TCP Application Recovery Improvement After Handover in Mobile Networks

Hajer Souri[1], Amine Dhraief[1], and Abdelfettah Belghith[2]

[1] HANA Lab, University of Manouba, Tunisia
[2] College of Computer and Information Sciences, King Saud University, Saudi Arabia
{hajer.souri,amine.dhraief}@hanalab.org, abelghith@ksu.edu.sa

Abstract. Host and network mobility support mechanisms aim to provide a seamless access to Internet while end users are changing their point of attachment. When a mobile entity roams from a network to another, a handover occurs causing communication session discontinuity degrading real-time application performance. Assuming that such inconvenience is a congestion, TCP recovery is slow and careful. We have previously proposed a cooperation between TCP and MIPv6 and proved its efficiency through OMNeT++ simulations. In this paper, We study its efficiency in the context of mobile networks. Therefore, we implement xNEMO; a compliant implementation of NEtwork MObility Basic Support (NEMO BS) in OMNeT++. Besides, in order to enhance the performance of TCP application, we propose to modify the TCP behaviour after a recovery from a handover. Simulations show that the cooperation between TCP and NEMO BS reduces the Application Recovery Time and enhances the performance of the TCP application.

Keywords: Network Mobility, TCP, Application Recovery Time, OMNeT++.

1 Introduction

The constant reachability of mobile IP devices is no more a vision but a dominant reality. The continuous connectivity to Internet requires a good management of frequent changes from a network to another in order to guarantee the good functioning of Internet applications, especially the real time ones. Therefore, it is important to assign a unique identifier to each mobile entity which allows to maintain communication sessions with other peers. Mobile IPv6 (MIPv6)[7] and Network Mobility Basic Support (NEMO BS)[2] are the IETF proposed protocols to manage respectively host mobility and network mobility. They enable mobile entities/networks to ensure session continuity while roaming between IP networks . However, keeping the roaming seamless still a major topic of research.

Guaranteeing reliability for Internet applications is ensured through the use of TCP, a major Internet transport layer protocol. It controls data transmission and ensures the stability of Internet application. Improving the performance of TCP is a key element to guarantee the reliability of application and even the

M. Kassab et al. (Eds.): Nets4Cars/Nets4Trains/Nets4Aircraft 2015, LNCS 9066, pp. 60–71, 2015.
DOI: 10.1007/978-3-319-17765-6_6

transparency of network roaming. In this context, it is important to mention that TCP different algorithms confuse between movement and congestion when recovering after packet losses. This was highlighted in our previous work[4,3] when studying the impact of the mobile node's movement on TCP sessions. The reaction of TCP after the packet loss caused by devices movement is the same as its reaction at the diagnosis of a network congestion, *i.e.* TCP recovery after network switching takes a long time to occur.

In order to address such misbehaviour in the context of host mobility, we have previously proposed to establish a novel solution that enhances the Application Recovery Time (ART) of TCP applications for MIPv6 nodes[5]. We proposed a cross-layer cooperation between TCP and MIPv6 layers notifying TCP of the end of an ongoing handover. Thus, the TCP activity is recovered. Our solution was proven to efficiently reduce up to 43% of the ART.

In this paper, we extend our previous work[5] in order to deal with mobile networks. The focus of our study is the efficiency of our proposed cross-layer cooperation when it is between TCP and NEMO BS. Hence, we implement a compliant version of NEMO BS by extending xMIPv6, the framework simulating the behaviour of MIPv6 nodes in OMNeT++[10]. Besides, being aware that the reason of the transmission interruption was not a congestion but a handover,we also propose to modify the behaviour of TCP after a recovery from a handover by resettling the state of TCP as if no interruption occurred and thus the careful and slow recovery is avoided. The performance evaluation of our proposals shows that the TCP application performance is significantly improved in the context of mobile networks.

