WIRELESS COMMUNICATIONS AND MOBILE COMPUTING
Wirel. Commun. Mob. Comput. 2006; **6**:547–557
Published online in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/wcm.410

A joint performance model of TCP and TFRC with mobility management protocols

Antonios Argyriou*,†

School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332, U.S.A.

Summary

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. Copyright © 2006 John Wiley & Sons, Ltd.

KEY WORDS: TCP; TFRC; mobility; analytical model

1. 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]. The Mobile IP-based design offers a solution to these two problems at the same time. First, it allows a host to be reached through a static IP address (location management), which is called the home address (HoA). Second, it allows transport layer sessions like TCP connections to continue when the underlying host moves (in mobile applications) and changes its IP address (handoff management). The latter is important to higher layer protocols that offer reliable service delivery [2,3]. Despite, however, the elegant solution that Mobile IP offers, TCP performance suffers from several problems due to handoffs caused by host mobility: The first problem is related to blackouts that lead to successive timer expirations and increase of the RTO every time an unsuccessful transmission takes place [4,5]. A mechanism for partly resolving this problem is through explicit layer 2 notifications to TCP, so that it can freeze the RTO [6,7]. The second problem is the long and

*Correspondence to: Antonios Argyriou, School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332, U.S.A.

†E-mail: anargyr@ece.gatech.edu

fluctuating delays due to local retransmission in the wireless link [8,9] or due to deep buffering in cellular access networks [9]. This situation can result into the invocation of congestion control algorithm and substantial decrease in the throughput. Third, the packet losses due to wireless errors have a similar effect since TCP is unable to distinguish them from congestion-induced losses.

There are several models for quantitatively analyzing TCP throughput [10,11], and latency [12], specifically in the context of the wired internet. Considerable amount of work on modeling mobility management protocols is available, and especially for mobile IP and its derivatives [13–17]. The main focus of the these mobility management modeling approaches has been the characterization of the signaling and processing loads as a measure of protocol performance [15,18,19]. Some related recent efforts include [17] where the authors investigate the performance of two IETF handoff protocols, namely the pre/postregistration mobile IP extensions under the assumption of an unregulated UDP source. They develop a simple analytical model that estimates the packet loss and the delay. Their results are restricted to the case of CBR traffic sources. Similar approaches, that do not consider complex transport protocols, and instead focus on UDP, have been published for mobile IP and mobile IP with route optimization [13,16]. While TCP-compatible rate control protocols have been studied extensively (e.g., the IETF standardized TCP-friendly rate control protocol (TFRC) [20]), there has not been much study of their behavior during wireless handoffs. We are aware of only one recent study, reported in [21], where the authors evaluate the performance of TFRC during handoffs between asymmetric networks. The authors identify performance degradation in TFRC due to its slow-responsiveness, and suggest the use of layer 2 triggers so that the protocol can adjust faster to the conditions of the new link. However, they do not attempt to model or express analytically this behavior.

The rest of this paper is organized as follows: In Section 2 we present an overview of the system model. Our analysis starts in Section 3 were we present a simple model of two mobility management protocols based on mobile IP. In Section 4 we present the stochastic model of TCP that incorporates the results from the previous section. Section 5 presents the throughput and latency model for the rate control protocol TFRC. Section 6 presents a set of comprehensive simulation results for validating the developed models, while Section 7 concludes this paper.

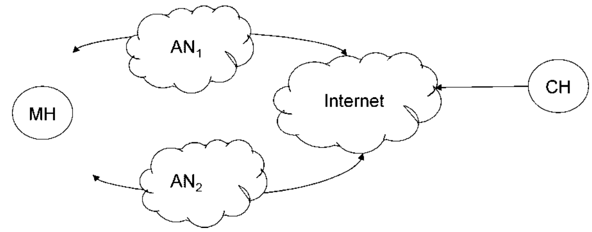

Fig. 1. End-to-end path model for transport protocol characterization during handoffs.

2. System Model and Assumptions

The proposed system model assumes an end-to-end session with the data flowing from the correspondent host (CH) to the mobile host (MH) (Figure 1). 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 MH 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 versus the total number of packets lost or received for each of the two paths:

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

However, to capture the effect of the handoffs we define the handoff packet loss rate P_{H}, which represents the packet losses only due to handoffs.

In addition, each end-to-end path that corresponds to one of the two access networks and the CN, is characterized by an RTT which has a constant value. This path model is general enough so that it can accommodate the modeling of different handoff management protocols.

