New flow control schemes of TCP for multimodal mobile hosts

Kazuya Tsukamoto, Yutaka Fukuda
Dept. of Computer Science
and Electronics,
Kyushu Institute of Technology
Kawazu 680–4, Iizuka, 820–8502 Japan
Telephone:+81-948-29-7687
Fax:+81-948-29-7651
Email: {kazuya,ytk}@infonet.cse.kyutech.ac.jp

Yoshiaki Hori
Dept. of Art and Infomation Design,
Kyushu Institute of Design
Shiobaru 4-9-1, Minami-ku,
Fukuoka, 815–8540 Japan
Telephone:+81-92-553-4443
Fax:+81-92-553-4440
Email: hori@kyushu-id.ac.jp

Yuji Oie
Dept. of Computer Science
and Electronics,
Kyushu Institute of Technology
Kawazu 680–4, Iizuka, 820–8502 Japan
Telephone:+81-948-29-7687
Fax:+81-948-29-7651
Email: oie@cse.kyutech.ac.jp

I. Abstract

Two control schemes designed specifically to handle changes in the datalink interface for 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. However, the existing TCP architecture is unable to maintain stable and efficient throughput as a result of such interface changes. The main issues are how to handle the change in host IP address when the datalink interface changes, and how to ensure reliable and continuous TCP flow. Although architectures addressing the first of these issues have already been proposed, the problem of flow control still remains. The first of the proposed schemes, Immediate Expiration of Timeout Timer, detects such an interface change, and begins retransmission immediately without waiting for retransmission timeout as in conventional TCP. The second scheme, Bandwidth Aware Slow Start Threshold, detects the interface changes, and also estimates the new bandwidth so as to set an appropriate slow start threshold when beginning retransmission. Through simulations, the Bandwidth Aware Slow Start Threshold scheme is demonstrated to provide marked improvements in performance over conventional TCP, particularly in situations where the bandwidth changes significantly and the interface is changed consecutively at short intervals. The Immediate Expiration of Timeout Timer scheme is also shown to be effective when the communication interruption associated with interface change is long.

II. 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 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 and 11b wireless LAN are a widely implemented technology that can provide high-speed access to the Internet within a limited area.

As the various datalink technologies become widely deployed, mobile hosts will be very likely to be capable of multiple datalink interfaces. It will therefore become necessary for the mobile host to select automatically one of the available datalink interfaces based upon the transmission capacity, signal intensity, bit error condition, and other factors. This introduces the concept of multimodal communication for mobile devices, raising the concern of how to ensure efficient TCP flow in this context. The most important issues are potential changes in the IP address of the mobile host when the interface mode is switched, and the ability for TCP flow control to handle interface handovers. In the former case, as a TCP flow is uniquely identified by the four parameters of source IP address, source 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 case, the time required to switch from one datalink interface for another may exceed the timeout for current TCP flow control, which may also not be able to accommodate the potentially large change in available bandwidth. Although a number of technologies have already been proposed to resolve the problem associated with the change in IP address, these existing architectures do not specifically address the concerns of TCP flow control. In the present paper, the authors propose two new flow control schemes that offer improved performance over conventional TCP when considering changes in mobile datalink interfaces.

III. Existing technologies for multimode mobile hosts

As stated in the introduction, a number of technologies have been proposed to handle host IP changes. One example is Mobile IP [1], which has matured rapidly to a stage where it is being proposed as a standard by the Internet Engineering Task Force (IETF) for supporting mobility on the Internet. This technology provides transparent support for host mobility by concealing the changes in IP addresses of the mobile host 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. This occurs when the mobile host informs the HA of its new IP address, allowing continuous delivery of datagrams even when the IP address of the mobile host changes.

Another proposed technology is a Massachusetts Institute of Technology (MIT) architecture [2] that eliminates the dependence of TCP connection upon the host IP address, thereby preserving the TCP connection in the event that the host IP address changes. This approach, proposed by Snoeren et al., adds two new TCP options; a migrate option, and a migrated-permitted option, to the current TCP. This architecture preserves a TCP connection by sending packets that contain these two TCP options.

These two technologies do not modify the existing TCP flow control scheme, and are referred to in this paper as "TCP technologies".

In contrast, Freeze TCP [3] entails a new 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 interval before reconnection. However, although this scheme achieves the desired result with regard to disconnection/reconnection by preventing the retransmission timeout timer from expiring, it does not consider the possibility that the bandwidth/latency changes after reconnection. Therefore, significant packet loss may occur after the TCP communication restarts.

In this paper, two new schemes are proposed to address these issues related to mobile TCP/IP connectivity. 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 further approaches the problem by detecting the interface change as a trigger for adapting its flow control to the changes, then sets the transmission window based on an estimation of the new bandwidth.

