TCP Performance Enhancement in Wireless/Mobile Communications

S. Hadjiefthymiades, S. Papayiannis, L. Merakos
Communication Networks Laboratory, Department of Informatics and Telecommunications, University of Athens, Athens 15784, Greece
shadj@di.uoa.gr, stpap@di.uoa.gr, merakos@di.uoa.gr

Abstract

The operation of TCP in wireless - mobile environments is considered. After briefly presenting previous efforts to ameliorate TCP performance in the considered environments, we propose a new mechanism for tackling the problems caused by handovers. Our mechanism is based on stochastic datagram relocation. Traffic destined to the mobile terminal is tunneled to adjacent cells according to the output of a path prediction algorithm. To reduce the associated overhead, only percentages of inbound traffic are copied to the cell's neighborhood on the basis of estimated probabilities. Through simulations, we have measured the effects that stochastic datagram relocation has on TCP dynamics.

1. Introduction

The operation of TCP in wireless/mobile communications has been an important research issue in recent years, owing to the impressive growth experienced in that area of modern telecommunications during the past decade. Significant contributions, such as the one presented in [1], indicate that the unmodified, standardized operation of TCP is not well aligned with the peculiarities of cellular environments. Terminal movement across cell boundaries, leading to handover, is misinterpreted by common TCP implementations as sign of congestion within the fixed network. To handle such congestion, TCP unnecessarily slows down transmission by reducing window sizes, and performing retransmissions, if relevant need arises.

A number of efforts have been reported for the resolution of the above-mentioned problems. Notable contributions are the Snoop protocol [2, 3], the Indirect-TCP (I-TCP) scheme [4] and METP (Mobile End Transport Protocol) [5]. The Snoop protocol tries to conceal potential problems caused by the peculiarities of the wireless interface. Snoop involves the installation of a specialized module (the Snoop Agent) in the forefront of the fixed network namely the base station (BS or Access Point). The agent maintains a local cache memory, monitors the exchange of TCP packets and acknowledgments, and performs local retransmission as needed. The I-TCP scheme belongs to the so-called "split-connection" solutions. The end-to-end TCP connection between the mobile and the fixed node is split into two separate connections: one between the fixed node and the BS and the other between the BS and the mobile node. METP is also member of the "split-connection" family of schemes. In METP, TCP is replaced in the wireless part of the connection by a much simpler protocol with reduced headers.

The Snoop scheme has been used as a basis for the work presented in this paper. Snoop does not cover the cases where the mobile terminal (MT) gets involved in handovers. In those cases, the Snoop cache accumulated in the BS remains unused since the MT's point of attachment to the fixed infrastructure changes. In the new BS, no TCP datagrams pertaining to the network dialogs disrupted by the handover were ever cached. Hence, the problems that Snoop tries to resolve are seriously aggravated.

To address the problem described in the previous paragraph we suggest an alternative architecture. Specifically, we adopt the use of a path prediction algorithm and the proactive formulation of datagram caches in adjacent BSs. This scheme is quite similar to the Daedalus project approach, reported in [6], where traffic destined to the mobile terminal is multicast (by the Home Agent entity of Mobile IP) to all the BSs being adjacent to the one currently used.

To render our algorithm as efficient as possible (bandwidth waste is one of our considerations), we suggest the stochastic relocation of datagrams according to the probabilities assigned to adjacent BSs by the path prediction algorithm. A Path Prediction algorithm based on the AI scheme of Learning Automata has been proposed in [7]. In this paper we adopt the simulation results (prediction efficiency) reported in the above-mentioned paper. The algorithm discussed in the current paper could fit into any path prediction algorithm provided that the latter assigns probabilities to the entire neighborhood of the present cell (i.e., does not simply identify the most probable cell but provides a vector of probability values).

We have simulated the suggested architecture to evaluate the effects that datagram stochastic relocation has on TCP dynamics. We assume that BSs use a simple caching scheme rather than the complex Snoop algorithm itself. The simulation was performed using the MIL3 Opnet Modeler [15]. The obtained simulation results show considerable benefits for TCP connections.

