Mobile-PEP: satellite terminal handover preserving service continuity

A. Caponi, A. Detti, M. Luglio, C. Roseti and F. Zampognaro[1]
University of Rome "Tor Vergata"
Department of Electronics Engineering
zampognaro@ing.uniroma2.it

***Abstract*—Mobile user terminals allow to access different networks through several interfaces. Seamless communications is an essential requirement and service continuity is its main metric from user perspective. Mobile IPv6, ad-hoc routing, dynamic link layer protocol, SDN paradigm greatly facilitate mobility and network flexibility. Nonetheless, full mobility is limited by NAT routers or proxy agents, which break end-to-end semantic, as Performance Enhancing Proxies (PEPs), mandatory component over satellite networks to optimize performance. PEP spoofs TCP connections to end-users and hides connection context to the end-user control. Thus, any dynamic path change leads to the drop of the ongoing connections impairing service continuity. In this paper, we present an enhanced PEP implementation, Mobile-PEP, able to manage handovers without connection context transfer. Main operations and added value in several satellite-based operational scenarios are herein shown, leveraging on a Mobile-PEP prototype implementation.**

***Index Terms*— Mesh Networks, Satellite Networks, PEP, Handover, TCP.**

I. INTRODUCTION

THE need of ubiquitous broadband connectivity and the availability of mobile broadband access devices and mobility-oriented applications are continuously increasing. This trend is supported by the deployment of Wireless Mesh Network or even Mobile Ad-hoc NETworks (MANET) [1], aimed to provide a highly flexible and dynamic access to Internet for common mobile devices. Several initiatives [2][3] allow private xDSL customers, with a specifically wireless modem configuration, to share part of their bandwidth with the community, resulting in a distributed access ad-hoc network. Similar approaches are adopted for tactical and/or sensor networks, where a MANET is dynamically deployed on the field, while a backhaul allow connection to core networks and Internet.

In case of isolated/rural areas, where a fixed infrastructure is not economically viable, satellite systems play an important role for backhauling. Furthermore, in the recent years, broadband Satellite platforms, based on Ka-band technology in multi spot-beam configuration [4], lowered the service costs and increased throughput, coverage and availability. In such a scenario, the satellite must be integrated with Wireless Mesh Networks/MANET, with Satellite Terminals (STs) acting as core networks gateway, which are required of dynamic changes due to mobility, resulting in a ST handover (HO).

TCP connections are managed respecting end-to-end semantic. Either Layer-2 or Layer-3 mobility technology is adopted to ensure a seamless HO with no impact on Quality of Experience (QoE), but network agents manipulating end-to-end context impact the mobility management. One of such agent is the Performance Enhancement Proxies (PEPs), usually placed at the edges of the satellite links [5], with the aim of accelerating TCP performance.

PEP involve a set of techniques usually based on TCP spoofing/splitting techniques which terminate TCP connections, transparently to end-users, and establish new ones using optimized transport protocols to efficiently transport data over the long-latency satellite links [6][7]. Thus, part of the connection context is managed by the PEP, forcing TCP flows to cross it and impact on mobility procedures is straightforward. In fact during HO, cached packets already spoofed by the PEP will be not able to reach the other end of connections, so that the continuation of the TCP session through a new unaware PEP is impossible (TCP de-synchronization [8]). In addition, the new attached PEP is not able to establish a new satellite connection without spoofing the initial three-way handshaking SYN packets.

This paper deals with practical operational scenarios involving satellite and Wireless Mesh Network integration, introducing meaningful use cases and network configurations. The main objective is to investigate on possible solution allowing user mobility with several PEPs. Solutions proposed in [9][10] are considered impracticable since they require additional architectural components and protocols to perform a context exchange among nodes to seamlessly continue transmissions. In fact, the addition of additional components can make integration and actual deployment of such systems more critical. Thus, a mobile-PEP is introduced to manage the transfer TCP context among two PEPs involved in a HO, transparently to the end-device not adding explicit signaling.