IV. Proposed Technologies

A. 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. This scheme forces the timer to immediately expire, thereby avoiding unnecessary communication interruption. The existing TCP is then applied for data transmission after restart. This scheme is referred to in short as the Immediate Expiration scheme.

B. Bandwidth Aware Ssthreshold

Whenever the interface is changed, 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. This scheme is referred to here as the Bandwidth Aware scheme.

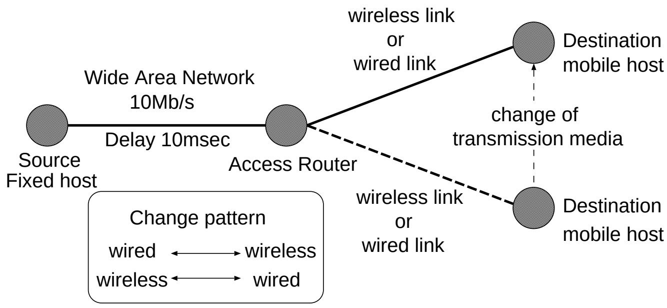

Fig. 1. Simulation Model.

In the proposed Bandwidth Aware scheme, the *ssthresh* value is set in part according to the method proposed by Hoe [4]. This scheme adjusts the initial *ssthresh* value to the optimized value given by the estimated bandwidth-delay product in the start-up period of a TCP connection as follows:

$$BW - Delay\, product = RTT \times \frac{Packet\, Size}{Lag\, of\, ACK}. \quad (1)$$

$$ssthresh = \frac{BW - Delay\, product}{Packet\, Size}. \quad (2)$$

where $\frac{PacketSize}{Lag of ACK}$ represents the bandwidth. Data packets, which are sent closely spaced, arrive at the receiver at the rate of the bottleneck link bandwidth. If the second and third ACKs arrive at the sender with approximately the same spacing, the approximate bandwidth can be calculated from the respective arrival times. The round-trip time (RTT) can be approximated by timing the first segment. This gives a calculation of the bandwidth-delay product, and *ssthresh* is set according to the byte equivalent of this product. This method is applied whenever the interface is changed.

Another method proposed by Aron et al. [5] is also applied to update *ssthresh* dynamically. In Aron's method estimates the *ssthresh* value for a given interval time whenever some predetermined conditions are satisfied, and modifies the *ssthresh* value dynamically.

V. Simulation

Simulations of the proposed scheme were developed using the Virtual InterNetwork Testbed (VINT) network simulator (NS) version 2.

In the simulation, the bandwidth and propagation delay of the link between the access router (AR) and the destination change dynamically according to the channel conditions, as shown in Figure 1. 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.

Simulations were conducted for 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. A single TCP flow is considered, with a TCP packet size of 1000 bytes. The TCP variant employed here

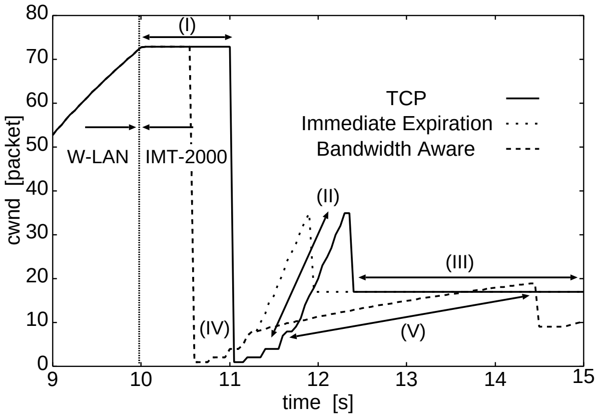

Fig. 2. cwnd (20Mb/s → 384Kb/s, 0.5seconds).

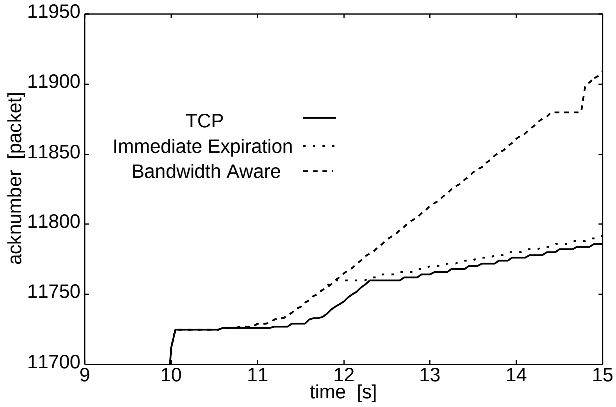

Fig. 3. cwnd (20Mb/s → 384Kb/s, 0.5seconds).

is NewReno, and the TCP traffic considered is greedy file transfer such as ftp-data. The propagation delay and bandwidth change drastically with the change in datalink interface. The propagation delay includes processing delay in the AR and the mobile host.

