Enhancing QoS Support for Vertical Handoffs Using Implicit/Explicit Handoff Notifications[*]

Ling-Jyh Chen, Guang Yang, Tony Sun, M. Y. Sanadidi, and Mario Gerla
Department of Computer Science
University of California, Los Angeles
{cclljj,yangg,tonysun,medy,gerla}@CS.UCLA.EDU

Abstract

Vertical handoffs between different wireless technologies usually lead to dramatic changes in the link capacity. A successful QoS solution for vertical handoffs must be able to fast track the capacity changes and agilely adapt the delivery rates and qualities of the ongoing applications. Though traditional AIMD-based source adaptation schemes (as found in TCP, TFRC, etc.) have been well designed for mild, gradual rate adjustments required by load fluctuations and network congestion, their response time is inadequate when the rate must be adjusted to the drastic network capacity changes that are typical in vertical handoff scenarios. To expedite the response to such changes, we propose in this paper two adaptive algorithms, named the Fast Rate Adaptation (FRA) and Early Rate Reduction (ERR), that are launched when the handoff is from low to high capacity (LOW-to-HIGH) or from high to low capacity (HIGH-to-LOW), respectively. We also propose two vertical handoff notification mechanisms to work with FRA and/or ERR, i.e. the Implicit Handoff Notification (IHN) and Explicit Handoff Notification (EHN). We show by simulation that our proposed schemes are able to provide better QoS support than the traditional AIMD based schemes during vertical handoffs.

1. Introduction

With the emerging wireless and mobile network applications, QoS support has become increasingly important in the last few years. Network applications, with QoS support enabled, are able to better utilize the network resources (e.g. best-effort data transfer), and optimize the user-perceived service qualities (e.g. real-time streaming). To achieve this goal, the system is required to be capable of detecting and adequately responding (or adapting) to the network resource changes. A prominent and well studied example of adaptive scheme is the Additive Increase Multiplicative Decrease (AIMD) algorithm [7]. In TCP, the sender adapts its congestion window size in AIMD fashion in reaction to the dynamic load (signaled by buffer overflow and packet loss) on the path between sender and receiver. More recently, TFRC has been proposed to provide smoother rate control than TCP while adjusting its sending rate according to an equation that mimics the long-term throughput of TCP [10]. The simplicity and elegance of the AIMD principle has contributed significantly to the stability of the Internet.

AIMD and its compatible derivatives have implicitly assumed that changes in available bandwidth are only due to changes in channel load and can be effectively measured by buffer occupancy and packet loss (or lack of it). Such assumption is usually true in fixed and wired networks. However, with the emerging mobile and wireless network applications, the path characteristics encountered by a sender (e.g. end-to-end path capacity and delay) may change dramatically and very rapidly due to vertical handoffs [29], wireless auto-rate adaptations [14, 17], or route changes. In such scenarios, traditional schemes are not able to react fast enough to the rapid changes in network resources. As a result, they behave too conservatively in utilizing the dramatically increased network resources, and conversely fail to respond to the resource reduction in a timely manner.

Therefore, in scenarios where network resources change drastically due to causes other than congestion or random loss, a more agile solution is needed. This is challenging: on the one hand, the solution should respond to changes in network resources promptly and accurately; on the other hand, it must also maintain the stability and inter/intra-protocol fairness as legacy AIMD-based protocols do. In this paper we focus on vertical handoff scenarios and propose an agile solution to improve the quality-of-service when handoffs occur.

We use TCP and TFRC as examples of adaptively controlled flows in this example. While the main motivation in

[*] This material is based upon work supported by the National Science Foundation under Grant No. ANI-0335302 and CNS-0435515.

this paper is to study agile controls for TFRC, we are also including a parallel study on TCP since TFRC is modeled on the behavior of the latter, thus a comparison is appropriate. We propose two algorithms:

1. the Implicit Handoff Notification (IHN) algorithm which detects the occurrences of vertical handoffs by passively monitoring the link capacity with end-to-end estimation tools (e.g. TCP Probe and TFRC Probe).

2. the Explicit Handoff Notification (EHN) algorithm, which relies on an intelligent handoff manager (such as proposed in [4]) to monitor the physical link characteristics and trigger handoff events. EHN explicitly notifies ongoing application senders when a handoff event is generated by the manager.

In parallel with IHN and EHN, we propose a Fast Rate Adaptation (FRA) algorithm that is triggered when the handoff is from low to high capacity (denoted as LOW-to-HIGH). The aim is to promptly utilize the newly materialized capacity. When the handoff is from high to low capacity (denoted as HIGH-to-LOW), another algorithm called Early Rate Reduction (ERR) is launched to prevent bulk packet losses. Using simulation experiments, we show that the proposed algorithms are able to provide better QoS support during vertical handoffs.