II. SCENARIOS DESCRIPTION

We address a scenario where Mobile Node (MN) is a full-IP device that roams through the network and runs most popular TCP-based applications. On the other side of communication, a Correspondent Node (CN) can be assumed as an application server, which MN is connected to. One or a pool of STs acts as Wireless Mesh Network gateway to interconnect to the Internet. A star-based architecture is assumed for the satellite system, where STs are connected to a satellite hub (herein including interconnection gateway (GW) functionalities for

---

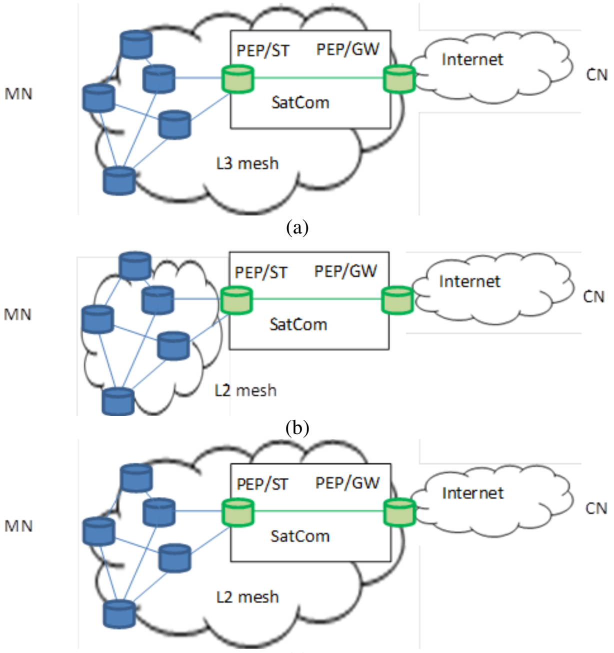
[1] Authors in alphabetical order

interfacing to terrestrial networks). Therefore, all the traffic generated from STs is gathered and managed by the Hub. In case of geostationary satellite links, installation of PEP agents up on both ST and Hub side is mandatory to optimize TCP performance.

The integration strategy is selected according to the target application scenario, system mission, area of deployment (from both extension and location point of view), performance and security requirements, etc. Two strategies at layer 3 and 2 are presented as reference. *Mobile IP* [11][12] (MIP), current standard for supporting IP mobility in wireless networks with infrastructure, enables the mobile node to access Internet and changes its access point without losing the connection. It suffers in our scenario from many drawbacks [13][14], such as high handoff delay. Many solutions have been developed to efficiently support local mobility inside IP wireless networks such as Cellular IP [15], Handoff-Aware Wireless Access Internet Infrastructure (HAWAII) [16], and Hierarchical Mobile IP (HMIP) [17]. *Batman_adv* protocol [18] as a pure Layer2 HO protocol provides signaling to determine the best route along the MANET domain.

Specifically, it is possible to define three main strategies to support mobility (Fig. 1):

a. Mesh networking at layer 3 (e.g. MIP) with MN/CN and PEP nodes actively participating to the network re-configuration during HOs;
b. Mesh networking at layer 2 for the access network only, with MN/CN unaware of the L2 underlying operations, and dynamic routing out of satellite domain;
c. Mesh networking at layer 2 with satellite PEP-enabled nodes modified to support mesh-routing and acceleration.



(a)

(b)

(c)

Fig. 1. Possible integration scenarios (GW = Satellite Hub)

III. SELECTED SCENARIO AND PROBLEM FORMULATION

We assume an application scenario similar to that in [2][3] with xDSL, but with some STs configured as mesh alternative gateways to benefit of satellite capability to cover large areas.

Depending on MN position, mesh routing can convey traffic to a particular gateway (e.g. minimizing terrestrial wireless hops), while allowing gateway HO when MN moves. Mesh networking at layer 2 scenario has been selected, previously referred as (b) in Fig. 1, since it avoids installation of mesh capabilities at the satellite Hub and represents the most flexible solution. In fact, configuration (a) would bring the problem of a handover delay due to network reconfiguration involving both MN and ST, while the configuration (c) requires modifications in satellite L2 extending to satellite Hub mesh capabilities. The latter implies ad-hoc agreements between local network manager and satellite operator, which result critical from both technical (additional signaling over satellite) and commercial point of view.

