Improving TCP Performance During the Intra LTE Handover

D. Pacifico, M. Pacifico, C. Fischione, H. Hjalrmasson, K. H. Johansson
School of Electrical Engineering
KTH Royal Institute of Technology, Stockholm, Sweden
Email: {davidep|pacifico|carlofi|hjalmars|kallej}@ee.kth.se

Abstract—Ensuring a seamless connection when users are moving across radio cells is essential to guarantee a high communication quality. In this paper, performance of TCP during the handover in a Long Term Evolution (LTE) network is investigated. Specifically, mobile users with high bit rates TCP services are considered, and the impacts of the intra LTE handover over their perceived throughput are studied. Due to the mobility of the users across radio cells, the high bandwidth required, and possible network congestions, it is shown that the handover may cause sudden degradation of the quality of the communication if the process is not correctly controlled. To alleviate these problems, three solutions are proposed: fast path switch, handover prediction, and active queue management. The first two solutions avoids excessive delay in the packet delivery during the handover, whereas the second solution acts at the transport network with an active queue management. Simulation results, obtained by an extension of the *ns-2* simulator, show that the proposed solutions present advantages, and that the handover prediction used with the active queue management increases TCP performance significantly.

I. INTRODUCTION

The recent increase of mobile data usage and the emergence of new applications, such as multimedia online gaming, mobile TV, Web 2.0, and streaming contents, have motivated the development of LTE, the Long Term Evolution of the UMTS terrestrial radio access network [1]. LTE supports a high throughput with low latency by an IP based transport network. TCP, the Transmission Control Protocol, is expected to play a significant role in LTE, given its high popularity on IP networks.

To achieve a seamless mobility across radio cells, LTE implements a hard handover algorithm, which moves a mobile user from one base station serving a radio cell to another one. However, considering the high bandwidth required by the users, the handover may cause a sudden degradation of the throughput of the TCP connections. Such a problem can be further exacerbated if the wired transport network is congested. Since the handovers may occur frequently for LTE networks, we conclude that the quality of service perceived by the users that are using TCP connections may be low.

There is a large literature on research for improving TCP over wireless links [2]. The main approaches proposed can

be grouped into three categories: end-to-end solutions (e.g., Eifel [3]), physical layer solutions (e.g., [4], [5]), and cross-layer solutions (e.g., [6]–[9]). However, the high bit rate offered by LTE, which is comparable to or even larger than wired links, alleviates substantially the problems of TCP over more traditional wireless networks investigated in the literature. Performance of TCP over LTE is affected mostly by the links of the wired network and total bandwidth available at the serving base station. In [10], an end-to-end solution to improve TCP during the handover has been proposed, but the drawback is that it requires modification of existing TCP protocols. The impact of LTE handover on user connection has been considered in [11], where some scheduler modifications to improve the user throughput are suggested. However, the core network status is not considered, where the network load plays a significant role on performance. As a consequence, scheduler modifications may have limited effect on the TCP throughput. In [12] a user data forwarding during the handover is investigated to increase the throughput.

In this paper we investigate TCP performance during the intra LTE handover procedure in a complete network scenario. We consider the data forwarding technique studied in [12] and propose some mechanisms to increase performance significantly. We show that TCP suffers of throughput degradation during the handover. Consequently, we propose some improvements of different complexity, namely: two forwarding avoidance algorithms (a simple one, and a second one that uses a predictor) and active queue management. These solutions require only a slight modification of the existing LTE handover procedure, and act mostly at the network layer. Differently from previous studies, we improve performance of TCP during LTE handover by acting inside the transport and not at the access network. We evaluate performance of the proposed solutions by an extension of the *ns-2* simulator. The forwarding avoidance algorithms with the predictor shows the largest improvement, but at the cost of a slightly increased complexity of the handover algorithm.

The remainder of the paper is organized as follows: In Section II, we describe in detail the problem we are addressing. In Section III background information on TCP and handover is given. A performance analysis of TCP over LTE is developed in Section IV. Improvements are proposed in Section V. In Section VI, performance of these solutions are investigated.
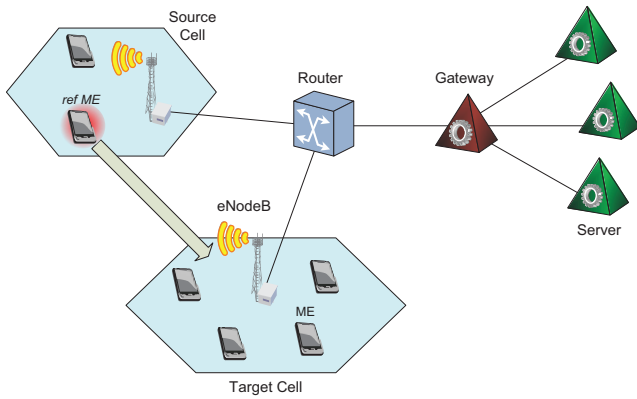
Fig. 1.   System Model.

Finally, in Section VII, conclusions and future perspectives are given.

II. PROBLEM FORMULATION

We consider a system scenario with two cells served by two different evolved Node B (eNodeB), a variable number of Mobile Equipments (MEs), divided among the source and the target cell (see Fig. 1).

An ME is doing an intra LTE handover from the first eNodeB to the second one. Let us denote such an ME by *ref ME*. *ref ME* is using a TCP connection to download data from a server. The remaining MEs are competitors for *ref ME* for the usage of the available bandwidth. A core-network router connects the two eNodeB to a server pool in which each server works as traffic source for a number of MEs. Let SeNodeB and TeNodeB be the Source and Target eNodeB for *ref ME*, respectively.