The reminder of this paper is organized as follows: an overview of Network Mobility Basic Support as well as a quick review of TCP performance in mobile environments are given in section 2. Section 3 is dedicated to describe the issue of TCP applications while moving from a network to another and present our revised solution to enhance the functioning of TCP. In section 4, we present the evaluation of the performance of our overall proposal in the framework of mobile networks after implementing the NEMO BS specification on OMNeT++. Conclusion and perspectives are presented in the section 5.

2 Related Works

In the following section, we first present the Mobile Network Basic Support details and then brievly prensent TCP and its performance in mobile networks.

2.1 Network Mobility

A mobile network, NEMO, is the association of several devices connected and moving together forming a mobile network using each the same communication medium. In NEMO, each device, a Mobile Network Node(MNN), has its own IPv6 address but the same IPv6 prefix advertised by a specific router and shall be permanently reachable and connected.

NEMO Basic Support(NEMO BS) [2] was proposed as an extension to MIPv6 in order to manage the Network Mobility. There are different scenarios of mobile networks in the current daily life. Among them, we mention:

- Personal networks: People daily carry multiple IP devices such as smartphones, PDAs, cameras and laptops. A cell phone or a 3G key could act as a mobile router managing their connectivity and continuous reachability,
- Public transport systems are moving towards being equipped with mobile routers in order to enable passengers handheld devices to hold their connectivity,
- Car area networks: Intelligent cars are a strategic target of network mobility, as car industry is likely to equip cars with more connected IP devices. A car, hence, becomes a mobile network where different IP devices, already installed in the car or belonging to the passengers, wish to be constantly connected. Public transport vehicles where many passengers carrying their mobile devices seek Internet access, are also typical mobile networks.

A NEMO changes its point of attachment to the Internet with time. A Mobile Router (MR), a MIPv6 enabled mobile node endowed by routing capabilities, is responsible of providing Internet connection to the mobile and fixed nodes.

The home network has an assigned bloc of prefixes , Mobile Network Prefixes (MNPs), that a special entity called home agent advertises,hence, theoretically, MR may have one or few home addresses. These configured addresses are permanent and valid even when the NEMO is out of its home network but topologically correct only in the home link. Besides, MR advertises one or few prefixes in its subnet that are used by nodes on its link to configure their addresses.

When the NEMO leaves its home network and attaches to a new access point, it acquires a topologically correct address called Care-of Address (CoA) configured using its own ID combined with the prefix being advertised on the foreign link. Forthwith, MR sends a Binding Update (BU) with mobile network prefix

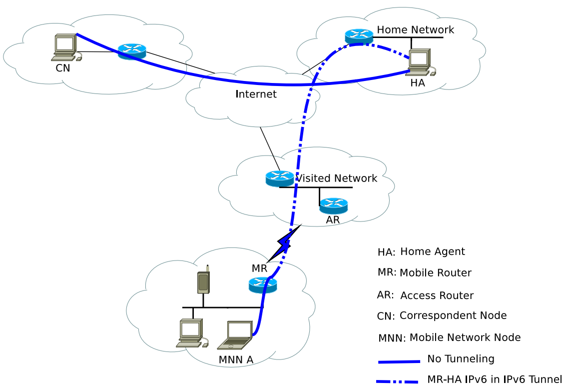

Fig. 1. NEMO Basic Support Protocol operation

information included to its home agent to inform about its newly configured address.

At the reception of the MR's BU, the home agent must first validate it, creates a new binding cache entry if no relevent one already exists. Then, it sends a Binding Acknowledgement (BA) to MR and establishes a bi-directional tunnel to MR for the requested MNP and starts advertising reachability to the mobile network. Hence, at the reception of a BA, MR may resume flowing data in the tunnel. Packets exchanged between the MNNs and their Correspondent Nodes (CNs) are routed through the home agent via the established tunnel (Figure 1).

