Which Transport Protocol for Hybrid Terrestrial and Satellite Systems?

Ihsane Tou, Pascal Berthou, Thierry Gayraud, Fabrice Planchou, Valentin Kretzschmar, Emmanuel Dubois, Patrick Gélard

Which Transport Protocol for Hybrid Terrestrial and Satellite Systems?

Ihsane Tou[1,2], Pascal Berthou[1,2], Thierry Gayraud[1,2],
Fabrice Planchou[3], Valentin Kretzschmar[3],
Emmanuel Dubois[4], Patrick Gelard[4]

[1] LAAS-CNRS, 7, Avenue du Colonel Roche, 31077 Toulouse, France
[2] Université de Toulouse; UPS, INSA, INP, ISAE; LAAS; F-31077 Toulouse, France
[3] EADS Astrium, 31 rue des Cosmonautes, Z.I. du Palays, F-31402 Toulouse Cedex 4, France
[4] CNES, Centre National d'Etudes Spatiales 18 Av. Edouard Belin, 31400, Toulouse, France
{firstname.lastname}@{laas|astrium|cnes}.fr,

Abstract. Satellite systems will complement terrestrial networks where the network could not be deployed for technical or economical reasons. Moreover, the natural broadcasting capacity of satellite networks makes it a good companion to terrestrial networks. Then, future services will be deployed over networks that combine terrestrial and satellite systems. The infrastructure heterogeneity could be problematic, for instance because of the delays variety. This article presents the problem from the point of view of the transport layer, the layer directly connected to the application, and compares several solutions to help future service developers using such network configuration.

Keywords: TCP, Hybrid networks, satellite, PEP.

1 Introduction

During the last decade a wide range of network access technologies have been developed to extend the access to the Internet services. In parallel the cellular networks, originally designed for voice/telephony mobile services have evolved to offer more services, and in particular Internet access. The convergence of fixed and mobile services has been achieved and standardized. Also, there has been significant progress on the level of the terminal handset (mobile phone, Smartphone and laptop) whose size has been significantly reduced, while providing more capabilities and wireless interface support.

With NGN and 4G architectures, services or applications are designed independently of the underlying access network (wireless, cellular, wired, optical, etc) based on the IP core technology, which is the convergent corner-stone of telephony and data services. The always-on paradigm is conceived as a generalized mobility for user services, allowing seamless service switching across any compliant network access technology.

This means that applications (thus the underlying transport protocols) have to be persistent to the network switching. This is a challenging objective since the access media are heterogeneous and potentially operated by various actors.

This paper exposes the problems met by the transport layer and especially TCP with such heterogeneity and compares several solutions to help future service developers using hybrid networks. The performances of the TCP protocol are compared according to different TCP versions, in the specific case of hybrid handover.

However, just before, the next section presents several scenarios envisaged for the integration of satellite access systems in the future LTE networks, what we call hybrid satellite and terrestrial networks.

2 Hybrid terrestrial and satellite systems

The satellite integration with terrestrial network can be achieved in several manners. Three complementary visions are presented. A **tightly coupled architecture**, an integrated approach in which a given mobile system (3G, LTE, WIMAX) is extended to support the satellite media, as an alternate radio access channel for the mobile user, in a completely transparent way. A "**relay integration",** in which the satellite is integrated within the mobile network infrastructure, not directly at the mobile air interface, but beyond a specific satellite relay enabling the access to the core mobile infrastructure. And a "**loosely coupled integration**", in which a satellite specific interface is added on the mobile terminal enabling the connection to IP data network, through access network. This involves multi-technology (multimodal) mobile terminals, handling several interfaces and running the convenient protocol (specific media access: DVB-RCS+M for instance).

