IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received November 29, 2021, accepted January 13, 2022, date of publication January 26, 2022, date of current version February 1, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3146770

ns-3-leo: Evaluation Tool for Satellite Swarm Communication Protocols

TIM SCHUBERT[1], LARS WOLF[2], (Senior Member, IEEE), AND ULF KULAU[3,4]
[1]Technische Universität Braunschweig, 38106 Braunschweig, Germany
[2]Institute of Operating Systems and Computer Networks, Technische Universität Braunschweig, 38106 Braunschweig, Germany
[3]Department of Smart Sensors, Technische Universität Hamburg, 21073 Hamburg, Germany
[4]DSI Aerospace Technologie GmbH, 28199 Bremen, Germany

Corresponding author: Ulf Kulau (ulf.kulau@tuhh.de)

We acknowledge support for the Open Access fees by Hamburg University of Technology (TUHH) in the funding programme Open Access Publishing.

ABSTRACT We present *ns-3-leo*, a module for the discrete event network simulator *ns-3*, which includes models for the network mobility and link characteristics of Low Earth Orbit (LEO) satellite mega-constellations. The initial goal has been to create a simulation environment in which existing routing protocols from the Wireless Sensor Network (WSN) and Mobile Ad-hoc Network (MANET) areas can be evaluated in these types of networks. To show its capabilities, we performed simulations of the planned satellite constellations of Starlink and Telesat based on publicly available parameters. Through simulations of Ad hoc On-demand Distance Vector Routing (AODV), we were able to gain valuable insights into the unique challenges presented to these protocols by satellite networks. It is unlikely that existing ad-hoc protocols using distance-vector routing or flooding can be used for broadband internet connectivity and alternative methods of handling large-scale mobility in these networks are needed. However, with ns-3-leo a much needed extension for supporting the development of protocols fulfilling the needs of satellite constellations is provided.

INDEX TERMS Low earth orbit satellites, satellite communication, routing protocols, internet, mobile ad hoc networks, AODV, disruption tolerant networking, computer simulation.

I. INTRODUCTION

Modern terrestrial and fiber-optics networks provide fast broadband internet connectivity, but in 2019 they served only around 54% of the global population [1]. For people living in remote areas or in countries with insufficient infrastructure, access is limited.

Low Earth Orbit (LEO) satellite constellations such as Iridium [2] and Globalstar [3] provide telephony and internet coverage in remote areas. While providing reliable coverage to most of the world, these older types of networks do not support the data rates needed by modern internet applications and their network capacity is quite limited. Another way of providing broadband internet connections is via Geo-Stationary Earth Orbit (GEO) satellites [4]. The position in GEO has the advantage of needing considerably fewer satellites to achieve global coverage. While the data rates of these satellites support modern internet applications, the considerable distance from earth has the disadvantage of long end-to-end delays and the bandwidth that is offered to end-users can not compete with fiber-optics or terrestrial networks where those are available.

In recent years, the idea of using many LEO satellites for this purpose has been promoted by a number of companies, among which SpaceX [5], Amazon [6], Telesat [7], OneWeb [8] are some of the more notable competitors. An example of such a mega-constellation can be seen in figure 1.

Based on theoretical calculations, the network capacity would be sufficient to provide a high-bandwidth internet connectivity [9] comparable to that of fiber-optics networks. While the low altitude of LEO satellites results in very low latency [10], because of the small signal propagation delay compared to GEO satellites, a significantly larger number of stations are required to relay data over the same distance, as Barrios *et al.* [9] point out. To overcome part of this challenge, Inter-Satellite Link (ISL) have successfully been used in the past to extend the connectivity of individual

The associate editor coordinating the review of this manuscript and approving it for publication was Rute C. Sofia.

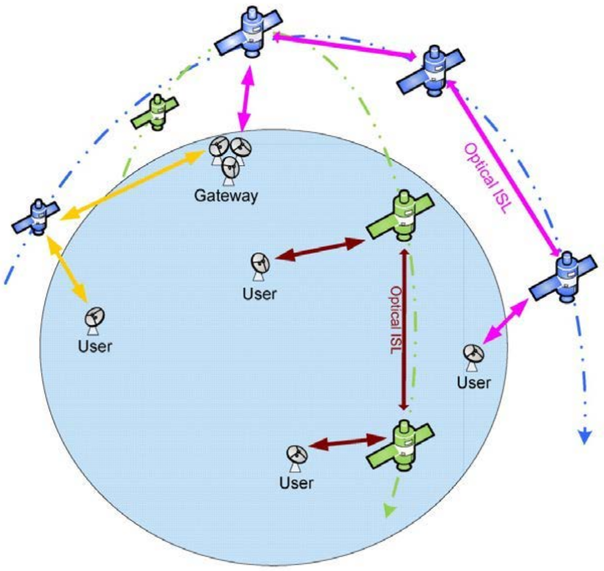

FIGURE 1. Architecture of the Telesat LEO constellation (from Telesat's application documents [7]).

satellites beyond the horizon [11]. ISLs allow for the direct transmission of data between satellites either using directed radio links or laser terminals. This removes the need for additional ground relays, where such links are possible. Figure 1 shows an example of a multi-hop path between a user terminal and a gateway that contains multiple ISL.

ISLs also have the potential to increase the overall throughput [9], reduce operational costs and allow satellite operators to scale their networks by adding additional orbits and satellites without the need for additional ground stations. The downside of this is, that with such ISL-based multi-hop networks there may exist various alternate paths. Hence, the network operator has to introduce appropriate routing algorithms to optimize the available network capacity. Routing is a classical problem in many networking contexts. The solution always depends on various parameters of the network such as nature of interconnects of the individual nodes, data and error rates, as well as mobility (if relevant).

For LEO satellite mega-constellations, the orbits of individual satellites are deterministic, but the topology of the network is constantly changing, interconnects (e.g., via ISL) change and individual satellites may disappear from or be added to the network at any time, due to equipment failure or traffic volume. As of today, there is only little published information about the routing protocols satellite operators might use to support these types of networks. At the same time, LEO networks share characteristics with other network types such as from MANET and WSN as they have to deal with mobility, a large volume of traffic, limited resources and certain requirements for the quality of service. There is a wide variety of candidate protocols that each solve some of these challenges [12], but a more specific evaluation has yet to be done.

A. PROBLEM STATEMENT

To learn more about the performance of existing MANET and WSN protocols in the context of LEO networks, it would be of interest to see how individual protocols perform inside a discrete event network simulation environment. This simulation requires detailed models of the network topology, radio technologies and network protocols involved in LEO mega-constellations.

While all of these individual topics have been theoretically covered in previous research and are relatively well understood, some building blocks for a simulation are still missing. As a first step towards a practical assessment of possible protocol options, we created ns-3-leo, a simulation tool for the network topology and link characteristics of LEO networks. Using this tool, we were able to identify future challenges for routing protocols in LEO mega-constellations, but more important we support developers to implement and investigate new protocols.