The rest of the paper is organized as follows. In section 2, we present an overview of vertical handoffs and summarize related work on service agility schemes. In section 3, we recapitulate the passive capacity monitoring tools, namely TCP Probe and TFRC Probe, and present the IHN with FRA algorithm. In section 4, we present the ERR and FRA algorithms with the EHN. The simulation experiments are presented in section 5, and section 6 concludes the paper.

2. Background and Overview

2.1. Vertical Handoff

Handoff occurs when the user switches between different network access points. Depending on the wireless technologies involved in the process, the handoff can be characterized as either vertical or horizontal [29], as depicted in Fig. 1. A vertical handoff involves two different network interfaces for different wireless technologies. For example, when a mobile device moves out of an 802.11b network and into a 1xRTT network, the handoff is considered vertical. A horizontal handoff occurs between two network access points that use the same wireless technology and interface. For example, when a mobile device moves between two 802.11b network domains, the handoff is considered horizontal.

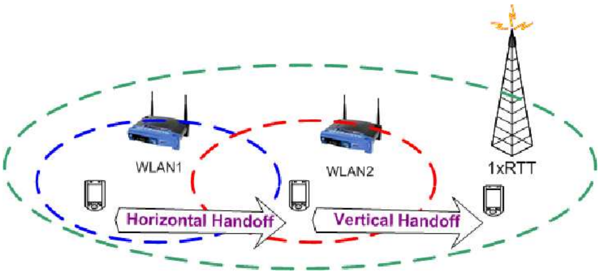


Figure 1. Illustration of horizontal and vertical handoff

Various vertical handoff solutions have been proposed [9, 15, 16, 19]. These proposals can be classified into two categories: network layer approaches and upper layer approaches. Network layer approaches are typically based on IPv6 [8] or Mobile IPv4 [23] standards, requiring the deployment of agents on the Internet for relaying and/or redirecting the data to the moving host (MH). Most upper layer approaches implement a session layer above transport, making connection changes at underlying layers transparent to the application [11, 13, 20, 26, 27]. Other upper layer approaches suggest new transport layer protocols (e.g. SCTP [30]) or modifications to existing transport layer protocols (e.g. TCP-MH [21] and TCP Migrate [28]) to provide necessary handoff support.

Further detailed discussion on vertical handoff solutions is beyond the scope of this paper. Hereafter we assume that a seamless vertical handoff solution is deployed on the mobile system, and assume that the latency caused by handoff is negligible.

2.2. Service Agility

Service agility is a key property of an adaptive system: it enables the system to quickly detect and respond to changes in network resource availability [22]. Though a less agile system may suffice in a more stable environment (e.g. leased lines or enterprise intranets), a highly agile system is necessary for effective operations in volatile networks that may experience large and erratic resource changes (e.g. mobile systems or wireless networks).

Service agility can be implemented using end-to-end feedback control. Additive Increase Multiplicative Decrease (AIMD) is the most popular control mechanism due to its stability properties. Numerous AIMD based protocols have been designed and deployed on the Internet, TCP [25] being the most prominent example,. Recently, TFRC [10] is gaining popularity as a smooth rate control mechanism suitable to multimedia applications. The core equation in TFRC mimics TCP Reno behavior, thus it can also be classified as AIMD compliant.

It turns out that the AIMD based input rate controls currently implemented in the Internet function effectively only when changes of network resources are caused by congestion or mild random losses. When a dramatic change of network resources occurs, e.g. a vertical handoff from 1xRTT to 802.11b where the link capacity changes from 150Kbps to around 5Mbps, AIMD reacts very slowly [2] and the performance is poor [12]. A new solution is needed to provide high agility for applications in such scenarios.

3. Proposed Approach - I: Implicit Handoff Notification

In this section, we propose using Implicit Handoff Notification (IHN) to provide service agility in vertical handoffs. IHN is based on the observation that *a vertical handoff usually results in a drastic change in the path properties (e.g. link capacity and delay).* By directly monitoring the capacity of the wireless link, one can "classify" and detect a vertical handoff when the capacity change is above a certain threshold. In the following subsections, we recapitulate two recently designed passive capacity monitoring techniques, namely TFRC Probe [5] and TCP Probe [24], and present the "Fast Rate Adaptation" (FRA) algorithm which is able to better react to the drastic capacity changes during vertical handoffs.

3.1. TFRC Probe

TFRC Probe [5] is an improved version of TFRC relying on CapProbe techniques [18] to passively monitor and estimate the bottleneck link capacity. Fig. 2 illustrates the differences between the original TFRC and TFRC Probe. In the original TFRC (Fig. 2-a), transmission of data packets is paced and evenly distributed. This is beneficial to multimedia applications which require a smooth sending rate. However, CapProbe-based estimation requires that packets be sent back to back (packet pairs). In order to perform such estimation, we have modified TFRC such that after every n-th data packet is sent out, the TFRC Probe sender immediately transmits the next data packet without waiting for the pacing interval (Fig. 2-b). In other words, TFRC Probe creates a back-to-back sampling packet pair every n packets. The default value of n is set to 20 in our experiments.

