Mobile Networking

Chris Blondia (Ed.)[1], Nik Van den Wijngaert[1], Gert Willems[1],
Olga Casals (Ed.)[2], Llorenc Cerda[2], Marcelo Bagnulo[3], Ignacio Soto[3]

[1] University of Antwerpen, Department of Mathematics and Computer Science,
Universiteitsplein 1 B-2610 Antwerpen, Belgium
{chris.blondia, nik.wijngaert, gert.willems}@ua.ac.be
http://win-www.ruca.ac.be/u/pats/
[2] Technical University of Catalonia, Department of Computer Architecture,
Jordi Girona, 1-3, Modul D6 E-08034 Barcelona, Spain
{olga, cerda}@ac.upc.es
http://www.ac.upc.es/recerca/
[3] Universidad Carlos III, Departamento de Ingeniera Telematica
Madrid, Spain

Abstract. We point out the different performance problems that need to be addressed when considering mobility in IP networks. We also define the reference architecture and present a framework to classify the different solutions for mobility management in IP networks. The performance of the major candidate micro-mobility solutions is evaluated for both real-time (UDP) and data (TCP) traffic through simulation and by means of an analytical model. Using these models we compare the performance of different mobility management schemes for different data and real-time services and the network resources that are needed for it. We point out the problems of TCP in wireless environments and review some proposed enhancements to TCP that aim at improving TCP performance. We make a detailed study of how some of micro-mobility protocols namely Cellular IP, Hawaii and Hierarchical Mobile IP affect the behavior of TCP and their interaction with the MAC layer. We investigate the impact of handoffs on TCP by means of simulation traces that show the evolution of segments and acknowledgments during handoffs.

1. Introduction

Mobility support in IP networks is in the final steps of the standardization process within the IETF. This specification defines basic tools needed to cope with mobile nodes. However, it is well known in the research community that multiple ulterior optimizations are required in order to provide a service comparable with those provided in a non-mobile environment. In this chapter, a number of critical issues with respect to mobility are addressed.

The state-less auto-configuration procedure is not optimized for real-time services, since obtaining a Care-of-Address requires a Duplicate Address Detection, which may take a default value of one second. This latency is unacceptable in an environment for interactive voice communications. Section 2 presents a possible modification of the

M. Smirnov et al. (Eds.): COST 263 Final Report, LNCS 2856, pp. 180-206, 2003.

Duplicate Address Detection mechanism of IPv6 with the aim of reducing the time needed to perform a handover.

In section 3, protocols are investigated that aim at realizing seamless handover, i.e. handovers without packet loss and without introducing extra packet delay and delay jitter. Their performance is studied and the influence of several system parameters is investigated.

The TCP transport protocol used in the Internet is a variable window protocol where losses are considered as congestion signals. This may cause TCP to behave poorly in wireless networks, which are characterized by losses due to transmission errors and handoffs. In section 4 we investigate the impact of transmission errors on TCP and the causes of packet losses during the handoffs.

2. Optimizations for Mobility Support in IPv6

2.1. Introduction

IPv6 introduces enhanced facilities for mobility support, among which we can find State-less Auto-configuration mechanism. However, general auto-configuration procedure is not optimized for mobile nodes as it will presented next. Since in order to take full advantage of multimedia capabilities of current mobile devices, the network infrastructure must provide an uninterrupted flow of information to appropriately support real time traffic. However, the requirement for performing Duplicate Address Detection (DAD) in the address autoconfiguration mechanism limits the performance of mobility in IPv6, provided by Mobile IPv6 [1]. Using this protocol, a Mobile Node (MN) that joins a subnet must configure an on-link address in that subnet, the Care-of-Address (CoA), before being able to communicate. According to the Stateless Address Autoconfiguration mechanism presented in [2], before using the CoA the MN must perform DAD for that address in order to guarantee its uniqueness on the link. It should be noted that DAD is a time consuming process. Although this it is not an issue for a desktop computer that is booting, the time required for DAD is critical in a mobile environment, since during this time the MN can not communicate and besides, all active communications of the MN are interrupted. The time required to perform DAD has a default value of one second [2], [3], a value subjectively deemed as not acceptable for interactive voice communications [4]. The Mobile IPv6 (MIPv6) specification [1] identifies the aforementioned problem and states that a MN can decide not to perform DAD, pointing this as a trade-off between safety and the time needed for the DAD procedure.

In this section, the usage of random numbers to create the Interface Identifier part of the IPv6 addresses is explored, and the risk of using these addresses without previously performing DAD is assessed. It should be noted that this solution is not restricted to a particular data-link layer technology, although it can be optimized in particular cases, such as GPRS, in which collision can be avoided by the GGSN (Gateway GPRS Support Node).

2.2. Risk Assessment

In this sub-section we will asses the risk of using randomly generated Interface Identifiers (IIDs) in IPv6 aggregatable addresses [5] without previously performing DAD. In order to do that, we will quantify the probability of a duplicate address event in several relevant scenarios and we will compare it with the probability of other critical events.

2.2.1. Duplicate Address Event Probability Calculation and Bounding

In the considered hypothesis, the Interface Identifier part of the IPv6 address is generated randomly, meaning that the node will use a 64 bit long random number as the IID. Actually, only 62 bits of the IID will be generated randomly, since the IID´s semantics defined in [6] imposes that the u bit must be set to "local" and the g bit must be set to "individual".

Considering that *n* is the number of possible IIDs (i.e. $n = 2^{62}$) and *k* is the number of interfaces (i.e. mobile nodes) on the same link, we will now calculate the probability of collision of two or more randomly generated IIDs:
We will represent the *k* IIDs in a link as a sequence of 62 bit long random variables I_i:

$I_1, I_2, \dots, I_k$ sequence of random integer variables with uniform distribution between 1 and $n\,(k \le n)$

We would like to obtain the probability that two or more I_is collide, i.e. $I_i = I_j$

This is well known mathematical problem, called the "birthday problem", whose classical formulation is as follows: we want to calculate the probability that in a group of *k* people, al least two of them have the same birthday.

We model the birthday as a integer random variable, with uniform distribution between 1 and *n* (in this particular case *n* is the number of possible birthdays i.e. *n*=365)

Then, the number *N* of ways that we can choose *k* values out of *n* with no duplicates would be:

$$N = n \cdot (n-1) \cdot \dots \cdot (n-k+1)$$

On the other hand, the number of possibilities for choosing *k* elements out of *n*, without the restriction of not having any duplicates is n^k

Then, the probability of not having a collision when we select *k* elements out of *n* is:

$$P_{NO}(n,k) = \frac{n!}{(n-k)! \cdot n^k}$$

So, the resulting expression for the probability of the collision of one or more I_i is:

$$P(n,k) = 1 - \frac{n!}{(n-k)!\,n^k} \qquad \textbf{(1)}$$

In our particular case, $n = 2^{62}$, and k may vary depending on the considered scenario. We will now obtain an upper bound to $P(n,k)$ in order to simplify calculations (especially to avoid $n\,!$ computation)

Performing simple computations in equation (1), we easily obtain:

$$P(n,k) = 1 - \left\{ 1 \cdot \left(1 - \frac{1}{n}\right) \cdot \left(1 - \frac{2}{n}\right) \ldots \left(1 - \frac{k-1}{n}\right) \right\} \tag{2}$$

Since

$$\forall i \in [1,2,\ldots,k-1] \Rightarrow \frac{i}{n} \le \frac{k-1}{n},$$

and considering that $k < n$, then

$$1 - \left(1 - \frac{1}{n}\right)\left(1 - \frac{2}{n}\right) \ldots \left(1 - \frac{k-1}{n}\right) \le 1 - \left(1 - \frac{k-1}{n}\right)^{k-1}$$

Applying this last result to equation 2, we can obtain the following bound B:

$$P(n,k) \le \frac{n^{k-1} - (n-k+1)^{k-1}}{n^{k-1}} = B \tag{3}$$

We will next perform some calculations in order to quantify the order of magnitude of the probabilities involved:

We will bound $P(n,k)$ for the following values of *k*, which we consider to be representative of usual situations

$$P(2^{62},20) \le 7.8e - 17$$

$$P(2^{62},100) \le 2.1e - 15$$

$$P(2^{62},500) \le 5.4e - 14$$

$$P(2^{62},1000) \le 2.2e - 13$$

$$P(2^{62},5000) \le 5.4e - 12$$

In order to fully seize the magnitude of the probabilities stated above, we can compare them with the probabilities of some rare events. For instance, according to Table 1.1 in [7], the probability of being killed by a lightning (per day) is about 1.1 e-10. Then, a mobile phone user should be more worried about being killed by a lightning in a given day than to have an interface identifier repeated when he performs a handoff.

Another relevant parameter that can be considered when evaluating the above probability, is the probability of a failure in a network device, since this failure would have similar effects i.e. the user can not continue the communication. So, the probability that a network device were not working properly in a given moment (when the user joins the network, for instance) can be calculated as follows:

$$P_{NEFails} = \frac{MTTR}{MTBF + MTTR}$$

Being MTTR the Meat Time To Repair and MTBF the Mean Time Between Failures.

