



Open Archive Toulouse Archive Ouverte (OATAO)

OATAO is an open access repository that collects the work of some Toulouse researchers and makes it freely available over the web where possible.

This is an author's version published in: <https://oatao.univ-toulouse.fr/1862>

To cite this version :

Gineste, Mathieu and Thalmensy, Hervé and Dairaine, Laurent and Sénac, Patrick and Diaz, Michel Active emulation of a DVB-RCS satellite link in an end-to-end QoS-oriented heterogeneous network. (2005) In: 23rd AIAA International Communications Satellite Systems Conference (ICSSC), 25 September 2005 - 28 September 2005 (Rome, Italy).

Any correspondence concerning this service should be sent to the repository administrator:

tech-oatao@listes-diff.inp-toulouse.fr

Active Emulation of a DVB-RCS Satellite Link in an End-to-end QoS-oriented Heterogeneous Network

Mathieu Gineste^{*} and Hervé Thalmensy[†]
ENSICA, 1, place Emile Blouin 31056 Toulouse France

Laurent Dairaine[‡]
NICTA, National ICT Australia, Bay 15 Locomotive Workshop, Australian Technology Park, Eveleigh NSW 1430, Australia
ENSICA, 1, place Emile Blouin 31056 Toulouse France

Patrick Senac[§]
ENSICA, 1, place Emile Blouin 31056 Toulouse France

and

Michel Diaz^{**}
LAAS / CNRS, Avenue du Colonel Roche 31000 Toulouse France

I. Introduction

Emulation platforms are a classical way to achieve protocol experiments, particularly into the expensive and complex satellite environment. In the context of low level protocol developments, satellite emulation systems are based on real protocol implementations (e.g., DAMA, adapted routing protocols, etc), the emulation being achieved for the air interface. Such systems are often very accurate in the emulation service they provide, but are also very complex and expensive to develop and to maintain when protocols evolve. In the context of end-to-end communication, protocol experimentations do not need such low level emulation. Existing emulation tools are usually driven by very simple and static emulation model that does not fit the required level of realisms. Other solutions such as nse rely on discrete event simulation models to drive the emulation, but performance issues drastically limits this type of approach.

Our contribution aims to propose a platform integrating an emulation processor and potentially complex emulation model into an easy-to-deploy system. Its main originality is its flexibility of use and the higher level of abstraction proposed compared to existing emulation platforms. Instead of reproducing access to the satellite link according to a heavy implementation of low level protocols behaviour, only resulting effects on data transfer are emulated by the way of a adapted emulation model. This leads to a simple and performing implementation that can be thus combined with more complex emulation scenarios (e.g. for emulating a wider network including a satellite link) or operational networks. To achieve this goal, the emulation model must react in real-time to various external events such as time but also processed traffic, leading to the concept of active emulation. This type of emulation also enables to evaluate more precisely communication protocols thanks to its capacity to react finely to specific events of the protocols (e.g. dropping a specific packet, or emulation bandwidth adjustment according to traffic crossing the emulation processor). We study this approach in the context of quality of service services over satellite link.

^{*} Ph.D. Student, LIP6-ENSICA.

[†] PhD. Student, LAAS-ENSICA.

[‡] Researcher, NICTA, Australia.

[§] Assitant Professor, ENSICA, France.

^{**} Research Director, LAAS-CNRS, France.

DAMA as the access scheme on the return link enables to maximize the use of network resources thanks to a dynamic allocation method based on a channel-state-dependent scheduling scheme. This dynamic assignment method permits to ensure QoS guarantees as well as high utilization of the global resources of the satellite return links. To reach this objective, the return link scheduler defines four main capacity assignment types, which are guaranteed or not, always granted or on demand and thus experiencing different delays to cross the satellite link. A combination of this assignment types is often used on real satellite SLAs (Service Level Agreements). The main difficulty to emulate this access scheme without implementing the real DAMA system is the channel-state-dependant property, based on data buffered in the Satellite Terminal queues in order to request resources. Consequently, an active emulation is needed to take into account the amount of data reaching the satellite. The active emulation consists in reacting in real-time depending on the type and on the amount of data crossing the emulator: rules will be applied by a high level emulation module to modify the behaviour of emulated networks based on the current traffic specificities. Traffic will be experiencing immediately the resulting networks characteristics defined by these rules.

The paper will first introduce emulation approach as one of the possible solution of experimentation with protocols. Then, the integration of a QoS Satellite link into the EuQoS project and the DAMA access scheme characteristics are detailed. The emulation solution is proposed using the active emulation architecture. Finally, early performance results using the emulation platform are proposed to validate the approach.

II. Network emulation for experimenting with protocols

Rapid growth of Internet network support makes most of recent computer applications to integrate more and more important part of distributed functionalities. These distributed functionalities either make use of transport layer services, by the way of well-known transport protocol such as TCP or UDP, but also have to implement specific application and/or transport protocols in such a way to meet user's wishes in terms of functionalities and quality of service requirements. But these applications, and particularly their distributed components, have to be carefully tested to determine if they have good performance, and are reliable and robust. This involves the expensive task of testing and debugging the produced software that is expected to run on distributed computers (two or more) interconnected by either a large area networks such as Internet or by specific networks technologies such as satellite networks.

A classical mean to test and debug distributed software is to use real technology. These techniques are generally expensive and time-consuming. The first possibility is to experiment with a real technology, when possible, the target operational network, e.g., the satellite system. But lack of control on the network experimentation conditions makes it very difficult to achieve and still reduce the value of the resulted measurements and tests. The software can also be tested on ad-hoc test-bed using real equipment and implementing a specific technology. This technique can be expensive, particularly in the context of large area heterogeneous network but also in specific technological conditions, particularly in the context of satellite network. Moreover, it is sometimes impossible to implement this method because the technology support is not available (e.g., a new satellite transmission technology). This high experimentation cost is mainly due to the technology cost involved, but also to the distributed man-in-the-loop manipulations and synchronization it requires. Moreover, it has a limited value due to the inherent discrepancies between a particular test network and the much broader range of network imperfection that will be encountered by the software users.