Therefore, selecting configuration (b), the only modifications involve PEP agents, which run on top of both ST and Hub PEPs: PEP is not required to be aware of layer 2 routing, it must be simply “ready” to accept and properly process TCP packets not referred to any flow previously “spoofed”. Nevertheless, this configuration requires modifications on the underlying L2 mesh protocol (*Batman-adv* is considered [18]) to manage multiple alternative Internet gateways (several ST are possible during handover).

The selection of a L2 mesh setup allows MN to keep IP address over time, leaving to the ARP L2 resolution the task to update information about MN point of attachment to Wireless Mesh Network/MANET. Therefore, MN and ST-PEP pool lie on the same local domain, as connected directly through a “smart” switch. Furthermore, each PEP is integrated to the corresponding ST, which in turn has its own IP subnetting according to the SatCom operator configuration. The detailed scenario is represented in Fig. 2.

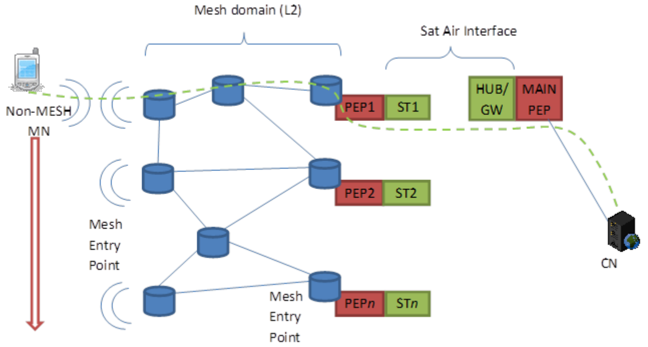


Fig. 2. Detail of the selected scenario

An excerpt from [9] states that: “*In cases where there is a change in subnet, IP packet delivery can be optimized if context (e.g., change in routing information) from the old Access Router (AR) to the target AR is transferred […]*”. In our configuration, this would imply the explicit transfer of TCP context between PEPs involved in the HO. In fact, the new PEP is completely unaware of the status of TCP connection managed by the old PEP. On the other hand, PEP installed on the satellite GW/Hub must be able to setup and manage two simultaneous connections with both old and new PEP, referred to the same end-to-end TCP connection between

MN and CN, during HO operations. This approach is considered a disadvantage, since PEP agents need to implement explicit signaling during a HO process, in order to preserve end-to-end transfer, requiring changes on PEP architectures as well as the definition of context synchronization distributed methods.

To easily integrate satellite and Wireless Mesh Network, the scheme proposed in this work instead relies on PEP operations with an implicit TCP context derivation HO. We can only leverage on the star-architecture assumed for satellite network, which implies that PEP on the Hub side can overview and manage all pending connections with the various ST PEPs.

IV. PROPOSED PEP SOLUTION

A. *Requirements and Design*

Keeping common PEP operations, the design of the mobility PEP support was based on these requirements:

- Send and receive TCP/IP packets out of any tracked (previously spoofed) connection, leveraging on the mesh Layer 2 methods for the reachability of MN;
- Manage possible multi-path satellite connections referred to a single end-to-end transfer, avoiding unnecessary re-transmissions and session tear-down;
- Keep the acceleration of TCP connections over satellite interface (splitting/spoofing) also in case of handover.

The proposed solution is based on a symmetric approach for distributed PEPs, with a set of functions triggered according to the state of the connections and the direction of traffic.

This approach allows to define a general framework to be tailored according to specific needs of the target application scenario. We assume a pool of *N* PEP agents interfaced to the Wireless Mesh Network. Each PEP can receive packets generated by the MN either from the beginning of the transfer, or in the middle of an end-to-end transfer. Dynamics of L2 switching changes are out of the scope of this work, although their characteristics have been verified through a real-time demonstrator based on CORE emulator [19].

The proposed *Mobile-PEP* is designed to work at the edges of the satellite link and to process TCP traffic only, as any normal PEP. The *Mobile-PEP* foresees an internal connection manager able to intercept packets passing through, triggering actions accordingly. From MN perspective, TCP connections must be established and successfully terminated transparently, without any relevant impact from *Mobile-PEP* actions.

