



HAL
open science

IP-Level Satellite Link Emulation with KauNet

Tanguy Pérennou, Amine Bouabdallah, Anna Brunstrom, Johan Garcia, Per Hurtig

► **To cite this version:**

Tanguy Pérennou, Amine Bouabdallah, Anna Brunstrom, Johan Garcia, Per Hurtig. IP-Level Satellite Link Emulation with KauNet. International Workshop on Satellite and Space Communications, 2009. IWSSC 2009., Sep 2009, Siena, Italy. pp.349-353, 10.1109/IWSSC.2009.5286348 . hal-00388811

HAL Id: hal-00388811

<https://hal.science/hal-00388811>

Submitted on 27 May 2009

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.

IP-Level Satellite Link Emulation with KauNet

Tanguy Pérennou^{1,2} Amine Bouabdallah^{1,2} Anna Brunstrom³ Johan Garcia³ Per Hurtig³

¹CNRS ; LAAS ; 7 avenue du Colonel Roche, F-31077 Toulouse, France

²Université de Toulouse ; UPS, INSA, INP, ISAE ; LAAS ; F-31077 Toulouse, France

³University of Karlstad, Sweden

Abstract—Distributed applications and transport protocols communicating over a satellite link may react very strongly to conditions specific to that kind of link. Providing a evaluation framework to allow tests of real implementations of such software in that context is quite a challenging task. In this paper we demonstrate how the use of the general-purpose KauNet IP-level emulator combined with satellite-specific packet loss patterns can help by reproducing losses and delays experienced on a satellite link with a simple Ethernet LAN setup. Such a platform is an essential tool for developers performing continuous testing as they provide new features for e.g. video codecs or transport-level software like DCCP and its congestion control components.

Index Terms—IP-level Emulation, DVB-H, Markovian LMS channel

I. INTRODUCTION

Testing application-level or transport-level software under satellite conditions is a challenging task. For example, how can one evaluate the correctness or the performance of a video codec using RTP/UDP over a satellite link that introduces specific delays and bit error patterns on the exchanged packets?

The central mechanism, the core idea that makes the essence of the codec (for instance) under development can generally be validated using well-established and dependable simulation tools such as Opnet or ns-2. But once it gets implemented as a Windows or Linux application or a kernel module, such simulators cannot be used anymore to test the application, because these simulators run as single processes integrating an elaborated model of the mechanism. Modern developers willing to perform continuous testing on the features they deliver need to have a practical and highly available tool to test their implementation. This tool must simulate the behavior of a satellite link, as opposed to the full simulation where both the satellite system and the endpoints are simulated. This link must be simulated because a real satellite link is not easily available, and would anyway be impossible to control. Such a simulation environment is generally referred to as a *satellite link emulation*.

A number of researchers have proposed the use of link-level emulators and IP-level emulators. The main advantage of network level emulators is their ease of deployment and usage. KauNet is a recent extension of the well-known Dummynet IP-level emulator that makes it deterministic regarding packet losses and offers a high level of control and accuracy through its notion of patterns. Working with patterns makes the testing task easier. For instance, a developer wanting to know how a codec behaves when the receiver is in a forest simply needs to load a forest-type packet loss pattern into KauNet. If he wants

to compare the performance of several codecs under stormy weather conditions, he just has to load the appropriate bit error or packet loss pattern into KauNet and compute the PSNR of testbench videos for each codec. The main difficulty left at the tester's hand is to find the appropriate pattern, or to design a new one when none is available.

The main contributions of this paper are listed below.

- We define the architecture of a satellite-specific testing platform for developers of real application-level or transport-level implementations. This platform is an instantiation of the KauNet emulator augmented with satellite-specific patterns. KauNet design and functionalities are discussed in Section II. The companion satellite-specific patterns used by KauNet have been produced using the Markovian Land Mobile Satellite model after a measurement campaign led by CNES, the French Space Agency. The model, traces and how they were distilled to KauNet patterns are discussed in Section III.
- We demonstrate the practicality of the platform by implementing a demonstration software dynamically coupling the trajectory of a mobile node on a large-scale map with appropriate packet loss patterns over time (Section IV).