During the handover procedure, packets waiting in the SeNodeB to be sent to the *ref ME* are forwarded from the SeNodeB to the TeNodeB via a core-network router. This mechanism has negative effects for the TCP round trip time of *ref ME*. The forwarding may cause an increase of delay in the packet delivery, especially when the network is congested. This could cause a retransmission timeout expiration of TCP and the congestion window could drop to 1 segment. As a result, the *ref ME* could experience a dramatic drop of the throughput during the handover. Therefore, in this paper we tackle this problem and propose some control mechanisms to counteract such an adverse situation.

In the next section, we give some background information needed for the evaluation of performance loss and the design of these control mechanisms.

III. BACKGROUND

In this Section we give an overview of LTE, TCP, and handover.

A. TCP over LTE

LTE is expected to improve substantially end-user throughput, cell capacity, and transmission latency. Given the popularity of the Transmission Control Protocol TCP, and Internet

Protocol (IP) for carrying all types of traffic, LTE supports TCP and IP-based traffic with end-to-end quality of service.

TCP [13] is a reliable transport protocol that uses a congestion window cwnd to send packets. The TCP reliability is achieved using ARQ mechanism based on positive acknowledgments. TCP packets are cumulatively acknowledged when they arrive in sequence; out of sequence packets cause the generation of duplicate acknowledgments. TCP sender detects a packet loss either when multiple duplicate acknowledgments (3 is the default value) arrive or when a retransmission timeout (RTO) expires. The RTO value is calculated dynamically based on RTT measurements.

LTE assumes that the end-to-end TCP protocol governs the congestion control and adapts to the varying network conditions and handle packet loss. The LTE system supports quality of service mechanisms over radio and in the transport network, but no flow control mechanisms are supported. This implies that packet buffering/dropping for TCP could occur during congestion in any node of the core network (e.g., in the base station or a core-network router).

In LTE, the connection from a server to the MEs crosses the core network exploiting an IP tunnel [14], which goes between the serving gateway and eNodeB, and is based on the GPRS Tunneling Protocol (GTP). Due to this tunneling protocol, an explicit congestion signaling is not allowed even if the user connection were able to handle it. The IP header handled by the core network router belongs to the tunnel application rather than to the user, and the signaling will be trashed once the packet has reached the end of the tunnel. We see in the next subsection that such a tunneling protocol puts some constraints to the improvement of TCP performance during LTE handover.

B. Handover

During a handover, the TCP protocol endpoints must be moved from the SeNodeB to the TeNodeB. The message chart of the procedure is reported in Fig. 2 [14]. The SeNodeB collects the *ref ME* measurements about the link quality, and assists the function controlling the *ref ME*'s handover. Next, SeNodeB makes a decision based on measurement reports and RRM information to hand off the *ref ME*, and it issues a handover request message to the TeNodeB.

After the last phase described above, an admission control may be performed by the TeNodeB according to the received quality of service information. The TeNodeB prepares the handover and sends the handover request acknowledge to the SeNodeB. Such a message may also include information about Radio/Transport Network Layer (RNL/TNL) for the forwarding tunnels. When the handover request acknowledge is received, data forwarding from the SeNodeB to the TeNodeB through the router (recall Fig. 1) may be initiated. The SeNodeB generates the handover command (RRC message) towards the *ref ME*. Then, the *ref ME* performs a synchronization to TeNodeB and accesses the target cell. It then sends the handover message to indicate that the handover procedure is completed.
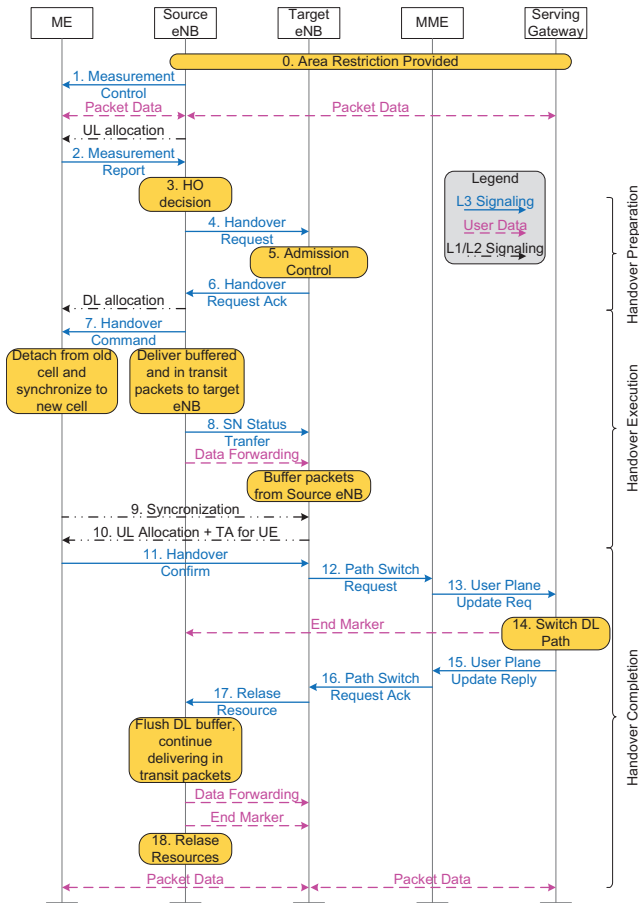
Fig. 2. Message chart of the LTE handover procedure. The control plane messages (solid and dot-dashed arrows) and the flow of the user packets (dashed arrows) are reported [14].