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.

VI. Results and Discussion

A. Effect of bandwidth change

The first case considered here is when the mobile datalink interface is changed from 20-Mb/s wireless LAN (IEEE802.11a) to 384-kb/s IMT-2000, with a communication interruption time of 0.5 s and a propagation of 1ms (wireless LAN) and 70 ms (IMT-2000).

Figure 2 shows the change in *cwnd* over this period, and Figure 3 shows how the sequence number increases after the datalink interface is changed. Communication under TCP stops at 11 s due to retransmission timeout (I), even though the new datalink interface becomes available at 10.5 s, and 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).

TABLE I

Throughput(20Mb/s → 384Kb/s, 0.5seconds)

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

However, this rapid expansion of *cwnd* results in multiple packet loss if the bandwidth is reduced as in this case. This is shown by 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.

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 as in TCP. This is because Immediate Expiration also uses existing TCP flow 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 prevents excess packet loss after hand-off, allowing *cwnd* to be expanded effectively during slow start. The throughput of the Bandwidth Aware scheme is far greater than either Immediate Expiration or conventional TCP.

Table 1 shows the average throughput of TCP for the period between 10 and 15 s, with throughput normalized against the bottleneck 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 (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 due to a change in the datalink interface in this context.

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

Figure 4 shows how *cwnd* changes after the datalink interface is changed. In this case, TCP is enforced to wait for the expiration of retransmission timeout timer until 11 s (I). After that, packets are first transmitted in the slow start phase and then immediately in the congestion avoidance phase (II). This prevents *cwnd* from increasing efficiently, even though the bandwidth was expanded to 100 Mb/s. The inability for existing TCP to reset *ssthresh* to an appropriate value when the datalink interface is changed leads to an 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 conventional TCP.

In contrast, the Bandwidth Aware scheme sets *ssthresh* to an appropriate value based on the estimated bandwidth

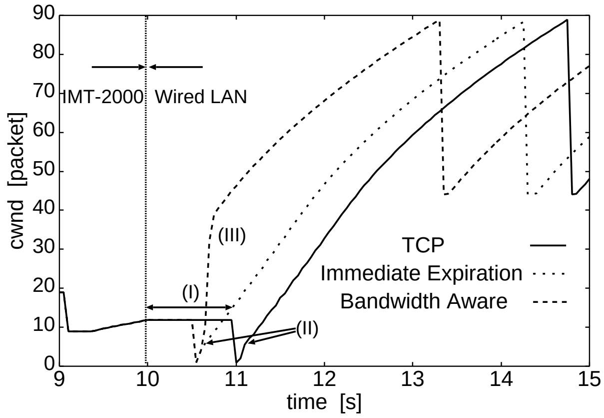

Fig. 4. cwnd(384Kb/s → 100Mb/s, 0.5seconds).

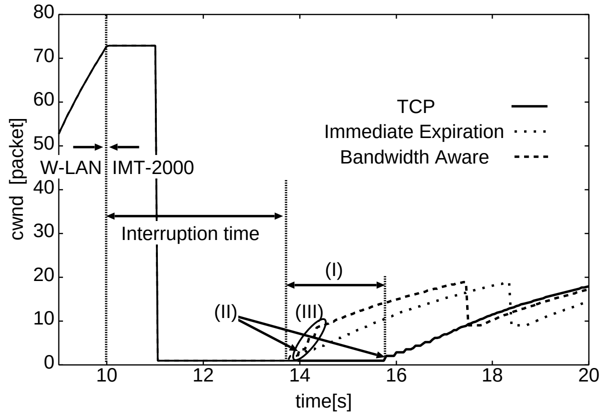

Fig. 5. cwnd(20Mb/s → 384Kb/s,3.5 seconds).

TABLE II
THROUTHPUT(384KB/S → 100MB/S, 0.5SECONDS)

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

TABLE III
THROUGHPUT (20MB/S → 384KB/S, 3.5SECONDS)

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

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 2, with values normalized to a bottleneck bandwidth of 10 Mb/s given in parentheses. The Bandwidth Aware scheme achieves a throughput of 8.218 Mb/s, which is about 1.3 times that of TCP (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.

B. 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) and 70 ms (IMT-2000). This interruption of 3.5 s simulates typical behavior for the current IMT-2000 architecture, which requires a PPP connection to be established to the gateway connecting the radio access network (RAN) and the Internet in advance.

Figure 5 shows the change in *cwnd* after the datalink interface is changed to IMT-2000 with a 3.5 s interruption. Communication under TCP 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. Then, *ssthresh* is reset to the minimum because of the multiple timeouts, resulting in congestion avoidance soon after slow start (II). This results in a very inefficient increase in *cwnd* and poor network performance. 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 in this case. Although performing better than regular TCP with regard to the longer delay, 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, but 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 others schemes.