Simulation and particularly event-driven simulation is a powerful tool to achieve cheap and fast experimentation. The main particularities of this approach are to rely on an ad-hoc model and to use a logical event-driven technique to achieve the simulation. The modelling techniques allow simplifying the studied problem, in such a way to concentrate on most critical issues. The logical time make the model execution not to be linked to real-time then, depending on model complexity; it is possible to either simulate a logical hour in few real-time milliseconds or a logical second in several real-time days. Due to both characteristics, no real implementations involving man in the loop are directly possible to achieve such a simulation. Then, a careful link between tested protocol model and real implementation has to be achieved in such a way to ensure that offered services and performance results between simulation model and the experimental implementation protocol will be consistent.

More recently, improvements in high speed processing and networking allows the development of network emulators. Network emulation allows reproducing, in real-time, the service and the quality of service of a target network. It is then possible to test on the emulation system, real implementation of protocols and applications. Then, network emulation is used to achieve experiments using both real protocols and distributed applications implementation and network model allowing to create a controlled communication environment. This communication environment can produce specific target communication behaviour in term of quality of service,

allowing reproducing real underlying network architecture but also artificial network impairments aiming at testing particularities of the experimented protocol. Mixing real-time experiment, real and simulated functionalities and emulation model, make this approach to be considered as an efficient and useful compromise between simulation and live testing. It provides a mean of experimentation using real data and a network model.

By allowing the real world traffic, protocols, application or even human to interact with these models, network emulator avoids the traffic modelling problem. Real software are evaluated, leading to a greater confidence on obtained results as well as avoiding re-implementing the real systems as it is the case when using the simulation approach. In addition, the emulation approach allows experiment flexibility and repeatability by a complete control on the simulation environment within the emulator conversely to real test-beds.

A simple (but very used) way to emulate a target network is a static emulation approach consisting in applying constraints on data flows according to not evolving parameters such as topology or technology capacity. Once defined, emulation parameters do not evolve during the experiment. A number of tools enable static network emulation to be implemented. These tools are traffic shapers constraining data flows between real end users. ENDE¹ emulates accurate end-to-end delays between two hosts for multimedia applications. DelayLine¹² allows emulating WAN characteristics. These tools, being implemented in end-hosts, require the user application to be linked with DelayLine library. Dummynet³ is a classical way to avoid any modification on evaluated protocols and applications. This tool is implemented into the Unix FreeBSD kernel providing efficiency and accuracy. Dummynet works by intercepting packets bridged or routed in the kernel, and introduces impairments such as delay, bandwidth limitation or packet loss rate on the flow of packets. Similar approach is developed into NISTnet² but provide better control on the emulation impairments. It allows e.g., packet duplication, and also setting up various distribution for delays. ONE⁷ is based on the same idea and uses three parameters to simulate a communication network (transmission delay, propagation delay and packet queues).

Netbed/Emulab⁹ or ModelNet¹⁰ are distributed emulation platforms using these tools as link level emulator and rely on a gigabit cluster to emulate high speed WAN links, but this approach is expensive and complex to implement.

These tools are mainly used as a black box aiming to reproduce a network link, network, or even inter network. The principal advantages of these static emulation tools are their simplicity of use while obtaining precise results as well as good performances. However, key functionality traffic shapers are missing, is their lack of control on the emulated behaviour. Their simple configuration does not allow introducing complex algorithms controlling the impairments in such a way to produce an accurate network behaviour. The network characteristics can change drastically due to external parameters such as traffic load, congestion, weather conditions in the case of satellite communication environment and mobility in the case of wireless network. Moreover, number of experimentation needs to observe the evolution or the impact of a particular parameter on time (e.g., bandwidth or delay).

Various types of dynamic emulation can be distinguished: Time dynamic emulation [Herrscher⁴] associates emulation scenario [NINE¹¹] based on time to the traffic to shape. Actual impairments that drives the emulation can e.g. result from traces coming from real network experiments. Emulators built on a trace-based approach reproduce the behavior of a real network (wired or not), according to previously captured traces [Noble¹³]. Those traces have to be processed to produce an emulation model that will be interpreted by the traffic shaping interface. However, capturing traces can be difficult and expensive in time. Moreover, traces are a snapshot of specific network conditions at some moment in time. Another approach to drive dynamically the emulation consists in coupling the emulation with real-time simulation. The emulation mode provided into ns-2⁵ operates with a modified scheduler instance and is able to process real packets. Two modes are proposed: an opaque mode where real packets are not modified and a protocol mode where real packets can interact with the simulated model. However the real-time scheduler make emulation mode of ns-2 very power consuming, leading to very low emulation capacity. In the context of wireless emulation, SWINE⁶ is a two-stage solution based on the real-time control of Dummynet. An offline simulation stage is used prior to the emulation to compute emulation scenarios, followed by an IP level emulation stage that reproduces these scenarios in soft real-time.

In this paper propose to extend the notion of dynamicity in such a way to drive the emulation with the emulated traffic itself. Active emulation (in reference to active networks) provides then the ability to decide how to impair the flows depending on packet or flow characteristics. This type of emulation enables to produce more precisely communication protocols thanks to its capacity to react finely to specific events of the protocols. This paper particularly focuses on the use of active emulation into the practical study case of emulation of quality of service services over satellite link. This emulated satellite link will be integrated into an heterogeneous QoS-aware testbed proposed in the context of EuQoS European Project.