In the next section, we derive simple models for hierarchical mobile IP (HMIP), and mobile IP with route optimization (MIP-RO), that can inter-operate with our path model. From these simple models we derive the average disruption time T_{H} for each mobility management protocol. For the rest of the analysis in this paper, we assume that the MH is using at least Mobile IP so that the TCP connection does not break due to changes in the IP address.

3. Modeling of Mobility Management Protocols

In this section we define a simple analytical model that captures the latency induced by all the operational phases (registration, tunneling, and packet delivery) of mobility management protocols. Specifically, we develop analytical models for two promising mobile IP optimized protocol versions: HMIP and MIP-RO.

3.1. Movement Cost Model

A simple movement model was adopted based on the assumption that the MH will move randomly between subnets [2]. We followed a typical configuration of a Mobile IP based network, that involves a hierarchy of foreign agents (FA) and gateway FAs (GFA). A MH initially registers with a FA, and the GFA, which subsequently informs the home network or home agent (HA) about the current location of the MH. When a CH wants to contact the MH it is redirected from the HA to the current MH location. Given that a network consists of N subnets, and that each subnet consists of k regional networks in case of HMIP, the expectation of a node going out of a regional network at movement M, requiring thus notification of the home network, is given by [15]:

$$E[M]_{\mathrm{hmip}} = 1 + \frac{N-1}{N-k} \tag{2}$$

In addition the residence time of the MH in a specific cell is denoted as T_{f}. In the case of MIP-RO, there are no regional networks. This means that every time the MH moves out of a subnet it must inform the home network. Therefore:

$$E[M]_{\mathrm{mipro}} = 2 - \frac{1}{N} \tag{3}$$

3.2. Packet Delivery

Packet delivery cost is crucial overhead in Mobile IP's performance, as packet tunneling is necessary even when the MH moves infrequently [2]. We calculate the overhead from packet delivery for MIP, HMIP, and MIP-RO. As in [13,15] we also define the following variables: L_{ch}, L_{hg}, and L_{gf} are the latencies for delivering a packet from the CH to HA, HA to GFA, and GFA to FA, respectively, while s_{h} and s_{g} are the packet processing delays at the HA and GFA respectively. HMIP has a packet delivery overhead [15]:

$$L_{\mathrm{PD}}^{\mathrm{hmip}} = s_{\mathrm{h}} + s_{\mathrm{g}} + L_{\mathrm{ch}} + L_{\mathrm{hg}} + L_{\mathrm{gf}} \tag{4}$$

The transmission delay from GFA to FA named L_{gf}, will be equal to $l_{\mathrm{gf}}\delta$, where δ is a proportionality constant and l_{gf} is the GFA/FA distance. We also assume that the distance between CH-HA and HA-GFA is the same, making thus $L_{\mathrm{ch}} = L_{\mathrm{hg}} = l_{\mathrm{hg}}\delta$. Now if λ_a is the data packet arrival rate at the HA, then the packet processing delay is analogous to λ_a with $s_{\mathrm{h}} = \nu\lambda_a$, where ν is also a proportionality constant. Concerning packet processing delay at the GFA it will consist of the lookup overhead of the IP routing table and of course the serving time for each packet. For the first parameter, we assume the worst case of being analogous to the routing table length e, with the number of MHs in the subnet ω, and of course the packet arrival rate λ_a. Therefore this cost is $e\lambda_a \log(k)$. With k subnets and ω hosts per subnet the packet processing delay will be $e\lambda_a\omega k$. So Equation (4) becomes:

$$L_{\mathrm{PD}}^{\mathrm{hmip}} = \nu\lambda_a + e\lambda_a(\omega k + \log(k)) + (l_{\mathrm{gf}} + 2 * l_{\mathrm{hg}})\delta \tag{5}$$

On the other hand MIP-RO does not suffer from triangular routing ($s_{\mathrm{h}} = 0$) but it still has to suffer the tunneling overhead. In addition, packets are routed directly from the CH to the GFA (distance l_{cg}) and then from the GFA to the FA. So the packet delivery cost will be:

$$L_{\mathrm{PD}}^{\mathrm{mipro}} = s_{\mathrm{g}} + L_{\mathrm{cg}} + L_{\mathrm{gf}} \tag{6}$$

Also $s_{\mathrm{g}}^{\mathrm{mipro}}$ is the same with $s_{\mathrm{g}}^{\mathrm{hmip}}$ because in addition to the routing overhead at the GFA, there is also the tunneling cost present (only s_{h} is avoided). With MIP-RO, packets also have to be routed to the MH and so the IP routing overhead is inevitable, and that is why $s_{\mathrm{g}} \neq 0$. Thus we have:

$$L_{\mathrm{PD}}^{\mathrm{mipro}} = e\lambda_{\mathrm{a}}(\omega k + \log(k)) + (l_{\mathrm{cg}} + l_{\mathrm{gf}})\delta \tag{7}$$