II. IP-LEVEL EMULATION WITH KAUNET

As discussed in the introduction IP-level emulation is an effective performance evaluation method for application and transport layer software targeting satellite environments. Our work is based on KauNet, which enables emulation-based experiments with a high degree of control and reproducibility of the emulated conditions. Before describing the details of satellite link emulation, an overview of the KauNet emulation system is provided in this section. An illustration of the system is provided in Figure 1.

The design of KauNet is centered around a number of pattern-handling extensions to the well known kernel-based Dummynet emulator [14], together with user-space programs for pattern creation and management. The use of Dummynet as a starting point provides a stable codebase that has been in wide-spread use for several years, as well as the integration with the `ipfw` program for emulation setup and management. Dummynet has the ability to lose packets, and to apply bandwidth restrictions and delays to the packets thus emulating the desired link or network conditions. KauNet extends these abilities by also including the ability to introduce bit-errors and packet reordering. Furthermore, KauNet allows deterministic packet losses in addition to the probabilistic losses provided

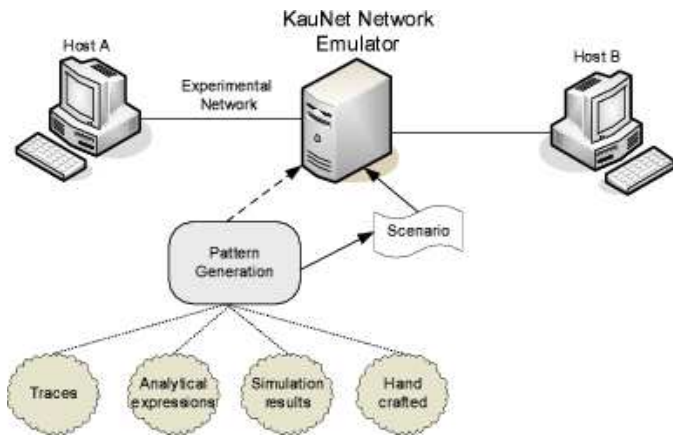


Figure 1. Overview of the KauNet Emulation System.

by DummyNet. In fact, KauNet allows bit-errors, packet losses, packet reordering and delay and bandwidth changes to be exactly and reproducibly controlled on a per-millisecond or per-packet basis with the use of patterns.

Compressed patterns are created ahead of time and then inserted into the kernel space under the control of KauNet. During emulation these patterns are then played to control the emulated behavior. For emulations controlled on a per-millisecond basis KauNet operates in time-driven mode and for emulations controlled on a per-packet basis in data-driven mode. In time-driven mode the index for the pattern is advanced once per millisecond regardless of whether or not any data is transferred. For time-driven bit-errors the amount of movement by the index is coupled to the bandwidth restriction used in the same pipe. This restriction indicates how many bits to process per millisecond. Other kinds of time-driven patterns do not require a bandwidth restriction to be set since they work only on a millisecond level and individual bits do not need to be accounted for. For data-driven patterns the index move forward one step for each packet, except for bit-error patterns where the index is increased according to the number of bits in the packet.

With regards to emulating multiple connections or links, KauNet inherits the DummyNet firewall rule capabilities that are specified with `ipfw`. These capabilities make it easy to create multiple emulated flows, emulating for example the effective bandwidths and delays for a set of nodes in a mobile network. While DummyNet allows emulated conditions to be changed during the run of the emulation, these changes must be done using command line tools (typically under the control of a script). Using regular DummyNet, it is thus very difficult to exactly control or reproduce an experimental run. As discussed above, KauNet provides enhanced functionality in this area through the use of patterns. However, the use of patterns and command line control can be effectively combined in KauNet. Even though the fine-grained behavior of the emulation is under the control of the patterns, higher-level dynamic events can still be incorporated by using the command line tools to dynamically switch between multiple patterns. This will be

further illustrated later in the paper.