B. *Mobile-PEP Implementation*

A *connection manager* on each *Mobile-PEP* instance is activated upon any new TCP flow detection, either a new established connection (interception of SYN/SYN-ACK packets) or already established connection (handed over from another PEP). *Connection manager* creates a new entry in its local table related to the *TCP context* information extracted from the processed packet. When receiving a *new unhandled packet*, it computes/collects the following parameters:

- **Connection signature** a hash value to identify the connection using as inputs Source/Destination IP and ports; such a signature allows to associate each end-to-end TCP connection to the PEP-to-PEP ad-hoc tunnel;
- **Time stamp** necessary to store its initial value to generate following spoofed packets; otherwise, if not correct, the end-system kernel discards spoofed packets;
- **IP address of corresponding Mobile-PEP endpoint** is the IP of the PEP on the other side of satellite link, which a tunnel must be established with; for Mobile-PEP at the ST side, it will be always the satellite Hub IP.

Then, the IP packet is forwarded unaltered in case of SYN/FIN-flag enabled, independently from input interface, or when coming from terrestrial interface. To opposite, new packets received from satellite interface are dropped.

With reference to state-machine illustrated in Fig. 3, upon the reception of a *new unhandled packet*, Mobile-PEP exits from IDLE state starting TUNNEL OPEN procedure with the corresponding Mobile-PEP end-point and then goes in TUNNEL OPEN CONFIRM. Tunnel establishment ends when an OK feedback is received from the corresponding Mobile-PEP. Thus, a PEP-to-PEP socket associated to the *connection signature* is up and ready to be exploited for data transport over satellite link. *Connection signature* – Satellite *Socket ID* association terminates setup phase and Mobile-PEP enters in ESTABLISHED phase, where PEP operations on the packets are performed. Together with *Spoofing*, Mobile-PEP performs *Caching* and *Handover (HO)-handling* operations.

<u>Spoofing</u>

Packets received from terrestrial connections are buffered in a local *cache* and acknowledged through "Spoofed ACK" locally generated. Cached packets are then processed over the proper satellite tunnel with Socket ID associated to the connection signature (e.g. using a TCP-based protocol optimized for satellite). Packets received from satellite tunnel, instead, are forwarded to terrestrial interface after restoring TCP/IP headers on the basis of connection signature info.

<u>Caching</u>

Mobile-PEP operations rely on the management of a local *cache*, which in turn leverages on two pointers:

- **Last SN sent:** the Sequence Number (SN) of the last packet over satellite tunnel; all cached packets with a higher SN are eligible for transmission;
- **Last SN ACKed**: The last packet acknowledged over satellite tunnel; all packets in the cache with lower SN can be definitively deleted.

All these parameters are managed and updated by the *connection manager* on a per connection basis. When receiving, Mobile-PEP is in charge also to manage re-ordering of packets arriving from different satellite links during handovers.

<u>HO Handling</u>

When a Mobile-PEP receives a TUNNEL OPEN request referred to an already managed connection signature (a tunnel with another Mobile-PEP is up), its *connection manager* performs the following HO procedure:

1. sets up the tunnel with the "new" requiring Mobile-PEP;

2. registers SN of the first packet received from the "new" Mobile-PEP, matching target *connection signature*;
3. waits for reception of packets matching target connection signature from two parallel tunnels: up to (SN-1) from tunnel with the "old" Mobile-PEP, and from SN and forth from tunnel with the "new" Mobile PEP;
4. when receives (SN-1) packets from "old" mobile-PEP, it forces TUNNEL CLOSE through a FIN/RST message;
5. continues transfer with the "new" Mobile-PEP only.

When connection FIN packets are detected, Mobile-PEP enters in TUNNEL CLOSE, where tunnel is terminated and corresponding entry in the connection manager table is deleted. FIN packets are forwarded transparently in order to trigger the correct closing of the end-to-end connection.

