Computer Networks 56 (2012) 412–434

Contents lists available at SciVerse ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

# XPLIT: A cross-layer architecture for TCP services over DVB-S2/ETSI QoS BSM

Juanjo Alins, Jorge Mata-Diaz, Jose L. Muñoz *, Elizabeth Rendón-Morales, Oscar Esparza

*Universitat Politècnica de Catalunya (UPC), Department of Telematics Engineering, Barcelona, Spain*

ARTICLE INFO

*Article history:*
Received 1 October 2010
Received in revised form 19 April 2011
Accepted 7 September 2011
Available online 13 October 2011

*Keywords:*
Cross-layer
TCP
Satellite
DVB-S2
QoS
ETSI BSM

ABSTRACT

This article proposes XPLIT, a new architecture based on TCP cross-layering and splitting for optimizing the transport layer performance in a DVB-S2 satellite link that employs the ETSI QoS Broadband Satellite Multimedia Services (BSM) standard. The main novelty of our proposal is a complete architecture that perfectly fits this new DVB-S2/ETSI QoS BSM scenario. Our architecture includes the design of satellite-optimized cross-layer TCP protocol, called XPLIT-TCP that uses two control loops to properly manage the system load. The proposal has been implemented to be tested in the NS-2 simulator and we include the most interesting performance evaluation results, which show the excellent performance of our architecture for the intended scenario.



## 1. Introduction

GEO satellite networks will play a crucial role in future Internet due to the necessity of communication services anytime and anyplace. GEO satellite communications are suitable for scenarios where it is difficult or impossible to deploy wired communication infrastructures. Typically these scenarios are rural environments, developing countries or airplane/boat communications. Nevertheless, GEO satellite communications have also some drawbacks. The main drawbacks are: variable link capacity, high propagation delays, packet corruption and channel asymmetry. In particular, applications that use the reliable transport protocol TCP (Transmission Control Protocol) have a significant degradation over satellite links.

TCP degradation is mainly due to the fact that the congestion control algorithm of standard TCP is not suitable for overcoming the impairments of satellite links [1]. The objective of the standard TCP congestion control is to keep the load of the network close to the system capacity and to fairly share the available capacity between flows. Standard TCP congestion control is implemented using probing phases: a coarse-grained phase called slow start and a fine-grained phase called congestion avoidance [2,3]. The characteristic of this congestion control algorithm is that it measures congestion causing congestion. TCP increases its congestion window until there is a drop, and then the delivered load to the system is reduced. Therefore, TCP has a tendency to put as much data in the network as possible, filling the queues to the limit. In a GEO satellite network there is a long two-way propagation delay or minimum round trip time (RTT). The problem is that a long RTT causes TCP connections to take up to several seconds before reacting to or recovering from congestions.

A variety of solutions have been proposed to address the mentioned problems. These solutions can be roughly classified as black-box and cross-layering. In black-box solutions, TCPs considers that elements of the network are not accessible and thus, it views the network as a black box. On the other hand, approaches with some degree of explicit knowledge about the network state permit a better optimization of the TCP performance. In this type of

* Corresponding author. Address: Campus Nord, C3 Building, Office 310, Barcelona, Spain. Tel.: +34 934010804.
*E-mail address:* jose.munoz@entel.upc.edu (J.L. Muñoz).

1389-1286/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.comnet.2011.09.005