Good network equipment can have an MTBF of 300,000 hours and if we suppose that some backup device is available, the MTTR stands for the time needed to replace the faulty element, e.g. 0.1 hour (6 minutes). In this case, $P_{NEFails} = 3.3e - 7$.

We can see that $P_{NEFails}$ is several orders of magnitude higher than $P(n,k)$ in the cases calculated above.

2.2.2. Scenarios

We have quantified the probability of a collision of two or more IIDs. However, this probability is not the most relevant parameter when we try to evaluate and compare the probability of failure of the system, since a mobile user will join multiple links in a given period. So, it is relevant to quantify the probability of at least one collision when a user performs multiple handoffs.

As we stated above, $P(n,k)$ is the probability of a collision of two or more IIDs when there are *k* interfaces in the same link. Hence, it can be derived that the probability of having at least one collision after joining *m* links is:

$$P(n,k,m) = 1 - (1 - P(n,k))^m \quad \textbf{(4)}$$

According to the bound *B* presented in equation 3 and considering that both $P(n,k)$ and *B* are lower than 1, we can infer the following bound:

$$P(n,k,m) \leq 1 - (1 - B)^m \quad \textbf{(5)}$$

Therefore, in order to estimate the probability of a collision event during a given period, for instance a year, we must first establish a reasonable number of handoffs per day. If we consider 140 handoffs per day, which seems to be a considerable number, this would mean about 50.000 handoffs per year, i.e. *m*=50.000. Then the probability of having at least one collision over a year during which the mobile node has joined 140 links of 500 nodes per day is:

$$P(2^{62}, 500, 50.000) \leq 2.7e - 9 \quad \textbf{(6)}$$

And, if the considered links have 5.000 nodes each, instead of 500, the probability is:

$$P(2^{62}, 5.000, 50.000) \leq 2.7e - 7 \quad \textbf{(7)}$$

Considering that each time there is a collision there are two users affected, and not taking into account the collision of 3 or more IIDs for this estimation, there will be 6 users out of 1.000.000.000 that will have a communication problem during this year, if users make 140 handovers per day in networks containing 500 interfaces. In the case that users make 140 handovers per day in networks containing 5.000 interfaces, there will be 6 users out of 10.000.000 that will have a communication problem during this year. This probability could be contrasted with some network availability data provided by, for instance, mobile operators, but this data has proven to be extremely difficult to find.

2.3. Implementation Issues

In this section, we will address some implementation issues regarding random IIDs generation and related security concerns.

2.3.1. Random Numbers Generation

When considering the usage of random IIDs, the random number generation process must be properly addressed since it is essential to guarantee the statistic uniqueness of the identifier. Several methods have been proposed [8] to generate pseudo-random numbers based on information available in most platforms, such as laptops. However, in some cases, such as mobile phones, the resources required to perform appropriate random number generation may not be available. In such cases, it should be noted that it is not necessary to create the identifier in real time, as long as randomness were guaranteed. This means that when the node joins the network the identifier could have been created already in advance to a network change. It could even be pre-configured in the interface driver with the node using the same identifier without changing the probabilities calculations stated above; this is analogous to the day of birth in the birthday problem. This would reduce the complexity in the nodes, although a mechanism should be provided in order to solve recurrent collisions, caused for example, when two habitual users of the same segment collide.

2.3.2. Security Concerns

Randomly generated IIDs have also been considered in order to improve security. In particular, its usage has been proposed to ensure anonymity [9] and even to carry cryptographic information when Cryptographically Generated Addresses are used [10]. These proposals are fully compatible with the solution of this document so they can get the benefit of better performance by avoiding DAD.

3. Mobility Management in IP Networks

3.1. Introduction

In this Section, we investigate methods to solve the problem of host mobility on layer 3. The basic problem is how to route packets to a Mobile Node (MN) when it moves from one point of attachment to another. The dual character of an IP address plays a central role: on the one hand the IP address uniquely identifies a MN, and on the other hand it identifies the location of the MN.

In what follows, we discuss several solutions to this host mobility problem. These solutions should ensure (as much as possible) a seamless handover, i.e. a handover without packet loss and without introducing extra packet delay or delay jitter. In addition, the messages required to implement these solutions and their storage and processing needs in routers should be minimal. Moreover, these mechanisms should have a maximal compatibility with other Internet protocols, in particular with QoS provisioning schemes. Scalability is another important requirement that has to be fulfilled.

3.2. IP Mobility Protocols

In this subsection, we give an overview of important IP mobility protocols. We start with the standard IETF solution, namely Mobile IP. Next we discuss two important classes of micro-mobility mechanisms, namely hierarchical tunnelling mobility protocols and host-specific routing protocols. Finally we discuss some low latency handoff schemes.

3.2.1. Mobile IP

Mobile IP (MIP) is the standard IP solution for host mobility. MIP uses two addresses, a home address (HoA) acting as a permanent Layer 3 identifier and the Care-of-Address (CoA) acting as a temporary routable address to indicate the actual location of the MN.

An agent located in the MN's home network, called the Home Agent (HA) maintains the binding <HoA, CoA>. The HA intercepts packets destined for the MN and tunnels them to the MN's CoA. When the MN changes subnet (and hence CoA), it needs to registrer the new CoA with the HA (i.e. a binding update). This registration may take a considerable time which, together with the delays introduced by the establishment of the new tunnels, may cause unacceptable packet loss and the introduction of an important signalling overhead. For the basic operation of MIP, we refer to [13] and [1].

Due to its robustness and simplicity, MIP seems to be a suitable protocol for handling global or macro-mobility (e.g. mobility between domains owned by different operators). Local (and hence more frequent) handovers need to be handled by more optimized, so-called micro-mobility protocols. In what follows, we consider two classes of micro-mobility protocols, namely schemes based on hierarchical tunnelling and protocols based on host-specific routing.

3.2.2. Hierarchical Tunneling Mobility Protocols

Hierarchical tunneling schemes, such as MIP Regional Registration (IPv4) ([14]) or Hierarchical Mobile IP (MIPv6) ([15]), handle local movements by maintaining the MN's location information in a distributed form by a hierarchy of Foreign Agents (FA) organized in a tree structure. Movements of the MN between different points of attachment only involve binding updates at the optimal point in the tree. In such a system, the HA tunnels the traffic to the root of the tree and each intermediate FA on the route to the MN decapsulates and re-encapsulates the packets as they are forwarded down the tree towards the MN's actual position.

3.2.3. Host-Specific Routing Protocols

This class of micro-mobility schemes avoids the overhead introduced by the tunnelling schemes by using host-specific routes towards the current location of the MN. These host-specific routes are created and maintained by explicit signalling or by snooping data packets. We discuss two important examples, namely Cellular IP (CIP) ([16]) and Handover Aware Wireless Access Internet Infrastructure (HAWAII) ([17]).

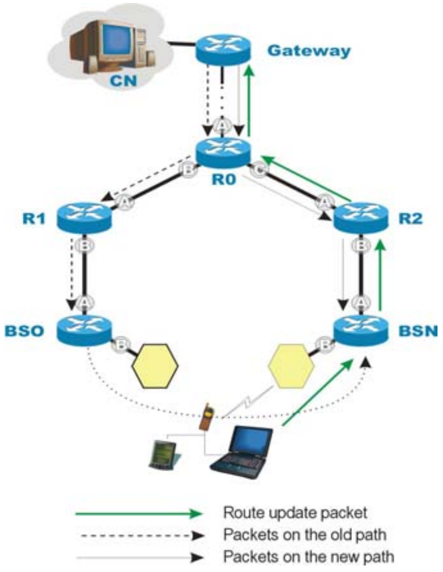

Figure 1: Cellular IP

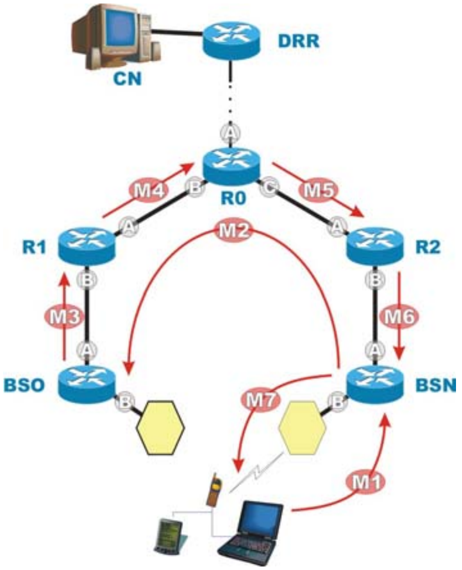

Figure 2: HAWAII

Cellular IP

A Cellular IP ([16], [18], [19]) access network is connected to the Internet via a gateway router. A MN attached to an access network will use the IP address of the gateway as its Mobile IP CoA.

After power up a MN has to inform the gateway router about its current point of attachment by means of a route update message. This message is received by the base station and forwarded to the gateway on a hop-by-hop basis. Each Cellular IP node maintains a route cache in which it stores host based routing entries. The path taken by uplink packets is cached by all intermediate nodes. To route downlink packets addressed to the MN, the path used by recently transmitted packets from the MN and which have been cached is reversed.