In order to achieve *one-way* capacity estimation, the back-to-back sampling packets are time-stamped with the sending time (T_0). Upon their receipt, the TFRC Probe receiver measures the *one-way delay* of each packet in the pair (T_1 and T_2) by subtracting T0 from its respective receiving time. The dispersion ($T_2 - T_1$) and delay sum ($T_2 + T_1$) are then calculated, and the capacity estimate is calculated following CapProbe [18]. The capacity estimation results can be reported to the sender using either the original ACKs or "out-of-band" reporting packets. In our implementation we chose to use ACKs, incurring no extra traffic overhead.

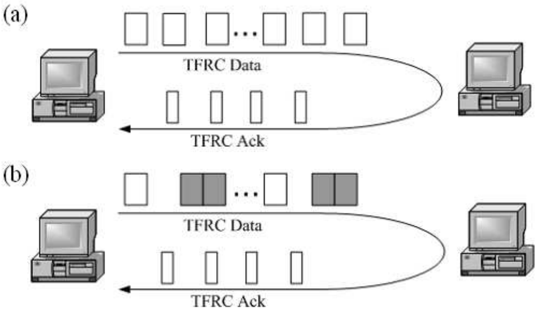


Figure 2. Original TFRC and TFRC Probe

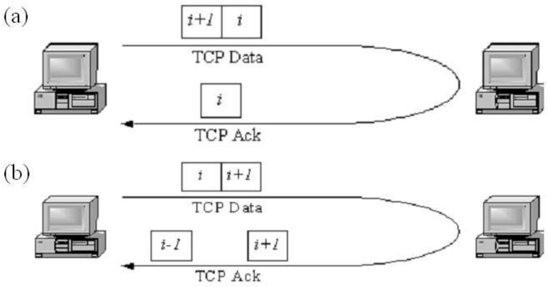


Figure 3. Original TCP and TCP Probe

3.2. TCP Probe

TCP Probe [24] has been proposed to passively monitor and estimate the bottleneck link capacity within TCP protocol. The design of TCP Probe is based on the observation that TCP does occasionally send back-to-back data packets, mostly in Slow Start but also in Congestion Avoidance. These instances are often sufficient to produce an adequate number of back to back packet pair samples for capacity estimation.

However, in order to make the modifications required in TCP Probe to be sender-side only, the capacity estimation of TCP Probe must be round-trip. As a result, two problems must be addressed. One is the result of the often deployed "Delayed ACK" (DelACK) option [3], the other is the result of the different sizes of TCP data packets and ACK packets.

A DelACK enabled TCP receiver may acknowledge every other data packet. Therefore if two TCP data packets i and $i + 1$ are sent back-to-back from the sender, a single DelACK may be sent after one of the two packets, as shown in Fig. 3-a. This coalescence of ACKs disables the sender ability to estimate the path capacity. This problem can be easily solved by using the "inverted packet-pair" technique. When TCP, for capacity estimation purposes, needs to send back-to-back data packets with sequence numbers i and $i + 1$, it swaps their order, i.e., packet '$i + 1$' is sent before packet 'i'. This will generate back-to-back ACKs on sequence numbers $i - 1$ and $i + 1$. The DelACK receiver is, thus, forced to send an individual ACK for each data packet, as shown in Fig. 3-b. This enhancement is applicable to all TCP variants.

The second problem stems from the fact that data and ACK packet sizes are different. This does not comply with the original CapProbe algorithm where packet pairs in both directions are equal. This problem has been addressed in [6]. To summarize, denote the capacities of the bottleneck links on the forward and backward paths by C_1 and C_2, respectively. Assume that TCP data packets are 1500 bytes, and that ACKs are 40 bytes each. TCP Probe can correctly measure the forward link capacity when $\frac{C_1}{C_2} < \frac{1500}{40} = 37.5$ or the backward link capacity when $\frac{C_1}{C_2} \geq 37.5$. To the best of our knowledge, most of the current Internet links fall into this range. Therefore, TCP Probe is able to work well in all but a few extreme cases.

3.3. Service Agility based on IHN

The primary goal of the AIMD algorithm is to maintain stability, inter-protocol friendliness, and intra-protocol fairness when adjusting the effective sending rate. As a result, AIMD schemes often probe conservatively for additional available bandwidth and are suitable only when bandwidth changes are slight. When the bandwidth change is drastic (e.g. caused by vertical handoffs to faster link technology), AIMD does not adapt fast enough to match the suddenly materialized bandwidth. Recent studies have indicated that a highly agile system may not be possible unless handoff notification is implemented [12].