2.2 TCP in Wireless and Mobile Networks

TCP is the major protocol that ensures Internet services reliability, congestion control, flow control and connection management. Mainly, it is able to control congestion by managing the transmission rate in order not to exhaust the available network bandwidth. Wireless and mobile networks suffers from the variability of the radio link quality and signals fluctuation so that transmission speed is lower and the Bit Error Ratio (BER) is higher.TCP performance are likely to be affected by mobility (*i.e.* handovers) and congestion that cause variable or even high delays (RTT)[9]. As a consequence, the retransmission timer expires. Hence, most TCP variations reduce their congestion window (cwnd) to 1 and enter a careful and slow transmission phase which degrades the application performance. [6], [1], [8] and others proposed solutions to improve the performance of TCP in wireless environments. Our work in [5] briefly descibes some of them.

3 Explicit Handover Cross-Layer Triggers and Its Implications in Mobile Networks

In [5], we defined the Application Recovery Time (ART) as the time that elapses between the last data packet received before a handover and the first data packet received after a handover. We showed that a handover usually finishes before the recovery of application as depicted in Figure 2.

This is due to the lack of communication between the IP and the transport layers. TCP continues executing its instructions waiting for the expiry of the retransmission timer that could have been doubled few times whilst the handover end had already took place. The adopted TCP algorithm keeps transmitting packets as far as needed while setting the retransmission timer (RTO) at each time. When an acknowledgment is received, RTO is canceled. But, when a handover occurs, Acks are lost and hence RTO expires triggering the detection of a problem. TCP, then, keeps retransmitting the packet that is suspected to be

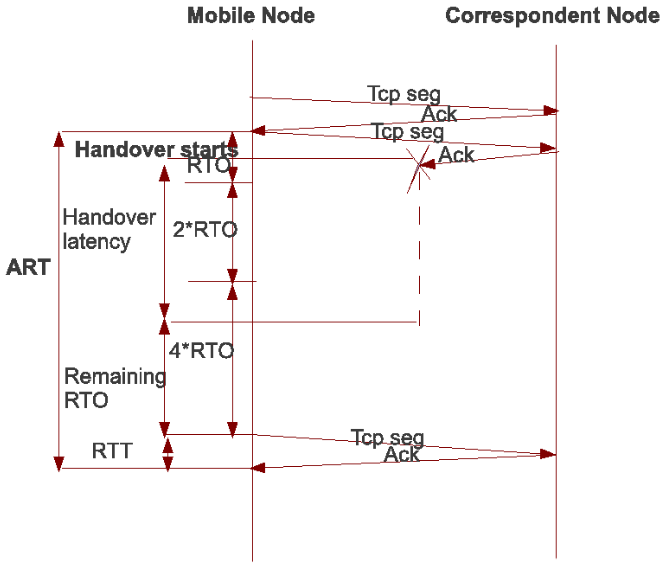

Fig. 2. Application Recovery Time

lost and resetting RTO to the double of its previous value at each expiry until the reception of an acknowledgment. Meanwhile, the handover may finish and a tunnel between the MR and the HA may be established but the ongoing TCP session would not notice it and keeps waiting the RTO expiry to resume retransmission.

To deal with this inconvenience, we proposed to establish a cooperative behaviour between the IP layer and TCP allowing the notification of TCP at the end of a handover and the establishment of a tunnel with the home network. Thus, TCP cancels the retransmission timer and retransmits the earliest non acknowledged segment. Our simulations showed that the proposal significantly reduces the ART in the case of MIPv6. In this paper we intend to prove its efficiency in the context of mobile networks by adopting the same cooperation mechanism.

Furthermore, canceling the retransmission timer will only speed the resumption of the TCP algorithm functioning. However, almost all of the TCP algorithms still consider the retransmission timer expiry as a congestion problem and then reduce remarkably its state variables. Most of the algorithms enter the Slow Start phase at the expiry of RTO. This implies the reduction of the cwnd and the ssthresh as follows:

$flightSize = min(awnd, cwnd)$
$ssthresh \leftarrow max(flightSize/2, 2)$
$cwnd = 1MSS$

where awnd is the TCP advertised window MSS is the maximum segment size. Then, TCP starts sending one single packet and waits until the reception of its acknowledgment to send two new other packets and so on until reaching the newly set ssthresh strikingly lower than the former value. Such reduction instantly degrades drastically the performance of the TCP transmission reducing significantly the rate of the mobile node's transmitted packets. Yet, the congestion estimation is completely wrong.