As illustrated in Figure 1, the KauNet system is flexible with regards to the origin of the patterns, which can be created from collected traces, previous simulations or analytical expressions. In order to simplify creation and management of patterns, a command line tool has been developed to create and manage patterns. The `patt_gen` tool can generate patterns according to several random distributions and their parameters. It is also capable of importing uncompressed pattern descriptions from simple text files. These text files can be generated by arbitrarily complex models, off-line simulators or trace collection equipment.

One pattern is necessary for each of the controllable parameters, i.e. bit-errors, packet losses, packet reordering, bandwidth changes and delay changes. Since many experiments require simultaneous control of several parameters, multiple patterns need to be managed. A satellite link may for example introduce specific delay characteristics as well as specific bit error or packet loss patterns. To simplify the management of patterns, KauNet supports the grouping of multiple patterns into so-called scenarios. A scenario is a combination of several patterns with some additional meta information. Scenarios are also created and managed using the `patt_gen` utility.

In this work KauNet is used for satellite link emulation, but its scope of applicability is of course much wider. Other uses of KauNet include transport protocol implementation verification (see for example [5]) as well as different transport layer and application layer performance evaluations (see for example [12], [8]).

III. SATELLITE LINK MODEL AND PATTERNS

In this Section we describe the Markovian Land Mobile Satellite (LMS) satellite link model that we used and how we obtained the corresponding KauNet packet loss patterns.

A. LMS Channel Model

Concerning the LMS channel, the model is a three-state model (LOS/Shadowing/Blockage) based on Markov chains integrating three scales of propagation effects [6]. The large scale effects correspond to the changes between the three states. The mid scale effect or “Slow” fading corresponds to shadowing. The small scale effect or “Fast” fading is due to multipath. Based on extensive measurement campaigns, [6] defines transition matrices between the three states for five environments (Open, Intermediate Tree Shadowed, Heavy Tree Shadowed, Suburban, and Urban), for elevation angles above 40° . For each state, slow fading and fast fading variations are defined according to a Loo distribution [13]. The Loo distribution considers that the received signal is made up of the sum of two components, the direct signal and multipaths. Within one state, the dynamics of the shadowing affecting the direct path is given by its correlation distance, equal in e.g. Intermediate Tree Shadowed environment to 1.5 m.

Moreover, a perfect demodulation is assumed. We also assume a TDM signal, which is expected to be representative for both TDM and OFDM cases. Indeed, for OFDM signals,

the symbol duration is equal to a few hundreds of microseconds, which corresponds to a rather stable propagation attenuation even for the highest speed considered. Noise level adjustment compared with LOS signal level is based on the assumption of a margin of 10 dB, which corresponds to a typical link budget.

B. LMS Channel Data

According to the above LMS channel model, the receiver carrier-to-noise ratio (C/N) has been simulated by CNES (French Space Agency) over a distance of 1 km, in order to limit the duration of the simulations performed in the upper layers. The simulator accounts for the modulation, the propagation part, and the demodulation. The considered simulation distance is split into 220000 segments of 0.004545 m each. The simulated data, represented in Figure 2, cannot be considered as perfectly representative for the complete environment but give a good idea of the channel behavior. The distance may then be covered at various speeds.

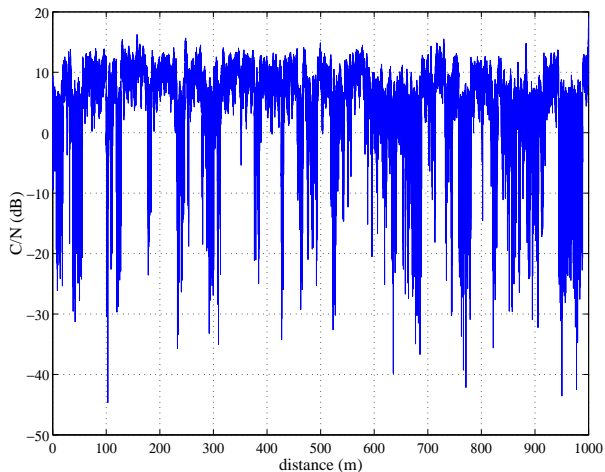


Figure 2. Evolution over 1 km of the receiver signal-to-noise ratio for the Intermediate Tree Shadowed area