3.2.1. Binding Updates

As mentioned earlier, binding or location updates are a requirement in Mobile IP even when the MH does not change its current address [1]. The processing latencies s_f, s_g, and s_h will also have to be encountered for the binding updates at the FA, GFA, and HA respectively. In addition LU_{hg}, LU_{gf}, LU_{fm}, are the transmission costs of binding updates between the HA-GFA, GFA-FA, and FA-MH respectively. Also δ_U is a distance cost unit. So the latency for a binding update from the GFA to the HA, and for a regional binding update from the MH to the GFA are given by the following two equations:

$$BU_h^{hmip} = s_h + 2LU_{hg} \tag{8}$$

$$BU_r^{hmip} = 2s_f + s_g + 2l_{gf}\delta_U \tag{9}$$

For the MIP-RO case BU_h^{mipro} will have to include the delay to notify the CH which is equal to $2l_{cg}$, while BU_r^{mipro} will be the same. So BU_h is:

$$BU_h^{mipro} = 2s_f + 2s_g + 2(l_{gf} + l_{cg})\delta_U \tag{10}$$

These values capture the overall binding update delays for all the MHs inside the subnets of the regional network. In order to calculate the average per host cost of a binding update, we have to consider the average residence time inside a subnet T_f. For both MIP-RO and HMIP we have:

$$BU = \frac{BU_h + BU_r}{T_f} \tag{11}$$

In order to calculate the expected disruption time for a specific scheme due to the latency caused by binding updates, we have to calculate first the latency of a specific network configuration. More specifically, during handoff the only latency that the MH will suffer is that of the local binding update BU_r^{hmip}. We can easily see that the average disruption time becomes $T_H^{hmip} = BU_r^{hmip}$. Similarly for MIP-RO the $T_H^{mipro} = BU^{mipro}$.

4. 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 and latency. More specifically, we assume that TCP is in the congestion avoidance phase when a handoff takes place. We think that this is a reasonable assumption given that a TCP connection usually spends most of its lifetime in this phase. We also model a TCP connection between two endpoints by considering rounds (as in Reference [22]), that have a duration of one RTT. We name the number of RTT rounds that pass until there is a packet loss as the NL round (Figure 2). 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.

4.1. Handoff Packet Loss Rate (P_H)

Figure 2 will be used for explaining the behavior of TCP during a handoff event. The variables FTT_1/BTT_1, and

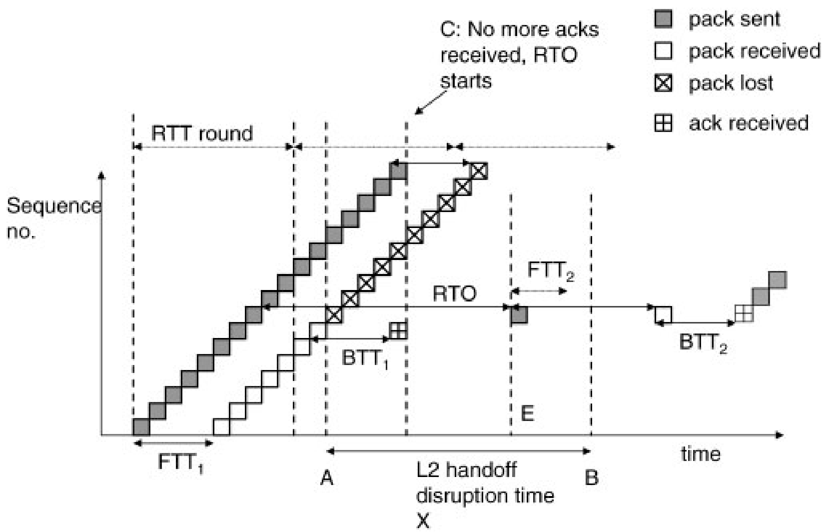

Fig. 2. Packet-level TCP behavior at the sender during a handoff.

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 $t_A + \mathrm{RTO}$, the TCP sender retransmits the first lost packet. Even more, if $t_B > t_C + \mathrm{FTT}_1$, 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 $t_B \leq t_C + \mathrm{FTT}_1$ 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 $t_C = t_A + \mathrm{BTT}_1$): $t_B \leq t_C + \mathrm{FTT}_1 \Rightarrow T_H \leq \mathrm{RTT}_1$. 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:

$$P_H = P[L < T_H] \tag{12}$$

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 [23]. On the other hand, the disruption time T_H is a parameter that depends of the mobility management a protocol, and was derived in the previous section. Therefore, we can easily calculate P_H from Equation (12).