The first two architectures hide the network heterogeneity with a dedicated layer 2 transparent for the upper layers. In the tightly coupled vision, the terminal can be aware of the satellite access; in the relay architecture the heterogeneity is embedded in the network. The loosely coupled architecture needs a third protocol to assure the network change, mobile IP would be the solution, however the network characteristics will be hidden too. Also the heterogeneity is a problem for the transport layer, it appears that an agnostic transport protocol (i.e. it does no know the underlying networks) should be a better solution as the knowledge of the underlying network is a risky hypothesis.

2.2. Impact on the transport layer

The hybridization of the networks may cause several troubles with the transport protocols. The firsts occurs when changing network (i.e. leaving a terrestrial network for a satellite network), the others when the transport protocol uses old parameters (Congestion WiNDow size, timers) on a new network (i.e. a TCP connection with its parameters set according to a terrestrial network uses a satellite network with a limited bandwidth and a long delay).

The handover model: According to the network specifications and the reception conditions, the handover between two networks can generate service interruption or not. It can vary from several packet losses (soft handover), to short interruption

(several seconds), to a long break with network addressing change (mobile IP should resolve this problem). The behavior of the transport layer will depend on this. In this paper we consider seamless handover and short interruptions that could be implemented using tighten or relay architecture. We exclude mobile IP scenarios as we previously show that such handover in a satellite system is too long for the transport layer (TCP connection reset) [1].

We call the firsts scenarios "make before break" and the last one "break before make" as the adaptation is made after the handover. In the first cases the new layer 2 is set-up before the old one is deactivated. Two links are available during a short period of time when the hand-over arises. This can provoke a disordered reception of packets and then trigger TCP congestion control algorithms. On the opposite, in the last case, the new link is established only when the old is down. The effects are a risk of disconnection if the disruption is too long (application or TCP timeout), the losses could be interpreted as a congestion an trigger the TCP congestion control algorithms, or an expiration of the TCP retransmission timeout (RTO) that will starve the connection throughput.

Impact of the parameters variation: Its obvious that delay, bandwidth and in smaller proportion the loss rate will vary according to the connected network. The transport layer "guess" these parameters through *timers* for the delay, *congestion windows* for the bandwidth. Loss rate is slightly different, as it will impact the reaction of the protocol. When changing network, the transport protocol could be duped by the old parameters and do not adapt properly to the new network. Considering TCP, the useless retransmissions could increase because of a to small value of the RTO, or spurious RTO may occur as side effect. It could increase the unused bandwidth (i.e. waste of resources): TCP may not be able to use the increased bandwidth of new network or in the opposite provoke a real congestion if the congestion window is too big for the network capacity.

Many TCP versions have been proposed to the IETF to adapt to a particular network. Some are general (as TCP options), while others are designed to change a particular TCP behavior, or tends to adapt TCP functioning in the conditions of a given media technology (as, "Wireless context"), and some are completely specific to the vertical handover purpose. In this paper we focuses on the hybridization problematic, but in the following part we briefly summarize the classical TCP versions to compare it when testes in a hybrid context.

3 Transport protocol alternatives

TCP is a byte-stream connection oriented protocol. The connection management is handled by the sole endpoints. TCP uses a sliding-window based congestion control scheme: the size of the transmit window is the lesser of the receiver's advertised window size *awnd* and the sender's congestion window size *cwnd*.

The size of *cwnd* is determined by well-known algorithms designed to limit the effects of congestion in the Internet, including slow start, congestion avoidance, fast retransmit, and fast recovery.

The main element that differentiates the TCP implementations is the management of the size of its emission window. Many different approaches have been proposed, and published at IETF, without reaching the standard status. Therefore each operating system may use its own, as preferred release. This section presents a brief overview of different versions that can be interesting because they are well spread and therefore efficient or because they are adapted to a specific domain of application like satellites, mobility or high-bandwidth media.

