

An Integrated Prototyping and Simulation Architecture for Space Specific Protocol Developments and Verifications (Extended Abstract)

Marc Emmelmann
Fraunhofer Institut FOKUS
Competence Center for Advanced Network Technologies
Berlin, Germany
Email: emmelmann@ieee.org

Abstract—The design and development of future satellite networks includes preferably, besides a thorough system definition phase, the simulation of the entire satellite network with special interest on the network behavior and the (prototype based) development of system components. Especially with respect to the development of MAC protocols which should preferably included at an earliest stage in the network simulation workflow, an integrated approach of simulating and rapid prototyping of these protocols is desirable.

This paper presents such an integrated workflow to develop future satellite networks and protocols. It consists of a simulation based part including the well-known toolsets STK and OPNET, and a part embedding a rapid prototyping and testing environment developed by Fraunhofer FOKUS.cats. Emphasis is put on the latter prototyping and testing unit and its possible use cases, e.g., protocol development (as done within the ATM-Sat project [1]) or mere satellite channel emulation (as used for the evaluation of connecting the ISS-Columbus module via ATM to earth).

I. INTRODUCTION

Space system engineers will increasingly face the need to reduce development costs while still designing failure safe systems. One possible approach to fulfill this requirement is the usage of commercial off the shelf equipment [2] in combination with standard software and operating systems (e.g.: flight linux [3]), and the re-use of previously developed soft- and hardware modules. Additionally, a time-saving development cycle helps to meet this requirement.

Especially during the system design phase, intense simulation of the entire satellite system including existing network hard- and software components as well as the verification of the developed system parts is beneficial. This paper focuses on one possible simulation and prototyping based development environment which consists of commercial off the shelf equipment.

The paper is structured as follows: Section II describes the architecture of this integrated system and its two components, i.e. the simulation driven work flow and the actual rapid prototyping and verification environment for protocol development. Two possible use cases (the development of a

MAC protocol for an ATM-based LEO multimedia satellite system, and an evaluation of a possible future connectivity of the ISS via ATM) are illustrated in section III. Finally, section IV summarizes the proposed architecture.

II. SYSTEM ARCHITECTURE

The promoted integrated simulation and prototyping / development architecture consists of a mainly simulation driven workflow using the well-known tool OPNET [4] and an implementation focused approach which incorporates a development environment created at Fraunhofer FOKUS.cats (Fig. 1).

Starting with the definition of the target system (including satellite constellations, protocols to be developed, etc.), STK is used to gain either a description of the satellites and their orbits (as an input to OPNET) or a mere table with elevation angle and propagation delay data for a specific link (as an input to the rapid prototyping / implementation unit). The simulation driven part focuses on the evaluation of the entire satellite network to be developed including the implementation of new protocols, whereas the implementation driven workflow focus on the development of new protocols with respect to the target system. Protocol specific, critical algorithms can be exchanged in between the two workflows.

A. Simulation on Network Level

The OPNET network simulator has its strengths in simulating large networks including specific user / application characteristics [4]. In addition, it allows the evaluation of standard protocols over satellite networks, e.g. TCP [5], and the integration of future satellite network specific protocols, e.g. used on the MAC layer [6]. The latter may mostly be implemented in standard C which allows an easy exchange of critical, protocol specific algorithms with the rapid prototyping environment targeted at the final system implementation. As OPNET itself is known to the research and mission engineering community, the tool is not further discussed in this context.