The rest of the paper is structured as follows. In Section 2 we give an overview of prior work in the area of TCP enhancements for wireless/mobile communications. In Section 3, we discuss the details on the proposed scheme for wireless environments. In Section 4, we give a brief synopsis of the path prediction algorithm assumed for our study. In Section 5, we present the results of the simulation of the suggested relocation scheme. We conclude the paper in Section 6, where we outline directions for future work in the area.

2. Related Prior Work

Recently, a number of schemes have been proposed to cope with the transport protocol impairments over wireless links of high BER channel and many lost packets. Such schemes are classified into two main categories:

- the end-to-end solutions, and
- the split-connection approach.

The end-to-end approach involves the use of selective acknowledgments (SACK) in TCP dialogs informing the sender for the correct delivery of up to three non-contiguous packets [8]. Another similar technique involves the use of an explicit loss notification (ELN) distinguishing between a congestion condition within the network and actual packet losses. On the other end, the split-connection approach involves the establishment of two separate connections between the communicating hosts (i.e., the fixed and the wireless node) and the base station. Among the protocols of this category, the I-TCP (Indirect TCP) and the METP (Mobile End Transport Protocol) are described below.

I–TCP involves the splitting of the normal TCP connection into two parts, one between the base station and the fixed host (the wired interface), and the other between the base station and the mobile host (the wireless interface). Separate plain-vanilla TCP connections are established in each interface and handled individually. The advantage of splitting the original connection into two separate flows is the decoupling of congestion control that allows better performance in both interfaces. The main drawback of this approach is that the semantics of end-to-end acknowledgements are violated in the sense that an acknowledgement may have reached the sender even before the packet has been delivered to the destination host. Another drawback is the overhead that I-TCP incurs since packets are being processed twice. The handover procedure is also an issue that deserves attention, since the base station maintains state information for each TCP connection. Such information needs to be transferred to the new base station in the event of a handover, a complicated and time-consuming operation [4]. An improved version of I-TCP permits the mobile node to adapt to the peculiarities of the wireless link, such as smaller MTUs, fading errors and lower bandwidth [9].

The Mobile End Transport Protocol (METP) is a split-connection protocol that replaces the TCP/IP protocol stack over the wireless interface with a simpler protocol with smaller headers [5]. In order to set up a communication with a fixed host, the routing and flow control functionality of the mobile host is undertaken by the BS (this is further facilitated by the one-hop distance between the BS and the mobile node). An interesting feature of METP is that it exploits a link layer ACK and retransmission mechanism to deal with the errors occurring in the wireless link. From a certain point of view, the mobile host behaves as an application module forwarding requests to the BS acting as a transport layer. The BS maintains a sending and a receiving buffer along with state information per connection. It delivers ACKs corresponding to TCP packets back to the source host and, then, a separate process undertakes the transfer of buffered packets to the mobile host. Since METP enhances the BS with an acknowledgement mechanism, the only congestion control that takes places concerns the wired TCP connection. In the event of a handover, the new BS must be supplied with the state information for pending TCP connections along with sending and receiving buffers.

The Snoop protocol does not belong to either of the two categories discussed above. The main objective of Snoop is to leave existing implementations in the mobile and fixed nodes unmodified. The Snoop agent preserves a cache for the unacknowledged packets transmitted in either direction and performs local retransmissions according to the timeouts and the policy of unacknowledged packets. The BS monitors the packets exchanged through the wireless and the wired connection, while there is no need of extending the BS with transport layer functionality. Each packet arriving at the BS is cached and routed to the mobile host. Snoop keeps track of the last packet sequence number that has been acknowledged by the mobile. According to the sequence number of the incoming packet, the BS forwards the packet to the mobile host, or drops the packet, and sends an ACK to the fixed host confirming that this packet has already been delivered. In addition, Snoop monitors and processes the acknowledgements received by the mobile host and performs the required operations. Upon reception of a new ACK, it flushes the cached packets that have just been acknowledged, propagates the ACK to the fixed host and updates internal counters. When a duplicated ACK arrives, the agent has a different option to execute depending on whether the indicated packet is cached or not. Hence, a retransmission may take place to the mobile host, or a forwarding of the duplicated ACK to the originating fixed host, or even the ACK may be discarded. Whenever a handover is performed, the new BS starts to collect information about the pending connection similarly to the previous approaches. During handover execution, a number of packets and acks are missed.