As can be seen on Figure 2, the received carrier-to-noise ratio ranges from approximately 10 dB to less than -25 dB during deep shadowing. Moreover, deep shadowing effects appear sometimes over distances larger than 50 m.

C. Mapping Channel Data to IP Packet Losses

The time elapsed in each segment of the above channel varies according to the speed of the mobile receiver. Then, in each segment, we computed the bit sequences and the corresponding bit reception statuses (success or failure) according to the received carrier-to-noise C/N data and the land mobile receiver speed. After that, we mapped the physical channel data described above to packet losses at the IP level through a stand-alone satellite simulator that we have developed separately in the context of a research study with CNES.

The protocols stack simulated is largely inspired from the DVB-H protocols stack. All components from physical to IP

layer are simulated using NEWCOM's IT++ libraries [1]. The simulator includes a QPSK modulator/demodulator, a block interleaver to reduce errors burst length and the 3GPP turbocode coder/decoder to correct errors. The turbo coder operates on the usual MPEG2 transport stream (TS) packets that come across most of the DVB systems. Above MPEG2-TS, there is the Multi-Protocol Encapsulation (MPE). It is the adaptation layer which aims at fitting the variable-length IP packets into the fixed-length MPEG2-TS packets. MPE adds 12 bytes to the header and appends a 4-byte long CRC to detect errors and discard erroneous packets. At this layer an optional erasure correcting code (MPE-FEC) is also available. Full details can be found in [3].

IP-level packet loss data consists of sequences of logical markers which inform whether IP packets are received or not. In practice, CNES has been providing us C/N data only for the Intermediate Tree Shadowed, Heavy Tree Shadowed and Suburban environments; Urban areas can be considered as Intermediate Tree Shadowed areas, while Open areas can be considered as suburban areas. Three distinct land mobile speeds were considered: 1.5, 15 and 30 m/s. For each given area type and mobile speed combination, a specific KauNet packet loss pattern has been produced. The pattern must be loaded in the time-driven mode.

IV. MAPPING GROUND CONDITIONS TO KAUNET PATTERNS

The use of KauNet and Markovian-LMS-derived patterns makes it easy to emulate a satellite link at the IP-level. This section describes the demonstrator that was developed by first year Master students in Aeronautics and Space Engineering at ISAE over only a few weeks.

The demonstrator is named HoSE (H.264 over a Satellite Emulator) and is mainly composed of (see also Fig. 3):

- an Interactive Travel Map (ITM), where the user defines a trajectory followed by a virtual mobile node, typically by car;
- a pattern loader colocated with KauNet, that dynamically loads a packet loss pattern reflecting the speed and the ground conditions encountered by the mobile node;
- a Darwin streaming server and a VLC client connected by RTP/UDP/IP and using the H.264 codec; the video stream packets generated go through KauNet and are diverted into a pipe where the pattern has been loaded.

The interactive travel map and the pattern loader are described hereafter. We then describe how a demonstration unfolds.