B. DOCUMENT STRUCTURE

First, we present an overview of previous work on the simulation LEO satellite networks. In section III, we describe our mobility model and channel models. The implementation of a *ns-3-leo* as a module of *ns-3* is further detailed in section IV. In section V, we present the results of our analysis of the performance of *ns-3-leo*. We then use it to evaluate AODV for a set of scenarios involving the Starlink and Telesat networks, to showcase its capabilities. In section VI, we discuss the current limitations of our approach, identify challenges for new and emerging routing protocols in LEO satellite networks and then present our conclusion.

II. RELATED WORK

The following section highlights some research related to LEO mega-constellations, the involved radio technology and related routing protocols. We also discuss similar simulations from the literature and compare them to our own.

A. STATISTICAL MODELS

The Federal Communications Commission (FCC) filings for the Starlink [5], OneWeb [8] and Telesat LEO [7] constellations contain detailed information on their link characteristics and orbital parameters. All of them have similar architectures with many ground stations and satellites, but with some notable differences. OneWeb's constellation does not feature ISL in its first generation, while Starlink plans to have significantly more satellites when fully deployed. An exemplary overview of the network architecture of Telesat LEO constellation is shown in figure 1.

Barrios *et al.* [9] described and compared some of these planned LEO constellations in detail. They combined the models of the topology and link budgets with the Iternational Telephony Union (ITU) model for atmospheric attenuation [13] to create a statistical model of the throughput and capacity for each constellation. Using this model, they estimated the total capacity relative to the capacity of the ISLs, the number of gateway antennas and the number of ground stations. Additionally, they used global census data [14] to create a demand model of the global network traffic and designed a basic genetic algorithm to determine the optimal number and placement of the ground stations. Based on the bandwidth utilization of the satellite links, the maximum system throughput and the total network capacity, they analyzed the impact of ISLs on the number of ground stations required and found that the use of ISL reduces the required number of ground stations and increases the total throughput. Since they analyze the same constellations, but use a statistical model instead of a simulation, their findings can be used to verify the validity of our simulation.

B. INTER-SATELLITE COMMUNICATIONS

One of the more prominent examples of satellite communication networks that make use of ISL is Iridium [15] and Iridium Next [16]. In these networks, only the communication between satellites with opposing directions on neighboring polar orbits requires the use of relaying ground stations.

Modern Ka-Band ISLs are capable of very high data rates of multiple gigabits per second over long communication distances. In the future these capabilities may be even further extended using optical ISL. At the time of writing, the off-the-shelf hardware required for optical ISLs in commercial applications was still uncommon [17], so our simulation focuses on ISLs using Ka-Band transmitters.

The combined use of ground relays and ISL is further investigated by Handley [10], [18]. He finds that paths using a series of space-ground links, even without ISLs, generally have a shorter delay than paths that exclusively use terrestrial networks. If the paths use ISLs, this further reduces the propagation delay, decreases the number of required ground-relays and increases the total capacity of the network. This supports the findings of Barrios *et al.* [9]. Handley uses his own visualization based on Unity 3D [19] to simulate and display the shortest paths inside the Starlink LEO network. The software ignores some connectivity constraints which, according to Handley, makes some east-west paths unrealistically good [20].

The statistical model of Barrios *et al.* [9] does not consider the effects of the protocols involved in packet routing, frequency coordination and medium access control. All three may play a significant role in how efficiently the available network capacity can be utilized and what the resulting quality of service will be. We attempt to add to their analysis through our own simulation of the effect of ISLs on existing routing protocols.

C. SPACE COMMUNICATION PROTOCOLS

The Consultative Committee for Space Data Systems (CCSDS) compiled an overview of existing space communication protocols [21]. Protocols like CFDP [22] are less relevant to LEO networks, since they are designed for reliable file transfers in scientific missions, not broadband internet connectivity.

Iridium and Geosat provide well researched examples of frequency coordination and Medium-Access Control (MAC) protocols in satellite networks. Some technical details can be found in satellite broadcasting standards such as the specification for Digital-Video-Broadcast - Satellite 2 X (DVB-S2X) [23]. A practical implementation of the predecessor of this standard, DVB-S2X, although only for a single geostationary satellite, can be found in *SNS3* [24]. Finding and implementing a suitable frequency and MAC coordination algorithm for LEO mega-constellations remains a topic for future research.

In an extensive survey of available protocols, Azúza *et al.* [12] identified potential candidate protocols for satellite networks. Their envisioned usage scenario is a very heterogeneous, federated architecture of multiple satellite networks that communicate across a range of different technologies. For this, they compared routing protocols from many network types, including MANETs, snapshot networks, existing LEO satellite networks, multi-layered networks, WSN and Delay Tolerant Networking (DTN). They compiled a list of properties that are critical for routing protocols in satellite networks. According to them, such a protocol should be based on distance-vector routing, be adaptive to topology changes from network mobility, and support resource aware routing.

It is important to note that the federated architecture of multiple satellite networks proposed by them is very different from the architecture that is planned for the LEO mega-constellation. This loosens some requirements for candidate protocols, such as the ability to deal with a heterogeneous architecture or the support for multiple different routing protocols, but puts more emphasis on others such as the optimal distribution of available bandwidth and resources, support for mobility and minimal latency. As some possible candidates for future investigations they suggest Load-Aware On-Demand Routing (LAOR), Zone Routing Protocol (ZRP), Ad-Hoc On-Demand Distance-Vector Routing (AOMDV), Routing Protocol For Low-Power And Lossy Networks (RPL), Energy Aware Epidemic Routing (EAEpidemic) and Probabilistic Routing Protocol For Intermittently Connected Networks (PROPHET).

Our exemplary evaluation of *ns-3-leo* uses AODV, since it matches most of the criteria and implementations for *ns-3* are already available. Later in section V, we compare the results of our preliminary evaluation to the expectations of Azúza *et al.* [12].

D. DISCRETE EVENT NETWORK SIMULATION

Bedon *et al.* [25] investigated the performance of Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) when used with ISL links. They simulated a network of *Cubesat* satellites along one polar orbit with different traffics flows and distances between the satellites. The simulation uses *SaVi*, which is a visualization tool based on the now outdated *ns-2*. While they also simulate a LEO network, they do not consider mobility or communication between satellites on different orbits.

ns-3 is a discrete event network simulator [26]. Discrete event network simulators process events, like the transmission of a packet, the progression of a node to a different waypoint or a route expiration of a protocol timer, in the order they are scheduled using a central simulator control loop. *ns-3* provides the basic components including the simulator event loop and interface definitions for network devices, channels, mobility model and routing protocols. The simulator is easily extendable using a module system, where individual modules may provide additional components. Users may build simulations by combining them with parts of *ns-3* itself and other third party modules. This provides our simulation with a large framework that already includes many protocol implementations. *ns-3* includes extensible tracing facilities using various types of event sources and sinks. We may also obtain Packet Captures (PCAPs) for further analysis in network protocol analyzers such as *tcpdump* and *Wireshark* [27].

Building a more accurate mobility model, requires the positions of the ground stations and satellites at any given point in time. The simulation of orbital dynamics of satellites often involves mission-planning software like *FreeFlyer* [28]. This software is too complex to use and not easily integrable into existing simulation tools. *ns-3-leo* provides both a simple mobility model based on circular orbits that encourages fast experimentation and can optionally import mobility traces from external files.