Indeed, as a handover lasts usually a few minutes at most, we assume that we have a quite restricted distance traveled during the movement of the mobile node/network. This assumption implies that we could consider that the density in the network that the mobile entity have just left is quite similar to the network area to which it currently belongs. Therefore, network congestion state is assumed to be relatively stable during the handover. Thereby, forcing TCP to reduce its state variables and enter Slow Start at the end of a handover is useless.

Therefore, we judged worthwhile to adapt the behaviour of TCP in mobile environments when it is running on a Mobile Node side. All the same, it is important to maintain it independent of any intermediaries. As the reception of an explicit handover cross-layer trigger informs the TCP layer about a handover taking place, it becomes simpler to amend the behaviour of TCP only when needed. So, we propose to resettle the TCP state variables after the handover to its recorded values just before the handover *i.e.* the Slow Start phase is eventually avoided and packet transmission rate would keep increasing just like the way it was increasing before the handover. At each expiry of an RTO, the RTO is doubled and the unacked TCP segment having the smallest sequence number is retransmitted. If the mobile node receives an ACK, that means that the network is congested. So, the mobile node must react conventionally by reducing the ssthresh and the cwnd and re-entering Slow Start smoothly increasing its packet transmission. However, if a handover cross-layer trigger is received, then we have a handover that has just been finished. Thus, our proposal is that the mobile node sends the unacked TCP segment having the smallest sequence number and when the ACK is received it checks whether it acknowledges some or all the data transmitted before the handover to decide data to be transmitted next. Besides, TCP resets cwnd and ssthresh to their values just before the handover and continues its normal operation. The TCP algorithm becomes as follows.

```
while RTO Expires do
  RTO ← RTO * 2
  Enter Slow Start
  Retransmit one segment
  if explicit handover cross-layer trigger then
    {Handover is the cause of the RTO expiry}
    Retransmit one segment
    if received ACK then
      cwnd ← recordedCwnd
      ssthresh ← recordedSsthresh
      if All data acknowledged then
        Send new data
      else
        Retransmit all unAcked data
      end if
    end if
  else if received Ack then
    {Congestion is the cause of the RTO expiry}
    Continue Slow Start {Congestion is the cause of the RTO expiry}
  end if
end while
```

4 Performance Evaluation

The performance evaluation of our proposal is the objective of this section. The influence of the cooperation between TCP and NEMO Basic Support as well as the modified TCP behaviour are investigated.

4.1 Simulation Setup

we perform evaluation on the OMNeT++ simulator [10] using the INET framework[1]. To achieve simulations, we use the Tahoe, Reno and NewReno implementation of TCP in INET.

Among the goals of this work, we want to test our proposed enhancements in mobile networks. However, there is no NEMO BS compliant implementation known for the OMNeT++ simulator. Therefore, we have had to design and develop our own implementation of the protocol according to the standards and specifications of the IETF. Having dissected the different features and specifications of the NEMO BS and carefully studied the xMIPv6 implementation, we have decided to be based on it to develop our NEMO BS protocol by extending its functionalities, data structures and messages. As a result, we developed the xNEMO, the extension of INET, that implements the NEMO Basic Support protocol according to the specification of the IETF[2].

To perform the simulation and evaluation, we use two mobile routers MR1 and MR2 that are roaming from their home networks to new foreign networks.

[1] http://inet.omnetpp.org/

Besides, we dispose of a mobile node MN that is moving between its home network and the two mobile networks related to the two mobile routers MR2 and MR1. A second scenario is also considered. It extends the first one by adding few extra nodes in order to take into consideration relatively dense networks. Nodes are scattered in the different subnetworks. Each node is communicating wirelessly with a correspondent node exchanging a UDP data flow.

