Grant Agreement No.: 7750824

Call: IDEJE

Hybrid Integrated Satellite and Terrestrial Access Network

D6.2: Complete PoC framework (HUT+CE+GW)

Work package	WP 6
Subactivity	T6.3
Due date	05/05/2025
Submission date	01/05/2025
Deliverable lead	SEE
Version	1.0
Authors	Dragomir El Mezeni, Haris Turkmanović
Reviewers	Zoran Čiča, Predrag Ivaniš

Document Revision History

Version	Date	Description of change	List of contributor(s)
V0.1	25/04/2025	1st version of D6.2	Dragomir El Mezeni
V0.2	28/04/2025	2nd version of D6.2	Haris Turkmanović
V0.3	30/04/2025	3rd version of D6.2	Predrag Ivaniš
V1.0	01/05/2025	The final deliverable	

COPYRIGHT NOTICE

ACKNOWLEDGMENT

This deliverable has been written in the context of hi-STAR project who has received funding from the Science Fund of the Republic of Serbia, Programme IDEJE under grant agreement n° 7750284.

EXECUTIVE SUMMARY

The hi-STAR project addresses one of the most critical challenges for the next generation wireless networks, which is integration of non-terrestrial networks with terrestrial 5G network. The general objective of WP6 is to integrate an ITCU developed in WP4 with satellite and 5G modems developed in WP3 into the HUT module, alongside the accurate channel emulator for 5G/satcom signal propagation.

This deliverable is a result of the work done in the context of WP6 Subactivity 6.3 (Integration of hybrid modem and channel emulator). We provide detailed description of the proposed simulation framework that integrates all earlier developed components into flexible and versatile proof-of-concept that will be used for demonstration and analysis of different hybrid access scenarios.

TABLE OF CONTENTS

LIST OF FIGURES

SECTION 1 - INTRODUCTION

User mobility support is one of the main and the most attractive features of mobile wireless technologies. Furthermore, user devices typically support more than one wireless technology. Smartphones, for example, support access to mobile networks like 4G or 5G, but at the same time support WiFi access as well. Modern wireless technologies also represent the main driving force for emerging applications in areas such as V2X (Vehicle-to-everything), IoT (Internet of Things), eHealth, etc [1]. The goal of modern communications is to utilize these available multiple network access to provide users better QoE (Quality of Experience), higher communication reliability and availability, and seamless communication sessions [2,3]. 5G technology has recognized this trend and has defined ATSSS (Access Traffic Steering, Switching and Splitting) paradigm that is in line with the aforementioned goal [4].

Satellite technologies have always offered great coverage and access to communication in difficult-to-cover zones. However, the usual downsides were the price, limited upstream bandwidth, sensitivity to weather conditions and high latency. Having this in mind, modern LEO (Low Earth Orbit) satellite communication systems have been emerging to provide much lower latencies, and good bandwidth in both directions (downstream/upstream) at affordable prices that will probably be even lower in the near future. Starlink and OneWeb are typical and successful examples of such systems [5,6]. Also, drone technology evolution and expansion will provide UAV (Unmanned Aerial Vehicle) based extensions of mobile communication networks to support congested cells traffic offload and coverage expansion [7,8].

Given the advancements in LEO satellite communication systems, these systems have been recognized as a good complement to 5G terrestrial systems to provide increased coverage, better reliability and availability, and bandwidth increase to mobile users [9,10]. For these reasons, hybrid satellite/terrestrial network access is gaining attention as a good strategy approach in the near future that would enable many benefits to users [11,12]. Vertical handover can be used to move the ongoing user session between different network technologies like between terrestrial and satellite network access [13]. However, the ATSSS paradigm expands this approach by enabling users to utilize both accesses at the same time. Steering and switching functions can be observed to utilize only one access at a given time, but splitting by definition assumes the usage of multiple network accesses at the same time. ATSSS implementation has two approaches – low level (data link layer) and high level (transport layer). Low level requires that networks in-volved share the same 5G network core, while high level may even be implemented in a way that access networks are unaware of it. Also, it is worth noticing that machine learning and artificial intelligence are becoming more and more popular approaches in optimizing multiple network access utilization according to user preferences such as price, latency, packet loss rate etc. [14,15].

To estimate the quality of multiple network accesses approaches and/or used techniques (optimization techniques, machine learning...) it would be very beneficial to have simulation frameworks and platforms in order to generate diverse and plentiful data [16]. Simulations can efficiently and economically provide insight in performances of the tested approaches [17]. Based on these insights, further tuning and elimination of detected issues can be done. Software simulations are a more economical approach, but hard-ware-based simulations can provide shorter simulation time and can use hardware accelerators to not only reduce the simulation time but also to enable inspection of features that are not feasible in software (for example, LDPC (Low Density Parity Check) coding/decoding [18]).

In this deliverable, we present a flexible framework that is intended for an evaluation of hybrid terrestrial/satellite access performances. Also, we introduce an intelligent selection of network access using the predictions done by long-short-term neural networks (LSTMs) based on the measured SNR (Signal Noise Ratio). Appropriate channel models are used in the simulation process, generating SNR data for both satellite and terrestrial network channels. Demonstration is done for UDP streaming based session where intelligent selector, based on SNR measurements, selects better access that provides higher bandwidth.