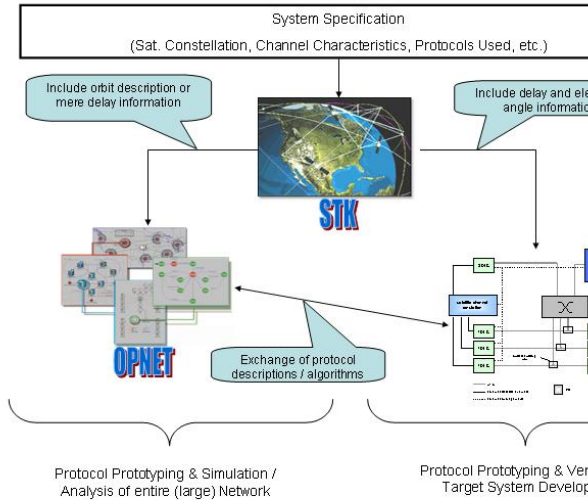


Fig. 1. Integrated Development Flow

B. Rapid Prototyping and Verification of Protocols

The second workflow (right half of Fig. 1) focuses on the development of new protocols with respect to the target system. The discussed environment developed by Fraunhofer FOKUS.cats consists in its core of a *satellite channel emulator* (SCE), a *protocol prototyping and development entity* (PDE), and a *control station* (CS). Figure 2 depicts the components as used within a project targeted at the development of a MAC protocol for ATM-based LEO satellite networks [1]; the PDE is used for the development of the latter protocol and acts therefore as a data link unit (DLC). For testing and demonstration purposes, any kind of user terminal with an appropriate network interface (e.g. a Windows PC or workstation) may be connected to the PDE.

Both functionalities, i.e. satellite channel emulation and the implementation of protocols, are split for better independence. This approach does also allow the SCE to be used to merely connect two networks while imposing a satellite specific delay and error rates to the forwarded packets (this functionality is similar to the Ohio Network Simulator (ONE) [7]).

1) *Satellite Channel Emulator*: The SCE is configurable via snmp and adds a variable propagation delay to the forwarded packets. Additionally it may either corrupt packets by a predefined BER or discards packets according to this error rate. As an input, the SCE accepts STK-based AER tables enhanced with precalculated error rates for certain times (or elevation angles). Depending on the configuration, it also

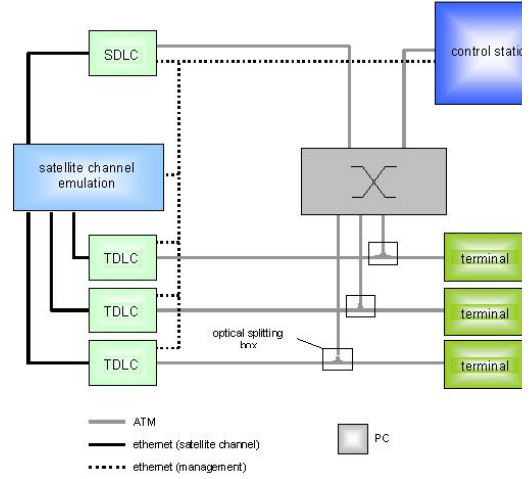


Fig. 2. Protocol Implementation and Channel Emulation Environment (set up for a MAC development for satellite ATM networks)

implements contention for terminals transmitting at the same time. The functionality is independent of the network interface to connect the SCE to the DLC. If the SCE is used in conjunction with a protocol development entity (here used for DLC development), MAC packets are encapsulated in Ethernet frames for transmission between SCE and DLC. If used in bridging mode, any network interface may be used.

2) *Protocol Prototyping Entity*: The protocol prototyping entity (PDE) consists of a FreeBSD [8] based system which is used as a development platform. Its concept known as “netgraph” [9] allows to link different protocol implementation entities together thus making it possible to easily replace parts of an existing protocol stack. Protocols are implemented in C which makes it possible to exchange source code of critical algorithms with OPNET [4].

3) *Control Station*: The control station (CS) initializes the SCE and PDE and acts additionally as an NTP-server which is used by the latter two units for time synchronization.

III. USE CASES

This section will deepen two possible use cases of the described prototyping and simulation architecture: its usage within an entire MAC protocol development and verification workflow, and its mere usage as a satellite channel emulator to analysis the impact of variable delays and error rates on an ATM connection between the ISS Columbus module and the earth.

A. ATM-Sat MAC Protocol Development

Fraunhofer Fokus, German Aerospace Agency (DLR), and Tesat Spacecom jointly conducted a project which targeted at the development of an ATM-based multimedia LEO satellite network. The described prototyping and simulation environment was used within this project for the development of the MAC protocol. This use case illustrates the most complex application of this integrated development environment.

The entire work flow (simulation with OPNET and protocol prototyping and testing), as illustrated in Fig. 1, was applied within the project. Simulations were conducted at an early stage within the project and are well documented in [5], [6], [10].

