



## **On the sustainability of PEPs for satellite Internet access**

Renaud Sallantin, Emmanuel Chaput, Emmanuel Philippe Dubois, Cédric Baudoin,  
Fabrice Arnal, André-Luc Beylot

### **► To cite this version:**

Renaud Sallantin, Emmanuel Chaput, Emmanuel Philippe Dubois, Cédric Baudoin, Fabrice Arnal, et al.. On the sustainability of PEPs for satellite Internet access. 30th AIAA International Communications Satellite Systems Conference (ICSSC 2012), American Institute of Aeronautics and Astronautics (AIAA), Sep 2012, Ottawa, Canada. pp.1-8, <10.2514/6.2012-15047>. <hal-03884465>

**HAL Id: hal-03884465**

**<https://hal.science/hal-03884465v1>**

Submitted on 6 Dec 2022

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

# On the sustainability of PEPs for satellite Internet access

Renaud Sallantin\*, Emmanuel Chaput†, Emmanuel Dubois‡,  
Cédric Baudoin§, Fabrice Arnal§, André-Luc Beylot†

TCP protocols such as New Reno suffer from very poor performances in satcom. In reaction, several solutions have been deployed to improve the global performance. PEP (Proxy Enhancement Performance) is the most widely adopted. However, this improvement has a cost, and using PEPs adds new constraints to face. Recently, many works not specific to the satellite scope have been done to develop new TCP protocols leading to new TCP releases such as Compound and Cubic TCP. This article investigates the behavior of those new protocols in satellite communication. The main objective of our work is to determine if using specific satcom techniques is still a sustainable approach.

## I. Introduction

Satellite communications are an attractive way to communicate efficiently with large or unreachable areas. As a consequence, they deserve to be part of the global communication network. However, satcom systems have to deal with some very challenging and restrictive constraints impacting on different level of the network protocol stack. Among this, for the particular scope of Transport Protocol, long Round Trip Times (about 600 ms for GEO system) are strongly damaging the global behavior. Indeed, they impair directly some congestion avoidance mechanisms such as slow start that are an essential part of reliable transport protocol (e.g. TCP), and so decrease the throughput. Moreover, satellite telecommunication system is a small niche of the global communication system. A new transport protocol version dedicated to satellite is then unthinkable, especially if it could weaken the overall performance of the legacy protocol.

Considering those impairments, Performance Enhancement Proxy (PEP) solution has been developed and is currently the most widely adopted architecture for satellite links. PEPs are a pragmatic solution using a transport level proxy to separate the satellite segment from the others (notably using splitting and spoofing methods). The segment becomes transparent for the rest of the network, and so the global behavior of the transport protocol is almost unchanged for the extremities, which means better throughput and performance.<sup>4</sup> Moreover, the end-to-end implementation remains unaltered.

During the last few years, new elements became apparent which are likely to cast doubt on the PEP solutions. Thus, some limits, mostly imputable to the break of the end-to-end semantics have been discovered, and notably, that PEPs behave badly under network mobility scenarios or some security context (e.g. IPsec tunnel). Alongside this, new TCPs like Cubic and Compound have been developed for a non specific satellite use. And they seem to behave well in an hybrid scenario and already better than end-to-end specific satellite transport protocol such as TCP Hybla.<sup>14</sup>

This study aims to determine the usefulness of transport protocol PEP in two different scopes. To do that, emulation of TCP New Reno and modern TCP in an hybrid scenario with and without PEP are used. First of all, for a short-term optimization purpose, the aim is to define if modern TCPs perform well enough without PEP (throughput, fairness, ...) to question the usefulness of current PEPs. Secondly, for a sustainable purpose, this study tries to determine if PEPs can still remain performant

\*TéSa, University of Toulouse

†IRIT-ENSEEIH, University of Toulouse, 2 rue Charles Camichel, Toulouse — France

‡CNES, 18 Avenue Edouard Belin, Toulouse — France

§Thales Alenia Space, 26 avenue Champollion, Toulouse — France

despite the current and future TCPs evolutions. Indeed, TCP protocols are going to evolve and PEPs implementations will have to deal with more and more different versions. So, eventually, PEPs will be less suitable for new transport protocols than for New Reno. In this case, local interactions between PEPs and the TCP version may impact the network global behavior. The enhancement obtained thanks to PEPs would then be degraded and the question of the sustainability of the current PEPs but also of the PEP concept would occur.

