이기종망간의 수직적 핸드오프에 대한 상태전환 방식의 TCP 혼잡제어방안

정회원 석 우 진*, 최 영 환**, 박 귀 순***, 나 지 현****, 종신회원 김 상 하**

TCP Congestion Control based on Context Switch in Heterogeneous Wireless Networks

Woojin Seok*, Younghwan Choi**, Guisoon Park***, Jee-Hyeon Na****, ***Regular Members***,
SangHa Kim** ***Lifelong Member***

요 약

이기종 무선망에서의 수직적 핸드오프는 전송지연시간과 전송대역의 큰 변화를 가져온다. 이로 인하여, 오인된 재전송과 대역의 비효율적 사용을 유발시키며, 결국 TCP 의 성능을 크게 저하시킨다. 본 논문에서는 이러한 문제점을 해결하기 위하여, 각 무선망에서 사용되었던 TCP 상태 변수 값들을 분리하여 저장하는 방법을 제안하였다. 이기종 무선망간의 수직적 핸드오프 시, 해당 무선망에서 사용되어 저장되었던 TCP 상태 변수 값들이, 현재 동작하는 TCP 상태 변수 값들로 전환되어 사용된다. 시뮬레이션 분석을 통하여, 제안된 TCP 혼잡제어가 수직적 핸드오프에 대하여 TCP 성능 저하가 발생하지 않았으며, TCP SACK과 비교하여 전송률에서 좋은 성능을 보여주었다.

Key Words : Vertical Handoff, Tcp, Heterogeneous Networks.

ABSTRACT

The heterogeneous wireless access networks has been envisioned to characterize the future wireless networks. In such environments, TCP(Transmission Control Protocol) has to experience poor end-to-end performance because bandwidth and link delay change suddenly when a mobile node moves over different types of wireless networks, which is called vertical handoff. In this paper, we propose a new TCP which maintains each set of congestion control variables, which we call TCP context, for each type of wireless network. The proposed TCP can switch the TCP context against vertical handoff in order to adjust quickly to a newly arrived network. In simulations, the proposed TCP has higher throughput than TCP SACK(Selective Acknowledgment Options) due to its great features to vertical handoff situations.

Ⅰ. Introduction

In coming near future networking era, multiple different types of wireless networks are expected to be overlaid, with each providing varying access bandwidth and coverage level. A mobile node in a vehicle or a human-hand moves over two or more different types of the wireless networks, and we refer to such a procedure that handles handoff between them as vertical handoff[1-4]. Specially, we call it downward vertical handoff when a mobile node moves from a lower bandwidth

---

* 한국과학기술정보연구원 (wjseok@kisti.re.kr), ** 충남대학교, *** 국방과학연구소, **** 한국전자통신연구원
논문번호 : KICS2007-03-111, 접수일자 : 2007년 3월 7일, 최종논문접수일자 : 2007년 6월 22일

providing network to a higher, and upward vertical handoff when vice versa.

A mobile node has to experience drastic changes of bandwidth and link delay by the vertical handoff. This produces a bunch of packet reordering, link underutilization, packet overflows, and spurious timeout for TCP. This results in serious degradation on end-to-end performance [5-8]. TCP SACK shows better performance than TCP Tahoe and TCP Reno to the vertical handoff due to its great performance against the multiple packet losses that can be produced by lots of packet reordering[9]. Regardless of its superiority to vertical handoffs among TCP variants, TCP SACK cannot avoid completely the problems of link underutilization on downward vertical handoff and spurious timeout on upward vertical handoff, respectively.

To reduce performance degradation against vertical handoff, TCP has to adapt quickly its sending rate to a newly arrived network without producing spurious timeout[10]. proposed to revise TCP receiver. The revised receiver recognizes vertical handoff and regulates the number of ACK(Acknowledgment)s sending back to TCP sender, which results in not only controlling the sending rate from the sender, but also eliminating spurious timeout. To recognize the vertical handoff incident, and adapt quickly to the changed bandwidth, the revised receiver estimates bandwidth exactly, and the estimated bandwidth gives a hint for the vertical handoff incident. However, the estimation of bandwidth which was just assumed in[10] requires long measuring time to be accurate, so much considerations should be required for the estimation.