Reno/New Reno: TCP Reno uses a loss-based congestion control window. It uses the TCP's mechanism of increasing the cwnd by one segment per RTT for each received ACK and halving the cwnd for each loss event per RTT. It uses all well-known congestion control. The TCP New Reno improves retransmission during the fast recovery phase of TCP Reno.
Compound: Compound TCP (CTCP) is designed to fit quickly to the bandwidth available while staying TCP friendly. Its particularity is to be both delay-based and loss-based. It manages two cwnds*;* the classic one like in TCP Reno and the delay based one dwnd which is used only during congestion avoidance phase.
CTCP is currently implemented over Windows Vista, Seven and Server 2008.
Cubic: The Window is a cubic function of time since the last congestion event, with the inflection point set to the window prior to the event
Cubic does not rely on the receipt of ACKs to increase the window size. Cubic's window size is dependent only on the last congestion event.
Cubic TCP is implemented and used by default in Linux kernels 2.6.19 and above.
Hybla: TCP Hybla is a TCP enhancement conceived with the primary aim of counteracting the performance deterioration caused by the long RTTs typical of satellite connections. It adopts the Hoe's channel bandwidth estimate, timestamps and packet spacing techniques.

Fortunately a number of TCP extensions have been proposed at IETF to improve TCP behavior. Some are general (as TCP options), while others are designed to change a particular TCP behavior or tends to adapt TCP functioning in the conditions of a given media technology (e.g. TCP Sack, Quick-Start), and some are completely specific to the vertical handover purpose (Freeze TCP [2], Fast Adaptation Mechanism [3]).

In the following section, we compare the TCP performances of different real TCP stacks during a handover in a hybrid network.

4 Experimental test bed description

The testbed relies on a satellite emulator (SATEM) developed by ASTRIUM that emulates the handover between satellite and wired network by selecting the available bandwidth, the propagation delay and the Packet Loss Ratio (PLR); This emulator is based on the layer 3 linux network emulator (netem). Client and Server (iperf), a network sniffer (Wireshark) and a TCP proxy (PEPsal) allow to generate, capture and produce the results. The Figure 1 shows a detailed layout of our testbed.

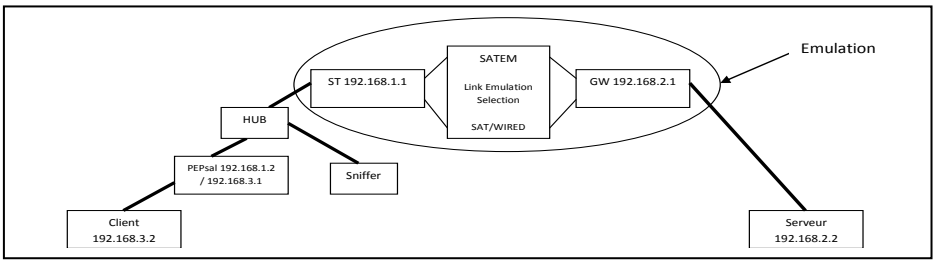

Figure 1 Testbed description

A Performance Enhancing Proxy as been added to be compared to end-to-end solutions. PEPsal [4] is an integrated PEP scheme based on the TCP-splitting technique that adopts TCP Hybla.

The following metrics has been chosen to evaluate the TCP performances: The *sequence number* evolution of the transmitted packets is useful to monitor the connection efficiency and regularity. It also tells information about packet loss, connection cuts and the amount of data transferred. The comparison of the sequence number evolution is interesting because it can be observed independently of the TCP version used. The *Congestion Window size* (cwnd), the *application end-to-end delay* and *throughput* where also used but not presented in this paper. Theses metrics have been captured and processed using the WEB100 Project and TCPTRACE [5].

We assume the satellite and the wired network parameters on emulated SATEM as:

- Emulated satellite link: Bandwidth 512 Kbps, RTT 500 ms, P.LR. 10-e5
- Emulated wired link: bandwidth 2 Mbps, RTT 50 ms and free P.L.R.

We choose to present first results according the variation of only one parameter at one time, to focus on the problems it raise.