As the routing entries in the nodes are soft state, a MN that has no data to transmit, has to send periodically special IP packets to maintain its routing path. Paging is used to route packets to MNs that have not been transmitting or receiving data for a while and whose routing cache entries have timed out. Paging caches are not maintained in each node and have a longer timeout value. If a node without a paging cache, receives a packet for a MH for which it has no routing cache entry, it will broadcast it to all its downlink neighbours.

In Cellular IP a mobile host initiates a handoff (based on signal strength measurements) by sending a route update packet to the new base station (see Figure 1). This packet travels in a hop-by-hop manner from the base station to the gateway router and reconfigures the route cache entries in the Cellular IP nodes along its way. Cellular IP supports different types of handoff schemes. In the hard handoff scheme, the wireless interface of the MN switches from one BS to another at once. In the semi-soft handoff scheme, the MN is assumed to be able to transmit a route update packet to the new base station while still listening to the old base station. By duplicating packets, the packet loss observed in hard handoff may be reduced.

HAWAII

HAWAII ([17]) creates host-specific routing entries in the routers by explicit signaling messages triggered by the MN. Four different path set-up schemes have been defined which are classified into two different classes: two forwarding schemes, namely Multiple Stream Forwarding (MSF) and Single Stream Forwarding (SSF) and two non-forwarding schemes, namely Unicast Non-Forwarding (UNF) and Multicast Non-Forwarding (MNF). We limit the description and the analysis (see Section 2.3.1.) to the MSF scheme.

The MSF schemes is described by means of the following messages (see Figure 2). At the instant of handoff, the old base station, BSO, loses contact with the MN and at the same time the MN sends a MIP registration message (M1) to new base station, denoted by BSN. The latter sends a path setup update message M2 to the BSO. When M2 arrives at BSO, the BSO starts to forward all packets with destination MN via router R1, including those packets that arrive after the handoff instant and that were stored in a forwarding buffer at the BSO. For that purpose, BSO adds a forwarding entry to its routing table indicating that packets for MN should leave the BSO via interface A. BSO sends the path setup message (M3) to R1, who adds a forwarding entry to its routing table indicating that packets for the MN should leave the R1 via interface A. R1 sends the path setup message (M4) to R0, who adds a forwarding entry indicating that packets for the MH should leave the R0 via interface C. From this instant on, all packets arriving at router R0 are sent directly to BSN. The path setup message continues (M5 and M6) triggering similar actions until it reaches BSN.

3.2.4. Low Latency Handoff Schemes

In this paragraph we discuss three different handoff schemes that aim at low latency handoffs: smooth handoff, pre-registration and post-registration.

Optimized Smooth Handoff

Optimized smooth handoff was proposed in [20]. We make the assumptions that the route optimization problem (i.e. the triangle routing problem) is assumed solved adequately. Furthermore, in what follows, we assume a network with a hierarchical Foreign Agent (FA) architecture. As soon as the MN has obtained its new regional CoA, it will register this address with its GFA. This is achieved by sending a

Registration Request Message to the new FA, who on his turn sends a registration message to the GFA. As part of this registration procedure, the MN may add to the registration request message a Previous Foreign Agent Notification extension. The new FA will then send a Binding Update Message to the previous FA, with the request to reply with an acknowledgement. This acknowledgment is tunnelled to the new FA, who should forward it to the MN (this may involve unauthorized traffic to the MN). In this way the MN is notified that the previous FA has the new CoA of the MN. This binding update the previous FA receives from the new FA, allows packet forwarding mechanism. When a packet arrives in the previous FA, the binding cache is checked and the packet is tunnelled to the new FA, who delivers it to the MN. However, packets arriving at the previous FA after the MN left and before the binding update message from the new FA is received are lost. In order to avoid this packet loss, FAs are provided with a circular buffer referred to as the Forwarding Buffer. When a tunnelled packet arrives at the previous FA, it is decapsulated, delivered to the MN (if possible) and copied into a buffer. When the previous FA receives a binding update originating from a previous foreign agent notification, these buffered packets are re-tunnelled to the new FA and all packets arriving at the previous FA with destination the MN are immediately tunnelled to the new FA. In order to avoid duplicate packets, the MN includes the pair of source address and datagram identification of the most recent received packets in the registration request that is sent to the previous FA, who uses this pair to drop the buffered packets that have been received by the MH.

Pre-registration

The Pre-Registration handoff scheme ([21]) is based on an anticipated Layer 3 handoff by means of Layer 2 triggers. The L2 trigger contains the new FA's IP address and this allows the Mobile Node (MN) to communicate via the previous FA with the new FA while still being connected with the previous FA. The MN sends a registration request to the new FA via the previous FA (if the L2 handoff is not completed) and the new FA issues a regional registration to the Gateway FA (GFA). The latter sends a registration reply message to the new FA. Until the MN actually completes the L2 handoff to the new FA and establishes the new L2 link, the new FA can receive packets for which it does not have a link layer connection. These packets may be buffered in the new FA.

Post-registration

In the Post-Registration scheme ([21]), the registration occurs after the L2 handoff is complete. The L2 trigger initiates the set-up of a bi-directional edge tunnel (BET) between the previous FA and the new FA. When the previous FA receives the L2 Line Down (LD) trigger, it starts forwarding packets destined to the MN via the BET to the new FA. When the new FA receives an L2 Line Up (LU) trigger, it delivers the packets received from the previous FA to the MN. Eventually the MN performs a formal MIP registration.

3.3. A Generic Analytical Model for IP Mobility Protocols

We propose an analytical model to compute the packet loss and the delay experienced by a UDP stream (i.e. a constant bit rate packet stream) originating from a Corresponding Node (CN) with destination the MN, when the MN switches from one access point to another. For computational tractability reasons, all routers are modelled as simple M/M/1 queues. From a performance modelling point of view, the packets can be subdivided into different classes. The time intervals that determine to which class a packet belongs are obtained from the end-to-end delay a message experiences when travelling form one node in the network to another node. Since all routers are modelled as M/M/1 queues, the length of these time intervals is the sum of exponentially distributed random variables and constants, and therefore computable in a fairly straightforward way. While travelling to the MN, each packet follows a specific path of routers according to the class it belongs to. Again due to the M/M/1 assumption, this path is the sum of exponentially distributed random variables and constants, and therefore the end-to-end delay can be computed easily. The model also allows the computation of the buffer requirements and possible packet loss probabilities. The models for the handoff schemes discussed in the previous section are found in the following papers: Cellular IP [22], HAWAII [23], Smooth Handoff [24], Pre-Registration [25] and Post-Registration [26].

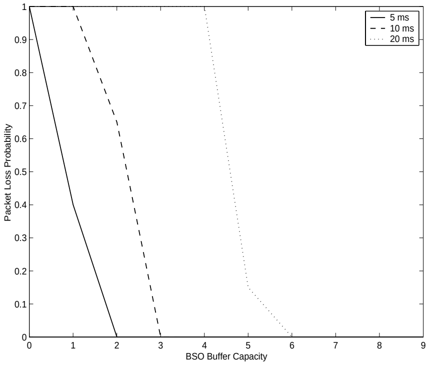

Figure 3: Packet loss probability in the Forwarding Buffer

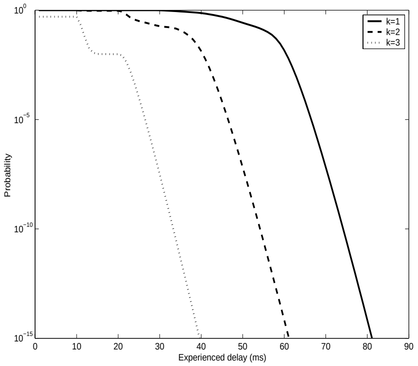

Figure 4: Delay experienced by packets due to forwarding

3.4. Performance Analysis of IP Mobility Protocols

In this Section we apply the generic analytical model for IP Mobility protocols described in the previous Section to HAWAII, Cellular IP, Smooth Handoff, Pre- and Post-Registration Handoff. A comparison of these mobility protocols can be found in [27].

3.4.1. HAWAII

Consider the system depicted in Figure 2, where each router (including the Base Stations) has the following characteristics: the service rate μ equals 10 packets per ms, the load is given by $\rho = 0.8$, and the propagation delay to each neighboring router is variable (5 ms, 10 ms and 20 ms). We consider a stream of packets arriving at Router R0 with a constant packet interarrival time of T=20 ms.
For this system, Figure 3 shows the packet loss probability as a function of the capacity of the forwarding buffer at the Old BS for the different propagation delays between neighboring routers (5 ms, 10 ms and 20 ms). Next, we compute the delay due to forwarding for this system. Figure 4 shows the probability that the *k*-th packet after handoff experiences a delay of at least *t* ms due to the forwarding scheme.