The TeNodeB can now start sending data to the *ref ME* and, at the same time, send a path switch message to the server MME to inform that the *ref ME* has changed cell. The MME sends a user update request message to the serving gateway which switches the downlink data path to the TeNodeB and sends one or more "end marker" packets on the old path to the SeNodeB, which can then release any resources previously associated to the *ref ME* (steps 7 to 16 in Fig. 2).

During the handover there is a time interval when packets on the new direct path (serving gateway – TeNodeB) and packets on the forwarding path (serving gateway – SeNodeB – core-network router – TeNodeB) may arrive in parallel at the TeNodeB. This gives rise to a possible increase of the RTT of the forwarded packets, since they have to do twice a queueing in the core-network router. Moreover, there is also the potential problem of out of order packet delivery to the *ref ME* when it is attached to the TeNodeB. In [12], a packet reordering feature inside the TeNodeB has been proposed to avoid the out of order delivering. Nevertheless, the potential increase of the RTT is still present, as we investigate in the following.
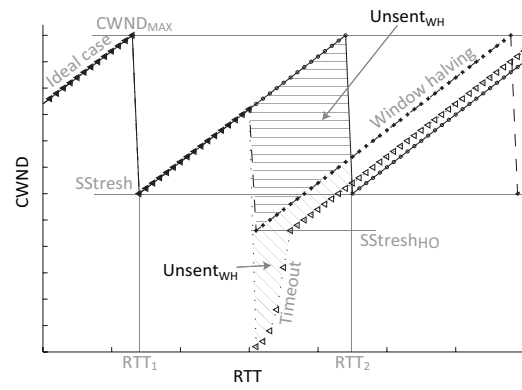

Fig. 3. Difference between ideal case and timeout occurrence.

IV. PERFORMANCE ANALYSIS

In Section III-B we have shown that user data are forwarded from the SeNodeB to the TeNodeB via a core-network router during the handover procedure. This mechanism has detrimental effects on the RTT, because the forwarded packet may experiments a dramatic increase of the delay, especially when the receiving network is congested. In particular, the forwarded packets increase the RTT as measured at the server due to two queuing phases. This first queuing is experienced by the packet transmission from the server to the core-network router, whereas the second queuing is due to the forwarding from the SeNodeB to the router during the handover. If these queuing increase the RTT significantly, the cwnd could halve, or drop its value to 1 segment at worst. In the following, we analyze such a performance degradation in detail. In particular, we examine both cwnd halving and timeout occurrence, compared with the optimal case, where cwnd does not suffer these problems.

Fig. 3 shows the behavior of the cwnd during the handover. We assume that the cwnd is in the steady state and we focus on the time interval between an initial instant RTT_1 and a final instant RTT_2. During such an interval we want to quantify the reduction of amount of data that is not transmitted due to the handover. For performance analysis, we define as *throughput* during the handover such an amount of data. We distinguish three possible situations: an optimal case, a window halving case, and a timeout occurrence case. We analyze them in the sequel.

In the optimal case the congestion window keeps on following the steady-state course regardless of the handover event. The amount of data delivered during the period of interest is

$$D_I = \sum_{i=RTT_1}^{RTT_2} \text{cwnd}(i) f_{D_I} = \sum_{i=SSthresh}^{\text{cwnd}_{MAX}} i f_{D_I}$$

$$= \frac{3}{2} \cdot \frac{\text{cwnd}_{MAX}}{2} \left(\frac{\text{cwnd}_{MAX}}{2} + 1 \right) f_{D_I}, \quad (1)$$

where cwnd_{MAX} is the maximum value assumed by the cwnd in the steady-state phase, $SSthresh = \text{cwnd}_{MAX}/2$ according to the TCP standard, and f_{D_I} is the probability of the steady-state case occurrence.

The window halving case is due to an out-of-order delivering. It happens if the router buffer exceeds the maximum length during the packet forwarding. The difference of throughput between the optimal case and this case is shown in Fig. 3 with horizontal pattern. Specifically, we need to subtract the horizontal patterned area $Unsent_{WH}$ in Fig. 3 to (1) to obtain the amount of data delivered:

$$
\begin{aligned}
D_{WH} =&(D_I - Unsent_{WH})F_{D_{WH}} = D_I f_{D_{WH}} + \\
&- \sum_{i=\texttt{cwnd}_{HO}}^{\texttt{cwnd}_{MAX}} (\texttt{cwnd}_{HO} - SSthresh_{HO}) f_{D_{WH}} \\
=& \left[D_I - \frac{\texttt{cwnd}_{HO}}{2} \cdot (\texttt{cwnd}_{MAX} - \texttt{cwnd}_{HO} + 1) \right] f_{D_{WH}} ,
\end{aligned}
\tag{2}
$$

where $\texttt{cwnd}_{HO}$ is the value assumed by the $\texttt{cwnd}$ at the handover start-time, $SSthresh_{HO} = \texttt{cwnd}_{HO}/2$, and $f_{D_{WH}}$ is the probability of the window halving case occurrence. By using such a model, the allowed values for $\texttt{cwnd}_{HO}$ are those belonging to the $[SSthresh, \texttt{cwnd}_{MAX}]$ interval that corresponds to all real possible values.