4.2. 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 timeout (TO) and we will also calculate the number of packets sent during the TO events.

4.2.1. 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 [10], that describes congestion window evolution as a function of the packet loss rate, we have:

$$E[W_x^h] = \frac{2}{b}\left(\frac{2+b}{6} + \sqrt{\frac{2b(1-P_x-P_H)}{3(P_x+P_H)} + \left(\frac{2+b}{6}\right)^2}\right) \tag{13}$$

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 (TD) events, the throughput formula would be of course similar to [10]:

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

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

4.2.2. 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:

$$P_{HW}(w, i) = \frac{(1-P_H)^i P_H}{1-(1-P_H)^w} + \frac{(1-P_1)^i P_1}{1-(1-P_1)^w} \tag{14}$$

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 [10]:

$$G(n, m) = \begin{cases} P_H^m(1-P_H) & \text{if } m \leq n-1 \\ P_H^m & \text{if } m = n \end{cases} \tag{15}$$

Now, the probability that at most two of them were acknowledged is:

$$g(k) = \sum_{i=0}^{2} G(k, i) \tag{16}$$

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:

$$P_{\mathrm{TO}_x}(w) = \sum_{k=0}^{2} P_{\mathrm{HW}}(w, k) + \sum_{k=3}^{w} P_{\mathrm{HW}}(w, k) g(k) \tag{17}$$

Note also that $P_{\mathrm{TD}_x}(w) = 1 - P_{\mathrm{TO}_x}(w)$.

4.2.3. TO duration

Besides the probability for a TO to happen, we have to calculate the average duration of a TO, based on the probability that $t_{\mathrm{B}} > t_C + \mathrm{FTT}_1$ (Figure 2). 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 $T_{\mathrm{H}} = t_{\mathrm{B}} - t_{\mathrm{A}}$, the number of retransmitted packets depends on the duration of the TO 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 [4]:

$$L_{\mathrm{h}} = \begin{cases} (2^j - 1)\mathrm{RTO}_0 & \text{if } j < 6 \\ (63 + 64(j - 6))\mathrm{RTO}_0 & \text{if } j \geq 7 \end{cases} \tag{18}$$

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

$$j = \begin{cases} \log_2\left(\frac{L_{\mathrm{h}}}{\mathrm{RTO}_0} + 1\right) & \text{if } L_{\mathrm{h}} \leq (2^6 - 1)\mathrm{RTO}_0 \\ \frac{L_{\mathrm{h}}}{\mathrm{RTO}_0} + 5 & \text{otherwise} \end{cases} \tag{19}$$

Therefore, the number of experienced TOs will be obtained by taking the $\lceil\ \rceil$ of $(L_{\mathrm{h}})/(\mathrm{RTO}_0)$. 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 $t_{\mathrm{C}} < t < t_{\mathrm{B}}$:

$$E[S_{\mathrm{h}}] = \begin{cases} \log_2\left(\left\lceil \frac{T_{\mathrm{H}}}{L_{\mathrm{h}}} \right\rceil + 1\right) & \text{if } T_{\mathrm{H}} \leq (2^6 - 1)\mathrm{RTO}_0 \\ \left\lceil \frac{T_{\mathrm{H}}}{L_{\mathrm{h}}} \right\rceil + 5 & \text{otherwise} \end{cases} \tag{20}$$

and the duration of the induced TOs is $E[L_{\mathrm{h}}] + T_{\mathrm{H}}$.

4.2.4. Final throughput formula

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

$$T_x^h = \frac{\frac{1-P_x}{P_x} + E[W_x^h] + P_{\mathrm{TO}} E[S_{\mathrm{h}}]}{\mathrm{RTT}_x(E[W_x^h]b/2 + 1) + P_{\mathrm{TO}}(T_{\mathrm{H}} + E[L_{\mathrm{h}}])}$$

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}. Concerning the latency estimate, we used this throughput formula and the size of the transferred file in order to derive the expected transfer time.

