Modeling the Effect of Mobile Handoffs on TCP and TFRC Throughput

Antonios Argyriou and Vijay Madisetti

School of Electrical and Computer Engineering
Georgia Institute of Technology
Atlanta, Georgia 30332–0250, USA
Email: {anargyr,vkm}@ece.gatech.edu

***Abstract*— At the forefront of the recent advances in mobile networks is the development of sophisticated mobility management mechanisms that are usually based on Mobile IP and its derivatives. Based on these mobility management protocols, several studies that characterize transport protocol performance have been presented. In this work we move one step further, and present a joint performance evaluation model of TCP and TFRC, with the underlying IP-based mobility protocols. We develop stochastic models that can characterize the protocol performance during handoffs between heterogeneous wireless networks like WLAN, cellular, or WMAN. We present performance evaluation results for validating the developed models under a set of different handoff scenarios. The developed model can be utilized as a basis for further analytical evaluation of new mobility management protocols, allowing thus a fast and accurate comparison.**

I. Introduction

The wide-spread success of IP-based mobile and wireless devices, has created a need for new network protocols and architectures so that revenue-generating and seamless services can be provided to the end user. One of the first challenges that has to be resolved is that of mobility management. The functionality that mobility management defines consists of two separate operations — location management and handoff management. Currently, the protocol that is considered to be a practical approach to the above problems is Mobile IP [1]. Despite however the elegant solution that Mobile IP offers, TCP performance suffers from several problems due to handoffs caused by host mobility [2].

Nevertheless, modeling the performance of TCP is such a mobile environment has received little attention. There are several models for quantitatively analyzing TCP throughput [3], [4], specifically in the context of the wired internet. On the other hand, considerable amount of work on modeling mobility management protocols is available, and especially for mobile IP and its derivatives [5], [6]. The main focus of the these mobility management modeling approaches has been the characterization of the signalling and processing loads as a measure of protocol performance. We are aware of only one recent study [7], where the authors evaluate the performance of the TCP-friendly rate control protocol (TFRC) [8] during handoffs between asymmetric networks. However, they do not attempt to model or express analytically this behavior.

II. System Model and Assumptions

The proposed system model assumes an end-to-end connection with the data flowing from the correspondent host (CH) to the mobile host (MH). According to the mobility scenario that we want to analyze, the MH is initially associated with the first access network (AN_1), and at some time in the future it may move towards AN_2. The two access networks are modeled separately, while they are both attached to the core network (CN) which is the Internet. We make the assumption that while the mobile host is connected to AN_1, the transport protocol is in steady state. Subsequently, after handoff is performed to AN_2, the protocol will reach the steady state at some time in the future.

Many studies have shown that the first-order two-state Markov chain (i.e. the Gilbert path model) can approximate fairly well the behavior of the Internet. However, in this paper we adopt the Bernoulli path model which is a simplification of the Gilbert model. The reason why we selected a simpler path model comes from the need to model TCP on the packet level and obtain a closed form result. By following a more complex path model, and therefore TCP model, the overall system complexity will be unnecessarily increased. With the Bernoulli model the only quantity needed to model the channel is the average packet loss rate. The average packet loss rate is simply calculated by dividing the number of lost packets vs. the total number of packets lost or received for each of the two paths:

$$P_{1,2} = \frac{\#lost}{\#(lost + received)} \tag{1}$$

However, in order to capture the effect of the handoffs we define the handoff packet loss rate P_H, which represents packet losses caused only by handoffs.

III. TCP model with mobile handoffs

Objective of this performance model is to evaluate how the specific parameters related to handoffs affect the transient packet loss, and ultimately the actual TCP throughput. More specifically, we assume that TCP is in the congestion avoidance phase when a handoff takes place. We also model a TCP connection between two endpoints by considering rounds, that have a duration of one RTT. We name the number of RTT

1-4244-0357-X/06/$20.00 ©2006 IEEE
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE GLOBECOM 2006 proceedings.

rounds that pass until there is a packet loss as the NL round (figure 1). During this round, TCP sends a burst of packets equal to the allowed window, and waits for acknowledgments. This approach, which is based on renewal theory can lead to a closed form solution for the average throughput.