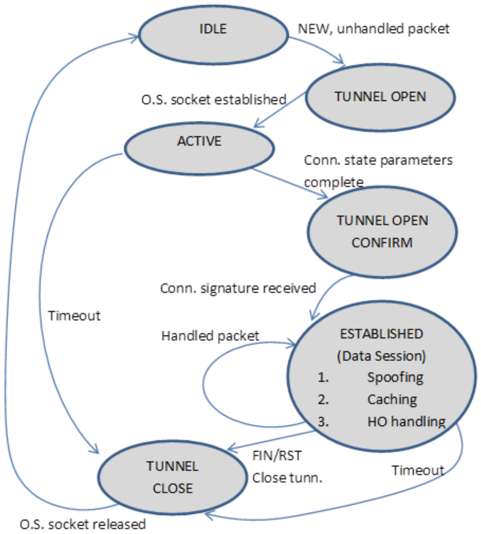


Fig. 3. Mobile-PEP state machine

V. DEVELOPMENT TESTBEDS

To develop and validate the proposed *Mobile-PEP* in the target scenario, two software tools were used: i) a Linux based implementation for Mobile-PEP development and feature testing ii) a *Batman adv* based network running on CORE [19] emulator to assess HO procedures with realistic timings and routing dynamics. Using i), a Mobile-PEP prototype has been implemented in Python. Three Mobile-PEP instances, two at the ST side and one at the satellite Hub side, are run over different Virtual Machines, connected to a satellite DVB-RCS-compliant emulator (SNEP [20]). An application machine can route its traffic to one of the two ST Mobile-PEPs towards a sink behind Hub Mobile PEP. A configuration file defines scheduling for routing changes from application machine to one of the two ST Mobile-PEP.

To validate the integration of Mesh networking with satellite links using the L2-mesh configuration, the additional setup ii) was extensively used, adopting a modified version of Batman-adv over CORE emulator [19].

A screenshot related to the adopted core setup is shown in Fig. 5. As CORE simulation output, HO timings for different MN mobility patterns and Wireless Mesh Network geometry setup are achieved.

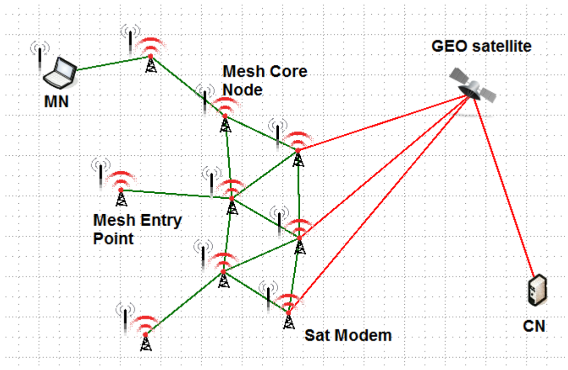


Fig. 5. Core testbed configuration

VI. TESTS AND RESULTS

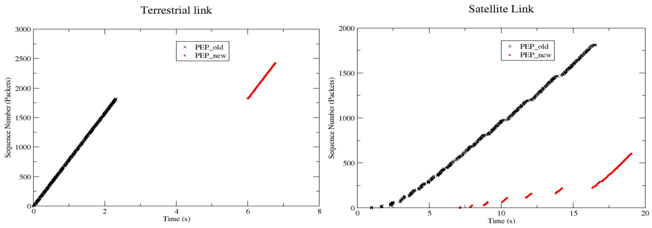
Main results concern the validation of functionalities envisaged in the proposed Mobile-PEP to support seamless TCP HOs. First, the mesh scenario has been realized adopting CORE emulator [19], adapted with *Batman-adv* integration to support MN HO on different satellite gateways. The first goal was to prove feasibility of a multi-satellite gateway configuration using *batman-adv* protocol. Since Internet architecture relies on a clean network layer separation, to be compliant with such a principle, default gateway configuration is disabled by default in *batman-adv*, and when enabled it operates on top of DHCP. In the configuration of Fig. 1 (c), L2 mesh is spread on the whole satellite domain, and the only default L2 gateway is set behind satellite GW/Hub. Thus, routing towards one of the possible STs is completely managed through L2 signaling without triggering any L3 mechanism. To exploit these benefits, while still using Fig. 1 (b) configuration, a L2/L3 "virtual" satellite gateway is registered as next-hop on all L2 ST nodes. Actually, the latter responds on behalf of "virtual gateway" indiscriminately and best route achievement completely relies on *batman-adv* signaling.