5. TFRC Throughput and Latency

After dealing with TCP, we proceed with the characterization of the rate control protocol TFRC. Our goal is quantify the effect of mobile handoff on a rate control protocol which 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:

$$T_{\mathrm{TFRC}} = \frac{s}{\mathrm{RTT}\sqrt{\frac{2p}{3}} + t_{\mathrm{RTO}}\left(3\sqrt{\frac{3p}{8}}\right) p(1 + 32p^2)} \tag{21}$$

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:

$$t_{\text{inter}} = \frac{s\sqrt{\text{RTT}_0}}{T * M} \tag{22}$$

where M is the average of the square roots of the RTTs calculated using an explicit window moving average (EWMA), and RTT_0 is the most recent RTT sample [20]. Given that S_i and S_j are the transmit times of two packets sent successively, the inter-packet spacing at the source is given as $S_i - S_j = t_{\text{inter}}$. However, the packet spacing at the receiver $R_i - R_j$, depends on the network queuing delays, which is the only cause of jitter in TFRC. Note that, Equation (21) 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 5. 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 s 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 nth RTT estimate is calculated as follows:

$$\text{RTT}_n = \mathrm{d}f * \text{RTT}_{n-1} + (1 - \mathrm{d}f) * \sqrt{\text{RTT}_{\text{cur}}} \tag{23}$$

Algorithm 1 TFRC rate estimation.

if $p > 0$ **then**
  $X\_calc = T(p, RTT, T\_0)$
  $tfrc\_x = max\left[min(X\_calc, 2 * X\_recv), \frac{s}{tmbi}\right]$
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

with RTT_{cur} be the last real RTT measurement. 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:

$$E[S_{\text{h}}] = \sum_{i=0}^{\left\lceil \frac{T_{\text{H}}}{\text{RTT}_{\text{cur}}} \right\rceil} \frac{1}{2^i} T_{\text{TFRC}} \times \text{RTT} \tag{24}$$

Therefore, for two links with different characteristics, p_1, RTT_1 and p_2, RTT_2, for links 1,2 or when $t < t_{\text{A}}$ or $t > t_{\text{C}}$, the actual rate at the sender will be given by the basic TFRC formula:

$$T_{\text{TFRC}}^{A^- B^+} = \max\left(2 \min(T_{\text{TFRC}}, 2(1 - P_{1,2})T_{\text{TFRC}}, \frac{s}{tmbi}\right) \tag{25}$$

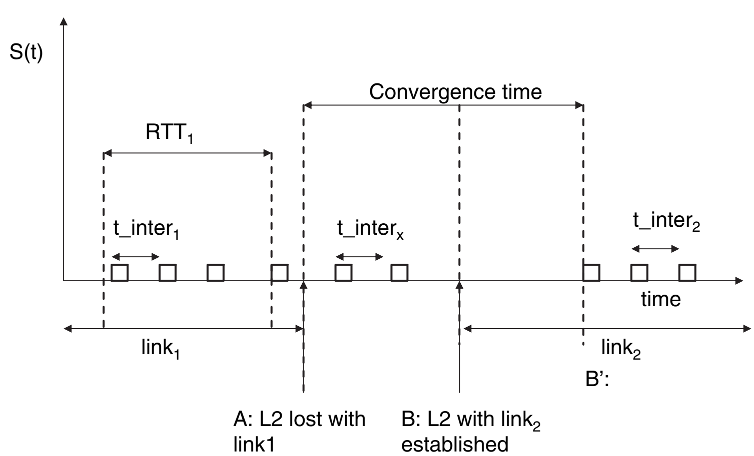

Fig. 3. TFRC behavior during IP layer handoff.

when $t_A < t < t_B$ then:

$$T_{\mathrm{TFRC}}^{\mathrm{AB}} = \frac{1}{T_{\mathrm{H}}}\left(\sum_{i=0}^{\left\lceil \frac{T_{\mathrm{H}}}{\mathrm{RTT_{cur}}} \right\rceil} \frac{1}{2^i} T_{\mathrm{TFRC}} \times \mathrm{RTT}\right) \quad (26)$$

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

$$T_{\mathrm{TFRC}}^{\mathrm{BC}} = \frac{1}{t_{\mathrm{cv}}}\left(\sum_{n=1}^{n_{\mathrm{cv}}} T_{\mathrm{TFRC}}(\mathrm{RTT}_{n\_}cv) \times \mathrm{RTT}_n\right) \quad (27)$$