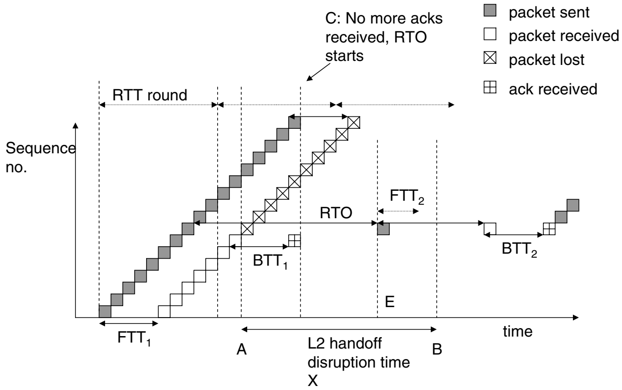

Fig. 1. Packet-level TCP behavior at the sender during a handoff.

A. Handoff packet loss rate (P_H)

Figure 1 will be used for explaining the behavior of TCP during a handoff event. The variables FTT_1/BTT_1, and FTT_2/BTT_2 describe the forward and backward trip times for the two access networks respectively. According to this figure, at time instant t_A, L2 connection is lost and at time t_A+RTO, the TCP sender retransmits the first lost packet. Even more, if $t_B > t_C + FTT_1$, then clearly no duplicate acknowledgments will be received at the sender, and the only way for TCP to resume the data flow is by expiration of the RTO of the first lost packet. On the other hand, as T_H shrinks, and if $t_B \leq t_C + FTT_1$ the probability to receive a number of the last packets (close to time instant t_C) is increased. If this happens, then the sender would fast retransmit the first lost packet, resulting into a faster recovery. If we rewrite the previous equation we have (and because $t_C = t_A + BTT_1$): $t_B \leq t_C + FTT_1 \Rightarrow T_H \leq RTT_1$. Therefore, the probability of handoff induced packet loss is the probability of that the one way network latency L_N, is smaller than the handoff duration T_H, or:

$$P_H = P[L \leq T_H] \tag{2}$$

We define as f_{L_N} the distribution of the end-to-end network induced latency. However, the selection of f_{L_N} is a decision orthogonal to our work in this paper. In our case we will model the one-way Internet latency as a Gamma distribution that occurs mainly due to buffering at the routing infrastructure. On the other hand, the disruption time T_H is a parameter that depends of the mobility management a protocol.

B. Throughput

Having calculated the packet loss probability due to handoff P_H, we now have to calculate the actual number of packets sent and lost during the handoff events. First, we will see how the congestion window is evolved when packet loss leads only to triple-duplicate events. Then we will calculate the probability that a packet loss event led to recovery of the lost packet with TO and we will also calculate the number of packets sent during the TO events.

Congestion window in congestion avoidance: Concerning the evolution of the congestion window, we next show the equation that describes its evolution based on the average end-to-end packet loss rate. By re-using the formula developed at [3], that describes congestion window evolution as a function of the packet loss rate, we have:

$$E[W_x^h] = \frac{2}{b}\left(\frac{2+b}{6} + \sqrt{\frac{2b(1-P_x-P_H)}{3(P_x+P_H)} + \left(\frac{2+b}{6}\right)^2}\right) \tag{3}$$

where $P_x + P_H$ is the aggregate end-to-end packet loss rate for either path 1 or 2 ($x = 1, 2$). Now if packet losses caused by handoff or other reason in the end-to-end path, and these losses create only triple duplicate events (TD), the throughput formula would be of course similar to [3]:

$$T_x = \frac{\frac{1-P_x}{P_x} + E[W_x^h]}{RTT(E[W_x^h]b/2+1)} \tag{4}$$

However, there is the probability that packet losses caused by the handoff event or the end to end paths lead to a timeout (TO). This is what we calculate next.

Probability of TO and TD events: Assume that the sender has sent a window w worth of packets in the current RTT round. The probability that the first i packets are acknowledged in this round, given that the rest are lost because of a handoff or buffer overflow in the wireline path 1, is given by $P_{HW} = p_H + p_1$. Because packet loss in the wired path and handoff loss are independent, the previous equation becomes:

$$P_{HW}(w,i) = \frac{(1-P_H)^i P_H}{1-(1-P_H)^w} + \frac{(1-P_1)^i P_1}{1-(1-P_1)^w} \tag{5}$$

In the above equation the term $(1-P)^i P$ expresses the probability that i packets are received before one is lost on a channel with packet loss rate P. Also $1-(1-P)^w$ is the probability that a loss happens in a NLR where w packets were sent.