The network configuration is as follows; in both scenarios we use the IEEE 802.11b for wireless communication with 2Mbps bitrate. The mobile node is moving from its home network to the visited network with a velocity of one meter per second. We use the rectangle mobility model to simulate the MN movement from the home network to the visited network. The delays separating two routers or a hub and a correspondent node are 50ms, while, the delays separating a router and a hub or an access point are $10^{-3}\mu$s. The access points beacon interval is 0.1s. The transmission range of each access point as well as the different nodes is 177m. Figure3 shows the basic simulation network without the extra nodes with the UDP flow. To observe results during the movement of the Mobile Node as well as the Mobile Routers, simulations lasts 600 seconds.

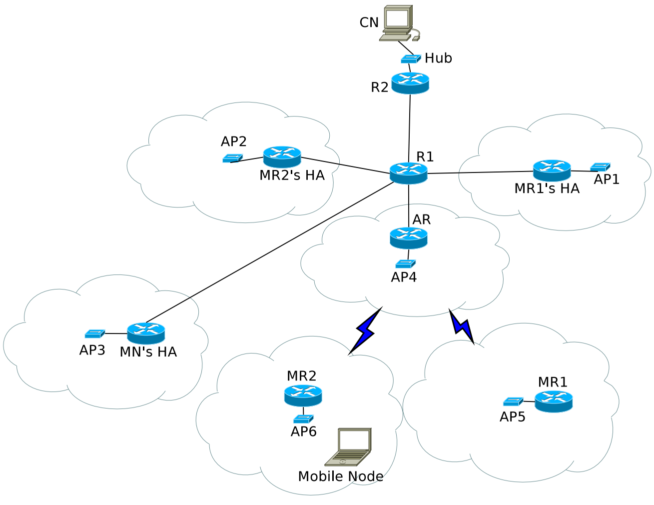

Fig. 3. Basic NEMO simulation network

4.2 Metrics

The purpose of simulations is to evaluate the impact of the explicit handover cross-layer triggers and the new behaviour of the TCP algorithm upon the reception of this cross-layer trigger on the TCP application recovery in the context of mobile networks.

TCP Application Recovery Time (ART) is an important metric we have to keep track of in order to evaluate the cross-layer trigger. We also keep track of the congestion window (cwnd) and the TCP packets sent by the mobile node during the simulations in order to evaluate the modification in the behaviour of the TCP algorithm.

Besides, the evaluation of the performance of our optimizations have been done while varying the density of our networks. Simulations are done using empty networks as well as loaded networks.

4.3 Simulation Results and Evaluation

Through Fig.5 and Fig.4 we can notice the difference between the round trip time (RTT) in the home network and the round trip time in the visited mobile network(Fig.5) and the round trip time in a simple foreign network (Fig.4), the difference is important regarding the number of tunnels in NEMO (nested tunnels) and then the number of subnetworks traveled by packets. And as the RTT is a component of the Application Recovery Time (ART)[5], that explains partially the importance of the ART in the Mobile Networks.

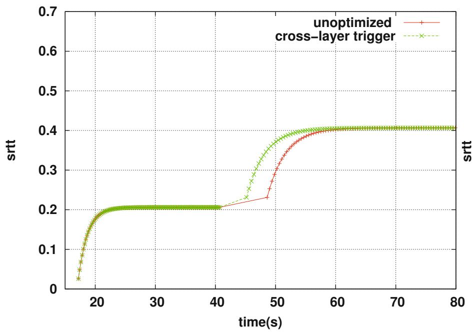

Fig. 4. TCP Smoothed RTT in MIPv6 (seconds)

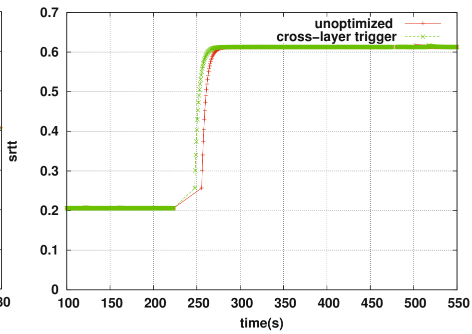

Fig. 5. TCP Smoothed RTT in NEMO (seconds)

Fig.6 and Fig.7 illustrate the evolution of the TCP congestion window size with three configurations; the red line for the unoptimized TCP, the green line is for TCP enhanced by the cross-layer triggering and the blue line is for cross-layer triggering plus modified behaviour of TCP. In all the scenarios, handover between the mobile node's home network and the first mobile network is triggered approximately at the 243rd second and the second handover between the two mobile networks is triggered at roughly the 461st second.