SNS-3 [24] is a sophisticated module for *ns-3* that models a geostationary Digital Video Broadcast (DVB) satellite systems, including its mobility, physical-layer and link-layer. Satellite module for ns-3 (SNS-3) is not applicable to LEO satellite constellations for multiple reasons. In LEO, the satellites do not remain stationary relative to a point on the earths surface, but have a high degree of mobility. As such, the channel model has to consider the mobility of multiple satellites, which is not the case for the single geostationary satellite included in the model of SNS-3. To support multi-hop paths inside the constellation, each node must be able to regenerate the payload and update the forwarding information inside each packet it forwards. The channel-model of SNS-3 is limited to a bent-pipe design, which tightly ties the uplink and downlink transmissions to each other and does not allow any modifications to the packet by a higher protocol layer. At the same time, the channel and link-layer models of SNS-3 use very similar transmission technologies to that of LEO satellites. This is why we initially attempted to make these models useable outside of SNS-3 through code refactoring. While this would have required too much effort to be suitable, SNS-3 influenced the design of the *ns-3-leo* module.

OS^3 is a satellite simulator that is available as a module for *OMNET++*, another discrete event network simulator. OS^3 models the satellite positions using Simplified General Perturbations Version 4 (SGP4) [29] and includes a detailed model of the propagation loss using current weather information. One downside of SGP4 is that is requires a description of the satellite's orbit in the form of Two-Line Element (TLE), which usually is only available for already launched satellites [30]. *ns-3-leo* can additionally simulate approximate circular orbits and use externally generated waypoint files. This makes it possible to simulate the orbits of satellites in the planned mega-constellations that have not been launched yet. An advantage of OS^3 is the detailed path-loss model, which our simulation does not yet include. Instead, we rely on estimates from the literature [9]. OS^3 does not model ISLs, which is necessary for a simulation of upcoming LEO networks.

III. MODELS

In this section, we discuss the theoretical background of our mobility model of nodes inside LEO networks, and how we modeled satellite-ground and satellite-satellite interconnects.

A. COORDINATE REFERENCE FRAME

To make our mobility model usable in practical research, it was important to find a coordinate reference frame that allows us to express the positions of both ground stations and satellites with a tolerable loss of precision. Positions for ground stations, and other points on the earth's surface, are generally provided as pairs of longitude and latitude, whereas the position of LEO satellites is often specified as their orbit's inclination from the equatorial plane and their mean altitude from earth's surface, or even using TLEs.

There exist a variety of coordinate systems to choose from when doing orbital calculations, the main difference between them being their point of reference. For instance, the International Celestial Reference Frame (ICRF) [31] centers around the sun, whereas many other systems such as International Terrestrial Reference Frame (ITRF) [32] have their origin at the earth's center of mass. The center of mass aligns mostly with that determined by Galileo Terrestrial Reference Frame (WGS84) [33] and newer versions thereof. This means that orbit propagation algorithms, such as SGP4, that rely on this version of the earth's gravitational model will produce results that can be directly translated into the ITRF and we can specify the positions of ground stations as pairs of latitude and longitude with a reasonable loss of precision. An additional advantage of the ITRF is that points on the surface of the earth (e.g., ground relays) stay fixed over time. This means only the positions of the satellites have to be kept up to date, which reducing computational demands.

We therefore use ITRF as a universal reference frame for the positions of both ground stations and satellites within the simulation.

B. MOBILITY MODEL

For the mobility model, it is necessary to know the speed, heading and current position of the satellites with sufficient accuracy to obtain precise timings for interconnect events

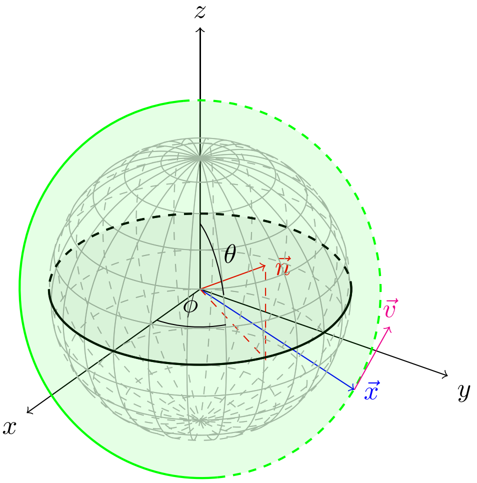

FIGURE 2. Simplified calculation of the position, speed, and heading of a satellite at $\vec{x}$ using its height, inclination θ and the azimuth ϕ of its orbital plane.

between the ground stations and satellites and between the satellites themselves.

Using algorithms for orbit propagations like SGP4, it is possible to create a mobility model from the current orbit parameters. Up-to-date TLE data for many satellites can be obtained from Celestrak [30]. Only providing mobility models for existing satellites is insufficient for a general simulation environment for arbitrary LEO satellite constellations in which the experimentation with arbitrary satellite orbits should be possible.

Our simplified mobility model supports circular orbits in LEO based on the inclination, azimuth, and height of the satellites. This has the downside of being less precise than the propagation of the orbits with e.g., SGP4 or even more precise models such as the Hybrid Simulation Platform for Space Systems (HPS) [34], since it excludes factors such as atmospheric drag and the unevenness of the earth's gravitational field, but it still is precise enough to accurately describe the satellites positions within a network topology. This enables the user to specify arbitrary orbits (e.g., that of not yet launched Telesat satellites) only using the orbit parameters published in the technical specifications [7]. It also has the advantage of being able to dynamically generate the positions, speed and heading at runtime, so that arbitrary spans of time can be simulated with any starting point without the loss of precision or the deterioration of the orbit.

Figure 2 shows how the position, speed, and heading of a satellite can be obtained using the height, inclination θ and azimuth ϕ of its orbital plane.

The velocity v of a satellite can be obtained from its height h using the gravitational mass product GM of the earth.

The norm n of an orbital plane of the satellite, which is shown in green, can be obtained by rotating the equatorial plane by the angle θ.

The progress within the orbit can then be obtained using v, and the time since the satellite last crossed the orbital plane t.

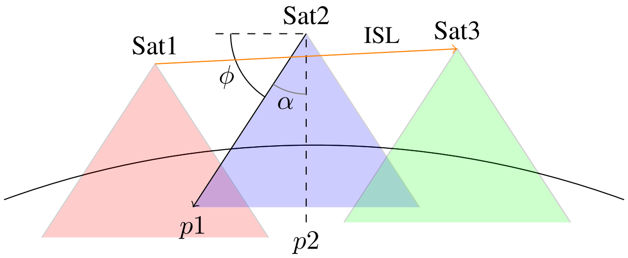

FIGURE 3. Beams of three satellites above the earth's surface.

C. CHANNEL MODEL

The channel model has the task of deciding if two nodes have an opportunity to transmit data, depending on their position within the mobility model and the characteristics of the radio channel. Nodes can be either satellites or ground stations.