We define as convergence time, t_{cv}, the time needed for TFRC to obtain its fair share of the bandwidth on the new link. Equation (23) is be expanded and written as:

$$\mathrm{RTT}_n = \mathrm{d}f^n * \mathrm{RTT}_0 + \mathrm{d}f^{n-1}(1-\mathrm{d}f) * \sqrt{\mathrm{RTT_{cur}}} + \cdots + (1-\mathrm{d}f) * \sqrt{\mathrm{RTT_{cur}}} \quad (28)$$

which if we solve for n gives:

$$n_{\mathrm{cv}} = log_{\mathrm{d}f}\left(\frac{\mathrm{RTT}_n - (1-\mathrm{d}f) * \sqrt{\mathrm{RTT_{cur}}}}{\mathrm{RTT}_0(1-\mathrm{d}f) - (1-\mathrm{d}f) * \sqrt{\mathrm{RTT_{cur}}}}\right) \quad (29)$$

Therefore, n_{cv} will give the number of RTT rounds needed in order for the RTT_n estimate to converge to the $\mathrm{RTT_{cur}}$. In the handoff case, $\mathrm{RTT_{cur}}$ 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 $\mathrm{RTT_{cur}}$. So the total convergence time will be:

$$t_{\mathrm{cv}} = T_{\mathrm{H}} + n_{\mathrm{cv}} * \mathrm{RTT}_n \quad (30)$$

While the throughput is calculated from Equations (25)–(27). The expected latency will depend on the size of the file and when the transfer ended. For example a chunk of the file might have been transferred before the handoff (t_{A-}), and whether the rest will finish between t_A and t_B will depend on the duration of handoff $T_{\mathrm{H}} = t_B - t_A$, and of course the throughput Equation (26) that determines that period.

6. Experiments for Model Validation

The WLAN topology that forms the basis of the simulations described in this section is shown in Figure 4. The values for bandwidth and delay of the links are 1 Mbps/100 ms and 500 Kbps/100 ms for the GFA links, and two FA links, respectively. The case study simulated is now described—Initially, the MH initiates an FTP data flow from the CH. According to the scenario, at the time instant 50 s, the MH starts moving away from the first access point (AP), at a speed of 10 m/s, 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 [24] network simulator while the simulation parameters can be seen in (Table I).

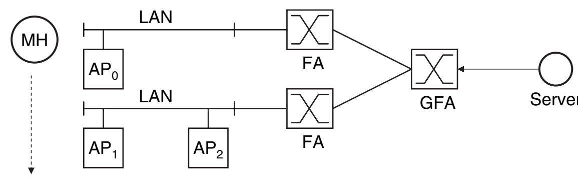

Fig. 4. Simulation topology for WLAN handoff experiments.

6.1. HMIP and MIP-RO

With this experiment, we want to evaluate the effect of the handoff disruption time T_{H}, on the throughput and latency of a session between the server and the client. Throughput results are shown in Figure 5. 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 5 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,

Table I. Simulation parameters.

Transport protocol		Mobility protocol	
RTT	100/500 ms	FA processing delay (s_f)	10 ms
$W_{\max}$	6 MB	HA processing delay (s_h)	25 ms
W_0	1 segment	GFA processing delay (s_g)	15 ms
MSS	1460	Wireless cost multiplier (ρ)	10
RTO_0	0.2/1 s	Distance cost multiplier (δ)	0.05

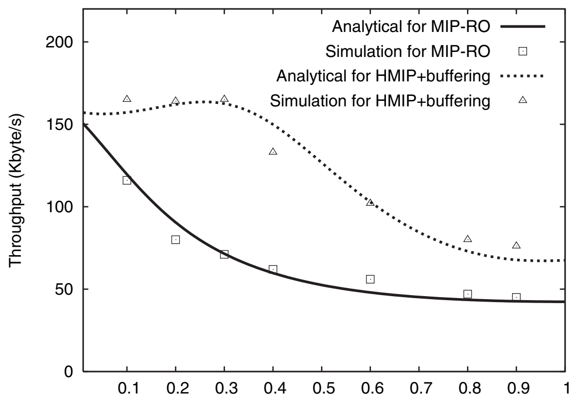

Fig. 5. Throughput results for a TCP session of 15 s and handoff between two different WLAN subnets.

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 6 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 6, we see that 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 in time, the buffer at the previous AP should not overflow frequently since it does not receive packet bursts as with TCP.