[8] and [11] proposed to revise TCP sender, and they both used Slow-Start scheme to adapt quickly to a newly arrived network. However, they did not consider SSTHRESH(Slow-Start THRESHOLD) for the newly arrived network, so the Slow-Start without proper SSTHRESH can results in link underutilization or network overload. [12] also proposed to revise TCP sender and use SlowStart scheme. Moreover, they proposed to reconfigure SSTHRESH on every vertical handoff. They estimated bandwidth-delay product to figure out the new SSTHRESH. But they used only a few ACKs to estimate it, so the accuracy could be a problem because it needs many packets with enough time to measure and estimate bandwidth-delay product accurately.

Estimating bandwidth and link delay needs enough time to be accurate, but TCP has to respond very quickly against vertical handoff because the poor end-to-end performance situation occurs immediately after vertical handoff. To meet these requirements, we propose a new TCP, named Context-Switching TCP, which does not estimate bandwidth or link delay to adapt to a newly arrived network. It maintains each set of TCP variables dedicated for each type of wireless networks. If a mobile node moves over two types of wireless networks, for example, two separated sets of TCP variables are maintained.

Context-Switching TCP replaces the running set of TCP variables by the proper set on vertical handoff. Then, it does not suffer from the drastic changes of bandwidth and link delay by the vertical handoff. To detect vertical handoff incident, it uses handoff triggering information directly from link layer. So it can detect the vertical handoff incident promptly and what network interface is turned on exactly. This enables Context-Switching TCP to select the proper set of TCP variables.

Ⅱ. The Proposed TCP

We define a new term in this paper, context, as a set of variables which represent an end-to-end status and are used for TCP congestion control. For example, a pair of CWND(Congestion WiNDow) and RTO(Retransmission TimeOut), two of TCP variables, can be a context.

In the environment where WLANs(Wireless LANs) are overlaid by a cellular network, WLAN context and cellular context can be expressed by an area of CWND-RTT domain shown in Fig.1. WLAN provides higher bandwidth and shorter RTT(Round Trip Time) with smaller coverage than cellular network, so WLAN context locates at left top area and cellular context locates at right bottom area. The two contexts are not quite close with each other, so TCP has to adapt to the extremely different context immediately after downward and upward vertical handoff.

In a conventional wired network, the context in a TCP connection is changed only by network congestion. The context change is recognized by 3 duplicate ACKs or RTO expiration. In heterogeneous wireless networks, vertical handoff causes the total change of the context as shown in Fig.1. However, TCP cannot recognize the vertical handoff, instead it is recognized as the conventional network congestion like 3 duplicate ACKs or RTO expiration. If we can differentiate the context change by vertical handoff from by network congestion, we can avoid unnecessary degradation of TCP performance against vertical handoff.

Context-Switching TCP provides a means to receive a handoff triggering signal from link layer, and it can know where a mobile node stays and which context it has to use there. Then it can change its context between WLAN and cellular network immediately after downward and upward vertical handoff without performance degradation. In this paper, we assumes that we have two overlapping wireless networks, cellular network and WLAN.

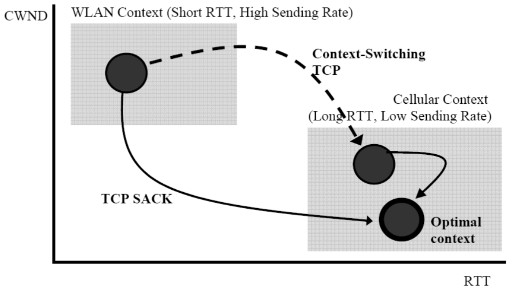


Fig. 1. Upward Vertical Handoff in CWND-RTT Domain: Context-Switching TCP canswitch the context from WLAN to cellular network immediately after upward vertical handoff (dashed line), then it has to adapt itself to cellular network within cellular context (solid line). TCP has to adapt the sudden changes between WLAN context and cellular context (solid line), so it has to experience the poor end-to-end performance on vertical handoff. The same scenario can be applied to downward vertical handoff in opposite direction.

2.1 Context Composition

On a downward vertical handoff, TCP suffers from underutilization, and that is caused by the change of bandwidth. After the downward vertical handoff, the bandwidth a mobile node can achieve is much higher than before, but CWND is still optimized to cellular network. So it cannot utilize the bandwidth as much as WLAN can provide. On an upward vertical handoff, TCP suffers from spurious timeout and buffer overload, and those are caused by the changes of RTT and bandwidth. The prolonged RTT causes RTO expiration, so TCP has to begin Slow-Start phase. During RTO period and Slow-Start phase, TCP cannot send data as much as the cellular network can provide. Based on the negative impact of vertical handoff on TCP performance, the context should be comprised of following three items.