1) INTER-SATELLITE LINKS

Two satellites, like *Sat1* and *Sat3* in figure 3, have a ISL with each other if they are sufficiently close for the signal to be strong enough at the receiver and if the Line-Of-Sight (LOS) between them is not blocked by other objects, like the earth. The satellites themselves can be assumed to be small enough at such distances as not to block the LOS. To compute the LOS, we used a method for line-sphere intersection that is often used in simple ray-tracers.

Depending on the value of the discriminant of the resulting quadratic equation and the relative positions of the satellites, and any intersection points of the ray with the earth, there may or may not be a LOS.

2) SATELLITE-GROUND LINKS

For the antenna beams, as shown in Figure 3, we specify the inclination ϕ based on the maximum relative angle α at which a satellite is still able to communicate with a ground station and vice-versa with $\phi = \pi/2 - \alpha$. ϕ can be obtained alongside other radio parameters from the technical specifications contained in the FCC applications [7] [6] [8] [5]. The equation for the ray cast by Sat2 to a point $p1$ on the surface of the earth, depending on ϕ and the altitude h_s is

$$p(t) = t \begin{pmatrix} 1 \\ -\tan\phi \end{pmatrix} + \begin{pmatrix} 0 \\ h_s \end{pmatrix} \tag{1}$$

This can be used to determine the maximum communication distance of a satellite and any ground station, again using the same line-sphere intersection.

Whether a device can receive a packet, also depends on the received power. It is given as

$$P_{RX} = P_{TX} + G_{TX} - L_{TX} - L_{FS} - L_M + G_{RX} - L_{RX} \tag{2}$$

where P_{TX} is the transmitter power, G_{TX} the transmitter gain, L_{TX} the transmitter loss, L_{FS} the free-space loss, L_M the link margin, $+G_{RX}$ the receiver gain and L_{RX} the receiver loss. All of these parameters can either be directly obtained from the technical specifications or as statistical estimates from the relevant literature [9].

IV. IMPLEMENTATION

The implementation of ns-3-leo is available for the community as download: https://gitlab.ibr.cs.tu-bs.de/tschuber/ns-3-leo. Besides the source code, the repository also contains useful examples on how to configure and run simulations.

We implemented the mobility model and link layer model as a module for *ns-3* version 3.30, which was the newest version at the beginning of the work on the module.

ns-3 provides models with a common way to declare their interface, which contains declarations of the name, attributes, constructors, and trace sources of the model [35]. This has the advantage that all these things can be referred to their semantic names, when the user configures the simulation. An example of such a name would be the trace source for course changes on all mobility models of all nodes in the simulation, which is written to by the *leo mobility model*.

```
/NodeList/*/$ns3::MobilityModel/CourseChange
```

Each component may additionally define a log component. These are primarily meant to be used for debugging and not for tracing, since traces provide better performance and allow for different log output formats. They provide direct access to the traced object and the trace sink may choose whatever output format is available to it (e.g., a plain-text file, or PCAP), while the log component always writes to the standard error output.

ns-3-leo provides these interface and logger definitions to *ns-3* and includes an assortment of classes that help with setting up more complex simulation scenarios, while sacrificing nothing of the extensibility and configurability provided by the model's interfaces.

A. NETWORK TOPOLOGY AND MOBILITY

There are generally two forms of mobility within a satellite network, the constant positions of the user-terminals and gateway stations and the continuously moving satellites, for which we implemented two mobility models. These models provide the channel models with the current position and velocity of each node and their distances to each other, depending on the simulation time.

We provide the simulation with the polar coordinates of each ground station. These are then converted into the ITRF by a custom position allocator. A set of helper classes is available to import polar coordinates from a file and insert them as ground stations into the simulation.

The first method of creating a mobility model for the satellites is by generating the positions ahead of the simulation using SGP4 and importing the waypoints into a waypoint mobility model. We provide helper classes that import the pre-generated waypoints into the simulation.

As mentioned previously in section II, we also want to be able to simulate additional satellites for which we have no orbit parameters in the form of TLE files. For this purpose, *ns-3-leo* contains a mobility model that provides the current position and velocity of satellites, as if they were moving on circular orbits. The model takes the altitude and inclination of the orbits as parameters. Additionally, the time interval after which the position will be recomputed can be configured. Figure 4 shows how the simulator event loop triggers periodic updates of the position stored in the mobility model.

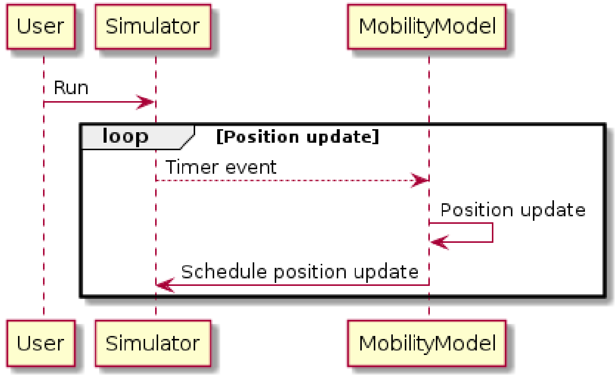

FIGURE 4. Sequence diagram of periodic position update using the simulator event loop.

Each satellite starts out at some position along its orbit at the beginning of the simulation. These positions are generated by the position allocator, which can be configured for each simulation through the provided helper classes. We implemented a simple position allocator that iterates through the orbits of a certain inclination and altitude and passes the position of each satellite inside the orbit and its orbit's location to the mobility model. *ns-3-leo* provides helper classes that import orbit definitions from external files. It is also possible to specify them directly in the user scripts.

B. LINK-LAYER

Since the MAC algorithms and frequency coordination functions that will be used in future LEO mega-constellations are still in the process of being specified [36], we implemented the network devices only as a thin wrapper around a send-queue with no added special behavior, except that it applies the configured error model, receive power thresholds, and maximum data-rates. Since only a few parameters of the ISL and satellite-to-ground links were published, we use the maximum possible data rates [9], representing a best-case scenario. Our network device implementations can handle both unicast and broadcast traffic, since these are necessary for AODV to work. Our network device implementation also provides support for both packet captures and tracing of packets through the simulation.

Network devices transmit packets using the channels they are attached to. Each channel represents a physical medium on which the data is transmitted (e. g. from a radio frequency or a laser-light beam). We implemented two types of channels, one for the inter-satellite links and one for the space-ground links.

We show the sequence of steps involved in the transmission of a packet using our channel implementation in figure 5 and figure 6.

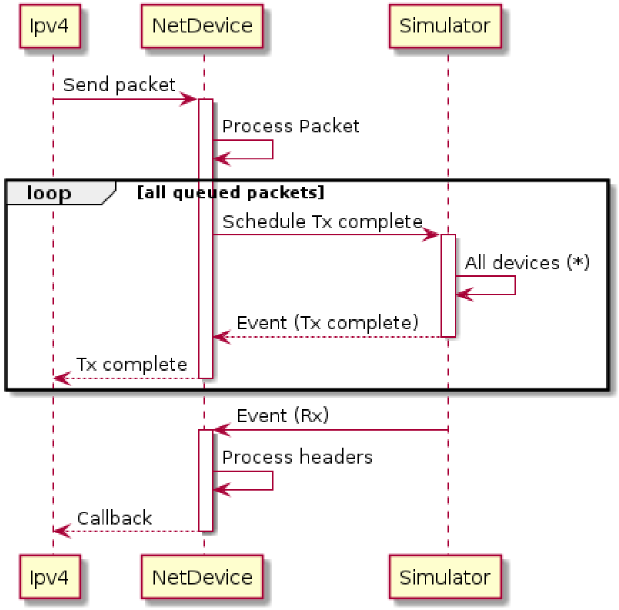

FIGURE 5. Sequence of processing steps in the transmission of a packet using a channel and its attached devices. *Further detailed in figure 6.