3. System Architecture – Algorithm Description

Figure 1 illustrates the wireless network assumed in this study. The environment consists of a number of base stations interconnected by means of a fixed LAN infrastructure, as well as mobile and fixed terminals. Each BS comprises a radio transceiver and, possibly, a support workstation (the latter may handle all of the signaling needed for the roaming of the mobile terminal). BSs maintain information regarding the mobile terminals that are currently under their control. We also assume that each BS maintains a Datagram Relocation Coordinator (DRC) agent per controlled mobile terminal.

In the event of a handover, the involved BSs (of the current and the target cells) need to consult their information bases, and collectively undertake specific actions, such as:

- reservation of bandwidth in the new path (e.g., in a QoS-aware network),
- release of resources in the old path,
- diversion of connections, and
- ARP updates (address configurations, etc.).

In the majority of wireless architectures, user profiles are stored in specialized nodes within the user's home sub-network (the part of the network the user administratively belongs to). When the mobile terminal migrates to a sub-network different from its home, the user profile database (home registry) is queried, and the relevant information is forwarded to the visited network by means of specialized inter-network signaling.

The home registry of a mobile terminal may also incorporate a path prediction algorithm (Figure 2) similar to the one presented in section 4. Such an algorithm predicts, with hopefully adequate precision, to which cells the terminal is more likely to be handed-over if it keeps on roaming. The invocation of the path prediction algorithm can be performed at some time after the entrance of the mobile terminal in the current cell (it is assumed that the mobile terminal will spent a certain minimum amount of time within the current cell). From that point, the current BS stochastically relocates datagrams to the BSs indicated by the path prediction algorithm. The mobile terminal is likely to attach to those BSs (also referred to as Target BSs) in the near future.

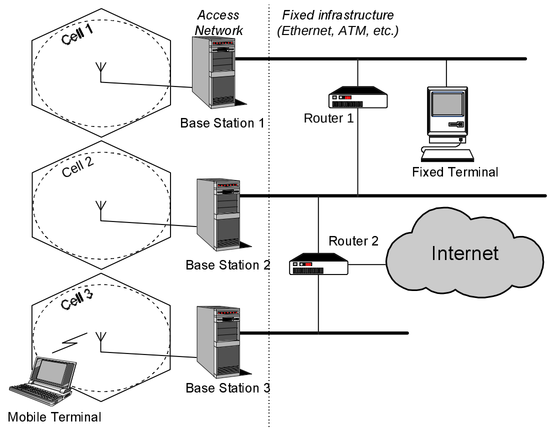


Figure 1. Network architecture

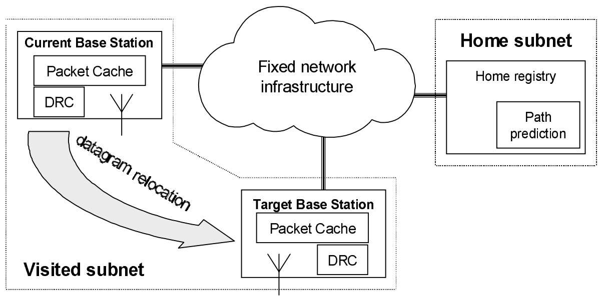


Figure 2. Packet Relocation Technique

In the current BS, when an inbound datagram[1] is processed (i.e., forwarded to the wireless interface), this datagram is also passed to the DRC for stochastic relocation (by means of IP tunneling) to adjacent nodes. The DRC has complete knowledge of the probabilities assigned by the path prediction algorithm to the adjacent BSs. For each incoming datagram and adjacent cell, the packet is independently processed and tunneled to the considered cell with probability equal to the assigned relocation percentage for that cell. There, the DCR feeds the incoming datagram to the local datagram cache. Caches are maintained per MT. DCR caches are based on a simple FIFO queue discipline. When inbound traffic exceeds the pre-configured maximum length of the queue, the oldest packets are dropped first (head-drop policy).

The sequence of actions undertaken by various network entities as well as the inter-network signaling

[1] In this paper we assume that no fragmentation is performed in the IP layer; hence there is a one-to-one correspondence between IP datagrams and TCP packets.

required for their completion are depicted in the Message Sequence Chart of Figure 3.