The timeout case is the worst event that can occur during the handover. It is a consequence of spurious timeout due to the high *jitter* increase, which is exacerbated by the data forwarding. Fig. 3 shows that in such a case we have more loss than the previous two cases. The unsent amount of data can be divided in two different contributions: $Unsent_{WH}$, which is due to the $\texttt{cwnd}$ halving, and $Unsent_{SS}$ (diagonal pattern) which is due to the slow-start phase. The throughput can be written as

$$
\begin{aligned}
D_{SS} =&(D_I - Unsent_{WH} - Unsent_{SS})f_{D_{SS}} \\
=& [D_{WH} - k \cdot (\texttt{cwnd}_{MAX} - \texttt{cwnd}_{HO} + 1) + \\
&- \sum_{i=0}^{k-1} \left(\frac{\texttt{cwnd}_{HO}}{2} - k + i - 2^i \right) \Bigg] f_{D_{SS}} .
\end{aligned}
\tag{3}
$$

where $f_{D_{SS}}$ is the probability of timeout occurrence. The $Unsent_{SS}$ quantity depends on k, which represents the total number of RTT in which the slow-start phase is on duty. The $SSthresh_{HO}$ belongs to the following interval

$$
SSthresh_{HO} \in \left(\frac{SSthresh}{2}, SSthresh \right) ,
\tag{4}
$$

where $SSthresh = \dfrac{\texttt{cwnd}_{MAX}}{2}$ and k are

$$
k = \lceil \log_2 SSthresh_{HO} \rceil + 1.
\tag{5}
$$

The theoretical evaluation of the probabilities of the optimal case, the window halving case, and the timeout case is difficult. Hence, we resorted to a simulation approach. In Subsection VI-B, we show that $f_{D_I} \leq f_{D_{WH}} \leq f_{D_{SS}}$. In other words, the timeout case has the highest probability. For performance comparison, in Fig. 4 we evaluate the normalized throughput by setting $f_{D_I} = f_{D_{WH}} = f_{D_{SS}}$. Fig. 4 shows the throughput for two handover situations as achieved by (2) (3) as function of $\texttt{cwnd}_{MAX}$ parameter and normalized with
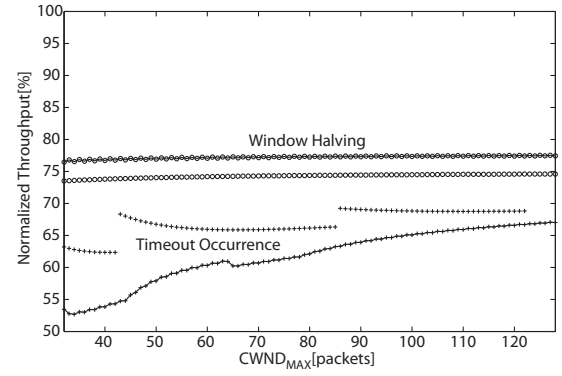

Fig. 4. Two possible averages of the throughput given by (2) (3) as function of $\texttt{cwnd}_{MAX}$ parameter and normalized with respect to (1). The continuous lines represent the average of the normalized throughput when the handover occurs when the the congestion window is at the maximum $\texttt{cwnd}_{MAX}$ (see Fig. 3). The lower circled and upper crossed lines assume that the handover occurs in the middle of the interval $(SSthresh, \texttt{cwnd}_{MAX})$ (see Fig. 3).
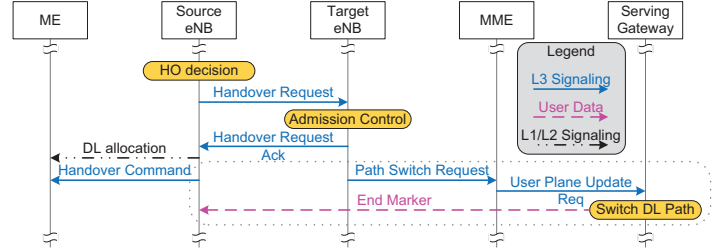

Fig. 5. Fast path switch modification to the handover message chart.

respect to (1). We observe a substantial performance degradation, which suggests the need of improving solutions for TCP during LTE handover.

V. IMPROVING HANDOVER

In this section, we propose some methods to enhance TCP performance during handover. The solutions are grouped into two categories: Forwarding avoidance, and active queue management. Details follow in the sequel.

A. Forwarding Avoidance

The basic idea is to avoid the packet forwarding from the SeNodeB to the core-network router and prevent that the RTT increases so much that a spurious timeout occurs. This can be accomplished in two ways: by a modification of the path switch time, or a prediction of the handover. In the current LTE handover procedure (see Fig. 2) the path switch command is sent by the TeNodeB only after it has received the handover confirm message from the *ref ME* when it is attached to the TeNodeB. The solutions proposed in the following subsections require modifications of such a handover procedure in that the resource allocation on the TeNodeB is done after a request made by the network and not by the *ref ME*.