The rest of the paper is organized as follows. A state of the art of PEP technology is discussed in section 2. In section 3, we described new TCP protocols and analyzed their behavior. Section 4 specifies the simulation environment and shows the result. Finally section 5 concludes the paper.

## II. Related works

PEPs systems have been developed to face the numerous constraints peculiar to the satcoms. Among them, a large RTT is probably the most damaging for standard TCP protocols such as New Reno that have been designed for wired scenarios. Indeed, slow start and congestion avoidance, their main mechanisms of bit-rate control use the reception of ack to increase their sending rate. Thus, they are RTT dependent and so very sensitive to the large RTT (around 0.6s) added by a geo-satellite segment.

### A. PEP benefits

Several notable studies are related on PEP technology. PEP is a generic term covering a wide range of protocols and mechanisms used to improve the global performances. Even if there is no common definition, all of them have for main objective to reduce the impact of the satellite segment on the global performances using an additional system transparent for the users. TCP is

To succeed in their objectives, most PEPs and notably the PEPs dealing with the TCP behavior efficiency use an architecture that splits the original end-to-end TCP connection into multiple connections. This can be done using integrated PEP or distributed PEP, but the result will be the same when strictly speaking of the RTT. Indeed, distributed PEPs and integrated PEPs are respectively a combination of a PEP server and PEP client located at each end of a satellite link and a PEP alone used at the gateway. So, in both case, the PEP located at the gateway will act as a TCP end-user and as such, will ack the received segments. Thus, the RTT used by the TCP mechanisms is reduced to the one corresponding to the path between the source and the PEP. So instead of suffering a large RTT resulting in a long slow start, the PEPs allow to fill in a faster way the whole satellite segment.

Another benefit of this splitting architecture is the isolation of the satellite segment. It allows the use of an optimized protocol adapted to this specific segment that can enable a better use of the return way or ameliorate the retransmissions.

However, even if those optimized protocols behave well when strictly speaking of performances, they are responsible of one of the system limit. Indeed, there are no common architectures or rules for the protocol used by the PEP on the satellite segment and so each constructor can build what he thinks to be the best. Finally, due to this lack of homogeneity, PEPs have to be considered as a black box on the middle of the network that can act in an unpredictable way.

### B. PEP flaws

Most of the PEPs drawbacks are due to the break of the end-to-end semantics of the transport control. The previous subsection laid the emphasis on the benefits of the splitting architecture, but using that process breaks the fundamental principle of the Transport Protocols. By doing so, it creates some incompatibility with other protocols and services from lower layers sized to deal with end-to-end properties.

Thus, the difficulty to insure the security, probably the most problematic flaw, is directly imputable to this last point. Security is a key functionality in scenario using IP-based data streams and notably in satcoms.<sup>5</sup> Indeed, satellites use broadcast to transmit the information, and so, need security to insure the confidentiality. Using PEPs, the connection is separated into two or three connections who need to exchange some informations (addresses, TCP fields...). In order to do that, PEPs need to have access to the fields contained in the IP packet and yet, this information is generally encrypted by security functionality such as IPsec. Current Ipsec does not allow the usage of PEPs. Various researchs have been done to create a couple

of updated Ipsec and trusted TCP-PEPs, but for now, results are not convincing. They are still generating an extra complexity for a weaker security.<sup>2</sup>

Complexity is an other drawback. PEP mechanisms have to be implemented and updated frequently by the operators. For example, the introduction of IPv6 will require the update of many PEPs. Moreover, and this is not only true in the peculiar scope of PEPs but for middleboxes in general, they can be a constraint in a possible extension of TCP. Indeed, PEPs are acting like a black box in the network, and so their behaviors in front of an unknown option are not easily predictable. Neither IETF nor networks operators or OS vendors can know what will or will not pass through middleboxes. Many researchs are conducted to determine if the deployment of middleboxes in the networks can be responsible for the end of the ability to extend TCP using the option fields.<sup>7</sup>

And finally, even if it is not strictly speaking a PEP drawback, the following deserves to be noted. Recent progress done to improve the BER on satellite link have made some of the main PEP improvements useless. Thus, with a BER better than  $10^{-7}$  on the return link and  $10^{-10}$  in forward link, the geo-satellite link can almost be considered as a free loss link. Mechanisms such as error recovery are not needed any more, and so, PEP benefits are deprived of one of their best asset.