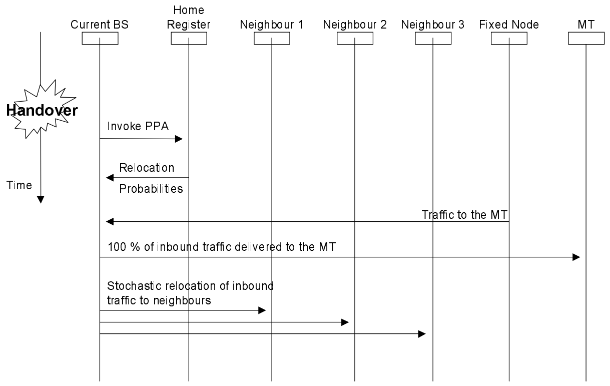


Figure 3. Message Sequence Chart for Algorithm Execution

The logic behind the operation of the DRC is simple. This entity

- invokes the path prediction algorithm in the home registry of the roaming terminal,
- stochastically duplicates traffic to adjacent BSs,
- receives copied traffic in adjacent BSs, and
- manages the queues deployed within those BSs.

For simplicity, we assume that the DRC has no knowledge of TCP's internal mechanisms for congestion avoidance neither inspects exchanged ACKs. A possible extension to the DRC functionality could be the inclusion of Snoop logic; this extension is expected to improve the performance exhibited by the system.

Stochastic Datagram Relocation

In this section, we argue on the need for a datagram relocation scheme which involves transmitting 100% of inbound traffic to the most likely target BS (Class A candidate) indicated by the path prediction algorithm. Furthermore, to deal with the cases of path prediction misses, the scheme involves the transmission of a high percentage of inbound traffic to some of the other neighboring cells (also referred to as Class B candidates). Class B candidates could be the second and third best predictions of the prediction algorithm. An even lower percentage of inbound traffic is relocated to the remaining neighboring cells (termed Class C candidates). The relocation percentages for Class B and Class C are design parameters and are assigned after performance assessment (see Section 5).

This approach is quite similar to the Shadow Cluster [10] technique proposed for bandwidth reservation in wireless ATM LANs. In the Shadow Cluster scheme, bandwidth is provisionally reserved in the cells adjacent to the one currently used by the roaming mobile terminal. Relocating the 100% of inbound traffic to all the adjacent cells is not a sound strategy, similarly to the bandwidth problem studied in [10], as it leads to increased overhead and non-optimum use of limited resources in the involved network links.

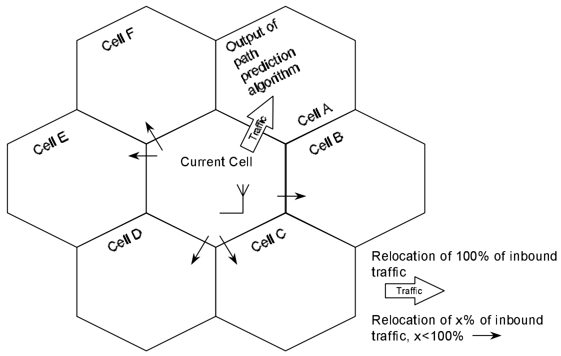


Figure 4. Datagram Relocation Strategies

A very important issue in the suggested scheme is the effect that stochastic relocation has on TCP dynamics. This issue is addressed in Section 5 where we present the results of our simulations.

4. Path Prediction Algorithm

The use of a path prediction algorithm in a wireless network architecture allows the efficient use of limited network resources such as bandwidth. A number of path prediction algorithms have been proposed in the networking literature. Notable examples of such work are the algorithm proposed by Liu et al. [11] and the Liu-Maguire algorithm [12]. The former example uses pattern matching techniques and Extended, Self Learning, Kalman filters to estimate the future location of mobile terminals. The Liu-Maguire algorithm is based on Mobile Motion Prediction (MMP) scheme for the prediction of the future location of a roaming user according to his movement history patterns. MMP is "based on the fact that everyone has some degree of regularity in his/her movement, that is, the movement of people consists of random movement and regular movement and the majority of mobile users has some regular daily (hourly, weekly, ...) movement patterns and follow these patterns more or less every day ...".