Fortunately, with the passive capacity monitoring techniques (i.e. TFRC Probe and TCP Probe), *implicit handoff notifications* (IHN) can be carried out by continuously estimating the link capacity. Suppose the most recent capacity estimate is C' and its previous estimate is C, a vertical handoff is identified when either $C' > \alpha C$ (LOW-to-HIGH handoff) or $C' < \beta C$ (HIGH-to-LOW), where α and β are two positive constants of threshold. For simplicity, we set $\alpha = 5$ and $\beta = 0.2$ throughout this study.

Once a drastic capacity change is observed, an IHN can be triggered to notify the ongoing applications to perform appropriate service adaptations. If the vertical handoff is LOW-to-HIGH, we propose the "*Fast Rate Adaptation*" (FRA) algorithm to aggressively take advantage of the dramatic capacity increase. More specifically, FRA forces TCP/TFRC Probe to enter the *slow start*[1] phase and probe the new link exponentially, rather than staying in *congestion avoidance*[2] with linear probing. FRA is able to help TCP/TFRC Probe achieve a higher throughput and better network utilization, especially when the capacity increase is large.

Additionally, when multiple data copies (e.g. video of different bit rates) are available on the server, the FRA algorithm should switch the ongoing data delivery to the higher quality data when it is possible. For instance, if two versions of stream video (say, 192kbps and 512kbps bit rates) are available on the video server, the FRA algorithm should switch the delivering data from the 192kbps version to the 512kbps version when the sending rate becomes larger than 512kbps. Therefore, the user-perceived stream quality can be greatly improved.

However, when the vertical handoff is HIGH-to-LOW, and assuming it results in a significant reduction in path capacity, most of the outstanding packets (transmitted but not yet received) will be lost. Ideally, in this case, a handoff notification is sent slightly before the handoff actually occurs, which IHN is not capable of doing. We will present a solution for this scenario in the next section.

4. Proposed Approach - II: Explicit Handoff Notification

For mobile systems incorporating an intelligent handoff manager, we propose Explicit Handoff Notification (EHN) to provide agile accommodations to vertical handoffs. With an intelligent handoff manager, such as the Smart Decision Model presented in [4], the decision on which network interface to use can be made in advance by considering user preferences as well as network parameters (e.g. link capacity, power consumption, remaining battery power, etc [4, 32]). The intelligent handoff manager automatically monitors the various physical link qualities, and triggers handoff event when appropriate. EHN gives advance handoff notifications to ongoing upper layer applications, allowing them to adjust faster to drastic changes in link capacity.

If EHN is deployed in a mobile system, it can be used in fact in both High-to-Low and Low-to-High handoffs. In case of Low-to-High handoff, upon receiving an EHN, the sender launches the FRA algorithm, enabling TCP/TFRC Probe to promptly utilize the newly materialized capacity. Since EHN is an explicit notification of a handoff event, FRA algorithm can be launched almost immediately after the EHN is received. IHN, on the other hand, depends on

[1] In TFRC [10], there is a slow start phase, during which the TFRC sender doubles the sending rate every RTT (i.e. exponentially) until the first packet loss occurs. After the first packet loss, TFRC sets its sending rate to half of the current rate and thereafter adjusts its rate based on the TCP response function.

[2] By congestion avoidance, in TFRC, we mean the TFRC sender adapts the rate based on the TCP response function.

the speed of capacity estimation to determine when handoff events have occurred, possibly causing some delay in launching the FRA algorithm. EHN can be expected to achieve a higher data throughput due to its faster reaction to vertical handoffs.

For HIGH-to-LOW vertical handoff events, we propose the Early Rate Reduction (ERR) algorithm to appropriately slow down the TFRC/TCP Probe sending rate before the actual vertical handoff event. More specifically, ERR slows down the TFRC/TCP Probe sending rate one OWD (one-way delay) before the handoff event takes place, reducing the amount of outstanding data packets that would be lost in the wireless portion of the path. To illustrate this concept, suppose a vertical handoff changes the capacity from C_1 to C_2 (where $C_1 > C_2$). Let the original TFRC Probe sending rate be R, and suppose the EHN is triggered one OWD before the actual handoff. The ERR algorithm first reduces the TFRC/TCP Probe sending rate from R to RC_2/C_1 (one OWD before the actual handoff), allowing the delivery of the outstanding packets before switching to a slower network interface. By the time the vertical handoff event actually occurs, the mobile host will already be sending at an adjusted rate RC_2/C_1, therefore avoiding potential bulk packet losses. Fig. 4 illustrates the ERR algorithm, the shaded region depicts the amount of outstanding data that is salvaged by employing the ERR algorithm.

