Satellite Network Simulator 3

Workshop on Simulation for European Space Programmes (SESP)
24-26 March 2015

ESA-ESTEC, Noordwijk, The Netherlands

Jani Puttonen(1), Budiarto Herman(1), Sami Rantanen(1), Frans Laakso(1), Janne Kurjenniemi(1)

(1)Magister Solutions Ltd
Sepänkatu 14, FIN-40720 Jyväskylä, Finland
Email:{firstname.lastname}@magister.fi

INTRODUCTION

Simulators, real-time emulators and real test-beds are possible tools for networking research. Simulators often provide easily controlled environment, configurability, repeatability, and scalability, but the accuracy of the simulator implementations and simulation results are often questioned. By definition, simulator implements models of real life protocols or applications which imitate reality. Simulation models have to balance between accuracy and complexity. Needed accuracy depends on the use case: in certain cases the simulation model may be implemented according to specifications, but it may also be a lot simplified only trying to capture the effects of a real implementation. A higher level of accuracy usually requires a more complex simulation models, and thus increases the simulation run time.

Real test beds, on the other hand, provide a high level of accuracy by allowing the use of real devices, real interaction with the operating system (OS), remote management and ability to run real applications. However, the repeatability and scalability may be somewhat limited. Moreover, the arrangement of mobility may be challenging. Also, one might find it difficult to acquire cutting edge software or hardware to build the test beds for research.

Emulations lie in between the above two extremes. Emulations may provide the high accuracy of test beds while providing a more controlled environment. Emulator implementations imitate real systems (protocols, applications) in a way that they can replace real software or hardware implementations. Thus, the accuracy level of emulation implementations is (usually) closer to reality if compared to simulation models. Simulations however, still provide the benefits of more easily built bigger scenarios, easy system-wide statistics gathering, and repeatability.

Simulators, emulators and test beds all have their specific role in networking Research and Development (R&D), which may be complementing with each other. Simulations may be utilized for analysis and research, when there is no software or hardware available. Simulators provide also rapid development cycle, thus a means of faster development of new models and performance tests. Emulators have, on the other hand, a significant role in more realistic protocol testing, where the implementations have to be more according to specifications. Emulators may be utilized as replacements of real software or hardware implementations, when they do not exist or are otherwise unavailable. Real test beds provide testing of real developed software and manufactured hardware, which have great importance in e.g. interoperability testing.

Satellite simulators and emulators owned by satellite vendors, operators and research organizations are usually proprietary, generally tailored for special purposes and/or may be based on commercial licenses. The Open Source Satellite Simulator (OS3) [1] is a satellite simulator based on OMNet++, complementing the INET framework. Its focus has been on satellite mobility, satellite constellations, and inclusion of weather data and channel models, but not on satellite communication protocols. A DVB-RCS model [2] has been developed on top of Network Simulator 2 (ns-2). However, it is clearly lacking some essential features, and it does not provide DVB-RCS2 or DVB-S2 models. In addition, it is based on ns-2, which has problems related to scalability, modularity, maintenance, and so on.

Besides the open source solutions mentioned above, there are several proprietary solutions: e.g. DLR (German Aerospace Center) has an OMNet++ based satellite simulator called *SatSim*, and Thales Alenia Space and CNES have an ns-2 based satellite simulator called *SMACSat* (System and MAC Satellite Simulator).

As for emulators, one example is *OpenSAND* [3], which is a DVB-RCS and DVB-S2 emulator developed initially by Thales Alenia Space. OpenSAND has been released with GPLv3 license and is freely available. *AINE* (Advanced IP Network Emulator) [4] is a proprietary emulator developed by Indra for both IP and link layer protocol emulation. A DVB-RCS2 emulator has also been developed within ESA project “DVB-RCS Next Generation Support and Verification Testbed” by DLR and VeriSat.

In this article, Satellite Network Simulator 3 (SNS3) is presented. SNS3 focuses on geostationary multi-spot beam satellite networks and DVB-RCS2 and DVB-S2 protocol stacks. SNS3 has been developed within ESA ARTES 5.1 project to function, e.g., as a common platform for ESA activities. It is intended to reduce consecutive simulation software development efforts. SNS3 may also run in real-time emulation mode, enabling both the usage of pure simulations and real-time emulations with the same implementation.

SATELLITE NETWORK SIMULATOR 3

Satellite Network Simulator 3 (SNS3) [5] is based on *Network Simulator 3* (*ns-3*) [6] platform, which is modular, widely used, scalable and fast open-source packet/system level network simulator for networking R&D. SNS3 models interactive multi-spot beam satellite networks with geostationary transparent star payload.

A single multi spot-beam satellite at geostationary orbit (GEO, at about 35786 km altitude) is considered. The system coverage is Europe using a single satellite located at 33 degrees East using Ka-band frequency (from 26.5 to 40.0 GHz) on the feeder and user links. As illustrated in Fig 1, the user link coverage consists of 72 spot-beams and the user beams are served by 5 gateways. The system allocates 2 GHz to the feeder link: from 27.5 GHz to 29.5 GHz to uplink and from 17.7 GHz to 19.7 GHz downlink. Full frequency re-use (reuse 1) is assumed in the feeder link, which means that each GW uses the same 2 GHz band on the feeder uplink; as well as in feeder downlink.