III. A QoS Satellite system into the EuQoS testbed

EuQoS is a project funded by the European Commission and involving a great deal of European research laboratories as well as Telecommunication key players. The objective of this project is to assess Quality of Service on an end-to-end basis, across heterogeneous networks. One of the multiple network types considered in this project corresponds to a QoS enabled satellite link as an access network. As shown in figure 1, the satellite link is based on the widely spread DVB-S standard for the Forward link (from the Hub Station toward the Satellite Terminal) and DVB-RCS standard for the Return link (from the Satellite Terminal toward the Hub Station). This satellite access network has to be provided in order to make the corresponding experimentations when a satellite link is part of the end-to-end communication. Due to the high cost of satellite resources, experimentations involving this access network will be done on an emulated satellite link. In order to develop this emulation platform, the characteristics of the satellite link have been studied carefully.

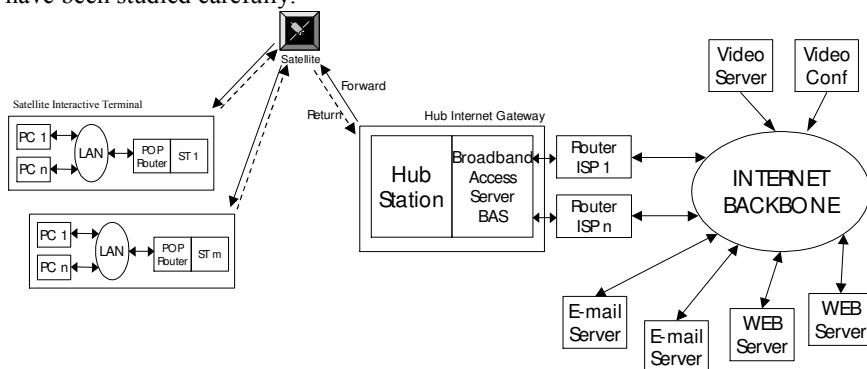


Figure 1: Satellite Link as an access network to the Internet

The main issue in the satellite communication context, and more particularly on the satellite return links, is to make an efficient use of the precious resources, scarce and costly. Thus, protocols have been designed to optimize the usage of these resources and especially to share properly and efficiently the return link resources accessed by multiple terminals. Some protocols have been proposed, using random or dedicated assignment techniques to access the shared resources on the return link. The former is not compatible with traffic having high Quality of Service requirements while the latter is doing a very inefficient usage of the satellite link resources.

Other techniques, based on dynamic bandwidth assignment, enable a high efficiency of the return link usage. The first technique consists in requesting resources when a connection starts which is identified by new data arrival. In a second technique, the Satellite Terminals get resources depending on the activity of their channel. This channel-state-dependent approach permits to allocate resources only when needed. Emerging protocols, such as DAMA (Demand Allocation Multiple Access) are integrating a combination of these existing protocols in order to both ensure a high utilization of the return link resources and to offer Quality of Service oriented capacity assignment types. This access scheme will be the one targeted for the satellite access network emulation.

A. General Description of Satellite Standards

The Digital Video Broadcasting Project (DVB) is an industry-led consortium of over 300 broadcasters, manufacturers, network operators, software developers, regulatory bodies and others in over 35 countries committed to designing global standards for the global delivery of digital television and data services. DVB technology has become an integral part of global broadcasting, setting the global standard for satellite, cable and terrestrial transmissions and equipment. DVB standards and specifications are available from ETSI. The DVB Return Channel System via Satellite (DVB-RCS) was specified by an ad-hoc ETSI technical group founded in 1999. This tracked developments by key satellite operators and followed a number of pilot projects organised by the *European Space Agency (ESA)*. The DVB-RCS system specifies a satellite terminal (sometimes known as a *Satellite Interactive Terminal (SIT)* or *Return Channel Satellite Terminal (RCST)*) supporting a *two-ways* DVB satellite system. The use of standard system components provides a simple approach and should reduce time to market. The satellite user terminal receives a standard DVB-S transmission generated by a satellite hub station. Packet data may be sent over this forward link in the usual way (e.g. MPE).

B. Technical characteristics of Satellite communication

a) DVB-S and MPEG-2

Initially designed for Audio and Video transmission, MPEG is a set of norms defining every transmission details from the system part, to video compression details. More and more, however, DVB is used to build Internet-compatible networks. In order to do this an MPEG-2 Transport Stream cell is used as a “container” for the IP packets with an added encapsulation header to allow the information to be adequately processed. Over the years a number of encapsulation methods have been explored to transmit data over broadcast media. The current standard for DVB is the Multi-Protocol Encapsulation (MPE). While this may be used to send Internet Packets, there are further issues that arise when used to build an Internet service.

MPEG system coding exists in two forms: The Transport Stream and the Program Stream, each being optimized for a certain type of application. Both kind of streams are logically constructed from a flow of PES (Packetized Elementary Streams) packets, they differ mainly in their timing and error requirements.

This definition stand for video and audio transmission, but pure data transmission has not been left apart. MPE (Multi Protocol Encapsulation) is especially meant for transporting data network protocols over the MPEG-2 TS in the DVB world. It has been somehow optimized for the transport of IPv4 packets: in this case the IPv4 packet is stored as the section payload.

b) DVB-RCS

The DVB-RCS system is based on the following principle: A *Return Channel Satellite Terminal (RCST)*, once powered on, will start to receive general network information from the *DVB-RCS Network Control Centre (NCC)*. The NCC provides monitoring and control functions and generates the control and timing messages required for operation of the satellite network. All messages from the NCC are sent using the MPEG-2 TS using private data sections (DVB SI tables). These are transmitted over the *forward link*. Actually the DVB-RCS specification calls for two forward links - one for interaction control, and another for data transmission. Both links can be provided using the same DVB-S transport multiplex.