We also write the probability that m packets acknowledged from the n sent in the last RTT round [3]:

$$G(n,m) = \begin{cases} P_H^m(1-P_H) & \text{if } m \leq n-1 \\ P_H^m & \text{if } m = n \end{cases} \tag{6}$$

Now, the probability that at most two of them were acknowledged is $g(k) = \sum_{i=0}^{2} G(k,i)$.

The probability for a TO to happen, will be one of course if $w < 3$, since not enough duplicate ACKs can be received. Now let us see what happens when $w \geq 3$. In that case a packet loss could lead to a TO in two cases: First, if TCP sends successfully less than three packets from a round of w packets send, this would lead to a TO because not enough duplicate acknowledgements will be received at the sender. Second, there is also the probability that the number of acknowledged packets is more than three in one RTT round, while a TO happens because in the next round less than three packets are acknowledged. The contribution of the aforementioned two cases in the probability that a packet loss is a TO for path x will give the final:

$$P_{TO_x}(w) = \sum_{k=0}^{2} P_{HW}(w,k) + \sum_{k=3}^{w} P_{HW}(w,k)g(k) \qquad (7)$$

Note also that $P_{TD_x}(w) = 1 - P_{TO_x}(w)$.

TO duration: Besides the probability for a TO to happen, we have to calculate the average duration of a timeout, based on the probability that $t_B > t_C + FTT_1$ (figure 1). The number of the retransmitted packets that will be lost leading to further RTO increase determine when the data flow will be resumed on the new link. Given that the duration of a handoff is $T_H = t_B - t_A$, the number of retransmitted packets depends on the duration of the timeout period, and subsequently on the number of exponential growths the TO timer experienced. We know that for TCP, j consecutive RTO events will have a duration:

$$L_h = \begin{cases} (2^j - 1)RTO_0 & \text{if } j < 6 \\ (63 + 64(j-6))RTO_0 & \text{if } j \geq 7 \end{cases} \qquad (8)$$

where RTO_0 is the initial value of the retransmission timer. By inverting this expression, we obtain the number of RTO expirations:

$$j = \begin{cases} log_2(\frac{L_h}{RTO_0} + 1) & \text{if } L_h \leq (2^6 - 1)RTO_0 \\ \frac{L_h}{RTO_0} + 5 & \text{otherwise} \end{cases} \qquad (9)$$

Therefore, the number of experienced TOs will be obtained by taking the $\lceil\ \rceil$ of $\frac{L_h}{RTO_0}$. This value will give the number of RTO expirations and the number of retransmitted packets. The previous equation finally gives the expected number of packets sent during $t_C < t < t_B$:

$$E[S_h] = \begin{cases} log_2(\lceil \frac{T_H}{L_h} \rceil + 1) & \text{if } T_H \leq (2^6 - 1)RTO_0 \\ \lceil \frac{T_H}{L_h} \rceil + 5 & \text{otherwise} \end{cases} \qquad (10)$$

and the duration of the induced TOs is $E[L_h] + T_H$.

Final throughput formula: Finally, by combining all the previous equations, the complete TCP throughput model that considers handoffs between asymmetric links is given by:

$$T_x^h = \frac{\frac{1-P_x}{P_x} + E[W_x^h] + P_{TO}E[S_h]}{RTT_x(E[W_x^h]b/2 + 1) + P_{TO}(T_H + E[L_h])}$$

With this formula we are able to estimate the throughput of TCP connection that is characterized by packet loss rate P_x and RTT_x after a disruption of T_H seconds happens. For the new path the P_H will be zero which means that the throughput estimate will only depend on the characteristics of the new path. It is interesting to note that we have described the TCP throughput as a function of the disruption time T_H, which is essentially controlled by the mobility management scheme in use, and the handoff packet loss rate P_H.

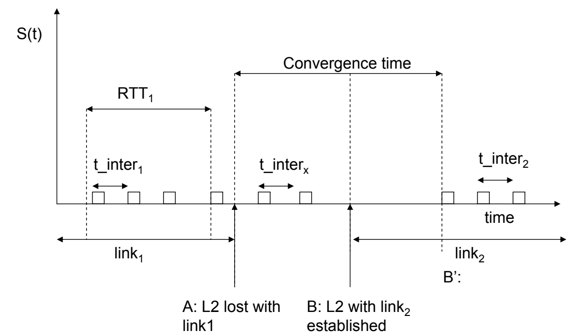

Fig. 2. TFRC behavior during IP layer handoff.

