemes for TCP over wireless networks," IEEE Trans. Mobile Computing, vol.3, no.2, April/June 2004.
[21] V. Kawadia and P.R. Kumar, "A cautionary perspective on cross-layer design," IEEE Wireless Commun., vol.12, no.1, pp.3–11, Feb. 2005.
[22] M. Aron and P. Druschel, "TCP: Improving startup dynamics by adaptive timers and congestion control," Technical Report TR98-318, Rice University Computer Science, 1998.
[23] VINT Project, Network Simulator ns-2, http://www/isi.edu/nsnam/ns
[24] N.A. Fikouras, K. EL Malki, S.R. Cvetkovic, and M. Kraner, "Performance analysis of Mobile IP handoffs," Proc. IEEE APMC'99, Asia Pacific Microwave Conference, Singapore, Dec. 1999.
[25] M. Kraner, C.N. Yap, S. Cvetkovic, and E. Sanchez, "Can Mobile IP be used as a link between IMT-2000 technologies and operators?" Karlsruhe Workshop on Software Radios, 2000.
[26] S. Floyd and T. Henderson, "The NewReno modification to TCP's fast recovery algorithm," RFC2582, April 1999.
[27] 3GPP TS 25.302 Services provided by the physical layer, version 5.2, 2002. Available via http://www.3gpp.org [2003-01-10]
[28] K. Fall and S. Floyd, "Simulation-based comparisons of Tahoe, Reno, and SACK TCP," ACM Computer Communication Review, vol.26, no.3, pp.5–21, 1996.

Kazuya Tsukamoto received M.E. degrees in computer science from Kyushu Institute of Technology, Iizuka, Japan in 2003. Since 2003, he has been a Ph.D. candidate at the Graduate School of Computer Science and Systems Engineering, Kyushu Institute of Technology. His research interests include performance evaluation of computer networks, wireless networks, and transport protocol. At present, he is a JSPS Research Fellow, and is also a student member of the IEEE.

Yutaka Fukuda received B.E., M.E. and D.E. degrees in computer science from Kyushu Institute of Technology, Iizuka, Japan in 2000, 2002 and 2005 respectively. Since October 2003, he has been a Research Associate in the Information Science Center, Kyushu Institute of Technology. His research interests include performance evaluation of computer networks, wireless networks, and transport protocol. He is a member of the IEEE.

Yoshiaki Hori received B.E., M.E., and D.E. degrees from Kyushu Institute of Technology, Iizuka, Japan in 1992, 1994, and 2002, respectively. From 1994 to 2003, he was a Research Associate in Common Technical Courses, Kyushu Institute of Design, Fukuoka. From 2003 to 2004, he was a Research Associate in the Department of Art and Information Design, Kyushu University, Fukuoka. Since March 2004, he has been an Associate Professor in the Department of Computer Science and Communication Engineering, Kyushu University. His research interests include network security, network architecture, and performance evaluation of network protocols on various networks. He is a member of IEEE, ACM, and IPSJ.

Yuji Oie received B.E., M.E. and D.E. degrees from Kyoto University, Kyoto, Japan in 1978, 1980 and 1987, respectively. From 1980 to 1983, he worked at Nippon Denso Company Ltd., Kariya. From 1983 to 1990, he was with the Department of Electrical Engineering, Sasebo College of Technology, Sasebo. From 1990 to 1995, he was an Associate Professor in the Department of Computer Science and Electronics, Faculty of Computer Science and Systems Engineering, Kyushu Institute of Technology, Iizuka. From 1995 to 1997, he was a Professor in the Information Technology Center, Nara Institute of Science and Technology. Since April 1997, he has been a Professor in the Department of Computer Science and Electronics, Faculty of Computer Science and Systems Engineering, Kyushu Institute of Technology. His research interests include performance evaluation of computer communication networks, high speed networks, and queueing systems. He is a fellow of the IPSJ and a member of the IEEE.