The system allocates 500 MHz to the user link: from 29.5 GHz to 30 GHz to uplink and from 19.7 GHz to 20.2 GHz to downlink. The 500 MHz user band is divided into four 125 MHz bands (i.e., four color re-use). Each spot-beam is allocated a 125 MHz band on the user uplink, and another 125 MHz band on the user downlink. Each GW can support an aggregate traffic to/from 16 beams. Thus, each GW is mapped to 16 beams. Since the system comprises of 72 spot-beams in total, five GWs are needed to support them.

The satellite system configurations are implemented to be easily changed to different kind of satellite systems: e.g. in terms of frequency configurations, coverage areas and number of spot-beams and GWs.

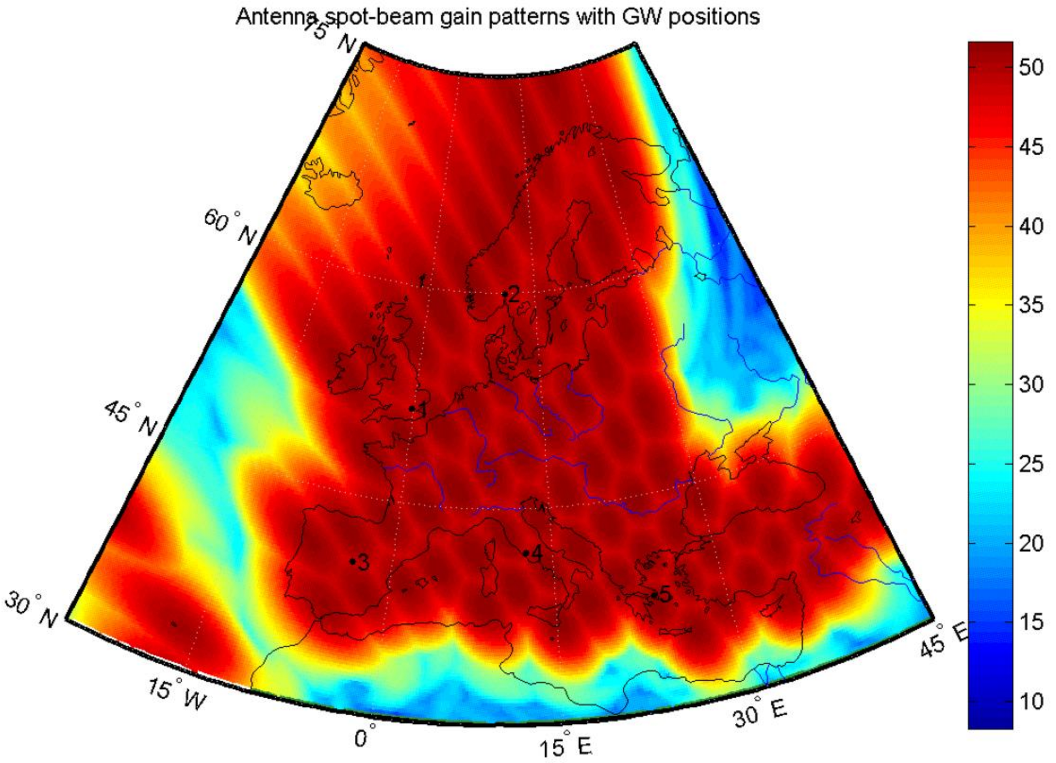


Fig 1. Antenna gain patterns of the satellite system.

Satellite model structure captures all the necessary functionality to be able to simulate multi-spot beam interactive transparent GEO satellite networks. This contains modules and nodes for *User Terminal (UT)*, *Satellite (SAT)*, *Gateway (GW)*, *Network Control Center (NCC)*, terrestrial nodes (end users) and their interaction (see Fig 2). The general simulator satellite module architecture is presented in detail in [5]. The satellite nodes have a new implementation of a *NetDevice* class (*SatNetDevice*), which implements the DVB-S2 [7] and DVB-RCS2 [8] protocol stacks. *SatNetDevice* consists of *Logical Link Control (LLC)*, *Medium Access Control (MAC)* and *Physical (PHY)* classes, specific for each node. In addition, a new implementation of the *Channel* class (*SatChannel*) is needed for it to support, e.g., satellite system received signal power calculations. More about the physical layer models of SNS3 can be found from [9]. In terrestrial domain, the used access technology may be anything supported by the ns-3 simulator, such as point-to-point, Carrier Sense Multiple Access (CSMA), Wi-Fi, and Long Term Evolution (LTE). This enables configurations for joint scenarios between satellite networks and terrestrial networks. NCC is modelled as a shared module with all GW nodes, which assumes that there is only one NCC entity and that there is an ideal communication channel between GW nodes and NCC.