A. Interactive Travel Map

The Interactive Travel Map (ITM) is a Java graphical user interface allowing the user to define the path he wants to virtually follow on a regional map divided into areas and featuring available roads. This map contains meta-data describing the nature of the areas: sea, plain, countryside, forest, town, tunnel; and roads: speed limit, one-way. The user simply selects departure and destination, and then the ITM software

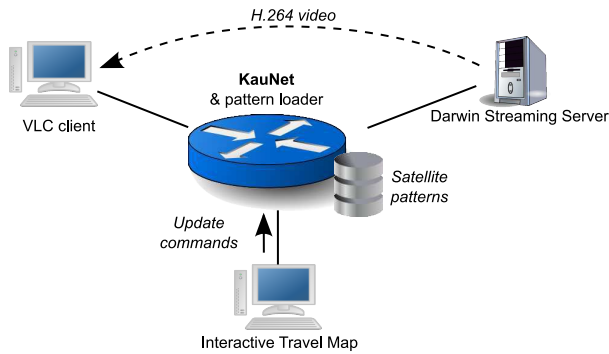


Figure 3. Demonstrator architecture.

computes the shortest path taking into account speed limits and one-way constraints. This is illustrated by Fig. 4.

Once the path is computed, the ITM sends update messages to the pattern loader so that KauNet uses a packet loss pattern reflecting the nature of the area newly entered by the mobile node and its current speed.

At any time after selecting the departure and destination, the user can interactively select a new destination for which the ITM computes the new path, merging or aggregating it to the current path without interrupting the demonstration. This allows for instance the user to return to an interesting zone at will, e.g. to illustrate a specific point to the audience.

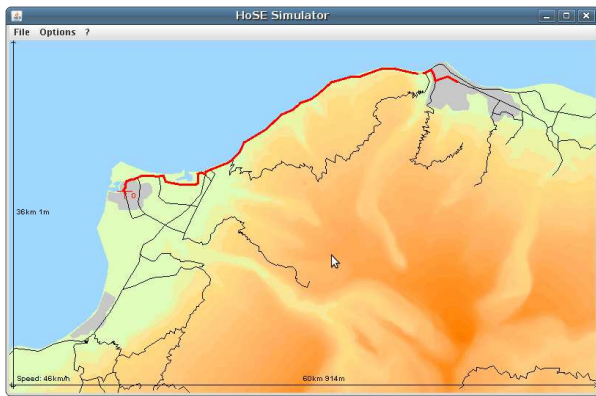


Figure 4. Interactive Travel Map

B. Pattern Loader

The pattern loader is a very lightweight component that simply interprets the ITM update messages in order to load the appropriate KauNet packet loss pattern. To keep things flexible, the update messages contain meta-data information extracted from the map describing the nature of the newly entered area, as well as the mobile node speed. In preliminary versions of the demonstrator, we have successfully used this information to retrieve Uniform or Gilbert-Elliott-derived packet loss patterns. In the version of the demonstrator described here, the same information allows to retrieve a Markovian-LMS-derived pattern.

First, the area nature information is looked up in a mapping table in order to find the appropriate Markovian LMS area. Then the speed is taken into account to compute the name of the KauNet packet loss pattern file. Finally the pattern is loaded into KauNet via a call to the `ipfw` command. Table I gives the currently used mapping table.

Table I
GROUND CONDITIONS MAPPING

| Nature of the area | Markovian LMS area |
|--------------------|----------------------------|
| Sea | Suburban |
| Plain | Suburban |
| Countryside | Intermediate Tree Shadowed |
| Town | Intermediate Tree Shadowed |
| Tunnel | Heavy Tree Shadowed |
| Forest | Heavy Tree Shadowed |

C. Demonstration Scenario

The typical scenario for a demonstration is as follows:

- Connect the Darwin streaming server and the VLC client to the KauNet router.
- Start the Darwin streaming server.
- Start the VLC client.
- At this time, KauNet is in its initial state and will not lose any packet. Configure KauNet so that the traffic from the server to the client is diverted to a specific server-to-client pipe. Set an appropriate delay on that pipe.
- Start the pattern loader for the server-to-client pipe on the KauNet machine.

At this point, the initialisation tasks are finished. You can call the attention of the audience and begin the real demonstration:

- Start the Interactive Travel Map and load a map. Define a trajectory on the ITM, so that different areas are crossed.
- Stream a video to the VLC client and comment the changes in the video quality according to the area changes.