In our suggested architecture, we assume the use of the path prediction algorithm presented in [7]. Such algorithm is based on a Learning Automaton [13], a well-established artificial intelligence technique for machine learning. Learning automata are finite state adaptive systems that interact continuously in an iterative fashion with a general environment. Through a probabilistic, trial-and-error response process, they learn to choose or adapt to the behavior that generates the best response. In the first step of the learning process, input is provided to the automaton from the environment. This input triggers one of a finite number of candidate responses from the automaton. The environment receives and evaluates the response and then provides feedback to the automaton. Such feedback is used by the automaton to alter its stimulus-response mapping structure to improve its behavior.

Generally, the operation of the learning automaton is based on a state transition matrix, which contains the one-step transition probabilities P_{ij} from the current state i to the next state j. For the updating of the state transition matrix after environment feedback, the Linear Reward-Penalty (L_{R-P}) scheme has been adopted. Upon invocation, the automaton selects, as candidate future state, the state with the highest probability. After consecutive interactions with the environment, some transitions will have probabilities close to 1 while others will have near-zero values (convergence). The adopted path prediction algorithm capitalizes on the time - space regularity of the nomadic user's movement. To accomplish that, the entries of the transition matrix that supports the operation of the automaton have the layout shown in Figure 5.

PreviousCell_ID	CurrentCell_ID	FutureCell_ID	TimeSlot	ProbabilityValue	TimeStamp

Figure 5. Layout of state transition matrix entries

PreviousCell_ID, CurrentCell_ID and FutureCell_ID are the identifiers of the previous, current and possible future cell of the mobile terminal, respectively. ProbabilityValue denotes the likelihood that the mobile terminal (previously located at PreviousCell_ID) migrates, within a specific time slot (denoted by TimeSlot), from CurrentCell_ID to FutureCell_ID. Hence, each state in the suggested algorithm is a triplet consisting of PreviousCell_ID, CurrentCell_ID and TimeSlot. Time is assumed divided in slots of 15 minutes. Whenever a prediction request arrives at the home registry of the mobile terminal, the entries pertaining to the chronologically closest time slot are taken into account provided that the distance to them does not exceed the 1 hour limit (i.e., 4 time slots). If no appropriate state transitions were found then new entries are introduced in the state transition matrix and a random one is chosen as the algorithm's output. The random selection is only influenced by the cell previously visited by the terminal. In the absence of other information, a linear movement of the terminal is assumed. Hence, the automaton, instead of making a completely random selection, points to the symmetrical cell to the one previously visited as the candidate cell.

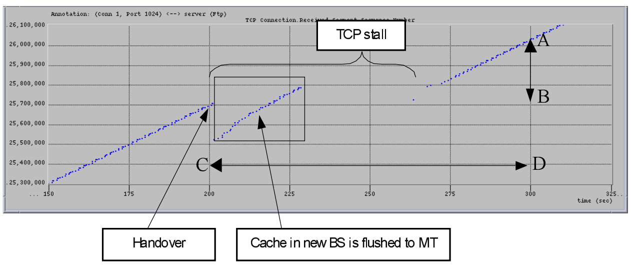
If the automaton decision is correct (positive feedback) then the matrix entry that contributed to this decision is rewarded while the probability values of the remaining entries are reduced (penalized). The reward/penalty procedure applies to all the entries of the matrix that refer to the same tuple of time slot, previous and current cell.

Simulations of the previously discussed algorithm have shown that it achieves a hit rate of 48% (for a hexagonal cell arrangement). The datagram relocation technique could also benefit from the second best guesses of the algorithm (Class B candidates), which seem to successfully match the real route of the mobile terminal with a probability of 0.2 (20%).