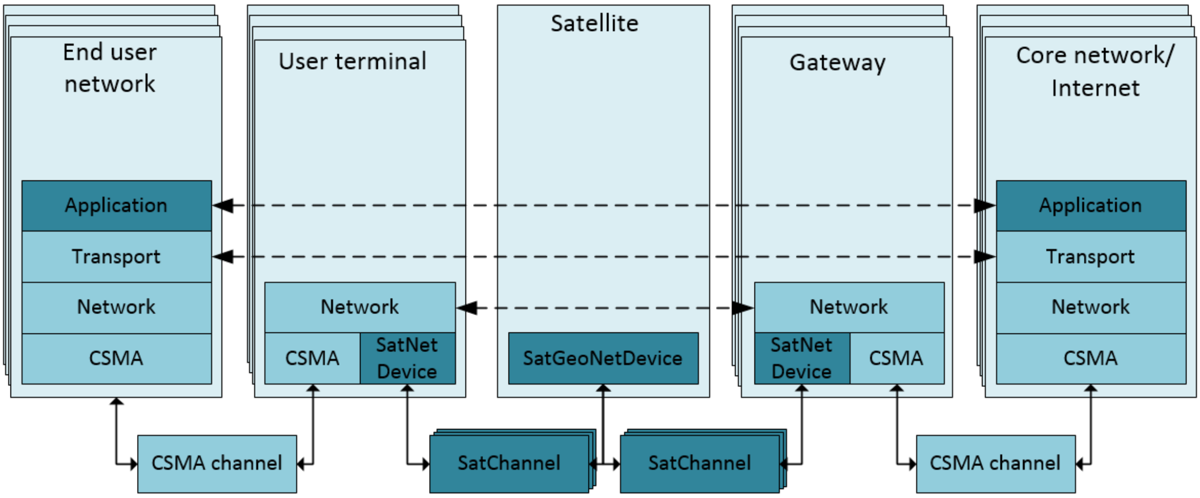


Fig 2. General architecture of SNS3.

Table 1 summarizes the features of SNS3 grouped by protocol layer and satellite link direction. Simulator adopts DVB-S2 Time Division Multiplexing (TDM) specifications [7] in forward (FWD) link and DVB-RCS2 MF-TDMA specifications [8] in return (RTN) link. Simulator features include, e.g., satellite channel models, physical layer abstraction based on link level Signal to Noise plus Interference (SINR) – Block Error Rate (BLER) curves, scheduling, Adaptive Modulation and Coding (ACM), Demand Assignment Multiple Access (DAMA) Capacity Allocation Categories, Random Access (RA) and encapsulation/fragmentation with Generic Stream Encapsulation (GSE) and Return Link Encapsulation (RLE).

Table 1. SNS3 feature table.

	FWD link	**RTN link**
Physical (PHY)/channel	Propagation delay, Land Mobile Satellite channel, Free-Space Loss (FSL), antenna gain patterns, Rx power calculation, weather traces, noise, interference, link budget parameterization, error modeling	
Medium Access Control (MAC)	FWD link scheduler, ACM, Base Band (BB) frame container	Random access (slotted ALOHA, Contention Resolution Diversity Slotted ALOHA (CRDSA)), UT scheduler, Terminal Burst Time Plan (TBTP) receiver
Logical Link Control (LLC)	GSE encapsulator, RLE decapsulator, Automatic Repeat request (ARQ), packet queue(s)	DAMA request manager, RLE encapsulator, GSE decapsulator, ARQ, packet queue(s)
Packet classifier	Differentiated Services Code Point (DSCP) to flow identifier mapping	
Satellite	Transparent star, non-flexible payload	
Network Control Centre (NCC)	-	RTN link burst scheduler
Network layer	IPv4, IPv6, Internet Control Message Protocol (ICMP), Mobile IP(*), Open Shortest Path First (OSPF)(*)	
Transport layer	User Datagram Protocol (UDP), Transport Control Protocol (TCP), Stream Control Transmission Protocol (SCTP)(*)	
Traffic models	On-off, HyperText Transfer Protocol (HTTP), Near-Real Time Video (NRTV), Constant Bit Rate (CBR)	
Statistics	Throughput, delay, signaling load, queue size, SINR, error probability, frame load, waveform usage, requested resources, granted resources, …	
Terrestrial networks	CSMA, Wi-Fi, Point-to-Point, LTE, Wimax	

(*) Support is achieved through Direct Code Execution (DCE) framework of ns-3.

Simulation Results

A simulation campaign has been performed to show some example statistics from SNS3. Note, that the simulation case is by no means optimized to get the maximum spectral efficiency and/or throughput. Simulation case consists of one simulated spot-beam with variable amount of UTs (one end user per UT) and traffic. The spot-beam is configured with 125 MHz bandwidth in both FWD and RTN links. The amount of user terminals (25-250) and offered load per UT (1-3 Mbps Constant Bit Rate) has been varied in either FWD or RTN direction. Spectral efficiency and average user throughput has been collected as KPIs. The channel conditions have been modeled by means of channel (weather) time traces.

Fig 3 and Fig 4 present example simulation results from SNS3 in FWD link. The maximum spectral efficiency in FWD link with 125 MHz bandwidth saturates at 3.3 b/s/Hz. This reflects the benefits of Adaptive Coding and Modulation (ACM), high probability of good channel conditions (Line-of-Sight) and thus high probability of using 32APSK, which is the MODCOD with the highest efficiency. FWD link supports more than 250 users at 1 Mbps, ~200 users at 2 Mbps and ~125 users at 3 Mbps without any significant degradation in the user throughput.