### III. Modern TCP protocols

In order to face the flaws due to the use of a middlebox, a large number of works have been done to implement a TCP able to deal with a High RTT. A first orientation has been to develop a satellite specific TCP protocol such as Hybla that enables to bypass this constraint. It allows long RTT connections to achieve the same transmission rate as a small RTT reference connection, but this solution still raises some troubles. Indeed TCP protocols fit to satisfy the satcom constraints are less performant or fair on the rest of the networks than the others TCP protocols. So, considering that satellite clients are a small user niche, another alternative has been explored and the emphasis has been laid on the improvement of a generic TCP. The objective is to improve the global performance including the satellite segment, but not to damage the average performances because of the satcoms.

Firstly, some TCP options have been implemented to improve the global TCP behavior. Thus, windows scaling deals with the issue concerning segment with a large Bandwidth Data Product (BDP). It enables to increase the maximum congestion window size allowed in order to reach a better performance over satellite segment when capacity is available. An other option called ECN (Explicit Congestion Notification) is also available. Thanks to ECN, the sender is notify that increasing or keeping its current sending bit-rate will create congestion. So the sender can react by decreasing its congestion window before the congestion occurs. So the answer to the congestion is quickened and there is no wasted time in retransmission of dropped packets.

Then, new TCP releases using optimized congestion control algorithm have been developed. The following section deals with 2 of them, Cubic and Compound TCP, which are respectively the default transport protocols in various Linux and Microsoft stacks.

#### A. Congestion Control algorithm

This part analyzes and compares the behavior of New Reno, Compound<sup>11</sup> and Cubic<sup>6</sup> TCP during the different stages of the congestion algorithm. Thus, after briefly speaking of the slow start, a more accurate study will be conducted on the congestion algorithm process. The idea is not to explain each protocol but the concepts they use.

##### 1. Efficiency

The slow start, as seen in the previous part, is used to determine in a fair spirit the available rate. This step is almost the same for each TCP protocol. Some versions of Cubic use a modified slow start with 2 successive stages. But the general idea is still the same.

The main differences are concerning the congestion avoidance algorithm. TCP New Reno has a very simple congestion algorithm with only one stage. Indeed, once in congestion avoidance, the protocol increases linearly its sending rate by incrementing each RTT its congestion window. This allows a slow but fair raise which shows some strong limits in a BDP environment (cf. section II). On the contrary, being able to behave well when meeting a large BDP network is one of the purpose of the modern TCPs. Thus the key idea is

for the TCP protocol to adapt its congestion control scheme in order to fit the current state of the network when this one evolves dynamically according to the traffic and flows.

That means that when there is some available bandwidth, the protocol has to be aggressive to reach its maximum sending rate in a minimum of time. However, once the link is fully used, the protocol has to stop to be aggressive. Otherway, it would only cause some congestion or unfairness. So during a while, the protocol keeps a constant sending rate. Then, because the available bandwidth evolves in a dynamic way, the protocol should break its balance to progressively raise its sending rate in order to see if there is some new available bandwidth or not.

The following will describe the behavior of each TCP (Compound and Cubic) in the three different stages questioning their efficiency :

- The discovery state : After a loss occurred, the CWND size is decreased to prevent a congestion. Then, Cubic and Compound play aggressively to fill the available bandwidth. So during this stage, efficiency is their only concern.
- The stable state :

Both protocols define a “loss free period”. During this period, they make the assumption that there is a balance between the different flows, and so, that keeping their current rate will share efficiently the resources. In order to decide when to change from the aggressive state into the stable state, both algorithms use a different strategy. Cubic implementation uses a static one.<sup>8</sup> It waits for the congestion window to reach the size it had when the previous loss occurred. In a different way, Compound uses a dynamic threshold.<sup>12</sup> It bases its state change on the increase of the RTT. So, when sensing a significant change in the packet delay, Compound in order to avoid a congestion that would only have for consequences to damage its current bit-rate, jumps into its stable state. The end of both stable state is calculated using the different congestion avoidance function.