All transmissions by a user terminal are controlled by the NCC. Before a terminal can send data, it must first join the network by communicating (logon) with the NCC describing its configuration. This dynamic resource admission control permits to assure an optimal use of the costly resources of the satellite. Dynamic resource control consists in assignment of resources (slots) to terminals based on their requests and limit values negotiated/set during connection establishment. The assignments are conditioned by the availability of resources (capacity) within defined return channels (as per system connectivity). The assignment is the responsibility of the MAC Scheduler (in the NCC), which implements a Demand-Assignment Multiple Access (DAMA) protocol.

The uplink scheduling consists of processes taking place in the Scheduler and in terminals (as described in [QoSRCs¹⁴]), namely:

- Calculation of capacity requests in terminals;
- Transmission of capacity requests to the scheduler (request signalling);
- Calculation of capacity assignment to terminals;
- Allocation (placement) of the assigned capacity based on the Terminal Requests and on the SLA (Service Level Agreement) passed between the Terminal and the Hub at logon;
- Generation and transmission of TBTPs (Terminal Burst Time Plan) carrying the allocations to terminals;
- Distribution (within terminals) of the allocated capacity to the end users and their applications (consistent with ST MAC queue architecture and service discipline).

The SLA between the Terminal and the Hub defined at logon specifies guarantees on different classes of access to the Return Link of the satellite. These classes are defined in the DVB-RCS standard as capacity allocation of different type. We consider that the DAMA (Demand Assignment Multiple Access) is used as the access scheme on the return path of the satellite. The DAMA implementation goal is to minimize time delays and to maximize the use of network resources using a dynamic allocation method of resources based on a channel-state-dependent scheduling scheme. This dynamic assignment method permits to ensure Quality of Service (QoS) guarantees as well as high bandwidth efficiency. The return link scheduler supports four main capacity assignment types to reach this objective. The definitions and characteristics of these four traffic assignment classes as defined in the DAMA are the following (see also Figure 2):

- Constant Rate Assignment – CRA). This traffic assignment is programmed directly in the hub and requires no dynamic signalling from the ST (Satellite Terminal). This capacity is guaranteed. If the connection set-up is admitted, the terminal will benefit of its requested bandwidth for the duration of the connection, so that the traffic is not subjected to any scheduling/queuing delay. The network controller must ensure (at the connection set-up time) that the total CRA of all the Satellite Terminals does not exceed the total system capacity. Then, the CRA is a rate capacity fully provided for the duration of the connection without any

DAMA request. The Satellite Terminal having CRA bandwidth in its contract (SLA: Service Level Agreement), does not have to make any request and therefore, the associated satellite network delay is the one associated to the geostationary round trip time, e.g., 300ms, a ping echo between the ST and the NCC is then around 600ms).

- Rate Based Dynamic Capacity – RBDC. This variable rate traffic assignment has two components – a prefixed ceiling rate (RBDC_{max}) and an instantaneous rate request. The ceiling rate is established during call set-up using similar methods to CRA set-up. The capacity is guaranteed at the requested rate, subject to the ceiling (RBDC_{max}). A request remains effective for a number of frames (until it is updated by the terminal or as long as it has not timed out). The Satellite Terminal requests a bandwidth. The NCC answers with a TBTP (Terminal Burst Time Plan) valid for a period of time. If the time between two capacity requests is lower than the TBTP validity, then the ST is able to anticipate the need. So, RBDC is very similar to CRA when the traffic rate is high (but lesser than RBDC_{max}) and regular. The RBDC bandwidth is guaranteed until the ceiling. A high rate ping echo profile is therefore 2000ms for the first pings (due to capacity requests: up to 500ms to launch the request, around 600ms for the requests RTT, up to 300ms for request processing in case of congestion, around 600ms for the data packet transfer) and then around 600ms. The network controller must also ensure that the total CRA and RBDC_{max} of all STs does not exceed the total available return path capacity. CRA and RBDC can be used in combination. Thus, the CRA gives a fixed minimum capacity per frame and RBDC gives a dynamic variation component on top of this minimum.
- Volume Based Dynamic Capacity – VBDC. This best effort traffic assignment is a capacity assigned when available in response to a request. The amount of capacity assigned in any frame is not guaranteed. Each request is dynamically signalled. A ST signals its request in terms of total number of slots required to empty its queue. These requests can be supplemented by new requests any time more traffic is queued. The scheduler accumulates all requests from each terminal. The total capacity of the queued requests is decreased as soon as slots are granted. The scheduler assigns capacity from these queue requests, within the constraints on capacity remaining after CRA and RBDC assignments and limits on maximum terminal transfer rate. Practically, the terminal must do a capacity request to the NCC based on the volume of data to transmit in its buffers. Then the NCC allocates some time slots through the Terminal Burst Time Plan (TBTP). When the ST receives its TBTP, it can use the satellite network capacity accordingly. The first impact is the associated delay. In VBDC, the delay depends on three parameters: latency to send a capacity request, Minimum Latency Scheduling to receive the TBTP (congestion dependant), Geostationary Round Trip Time. A typical delay varies between 1400 ms and 2100 ms or more in case of severe congestion. The VBDC is not a guaranteed traffic assignment but a potentially available one in response to a traffic burst at the ST. In the worst case scenario, the NCC could allocate no bandwidth at all to the ST requesting capacity.
- Guaranteed VBDC (G-VBDC) or High Priority VBDC (HP-VBDC). This traffic assignment is a capacity that can also be defined by setting a minimum value for VBDC (MinVBDC) per ST and configuring it in the MAC Scheduler and ST. VBDC capacity up to MinVBDC will be granted in every super-frame, and so VBDC capacity up to MinVBDC is treated as a third form of guaranteed capacity along with CRA and RBDC. Note, however, that it does not obey the RBDC rules, meaning that requests equivalent to VBDC shall be done when traffic is queued in the ST, but up to MinVBDC, there is a guaranty that the capacity will be granted. VBDC requests for capacity in excess of MinVBDC will compete for the balance of (standard) VBDC capacity with other STs.
- Free Capacity Assignment (FCA). This capacity category is assigned to STs on an “as available” basis, from capacity that would be otherwise unused. This is automatic and does not involve any capacity request from the terminal to MAC Scheduler and the ST has no say in FCA assignment. In the basic MAC Scheduler design, each terminal is limited to one FCA slot per superframe, but alternatively more than one slot can be freely assigned and thus redistributed, where the ratio of terminals to free capacity makes it meaningful. A terminal may qualify or not for FCA, based on a configuration parameter in the terminal profile, which is made available to the MAC Scheduler. FCA can improve the MAC Scheduler performance (throughput), especially in low loading conditions, but it can introduce jitter, therefore its use should be limited to applications that are jitter-tolerant. FCA is normally used to supplement VBDC.