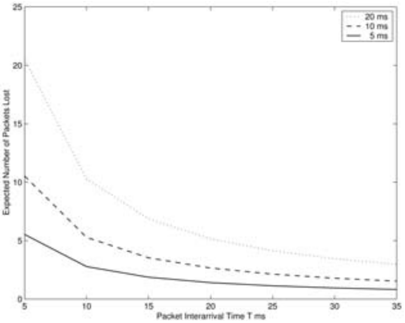

Figure 5: Hard handoff : Expected number of packets lost in BSO

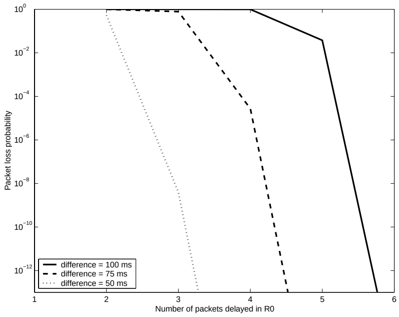

Figure 6: Packet loss probability as a function of number of delayed packets for different values of difference in length of new and old path

3.4.2. Cellular IP

First we evaluate the case of the Hard Handoff scheme. Consider the reference network as described in 2.3.1. We consider a stream of packets arriving at Router R0 with a constant packet interarrival time, taking values between 5 ms and 35 ms. Figure 5 shows the expected number of packets that are lost at the BSO due to the hard handoff, for variable packet interarrival times and different values of propagation delay. Clearly the expected number of lost packets increases when the propagation delay increases and the interarrival time decreases.
Next, for the Semi-soft Handoff scheme we compute the packet loss probability as a function of the number of packets that are delayed in the R0 buffer before the first packet is released, in order to cope with the different propagation delay between R0-BSO and R0-BSN. We assume this difference to be 50 ms, 75 ms and 100 ms. The three curves in Figure 6 show the loss probability of packets due to early arrival in BSN for different values of the buffer capacity in R0.

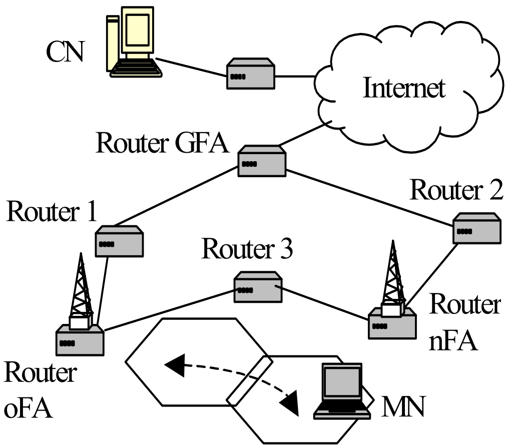

Figure 7: Reference Network for smooth handoff

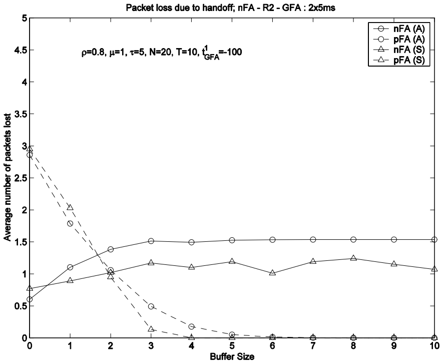

Figure 8: Packet loss as function of the buffer size

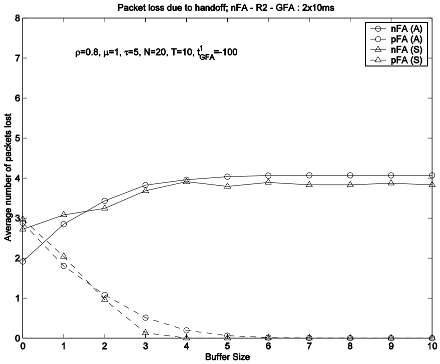

Figure 9: Packet loss as function of the buffer size

3.4.3. Optimized Smooth Handoff

Consider the network depicted in Figure 7 with the following system parameters. Each router is loaded up to 0.8, the propagation delay between routers is $\tau = 5$ ms, the average processing time of a packet in a router is 1ms.

In Figure 8 and Figure 9 the expected number of lost packets is shown as a function of the buffer size at the previous FA. The analytical results (A) are compared to simulation results (S). The expected packet loss due to buffer overflow is given by the dashed line, while the solid line represents the additional loss at the new FA, due to early arrival. Figure 8 shows the results for link delays equal to 5ms on every link, while for Figure 9, the 2 links on the nFA-GFA path are increased to 10ms each.

Obviously, the loss in the buffer at the previous FA diminishes when the buffer size is increased. The packets that are lost in case of very small buffer size do go through the buffer when this buffer size increases. But in the latter case they possibly contribute to the number of lost packets at the new FA, hence the rise of the solid curve for small buffer sizes. This is especially true for the case of 10ms link delay on the nFA-GFA path, because then the whole buffer is likely to arrive too early at the new FA (i.e. before the registration reply message from the GFA has arrived). In other words, in the case of long delay on the nFA-GFA path, if a packet is not dropped at the previous FA, it will most likely be lost at the new FA.

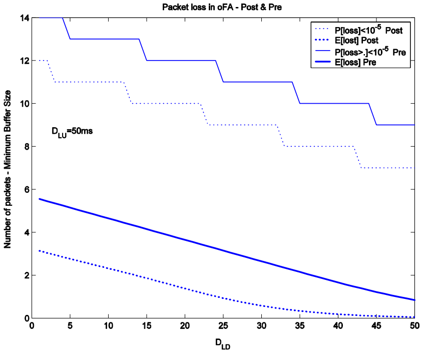

Figure 10: Packet loss in oFA

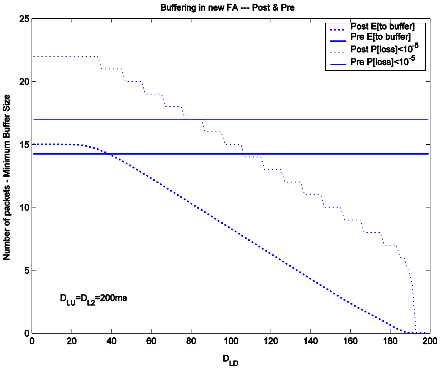

Figure 11: Buffers required at nFA

In order to avoid packet loss at the previous FA, the forwarding buffer need to be dimensioned such that it can store packets of the order of the product bit rate of the stream times delay (MN - new FA – previous FA). The loss at the new FA on the other hand depends on the difference between the distance (new FA – GFA) and (new FA – previous FA). If the latter is smaller than the former, then packets may get lost. A possible solution would be to provide the new FA with a buffer to store temporarily unauthorized traffic until the registration reply from the GFA arrives at the new FA.

Another solution consists of sending the binding update message from the new FA to the previous FA via the GFA in order to allow the registration reply message to arrive before the first forwarded packet. A similar solution has been applied in the Multiple Stream Forwarding scheme of the HAWAII.

3.4.4. Pre- and Post Registration

Consider the network depicted in Figure 7. Let τ_1 represent the propagation delay on the links connecting the Gateway and the oFA and also on the links connecting the Gateway and the nFA, and let τ_2 represent the propagation delay on the links connecting the oFA and the nFA. In order to compare the two schemes, we have to relate the respective triggers. We make the following assumptions. We let $t_0 = 0$ be the start of the handoff and we assume that $D_{L2} = D_{LU}$ (i.e. the handoff is completed when the nFA receives the LU trigger).

In a first example, we consider a system with the following parameters. A CN transmits 500 byte packets every T=10 ms in a network with μ=1, τ_1 =5 ms and τ_2 =3 ms. $D_{L2} = D_{LU} = 50ms$. Figure 10 shows the lost packets in the oFA for varying D_{LD}.

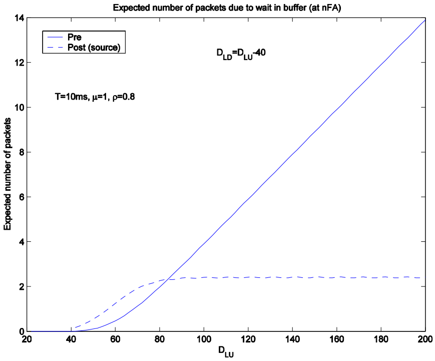

Figure 12: Buffers required in nFA

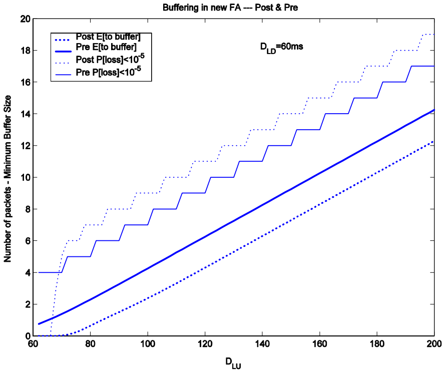

Figure 13: Buffers required in nFA

Next in Figure 11 we show the expected number of packets that need to be buffered in the nFA, in order to obtain zero packet loss (mean and 1-10^{-5} quantile). We let $D_{L2} = D_{LU} = 200ms$ and vary D_{LD}.