1) Fast Path switch: This solutions consists in sending the path switch command to the serving gateway immediately after the handover command is sent by the SeNodeB, instead of
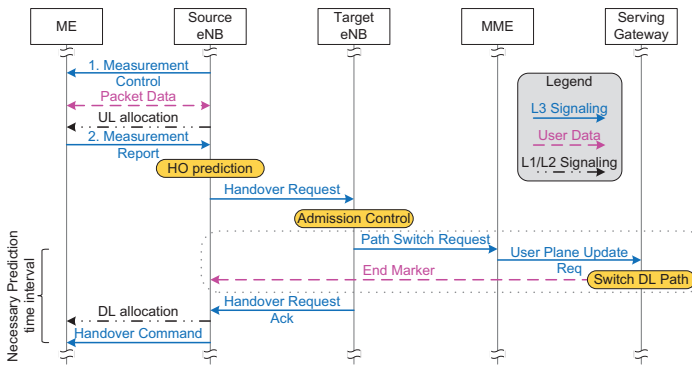
Fig. 6. Modification to the handover message chart for the handover prediction.

making the path switch upon the reception of the handover confirm message by the TeNodeB. Indeed, in the current LTE handover algorithm, even though the handover command is sent, the serving gateway keeps sending packets to the SeNodeB during the whole duration of the handover execution, as discussed in the end of Subsection III-B. Hence, the objective is to reduce the number of packets that do a double queuing in the core-network router, which translates in a reduction of the potential RTT increase. The modification of the first part of the message chart required by the fast path switch solution is reported in Fig. 5.

2) Handover prediction: The previous proposal has the advantage of simplicity. However, there can be packets doing a double queuing. To overcome this limitation, it is possible to further anticipate the path switch command of the previous solution by a prediction of the handover start-time. The prediction can be done easily up to the coherence time, which spans over hundred of milliseconds []. By doing such a prediction, the packets sent so far (and that are queued in the core-network buffer) follow the path given by the server – core-network router – SeNodeB and are sent to the *Ref ME* during the time interval between the anticipated patch switch command and the handover command. The new packets that are sent afterwards the anticipated path switch command follows the new path given by the server – core-network router – TeNodeB, thus avoiding a double queuing in the core-network router. This algorithm allows the SeNodeB to emptying its buffer containing the *ref ME* data before the handover command, and thus it avoids completely the packet forwarding procedure. The modification in the message chart are reported in Fig. 6, which is related to the time interval between the path switch command and handover command.

The prediction of the handover start-time can be done by using cross-layer information on the bandwidth available at the SeNodeB for the *ref ME*. Such a prediction is accurate to up some hundreds of milliseconds, which is above the typical RTT, namely the time needed to the SeNodeB to emptying its buffer. The drawback of this solution is that it is more complicated than the one presented in the previous subsection. Moreover, if the prediction is erroneous, it could lead to a total disconnection of the *ref ME* from the LTE network. However, such an event has a negligible probability of occurrence.

B. Active Queue Management

Active Queue Management (AQM) refers to the control taking place in core-network routers. There are several motivations for using AQM, such as reduced packet losses, reduced queueing delay and jitter, improved throughput. These improvements are achieved by using as control input the arrival rate of data packets and the queue size for a particular outgoing link, and as output a decision on how to mark or drop packets. The AQM mechanism selects the probabilities for marking and dropping packets. Then, each forwarded packet is marked/dropped with such a probability. Servers will then react to these marks and adjust their sending rates.

Several AQM mechanisms are evaluated in [15], with and without the use of Explicit Congestion Notification (ECN). In TCP, an option is available for notifying the congestion inside the network [16]. However, in the tunnel connection, the network is configured such that between the serving gateway and the eNodeB the UDP protocol is used, which does not allow for ECN notification. It follows that the network configuration is ECN-unaware even if the user transport protocol provides such a feature.

To overcome the limitation described above, the basic idea is to enable the AQM policy with ECN *marking* in the core-network router and forward such a marking to the user IP header located at the end of the tunnel. This solution means to provide a feature enhancement inside the eNodeB and gateway to mark/drop the packets depending on the user transport protocol. The *mark* is forwarded by the eNodeB to the user IP header only if it carries the ECN-aware sequence (see Fig. 7(a)). Otherwise, when the user IP header handles an ECN-unaware TCP the packet will be dropped, as shown in Fig. 7(b).
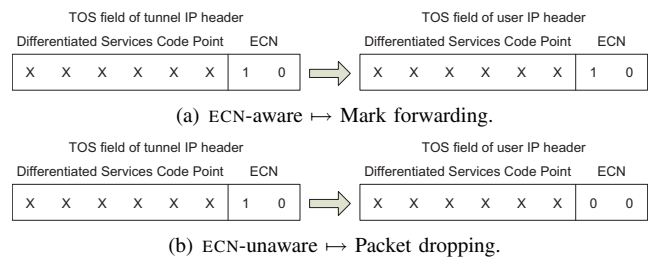

Fig. 7. Parameters of the reference ME connection during the handover [16].

VI. SIMULATION STUDY

In this section, we validate the performance improvements of TCP when LTE handover occurs as achieved by the solutions we have proposed in Section V. To this aim, we have extended the *ns-2* simulator.

A. The LTE Simulator

At present *ns-2* does not support multiple radio interfaces. It does not have flexible tools for the cross-layer control of

TABLE I
COMMON PARAMETERS OF ALL SIMULATION.