Figure 2 summarizes characteristics of the mentioned traffic assignment types:

| Traffic Assignment Type | Guaranteed | On Demand |
|-------------------------|----------------------|--------------------------------|
| CRA | Yes | No |
| RBDC | Yes, up to a maximum | Yes, Rate Based and Timer |
| VBDC | No | Yes, Volume Based |
| Guaranteed VBDC | Yes | Yes, Volume Based |
| FCA | No | No (automatically distributed) |

Figure 2: DAMA Traffic Assignment Types Characteristics

IV. Active network emulation

The following section outlines the concept of active network emulation and the application of this concept for emulating complex mechanisms of satellite communication described in the previous section. We will see how active emulation can be a realistic way to emulate complex network behaviors and in particular the satellite return link.

A. Concept presentation

Given networks produce particular network dynamic that may require complex models and mechanism to be emulated. The complexity of this type of emulation often comes from the highly dynamic behaviour of the protocols used in such networks. Protocols react depending on internal factors like their own mechanisms and external factors such as the traffic crossing the network. For instance, Satellite network and particularly access scheme such as DAMA proposes four different types of traffic assignment techniques that may be combined together. Some of them depends on the traffic characteristics while some others not. The behaviour of this access scheme is not predictable in advance, that is why dynamic configurations of the emulation system is required. The only way to have a sufficiently realistic emulation behaviour is to react in real time onto the emulation model according to various factors including the experimental traffic. In this sense the Emulation need to be active: the traffic will modify the configuration of the emulation and consequently the resulting modifications will have an effect on the traffic crossing the emulator. Active emulation is thus propose (in reference to active networks).

In this paper, an emulation architecture is proposed in order to implement this active emulation concept. This architecture is able to take into account the information on the traffic and to react accordingly using low-level modules. Then, the architecture proposes a high level of abstraction of the emulation in order to make it simple to use it and create any kind of target network emulation easily. High-level modules will permit to do this easy configuration and definition of rules. Besides taking into account new types of network behavior, our approach aims at providing the functionalities offered by the current emulation techniques such as dynamic configuration, traced based emulation, etc. The combination of these approaches can finally lead to emulate real complex networks into an easy-to-deploy system.

B. Architecture description

The architecture is divided into two sets of modules:

- High level modules: implementation-independent, managing emulation model and reacting depending on the characteristics of the data transferred;
- Low level modules: implementation-dependent, capturing and sending experimental traffic and applying low level rules according to the high level emulation model provided by the previous modules.

The figure 3. represents this active emulation architecture:

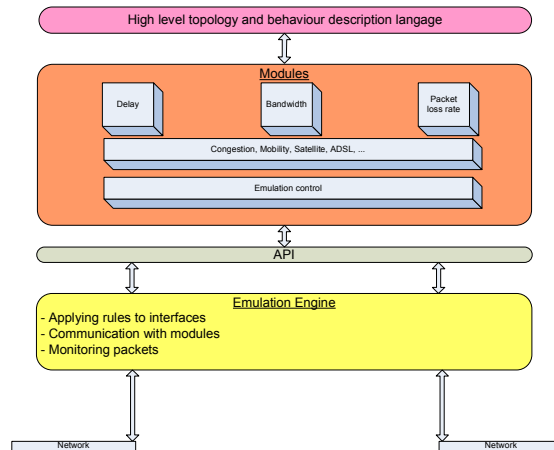


Figure 3: Active emulation general architecture

The High-level modules are divided into two sets of elements:

- The higher element is a high-level language aiming to configure the emulation platform from an abstract and simple point of view. This XML-based language enables to describe the network topology and the possible configuration of various network elements composing this topology as well as scenarios introducing an evolving and dynamic configuration of the network.
- The second set of elements corresponds to different pre existing network modules (new modules can be developed easily using basic pre existing modules). These modules can emulate basic network behavior or more complex ones including interaction with the traffic crossing the emulator leading to the active aspect of this architecture. Typical modules possibly provided correspond to satellite Link, DSL access, 802.11 wireless access, or core network for instance. But basic modules such as delay, losses or throughput limitation might be possible to use, in particular when developing new types of more complex network behavior. These high level modules remain though independent of the implementation characteristics of the emulation. The emulation control is also part of this set of elements and is the element in charge of managing information on the traffic (it will for instance apply mask in order to retrieve required information on the traffic).

The interface between these two modules is performed this way: The first step is to define the topology and the possible configuration of the various elements of this topology as well as a possible scenario using the high-level language. The resulting XML file is then parsed in order to configure and set the appropriate network modules as well as to retrieve scenario information to be used during emulation. The adaptation API enables to translate the rules defining the required behavior of the high-level modules driving the emulation toward implementation dependant rules, understandable by the emulation engine.

The emulation engine finally applies these rules to interfaces in order to configure the impairments on the traffic as needed. The emulation engine is also in charge to provide in real time required information about the traffic, to the high-level modules, such as sequence number of packets, packets' size or any kind of useful information for the high level modules in order for them to take some decisions on the evolution of the emulation and thus to apply appropriate rules to the ongoing traffic. A traffic shaper is in charge of applying constraints on traffic crossing the emulator in accordance with the current configuration done through high level modules.