- The probing state :

After the stable state, both TCPs are going to probe again the network capacity. Indeed, if no congestion occurred at the end of the stable period, then either the network state stays exactly the same, or, and its the most desirable and common case, there is new available bandwidth. So, both increase their CWND. Cubic does it aggressively, and Compound uses the linear function used by New Reno in congestion avoidance.

## 2. *RTT fairness*

In opposition to TCP New Reno, Compound and Cubic use mechanisms to take into consideration the consequences of having flows with different RTTs sharing a same bottleneck link. So they use mechanisms to reach a good RTT fairness rate.

Compound is a mixed between delay-based approach and loss-based. So, by using its delay-based approach, he can be RTT independent in the Discovery phase, and still be competitive thanks to its loss-based approach. Cubic has a loss-based approach but is RTT independant too. Its window growth depends only on the time between two consecutive congestion events.

## 3. *Benefits for the satcoms*

The emphasis lies by those TCP protocols on the RTT fairness has not as main objective to improve the satcoms, but to create a fairness between flows of different RTT. Nevertheless, this enables to react quickly in case of long RTT and so doing, is an important improvement for satcoms.

The ability to reach a high rate is an other point very interesting when it is question of high BDP network.

# IV. Emulation environment and results

## A. Emulation environment

In order to obtain results closer to the reality, we used an emulation platform of a satellite link connected to real computers and TCP stacks.<sup>1</sup> Moreover, to compare the behavior of the TCP protocols with and without PEPs, distributed commercial PEPs have been used.

However, instead of helping us to be more confident about the results obtained, it urges us only to be even more careful with the interpretations. Indeed, the emulation allows to consider the impact of several parameters peculiar to the satcoms and so doing to see some limits hidden from the simulation-based studies.

For example, the Figure 1 represents the evolution of the sequence number in a communication on the forward link when the Gateway buffer is the only parameter to change. The impact on the evolution and so on the throughput is really significant. Moreover the different TCP protocols are not damaged in a same way.

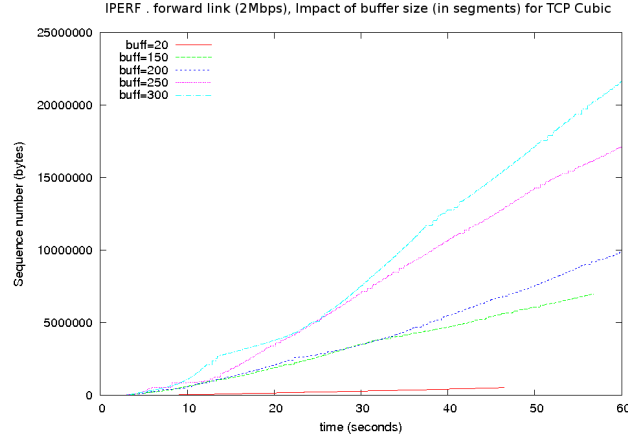


Figure 1. evolution of the sequence number for a fixed bandwidth

Because of those observations, we decided to present only results that can have been highly verified.<sup>9</sup> So, and because of the complexity met to analyze accurately Compound, which is mainly due to the lack of open tools, the continuation will present only the results obtained with New Reno and Cubic. However, researchs have been conducted to compare Compound TCP to others TCPs<sup>13</sup> and CTCP seems to be at least as efficient as Cubic in satcom.

Anyway, this article doesn't aim to compare the performance of the different TCP versions, but to identify what can justify the use of one system rather than another. So, the emphasis has been laid not on an accurate evaluation of the performance but on the behavior of the protocols in different contexts.