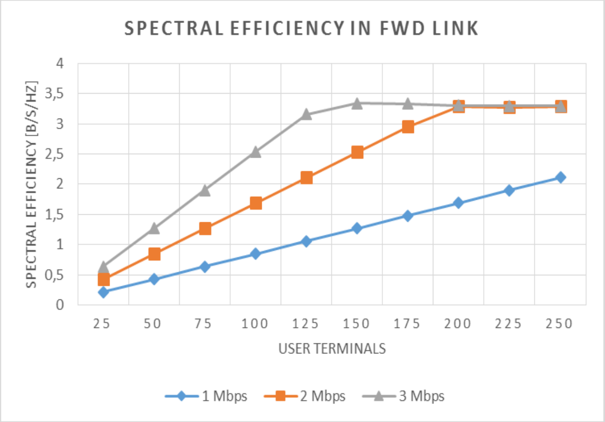


Fig 3. Spectral efficiency in FWD link.

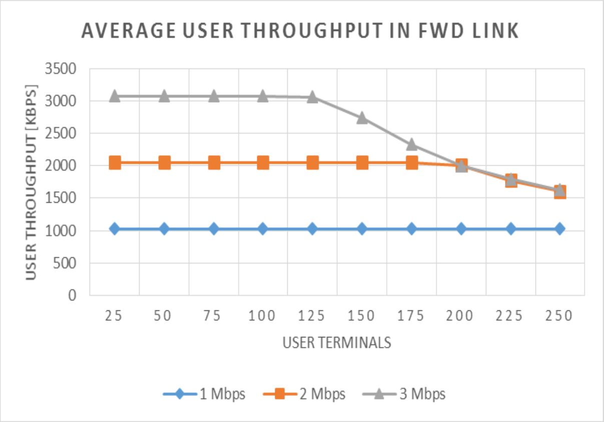


Fig 4. Average user throughput in FWD link.

Example simulation results from RTN link are shown in Fig 4 and Fig 5. The simulation case is otherwise similar to the FWD link case, except that for the frame configuration. In RTN link case, three simplified frame configurations have been assumed: 1 frame per superframe of size 125 MHz, roll-off 0.2 and carrier spacing of 0.3.

- Conf-1 – 625 kHz carriers (in total 200 carriers)
- Conf-2 – 1.25 MHz carriers (in total 100 carriers)
- Conf-3 – 2.5 MHz carriers (in total 50 carriers)

According to Fig. 5, the spectral efficiency of RTN link saturates at around 1.7 b/s/Hz. This reflects a high utilization of 16QAM, which is the most efficient MODCOD in RTN link. If compared to FWD link results, the RTN link spectral efficiency is quite much lower. This is because of several reasons: the highest usable modulation is 16QAM compared to 32APSK in FWD link, the carriers in RTN link are separated by carrier spacing so that consecutive carriers would not interfere each other, and UT is capable of being scheduled to only one carrier at a time, which limits the peak throughput for one single UT. However, even with the narrower carriers the system is providing the same peak spectral efficiency, if the amount of UTs is increased enough.

The effect of carrier bandwidth is visible in Fig 6. With 3 Mbps of offered load, only the carrier bandwidth of 2.5 MHz can offer 100% satisfaction, when UT can be allocated one full carrier.

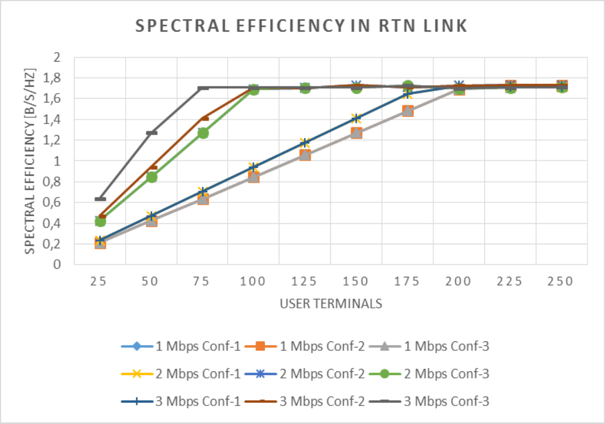


Fig 5. Spectral efficiency in RTN link.