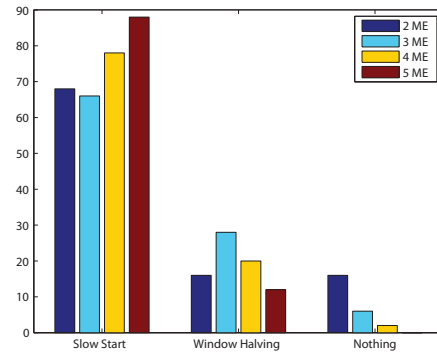| Parameter	Value
Cross-traffic	FTP
Bitrate of router links	12 Mb/s
Bitrate of server links	100 Mb/s
Latency of wired links	2 ms
eNodeB queue length	200 packets
Reordering	Enabled
Detach time	30 ms
eNodeB Scheduling policy	WFQ
IP Packet size	1500 byte

Fig. 8. Probabilities of the timeout case $f_{D_{SS}}$ (left), the window halving case $f_{D_{WH}}$ (middle), and the optimal case f_{D_I} (right) with the standard handover procedure.

TABLE II
SIMULATION PARAMETERS ABOUT FORWARDING AVOIDANCE MECHANISM.

Parameter	Value
Total simulation time	5 s
Nodes attached to SeNodeB	1
Nodes attached to TeNodeB	5
Core network router queue size	90 packets
$cwnd_{MAX}$	$44 \simeq 64$ KB
Prediction Step	130 ms

communication systems and, moreover, it is not possible to reuse the existing implementations of the wireless channel because they do not work for LTE radio interfaces (they are all *ad-hoc* solution to the specific problem which they are implemented for). Therefore, we have implemented an LTE simulator by using the framework *Multi InteRfAce Cross Layer Extension for ns-2* (nsMiracle) [17], which is conceived as a set of dynamic libraries that are loaded to add support for multi-technology and cross-layering. It exploits also a patch that facilitates the use of dynamic libraries in *ns-2* [18], which allows us to load in the simulator only the necessary modules and makes the simulation faster.

The TCP version used in our simulations is TCP Reno since it is widely used in the Internet. Other versions of TCP, such as SACK, could lead to better performance [19], but TCP Reno plays a significant role in the mobile applications [20]. The RTO parameters have been set correctly according to the specifications RFC2988.

The reference scenario of the simulator is depicted in Fig. 1. In the simulator, the starting of a communication between each ME and server follows the TCP *three way handshaking* [13] and typically the file transfer is started by the server without any request.

In Tab. I we have reported the parameters common for all simulations. The cross traffic on the core-network router is assumed FTP. The detach time is the interval time of the hard handover. WFQ is the LTE scheduler used by the eNodeB. Other schedulers give basically same performance results as those described in the following.

B. Simulation Results

Fig. 8 shows the probabilities of the optimal case f_{D_I}, the window halving case $f_{D_{WH}}$, and the timeout case $f_{D_{SS}}$ (see Section IV). The source cell serves the *ref ME*. The number of mobiles in the target cell varies among 2 and 5. A set of 50 simulations was done for every network configuration. Then simulation data are collected according to three different classes: The first class is timeout occurrences, in which the number of times that the *ref ME* experiences a RTO event are collected. The second class is the window halving occurrences and collects the times that the *ref ME* experiences a cwnd halving. Finally the last class represents the case in which

the handover is transparent to the TCP protocol. From Fig. 8, we observe that the RTO expiration is very frequent, which confirms that the mechanisms to improve TCP performance during the LTE handover are essential. The improvements offered by the solutions presented in Section V are discussed next.

1) Forwarding Avoidance: Here we investigate the benefits of the two forwarding avoidance mechanisms. Tab. II shows the key parameters used in the simulations. The prediction step of 130 ms gives a reasonable path switch in advance with respect to the actual handover. The figure was chosen with respect to the typical RTT (around 100 ms) to empty the pipe of the core-network router.

Figs. 9 and 10 show the congestion window update and the RTT experienced by the received packets. Figs. 9(a) and 10(a) describe the case of standard LTE handover procedure, in which a spurious timeout occurs due to RTT increasing; 9(b) and 10(b) are referred to the application of the fast path switch procedure, which is only partially able to avoid the spurious timeout event; better performance is achieved by the handover prediction (Fig. 9(c) and 10(c)), which avoids always the spurious timeout and causes only a window halving due to a packet dropping done by the router buffer.

Fig. 11 shows the probabilities of timeout, window halving and ideal case similar to Fig. 8, but with the difference that the handover prediction algorithm presented in in this paper is employed. We observe a substantial reduction of the slow start
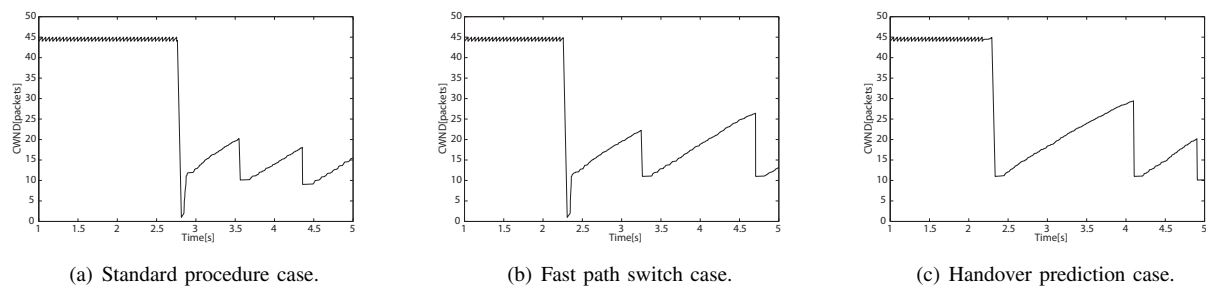
(a) Standard procedure case.

(b) Fast path switch case.

(c) Handover prediction case.