5. Simulation Results

A series of simulation experiments were realized in order to quantify the effects that stochastic relocation of datagrams has on TCP dynamics. Specifically, we measured the behavior exhibited by a single FTP flow that has been interrupted by a handover occurrence. The experiments involved the FTP transfer of a 2 MB file. This flow was handed over, in time t=200 sec (from simulation start), to an adjacent, cache-enabled, base station. The adopted queue size in the adjacent base stations of the architecture was 30 datagrams. Since our main focus in this paper is on Class B and C candidates (candidate BSs receiving less than 100% of inbound traffic), we performed experiments for relocation percentages of 80%, 70%, 60%, 50%, 40% and 30%. For each of these scenarios, through the simulation environment we have monitored the following metrics:

- increase of packet sequence numbers, and,
- time of TCP stall



$$SeqNumRate = \frac{AB}{CD}$$

Figure 6. Metrics monitored for stochastic packet relocation

To assess the increase of packet sequence numbers during the handover period we calculate the slope in the sequence number (SeqNum)/time plot over a 100sec period right after the occurrence of the handover. We call this metric SeqNumRate. A fast recovery scenario is characterized by an increased SeqNumRate value. As time of TCP stall we consider the time taken by TCP to reach, through the new path, the same sequence number as the one signaled in the handover interrupted communication. Figure 6 shows the discussed metrics over the SeqNum/time plot. In Figure 7 and Figure 8 we present the simulation statistics with regard to the aforementioned metrics. Both plots are enriched with exponential fits to show the trend line of the underlying data set.

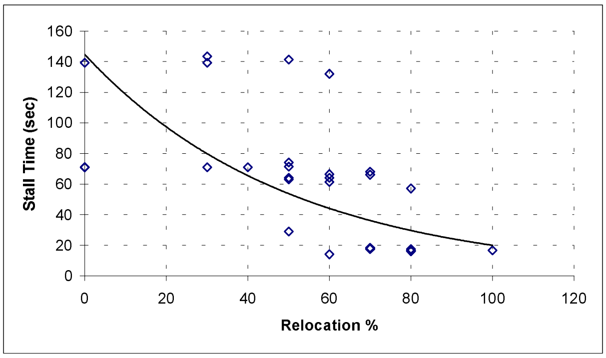


Figure 7. TCP Stall Time Vs Relocation Percentage

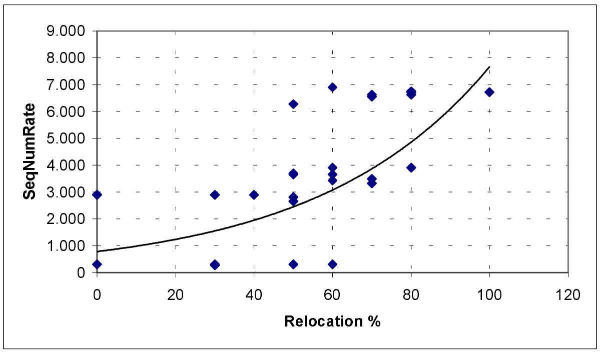


Figure 8. SeqNumRate Vs Relocation Percentage

As shown in Figure 7 and Figure 8, a relocation percentage of 70% - 80% demonstrates a very good behavior in terms of Stall Time and sequence number increase. For the said percentages, performance is comparable to the 100% relocation case. Hence, a relocation percentage in the range 70% - 80% is proposed for the Class B candidates that the prediction algorithm provides. On the extreme other end, relocating 30% of inbound traffic achieves a very poor performance. In certain trials such performance was similar to the 0% relocation case. For Class C candidates we propose the adoption of a relocation percentage in the range 40% - 50%.

Apart from the single-flow scenarios we have also monitored the behavior exhibited in multi-flow cases. Specifically, we have conducted a series of experiments involving three (3) FTP flows. Results of the stochastic relocation scheme were quite alike. Differences between the 50% and 80% scenarios are easily distinguished in Figures 9 and 10 respectively.