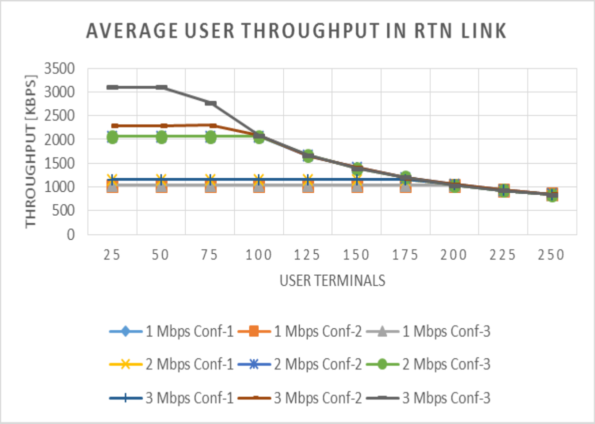
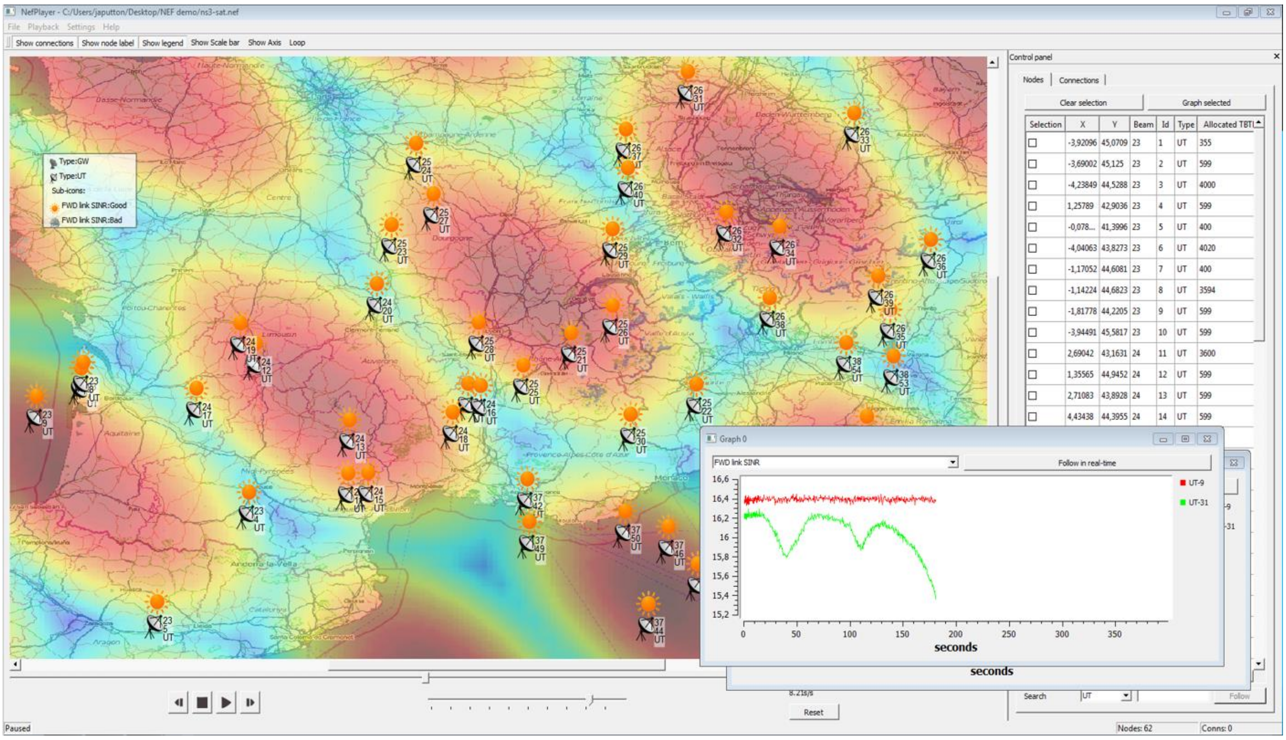


Fig 6. Average user throughput in RTN link.

NEF Player

Fig 7 presents a Network Event File (NEF) player with SNS3 specific use case. NEF player is a simulator independent visualization tool which can be easily integrated on top of any network simulator, and configured to visualize KPIs essential to the project in question. Meaningful events and statistics may be included in the visualization as well, such as node mobility, handovers, throughput, SINR, resource consumption and packet errors. An example video about NEF player can be found in [10].



Selection	X	Y	Beam	Id	Type	Allocated TBTL
□	-3,92096	45,0709	23	1	UT	355
□	-3,69002	45,125	23	2	UT	599
□	-4,23849	44,5288	23	3	UT	4000
□	1,25789	42,9036	23	4	UT	599
□	-0,078...	41,3996	23	5	UT	400
□	-4,04063	43,8273	23	6	UT	4020
□	-1,17052	44,6081	23	7	UT	400
□	-1,14224	44,6823	23	8	UT	3594
□	-1,81778	44,2205	23	9	UT	599
□	-3,94491	45,5817	23	10	UT	599
□	2,69042	43,1631	24	11	UT	3600
□	1,35565	44,9452	24	12	UT	599
□	2,71083	43,8928	24	13	UT	599
□	4,43438	44,3955	24	14	UT	599

Fig 7. Simulator independent visualization with NEF player.

SNS3 simulations have several possible use cases:

- Algorithm and protocol design, development and optimization, e.g. Radio Resource Management (RRM)
- Building Proof-of-Concept for new features
- Full satellite system performance analysis and end user quality of service verification
- Integration between satellite and terrestrial networks (e.g. for satellite 5G)

SATELLITE NETWORK SIMULATOR 3 IN EMULATION MODE

This section discusses how SNS3 can be integrated with a real-life test-bed environment [11]. The integration utilizes special *network layer to device interfaces* for connecting the simulator to the host machine's network interfaces. It is one of the features of ns-3 and is compatible with real network protocols such as Internet Protocol (IP).