Using the topology in Fig. 5 and with default values, a HO is performed within of a very large time-range, about 2-10 s, depending on different factors: MN mobility pattern, number of L2 mesh nodes, mesh topology, *batman-adv* default timers for signaling, etc. Once demonstrated the HO scenario consistency, reproducing dynamic ST changes, our primary goal was to test all the basic Mobile-PEP features to validate feasibility of L4 HO operations. Thus, in the rest of the tests with the Mobile-PEP prototype on the Linux testbed, a fixed HO time compatible with the previous tests is considered (about 5 s) and it is triggered on demand. Table I summarizes the validation result of the main features required to enable the Mobile-PEP context-less mobility. The Mobile-PEP prototype has been fruitfully used to perform some preliminary tests to characterize TCP flow level dynamics during HO. TCP-based application data is generated by *iperf* software. The rationale was to assess benefits coming from the distributed management of two Mobile-PEP sub-connections versus the stop-and-go effecting an end-to-end TCP connection during a HO. With previous PEP implementations, the loss of connectivity towards the other PEP causes a connection failure. Thus file transfer must restart from scratch once connection with a new PEP is established.

TABLE I
VALIDATION TESTS

Feature Test	Result and comments
Intercept of IP packets in transit	Ok – The library used was NFQUEUE packet interception wrapper for Python [21]
Break of end-to-end connection for processing packets	Ok – The NFQUEUE library allows to work on packet-per-packet basis and each packet can be intercepted for processing.
Spoofed ACKs and fake TCP packets	Ok – The used library is SCAPY, allowing RAW packets generation in Python [22].
Establishment of a dedicated tunnel for PEP-PEP data	Ok – On the tunnels a specific session protocol with data transfer was designed.
Recover of connection state on an unaware Mobile-PEP$_{new}$	Ok – The state variables defined can be recovered by new Mobile-PEP through inspection of unhandled TCP packets. New Mobile-PEP recovers operations normally.
Cache of outgoing packets at the PEP for acceleration	Partial – This feature is being implemented with limited cache size to perform back-pressure TCP flow control (advertised window). In addition, re-ordering is not performed yet.
Mobile-PEP transparency for both MN and CN.	Ok – Application running during handover continues operations normally and avoiding loss of previously received data.
Correct TCP termination	Partial – Since accurate re-ordering during HO is not performed yet, TCP-based applications are not closing successfully

A real packet dump has been collected and post-processed for the generation of a byte-transfer (Sequence Number) vs time plot with the proposed implementation. HO management with Mobile-PEP is summarized with plots in Fig. 6. Fig. 6 (a) shows the sequence number evolution of packets as received from Mobile-PEP terrestrial interfaces. The first set of points represents bytes received by the first Mobile-PEP, namely "old". Then, L2 HO conveys all remaining packets to the second Mobile-PEP ("new"). As observed, MN continuously transmits packets just experiencing a gap in the ACK reception during L2 handover operations. Note that with traditional PEP, after 2.5 s, MN would wait for timing out before forcing connection termination. Fig. 6 (b) reports sequence number of packets transmitted over Mobile-PEP satellite interfaces. Data transfer is split on the two Mobile-PEPs (on the ST side) involved in the HO. When traffic is forwarded on the new Mobile-PEP, sequence number restarts from 0, since referring to a new connection established and managed by the "new" Mobile-PEP.



(a) (b)

Fig. 6. Results of PEP handover on a TCP session

In addition, there is evidence that for a relevant time interval Mobile-PEP at the satellite Hub side receives packets from two different tunnels with two corresponding STs. This is because, at the time traffic is routed towards the new Mobile-PEP, the old one has still cached packets to be processed. A kind of distributed MP-TCP (Multi Path TCP) management of end-to-end connections is observed during Mobile PEP handovers.

VII. CONCLUSION

This paper addresses the problems and introduces a possible solution for mobility support of PEPs in a mesh network setup. PEPs are important components of a satellite link, and existing implementations are not able to keep transport protocol sessions during handover. A novel PEP architecture is introduced, detailed (with support of dedicated testbeds) and the preliminary results proved its applicability and correct functioning in the reference scenarios. Future works will address complete Mobile-PEP implementation on Linux, that will allow to perform additional performance-oriented test.