- CWND: After upward or downward vertical handoff, bandwidth becomes much smaller or larger than before, respectively. But CWND of TCP sender is optimized for the bandwidth of previously stayed network. This causes buffer overload or underutilization. Therefore, CWND should be considered as a component of context.
- sRTTand RTTvar: After upward vertical handoff, the drastic prolonged RTT causes RTO expiration. RTO calculation is based on sRTT (smoothed RTT) and RTTvar(RTT variation) which are the average and the variation of measured RTT[13]. To prevent RTO expiration, sRTT and RTTvar should be replaced by the proper ones immediately after upward vertical handoff. However, there is no need to consider RTO expiration on downward vertical handoff. Therefore, sRTT and RTTvar are considered as a component of context only for upward vertical handoff.

- SSTHRESH: While a mobile node stays at a new wireless network after a vertical handoff, it may experience packet losses and RTO expiration with SSTHRESH still optimized to the previously arrived network. Then Slow-Start after the RTO expiration may overload or underutilize the newly arrived network due to the wrong SSTHRESH. Therefore, SSTHRESH should be considered as a component of context.

As a result, WLAN context is comprised of CWND and SSTHRESH, and cellular context is comprised of CWND, sRTT and RTTvar, and SSTHRESH. They should be replaced by the proper ones at the moment of vertical handoff.

2.2 The Protocol Structure

Fig. 2 shows the proposed structure of Context-Switching TCP. It consists of WLAN context and cellular context, Vertical Handoff Management, and existing TCP variables. Context-Switching TCP uses handoff triggering information of link layer. Without the information, it cannot tell handoff incidents from network congestions. With the information, it can detect a handoff incident exactly and can differentiate them. Link layer and transport layer can communicate with a shared memory, else other inter-process communication methods, or interrupt-driven notification methods. We leave it without proposed in this paper because any possible way can be used for this purpose.

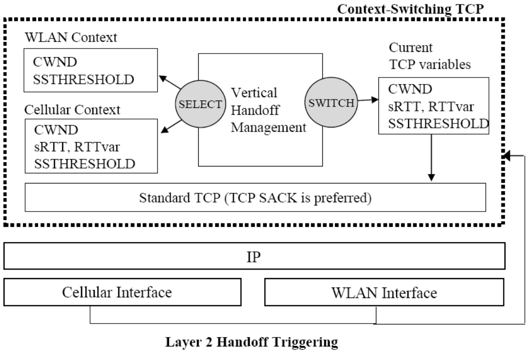


Fig. 2. The Structure of Context-Switching TCP: When a mobile node stays at cellular network, for example, cellular interface turns on and Context-Switching TCP uses cellular context for congestion control later on. One context per a network interface is maintained separately.

When a mobile node moves over cellular network and WLAN, the cellular interface or the WLAN interface in the mobile node detects a handoff and notifies it to Context-Switching TCP. Using the handoff triggering information, Context-Switching TCP can select the proper context for a newly arrived network. That is, a mobile node will use the recorded cellular context when it arrives at a cellular network, or the recorded WLAN context at a WLAN. The number of context to be maintained depends on the number of types of wireless networks over which a mobile node moves.

Vertical Handoff Management has two functions, SELECT and SWITCH. SELECT function selects the proper context with the received layer 2 handoff triggering. SWITCH function replaces current running TCP variables by the selected context. If it is the first visit to cellular network or WLAN, Context-Switching TCP does not have the corresponding context yet. Without it, Context-Switching TCP has exactly the same behavior with TCP SACK. On the second or subsequent visits to the networks, Context-Switching TCP can retrieve the context recorded, and it is expected to achieve performance gain because no RTO expiration occurs and CWND begins increasing from the proper context.