V. Use case: QoS Satellite Communication

Emulation platforms are a classical way to achieve protocol experiments, particularly in the expensive and complex satellite environment. Several techniques might be used to emulate a Satellite Link:

- In the context of low level protocol developments, satellite emulation systems can be mostly based on real protocol implementations (e.g., DAMA, adapted routing protocols, etc), using wire emulation model to avoid the use of real air interface. The complete protocols behavior and signaling is then actually implemented while only the physical link is emulated, introducing geostationary end-to-end delay, and possibly loss models. Such systems are often very accurate in the emulation service they provide, but are

also very complex and expensive to develop and maintain when protocols evolve. In the context of end-to-end communication, protocol experimentations do not need such low level emulation

- The active emulation approach proposed in this paper is an alternative: Instead of reproducing access to the satellite link according to a heavy implementation of low level protocols behavior, only resulting effects on data transfer are emulated by the way of a proper emulation model. This leads to a simple and performing implementation that can be thus combined with more complex emulation scenarios (e.g. for emulating a wider network including a satellite link) or operational networks. To achieve this goal, the emulation model must react in real-time to various external events such as time or processed traffic, leading to the concept of active emulation. The advantage compared to the first type of emulation is that it provides an “in a box” solution integrating an emulation processor and potentially complex and realistic emulation model (not limited to satellite links) into an easy-to-deploy system.

C. Emulation specification

1. Forward link

Concerning the emulation of the Forward link of a Satellite using DVB-S, a simple and realistic approach is used. Active emulation is not needed in this case but a simple added delay and throughput limitation will emulate well this link in this platform. (a packet loss rate model might also be included, but we are not going to detail it in this paper). The typical delay to cross a geostationary satellite link is around 300 ms, this will be the delay used for the packets going through the Forward Link. The limitation on the throughput depends on the type of satellite and the contract used on the Return Link. A possible configuration through high level modules will be possible in order to set this bandwidth limitation. Loss rate depends mainly on the possible weather condition. So different loss models representing different weather conditions shall be proposed for the configuration of the emulator.

2. Return link

Figure 4 presents the architecture used to emulate a DVB-RCS satellite link using DAMA as access scheme composed by several types of traffic assignment classes.

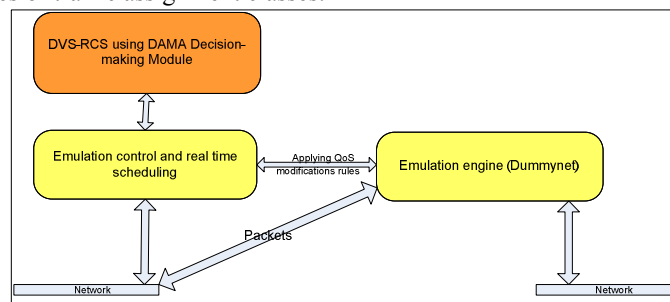


Figure 4: Active emulation of DVB-RCS satellite link architecture

Decision-making module decides bandwidth limitation, delay and packet loss rate values to apply to the emulation processor which actually impairs the flows, according to the initial configuration of the satellite link emulation as well as information concerning ongoing network traffic. Depending on the initial configuration corresponding to the type of contract existing with a potential Satellite network provider, several type of traffic assignment classes up to some limitations will be available and accordingly instantiated in the emulator in order for the traffic to experiment the proper behaviour. Depending on the traffic assignment types to emulate and on the ongoing traffic, the decision-making module will initially set some values for the parameters such as delay and throughput of the different pipes and will take decision to modify these values in real-time in order to emulate the time required by signalling process and the influence on the resource delivered to a particular Satellite Terminal. However, the signalling protocol is not really implemented, but just its resulting effects on the traffic. To take the emulation decisions, the emulation control need information about the traffic. The information on the traffic is provided by the emulation controller using a packet sniffer to retrieve the required information from the packets.

D. DVB-RCS emulation implementation

In the case of the DVB-RCS Satellite link using DAMA as its access scheme, a traffic assignment type is always granted namely the CRA and two classes of traffic assignment types need signalling protocols in order to obtain resources. The two classes mentioned corresponds to VBDC and RBDC who need to send requests when traffic is

reaching the satellite terminal in order to get resources on the satellite return link for this traffic. A third type of traffic is not requested but not guaranteed either and depends on the resources not used by the other terminal (FCA).

The first type of traffic corresponding to an always granted and guaranteed traffic (CRA) is very easy to emulate with a fixed delay applied to packets as well as a limitation on the total throughput. However, this type of traffic can be combined with on-demand traffic such as VBDC or RBDC; it then becomes more complex to emulate. On-demand traffic such as VBDC needs to send requests to the NCC in order to get the necessary resources on the satellite return link. This request takes a certain amount of time to cross the satellite link and to be processed. We don't want to reproduce the signalling process of sending requests and process them to keep the implementation simple and thus extensible. However all this delay might be emulated for a realistic behaviour of this type of traffic. The resources then granted will be emulated thanks to an evolving limitation of the throughput on the ongoing traffic.

The implementation of this system has been achieved using the Libpcap library for user-level packets capture, and FreeBSD/Dummynet as emulation processor. Because network monitors run as user-level processes, packet must be copied across the kernel/user-space. Libpcap provide access to information inside packets such as packets size for example. Then amount of traffic can be calculated by the module. However, few modifications on Dummynet have been achieved especially concerning bufferisation because the emulation was requiring some larger queue than hundred packets basically proposed in Dummynet.