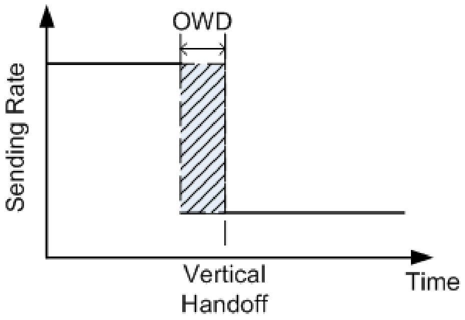


Figure 4. Illustration of the ERR algorithm for TCP/TFRC Probe

Note that, for real-time streaming applications, the ERR algorithm should also reduce the data bit rate (e.g. switch to the lower bit-rate video or perform real-time transcoding) so that the user-perceived stream quality can be largely preserved (i.e. reducing the delay of real-time streaming). It should also be mentioned that the EHN scheme applies exclusively to mobile hosts equipped with the intelligent handoff manager. Moreover, EHN is designed for the wireless last-hop scenarios. Therefore, EHN is not directly applicable to multi-hop networks that conduct vertical handoffs on intermediate nodes.

5. Simulation

In this section, we present simulation results to evaluate the performance of service agility using TCP/TFRC Probe. Both IHN and EHN were implemented in the NS-2 simulator [1]. IHN is triggered once a drastic capacity increase was observed from passive capacity monitoring; while EHN was explicitly triggered prior to a handoff event (i.e. assuming the intelligent handoff manager is given, and the EHN is sent directly to TCP/TFRC applications). Moreover, the FRA algorithm was implemented to utilize the increased capacity more aggressively after a LOW-to-HIGH vertical handoff, and the ERR algorithm was launched when an EHN of drastic capacity reduction was received for an imminent HIGH-to-LOW handoff.

Fig. 5 illustrates the simulation setup. All links except the one between node 5 and node 6 belonged to the wired Internet segment and had a capacity of 100 Mbps each. Node 5 and node 6 were connected via a wireless link, with a capacity of either 150 Kbps (1xRTT) or 5 Mbps (802.11b). One application flow (TCP or TFRC) was initiated from node 1 to node 6, such that the wireless link became the last hop. 16 Pareto distributed flows (with the parameter alpha = 1.9, and the total rate of 25Mbps) were created from node 7 to node 10 (4 flows), from node 8 to node 9 (4 flows), from node 11 to node 14 (4 flows), and from node 12 to node 13 (4 flows). These Pareto flows represented the long range dependent (LRD) traffic observed on the Internet [31].

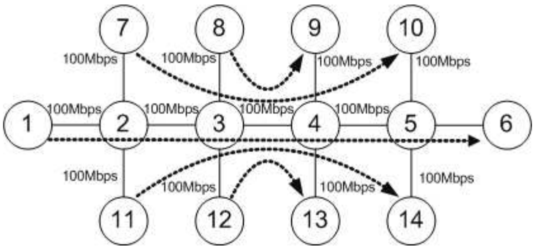


Figure 5. Simulation scenario

During the 1200-second simulation, a vertical handoff event was generated at the 600th second on the last hop (i.e. the link between node 5 and node 6). We now present the simulation results of the LOW-to-HIGH handoff in subsection 5.1 and those of the HIGH-to-LOW handoff in subsection 5.2.

5.1. Vertical handoff from LOW to HIGH

5.1.1 TFRC Probe

We first study the performance of FRA and IHN/EHN in the context of TFRC Probe when the handoff is LOW-to-HIGH.

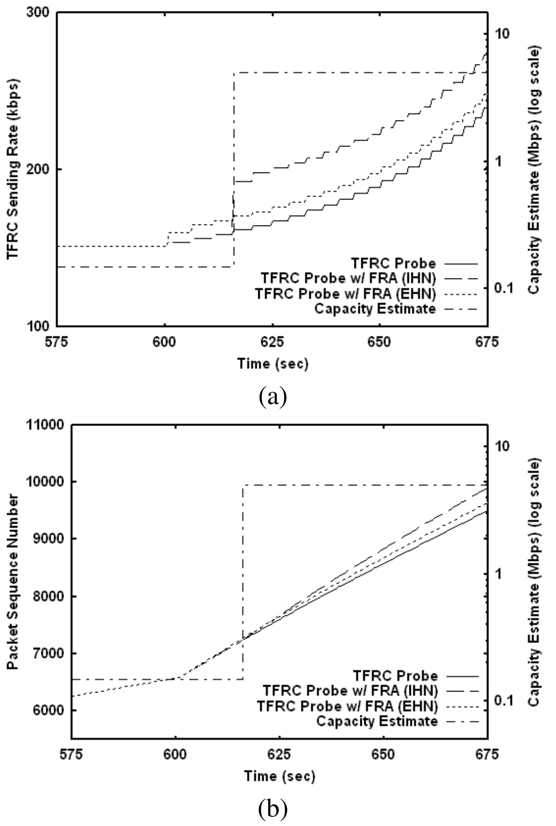


Figure 6. Simulation results of TFRC Probe (original, w/ FRA, and w/ EHN) during a vertical handoff from a 150kbps link to a 5Mbps link (the vertical handoff occurred at the 600th second): a) TFRC sending rate; b) TFRC packet sequence number.