We also need to consider the way of notifying a handoff incident to TCP sender if a mobile node is TCP receiver. In[5], they proposed to use TCP option field for this purpose. Using 2 bits of the option field, a mobile node can notify 3 states such as normal state, horizontal handoff, and vertical handoff to the sender. However, this idea cannot distinguish a downward vertical handoff and an upward vertical handoff. In[11], they proposed to monitor and estimate the link capacity to detect a handoff incident, and also proposed to use ACKs to piggyback the detected incident to the sender. However, this idea cannot distinguish a handoff from network congestion exactly and promptly because using traffic monitoring to estimate the link status needs long time for accurate decision.

```
if detecting handoff signal from cellular interface, OR
if receiving ACK piggybacking upward handoff from a receiver
    (recording the current status for WLAN context)
    $CWND_{WLAN} = CWND$
    $SSTHRESHOLD_{WLAN} = SSTHRESHOLD$
    if cellular context exists
        (retrieving the recorded context of cellular network)
        $CWND = CWND_{CELL}$
        $SSTHRESHOLD = SSTHRESHOLD_{CELL}$
        $RTTvar = RTTvar_{CELL}$, $sRTT = sRTT_{CELL}$

if no wireless network interfaces existing, AND
if receiving ACK piggybacking downward handoff from a receiver,
OR
if detecting handoff signal from WLAN interface, AND
if receiving ACK piggybacking downward handoff from a receiver,
    (recording the current status for cellular context)
    $CWND_{CELL} = CWND$
    $SSTHRESHOLD_{CELL} = SSTHRESHOLD$
    $RTTvar_{CELL} = RTTvar$, $sRTT_{CELL} = sRTT$
    if WLAN context exists
        (retrieving the recorded context of WLAN)
        $CWND = CWND_{WLAN}$
        $SSTHRESHOLD = SSTHRESHOLD_{WLAN}$
```

Fig. 3. The Pseudo Code of Context-Switching TCP at Sender

```
if normal state, (meaning 'no handoff')
ACK piggybacks the option field with 00.

if detecting signal from cellular interface, (meaning 'upward vertical handoff')
ACK piggybacks the option field with 01.

if detecting signal from WLAN interface, (meaning 'downward vertical handoff')
ACK piggybacks the option field with 10.
```

Fig. 4. The Pseudo Code of Context-Switching TCP at Receiver

We propose to use 2 bits of TCP option field as[5] did to notify handoff incidents to TCP sender.

We propose to use 3 states such as normal state, downward handoff, and upward handoff. The 2 states, downward and upward vertical handoff, can give a decision to select which context should be selected.

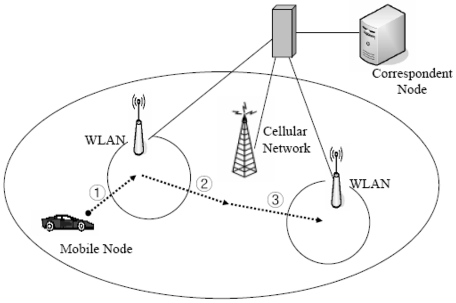


Fig. 5. Simulation Topology and Scenario

Additionally, we need to consider the case that both TCP sender and receiver can move and issue vertical handoff. The sender receives ACK piggybacking the notification of downward vertical handoff from the receiver, but the sender still stays at cellular network, or moves to cellular network. Then the sender of Context-Switching TCP has to select cellular context rather than WLAN context. For this purpose, Vertical Handoff Management of Context-Switching TCP has to decide to select WLAN context only when the sender and the receiver of Context-Switching TCP both stay at WLAN. The pseudo code of Context-Switching TCP is described in Fig.3 and Fig. 4.

Ⅲ. Performance Analysis

In this section, we evaluate the performance of Context-Switching TCP by comparing with TCP SACK. The adaptation of CWND and RTO against vertical handoffs are analyzed, and each throughput achieved at WLAN and cellular network is compared with TCP SACK.

3.1 Simulation Environment

Fig.5 shows the topology of nodes and the scenario used in the simulations in this section. For the data rate and latency within cellular network, CDMA2000 1X increase data rates to as much as 614Kbps. The first implementation of CDMA2000 1X is expected to yield data rates up to 153.6 Kbps. The data rate within GPRS cellular network is between 30 and 50 Kbps, as mentioned in[7]. 144 Kbps is used as the data rate of cellular network in[8]. The latency is mentioned between 500 and 3000ms as round trip time in[7], and 300ms is used for the simulation in[8]. For the data rate and latency within WLAN are mention between 5 and 7Mbps, and between 5 and 40ms, respectively, in[7]. 2Mbps and 100ms are used for data rate and latency of the simulation in[8].

From this previous uses for the data rate and latency, We suppose the wireless link of cellular network is 600 ms latency and 300 Kbps data rate, and the wireless link of WLAN is 60 ms latency and 10 Mbps data rate. This cannot emulate cellular network and WLAN exactly, but could be enough for the comparison with each other as simulation model. All other wired links are supposed to be 10 ms latency and 100 Mbps data rate. The maximum segment size is 1460 bytes which is the most popular used value. The simulations are done with ns-2.