E. Emulation of CRA and VBDC:

As stated previously, CRA emulation is pretty straightforward to implement as every packets experiment around the same delay of 300 ms to cross the return link of the satellite. This delay is emulated using a simple Dummynet delay pipe with a constant value representing the geostationary delay. Then a limitation on throughput is emulated using a Dummynet bandwidth pipe corresponding to the maximum bandwidth available on the satellite link for this type of traffic and set when configuring the emulator. A possible loss model can then be introduced to emulate losses on the Satellite Link. Thus in this case, active emulation is not necessary if only CRA is emulated.

However to implement VBDC traffic class, active emulation is required because in this case, the behavior is traffic-dependant. Consequently, to emulate VBDC class of traffic in standalone (not combined with other traffic classes), four Dummynet pipes have been chained as represented on the following figure

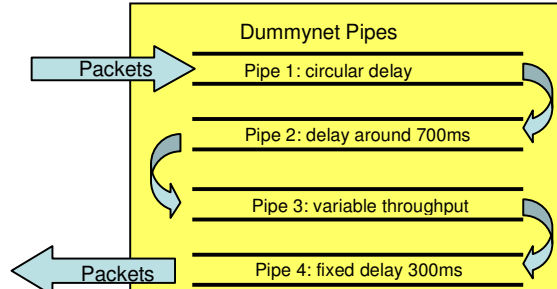


Figure 5: Pipes composition for VBDC traffic class emulation on the return link

The packets reaching the emulator toward the return link connection are experimenting four pipes introducing delay for three of them and bandwidth limitation for the fourth as seen on Figure 5. Each of these pipes is playing a role representing an aspect of the VBDC protocol behaviour. In particular, the signalling part of this protocol has an effect on the delay and bandwidth experimented by the traffic reaching a satellite result link. The real implementation of this signalling protocol is not going to be done, but just the resulting effects on the traffic in term of delay and bandwidth. Here is the detail of the roles of the four pipes:

- The first pipe corresponds to a circular delay periodically decreasing between 500 millisecond and 0 milliseconds (when 0 ms is reached the delay is then set to 500 ms again). It emulates the interval of time between two requests are sent from a Satellite Terminal toward the NCC.
- The second pipe corresponds to a delay around 700 ms. It emulates the time for the requests to cross from the Satellite Terminal to the NCC, to be processed and come back to the Satellite Terminal. The satellite link is thus crossed twice which is a fixed delay around 600ms, however the request treatment can vary depending on the congestion in the NCC. At least 100 ms is required to process the requests but this time can also increase.
- The third pipe corresponds to a variable throughput. It emulates the rate at which the Satellite Terminal is actually sending data on the link. Configuration of the 3rd pipe depends on the information retrieve from the

traffic. In particular, all packet length reaching the emulator to cross it on the return link will be passed to the DAMA decision making module that will then calculate the throughput to apply to this pipe depending on the rate reaching the emulator on the return link and up to the limitation of VBDC traffic contracted in the SLA (the guaranteed VBDC part). The configuration of the rate is calculated when packets arrive on the emulator but is applied after emulated requests came back meaning after the traffic has waited in the first two pipes. There is a delay introduced between the rate is calculated and is applied to the third pipe in order to put a constraint on the traffic on which the calculation has been done. It means that the queuing will be done in this part due to traffic limitation, and if the traffic is too high for the satellite link capacity, losses will be encountered when this pipe buffer will be full.

- The fourth pipe corresponding to a fixed delay of 300 ms emulates the delay encountered by the traffic to cross the satellite link.

VI. First Experimentation Results

Experimentations have been carried out using a Network Emulator platform called NINE (Nine Is a Network Emulator) developed at ENSICA. NINE manages a set of end-stations and is able to reproduce a specific network service between those stations. It is based on the Dummynet running FreeBSD and a set of software allowing the overall platform configuration and experiment controls. An active emulation model based on a set of rules defines how the emulator applies constraints to the traffic between the various end-systems in term of delay, bandwidth and packet error rate.

For the experiments, the application used corresponds to a constant bit rate of 128 kbps measured at the IP layer, the overhead is included. This stream is going through the return link of the satellite and a response is sent for each packets through the Forward link to measure the RTT delay. Dimensioning of the Forward link, implies that no congestion is encountered for the responses. Two scenarios are foreseen, one with a pure guaranteed VBDC configuration of the emulation system and a second one combining CRA assignment types with guaranteed VBDC. For both scenarios the packets RTT delay is measured. Dimensioning of buffers in the Satellite Terminal is big enough for no packets to be lost by overflow.

Pure VBDC (512 kbps guaranteed) results:

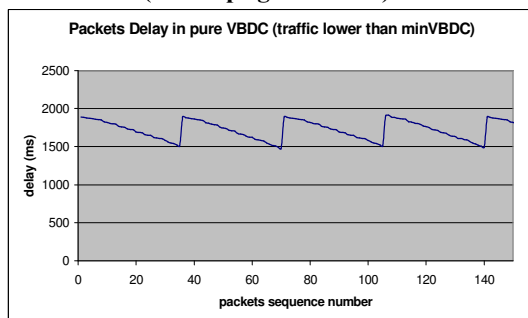


Figure 6 : Packets RTT delay in VBDC

just after the request has to wait for 530ms the following request is sent, implying for this packets an RTT increasing directly up to a delay around 1900ms.