The satellite-to-ground channel separates the attached network devices into two sets based on their type, one for ground stations and one for satellite-to-ground links on satellites. It is important to note, that any direct communication within these two sets must be prohibited by the channel. Such communication would only occur if one satellite or ground station would enter the beam of another. Since this is verify uncommon, we purposefully exclude such cases from the channel model. This also improves the performance of the simulation, since a smaller amount of nodes has to be iterated and fewer individual transmission events need to be simulated. This is different from the simulation of a shared medium like a wireless LAN channel.

The propagation loss model determines which nodes of the two sets are reachable for each transmission. Using the method explained in the specification III, it checks first if there is a LOS between the satellite and the ground station and then whether the ground station is inside the satellite's beam or not. For nodes that are in the LOS, the propagation loss model is then used to estimate the received signal strength. We used the statistical estimates from Barrios *et al.* [9] to create an approximate model of the path loss and estimated the propagation delay using the constant speed propagation delay model of *ns-3*.

Our implementation of the ISL channel checks if a LOS between two satellites is possible. For this it uses the line-sphere intersection between the earth and the path between each two satellites as described in section III.

V. EVALUATION

In this section we present our evaluation of *ns-3-leo* and the simulation results for AODV on the Starlink and Telesat LEO constellations. We also discuss how we verified the correctness of the simulation and analyze the performance of our code.

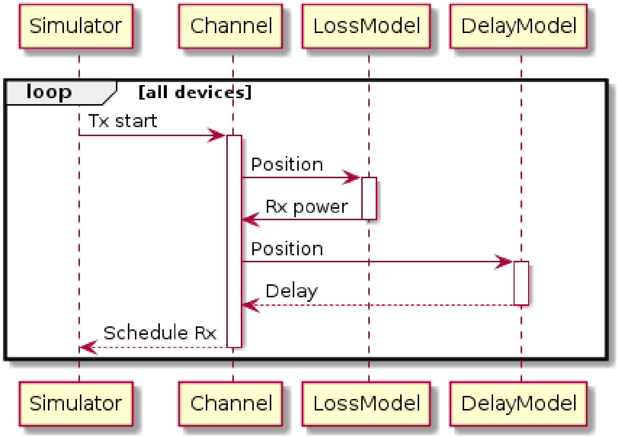

FIGURE 6. Sequence diagram of the processing steps done for each transmission on all devices attached to the channel. This corresponds to "All devices" in figure 5.

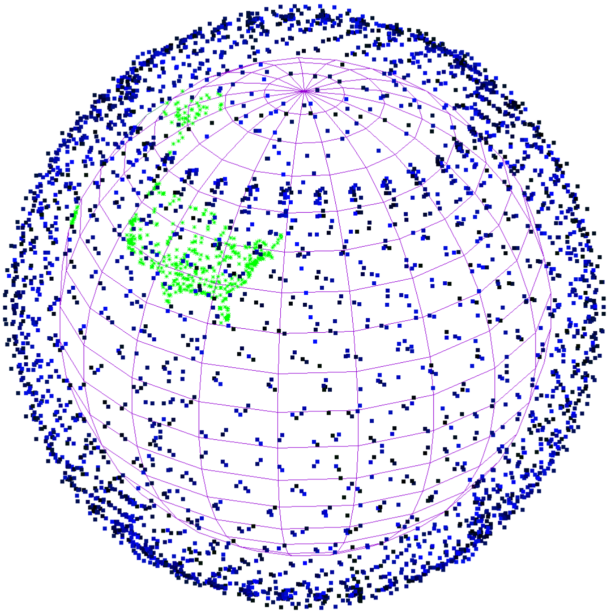

FIGURE 7. Still image from a 24 hours sped-up animation of the Starlink constellation. Airports in the USA are shown in green as an example of areas with a high population density. The full animation can be found online [37].

A. VERIFICATION OF ns-3-leo

To verify that the simulation is correct, the automated test suite covers both core components of *ns-3-leo* and some important helper classes with unit tests and functional tests.

We created an animation of the satellite positions during a simulation run using *gnuplot* and visually compared it to other animations of the same constellation such as the one created by Handley [18]. Figure 7 shows a still image of our animation.

TABLE 1. **Simulation scenarios used in the evaluation. For each scenario the parameters are listed, and a shorthand is assigned that is used to refer to the scenario in the evaluation.**

Script	Constel.	ISL	Acronym
leo-circular-orbit	Starlink	no	OS
	Telesat	no	OT
leo-delay	Starlink	no	DS
	Starlink	yes	DSI
	Telesat	yes	DTI
leo-bulk-send	Starlink	no	BS
	Starlink	yes	BSI
	Telesat	yes	BTI

The test cases for the models are based on the assumptions and expectations listed in the specification. The helper classes are verified by checking if they set up the simulation according to the model attributes and other parameters that are declared for the respective test scenarios. We test the integration of the components by running our example scenarios and check if the results change or errors occur during the test run.

B. TEST SCENARIOS

The evaluation uses three different simulation scripts. Each script is run either on the Starlink or the Telesat constellation and may have ISL enabled. Table 1 lists the different simulation scenarios and the acronyms assigned to their results in the evaluation.

The test script *leo-circular-orbit* calculates the positions of all nodes during the simulation and writes them to a log file. This means that no packets are send during the scenario *OS* and *OT* and only the topologies of the Starlink and Telesat constellations are simulated.

We did not optimize the ground station placement in any way. Instead a 20 × 20 grid of ground stations with equal latitude-longitude is used. For AODV, we entirely disabled the emission of link state *hello* messages, since the large number of resulting messages leads to an infeasible simulation time. To compensate for the longer paths via ground-stations as opposed to links via ISL, we set the threshold value for the progressive ring search of AODV to 20 hops and increase the net diameter to 40 hops to make it possible to still discover end-to-end routes for all nodes in the network. We set the data-rates of all ISL links to 2 Gb/s, which is the same data rate as used in previous simulations [9].

C. SIMULATION PERFORMANCE

We evaluated the performance of our simulation tool by measuring the memory consumption and analyzing the call graph with the instructions per function call using *valgrind*'s *callgrind* tool. *valgrind* is "an instrumentation framework for building dynamic analysis tools" [38] and it allows for a detailed profiling of the our simulation runs. Additionally, we compared the execution time using the *time* command of the Bourne Again SHell (BASH) that starts the simulation script. We performed all simulations using an Intel(R) Core(TM) i7-6700K CPU @ 4.00GHz running Ubuntu Linux 20.04 LTS. *ns-3* runs as a single-threaded process, so only one of the eight available threads of the Central Processing Unit (CPU) is used. Additional factors like the amount of competing tasks and the schedulers used by the Operating System (OS) also influence the performance of the simulation.