Latency results are presented in Figures 7 and 8. For the combination MIP-RO and TCP, latency is increased continuously, after the disruption time is larger than the

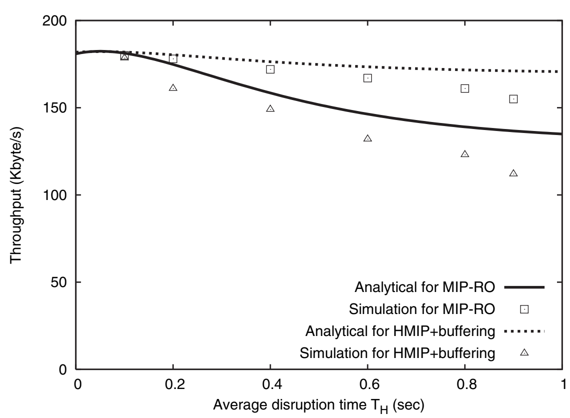

Fig. 6. Throughput results for a TFRC session of 15 s and handoff between two different WLAN subnets.

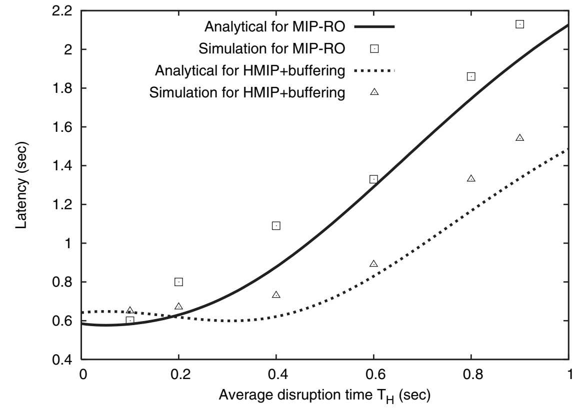

Fig. 7. Effect of disruption time on TCP latency for a session of 15 s. Parameters: RTT = 200 ms, $RTO_0 = 400$ ms, MSS = 1460 bytes, $W_0 = 1$ segment, $W_{max} = 4$ MB.

one way end-to-end delay. This results in the first packet losses and the first retransmitted packets. With HMIP, the latency incurred due to a handoff is bounded by the latency between GFA and MH. The use of buffering eliminates a series of packet losses, reduces RTO expirations and fast retransmissions leading to reduced latency as we expected from our analytical equation and can be seen in Figure 7. However, as the disruption time increases, packet losses are observed. Analytical results for TFRC are shown in Figure 8. We can see that the latency for the transport of a specific data chunk, is not penalized as severely as with TCP, due to the smooth variations of the TFRC rate control algorithm.

6.2. 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

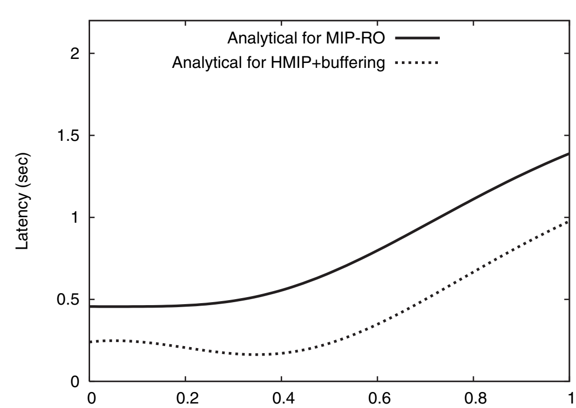

Fig. 8. Effect of disruption time on TFRC latency for a session of 15 s. Parameters: $p = 0.02$, $RTO_0 = 1$ s, $s = 1500$ bytes.

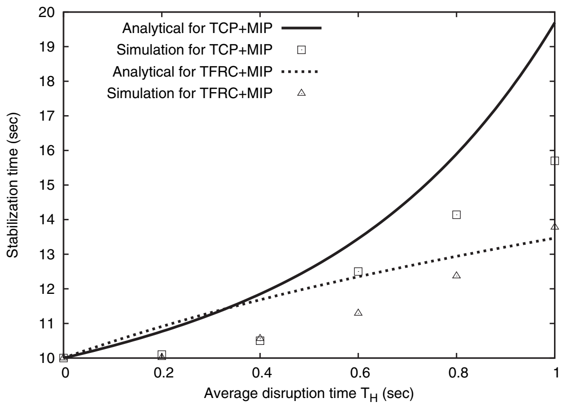

Fig. 9. Required recovery time versus disruption time for both TCP and TFRC.

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 9, 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 9 also indicates.

7. Conclusions

In this paper we presented a model for studying the effects of wireless handoffs in 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%. An important observation from the conducted experiments is that the use of a forwarding buffer at the old access network, can significantly improved the delivered throughput. If the requirements of the system specify that no packet loss should take place, the rule of thumb for TCP, is that the buffer size should be equal to the bandwidth-delay product of the old access path times the expected disruption time. Concerning TFRC, we found that the required forwarding buffer size should be surprisingly bigger by 60% than TCP. The reason for that is the slow responsiveness of TFRC which does not drastically cut its rate, resulting in the need of a larger buffer. We also introduced in this paper the notion of the “recovery period,” which is defined as the time required for the transport protocol to achieve the nominal throughput in a new link, after a handoff. 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, which is persistent on sending new packets to the network.