Three TFRC Probe variants, namely the original TFRC Probe, TFRC Probe with IHN+FRA and TFRC Probe with EHN+FRA, are compared in Fig. 6. Note that there existed a lag of approximately 15 seconds between the moment of handoff and its detection by IHN in TFRC Probe, after which TFRC Probe with IHN+FRA was able to enter slow start and increase its sending rate faster than the original TFRC Probe (see Fig. 6(a)). On the other hand, EHN was able to foresee the handoff right before it actually occurred (at the 600th second), so TFRC Probe with EHN+FRA entered slow start and increased the rate 15 seconds earlier. Both advanced TFRC Probe variants managed to utilize the increased capacity after handoff more efficiently, in terms of the sending rate (Fig. 6(a)) and throughput (Fig. 6(b)), than the original TFRC Probe. At the same time they exited the slow start phase quickly and continued in congestion avoidance without incurring any congestion.

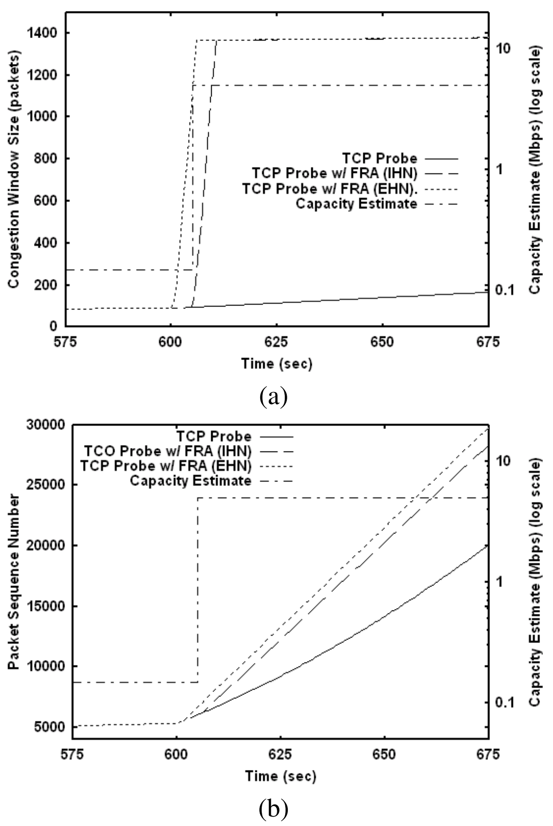
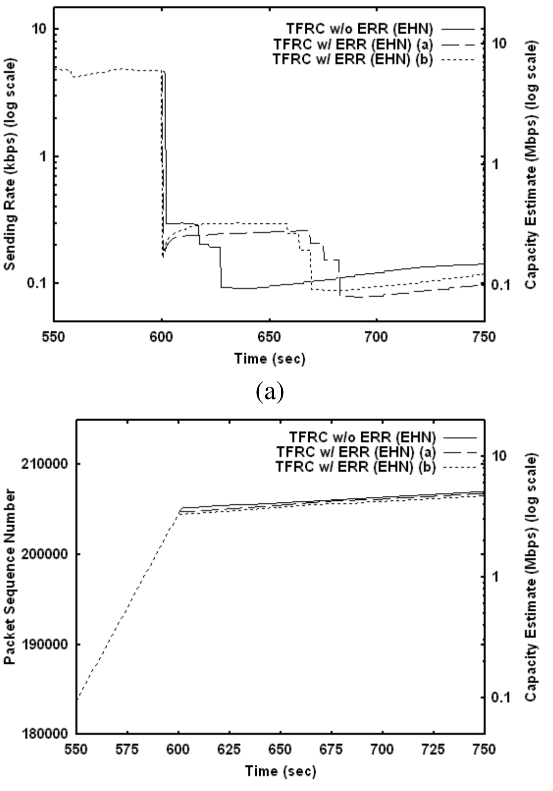


Figure 7. Simulation results of TCP Probe (original, w/ FRA, and w/ EHN) during a vertical handoff from a 150kbps link to a 5Mbps link (the vertical handoff occurred at the 600th second): a) TCP congestion window size; b) TCP packet sequence number.

5.1.2 TCP Probe

We then look at how FRA and IHN/EHN perform in TCP Probe. The simulation results, shown in Fig. 7, are similar to those with TFRC Probe. The two advanced TCP Probe variants, equipped with FRA+IHN and FRA+EHN respectively, were able to enter slow start and increase the window exponentially when the handoff was detected (IHN lagged behind EHN by 5 seconds in this case). In contrast, the original TCP Probe stayed in congestion avoidance and increased its window slowly (Fig. 7(a)). Fig. 7(b) shows that the throughput of TCP Probe with FRA+IHN or FRA+EHN was significantly higher than the original scheme lacking awareness of the vertical handoff.