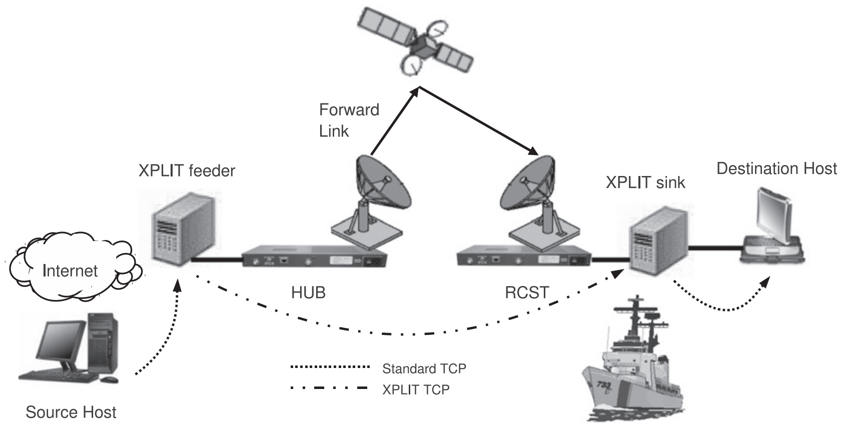


**Fig. 1.** XPLIT Scenario.

approaches, network devices send feedback to the TCP layer to enhance its performance. These TCPs are also called "*cross-layering* TCPs" [4]. All these approaches (both black box and cross-layering) require satellite-optimized TCP protocols to run on an end-to-end user basis. However, forcing all the users (including Internet servers) to run a satellite-optimized TCP is not achievable in practice because most operating systems do not include this type of TCP-flavors in their standard distributions. Furthermore, a satellite-optimized TCP might not be suitable for a terrestrial path. Thus, in this case end users should be able to foresee if the path includes a satellite link or not to apply the proper TCP-flavor, which is clearly impractical.

To mitigate the mentioned problems, the most promising solution for satellite networks is "*splitting*"[1] [5], which consists in dividing the TCP connection into a satellite portion and a terrestrial portion. The objective of splitting is isolating the long-latency satellite link with middle agents called Performance-Enhancing Proxies (PEPs). With splitting, the connection is divided in three TCP connections: sender-to-PEP, PEP-to-PEP and PEP-to-receiver. PEPs are responsible for intercepting, caching, acknowledge data received by senders and forwarding these data to receivers. Unlike proxy caches, splitting is transparent to both the sender and receiver. Splitting produces an enhancement of the end-to-end performance because it helps in reducing the probing time of each split connection.

In this article, we present XPLIT, an architecture based on splitting and cross-layering for the forward channel of a DVB-S2/ETSI-BSM QoS scenario. This scenario is shown in Fig. 1. As it can be observed, there are two elements that manage the physical medium and the data link layer of the satellite link: a satellite forwarder node (HUB) and a Return Channel Satellite Terminal (RCST). Each satellite subscriber has a RCST, while the satellite operator manages the HUB. The link HUB-to-RCST is called *forward* link and it is defined by the DVB-S2 standard [6]. The RCST-to-HUB link is called *return channel* and it is defined by the DVB-RCS standard [7]. In practice, the HUB is located in the infrastructure side (which belongs to the operator) where it is also connected to the Internet. Therefore, the forward link is intensively used by satellite terminals in their access to Internet applications and services. Satellite operators, service providers and hardware manufacturers are currently very committed to bringing the deployment of the DVB-S2/DVB-RCS standards to large-scale adoption [8].

PEP boxes are included to deploy the TCP-splitting architecture, connecting the HUB to the Internet access and the RCST terminals to the Destination Host. Taking into consideration the direction of the data flow, we call XPLIT feeder to the PEP box at the infrastructure side because this is the PEP box that feeds the satellite link with data in our architecture. In the same sense, the PEP boxes attached to RCST terminals are called XPLIT sinks.

On the other hand, our architecture is also intended for an ETSI-BSM QoS scenario [9]. The ETSI-BSM QoS group establishes a specification based on the Internet Protocol suite for providing quality of service to Broadband Satellite Multimedia Services. In this respect, the ETSI-BSM has been designed to be compatible with currently standardized IP DiffServ architectures, in which flows are aggregated in classes to obtain a QoS degree. In this scenario, the main concern is the management of variations of available bandwidth, satisfying the service level specifications for each DiffServ class.

Notice that available bandwidth not only varies due to the present traffic as it happens in terrestrial networks, but also because of the DVB-S2 link state, which is related to weather or other propagation conditions. Additionally, when considering QoS, the bandwidth of a particular flow of a traffic class can also change quite drastically if a considerable amount of traffic varies in a higher DiffServ class. To properly manage these changes, our architecture provides the buffer occupancy ($\beta_i$) and the service rate ($\mu_i$) of each traffic class to the congestion control algorithm. This algorithm manages the senders that generate split connections at the XPLIT feeder.

Our architecture also follows the general design rules of ECLAIR [10]. ECLAIR describes guidelines about how to design a cross-layering architecture for wireless protocol stacks. The ECLAIR main components, as shown in Fig. 2, are the OSS (Optimizer Sub-System) and the TL (Tuning

[1] However, splitting breaks the end-to-end semantic of TCP, which raises several philosophical issues [5] that are beyond the scope of this article.