Mixed CRA (64 kbps) and VBDC (64 kbps guaranteed) results:

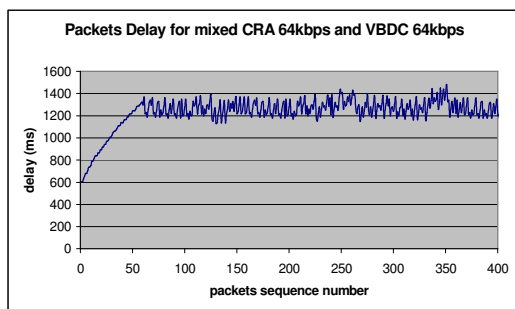


Figure 7 : Packets RTT delay in CRA and VBDC

The bit rate sent by the application, 128kbit/sec overhead included, is less than the minVBDC bandwidth guaranteed in the SLA. Requests for all the traffic buffered is done every 530 ms. This implies a pretty bursty behaviour of VBDC that is going to send at the higher rate as possible the traffic received in this 530 ms interval. This leads to a delay varying a lot depending on when the data arrived compared to when the following capacity request is sent. When the capacity is granted, all the traffic in the queue is sent very fast (at the rate of 512 kbps). The RTT delay of packets is thus decreasing as seen on Figure 6, because the traffic arriving just before a capacity request is sent, is waiting less in the buffer than the one arriving just after a request has been sent. Traffic arriving

The bit rate sent by the application, 128kbit/sec overhead included, is twice as much as the available CRA bandwidth of the SLA. Consequently, during the period corresponding to the transfer, processing and granting of VBDC requests, packets are buffered in the bandwidth queue. The number of packets in this buffer increases, implying an increase in the delay experienced by these packets as seen on Figure x. At the moment, first VBDC requests went through, the bandwidth is set to the one actually sent by the application, thus the bit rate entering the buffer is equal to outgoing traffic. Consequently, the buffer is neither emptying nor increasing in size but get stabilized as well as the delay

perceived by packets as seen on the Figure 7. Some fluctuations still persists coming from the bursty behaviour of VBDC requests process.

These first experimental results show some interesting behaviour of the satellite emulation systems close from results that could be obtained from an emulation platform implementing real Return link protocols (comparison is done with results obtained on an emulation platform in the context of an ESA project, TRANSAT, lead by Alcatel Space Industry).

VII. Conclusion and future works

This paper introduces a new concept, active emulation, enabling to achieve the DVB-RCS complex behavior, and particularly the DAMA access scheme. The concept of active emulation permits to take into account the traffic crossing the emulator in order to define dynamic emulation rules. This aspect is mandatory to emulate in a realistic way, the highly dynamic DAMA access scheme behavior. This emulation system after being more extensively tested, is going to be used as an easy-to-deploy solution to emulate a satellite access in the EuQoS project intending to study Quality of Service on an end-to-end basis, across heterogeneous networks. This will allow to integrate a satellite link in the end-to-end Quality of Service architecture and measure its effect on the global QoS perceived by users. The first experimental results show promising behaviours of the emulation system. These experiments need to be extended in order to evaluate the emulator with TCP traffic type as well as with several concurrent flows. Emulation need also to evolve toward traffic differentiation enabling the Quality of Service of traffic assignment types to be efficiently used, in accordance with the applications requirements.

References

- ¹I.Yeom and N.Reddy, *ENDE: An End-to end Network Delay Emulator*, Journal on Multimedia Tools and Applications, August 2001.
- ²M.Carson, D.Santay; NIST Net: *A Linux-Based Network Emulation Tool*, ACM Computer Communication Review, 2003.
- ³L.Rizzo, *Dummynet: A simple approach to the evaluation of protocols*, in ACM Computer Communication Review, vol.27, no 1, pp.31-41, 1997.
- ⁴D. Herrscher, and K. Rothermel: *A Dynamic Network Scenario Emulation Tool*. In Proceedings of the 11th International Conference on Computer Communications and Networks ICCCN '02, pp. 262--267, Miami 2002
- ⁵Kevin Fall, *Network Emulation in the VINT/NS Simulator*, in Proceedings of the the Sixth IEEE Symposium on Modeling, Analysis and Simulation of Comuter Telecommunication Systems, pp. 244-250, Red Sea, Egypt, July 1999.
- ⁶T. Pérennou, E. Conchon, L. Dairaine, M. Diaz, *Two-Stage Wireless Network Emulation*, Workshop on Challenges of Mobility (WCM 2004), August 2004.
- ⁷M.Allman, A.Caldwell, S.Ostermann *ONE: The Ohio Network Emulator*, TR-19972, School of Electrical Engineering and Computer Science, Ohio University, August 1997.
- ⁸C. Bolo, *End-to-End Packet Delay and Loss Behavior in the Internet*, ACM Computer Communication Review, vol.23, No. 4, October 1993, pp. 289-298.
- ⁹B.White, J.Lepreau, L.Stoller, R.Ricci, S.Guruprasad, M.Newbold, M.Hilber, C.Barb and A.Joglekar: *An Integrated Experimental Environment for Distributed Systems and Network*, in Proceedings of the Fifth Symposium on Operating Systems Design and Implementation (OSDI), pp. 255-270, Boston, MA, December 2002.
- ¹⁰A. Vahdat, K.Yocum, K.Walsh, P.Mahadevan, D.Kostic, J. Chase, and D.Becker: *Scalability and Accuracy in a Large-Scale Network Emulator*, in Proceedings of the Fifth Symposium on Operating Systems Design and Implementation (OSDI), Boston, MA, December 2002.
- ¹¹L. Lancerica, L. Dairaine, F. de Belleville, H. Thalmensy, C. Fraboul, *MITV, A solution for Interactive TV Based on IP Multicast over Satellite*, IEEE International Conference on Multimedia and Expo, (ICME), June 2004
- ¹²D. Ingham, *DelayLine, A Wide Area Network Emulation Tool*, Master Science Thesis, University of Newcastle, September 1992.
- ¹³Brian Noble, M. Satyanarayanan, Giao Thanh Nguyen, Randy H. Katz, *Trace-Based Mobile Network Emulation*, in proceedings of SIGCOMM, 1997
- ¹⁴S. Combes, L. Goegebeur, N.Sannier, A. Iuras, *Integrated Resources and QoS Management in DVB-RCS Networks*, June 2004