TCP versions over satellite comparison

We first choose to compare the TCP versions over the satellite part, even if it has been extensively studied, has the performances of Windows Seven Compound version are surprising.

The sequence number evolution graph (a) shows similar performances between hybla, cubic and compound, however compound if more aggressive and benefits rapidly of the available bandwidth without undergoing losses as hybla or cubic. The throughput evolution (b) confirms a better behavior of compound and shows that after 1 minute, the tail of the hybla and cubic curve is the result of large buffers flushing and retransmission management that is not a positive aspect for the application.

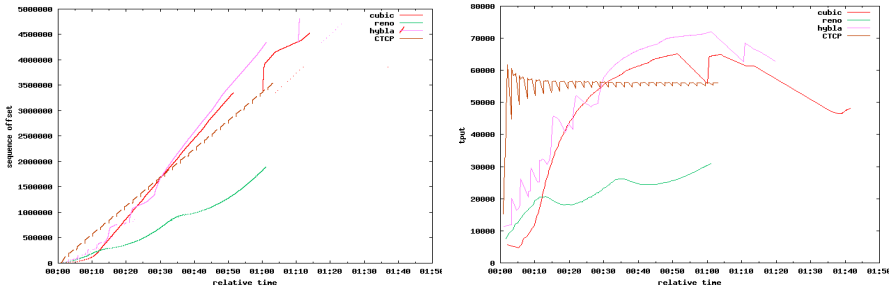

Figure 2 - TCP versions comparison (a) Sequence number (b) Throughput

TCP versions affording soft handover

Our experiences focus on changing one parameter during a handover as described above to see the influence of each parameter during the mobility processes.

Delay variation

The figure 3 (a) shows the evolution of sequence numbers with two handovers. The bandwidth has been set to 2 Mbps during the entire test (60s) and the propagation delay is changing (RTT ~ 500 ms in satellite to 50 ms the wired network). 3 phases have been defined, between 0-20 seconds, TCP streams through the satellite network, between 20-40 seconds, they cross the wired network and finally between 40-60 seconds they go back to the satellite network.

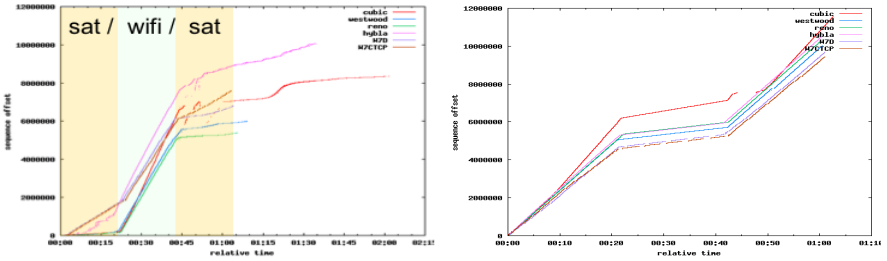

Figure 3 - TCP sequence number evolution with delay (a) and bandwidth (b) variation

In Figure 3 (a), as shown previously, hybla, cubic, compound perform better on the satellite part than reference TCP versions (westwood, reno). During the WiFi phase, no significant difference can be observed. The main difference resides when coming back to the satellite network. Cubic affords a bad congestion windows setting and generates losses, hybla too but in smaller proportions. Compound restarts faster and rapidly catches up its delay. The compound dual windows management proves to be effective after a handover.

Bandwidth variation

The Figure 3(b) shows the evolution of sequence numbers with two handovers. We set the RTT to 50 ms in the entire test (60s) and vary the available bandwidth (512

Kbps in satellite to 2 Mbps the wired network). The flows undergo 3 phases first they through the wired network, then they cross the satellite network and finally they go back to the wired network.

All streams have a quite similar behavior; expect Cubic which suffers at the end of the phase 2. The buffers are over and the cut at the second 43 is a consequence of the aggressiveness of the Cubic mechanism with low bandwidth. Cubic tries to growth up a maximum bandwidth.