With the topology, we have a scenario that a mobile node downloads data from a Correspondent node while it moves from a cellular network into a WLAN overlaid by the cellular network, then moves back to the cellular network, and finally gets to the WLAN in the coverage of the same cellular network. During this period, the mobile node issues one upward and two downward vertical handoffs while the TCP connection is sustained. The durations are all 60 seconds for the cellular network and the WLAN. The achievable bandwidth for the second visit to WLAN and cellular network is the same with the first one. In the subsection 3.3, we analyze the situation that the achievable bandwidth at the second visit is different with the first one.

3.2 Result and Discussion

Fig.6 shows CWND curves. In T1 and T2 when a mobile node visits a cellular network and a WLAN firstly, Context-Switching TCP and TCP SACK show the same shape. While TCP SACK just leaves the cellular network and the WLAN, Context-Switching TCP records the last contexts of the networks immediately before leaving, and uses them for next subsequent visits to them.

In the second visit to cellular network (at T3), TCP SACK goes into Slow-Start phase due to RTO expiration because RTO must be optimized to that of WLAN at T2, and RTT received at T3 should be much larger than the RTO. On the other hand, Context-Switching TCP can continue the last state of the first cellular network by retrieving the recorded context of the cellular network, so it does not experience RTO expiration(by change of sRTT and RTTvar) and can shrink sending rate promptly(by change of CWND). In the second visit to WLAN (at T4), TCP SACK linearly increases CWND from very low value, but Context-Switching TCP can continue the last high CWND(by change of CWND). The Fig.7 shows the RTO curves. sRTT and RTTvar are used to calculate RTO value, which is shown in Eq.(1), where α is 1/8, β is 1/4, and k is 4 [13,14].

$$\begin{aligned} &RTO = sRTT + k \times RTTvar && (1) \\ &RTTvar = (1-\beta) \times RTTvar + \beta \times |sRTT - RTT| \\ &sRTT = (1-\alpha) \times sRTT + \alpha \times RTT \end{aligned}$$

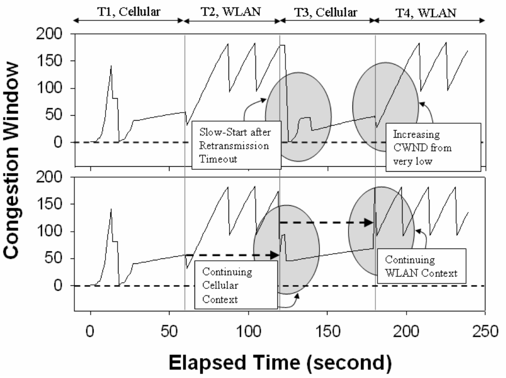


Fig. 6. CWND Curves of TCP SACK(upper) and Context-Switching TCP(lower)

TCP SACK experiences RTO expiration(noted by “Spurious Timeout Occurs” in Fig.7) immediately after a mobile node issues an upward vertical handoff. When a mobile node moves to the second cellular network, RTT becomes much long by the inherent delay of the cellular network. To prevent the RTO expiration, RTO should be large suddenly to wait for ACK segments that could arrive but takes long latency due to the suddenly changed path. But sRTT and RTTvar cannot become large suddenly enough to make RTO to be large as shown in Fig.7. On the other hand, Context-Switching TCP does not have any RTO expiration. It records the last sRTT and RTTvar when it leaves cellular network. Those two values are retrieved when it moves back to the cellular network, and they prevent RTO expiration.

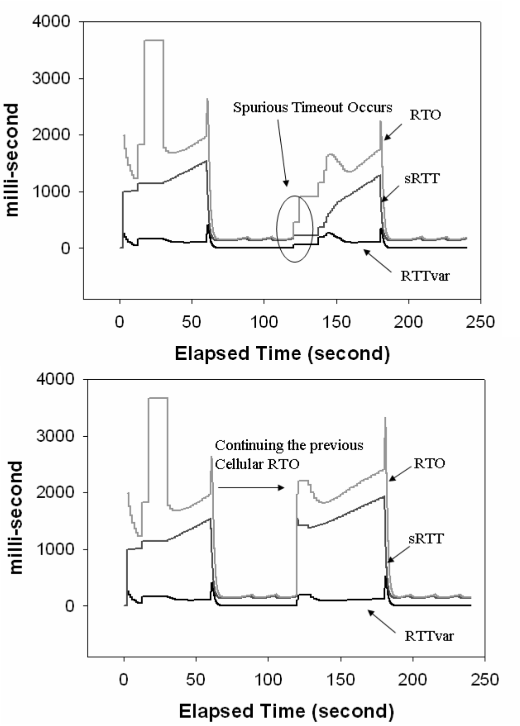


Fig. 7. RTO Curves of TCP SACK(upper) and Context-Switching TCP(lower): TCP SACK has a spurious RTO expiration, but Context-Switching TCP has no RTO expiration for the same situation.