TABLE 2. **Performance results of the simulation examples, showing the time needed to run the simulation, the peak memory usage, the amount of instructions spend inside the simulation, and the amount of instructions spend inside the code of *ns-3-leo*. The simulation time is 1000 s for all simulations. The profiler for *DTI* has been manually cancelled after 4 days.**

Scenario	time[s]	Peak[MByte]	#Instr.[10^9]	#I. mod.[10^9]
OS	8.49	29	15.005	0.539
DTI	15469.0	196.4	7928.064	138.257
BSI	525.369	128.0	1616.950	23.049

Simulations of scenarios have ISL enabled always run faster than those of scenarios without ISL, since they create fewer transmission and location update events. The same is true for the scenarios using the Telesat constellation, since it has considerably fewer satellites. The TCP stream of *leo-bulk-send* generates more packets compared to the single ICMP packets created during the script *leo-delay*. The performance evaluation focuses on worst-case scenarios *OS*, *DTI* and *BSI*. Table 2 summarizes the results.

It is important to note that almost the entire simulation time, even 87.90% for *DTI*, is spent on AODV sending, processing and receiving messages, while the network devices of *ns-3-leo* make up for less than 6% of instructions. Despite the relatively expensive operation, the update of the mobility model only takes around 4% of the total instructions. When *BSI* is run with mobility, the TCP connection closes due to a timeout early in the simulation. This results in a shorter simulation time for *BSI* with mobility than without, since after this point, the simulator has nothing to do.

To verify that the high instruction count and therefore larger running time has not been caused by a programming error, but is caused by the overhead of the AODV protocol under mobility, we simulated the same scenarios without updating the mobility model after the initial update. This resulted in 50 s for *DTI* and 199 s for *BSI* and considerably less traffic from AODV protocol messages.

D. PERFORMANCE OF AODV

To demonstrate that *ns-3-leo* can be used in the evaluation of routing protocols on LEO satellite networks, the *leo-loss*, *leo-delay*, and *leo-bulk-send* simulation scripts have been used with various parameters as described in table 1.

1) PACKET-LOSS

Figure 8 shows the percentage of lost packets during different scenarios for the Starlink network (*NM-DSI*, *DSI*, *DS*) and for the Telesat network (*DTI*). We include the artificial scenario

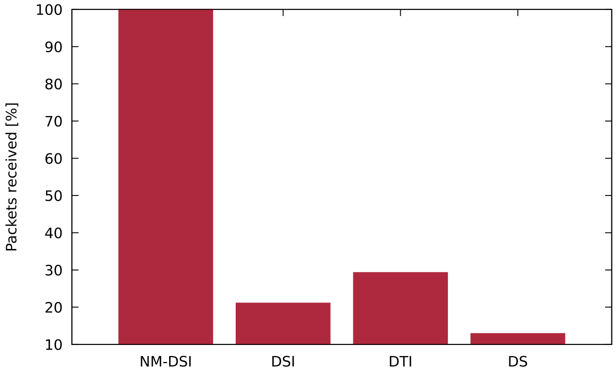

FIGURE 8. **Percentage of receive packets between Middle Europe (51.399, 10.536) and the east coast of North America (40.76, -73.96) over a 1000 s simulation for different scenarios with mobility (DSI, DTI, DS), and without (NM).**

NM-DSI, in which all nodes remain at a fixed position during the simulation, to be able to compare it to the same scenario with mobile satellite (DSI). Similarly, *DS* does not feature any links between satellites, while DSI and DTI do include ISLs. In-case of the Starlink network, AODV does not manage to deliver any packets without the use of ISL.

Without mobility, all packets are received successfully, which also indicates that the simulation software does not contain mistakes that unintentionally drop packets. Under mobility, AODV performs about the same for both constellations.

During any simulation run of AODV on the Starlink network, there were long time-segments during which *AODV* did not discover any viable path. This has been verified using packet captures of the simulation runs.

2) DELAY

The end-to-end delay has been measured together with the packet loss. Figure 9 shows the delay for different scenarios. The delay is the smallest if mobility is disabled (NM-BSI).

For all scenarios, AODV does sometimes not have a valid route at the time the traffic control layer hands down a packet. This causes some outlier values of the forwarding delay, which can be seen from the log output of the AODV module. AODV then buffers the packet and transmits it after a valid route has been acquired. This means that the buffering delay adds to the total delay in some cases. The rest of the variation presumably stems from the traffic of other routing protocol messages and variations in the path length due to mobility.

The statistical outliers with very high latency are significantly fewer for the Telesat constellation (DTI) than for the Starlink constellation (DSI). This indicates that AODV is able to faster acquire a path if there are fewer nodes. As seen with the packet loss, the addition of ISL significantly improves the performance of AODV and the removal of mobility reduces the time spent on finding new routes.

3) THROUGHPUT

We measured the throughput of an end-to-end TCP connection for some scenarios using the *leo-bulk-send* script.

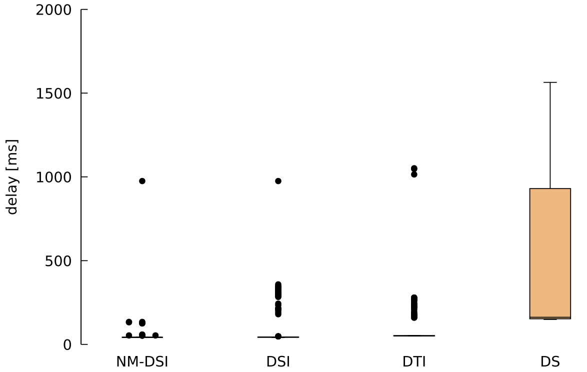

FIGURE 9. **Delay between Middle Europe (51.399, 10.536) and the east coast of North America (40.76, −73.96) over a 1000 s simulation for different scenarios with mobility and without.**

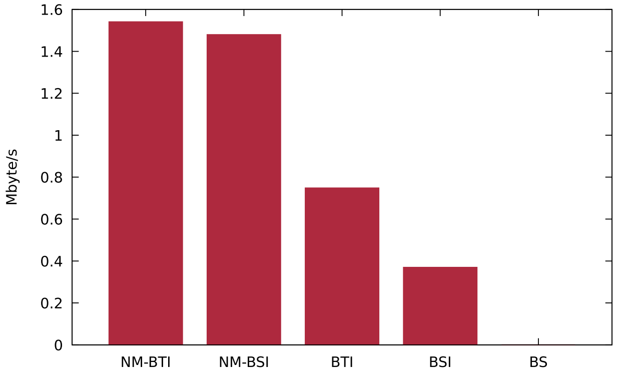

FIGURE 10. **Total throughput of a single TCP connection between Middle Europe (51.399, 10.536) and the east coast of North America (40.76, −73.96) over a 1000 s simulation for different scenarios with mobility and without.**

Figure 10 shows the measured throughput for each scenario.