Whereas in the previous example $D_{L2} = D_{LU}$ was kept constant, in the next two examples we let $D_{L2} = D_{LU}$ vary. First in Figure 12, also D_{LD} varies, such that $D_{LU} - D_{LD} = 40ms$. In the last example we let $D_{LD} = 20, 40 \text{ and } 60$ ms and the corresponding results for the number of packets that need to be buffered in nFA are shown in Figure 13.

4. TCP in Wireless Networks

4.1. Introduction

The TCP transport protocol used in the Internet is a variable window protocol where losses are considered as congestion signals. Basically, TCP increases the window as acknowledgments are received and reduces the window when losses are detected. Therefore, in order TCP to perform well, it is needed that packet losses are mainly due to congestion. This may cause TCP to behave poorly in wireless networks, which are characterized by losses due to transmission errors and handoffs.

In this section we investigate the impact of wireless networks on TCP from two points of view. First, in section 3.2, we consider the TCP performance degradation introduced by a wireless media having transmission errors. We also study the capability of several mechanisms to reduce this performance degradation. Then, in section 3.3, we investigate the packet losses due to the handoffs using some of the micromobility protocols analized in section 3.2.

4.2. TCP in a Wireless Environment

In this section we point out the problems of TCP in wireless environments and review some proposed enhancements to TCP that aim at improving TCP performance for wireless and mobile nodes.

We analyze the TCP performance degradation introduced by a wireless media having transmission errors and the capability of the following mechanisms to correct it (i) improvements over the TCP Reno release (SACK [29, 33], New-Reno [30] and FACK [34]); (ii) improvements over the classical “tail dropping” mechanism used by routers (RED [28, 31] and ECN [35]). A TCP implementation based on the usage of ECN routers that would be able to distinguish among losses due to congestion and losses due to transmission errors is analyzed in [38].

4.2.1. Simulation Framework

The simulator used to obtain the results shown in this paper is an event driven simulator written in C++. In the following the assumptions made for the simulation of the TCP module and the network topology are described.

The TCP Module

We assume a greedy TCP module (which has always segments ready to send). Our TCP module passes the maximum number of segments allowed by the TCP window to the network driver. The segments are immediately given to the driver after the TCP module initialization, when an ack allows the transmission of more segments, or when a time-out expires.

For the network driver we have assumed an infinite queue (i.e. with no losses) which stores the segments received by the TCP module until they are sent into the transmission link. However, we have considered the buffer occupancy at the TCP receiver due to segment reordering. Therefore, in the simulations the TCP receiver always advertises a window equal to the free buffer space. We have also assumed that the TCP receiver sends an ack for each received segment. A realistic simulation is done of the slow timer used by TCP for the segments retransmission control (see [37]). In fact, the slow timer function is called each time interval equal to the granularity used by the TCP module as in a real implementation.

For sake of simplicity, we have not considered queuing delays but only the propagation delay in the return path of the acks. However, in order to avoid phase effects we have added a small random component in the return path delay of the acks.

The sources run the different TCP implementations with the following parameters:

1. Segments of 576 bytes (this is the default value used by TCP if no MSS is indicated at the connection setup).
2. Buffer space at the TCP receiver equal to 50 segments.

3. The timestamp option is used for the round trip measurements (see [36]).
4. We use the retransmission time-out mechanism described in [32].

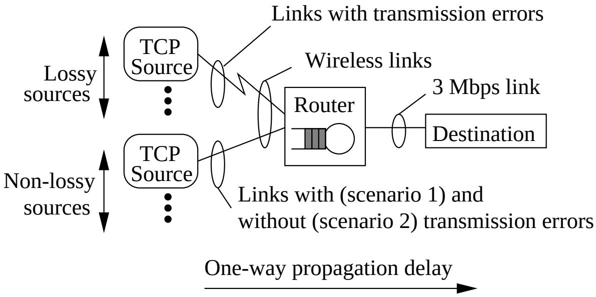
Links with transmission errors
TCP Source
Lossy sources
Wireless links
3 Mbps link
Router
Destination
TCP Source
Non-lossy sources
Links with (scenario 1) and without (scenario 2) transmission errors
One-way propagation delay

Fig. 14: Simulation scenarios.

Simulation Topology and Scenarios

Fig. 14 shows the network topology we have simulated in this paper. It consists of 20 TCP sources that compete for a congested link at the router output. The transmission links of the sources are 1 Mbps (we shall refer to these links as the wireless links), and the congested link at the router output is 3 Mbps.

We have considered two scenarios:

1. ***Lossy*** **scenario**: all the wireless links have the same transmission error rate,
2. ***half-lossy*** **scenario**: only half of the wireless links have transmission errors. We shall call *lossy* sources the sources having a transmission link with transmission errors and *non-lossy* otherwise.
1. In the simulation the errors have been introduced only in the forward direction, i.e. no acks are lost due to transmission errors. We have used a Bernoulli distribution in order to generate these transmission errors (i.e. each segment is lost due to transmission errors with probability *loss_p* and properly transmitted with.
3. probability 1 – *loss_p*). Several simulations have been carried out for each of these scenarios changing the behavior of the TCP sources and the router as follows:
4. **TCP sources**: (i) standard TCP-Reno [36], (ii) TCP with the New-Reno improvement [30], (iii) TCP with the SACK implementation [29] and (iv) TCP with the FACK implementation [34].
5. **Router**: (i) "tail dropping", (ii) with the RED discarding policy [31] and (iii) with the ECN-RED marking mechanism [35].

   Finally, simulations have been done varying the following parameters:
2. **Number of sources**: this has been set to (i) 10 sources (5 *lossy* and 5 *non-lossy* in scenario 2) and (ii) 40 sources (20 *lossy* and 20 *non-lossy* in scenario 2),
3. **end-to-end propagation delay**: this has been set to (i) 1 ms and (ii) 20 ms,
4. **TCP granularity**: this has been set to (i) 50 ms and (ii) 500 ms.

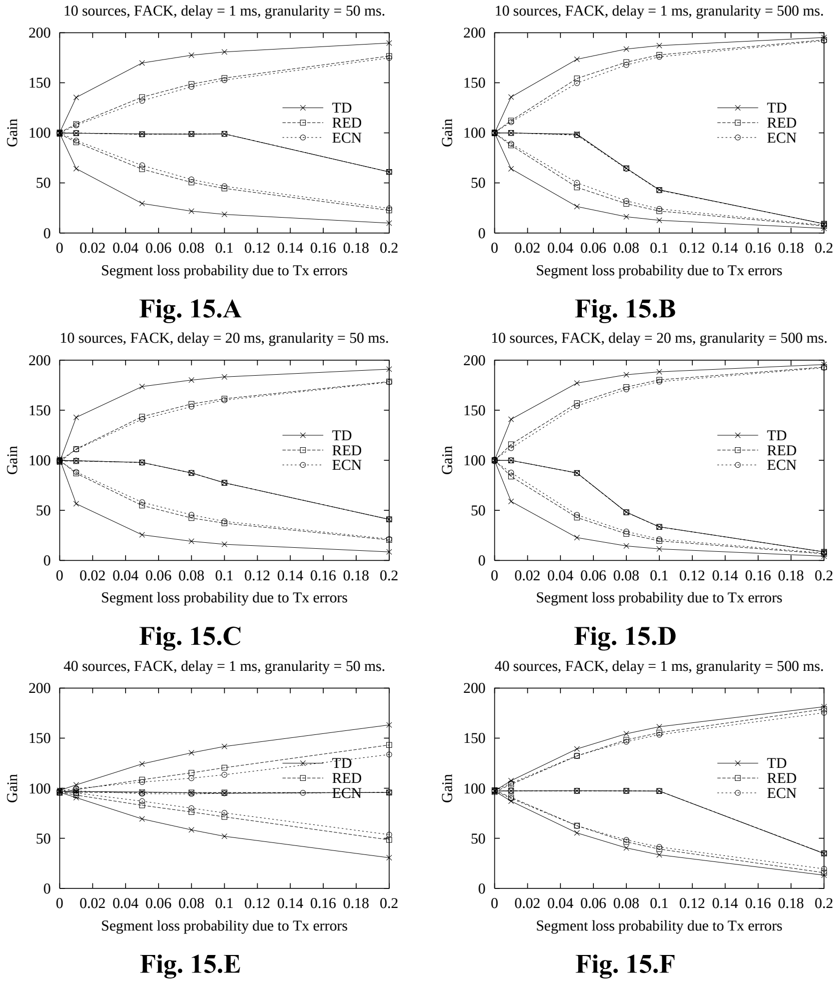

Fig. 15.A

Fig. 15.B

Fig. 15.C

Fig. 15.D

Fig. 15.E

Fig. 15.F

Fig. 15: Gain obtained in the lossy and half-lossy scenarios varying the number of sources, delay and granularity using a tail dropping (TD), RED and ECN router.