Fig. 9. cwnd for *ref ME* during the handover. The handover start-time is 2.3 s

(a) Standard procedure case.

(b) Fast path switch case.

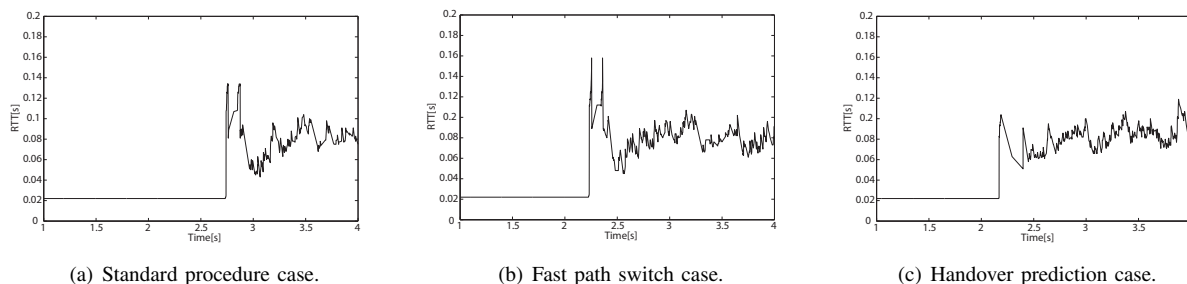(c) Handover prediction case.

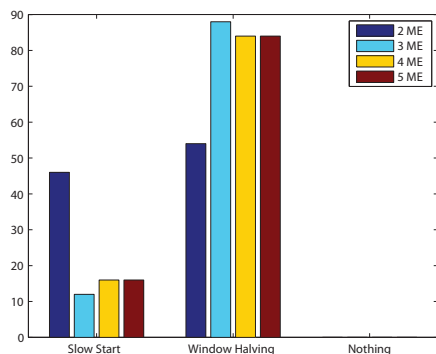Fig. 10. RTT for *ref ME* during the handover. The handover start-time is 2.3 s

Fig. 11. Probabilities of the timeout case $f_{D_{SS}}$ (left), the window halving case $f_{D_{WH}}$ (middle), and the optimal case f_{D_I} (right) when the handover prediction is enabled.

TABLE III
SIMULATION PARAMETERS ABOUT AQM.

Parameter	Value
Total simulation time	10 s
Nodes attached to SeNodeB	4
Nodes attached to TeNodeB	3
max p	0.1
thresh	20 packets
maxthresh	65 packets
Core network router queue size	90 packets

TABLE IV
ACTIVE QUEUE MANAGEMENT SIMULATION RESULTS.

Implemented Queue Management	Link R → SeNodeB Drop Rate [%]	Link R → TeNodeB Drop Rate [%]
Drop Tail	2.2	3.9
RED (Dropping)	2.2	3.6
RED (Marking)	2.1	3.3

frequency and an increasing of the window halving frequency, which are beneficial in terms of throughput. This is due to the reduced timeouts experimented by the *ref ME* during the handover thanks to the prediction technique. It is possible to show that the conclusions derived from the simulation results reported in Fig. 9, 10, and 11 hold for other network configurations, namely for networks with a different number of MEs.

2) Active Queue Management: Here we present simulations to investigate the advantages achieved by using RED queue management instead of the typical drop-tail policy. Tab. III shows the parameter settings of the simulator. The RED parameters have been set according to the suggestion given in [21].

In Tab. IV we report the probability of dropping packets on the link between the core-network router and the SeNodeB, and the link between the core-network router and the TeNodeB. Recall that a smaller dropping probability means a better quality of service for the *ref ME*. We observe that exploiting the RED queue management carries to more benefits in the link between router and TeNodeB. Indeed, even though after the handover such a queue handles four different TCP connections and then an increased congestion level, the dropping probability is reduced. According to [22] we can say that RED allows better performance with a congested network
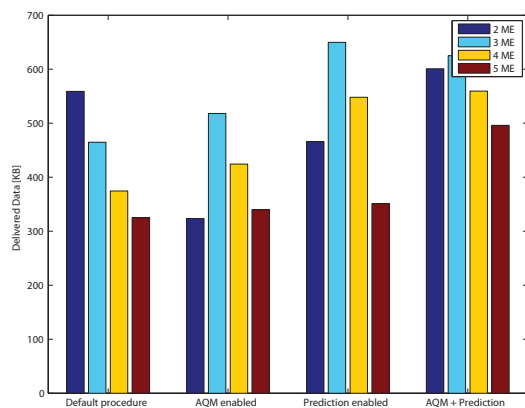
Fig. 12. Amount of data delivered successfully to the *ref ME* from the starting of the handover until one RTT after the *ref ME* connects with the TeNodeB. The leftmost bars are referred to the standard handover, the second group to the handover with AQM, the third group to the handover with prediction, and the rightmost group to the handover with a simultaneous use of handover prediction and AQM.

in which the early dropping works exactly to avoid more drops due to the incoming congestion. We conclude that if the cross network ECN is implemented, the marking benefits also can be exploited, this reduces the drop rate specially for the more congested link. The conclusions derived from the simulation results that we have reported in Tab. IV hold also for other network configurations.

Finally, in Fig. 12 we compute the amount of data delivered from the starting of the handover until one RTT after the moment in which the *ref ME* is connected to the TeNodeB for each improving solution and a combination of them. We see a substantial increase of data delivered, particularly when the prediction technique is associated with AQM.