The end-to-end throughput is shown in Table 1. Context-Switching TCP has no performance gain in the first cellular network and WLAN because it is the first visit and there is no context recorded yet. Context-Switching TCP, however, has 21.4% and 10.0% performance gain for the second visits to cellular network and WLAN respectively. Using the recorded context, Context-Switching TCP can adapt itself quickly to subsequently visited networks.

Fig. 8 shows TCP throughput to handoff frequency. The upper figure is the achieved TCP throughput after downward vertical handoffs, and the lower figure is after upward vertical handoffs. Context-Switching TCP has higher performance gain for more frequent handoffs. As handoff frequency increases from 0.5 to 2.0, the performance gain increases from about 5% to about 22% for downward vertical handoffs, and from about 8% to about 63% for upward vertical handoffs. The performance gain increases almost linearly as shown in Fig. 8. That is, the gap between two curves in two figures gets bigger linearly. A mobile node using Context-Switching TCP is expected to have linearly higher performance gain in a situation that needs more frequent handoffs. This means that Context-Switching TCP will be used beneficially for mobile nodes moving fast over heterogeneous wireless networks.

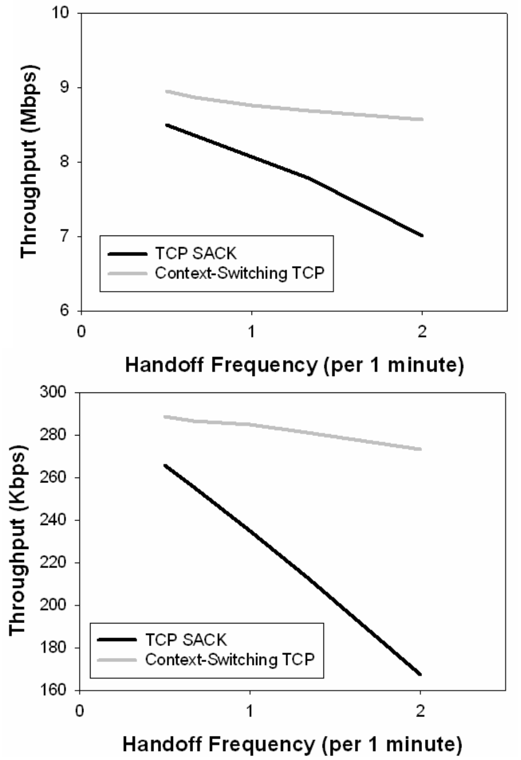


Fig. 8. Throughput Curves of TCP SACK and Context-Switching TCP

Table 1. Performance Gain of Context-Switching TCP; Gain means how much Context-Switching TCP earns more than TCP SACK. It is expressed by the percentage(%) of bytes transferred more than TCP SACK by Context-Switching TCP over the bytes transferred by Context-Switching TCP.

	Cellular	WLAN	Cellular	WLAN
TCP SACK	218.4K	8.15M	234.7K	8.06M
Context-Switching TCP	218.4K	8.15M	284.9K	8.87M
Gain(%)	0	0	21.4%	10.0%

3.3 Validation of Reuse of Recorded TCP context

The achievable bandwidth at WLAN depends on the distance between AP(Access Point) and a mobile node, how many other nodes are trying to send data or requesting bandwidth, and the version of 802.11[15,16]. And the bandwidth assigned in the cellular network is no longer fixed but is variable and is dependent on the channel allocation policy in the radio network controller and the condition of radio links, like EGPRS(Enhanced General Packet Radio Services) that offers packet-switched data transmission with rates ranging from 8.8 kbps to 59.2 kbps per channel by choosing appropriate coding schemes according to the condition of the radio link[17].

The recorded TCP context does not exactly represent the network condition at the moment on arriving at wireless network after vertical handoff. However, applying recorded TCP context is much better than just continuing the last TCP context of different type of wireless network. The difference between two TCP contexts before and after vertical handoff is so huge that it produces link underutilization on downward vertical handoff and RTO expiration on upward vertical handoff.

We vary achievable bandwidth at WLAN and measure TCP throughput. After downward vertical handoff, Context-Switching TCP can continue the last WLAN context by retrieving the recorded WLAN context. But the newly arrived WLAN can provide higher bandwidth or lower bandwidth than the bandwidth which can be expected by the recorded WLAN context. The throughputs which TCP SACK and Context-Switching TCP can achieve for the differently provided bandwidth are shown in Fig. 9. The achieved bandwidth at the previously arrived WLAN is 10Mbps.