Note that a non-expert audience may have some trouble interpreting the mere visualisation of the video: a codec like H.264 introduces its own mechanisms and makes it difficult to see the difference between a lot of packet loss or rare packet losses impacting important images and durably degrading the video even if no subsequent packet loss occur. Therefore, it may also be appropriate to show the evolution of the packet loss rate in real time.

V. RELATED WORK

Satellite link emulation can be performed at various levels of abstraction. When link layer emulators are used real satellite hardware is used and all lower layer protocols are implemented. In this case only the air interface is emulated using a wireless channel model. For instance, the low level emulation platform developed by Alcatel in the IST GEOCAST project completely implements the software and hardware of the Alcatel 9780 system architecture. The air link is the only emulated part of the system. Reference results for this platform are available in [10]. Low level emulation systems can provide

accurate results and are needed for verification of lower layer protocols. However, they are also complex and expensive to develop, maintain and use. Thus for testing of application and transport layer software in satellite conditions, IP-level emulation offers an appealing alternative.

When IP-level emulation is used the behavior of the satellite system is captured on a higher abstraction layer and only its effect on the data traffic, in the form of delay characteristics, loss patterns, etc., is emulated. The details of the lower layer protocols do not need to be implemented as long as their high level effects can be captured. This provides a more easily used and cost effective emulation. A number of generic IP-level emulators exist that can be used in particular for satellite link emulation, such as Dummynet [14], NIST Net [4] or NetEm [11]. For instance, the performance of VoIP with DCCP/CCID4 protocol has been studied using Dummynet over long-delay links [2]. NASA itself has developed a satellite channel emulator (CE) based on NetEm [15]. As compared to these IP-level emulators, KauNet's pattern-based approach provides a more fine grained and flexible control over the emulated network. A more in depth discussion on the specifics of KauNet in relation to other emulators can be found in [7]. Another approach consists in exploiting discrete-event simulators like ns-2 in real-time, as done in NCTUNs [17], which provides a DVB-RCST link. This feature has been used in the NovaVANET wireless vehicular ad hoc network simulator [16].

One limitation of KauNet and other generic IP-level emulators is that the emulated link behavior is typically assumed to be independent of the traffic traversing the link. In [9], it is illustrated how interactions with data traffic can be captured in the emulation of a DVB-RCS satellite channel using the NPBox emulator. DVB-RCS uses dynamic resource admission control based on a demand-assignment multiple access protocol, and the delay and bandwidth characteristics experienced over the link by a data flow are thus dependent on the traffic in the system. NPBox can react to incoming traffic based on a programmable finite-state-machine like approach. As compared to KauNet, the NPBox approach provides a more powerful and advanced emulation system. However, it is also more complex and difficult to use. As the characteristics of the LMS channel emulated in out work do not depend on the traffic, KauNet is a more appropriate emulation system.

VI. CONCLUSION

In this paper we showed that the appropriate use of the KauNet IP-level emulator provides a practical satellite link emulation for software developers working at the application or transport level. We focused on how to obtain packet loss patterns from a Markovian LMS channel model, and we described the HoSE demonstrator which makes an interactive use of such patterns. This demonstrator allows to illustrate the quality of streamed video reception with the H.264 codec, the receiver dynamically moving around in different environments.

Future work consists in improving practical aspects of the HoSE demonstrator, such as real-time visualisation of the

current packet loss pattern, the current packet loss rate and the current image type (I, B or P). We are also extending kauNet with *trigger* patterns that can be used to convey contact opportunities. Such a feature is needed for the emulation of delay/disruption tolerant networking scenarios, such as satellite networks with periodic connectivity.

Acknowledgements

We wish to thank Marc Aasebo, Abdendi Bouthiche, Antoine Cavallès and Gonzague Fontaine, who were the interns involved in the HoSE project, as well as Jérôme Lacan and Laurent Dairaine who co-supervised it. We also thank Gaël Scot and Caroline Amiot-Bazile who provided us LMS channel data on behalf of CNES (French Space Agency). This work was partly supported by the European Commission in the framework of the FP7 Network of Excellence in Wireless COMMunications NEWCOM++ (contract n. 216715).