VII. Conclusion

In this paper we presented a complete analysis of TCP performance during the LTE handover. We observed that the throughput of the mobile equipment doing the handover drops significantly. This is due to a forwarding procedure of packets from the source evolved NodeB to the target evolved NodeB. Therefore we proposed and evaluated several improving solutions. Specifically, we investigated a forwarding avoidance algorithms (which includes a prediction technique and a fast path switch) and active queue management. Extensive numerical simulations, obtained by an extension of the *ns-2* simulator, showed that the handover prediction algorithm is the most promising technique to improve TCP performance during the LTE handover.

Future work includes an extension of the theoretical analysis presented in this paper to model analytically the TCP flows and provide an optimization of the buffer size of the router. Furthermore, we plan to study the joint effect of the handover predictor and active queue management.

Acknowledgment

The authors are grateful to Andràs Ràcz, Ericsson Research, Budapest, for fruitful suggestions to design the simulator used in this paper, for providing us a trace file needed for the simulation implementation, and for the comments for writing the paper.

References

[1] K. Bogineni, R. Ludwig, P. Mogensen, V. Nandlall, V. Vucetic, B. Yi, and Z. Zvonar, "LTE Part I: Core network," *Communications Magazine, IEEE*, vol. 47, no. 2, pp. 40–43, February 2009.

[2] T. Ye, X. Kai, and N. Ansari, "TCP in wireless environments: problems and solutions," *IEEE Communications Magazine*, vol. 43, pp. 27– 32, March 2005.

[3] R. Lundwig and R. Katz, "Making TCP robust against spurious retransmission," *ACM Computer Communication Review*, vol. 30, pp. 30 – 36, January 2000.

[4] H. Balakrishnan, S. Seshan, and R. H. Katz, "Improving Reliable Transport and Handoff Performance in Cellular Wireless Networks," *ACM Wireless Networks*, vol. 1, no. 4, 1995.

[5] J. Moon and B. J. Lee, "Rate-adaptive snoop: a TCP enhancement scheme over rate-controlled lossy links," *IEEE/ACM Transactions on Networking*, vol. 14, pp. 603–615, June 2006.

[6] M. Ivanovich, P. Bickerdike, and J. Li, "On TCP performance enhancing proxies in a wireless environment," *IEEE Communications Magazine*, vol. 46, pp. 76–83, September 2008.

[7] M. Chiang, "Balancing transport and physical layers in wireless multihop networks: jointly optimal congestion control and power control," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 1, pp. 104–116, January 2005.

[8] F. Ren, X. Huang, F. Liu, and C. Lin, "Improving TCP Throughput over HSDPA Networks," *IEEE Transactions on Wireless Communications*, vol. 7, pp. 1993 – 1998, June 2008.

[9] N. Mller, "Window-based congestion control: Modeling, analysis and design," Ph.D. dissertation, School of Electrical Engineering, KTH, Automatic Control, Osquldas vg 10, 10044 Stockholm, July 2008.

[10] W. Liao, C. Kao, and C.-H. Chien, "Improving TCP performance in mobile networks," *IEEE Transactions on Communications*, vol. 53, pp. 569– 571, April 2005.

[11] L. Bajzik, P. Horvath, L. Korossy, and C. Vulkan, "Impact of Intra-LTE Handover with Forwarding on the User Connections," *Mobile and Wireless Communications Summit, 2007. 16th IST*, pp. 1–5, July 2007.

[12] A. Racz, A. Temesvary, and N. Reider, "Handover Performance in 3GPP Long Term Evolution (LTE) Systems," *Mobile and Wireless Communications Summit, 2007. 16th IST*, pp. 1–5, July 2007.

[13] J. Postel, "Transmission Control Protocol," Internet Engineering Task Force, RFC 0793, Sep. 1981. [Online]. Available: http://www.rfc-editor.org/rfc/rfc793.txt

[14] 3GPP, "Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access (E-UTRAN); Overall description; Stage 2," 3rd Generation Partnership Project (3GPP), TS 36.300, Sep. 2008. [Online]. Available: http://www.3gpp.org/ftp/Specs/html-info/36300.htm

[15] L. Le, J. Aikat, K. Jeffay, and F. D. Smith, "The effects of active queue management on web performance," in *SIGCOMM '03*. ACM, 2003.

[16] K. Ramakrishnan, S. Floyd, and D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP," Internet Engineering Task Force, RFC 3168, Sep. 2001. [Online]. Available: http://www.rfc-editor.org/rfc/rfc3168.txt

[17] "ns-MIRACLE: Multi InteRfAce Cross Layer Extension for ns-2." [Online]. Available: http://www.dei.unipd.it/ricerca/signet/tools/nsmiracle

[18] "Dynamic modules in ns-2." [Online]. Available: http://telecom.dei.unipd.it/ns/miracle/ns_dynamic_libraries/

[19] F. Xin and A. Jamalipour, "TCP throughput and fairness performance in presence of delay spikes in wireless networks," *Int. J. Commun. Syst.*, vol. 18, no. 4, pp. 395–407, 2005.

[20] S. Fu and M. Atiquzzaman, "Modelling TCP Reno with spurious timeouts in wireless mobile environments," Oct. 2003, pp. 391–396.

[21] S. Floyd, "RED: Discussions of Setting Parameters," November 2007. [Online]. Available: http://www.aciri.org/floyd/REDparameters.txt

[22] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Trans. Netw.*, vol. 1, no. 4, pp. 397–413, 1993.