The connection of the simulation and real world is done by installing an additional file descriptor (FD) *net device* (henceforth, *FdNetDevice*) in the simulated ns-3 node. *FdNetDevice* is able to interact with an external device by using an FD defined by the developer. The FD can be associated, for example, to a physical network device or a TAP device in the host computer.

ns-3 currently supports two types of *FdNetDevice*. The first is *TAP FdNetDevice*, which creates a virtual network tap interface on the host machine upon the start of the ns-3 simulation. This approach is suitable for scenarios with virtual end nodes running on the same host as ns-3. In this case, the TAP device will be attached to a Linux bridge that can be further utilized by, e.g., virtual Linux Container (LXC) machines.

The second type of *FdNetDevice* is *EMU FdNetDevice*. Instead of creating an additional interface, it directly connects to a physical interface of the host machine. This approach may be utilized in scenarios where no virtualization is used.

An example illustration of the SNS3 in real-time emulation mode is depicted in Fig 8. The scenario consists of 3 different devices: end user, simulation server and content server. Simulation server runs SNS3 in one process with a configured parameterization. All of the features of SNS3 which are available for simulation case are also available for this emulation case. By utilizing *EMU FdNetDevice,* real devices (end users) may be attached to any of the simulated UTs, regardless of the end user's OS, protocols or applications. Note that each UT must be associated with a unique physical interface of the simulation server. This limits the number of UTs in the configuration to the number of physical interface that the simulation server has.

Alternatively, one may utilize LXC and *TAP FdNetDevice*, i.e., integrating SNS3 with virtualized hosts within the simulation server. This approach is not affected by limited physical network interfaces at the simulation server.

By default, the main ns-3 process thread occupies only a single processing thread, which is very inefficient when maximum test-bed capacity is desired. The existing Message Passing Interface (MPI) in ns-3 allows the execution of single simulation on multiple cores. However, there are still some challenges and limitations, e.g., finding a proper logical process from the simulation scenario, and the fact that MPI supports only point-to-point links at the moment.

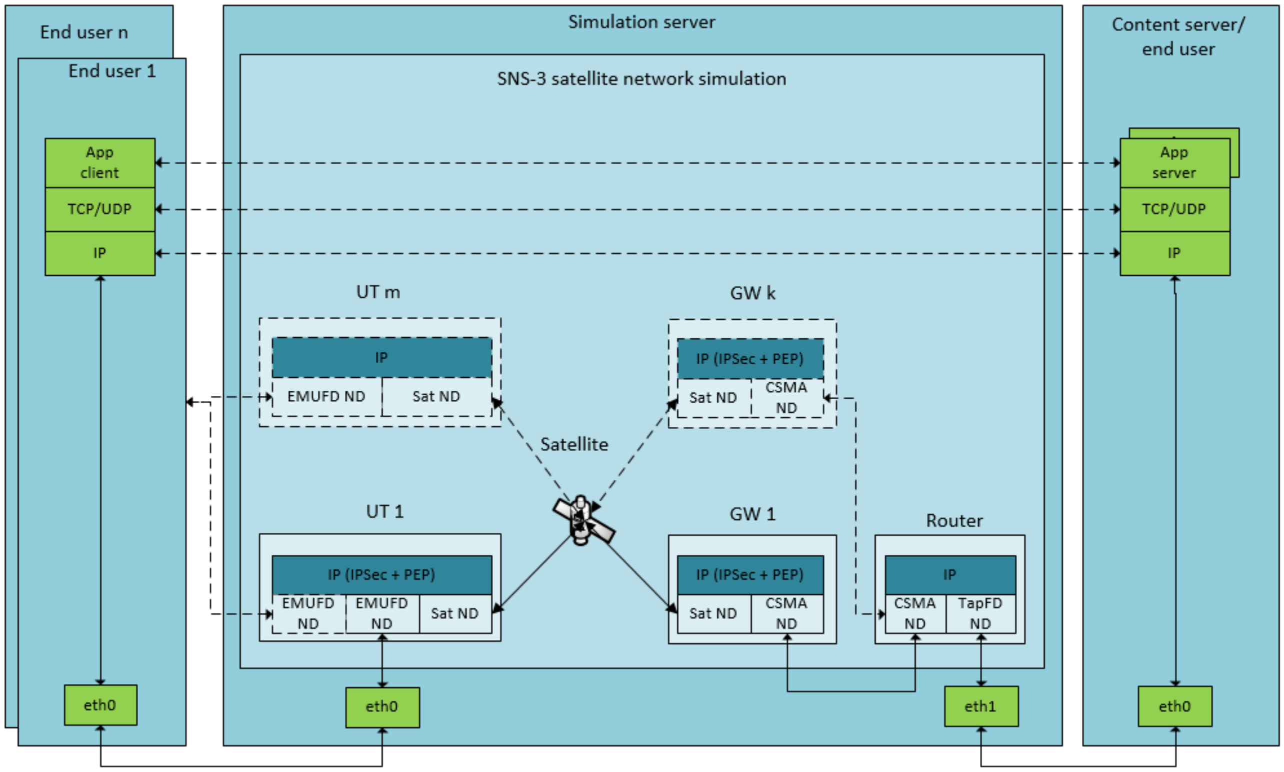


Fig 8. Example SNS3 emulation architecture (using EMU FdNetDevice approach).