IV. TFRC Throughput

After dealing with TCP, we proceed with the characterization of the rate control protocol TFRC [8]. Our goal is to quantify the effect of mobile handoffs on a rate control protocol which uses a completely different congestion control algorithm. In this paper we consider TFRC to be implemented on top of UDP. TFRC uses the closed form equation for TCP throughput in order to regulate the sender's output rate:

$$T_{TFRC} = \frac{s}{RTT\sqrt{\frac{2p}{3}} + t_{RTO}(3\sqrt{\frac{3p}{8}})p(1+32p^2)} \qquad (11)$$

where s is the packet size, RTT is the RTT estimate, and t_{RTO} is the value of the retransmission timer. In addition TFRC follows a packet spacing algorithm at the sender [8]. Note that, equation 11 does not represent the actual TFRC sending rate but only an upper bound for it. The actual output rate of TFRC is calculated using the algorithm 1. In this algorithm, s represents the packet size, tld is time when the rate was last doubled, $tmbi$ is the maximum back-off time (64 seconds by default), and X_{recv} is the average receive rate. If p is zero, no packet loss has yet been seen by the flow and in this phase, the TFRC sender emulates slow start of TCP by doubling the transmission rate every RTT.

However, the main feature of TFRC is that it reacts slowly and cautiously to RTT changes and this means that the throughput estimate changes slowly when compared with TCP. Therefore, a sudden change in the RTT and bandwidth of a link, as is the case with handoffs, will lead to considerable packet losses. We will now attempt to model this behavior. We start by identifying the RTT estimation procedure that TFRC uses. Given a decay factor df, the n-th RTT estimate is calculated as follows: $RTT_n = df * RTT_{n-1} + (1-df) * \sqrt{RTT_{cur}}$, with RTT_{cur} be the last real RTT measurement.

1-4244-0357-X/06/$20.00 ©2006 IEEE
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE GLOBECOM 2006 proceedings.

Algorithm 1 TFRC rate estimation.

if $p > 0$ **then**
  $X\_calc = T(p, RTT, T\_0)$
  $tfrc\_x = max[min(X\_calc, 2 * X\_recv), \frac{s}{tmbi}]$
else
  if $tnow - tld \geq RTT$ **then**
    $tfrc\_x = max(min(2 * tfrc\_x, 2 * X\_recv), s/RTT)$
  end if
  $tld = tnow$
end if

When the TFRC sender does not receive feedback during an entire RTT, it cuts the output rate in half. If we assume that at instant t_B, where handoff is over the sender sends at least on packet and receives feedback, then in a period of RTT_2 the sender will receive the first feedback report. So for the handoff duration T_H, the sender will reduce the rate in half every RTT_1, since this the last RTT estimate. Therefore the total number of packets sent during T_H will be gradually reduced with a total number:

$$E[S_h] = \sum_{i=0}^{\lceil \frac{T_H}{RTT_{cur}} \rceil} \frac{1}{2^i} T_{TFRC} \times RTT \quad (12)$$

Therefore, for two links with different characteristics, p_1, RTT_1 and p_2, RTT_2, for links 1,2 or when $t < t_A$ or $t > t_C$, the actual rate at the sender will be given by the basic TFRC formula:

$$T_{TFRC}^{A^- B^+} = \max\Big(2\min(T_{TFRC}, 2(1-P_{1,2})T_{TFRC}, \frac{s}{tmbi}\Big) \quad (13)$$

when $t_A < t < t_B$ then:

$$T_{TFRC}^{AB} = \frac{1}{T_H}\Big(\sum_{i=0}^{\lceil \frac{T_H}{RTT_{cur}} \rceil} \frac{1}{2^i} T_{TFRC} \times RTT\Big) \quad (14)$$

If $t_B < t < t_C$, the sender gradually starts to catch up with the new link characteristics:

$$T_{TFRC}^{BC} = \frac{1}{t_{cv}}\Big(\sum_{n=1}^{n_{cv}} T_{TFRC}(RTT_n\_cv) \times RTT_n\Big) \quad (15)$$

We define as convergence time, t_{cv}, the time needed for TFRC to obtain its fair share of the bandwidth on the new link. Then we can rewrite the equation for RTT_n:

$$\begin{aligned} RTT_n &= df^n * RTT_0 + df^{n-1}(1-df) * \sqrt{RTT_{cur}} \\ &+ \ldots + (1-df) * \sqrt{RTT_{cur}} \end{aligned} \quad (16)$$

which if we solve for n gives:

$$n_{cv} = log_{df}\Big(\frac{RTT_n - (1-df) * \sqrt{RTT_{cur}}}{RTT_0(1-df) - (1-df) * \sqrt{RTT_{cur}}}\Big) \quad (17)$$

Therefore, n_{cv} will give the number of RTT rounds needed in order for the RTT_n estimate to converge to the RTT_{cur}. In the handoff case, RTT_{cur} represents the RTT_2 of the new link while RTT_0 is the last estimate we had and it is RTT_1. Practically we would like for RTT_n to be close to 90% of the RTT_{cur}. So the total convergence time will be:

$$t_{cv} = T_H + n_{cv} * RTT_n \quad (18)$$

The throughput is calculated from equations 13, 14, 15.