REFERENCES

[1] MANETS (Mobile ad-hoc network) IETF Working Group, http://datatracker.ietf.org/wg/manet/charter/ last accessed Feb 2015.

[2] FON community mesh initiative, https://corp.fon.com/en and http://boingboing.net/2009/04/21/fon-releases-open-me.html.

[3] S.Proto, A. Ciruleo et al., Ninux Initiative – Routing Architecture Document, "Ninux Roma, The Routing Architecture", v.0, May 2012.

[4] Fenech, Amos, Tomatis, Soumpholphkakdy, KA-SAT and future HTS systems, IEEE 14th Vacuum Electronics Conference (IVEC), 2013.

[5] Interoperable PEP (I-PEP) Transport Extensions and Session Framework for Satellite Communications: "Air Interface Specification", Oct. 2005.

[6] M. Luglio, C. Roseti, F. Zampognaro, Performance evaluation of TCP-based applications over DVB-RCS DAMA schemes, JSCN, 2009.

[7] M. Luglio, C. Roseti, F. Zampognaro, Analysis and Performance evaluation of a burst-based TCP for Satellite DVB RCS links, on IEEE/ACM Transactions on Networking, Vol. 18, Issue 3, Jun. 2010.

[8] F. Arnal, T. Gayraud, C. Baudoin, and B. Jacquemin, "IP Mobility and Its Impact on Satellite Networking", 4th Advanced Satellite Mobile Systems, pp. 94 –99, 2008.

[9] IEEE Std 802.21 - IEEE Standard for Local and metropolitan area networks — Part 21: Media Independent Handover Services, 2013.

[10] E. Dubois; J. Fassony; C. Donny; E. Chaputy; "Enhancing TCP based communications in mobile satellite scenarios: TCP PEPs issues and solutions", 5th ASMS Conference, 2010.

[11] D. Johnson, C. Perkins, and J. Arkko, "Mobility Support in IPv6," RFC 3775, June 2004.

[12] C. Perkins, "IP Mobility Support for IPv4," IETF RFC 3344, 2002.

[13] A. T. Campbell and Javirer, "IP Micro-Mobility Protocols," ACM SIGMOBILE Mobile Computing and Commun. Review, ACM Press, vol. 4, issue 4, Oct. 2000, pp. 45–53 .

[14] N. A. Fikouras et al., "Performance of TCP and UDP during Mobile IP Handoffs in Single-Agent Subnetwork," Proc. Wire- less Commun. and Net. Conf., 1999 (WCNC), New Orleans, LA, Sept. 1999.

[15] A. Valko, "Cellular IP: A New Approach to Internet Host Mobility," ACM SIGCOMM Comp. Commun. Rev., vol. 29, no. 1, Jan. 1999.

[16] R. Ramjee et al., "HAWAII: a Domain-based Approach for Supporting Mobility in Wide-area Wireless Networks," IEEE/ACM Trans. Networking., June 2002, vol. 10, pp. 396–410.

[17] E. Fogelstroem, A. Jonsson, and C. Perkins, "Mobile IPv4 Regional Registration," IETF Internet-Draft, v 03, Aug. 22, 2006.

[18] Batman-adv, Open Source project Web page -Layer 2 Mesh networking: http://www.open-mesh.org/projects/batman-adv last accessed Feb 2015.

[19] Core Emulator, Open source Networking Emulator, http://www.nrl.navy.mil/itd/ncs/products/core last accessed Feb 2015.

[20] F. Belli, M. Luglio, C. Roseti and F. Zampognaro, Evaluation of TCP performance over emulated DVB-RCS scenario with multiple RCSTs, IEEE IWSSC 10th -11th Sep. 2009, Siena, Italy.

[21] NFQUEUE Pyhton library wrapper, https://pypi.python.org/pypi/NetfilterQueue/0.3, last accessed Feb 2015.

[22] SCAPY Python IP packet handling, http://www.secdev.org/projects/scapy/, last accessed Feb 2015.