According to [11], one of the key challenges of ns-3 emulations is to avoid reaching the upper bound of the host computer's processing capability. This may results in artificial packet drops and may thus have an effect on the emulation results. This issue is illustrated in the following example of real-time satellite emulations with different load conditions in both FWD and RTN link directions. The emulation scenario is loaded with CBR type traffic with different level of offered loads and with different number of end users. The aim is to study whether the system gets congested (satellite system capacity reached) or overloaded (emulation capacity reached).

Emulation Results

An emulation campaign was performed to show some example statistics from SNS3 emulations. The emulated case is similar to the simulated scenario, e.g. in terms of parameterization. However, now the end users are virtualized Linux users using LXC and running IPerf application with different offered loads. The emulation server is using Intel i7-3770 @ 3.40 GHz processor with 4 physical/8 logical cores.

The results show that with heavy load (and increased complexity) the emulation capacity is exceeded before the maximum capacity of the satellite system. This is apparent as reduced throughput (see Fig 9 and Fig 11), increased number of FD drops (see Fig 10 and Fig 12) and increased variance in the statistics.

In FWD link, the system is able to support at least 110 users at 200 kbps load, but only 50 users at 1 Mbps load. The overloading conditions begin at about 3500-4800 packets per second. Emulations with higher number of UTs become congested at lower level of load. As stated already in [11], the increased complexity of the emulation (e.g., the amount of simulated nodes) reduced the emulation capacity.

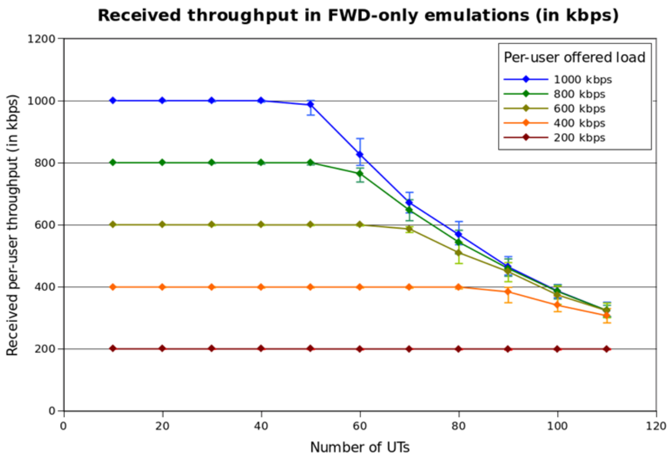


Fig 9. Emulated throughput in FWD link.

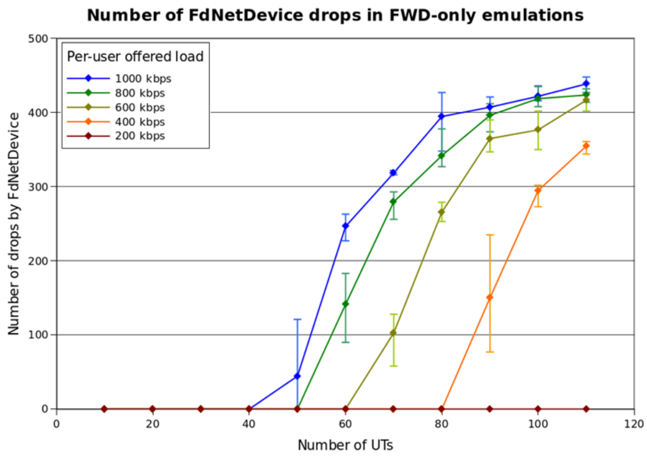


Fig 10. Amount of packet drops in FWD link.

In RTN link, the emulation shows slightly higher emulation capacity than in FWD link. More than 110 users can be emulated at 200 kbps load, followed by about 60 users at 1 Mbps load. This is caused by the higher complexity of the FWD link scheduler.

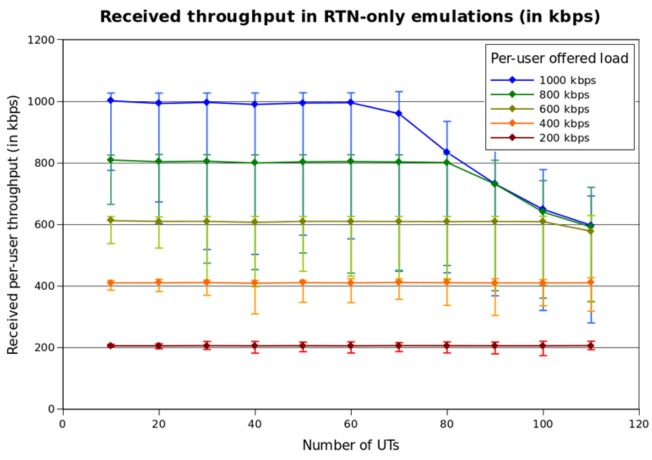


Fig 11. Emulated throughput in RTN link.