REFERENCES

- [1] C++ library of mathematical, signal processing and communication routines. <http://itpp.sourceforge.net>.
- [2] H. V. Balan, L. Eggert, S. Niccolini, and M. Brunner. An Experimental Evaluation of Voice Quality over the Datagram Congestion Control Protocol. In *Proceedings of IEEE INFOCOM 2007*.
- [3] A. Bouabdallah, M. Kieffer, J. Lacan, G. Sabeva, P. Duhamel, G. Scot, and C. Bazile. Evaluation of Cross Layer Reliability Mechanisms for Satellite Digital Multimedia Broadcasting. *IEEE Trans. on Broadcasting*, 53(1):391–404, March 2007.
- [4] M. Carson and D. Santay. NIST Net: A Linux-based Network Emulation Tool. *ACM Computer Communication Review*, 33(3):111–126, 2003.
- [5] E. Conchon, J. Garcia, T. Pérennou, and M. Diaz. Improved IP-level Emulation for Mobile and Wireless Systems. In *Proc. of IEEE Wireless Communications and Networking Conference (WCNC07)*, Hong Kong, China, March 2007.
- [6] F. Pérez Fontán, M. Vázquez-Castro, C. E. Cabado, J. P. García, and E. Kubista. Statistical Modelling of the LMS Channel. *IEEE Trans. on Vehicular Technology*, 50(6):1549–1567, November 2001.
- [7] J. Garcia, E. Conchon, T. PÁrennou, and A. Brunstrom. KauNet: Improving Reproducibility for Wireless and Mobile Research. In *MobiEval Workshop at ACM MobiSys 2007*, San Juan, Puerto Rico, June 2007.
- [8] J. Garcia, P. Hurtig, and A. Brunstrom. The Effect of Packet Loss on the Response Times of Web Services. In *Proc. of 3rd International Conference on Web Information Systems and Technologies (WebIST2007)*, Barcelona, Spain, March 2007.
- [9] M. Gineste and J. Garcia. Programmable active emulation of wireless systems - A DVB-RCS Example. In *Proc. of the 4th International workshop on Wireless Network Measurements (WiNMe 08)*, Berlin, Germany, March 2008.
- [10] M. Gineste and P. Sénac. Quality of Service Solutions for Satellite Communication. In *Proc. of the 4th IEEE International Conference on Networking (ICN 2005)*, April 2005.
- [11] S. Hemminger. Network Emulation with NetEm. In *Proc. of the Linux Australia Conference (linux.conf.au 2005)*, Canberra, Australia, April 2005.
- [12] P. Hurtig and A. Brunstrom. Enhancing SCTP Loss Recovery: An Experimental Evaluation of Early Retransmit. *Elsevier Computer Communications*, 31(16):3778–3788, October 2008.
- [13] C. Loo. A Statistical Model for a Land Mobile Satellite Link. *IEEE Trans. on Vehicular Technology*, VT-34(3):122–127, August 1985.
- [14] L. Rizzo. Dummynet: A simple approach to the evaluation of network protocols. *ACM Computer Communication Review*, 27(1):31–41, January 1997.
- [15] R. A. Slywczak and T. J. Kollar. Modeling Space-Based Communications using a Dynamic Channel Emulator. In *Proc. of AIAA Modeling and Simulation Technologies Conference and Exhibit*, Keystone, USA, August 2006.

- [16] S.-Y. Wang and C.-L. Chou. NCTUns Simulator for Wireless Vehicular Ad Hoc Network Research. In J. N. Turner and C. S. Boyer, editors, *Ad Hoc Networks: New Research*. Nova Science Publishers, 2009.
- [17] S.-Y. Wang and Y.-B. Lin. NCTUns network simulation and emulation for wireless resource management. *Wireless Communications and Mobile Computing*, 5(8):899–916, 2005.