Almost no payload data is transmitted in-case of the scenario for Starlink without ISL (BS). AODV fails to establish a stable path, as seen in the simulation scenarios that measure the packet loss and the delay. As a result, the TCP connection fails to perform an initial three-way handshake. This has been verified using the packet captures obtained from the simulation runs.

In the same scenarios with ISL enabled (BSI and BTI), AODV manages to establish a somewhat stable path for at least some time and transmits some packets. After about half of the simulation time, the connection is closed due to a timeout, since AODV was not able to reestablish a path in time. It is important to note, that this behavior correlates with the amount of mobility in the network. If the satellites remain stationary, as in NM-BSI, this results in a throughput of about 1.58 Mbit/s for the Telesat and 1.48 Mbit/s for the Starlink constellation on average. The Telesat constellation has fewer satellites overall, and therefore less mobility.

VI. LIMITATIONS AND FUTURE WORK

While *ns-3-leo* already provides a usable framework for simulations of LEO satellite networks, it remains open for extension using the module system of *ns-3*.

We purposefully excluded the behavior of any frequency coordination and medium access functions from our simulations, to focus on the effects of the mobility. This effectively corresponds to a fictitious best-case scenario, where no collisions and overhearing occurs. In the future, the frequency coordination and medium access functions for ISL and satellite-to-ground links, the number of antennas and some other details of the physical layer, such as the path loss through atmospheric effects, could be included into the simulation.

The modeled ISL is able to broadcast to all surrounding satellites. Real-world implementations of ISL use directed beams in e.g., Ka-Band or even laser-light communication, and are only able to transmit the broadcast to one destination at a time, or multiple destinations if there are multiple transceivers, incurring an additional delay while the transceiver switches between destinations. A future routing protocol might make use of these unidirectional links to reduce the amount of overhearing inside the satellite constellation and improve performance. Since AODV benefits from discovering all possible neighbors with as little delay as possible, approximating multiple individual transmission with a single broadcast transmission is sufficient in our case.

We placed ground stations into our simulations using a constantly spaced grid. Our preliminary evaluation suggests that the reliability of Starlink and Telesat would benefit if this placement was optimized in some way, as done by Barrios *et al.* [9].

VII. CONCLUSION

The goal of the creation of *ns-3-leo* was to provide methods which enable simulations of the mobility and link characteristics of LEO satellite mega-constellations as a first step towards an evaluation of the feasibility of WSN and MANET routing protocols for these networks.

ns-3-leo provides detailed insights into the behavior of these networks using various methods of observation, including live traces of network mobility and transmission events, and live captures of the network traffic. At the same time, the module is open to extension through *ns-3*'s interfaces, which makes it possible to reuse the models independently in third party modules and user scripts.

We validated our software using both automated software testing based on its specification and visualizations.

Ad-hoc protocols like AODV have difficulty with the continuously changing paths and intermittent loss of connectivity. This causes a high volume of control traffic and the limits of the reactivity of AODV are reached in many cases, leading to a large performance overhead. For more concrete results and realistic measurements, a more detailed model of the physical and medium access layer would be required, that also models the frequency allocation and medium access functions.

Nevertheless, the evaluation helped to show that routing protocols in LEO mega-constellations face unique challenges from the intermittent connectivity and fast-changing topology that comes with a high degree of mobility of the satellites.

All of our source code as well as useful examples how to make use of ns-3-leo are available online at: https://gitlab.ibr.cs.tu-bs.de/tschuber/ns-3-leo.

REFERENCES

[1] Telecommunication Development Bureau, International Telecommunication Union(ITU). (Jan. 2019). *Measuring Digital Development: Facts and Figures 2019*. [Online]. Available: https://www.itu.int/en/ITU-D/Statistics/Pages/facts/default.aspx

[2] S. R. Pratt, R. A. Raines, C. E. Fossa, and M. A. Temple, "An operational and performance overview of the IRIDIUM low earth orbit satellite system," *IEEE Commun. Surveys*, vol. 2, no. 2, pp. 2–10, 2nd Quart., 1999.

[3] R. A. Wiedeman and A. J. Viterbi, "The Globalstar mobile satellite system for worldwide personal communications," in *Proc. 3rd Int. Mobile Satell. Conf.*, 1993, pp. 285–290.

[4] T. Vega. (Sep. 2020). *FullSAT Space Broadband Solutions*. [Online]. Available: https://www.telespazio-vega.de/images/pdf/FullSAT_DE.pdf

[5] Space Exploration Holdings. (Nov. 2018). *SpaceX Non-Geostationary Satellite System Attachment a Technical Information to Supplement Schedules*. [Online]. Available: https://licensing.fcc.gov/myibfs/download.do?attachment_key=1569860

[6] Application of Kuiper Systems LLC. (2019). *Application of Kuiper Systems LLC for Authority to Launch and Operate a Non-Geostationary Satellite Orbit System Technical Appendix*. [Online]. Available: https://licensing.fcc.gov/myibfs/download.do?attachment_key=1773885

[7] Telesat Canada. (2016). *Telesat Ka-Band NGSO Constellation FCC Filing SAT-PDR-20161115-00108 Attachment Telesat Loi*. [Online]. Available: https://licensing.fcc.gov/myibfs/download.do?attachment_key=1158133

[8] WorldVu Satellites Limited. (2017). *OneWeb Non-Geostationary Satellite System Attachment a Technical Information to Supplement Schedule's*. [Online]. Available: https://licensing.fcc.gov/myibfs/download.do?attachment_key=1134939

[9] I. del Portillo, B. G. Cameron, and E. F. Crawley, "A technical comparison of three low Earth orbit satellite constellation systems to provide global broadband," *Acta Astronautica*, vol. 159, pp. 123–135, Jun. 2019.

[10] M. Handley, "Delay is not an option: Low latency routing in space," in *Proc. 17th ACM Workshop Hot Topics Netw.*, New York, NY, USA, Nov. 2018, pp. 85–91, doi: 10.1145/3286062.3286075.

[11] H. Hauschildt, S. Mezzasoma, H. L. Moeller, M. Witting, and J. Herrmann, "European data relay system goes global," in *Proc. IEEE Int. Conf. Space Opt. Syst. Appl. (ICSOS)*, Nov. 2017, pp. 15–18.

[12] J. A. R. de Azúa, A. Calveras, and A. Camps, "Internet of satellites (IoSat): Analysis of network models and routing protocol requirements," *IEEE Access*, vol. 6, pp. 20390–20411, 2018.

[13] I. del Portillo. (2017). *ITU-Rpy: A Python Implementation of the ITU-R P. Recommendations to Compute Atmospheric Attenuation in Slant and Horizontal Paths*. [Online]. Available: https://github.com/iportillo/ITU-Rpy/

[14] *Gridded Population of the World, Version 4 (GPWV4): Population Count, Revision 11*, Palisades, NY, USA, Center for International Earth Science Information Network—CIESIN—Columbia University, Apr. 2018, doi: 10.7927/H4JW8BX5.

[15] R. J. Leopold, "The iridium communications systems," in *Proc. Singapore ICCS/ISITA*, vol. 2, Nov. 1992, pp. 451–455.