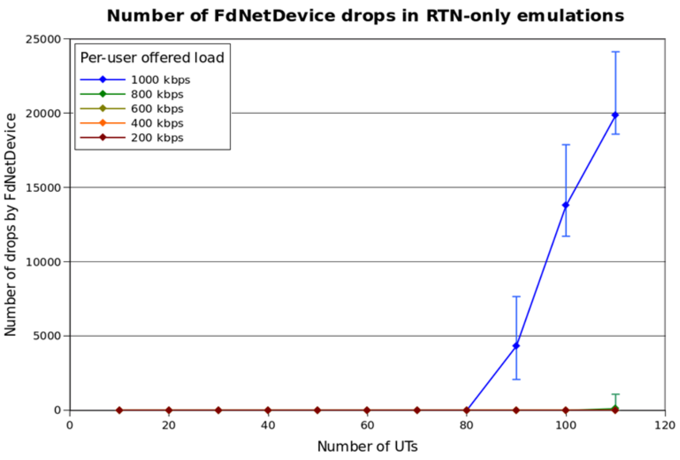


Fig 12. Amount of packet drops in RTN link.

The main benefit of the SNS3 emulations is that the user may experiment with and optimize the performance and behavior of real higher layer protocols, applications and hardware on top of a simulated satellite network. However, there may be performance related challenges in the emulation as well; the system load needs to be controlled not to cause overload to the ns-3 real-time scheduler. When comparing the simulation and emulation performance results, it is evident that the emulations are not as scalable as simulations. However, simulations and emulations have their own specific and valuable use cases.

CONCLUSIONS

Satellite Network Simulator 3 (SNS3) is developed to be highly modular and flexible to be able to match the future communications/networking R&D needs of satellite system vendors, operators, research organizations, as well as scientific community. The module aims at providing easy reconfigurations for other multi-spot GEO satellite systems and extensibility towards e.g. future communication standards. In addition, the Radio Resource Management (RRM) algorithms and protocols are intended to be easily replaced to enable satellite system R&D studies. SNS3 provides also a real-time emulation support, which provides even more flexible usage for different use cases. The satellite models and implementations are planned to be contributed to ns-3 community [6] during year 2015.

The simulator future use cases include e.g. algorithm, protocol, RRM and device development, proof-of-concept and bottleneck analysis, and full satellite system performance analysis and optimization. SNS3 could be used as a part of specification and design phases (0, A, B) of future satellite systems.

ACKNOWLEDGEMENTS

This work has been partly funded by European Space Agency in the scope of ARTES 5.1 AO6947 “Development of an open-source, modular and flexible satellite network simulator” project (contract no: 4000106081/12/NL/NR) and ARTES 5.1 AO7476 “Network Security on Multi-Spot Beam Broadband Networks for Multicast Based Interactive Systems” project (contract no: 4000109719/13/NL/EM).

REFERENCES

[1] B. Niehoefer, S. Subic and C. Wietfeld, “The CNI Open Source Satellite Simulator based on OMNeT++”, *6th International Workshop on OMNeT++*, March 2013.

[2] “DVB-RCS(2) for ns-3”, http://homepages.abdn.ac.uk/r.secchi/pages/dvbrcs_ns2.htm, referenced 2.3.2015.

[3] O. Alphand, P. Berthou, T. Gayraud, F. Nivor, S. Combes, “Platine: DVB-S/RCS tesbed for next-generation satellite networks”, *Advanced Satellite Mobile System (ASMS)*, 2006.

[4] D. F. Pinaz and C. Morlet, “AINE: An IP Network Emulator”, *10th International Workshop on Signal Processing for Space Communications (SPSC)*, 2008.

[5] J. Puttonen, S. Rantanen, F. Laakso, J. Kurjenniemi, K. Aho and G. Acar, ”Satellite Model for Network Simulator 3”, *7th International ICST Conference on Simulation Tools and Techniques (SIMUTools 2014)*, March 2014.

[6] ns-3 Consortium. *Network Simulator 3*, http://www.nsnam.org, referenced 2.3.2015.

[7] “Digital Video Broadcasting (DVB); Second generation framing structure, channel coding and modulation systems for Broadcasting, Interactive Services, News Gathering and other broadband satellite applications (DVB-S2)”, 2005.

[8] “Digital Video Broadcasting (DVB); Second Generation DVB Interactive Satellite System (DVB-RCS2); Part 2: Lower Layers for Satellite standard”, 2012.

[9] J. Puttonen, S. Rantanen, F. Laakso, J. Kurjenniemi, K. Aho, G. Acar, “A Packet Level Simulator for Future Satellite Communications Research”, *AIAA Space 2014*, August 2014.

[10] Magister Solutions, “NEF-player demo of satellite and mobile simulations”, https://www.youtube.com/watch?v=5SZgMyribXE.

[11] V. Hytönen, B. Herman, J. Puttonen, S. Rantanen, J. Kurjenniemi and O. Smeyers, “Satellite Network Emulation with Network Simulator 3”, *Ka and Broadband Communications, Navigation and Earth Observation Conference*, October 2014.