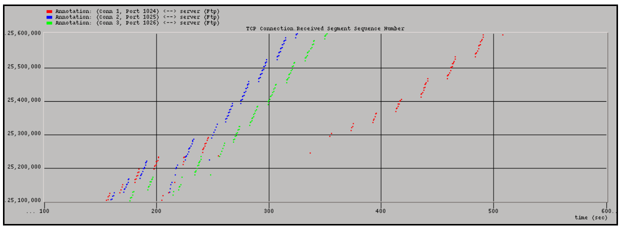


Figure 9. Multi-flow scenario/50% relocation percentage

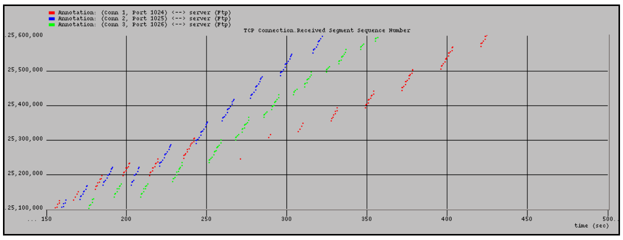


Figure 10. Multi-flow scenario/80% relocation percentage

6. Conclusions

In this paper, we have proposed an architecture for overcoming the problems associated with the operation of TCP in wireless networking environments. We suggested the use of datagram/packet caches in base stations. Inbound traffic is stochastically relocated by the current base station to all its adjacent base stations and cached there. Stochastic relocation is based on the outcome of a path prediction algorithm assigning probabilities to all the neighbors of the current base station. To render the operation of the scheme as efficient as possible and avoid excess bandwidth consumption in the access network, we relocate a high percentage of inbound traffic to more likely neighbors. Less likely neighbors receive a lower percentage of traffic. We have shown, through simulations, that a relocation percentage in the range 70-80% allows TCP to rapidly recover after the occurrence of a handover and a path prediction miss.

The integration of the proposed scheme with a mechanism such as Snoop seems promising as it could achieve even better results than the ones presented here and is currently under study.

7. References

[1] R.Caceres, and L.Iftode, "Improving the Performance of Reliable Transport Protocols in Mobile Computing Environments", IEEE JSAC, Vol. 13, No. 5, June 1995.

[2] H.Balakrishnan, S.Seshan, E.Amir and R.Katz, "Improving TCP/IP Performance over Wireless Networks", Proceedings of the 1st ACM International

Conference on Mobile Computing and Networking, (Mobicom), November 1995.
[3] E.Amir, H.Balakrishnan, S.Seshan, and R.Katz, "Efficient TCP Over Networks with Wireless Links", Proceedings of the 5h Workshop on Hot Topics in Operating Systems (HOTOS-V), Orcas Island, WA, May 1995.
[4] A.Bakre, and R.Badrinath, "Handoff and System Support for Indirect TCP/IP", Proceedings of Second Usenix Symp. on Mobile and Location-Independent Computing, April 1995.
[5] K.Wang, and S.Tripathi, "Mobile-End Transport Protocol: An Alternative to TCP/IP over Wireless Links", Proceedings IEEE Infocom 1998.
[6] S.Seshan, H.Balakrishnan, and R.Katz, "Handoffs in Cellular Wireless Networks: The Daedalus Implementation and Experience", Kluwer International Journal on Wireless Communication Systems, 1996.
[7] S.Hadjiefthymiades, and L.Merakos, "ESW4: Enhanced Scheme for WWW computing in Wireless communication environments", ACM SIGCOMM Computer Communication Review, Vol.29, No.5, 1999.
[8] M.Mathis, J. Mahdavi, S.Floyd, and A.Romanow "TCP Selective Acknowledgements Options", RFC-2018, 1996.
[9] R. Yavatkar, N. Bhagawat, "Improving End to End Performance of TCP over Mobile Networks", Proceedings of "Mobile '94", Workshop on Mobile Computing Systems and Applications, Santa Cruz, CA, December 1994.
[10] D. Levine, I. Akyildiz, and M. Naghshineh, "A Resource Estimation and Call Admission Algorithm for Wireless Multimedia Networks Using the Shadow Cluster Concept," IEEE/ACM Transactions on Networking, Vol.5, No.1, February 1997.
[11] T. Liu, P. Bahl, and I. Chlamtac, "Mobility Modeling, Location Tracking, and Trajectory Prediction in Wireless ATM Networks," IEEE Journal on Selected Areas in Communications, Vol.16, No 6, August 1998.
[12] G.Y. Liu, and G.Q. Maguire, Jr., "A Predictive Mobility Management Algorithm for Wireless Mobile Computing and Communications," Proceedings of ICUPC '95, Tokyo, Japan, November, 1995.
[13] K. Narendra, and M. A. L. Thathachar, "Learning Automata: An Introduction," Prentice-Hall, 1989.
[14] W.R.Stevens, "TCP/IP Illustrated, Volume 1", Addison Wesley, 1997.
[15] "OPNET Modeler - Modeling Concepts", MIL3 Inc., 2000.