The buffer size of the router has been fixed to 150 segments in all cases. RED and ECN-RED have been implemented with the following parameters (see [31]): `LowTh` = 20 segments, `HighTh` = 60 segments, `wq` = 0.5, `maxp` = 0.02

4.2.2. Numerical Results

Each of the graphs of Fig. 15 and Fig. 16 shows the results obtained for the lossy and half-lossy scenarios described in section 0 with a different set of parameters. What we call *gain* in the graphs is given by the average goodput obtained for each group of sources having the same ratio of transmission errors to the fair goodput (the link rate divided by the number of sources) multiplied by 100. We compute the goodput of one source as the sequence number increment achieved during the simulation multiplied by the segment size in bits divided the simulation time. Remember from section 0 that in the lossy scenario all the sources have the same segment loss probability due to transmission errors and in the half-lossy scenario only half of the sources have transmission errors. This loss probability is given in the abscissa and we shall refer to it as *loss_p*. Therefore, only one point is depicted in the graph for each simulation with the lossy scenario, and two points corresponding to the group of sources having and not having transmission errors are depicted in the half-lossy scenario. Note that these points are easy to identify: (i) those in the middle of the graph correspond to the lossy scenario and (ii) those in the top and the bottom of the graphs respectively correspond to the lossy and non-lossy sources of the half-lossy scenario.

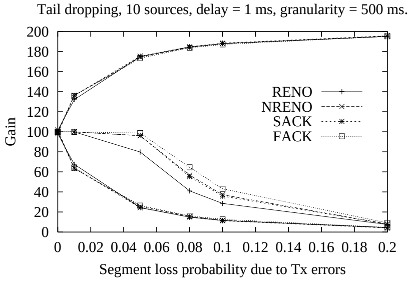

Fig. 16: Gain obtained in the lossy and half-lossy scenarios using different TCP implementations.

Note also that in Fig. 15 the curves obtained for each of the three types of routers (tail dropping, RED and ECN) have been superimposed for each scenario. The TCP implementation used for these graphs is FACK. Instead, in Fig. 15 the curves obtained for each of the TCP implementations (Reno, New-Reno, SACK and FACK) have been superimposed. In the following some general guidelines are derived from these graphs.

Unfairness: the first effect that becomes apparent from the graphs of Fig. 15 arises when comparing the results obtained with the lossy and half-lossy scenarios. In all cases the degradation of the gain obtained by the sources having losses due to transmission errors is higher when the congested link is to be shared with sources having no transmission losses. For example, Fig. 15.A shows that only for values of *loss_p* higher than 0.1 the gain starts to degrade in the lossy scenario. Instead, the gain of sources having transmission errors starts to degrade for small values of *loss_p* in the half-lossy scenario (e.g. using the “tail dropping” router this is only 20% when *loss_p* is equal to 0.1). This effect can be seen as an unfairness behavior of TCP when different transmission error rates occur among the sources, since the sources without transmission errors have the tendency to lock-out the sources with transmission errors.

RED and ECN advantage over tail dropping routers: Fig. 15 shows that the advantage of using RED and ECN arises in the half-lossy scenario. In this case, the unfairness effect described in the previous paragraph is reduced. For example, Fig. 15.A shows that when *loss_p* is equal to 0.1 the gain of sources having transmission errors in the half-lossy scenario is around 45% when RED and ECN is used, while it is only 20% when using tail-dropping. These graphs show also that this fairness benefit of using RED and ECN depends on the delay and the granularity. In fact it can be observed that the higher the granularity, the lower the benefit of using RED and ECN in terms of fairness. Furthermore, the lower is the RTT, the higher is the influence of the granularity. Note that there are other differences between the router algorithms analyzed that cannot be derived from Fig. 15. For example, using tail dropping the queue length has stronger oscillations than using RED. Therefore, the end-to-end transmission delays and their variance are reduced using RED and ECN-RED.

Granularity: the influence of this parameter is twofold since it determines the accuracy in the RTT measurement and the coarse of time-outs. Comparing Fig. 15.A and Fig. 15.B it can be seen that a higher value of the granularity not only reduces the fairness benefit of RED and ECN-RED as explained in the previous paragraph, but also increases the influence of *loss_p* on the gain.

Delay: comparing Fig. 15.A and Fig. 15.C it can be observed that the higher is the RTT, the higher is the influence of *loss_p* on the gain. However, Fig. 15.E shows that this effect is reduced when increasing the number of sources.

Number of sources: comparing the Fig. 15.A- Fig. 15.B respectively with Fig. 15.E-Fig. 15.F it can be observed that the higher is the number of sources, the lower is the reduction of the gain when *loss_p* increases. This is logical since the higher is the number of sources the lower is the average rate obtained for each of them, consequently, the higher is the ratio of discarded (or marked) to transmitted segments by the router in order to adjust the transmission rate governed by TCP. Therefore, the higher loss rate due to transmission errors is needed to interfere with segments discarded (or marked) by the router.

Reno, New-Reno, SACK and FACK: Fig. 16 depicts the gain obtained with each of these TCP implementations in the lossy and half-lossy scenarios using the same parameters than those of Fig. 15.B. The traces obtained using the other sets of parameters of the graphs of Fig. 15 are not shown because similar conclusions that those given in the following where obtained. The first conclusion that can be derived from Fig. 16 is that in the half-lossy scenario, nearly the same result is obtained regardless of the TCP implementation. Therefore, these TCP implementations are not effective to solve the unfairness problem described in the previous paragraphs. Furthermore, from Fig. 16 it can be observed that the influence of the granularity is even higher than the use of a refined TCP implementation. This can be explained since the most important benefit of better TCP implementations is reducing the number of time-outs, but the impact of time-outs is only predominant when the value of the granularity is much higher than the RTT. Finally, from Fig. 16 the following conclusions can be derived: (i) Reno is clearly outperformed by the other TCP implementations, (ii) New-Reno achieves the same performance as the basic SACK implementation (that one we refer to as SACK), (iii) using the more complex TCP implementation (FACK) does not represent a significant improvement over the much simpler New-Reno implementation.

4.3. TCP in a Wireless Network with Micromobility Support

As explained in section 3, several IP micro-mobility protocols have been proposed to enhance the performance of Mobile IP in an environment with frequent handoffs. In this section we make a detailed study of how some of these protocols, namely Cellular IP and Hawaii (see section 2.2.3), affect the behavior of TCP and their interaction with the MAC layer. We investigate the impact of handoffs on TCP by means of simulation traces that show the evolution of segments and acknowledgments during handoffs. A more detailed study can be found in [39].

4.3.1. Simulation Framework

All simulations were conducted using the *network simulator*, ns (see [40, 41]). Fig. 17 shows the topology used in the simulations. This topology consists of two BSs (BS1 and BS2) and a cross-over router. The cross-over router is the Gateway for Cellular IP and the Domain Root Router for HAWAII. The topology was chosen as simple as possible, in order to avoid undesirable complexity and keep the results comprehensible.

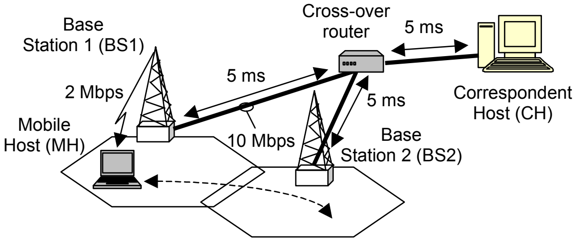

Fig. 17: Simulation testbed.

The Mobile Host moves between BS1 and BS2 such that handoffs are periodically produced. We will refer to the time between two consecutive handoffs as the *handoff period*. There is a cell overlapping of 1 second. BSs send router advertisements every 1 second. The MH uses these router advertisements as beacon signals to recognize the migration from one cell to another, and thus, initiate the handoffs.

In all the simulations a TCP download is simulated. The TCP-Reno implementation is used with a packet size of 1460 bytes and a maximum window size of 20 segments.

Several simulations have been carried out combining the following scenarios:

Radio links: Two kinds of radio links have been used (i) a shared media with an implementation of IEEE 802.11 that works like the 914 MHz Lucent WaveLAN DSSS radio interface, and (ii) an “ideal” wireless interface that consists of a fictitious non-shared media, that is, as if each sender were using a different channel at full link rate (with no collisions). We shall refer to these radio links as the *802.11 MAC* and *Ideal MAC* respectively.

The motivation of using the *Ideal MAC* was to eliminate the effect of a shared media access protocol as the 802.11. This allows to better observe the impact of the micro-mobility protocols. In both cases the bit rate of the radio link was set to 2 Mbps.