PLR variation

The variation of the Packet Loss Ratio did not bring significant results, as the reaction of TCP versions should be comparable when affording a loss (considered as a congestion). Cubic obtained the worsts results because of its aggressiveness.

PEP or not

One of the main question when dealing with specific medium (i.e. satellite) is does we need a PEP or not. Considering our previous result with the new TCP versions over satellite systems, we try to show in this part that PEP are no more mandatory in such configuration.

PEP with satellite system

The figure 4 presents the evolution of sequence number of 6 TCP versions through the satellite network. It is clear that the PEPsal solutions are better than end-to-end TCP (in brown is Reno). We notice the exemplary behavior of CTCP (without PEP), which remains stable, little bit under but holds the comparison with the PEP solutions.

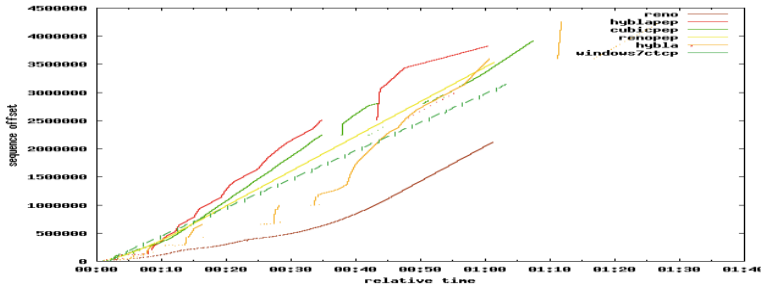

Figure 4 - Sequence number evolution with PEP influence over a satellite

PEP with mobility

The Figure 4 shows the evolution of sequence numbers with two handovers. In the graph (a), we fix the bandwidth to 2 Mbps during the entire test (60s) and vary the propagation delay (RTT ~ 500 ms and 50 ms). In the second graph (b), we fix the RTT to 50 ms during the entire test (60s) and vary the available bandwidth (512 Kbps in satellite to 2 Mbps in the wired network).

The 3 handovers are defined as previously, from satellite to wired network to satellite.

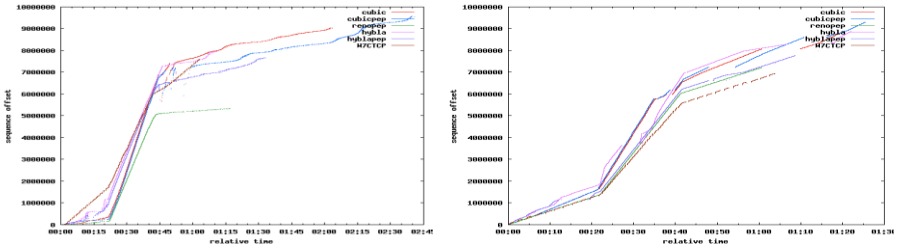

Figure 4 - Comparison between PEP TCP and e2e TCP during a handover (delay (a) and bandwidth (b) variation)

The graph (a) clearly shows the superiority of CTCP in Phase 1 and 3. PEPsal has a good result in both phases but with a CTCP advantage. During the transition to the wired network, all flows catch up from the bad performance (especially Cubic), because of the short propagation delay.

In the graph (b), the performance is quite identical between the solutions. A small advantage was noticed during the transition to terrestrial network for the PEPsal.

The impact of "break before make" handover

This paragraph compares the behavior of TCP versions when the handover is not error free (break before make scenario). The break was set to 500ms and 1000 ms.

Break (500/1000 ms) before make, fixed bandwidth (2Mbps) and delay variation (500-50-500 ms RTT)

The Figure 5 (a) shows the TCP connection behavior when the break takes 500ms. When the streams through the satellite link in the first phase, CTCP confirms the results we had before. The same applies to PEPsal and Hybla, which are just behind. After the first break, at second 20, the restart of all streams is at least about 5 seconds for Cubic, and Hybla PEPsal. CTCP reacts faster and recovers the transmission in about 2 seconds. We get the same conclusion at the second 40 when the second handover occurs.