From Fig.6 and Fig.7, we notice that the ART when roaming between the two mobile networks is less then the ART when roaming between the mobile node's home network and the first mobile network. This due to the fact that the mobile node's home agent does not perform the Duplicate Address Detection (DAD) when receiving a Binding Update from a mobile node having already a cache entry in the cache list which is the case of our scenario and thus the BU/BA phase composing the handover latency is decremented by one second.

The figures clearly show that with the cross-layer triggering the TCP application is recovered earlier *i.e* ART is reduced especially in the case of the quite loaded network (Fig.7). Besides, adopting a modified behaviour of the TCP algorithm allows a much better recovery of the application by recovering the state variables of TCP (cwnd and ssthresh).

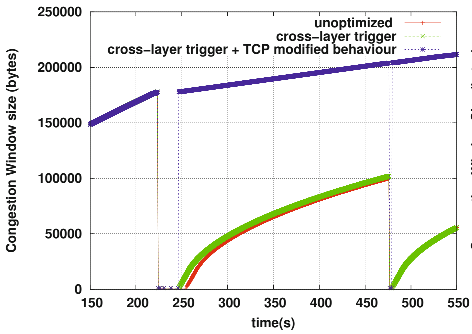

Fig. 6. cwnd size, empty cells

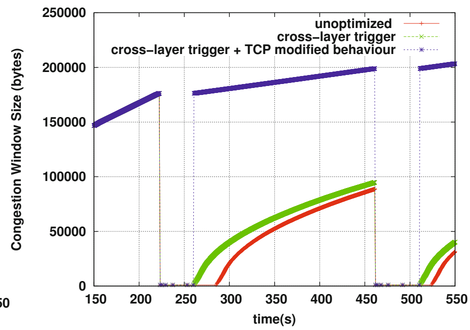

Fig. 7. cwnd size, 28 nodes,300kbps

Fig.8 and Fig.9 show the evolution of the number of the TCP packets sent by the mobile node. Both figures show that the number of packets sent increases faster when adopting the cross layer triggering and the TCP modified behaviour (the blue line).

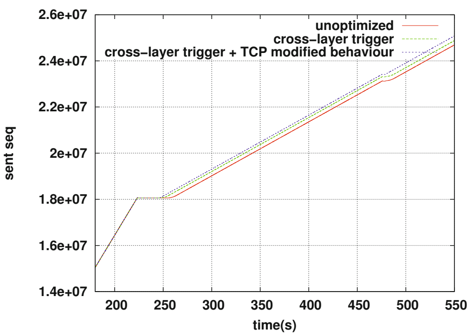

Fig. 8. TCP Sequence number, empty cell

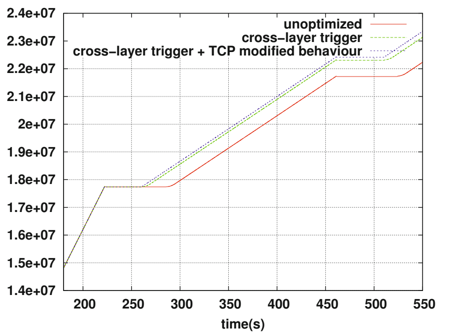

Fig. 9. TCP Sequence number, 28nodes, 300kbps

Tab.1 presents and compares the number of TCP packets sent in each simulation, it also demonstrates that the performance of the TCP application has been improved by adopting the proposed enhancements as the number of packets sent when TCP is augmented by the cross-layer triggers and the modified behaviour is much bigger (22176 packets in empty cells and 20492 packets in loaded cells) than the number of transmitted packets in case of unoptimized TCP (21703 packets in empty cells and 19403 packets in loaded cells).