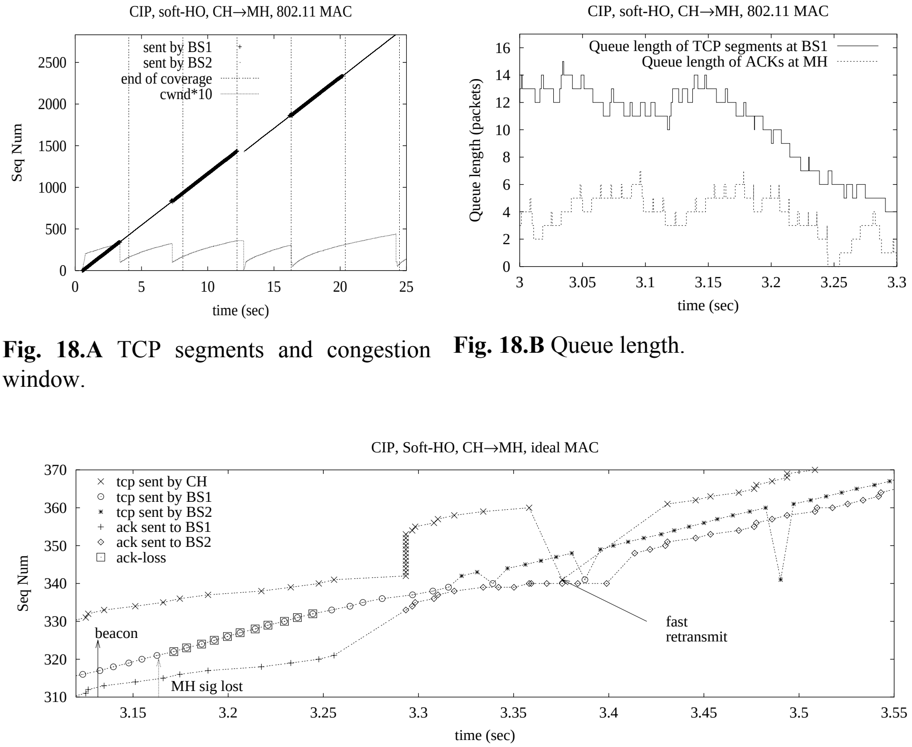

Fig. 18.A TCP segments and congestion window.

Fig. 18.B Queue length.

Fig. 18.C Zoom of a handoff.

Fig. 18: CIP with soft handoff traces using a 802.11-MAC

Micro-mobility protocols: we have tested the following micro-mobility protocols using the ns-2 simulator implementation of [41]: (i) Cellular IP with Hard Handoff and Soft Handoff and (ii) HAWAII with MSF and UNF Path Setup Schemes. In the HAWAII-MSF a forwarding buffer with capacity for 20 packets and a time out of 400 ms was used.

4.3.2. Numerical Results

All results discussed in this section have been obtained using the simulation topology described in the previous section. The section is organized as follows: First traces obtained using the 802.11-MAC are shown in order to discuss the impact of the radio link access mechanisms. Then, the dynamics of TCP using each of the micro-mobility protocols (CIP with hard and soft handoff, and HAWAII with MSF and UNF Path Setup Schemes) are presented using the Ideal-MAC.

Impact of the radio link access method

Fig. 5 shows traces obtained in a down-link transmission (from CH to MH) using Cellular IP with soft handoff and a 802.11-MAC radio access link.

Fig. 18.A shows the sequence number of each transmitted segment, the instants when the MH loses the coverage (end of coverage) with the old BS, and the evolution of the congestion window used by TCP (cwnd multiplied by 10 to see it better) at the CH. Fig. 18.B shows the queue length of TCP segments built up at the BS and the queue length of acks at the MH. Fig. 18.C is a zoom of Fig. 18.A. This figure shows: (i) Instants at which the TCP segments are transmitted by the CH (indicated as "tcp sent by CH" in the figure). (ii) Instants at which TCP segments arrive at the MH (indicated as "tcp sent by BS1/BS2"). These instants are marked differently according to the route taken to reach the MH (through BS1 or BS2). Note that these are also the instants when the acks are generated. Transmissions instants of lost acks are marked with a square. (iii) Instants at which acks arrive at the CH (indicated as "ack sent to BS1/BS2"). Again, these instants are marked differently according to the BS used to send them (BS1 or BS2). (iv) The transmission instant of the beacon from the new BS causing the handoff and the transmission instant of the route update message sent by the MH (indicated as *MH sig lost*).

The queue of acks built up at the MH shown in Fig. 18.B seems to be counterintuitive since the 10 Mbps links connecting the CH with the BS are much faster than the 2 Mbps radio link. Therefore, we would expect a queue of TCP segments at the BS, but not the queue of acks at the MH. The reason of this effect is that the acks sent by the MH tend to find the wireless medium occupied by the TCP segments, and their back-off makes that the number of acks per time unit that the MH is able to send into the shared media is lower than the number or TCP segments per time unit sent by the BS. The queue of acks is responsible of the long delay that occurs between the transmission of the ack by the MH and the reception at the CN, as shown in trace (c).

The beacon shown in Fig. 18.C is the first one received from the new BS (BS2). This beacon causes the MH to initiate the handoff. Therefore, the MH switches the radio connection from the old BS (BS1) to the new BS and sends the route update message through the new BS to the gateway router. As indicated in the figure, this route update message and the following 11 acks sent to the new BS are lost. These loses are caused by an address resolution failure motivated by the ack queue built up at the radio link driver of the MH. The IP module at the MH issues an ARP request to the new BS when the route update message is to be sent. The ARP packet is stored at the driver queue and the route update message is kept while waiting for the address resolution. Since the following acks sent by the MH are also addressed to the new BS, and the ARP module keeps only one packet waiting for the resolution of an address, each ack push out the previous packet waiting for the resolution of the new BS address. Only when the ARP query leaves the queue and the address is solved, the following acks are sent to the new BS. The first ack reaching the cross over router changes the routing cache to point to the new BS. During this time the TCP packets are still able to reach the MH through the old BS. These packets are not lost because the MH is able to simultaneously listen to both BSs.

Fig. 18.A shows that although no TCP segments are lost during this first handoff, the TCP sender reduces the congestion window. This is because some of the packets arriving from the old BS reach the MH later than packets reaching the MH through the new BS (as shown in trace (c)). These packets arrive out of order causing the MH to send duplicated acks that trigger the fast retransmit mechanism of TCP. In this case, the fast retransmit of TCP unnecessarily retransmits a packet and reduces the congestion window.

To avoid the described problems caused by the 802.11 MAC influencing the performance of the Micro-Mobility protocols, in the following evaluations we shall use an Ideal MAC.

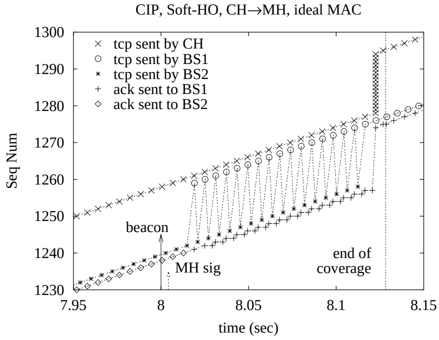

Fig. 19: CIP with soft handoff.

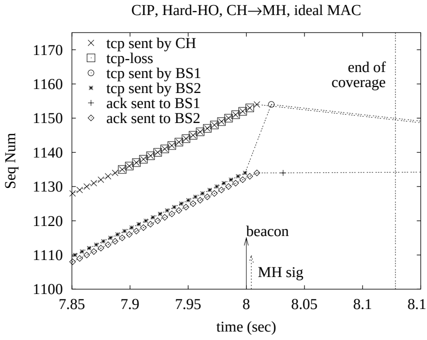

Fig. 20: CIP with hard handoff.

CIP Soft Handoff

Fig. 19 shows the trace obtained with CIP using Soft Handoff. When the MH receives a beacon signal from the new BS, a handoff is initiated while the connection with the old BS is maintained. The MH listens to both BSs during the overlapping of their cell coverage.

The trace shows the transmission instants of the data segments at the CH, and the transmission instants of the data segments at the old BS (BS2) and the new BS (BS1). When the new BS start transmitting TCP segments, the old BS remains transmitting the segments that are enqueued. Since the queue at the new BS is empty, the delay of the segments that go along the path to the new BS is smaller than the ones that go along the path to the old BS. As a result of this, every time a segment transmitted by the new BS arrives at the MH, the MH sends a duplicated ack since the segment is out of order. Only when the last expected packet is transmitted by the old BS, all segments arrived out of order are acknowledged at once and the TCP source sends a whole window of new segments.

CIP Hard Handoff

Fig. 20 shows a trace capturing a handoff obtained with CIP using Hard Handoff. The points depicted in this figure are analogous to those of Fig. 19 but now some TCP segments are lost as shown.

The hard handoff procedure has an important impact on the TCP dynamics. At each handoff, packets get lost when the MH switches the connection from the old BS (BS2) to the new BS (BS1). Packets waiting at the old BS when the connection is switched cannot reach the MH and are lost. This burst of lost TCP segments causes the TCP sender to wait for the retransmission time out and start with a slow start phase. This produces a goodput degradation.

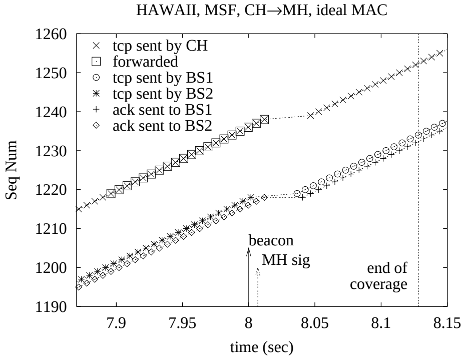

Fig. 21: Hawaii with MSF.