In the graph (b), when the break takes 1 second, we observed the same behavior after the handover, but with more difficulties for Cubic.

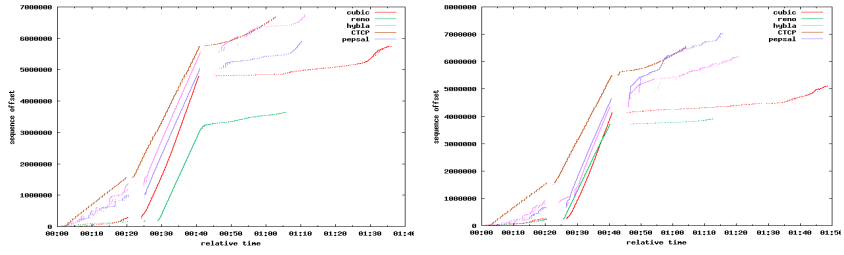

Figure 5 - Sequence number evolution with (a) 500 ms (b) 1000 ms break

Break (500/1000 ms) before make, fixed delay (50 ms) and bandwidth variation (512K-2M-512K bps)

In the Figure 6 (a), we see at the beginning of the transmission that Cubic is better than other versions, as the delay is low (but with a small bandwidth). During the two handovers, Cubic has a high cut, and restart is difficult. Hybla and PEPsal have naturally the same reaction during handovers. CTCP better manages the break, and recovers better the connection, which takes about 2 seconds.

When the break stays 1 second, all flows except CTCP, were completely broken after handover (b).

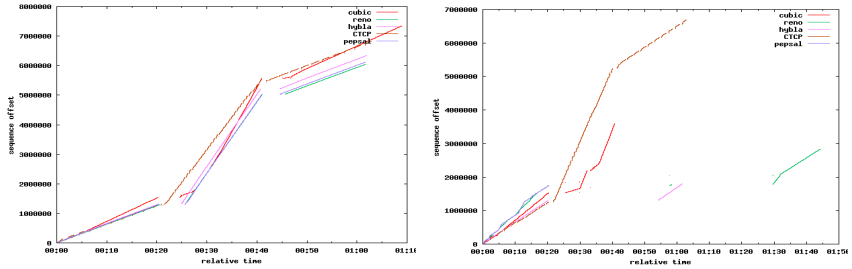

Figure 6 - Sequence number evolution with (a) 500 ms (b) 1000 ms break

6 Conclusions

This paper proposes the evaluation of different TCP versions, using real stacks, in the specific problematic of the handover between terrestrial and satellite networks. The good behavior of the transport layer is crucial to offer quality services over hybrid networks. The surprising conclusion of this study is that windows seven compound gives very good results, in some cases better than those with PEP's, especially over large propagation delays. Tests with the "break before make" showed the difficulties of some TCP restart compared to others, and attest CTCP as the best version for handovers with or without breaks.

References

[1] F. Arnal, T. Gayraud, C. Baudoin, and B. Jacquemin, "IP mobility and it's impacts on satellite networking," *ASMS*, 2008.

[2] T. Goff, J. Moronski, D.S. Phatak, and V. Gupta, "Freeze-TCP: A true end-to-end TCP enhancement mechanism for mobile environements," *INFOCOM*, 2000.

[3] D. Li, K. Sleurs, E. Vani Lil, and A. Van de Capelle, "A fast adaptation mechanism for TCP veritcal handover," *ATC*, 2008.

[4] C. Caini et al., "Analysis of TCP live experiments on a real GEO satellite testbed," *Performance Evaluation*, vol. 66, 2009.

[5] Shawn Ostermann. TCPTRACE. [Online]. http://www.tcptrace.org/