(a)

(b)

Figure 8. Simulation results of TFRC Probe with/without explicit handoff notifications during a vertical handoff from a 5Mbps link to a 150kbps link (the vertical handoff occurred at the 600th second): a) TFRC sending rate; b) TFRC packet sequence number.

5.2. Vertical handoff from HIGH to LOW

We have discussed earlier in this paper that when the vertical handoff is HIGH-to-LOW, IHN cannot detect it in a timely manner; a significant amount of packets will get lost before the sender reduces its sending rate. Therefore, we did not simulate IHN in this scenario. Instead we focused on EHN and assessed the efficacy of ERR in reducing the number of lost packets during the handoff.

Fig. 8 presents the comparison of three TFRC Probe variants. In addition to the original TFRC Probe with no handoff notification, there were two advanced variants of TFRC Probe with EHN: EHN(a): EHN was generated when the handoff occurred, i.e. without ERR; and EHN(b): EHN was generated one OWD before the handoff, i.e. with ERR.

As Fig. 8(a) shows, TFRC Probe with EHN(a) and EHN(b) both reduced the sending rate more drastically than the original TFRC Probe when the handoff occurred. They also experienced a second rate cut as the original TFRC Probe did, but the time between rate cuts was accordingly prolonged. Finally all TFRC Probe variants had their sending rate converged to the new last-hop capacity (150 kbps) and entered congestion avoidance.

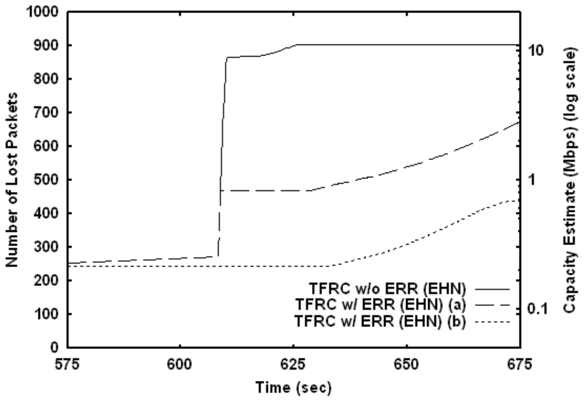


Figure 9. Simulation results of TFRC Probe with/without explicit handoff notifications during a vertical handoff from a 5Mbps link to a 150kbps link (the vertical handoff occurred at the 600th second).

Fig. 8(b) seemingly indicates that all TFRC Probe variants had practically the same throughput in our aforementioned simulation. This is true considering solely the number of received packets at node 6. However, Fig. 9 reveals more insight into the issue of QoS. TFRC Probe with EHN(b), namely ERR-equipped, had approximately 200 packets lost during the handoff. Without ERR, TFRC Probe with EHN (a) lost about 400 packets, twice as many as TFRC Probe with EHN(a). The original TFRC Probe suffered the most packet losses with a number of 650. It is clear that both EHN and ERR are effective algorithms that reduce the number of packets during a HIGH-to-LOW vertical handoff.

6. Conclusions

In this paper, we studied design issues and proposed corresponding solutions for providing service agility for mobile hosts in various vertical handoff scenarios. Using TCP and TFRC as examples, we proposed Implicit Handoff Notification (IHN), an algorithm to detect the occurrences of vertical handoffs by passively monitoring and estimating the link capacity with embedded end-to-end estimation tools (e.g. TCP Probe and TFRC Probe). End-to-end protocols such as TCP Probe or TFRC Probe are suggested to identify the nature of the handoff event estimate the amount of change in link capacity. For mobile systems equipped with an intelligent handoff manager, we purposed the Explicit Handoff Notification (EHN) scheme. The handoff manager automatically monitors the physical link qualities, and triggers handoff events when appropriate. EHN explicitly notifies the ongoing applications when/what handoff event is about to be generated by the manager, enabling the upper layer application to adapt its sending rates accordingly.

References

[1] Network simulator (ns-2). http://www.mash.cs.berkeley.edu/ns/.

[2] D. Bansal, H. Balakrishnan, S. Floyd, and S. Shenker. Dynamic behavior of slowly responsive control algorithms. In *ACM SIGCOMM*, 2001.

[3] R. Braden. Requirements for internet hosts communication layers. Technical report, IETF RFC 1122, October 1989.

[4] L.-J. Chen, T. Sun, B. Chen, and M. Gerla. A smart decision model for vertical handoff. In *The 4th ANWIRE International Workshop on Wireless Internet and Reconfigurability*, 2004.

[5] L.-J. Chen, T. Sun, D. Xu, M. Y. Sanadidi, and M. Gerla. Access link capacity monitoring with tfrc probe. In *E2EMON*, 2004.