In terms of the average throughput, the Bandwidth Aware scheme achieves a throughput of 0.213 Mb/s, which is about 1.7 times that of TCP (0.124 Mb/s). However in this case, the improvement offered by the Immediate Expiration scheme is also quite good (0.199 Mb/s).

These results have demonstrated 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 long.

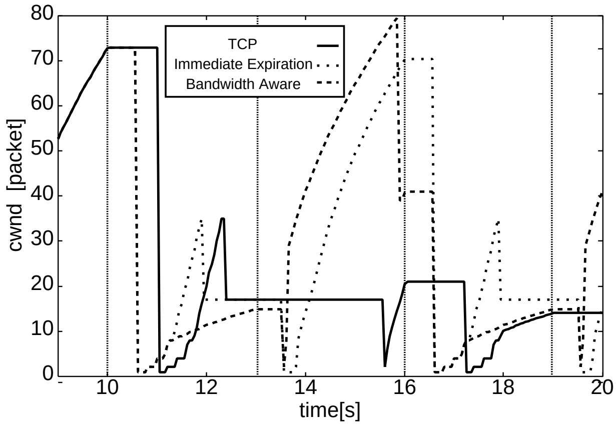

Fig. 6. cwnd (20Mb/s ⇔ 384Kb/s, 0.5 seconds).

TABLE IV
THROUGHPUT (20MB/S ⇔ 384KB/S, 0.5SECONDS).

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

C. Effect of multiple changes

The effect of multiple changes in datanlink 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) and 70 ms (IMT-2000).

Figure 6 shows how *cwnd* changes after the datalink interface is changed. Under TCP, 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, then increases *cwnd* exponentially. However, because of the reduced bandwidth, the scheme suffers multiple packet loss, entering fast recovery under NewReno and retransmitting lost packets just before the second hand-off at 13 s, as indicated by the constant *cwnd*. After the fast recovery algorithm has completed, this scheme starts data transmission by slow start, and *cwnd* is prevented from increasing efficiently. Furthermore, since existing TCP flow control schemes set *ssthresh* to half of the old *cwnd* value after retransmission timeout expires, *ssthresh* will become very small and TCP will thus enter congestion avoidance immediately.

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), then increases *cwnd* exponentially. This results in the improvement in throughput performance as shown in Table 4.

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 shows clearly the effect of the Bandwidth Aware scheme (Table 4). For the period between 10 and 20 s, the ideal throughput is 3.192Mb/s, as calculated by

$$Ideal\,throughput = \frac{sum\,of(BW \times duration)}{simulation\,time}. \quad (3)$$

The Bandwidth Aware scheme achieves a throughput of 2.670 Mb/s, which is about ten times that of TCP (0.253 Mb/s), and also 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.

VII. CONCLUSION

In this paper, two new flow control schemes suitable for future mobile environments in which the datalink interface may be changed dynamically to achieve an effective use of available network resource. 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 by simulation of a range of scenarios. The first control scheme, Immediate Expiration of Timeout Timer, offers only moderate performance improvements over conventional TCP in that the waiting time until start of retransmission is shortened, but subsequent flow control is same as in TCP. The second scheme, Bandwidth Aware Slow Start Threshold, picks up retransmission promptly like in the first scheme and further controls subsequent flow by slow start with an optimized value of *ssthresh* based on estimation of the new bandwidth. The simulations demonstrated that the network performance offered by the Bandwidth Aware scheme is considerably higher than either TCP or Immediate Expiration in situations where the available bandwidth changes drastically due to the datalink interface change, and when the 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.

REFERENCES

[1] C. Perkins, Editor. “IP Mobility Support”, *Networking Group*, RFC2002, Octover, 1996.

[2] Alex C. Snoeren and Hari Balakrishnan, “An End-to-End Approach to Host Mobility,” *ACM/IEEE MobiCom’00*, no. 6, pp. 155-166, Boston, Aug., 2000.

[3] T. Goff, J. Moronski, D. S.Phatak, and V. Gupta,“Freeze-TCP: a True End-to-end TCP Enhancement Mechanism for Mobile Enviroments,”In *Proc. IEEE INFOCOMM’00*, Tel Aviv, Israel, March 2000.

[4] Janey C. Hoe, “Improving the Start-up Behavior of a Congestion Control Scheme for TCP”, in *Proc. ACM SIGCOMM’96*, Aug., 1996, pp. 270-280.

[5] M. Aron, P. Druschel, “TCP:Improving Startup Dynamics by Adaptive Timers and Congestion Control”, *Technical Report TR98-318, Rice Unversity Computer Science*, 1998.