[16] Iridium Satellite LLC. (2013). *Iridium Next Engineering Statement*. [Online]. Available: http://licensing.fcc.gov/myibfs/download.do?attachment_key=1031348

[17] F. Heine, A. Sánchez-Tercero, P. Martin-Pimentel, N. Höpcke, D. Hasler, T. Marynowski, H. Zech, L. Alber, and J. Klement, "Status of Tesat laser communication activities," in *Free-Space Laser Commun. XXXII*, vol. 11272, H. Hemmati and D. M. Boroson, Eds. Bellingham, WA, USA: SPIE, 2020, pp. 19–25, doi: 10.1117/12.2545095.

[18] M. Handley, "Using ground relays for low-latency wide-area routing in megaconstellations," in *Proc. 18th ACM Workshop Hot Topics Netw.*, New York, NY, USA, 2019, pp. 125–132, doi: 10.1145/3365609.3365859.

[19] U. Technologies. (2020). *Unity 3D*. [Online]. Available: https://store.unity.com/

[20] M. Handley. (2020). *Starlink Simulator for Comp0031*. [Online]. Available: https://github.com/mhandley/Starlink0031

[21] "Overview of space communications protocols," Consultative Committee Space Data Syst., Washington, DC, USA, Tech. Rep. CCSDS 130.0-G-3, Jul. 2014. [Online]. Available: https://public.ccsds.org/Pubs/130x0g3.pdf

[22] "CCSDS file delivery protocol (CFDP) recommended standard," Consultative Committee Space Data Syst., Washington, DC, USA, Tech. Rep. CCSDS 727.0-B-4, Jul. 2014. [Online]. Available: https://public.ccsds.org/Pubs/727x0b5.pdf

[23] *Digital Video Broadcasting (DVB); Second Generation Framing Structure, Channel Coding and Modulation Systems for Broadcasting, Interactive Services, News Gathering and Other Broadband Satellite Applications; Part 2: DVB-S2 Extensions (DVB-S2X)*, document Rec. Technical Specification (TS) ETSI EN 302 307-2, ETSI, Aug. 2020, V1.2.1. [Online]. Available: https://www.etsi.org/deliver/etsi_en/302300_302399/30230702/01.02.01_60/en_30230702v010201p.pdf

[24] J. Puttonen, S. Rantanen, F. Laakso, J. Kurjenniemi, K. Aho, and G. Acar, "Satellite model for network simulator 3," in *Proc. 7th Int. ICST Conf. Simul. Tools Techn.*, 2014, pp. 86–91.

[25] H. Bedon, C. Negron, J. Llantoy, C. M. Nieto, and C. O. Asma, "Preliminary internetworking simulation of the QB50 cubesat constellation," in *Proc. IEEE Latin-Amer. Conf. Commun.*, Sep. 2010, pp. 1–6.

[26] T. R. Henderson, M. Lacage, G. F. Riley, C. Dowell, and J. Kopena, "Network simulations with the ns-3 simulator," *SIGCOMM Demonstration*, vol. 14, no. 14, p. 527, 2008.

[27] G. Combs. (Sep. 5, 2020). Wireshark. Wireshark Foundation. [Online]. Available: https://www.wireshark.org/

[28] A.I. Solutions. (Nov. 2020). *FreeFlyer Software*. [Online]. Available: https://ai-solutions.com/freeFlyer/

[29] D. Vallado and P. Crawford, "SGP4 orbit determination," in *Proc. AIAA/AAS Astrodynamics Spec. Conf. Exhib.*, Aug. 2008, p. 6770.

[30] T. Kelso. (Aug. 2020). *Celestrak*. [Online]. Available: https://celestrak.com/

[31] C. Ma *et al.*, "The second realization of the international celestial reference frame by very long baseline interferometry," Tech. Rep. ITN 35, 2009.

[32] Z. Altamimi, P. Rebischung, L. Métivier, and X. Collilieux, "ITRF2014: A new release of the international terrestrial reference frame modeling nonlinear station motions," *J. Geophys. Res., Solid Earth*, vol. 121, no. 8, pp. 6109–6131, Aug. 2016.

[33] F. J. Lohmar, "World geodetic system 1984—Geodetic reference system of GPS orbits," in *GPS-Techniques Applied to Geodesy and Surveying*. Berlin, Germany: Springer, 1988, pp. 476–486.

[34] R. Schwarz. (Sep. 2020). *HPS—Hybrid Simulation Platform for Space Systems*. [Online]. Available: https://www.dlr.de/irs/desktopdefault.aspx/tabid-11348

[35] N. Contributors. (Aug. 2020). *Ns-3 Manual*. [Online]. Available: https://www.nsnam.org/docs/release/3.31/manual/html/index.html

[36] *Study on Using Satellite Access in 5G*, document Rec. Technical Specification (TS) 22.822, Jun. 2018, V16.0.0. [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3372

[37] T. Schubert, U. Kulau, and L. Wolf. (Nov. 14, 2021). Animation Starlink Constellation. IBR. [Online]. Available: https://gitlab.ibr.cs.tu-bs.de/tschuber/project-thesis/-/blob/15c143de694bb2f6903e1c713ebd82595bbb4363/data/starlink-24h.gif

[38] N. Nethercote and J. Seward, "Valgrind: A framework for heavyweight dynamic binary instrumentation," *ACM SIGPLAN Notices*, vol. 42, no. 6, pp. 89–100, 2007, doi: 10.1145/1273442.1250746.

TIM SCHUBERT received the bachelor's degree in computer science from Technische Universität Braunschweig, in 2017. Since 2019, he has been working as a Software Engineer with IServ GmbH, Braunschweig. His interests include protocol design, wireless sensor networks, reliable systems in general, and mobile applications.

LARS WOLF (Senior Member, IEEE) received the Diploma degree in computer science from Universität Erlangen–Nürnberg, in 1991, and the Ph.D. degree in computer science from TU Chemnitz, in 1995. From 1991 to 1996, he worked at IBM's European Networking Center, Heidelberg, Germany. From 1996 to 1999, he was with Technische Universität Darmstadt as an Assistant Professor (Wissenschaftlicher Assistent). He joined Universität Karlsruhe (TH), Germany, in 1999, where he was an Associate Professor with the Computer Science Department and an Alternate Director of the Computer Center. Since Spring 2002, he has been a Full Professor of computer science with TU Braunschweig, Germany, where he was the Head of the Institute of Operating Systems and Computer Networks. His current research interests include wireless networking in general, sensor networks, vehicular networks, delay-tolerant networks, and mobile systems.

ULF KULAU received the bachelor's degree in engineering and the master's and Ph.D. degrees in computer science from Technische Universität Braunschweig, in 2010, 2012, and 2016, respectively. Since April 2018, he has been an Interim Professor at the Computer Engineering and Embedded Computing Research Group, TU Braunschweig, where he was working on hardware/software co-design in embedded systems. Since February 2019, he has been the Head of pre-development at DSI Aerospace Technologie GmbH, where he is investigating new trends in space electronics and technology transfer. Since October 2021, he has been an Assistant Professor at TU Hamburg, where he is leading the Smart Sensors Group. His current research interests include wireless sensor networks (WSNs), microcontroller systems in general, energy constrained computing, and electronics for harsh environments.