## B. Results

### 1. Comparison of Cubic and New Reno without PEP

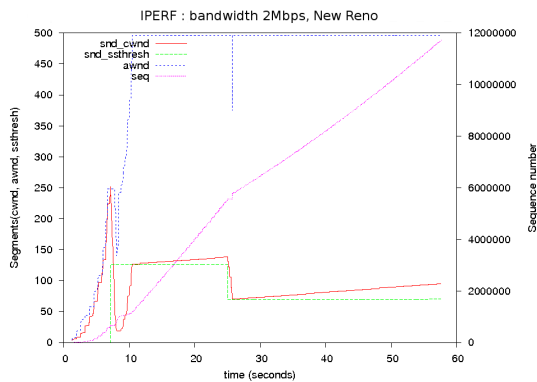


Figure 2. Reno : CWND and sequence number evolution

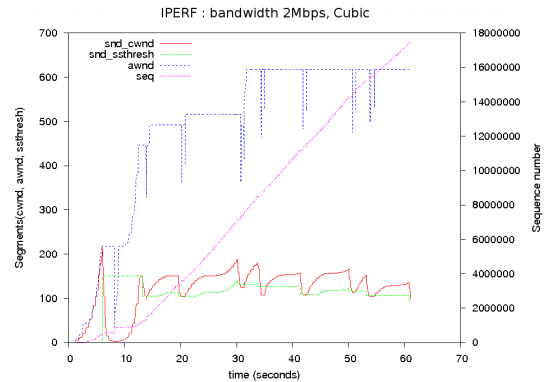


Figure 3. Cubic : CWND and sequence number evolution

The Figure 2 and Figure 3 illustrate the theory seen in the previous section by pointing out the correlation between the CWND and the throughput (represented by the gradient of the “sequence number” curve).<sup>3</sup>

TCP cubic uses the same slow start stage than New Reno and so during this time they have a similar throughput. Then, using its “three states” mechanisms (cf. section III), Cubic can reach a higher throughput and preserve it. Indeed even if suffering of more congestion events than New Reno, Cubic manages to keep a same throughput whereas each congestion event damages significantly the New Reno performance.

## 2. Experiments in the early stage

The previous theoretical analyses and tests show that there are few differences between both protocols during the beginning of the connection, and notably, that both can suffer from a large RTT. The Figure 4 confirms then that the combination of PEPs and TCP enable superior performances for the short connections that the TCP used alone.



Figure 4. evolution of the sequence number for a 102 kB transfer on the return link

## 3. Experiments with long connections

In opposition to the previous case, theory and tests show that modern TCPs allow significant improvements after the slow start stage. The interest of long connection scenarios is not only to see if Cubic behaves better than New Reno, but if its efficiency without PEP manage to counterbalance its bad start to reach a level close to the one with PEPs.

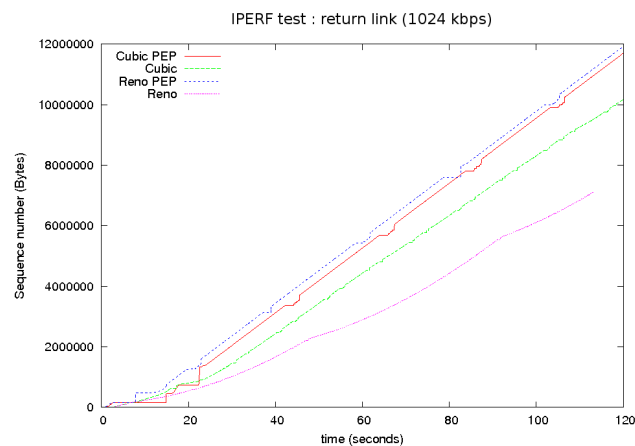


Figure 5. evolution of the sequence number for an Iperf transfer on the return link

Figure 5 and the different measurements show that even if Cubic is not able to fill the whole bandwidth, the gap between the performances with and without PEP is highly restricted with Cubic.



Another important point to notice is that the TCP performances using the PEPs have neither been improved by the use of a modern and more efficient TCP protocol nor been really damaged. Indeed, our experiments show that the TCP PEPs designed for a peculiar TCP protocol manage to keep high performances, even if a bit lower, when facing another TCP protocol, but are not able to improve them.

Moreover, our experiments on this kind of scenarios illustrated one of the PEP flaws seen in section II. Indeed, we noticed some very poor and unexpected performances when using PEPs on the forward link with a high bandwidth (strictly superior to 1 Mbps). The PEPs performance was limited to 1 Mbps. During their real experiments,<sup>10</sup> met the same problem. This seems to be an hardware limitation. If it is the case, this is typically an issue that can occur when using a blackbox in the middle of a network.

Thus, in a sustainable point of view, considering the important amount of current studies still dedicated to the improvement of the TCP protocol, it is likely that the future TCP protocols will allow if not to reach the performance obtained with the PEPs, at least to be close enough to consider that the balance between the performances and drawbacks is in favor of the abandonment of the PEPs systems.