The usage of each component of the development and testing system is shortly explained from a top-down approach. Two user terminals (any kind of terminal, e.g. a PC or workstation) are supposed to communicate via an ATM-based satellite network. They are connected via plain ATM to their terminal data link control (TDLC) entity. The latter may be seen as either an external VSAT system or an adapter card integrated in the final terminal (reducing the physical connection to a service access point in the protocol stack). The TDLC is based on a standard PC running FreeBSD 5.0-current [8] and implements the newly developed MAC protocol. The FreeBSD netgraph concept [9] allows an easy and fast integration of the new protocol in any existing protocol stack thus allowing the developer to concentrate on the implementation of the MAC protocol. When running and testing the protocol, the TDLC receives ATM cells from its interface to the terminal and transmits them via the MAC protocol to the satellite at a certain time. The transmission is actually emulated by encapsulating the transmitted MAC structure with an additional time stamp into a plain Ethernet package. The latter is forwarded to the SCE entity over a wired ethernet link. The satellite channel emulator receives all MAC packages from all terminals and merges them according to the time stamp included in the encapsulation package. Due to the time stamps, the SCE is well able to handle contention and additionally adding a predefined BER. The merged MAC packets are forwarded after a predefined, variable delay to the satellite (again by Ethernet encapsulation). The satellite DLC decapsulates the received frame and handles it in its DLC implementation.

The ATM switch depicted in this configuration is used to forward ATM cells received by the SDLC of the sending terminal to the SDLC of the receiving terminal (the two DLCs are identical for the depicted scenario). Additionally, the switch is used for measurement purposes. The control station acts as a boot-server for the DLCs and the SCE. In addition, the CS runs an NTP-daemon which synchronizes the DLC and SCE entities with an accuracy of less than $50\mu s$. Final (application specific) experiments may be run over the (partially) emulated satellite connection between two terminals.

B. ATM connectivity between ISS Columbus and Earth

The prototyping and simulation environment has been reduced to the mere satellite channel emulator for experiments conducted with Astrum Spacecom to test a possible connectivity of the ISS Columbus module with the earth. Therefore, the SCE was configured to act as a pure bridge imposing a variable delay and BERs. The FreeBSD netgraph concept allowed an easy exchange of the "ATM-encapsulation part" with a pure ATM driver thus allowing user terminals to directly communicate via the SCE.

IV. CONCLUSION

This paper described one approach to integrate both a simulation driven and rapid prototyping based development cycle of new space specific protocols. The entire system consists of commercial off-the-shelf equipment which can be configured for various use cases starting from mere satellite channel emulation up to the development and testing of upper layer protocols. The possible exchange of critical protocol algorithms between the simulation driven workflow of the entire satellite network and the target system specific implementation workflow allow the testing of new developments at various system levels at an early stage.

REFERENCES

- [1] (2003) Atm-sat: Atm-based multimedia communication via leo satellites. [Online]. Available: <http://www.fokus.gmd.de/research/cc/cats/satellite/Projects/ATM-Sat>
- [2] (2003) The micro-satellite BIRD. [Online]. Available: <http://spacesensors.dlr.de/SE/bird/>
- [3] (2002) NASA FlightLinux. [Online]. Available: <http://flightlinux.gsfc.nasa.gov>
- [4] (2003) OPNET — optimum network performance. [Online]. Available: <http://www.opnet.com>
- [5] M. Emmelmann, "Effects of advertised receive buffer size and timer granularity on TCP performance over erroneous links in LEO satellite networks," in *Proc. IEEE Globecom '02*, Taipei, Taiwan, Nov. 17–21, 2002.
- [6] J. Bostic, "Report on OPNET simulators," German Aerospace Agency (DLR), Oberpfaffenhofen, Germany, Tech. Rep. DLR-IB, IB 554-01/02, 2002.
- [7] M. Allman, A. Caldwell, and S. Ostermann, "ONE: The Ohio Network Emulator," Ohio University, Tech. Rep. TR-19972, Aug. 1997. [Online]. Available: <http://roland.lerc.nasa.gov/~mallman/papers/em.ps>
- [8] (2003) The freebsd homepage. [Online]. Available: <http://www.freebsd.org/>
- [9] A. Cobbs. (2002) All about netgraph. [Online]. Available: <http://www.daemonnews.org/200003/netgraph.html>
- [10] H. Bischl, H. Brandt, A. Dreher, M. Emmelmann, A. Freier, B. Hespeler, F. Krepel, E. Lutz, W. Milcz, L. Richard, P. Todorova, and M. Werner, "ATM-sat: ATM-based multimedia communication via LEO-satellites," German Aerospace Agency (DLR)," Final Report, Dec. 2002.