References

1. Perkins C. IP Mobility Support for IPv4. RFC 3220, 2002.
2. Stojmenovic I. *Handbook of Wireless Networks and Mobile Computing*. John Wiley and Sons: New York, 2002.
3. Pierre Reinbold and Olivier Bonaventure. A Survey of IP micro-mobility protocols. cite-seer.nj.nec.com/reinbold02survey.html, 2002.
4. Stevens R. *TCP/IP Illustrated* Vol. 1. Addison-Wesley: New York, 1994.
5. Balakrishnan H. *Challenges to Reliable Data Transport over Heterogeneous Wireless Networks*. PhD thesis, UC Berkeley, 1998.
6. Holland G, Vaidya N. Analysis of TCP performance over mobile ad hoc networks. In *ACM International Conference on Mobile Computing and Networking (MOBICOM)*, 1999.
7. Monks J, Sinha P, Bharghavan V. Limitations of TCP-ELFN for ad hoc networks. In *MOMUC*, 2000.
8. Chakravorty R, Cartwright J, Pratt I. Practical Experience with TCP over GPRS. In *GLOBECOM*, 2002.
9. Chan MC, Ramjee R. TCP/IP performance over 3G wireless links with rate and delay variation. In *ACM International Conference on Mobile Computing and Networking (MOBICOM)*, 2002.
10. Padhye J, Firoiu V, Towsley D. Modeling TCP reno performance: a simple model and its empirical validation. *IEEE/ACM Transactions on Networking* 2000; **8**(2): 133–145.
11. Sikdar B, Kalyanaraman S, Vastola K. Analytic models for the latency and steady-state throughput of TCP Tahoe, Reno and SACK. In *INFOCOM*, 2001.
12. Cardwell N, Savage S, Anderson T. Modeling TCP Latency. In *INFOCOM*, 2000.
13. Chen W, Ho J. Performance Analysis of Adaptive Location Management for Mobile IP. Technical Report 97-CSE-13, Southern Methodist University, 1997.
14. Akyildiz I, McNair J, Ho J, Uznalioglu H, Wang W. Mobility management in next-generation wireless systems. *IEEE Proceedings* 1999; **87**(20): 1347–1384.
15. Xie J, Akyildiz IF. A Novel Distributed Location Management Scheme for Minimizing Signaling Costs in Mobile IP. *IEEE Transactions on Mobile Computing* 2002; **1**(3): 163–175.
16. Blondia C, Van den Wijngaert N, Willems G, Casals O. Performance Analysis of Optimized Smooth Handoff in Mobile IP. In *MSWiM*, 2002.
17. Blondia C, Casals O, Cerda L, Van den Wijngaert N. Performance Comparison of Low Latency Mobile IP schemes. In *WiOpt*, 2003.

18. McNair J, Akyildiz I, Bender M. Handoffs for real-time traffic in Mobile IP Version 6 networks. In *IEEE ICC*, 2001.
19. Pack S, Choi Y. Performance analysis of hierarchical mobile IPv6 in IP-based cellular networks. In *PIMRC*, 2003.
20. Handley M, Floyd S, Pahdye J, Widmer J. TCP Friendly Rate Control (TFRC): Protocol Specification. RFC 3448, 2003.
21. Gurtov A. Effect of vertical handovers on performance of TCP-friendly rate control. In *ACM Computer Communications Review*, 2004.
22. Lakshman T, Madhow U. The performance of TCP/IP for networks with high bandwidth-delay products. *IEEE/ACM Transactions on Networking* 1997; **5**: 336–350.
23. Chakareski J, Girod B. Rate-distortion Optimized Packet Scheduling and Routing for Media Streaming with Path Diversity. In *IEEE Data Compression Conference (DCC)*, 2003.
24. Network simulator.

Author's Biography

Antonios Argyriou received his undergraduate degree in electrical and computer engineering from Democritus University of Thrace, Greece, in 2001, and the M.S. and Ph.D. degrees in electrical and computer engineering as a Fulbright scholar from the Georgia Institute of Technology in 2003 and 2005 respectively. His research interests include multimedia communications, modeling of wireless and mobile systems and network performance evaluation. He has published more than 20 papers in international journals and conferences. He is a member of IEEE and ACM.