V. Experiments for model validation

The topology that forms the basis of the simulations assumes the handoff between two WLAN access points. Each AP is connected to a LAN which is configured as follows: The values for bandwidth and delay of the LAN links are 1Mbps/100ms and 500Kbps/100ms for the GFA links, and two FA links, respectively. The selected parameters for TCP and TFRC in the simulations are: W_{max} is 6MB, W_0 is one segment, MSS is 1460 bytes, the RTO_0 is 200 msec, and the s is 1500 bytes. The case study simulated is now described – Initially, the MH initiates an FTP data flow from the correspondent host (CH). According to the scenario, at the time instant 0 sec, the MH starts moving away from the first access point (AP), at a speed of 10m/sec, and is heading toward the other AP. The FA follows the Mobile IP procedure in order to notify the HA after the MH registers with new FA. In the case of Hierarchical MIP, the GFA is the one that handles the handoff from the two FAs that correspond to the two access points: AP_1, and AP_2. Finally, for the MIP-RO case we configured the MH to send a binding update directly to the CH. For all these experiments we used the ns-2 [9] network simulator.

HMIP and MIP-RO: With this experiment, we want to evaluate the effect of the handoff disruption time T_H, on the throughput of a session between the server and the client. Throughput results are shown in figure 3. As expected, even with MIP-RO, TCP throughput suffers considerably when disruption time is increased. The proposed model, predicts a logarithmic decrease in throughput as the packet loss rate is increased, which of course depends on the duration of the disruption. Concerning the throughput for the combination of HMIP/buffering, we can observe in figure 3 that TCP throughput remains considerably stable until the point where the disruption time comes close to the average RTT of the end-to-end session. Packets are buffered in the old AP and forwarded to the new AP, but after the buffer overflows, packets are dropped and the throughput decreases. This means that after the point where the forwarding buffer is full, the throughput will continue to decrease very fast since any new packet will be dropped.

In figure 4 we present results for TFRC throughput. TFRC uses a rate control algorithm that reacts slowly to packet loss and RTT fluctuations. In addition TFRC makes use of a packet spacing algorithm that arranges the packets in time. As shown in figure 4, the combined MIP-RO/TFRC suffers from minimal packet losses and throughput degradation. With the addition of a buffer in the previous AP, these losses are reduced further. Also it is important to note that since TFRC spaces the packet

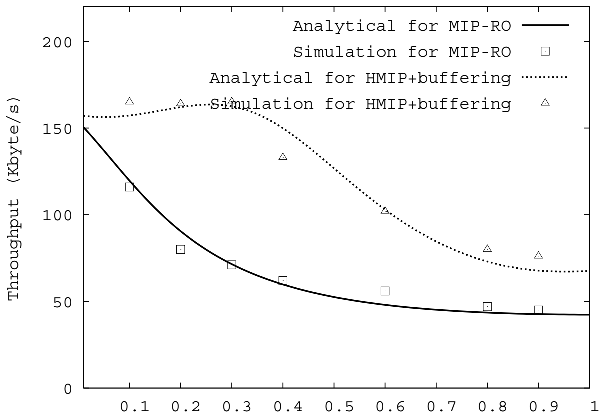

Fig. 3. Throughput results for a TCP session of 15 seconds and handoff between two different WLAN subnets.

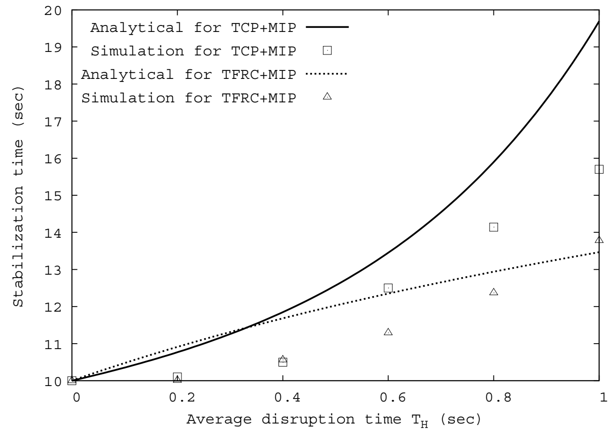

Fig. 5. Required recovery time versus disruption time for both TCP and TFRC.