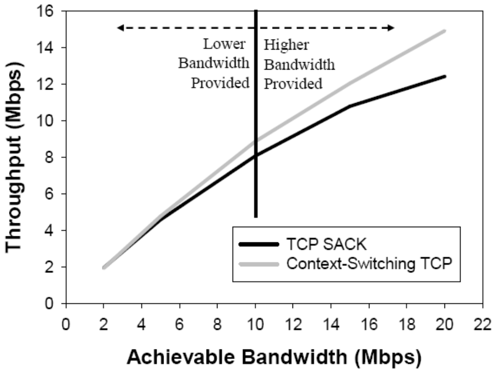


Fig. 9. Throughput Curves of TCP SACK and Context-Switching TCP

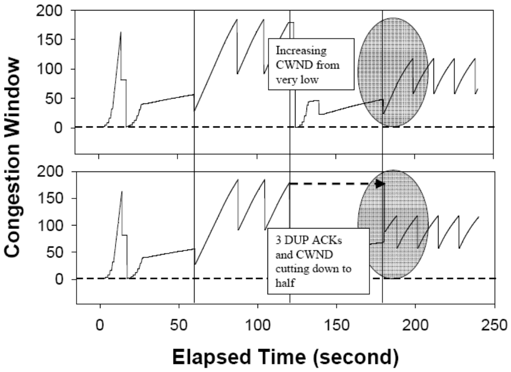


Fig. 10. CWND curves of TCP SACK(upper) and Context-Switching TCP(lower): A mobile node arrives a WLAN proving lower bandwidth(5Mbps) than that of previously arrived WLAN(10Mbps).

In case that the achievable bandwidth becomes higher than the last WLAN context, Context-Switching TCP can earn more throughput gain because TCP SACK requires longer time for CWND to get to the higher achievable bandwidth while Context-Switching TCP does not need the additional time to get to the higher achievable bandwidth. The throughput gap between Context-Switching TCP and TCP SACK becomes much bigger as the achievable bandwidth becomes higher.

In case that the achievable bandwidth becomes lower than the last WLAN context, Context-Switching TCP makes WLAN congested because it sends more segments than the achievable bandwidth. This results in 3 dupicate ACKs and cutting down of CWND by half as shown Fig. 10. Even though Context-Switching TCP experiences the congestion and cutting down of CWND, its performance still shows better than or at least equal to TCP SACK as shown in Fig. 9.

As a result, a mobile node using Context-Switching TCP is expected to have much higher performance gain in the situation that WLAN can provide more bandwidth than previously arrived WLAN, and also expected to have better or at least equal performance even in the situation that WLAN can provide lower bandwidth than previously arrived WLAN. The same results are expected for cellular network.

Ⅳ. Conclusions

In heterogeneous wireless networks, vertical handoff makes the performance of TCP degraded due to the sudden changes of bandwidth and link delay in an end-to-end path. In this paper, we propose a new TCP, named Context-Switching TCP, that can switch promptly its congestion controlling variables to adapt quickly to a newly arrived network after vertical handoff. Context-Switching TCP has better performance than TCP SACK for both downward and upward vertical handoff because it does not experience link underutilization, network overload, and retransmission timeout which TCP SACK has to experience. Context-Switching TCP achieves linearly higher performance gain as the frequency of handoff incidents increases. We can expect from the results that Context-Switching TCP will be used beneficially for mobile nodes moving fast over heterogeneous wireless networks. We also show that its performance is better than or at least equal to TCP SACK in the situation that it can be provided different bandwidth at WLAN from the previously arrived WLAN.

References

〔1〕 Q.Zhang, C.Guo, Z.Guo, and W.Zhu, "Efficient Mobility Management for Vertical Handoff between WWAN and WLAN," IEEE Communication Magazine, pp.102-108, November 2003.

〔2〕 J.McNair and F.Zhu, "Vertical Handoffs in Fourth-generation Multinetwork Environments," IEEE Wireless Communication, vol.11, no.3, pp.8-15, June 2004.

〔3〕 Q.Song and A.Jamlipour, "A Network Selection Mechanism for Next Generation Networks," IEEE ICC, vol.2, pp.1418-1422, May 2005.

〔4〕 W.Chen and Y.Shu, "Active Application Oriented Vertical Handoff in Next Generation Wireless Networks," IEEE WCNC, vol.3, pp.1383-1388, March 2005.