Hawaii with MSF/UNF

Fig. 21 shows the traces obtained using Hawaii with the MSF Path Setup Scheme. In this scenario the MH is able to maintain an ongoing connection with only one BS. Remember that this protocol tries to avoid losing the packets outstanding at the old BS when the connection is switched to the new BS. This is accomplished by forwarding these outstanding packets to the new BS. The trace shows how these forwarded packets effectively avoid TCP packets to be lost, thus solving the goodput degradation that occurs in CIP with hard handoffs.

Using Hawaii with the UNF Path Setup scheme the MH is able to listen to both the new and old BS. This scheme and CIP with soft handoffs have only small differences in the handoff procedures that are not relevant for their performance evaluation, showing similar behavior.

5. Open Research Issues

In this chapter a possible optimization of the Mobility support in IPv6 mechanism has been explored. Nevertheless, additional modifications to the general IPv6 behavior are to be considered in order to improve mobility efficiency. These modifications include different aspects of the specification such as the variation of frequency of Router Advertisement messages and mechanisms for improving micro-mobility support.

Hierarchical Mobile IP combined with a low latency handoff mechanism, such as Pre-Registration of Post-Registration seems to be promising solution to realize seamless handoffs in Mobile IP networks. However, further study is required to investigate how the required layer 2 triggers may be realized for different layer 2 access technologies (such as IEEE 802.11, Hiperlan, etc.). The timing of these triggers may have a major impact on the performance of the handoff and will determine the kind of protocol that is most appropriate, pre-, post-registration of a combination.

TCP has been tuned during many years for networks comprising wired links and stationary hosts. The evolution of TCP will have to take into account the introduction of wireless technologies, where the assumption of *congestion* as the primary cause of packet losses may not hold anymore. More research and practical experimentation with new wireless technologies need to be performed in order to standardize new TCP implementations that allow differentiating among congestion and other types of packet losses. In wireless networks it is also desirable that ongoing TCP connections do not degrade due to handoffs while the mobile moves within the access network. More research and experimentation is needed to better understand the main causes of packet loss during the handoffs, and the effectiveness to achieve seamless handoffs using the IP micro-mobility protocols that have been proposed in recent years.

References

1. Johnson, D. and C. Perkins, "Mobility Support in IPv6", Internet draft, Work in progress, July 2001.
2. Thomson, S. and T. Narten, "IPv6 Stateless Address Autoconfiguration", RFC 2462, December 1998.
3. Narten, T., Nordmark, E., Simpson, W., “Neighbor Discovery for IP Version 6 (IPv6)”, RFC 2461, December 1998
4. Gruber, J. and Strawczynski, L., “Subjective Effects of Variable Delay in Speech Clipping in Dynamically Managed Voice Systems,” IEEE Trans-actions on Communications, Vol. COM-33, No. 8, Aug. 1985.]
5. Hinden, R., O´Dell, M. and S. Deering, "An IPv6 Aggregatable Global Unicast Address Format", RFC 2374, July 1998.
6. Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 1998, 1998.
7. Schneier, B., "Applied cryptography", Wiley ISBN 0-471-12845-7,1996.
8. Eastlake, D., Crocker, S., Schiller, J., “Randomness Recommendations for security”, RFC 1750, December 1994
9. Narten, T., Draves, R., “Privacy Extensions for Stateless Address Autoconfiguration in IPv6”, RFC 3041, January 2001
10. Montenegro, G., Castelluccia, C., “SUCV Identifiers and addresses”, Internet draft, Work in progress, November 2001
11. Dommety, G., "Fast Handovers for Mobile IPv6", Internet draft, Work in progress, 2001.
12. Wasserman, M., “Recommendations for IPv6 in 3GPP Standards”, Internet Draft, Work in progress, April 2002
13. C. Perkins, ed., “IP Mobility Support”, IETF RFC 2002, October 1996.
14. E. Gustafsson, A. Jonsson, C. Perkins. “Mobile IP Regional Registration”, draft- ietf-mobileip-reg-tunnel-02.txt, March 2000.
15. Hesham Soliman et al., “Hierarchical Mobile IPv6 mobility management (HMIPv6)”, draft-ietf-mobileip-hmipv6-07.txt
16. A. Valko, “Cellular IP – a new approach of Internet host mobility”, ACM Computer Communication Reviews, January 1999
17. Ramjee, R., La Porta, T., Thuel, S., Varadhan, K., and Wang, S., HAWAII: a domain based approach for supporting mobility in wide–area wireless networks, Proceedings of International Conference on Network Protocols, ICNP’99.
18. A. Campbell, J. Gomez, C. Y Wan, S. Kim, Z. Turanyi, A. Valko, “Cellular IP”, IETF draft (draft-ietf-mobileip-cellularip-00.txt), January 2000.
19. A. Campbell, J. Gomez, S. Kim, A. Valko, C.-Y. Wan and Z. Turanyi, “Design, implementation and evaluation of Cellular IP”, IEEE Personal Communications, August 2000, pp.42-49

20. C. Perkins and K-Y. Wang. “Optimized smooth handoffs in Mobile IP”, Proceedings of IEEE Symposium on Computers and Communications, Egypt, July ’99.
21. K. El Malki and others, “Low Latency Handoffs in Mobile IPv4”, IETF draft-ietf-monileip-lowlatency-handoffs-v4-04.txt, 2002.
22. C. Blondia, O. Casals, P. De Cleyn and G. Willems, Performance evaluation of IP micro-mobility solutions, Proceedings Seventh IFIP/IEEE Workshop on Protocols for High-Speed Networks (PfHSN'2002), pp. 211 –226
23. C. Blondia, O. Casals, Ll. Cerdà and G. Willems, Performance analysis of a forwarding scheme for handoff in HAWAII, *Proceedings Networking 2002*, Pisa, Italy, Lecture Notes in Computer Science (LNCS) number 2345. Eds. Enrico Gregori, Marco Conti, Andrew T. Campbell, Guy Omidyar, Moshe Zukerman.
24. C. Blondia, O. Casals, N. Van den Wijngaert, G. Willems,Performance analysis of smooth handoff in Mobile IP, Proceedings MSWiM2002 Performance
25. C. Blondia, O. Casals, Ll. Cerdà, N. Van den Wijngaert, G. Willems, P. De Cleyn, Comparison of Low Latency Mobile IP schemes, to appear in proceedings WiOpt (Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks, March 2003
26. O. Casals, Ll. Cerdà, G. Willems, C. Blondia, N. Van den Wijngaert, Performance Evaluation of the Post-Registration Method, a Low Latency Handoff in MIPv4, to appear in Proceedings ICC 2003
27. P. Reinbold and O. Bonaventure, A Comparison of IP mobility protocols, Technical Report infonet-TR-13, 2001, www.infonet.fundp.ac.be
28. B. Braden et al. "Recommendations on Queue Management and Congestion Avoidance in the Internet". RFC 2309, April 1998.
29. K. Fall and S. Floyd. "Simulation-based Comparisons of Tahoe, Reno and SACK TCP". ACM Computer Communication Review, July 1996. ftp://ftp.ee.lbl.gov/papers/sacks.ps.Z
30. S. Floyd and T. Henderson. "The NewReno Modification to TCP's Fast Recovery Algorithm". RFC 2582, April 1999.
31. S. Floyd and V. Jacobson. "Random Early Detection Gateways for Congestion Avoidance". IEEE Transactions on Networking, August 1993.
32. V. Jacobson. "Congestion Avoidance and Control". ACM Computer Communication Review, 18(4), 314-329, August 1988. ftp://ftp.ee.lbl.gov/papers/congavoid.ps.Z
33. M. Mathis, J. Mahdavi, S. Floyd, and A. Romanow. "TCP Selective Acknowledgment Options". RFC 2018, October 1996.
34. M. Mathis and J. Mahdavi, S. Floyd. "Forward Acknowledgement: Refining TCP Congestion Control". ACM Computer Communication Review, 26(4), October 1996.
35. G. Montenegro, S. Dawkins, M. Kojo, V. Magret, N. Vaidya, “Long Thin Networks”, IETF RFC 2757, Jan. 2000
36. W.R. Stevens. "TCP/IP Illustrated, Volume 1: The Protocols". Addison-Wesley, 1994.
37. G.R. Wright and W.R. Stevens. *TCP/IP Illustrated, Volume 2, the implementation.* Ed: Addison-Wesley, 1995.
38. Ll. Cerdà, O. Casals. “Study of the TCP Unfairness in a Wireless Environment”. Proceedings of IEEE ICT 2001, Bucharest, Rumania, June 2001.
39. F. Vena, Ll. Cerdà, O. Casals. “Study of the TCP Dynamics over Wireless Networks with Micromobility Support Using the ns Simulator”. European Wireless 2002, Florence, Italy, February 2002.
40. The Network Simulator - ns-2, http://www.isi.edu/nsnam/ns
41. ns with micromobility support, http://comet.ctr.columbia.edu/micromobility