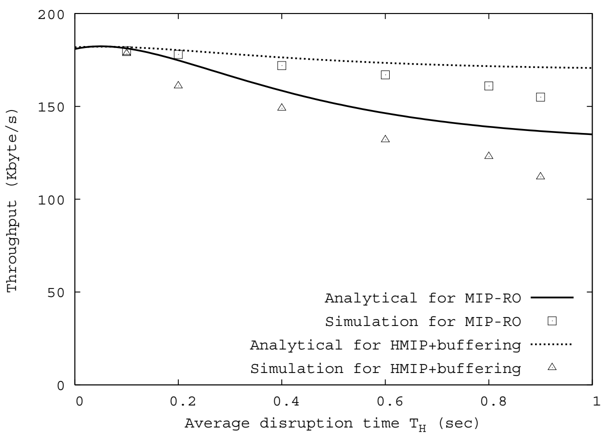

Fig. 4. Throughput results for a TFRC session of 15 seconds and handoff between two different WLAN subnets.

in time, the buffer at the previous AP should not overflow frequently since it does not receive packet bursts as with TCP.

Recovery period: We define as recovery period the time duration that the MH has to be out of the handoff state, so that the achieved throughput in this time period reaches the nominal for this link. Essentially we want to know how fast TCP will recover from a handoff with a specific duration T_H. The answer to this question can also give us useful information concerning the effect of the handoff rate on the TCP throughput. Figure 5, presents results for this experiment when baseline Mobile IP was used for both TCP and TFRC protocols. We see that for our model the required recovery period, is increased exponentially diverging from the real measurements which are not so pessimistic. We believe that this behavior is due the interpretation of more losses as a TO indication instead of TD. However, the TFRC model does not have to classify packet loss types, allowing thus a more accurate estimation as figure 5 also indicates.

VI. Conclusions

In this paper we presented a model for studying the effects of mobile handoffs on two transport protocols, namely TCP and TFRC. The model was found to be accurate for TCP in both the cases where HMIP and MIP-RO were used as the underlying mobility management protocols. However, the TFRC model predicts the expected throughput with even better accuracy, due to the simpler protocol algorithms. For example the worst case error for the TCP model was nearly 22% while for the TFRC model it was 13%. We also introduced in this paper the notion of the "recovery period". The slow-responsive rate control algorithm of TFRC, requires less time in order to recover when compared with TCP. However, we found that as the disruption time is increased, TFRC suffers from more packet losses than TCP, due to the slow-responsive algorithm.

References

[1] C. Perkins, "IP Mobility Support for IPv4," RFC 3220, January 2002.

[2] H. Balakrishnan, "Challenges to Reliable Data Transport over Heterogeneous Wireless Networks," Ph.D. dissertation, UC Berkeley, 1998.

[3] J. Padhye, V. Firoiu, and D. Towsley, "Modeling TCP Reno Performance: A Simple Model and Its Empirical Validation," *IEEE/ACM Transactions on Networking*, vol. 8, no. 2, pp. 133–145, April 2000.

[4] B. Sikdar, S. Kalyanaraman, and K. Vastola, "Analytic models for the latency and steady-state throughput of TCP Tahoe, Reno and SACK," in *INFOCOM*, 2001.

[5] C. Blondia *et al.*, "Performance Analysis of Optimized Smooth Handoff in Mobile IP," in *MSWiM*, 2002.

[6] ——, "Performance Comparison of Low Latency Mobile IP schemes," in *WiOpt*, 2003.

[7] A. Gurtov, "Effect of Vertical Handovers on Performance of TCP-Friendly Rate Control," in *ACM Computer Communications Review*, 2004.

[8] M. Handley, S. Floyd, J. Pahdye, and J. Widmer, "TCP Friendly Rate Control (TFRC): Protocol Specification," RFC 3448, January 2003.

[9] "Network simulator." [Online]. Available: http://www.isi.edu/nsnam/ns/

1-4244-0357-X/06/$20.00 ©2006 IEEE
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE GLOBECOM 2006 proceedings.