〔5〕 S.E.Kim and J.A.Copeland, "TCP for Seamless Vertical Handoff in Hybrid Mobile Data Networks," IEEE GLOBECOM, no.1, pp.661-665, December 2003.

〔6〕 A.Gurtov and J.Korhonen, "Effect of vertical handovers on performance of TCP-friendly rate control," ACM Mobile Computing and Communications Review, vol.8, no.3, pp.73-87, July 2004.

〔7〕 R.Chakravorty, P.Vidales, K.Subramanian, I.Pratt, and J.Crowcroft, "Performance Issues with Vertical Handovers - Experiences from GPRS Cellular and Wireless LAN Hot-spots Integration," IEEE PERCOM, pp.155-164, March 2004.

〔8〕 S.E.Kim and J.A.Copeland, "Interworking Between WLANs and 3G Networks: TCP Challenges," IEEE WCNC, vol.2, pp.1252-1257, March 2004.

〔9〕 W.Hansmann, M.Frank, and M.Wolf, "Performance Analysis of TCP Handover in a Wireless / Mobile Multi-Radio Environment," IEEE LCN, pp.585-594, November 2002.

〔10〕 Y.Matsushita, T.Matsuda, and M.Yamamoto, "TCP Congestion Control with ACK-Pacing for Vertical Handover," IEEE WCNC, pp.1497-1502, 2005.

〔11〕 L.J.Chen, G.Yang, T.Sun, M.Y.Sanadidi, and M.Gerla, "Enhancing QoS Support for Vertical Handoffs Using Implicit/Explicit Handoff Notifications," IEEE QSHINE, p.37, August 2005.

〔12〕 K.Tsukamoto, Y.Fukuda, Y.Hori, and Y.Oie, "New TCP Congestion Control Schemes for Multimodal Mobile Hosts," IEICE Trasaction of Communications, vol.E89-B, no.6, pp.1825-1836, June 2006.

〔13〕 V.Paxson and M.Allman, "Computing TCP's Retransmission Timer," RFC 2988, November 2000.

〔14〕 Jacobson, V.M.Karels, "Congestion avaoidance and control," ACM Computer Communication Review, vol.18, no.4, p.314-329, August 1988.

〔15〕 M.Heusse, F.Rousseau, G.Berger-Sabbatel, and A.Duda, "Performance Anomaly of 802.11b," IEEE INFOCOM, pp.836-843, March 2003.

〔16〕 Hayes and S.Ian, "Just Enough Wireless Computing," Prentice Hall PTR, pp. 305-309, August 2002.

〔17〕 J. Cai, L.-F. Chang, K. Chawla and X. Qiu, "Providing Differentiated Services in EGPRS Through Packet Scheduling," IEEE GLOBECOM, vol.3, pp.1515-1521, November-December 2000.

석 우 진 (Woojin Seok) 정회원

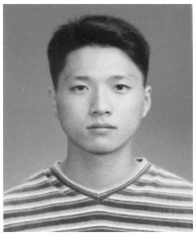
1996년 경북대학교 컴퓨터공학과 학사

2003년 Univ. North Carolina, Computer Science 석사

2003년~현재 한국과학기술정보연구원 선임연구원

<관심분야> 무선/이동 QoS, TCP 성능 분석

최 영 환 (Younghwan Choi) 정회원



2002년 충남대학교 컴퓨터과학과 학사

2005년 충남대학교 컴퓨터과학과 석사

현재 충남대학교 컴퓨터공학과 박사 과정 재학

<관심분야> 네트워크 이동성, 무선 멀티캐스팅, 센서망

박 귀 순 (Guisoon Park) 정회원

1986년 부산대학교 계산통계학과 학사

1988년 부산대학교 계산통계학과 석사

1991년 2월~현재 국방과학연구소 책임연구원

<관심분야> 무선/이동 QoS

나 지 현 (Jee-Hyeon Na) 정회원

1989년 전남대학교 전산통계학과 학사

2000년 충남대학교 컴퓨터과학과 석사

1989년~현재 한국전자통신연구원 책임연구원

<관심분야> 이동성 관리, Mobile IP, Paging, Ad hoc 네크워크,

김 상 하 (SangHa Kim) 종신회원

1980년 서울대학교 화학과 학사

1984년 University of Huston 석사

1989년 University of Huston 박사

1992년~현재 충남대학교 정보통신공학부 교수

<관심분야> 무선/이동 QoS, 무선 멀티캐스트