Table 1. Number of TCP packets sent by the mobile node during the different NEMO simulations

	empty cells	**loaded cells**
Standard	21703	19403
cross-layer triggering	21967	20279
cross-layer triggering+ modified TCP	22176	20492

5 Conclusion

Wireless mobile networks present few challenges due to the nature of wireless links. They have to frequently change their points of attachment while moving around because of the relatively short range of transmission of access routers. This requires different steps to be taken before becoming reachable again. Hence, the interruption of data transmission becomes frequent and the time that an application recovery takes to recover is considerably larger than the handover latency. It is fair to say that this inconvenience is due to the lack of cooperation between mobility protocols and TCP. We have been interested in this problem previously and proposed a cross-layer cooperation between these protocols and TCP. Host Mobility case was studied in an earlier work while this paper treats the networks mobility case. We implemented the xNEMO extension to INET framework on OMNeT++ deploying the Network Mobility Basic Support protocol (NEMO Basic Support) according to the the specification of the IETF. Our proposed cooperation was tested using our new INET extension (xNEMO). Simulations showed that our proposal significantly reduced the Application Recovery Time in the mobile platforms context and the introduction of the modified behavior of TCP after recovery improved the performance of the TCP application. As a future work, we intend to consolidate the simulation results by moving to a broad-scale implementation with simulation tools and in a second time with a real test beds. The vehicular networks, indeed, would be an ideal canvas to test our work. In fact, they are typical mobile environments where it is plausible to consider a vehicle as a mobile network. Whether its a private car or a public transportation, NEMO Basic Support would be an exemplary mean of managing devices connection inside the vehicle.

References

1. Casetti, C., Gerla, M., Mascolo, S.: Tcp westwood: End-to-end congestion control for wired / wireless networks. Wireless Networks 8(5), 467–479 (2002), http://portal.acm.org/citation.cfm?id=582460
2. Devarapalli, V., Wakikawa, R., Petrescu, A., Thubert, P.: Network Mobility (NEMO) Basic Support Protocol. RFC 3963 (Proposed Standard) (January 2005), http://www.ietf.org/rfc/rfc3963.txt
3. Dhraief, A., Belghith, A.: An experimental investigation of the impact of mobile ipv6 handover on transport protocols. The Smart Computing Revue, KAIS 2(1) (February 2012), http://www.smartcr.org/view/view.html
4. Dhraief, A., Chedly, Z., Belghith, A.: The impact of mobile ipv6 on transport protocols an experimental investigation. In: 2010 International Conference on Communication in Wireless Environments and Ubiquitous Systems: New Challenges (ICWUS), pp. 1–8 (October 2010)
5. Dhraief, A., Souri, H., Belghith, A.: Explicit mipv6 handover cross-layer triggers and its impact on tcp application recovery time. In: 2013 International Conference on Computer Applications Technology (ICCAT), pp. 1–8. IEEE (2013)
6. Fu, C.P., Liew, S.C.: TCP Veno: TCP Enhancement for Transmission Over Wireless Access Networks. IEEE Communications 21(2), 216–228 (2003), http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.2.1469
7. Johnson, D., Perkins, C., Arkko, J.: Mobility Support in IPv6. RFC 3775 (Proposed Standard) obsoleted by RFC 6275 (June 2004), http://www.ietf.org/rfc/rfc3775.txt
8. Lee, D.C., Kim, H.J., Koh, J.Y.: Enhanced algorithm of tcp performance on handover in wireless internet networks. In: NPC, pp. 684–690 (2004)
9. Tian, Y., Xu, K., Ansari, N.: TCP in wireless environments: problems and solutions. IEEE Communications Magazine 43(3), S27–S32 (2005), http://dx.doi.org/10.1109/MCOM.2005.1404595
10. Varga, A.: OMNeT++ User Manual, Version 4.1 (2010), http://www.omnetpp.org