## V. Conclusion

Middleboxes are used in many networks to accelerate applications by hiding or bypassing the worse constraints. In this paper, we examined the pros and cons of PEPs when applied to Satcoms. Because the majors flaws and advantages of the PEPs are well known, we focused on the improvements of the TCP protocol to see if they are or can become an efficient substitute. The emphasis has been laid to analyze what is responsible for the good behavior of modern TCP protocols through satellite links. The results confirm that the gap between the performances with and without PEP is thinner and thinner.

However, this study is not complete. Some complementary research needs to be done in order to test the fairness and friendliness of those modern TCPs in satcom and to compare their behaviors with and without PEPs in case of handovers. Then, it would be interesting to validate those emulation results with real experiments.

## VI. Acknowledgments

The authors are very grateful to Laurie Albaladejo and Patrick Gélard for their technical supports and pertinent comments.

## References

- <sup>1</sup>[www.opensand.org](http://www.opensand.org), 2012.
- <sup>2</sup>F. Arnal, T. Gayraud, C. Baudoin, and B. Jacquemin. Ip mobility and its impact on satellite networking. In *Proc. 4th Advanced Satellite Mobile Systems ASMS 2008*, pages 94–99, 2008.
- <sup>3</sup>R. Lacamera D Caini, C. Firrincieli. Comparative performance evaluation of tcp variants on satellite environments. *ICC'09 Proceedings of the 2009 IEEE international conference on Communications*, pages Pages 5161–5165, 2009.
- <sup>4</sup>E. Dubois, J. Fasson, C. Donny, and E. Chaput. Enhancing tcp based communications in mobile satellite scenarios: Tcp peps issues and solutions. In *Proc. 5th Advanced satellite multimedia systems conference (asma) and the 11th signal processing for space communications workshop (spsc)*, pages 476–483, 2010.
- <sup>5</sup>A. Fairhurst, G. Sathiaselan Arjuna, Haitham. Cruickshank, and C. Baudoin. Transport challenges facing a next-generation hybrid satellite internet. *International Journal of Satellite Communications and networking*, 29:249–268, 2011.
- <sup>6</sup>S. Ha, I. Rhee, and L. Xu. Cubic: A new tcpfriendly highspeed tcp variant. Technical report, 2005.
- <sup>7</sup>Y. Raiciu C. Greenhalgh A. Handley M. Tokuda H. Honda, M. Nishida. It is still possible to extend tcp? In *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*, 2011.
- <sup>8</sup>R.N. McCullagh G. Leith, D.J. Shorten. Experimental evaluation of cubic-tcp. In *Protocols for Fast Long Distance Networks*, 2007.
- <sup>9</sup>D. Shorten R.N. Li, Y. Leith. Experimental evaluation of tcp protocols for high-speed networks. *Networking, IEEE/ACM Transactions on*, 15:1109– 1122, 2007.
- <sup>10</sup>A. Sanadidi M.Y. Gerla M. Firrincieli R. Beering D. Romaniak G. Marcondes, C. Persson. Tcp in mixed internet and geo-satellite environments : Experiences and results. In *IEEE*, 2006.
- <sup>11</sup>K. Tan, J. Song, Q. Zhang, and M. Sridharan. A compound tcp approach for high-speed and long distance networks. In *Proc. 25th IEEE Int. Conf. Computer Communications INFOCOM 2006*, pages 1–12, 2006.
- <sup>12</sup>J. Tan.K, Song. Compound tcp: A scalable and tcp-friendly congestion control for high-speed networks. In *in 4th International workshop on Protocols for Fast Long-Distance Networks (PFLDNet)*, 2006.
- <sup>13</sup>P. Gayraud T. Planchou F. Kretzschmar V. Dubois E. Gelard P. Tou, I. Berthou. Which transport protocol for hybrid terrestrial and satellite systems? In *PSATS*, 2012.



<sup>14</sup>S. Trivedi, S. Jaiswal, R. Kumar, and S. Rao. Comparative performance evaluation of tcp hybla and tcp cubic for satellite communication under low error conditions. In *Proc. IEEE 4th Int Internet Multimedia Services Architecture and Application(IMSAA) Conf*, pages 1–5, 2010.