[6] L.-J. Chen, T. Sun, G. Yang, M. Y. Sanadidi, and M. Gerla. End-to-end asymmetric link capacity estimation. In *IFIP Networking*, 2005.

[7] D.-M. Chiu and R. Jain. Analysis of the increase and decrease algorithms for congestion avoidance in computer networks. *Computer Networks and ISDN Systems*, 17:1–14, 1989.

[8] S. Deering and R. Hinden. Internet protocol, version 6 (ipv6) specification. Technical report, IETF RFC 2460, December 1998.

[9] G. Dommety. Fast handovers for mobile ipv6. Technical report, draft-ietf-mobileip-fast-mipv6-04.txt, IETF Internet draft, March 2002.

[10] S. Floyd, M. Handley, J. Padhye, and J. Widmer. Equation-based congestion control for unicast applications. In *ACM SIGCOMM*, 2000.

[11] V. Ghini, G. Pau, M. Roccetti, and P. S. M. Gerla. Smart download on the go: A wireless internet application for music distribution over heterogeneous networks. In *IEEE ICC*, 2004.

[12] A. Gurtov and J. Korhonen. Measurement and analysis of tcp-friendly rate control for vertical handovers. *ACM MCCR*, 2004.

[13] M. Handley, H. Schulzrinne, E. Schooler, and J. Rosenberg. Sip: Session initiation protocol. Technical report, IETF RFC 2543, March 1999.

[14] G. Holland, N. Vaidya, and P. Bahl. A rate-adaptive mac protocol for multi-hop wireless networks. In *ACM MobiCom*, 2001.

[15] R. Hsieh, Z. G. Zhou, and A. Seneviratne. S-mip: a seamless handoff architecture for mobile ip. In *IEEE Infocom*, 2003.

[16] D. B. Johnson, C. Perkins, and J. Arkko. Mobility support in ipv6. Technical report, draft-ietf-mobileip-ipv6-17.txt, IETF Internet draft, May 2002.

[17] A. Kamerman and L. Monteban. Wavelan ii: A high-performance wireless lan for the unlicensed band. *Bell Lab Technical Journal*, Summer:118–133, 1997.

[18] R. Kapoor, L.-J. Chen, L. Lao, M. Gerla, and M. Y. Sanadidi. Capprobe: A simple and accurate capacity estimation technique. In *ACM SIGCOMM*, 2004.

[19] K. E. Malki. Low latency handoffs in mobile ipv4. Technical report, draft-ietf-monileip-lowlatency-handoffs-v4-03.txt, IETF Internet draft, November 2001.

[20] D. A. Maltz and P. Bhagwat. Msocks: An architecture for transport layer mobility. In *IEEE Infocom*, pages 1037–1045, March 1998.

[21] A. Matsumoto, M. Kozuka, K. Fujikawa, and Y. Okabe. Tcp multi-home options. Technical report, draft-arifumi-tcp-mh-00.txt, IETF Internet draft, October 2003.

[22] B. D. Noble, M. Satyanarayanan, D. Narayanan, J. E. Tilton, J. Flinn, and K. R. Walker. Agile application-aware adaptation for mobility. In *ACM Symposium on Operating Systems Principles*, pages 276–287, 1997.

[23] C. Perkins. Ip mobility support for ipv4. Technical report, IETF RFC 3344, August 2002.

[24] A. Persson, C. A. C. Marcondes, L.-J. Chen, M. Y. Sanadidi, and M. Gerla. Tcp probe: A tcp with built-in path capacity estimation. In *IEEE Global Internet Symposium*, 2005.

[25] J. Postel. Transmission control protocol. Technical report, IETF RFC 793, September 1981.

[26] M. Schlager, B. Rathke, S. Bodenstein, and A. Wolisz. Advocating a remote socket architecture for internet access using wireless lans. *Journal of Mobile Networks and Applications*, 6:23–42, 2001.

[27] A. Snoeren. *A Session-Based Approach to Internet Mobility*. PhD thesis, Massachusetts Institute of Technology, December 2002.

[28] A. C. Snoeren and H. Balakrishnan. An end-to-end approach to host mobility. In *ACM MobiCom*, 2000.

[29] M. Stemm and R. H. Katz. Vertical handoffs in wireless overlay networks. *ACM Mobile Networking (MONET)*, 1998.

[30] R. Stewart, Q. Xie, K. Morneault, H. Schwarzbauer, T. Taylor, I. Rytina, M. Kalla, L. Zhang, and V. Paxson. Stream control transmission protocol. Technical report, IETF RFC 2960, October 2000.

[31] M. S. Taqqu, W. Willinger, and R. Sherman. Proof of a fundamental result in self-similar traffic modeling. *ACM SIGCOMM Computer Communications Review*, 27:5–23, 1997.

[32] H. Wang, R. H. Katz, and J. Giese. Policy-enabled handoffs across heterogeneous wireless networks. In *ACM WMCSA*, 1999.