The main contribution of this deliverable is the design of flexible hardware-based simulation platform that is intended for inspection of high-level based multi-access approaches and protocols. Other contributions include performance testing for UDP based streaming sessions and performance measurements using the proposed hardware-based simulation platform.

SECTION 2 – POC ARCHITECTURE

The overall architecture of the simulation framework, intended to enable simulation of multi-RAT access, is illustrated in Figure 1. It contains basic elements, **modules**, that mimic multi-RAT access scenario:

- **HUT (Hybrid User Terminal)** – user equipment (UE) capable of utilizing two different RATs – in this paper these are terrestrial and satellite RATs.
- **Transmitter/receiver (T/R)** – to simulate Tx/Rx module of HUT
- **Channel** – to simulate selected radio channel models.

This approach enables flexible adaptations to the various use case scenarios. In the next section, simulation results for UDP streaming session are given. The initial development stage of the proposed framework has been presented in [19].

The core functionalities of the simulation framework utilized in this research are detailed in [19]. For this study, the framework has been modified to support terrestrial and satellite channel models. The overall architecture of the simulation framework is illustrated in Figure 1.

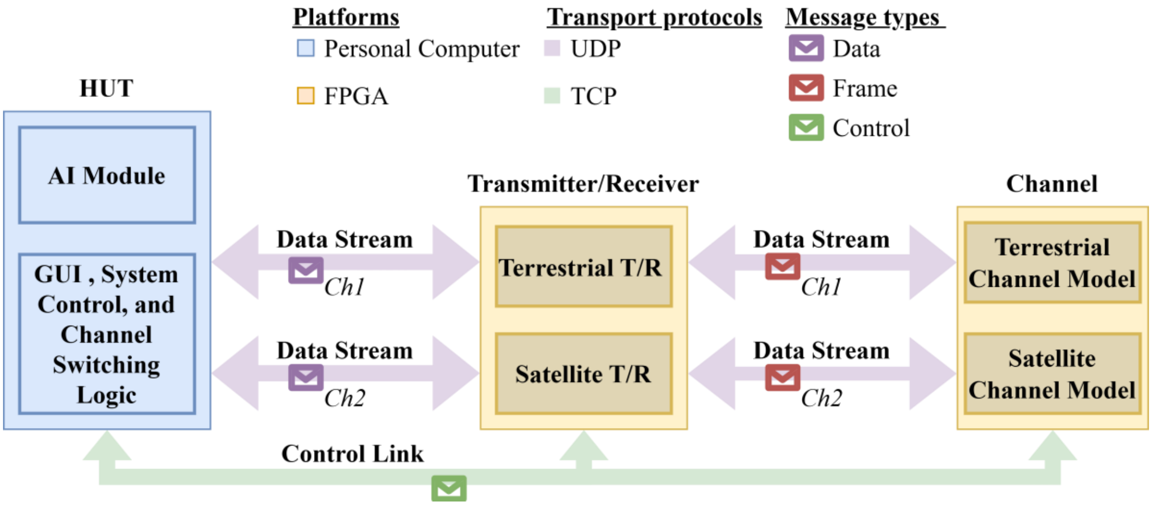

Figure 1. Architecture overview of Distributed Simulation Framework.

Two main building components of developed simulation framework are:

- **Modules** – software or hardware components that can generate, receive and process user data.
- **Messages** – encapsulations of user and control data that are exchanged between modules.

The simulation framework is implemented as a distributed solution that includes three software modules, deployed on separate physical platforms, that are in charge of generating, receiving, transmitting and processing user data. Two types of links are established between the modules of the simulation framework:

- **Data Stream Link** – UDP based
- **Control Link** – TCP based

User data that is transmitted within simulation framework are encapsulated by Messages that have a specific format detailed in the next section.

SECTION 3 – MESSAGES

Configuration and data flow between modules is implemented using Messages. Although the developed framework architecture is flexible and enables definition of different message types, for the purpose of the Proof of Concept for this project, three message types are defined: Data, Frame and Control.

Data Messages are generated by the HUT and are used to transfer information to and from the T/R module. These messages are transmitted over the UDP protocol in a binary format, as illustrated in Figure 2. The message format consists of two main parts: a header and a payload. The header includes five 4-byte fields. The first field, Req/Res, indicates whether the message is sent from the HUT to the Channel module (uplink), or a response sent by the Channel module to the HUT module (downlink). Each Data Message has a unique Data ID, which helps packet loss detection, since this mechanism is not provided by the UDP protocol. Additionally, the Slot ID field identifies the timeslot to which the Data Message belongs. HUT module makes predictions, and timeslot represents a time period for which one prediction is considered valid. The Metadata field is a bit-accessible field that contains other miscellaneous parameters and control signals needed for data processing across the modules. The MODCOD (modulation and coding) field contains an index representing the modulation and code rate used for the current transmission. The payload section consists of n bits of user data.

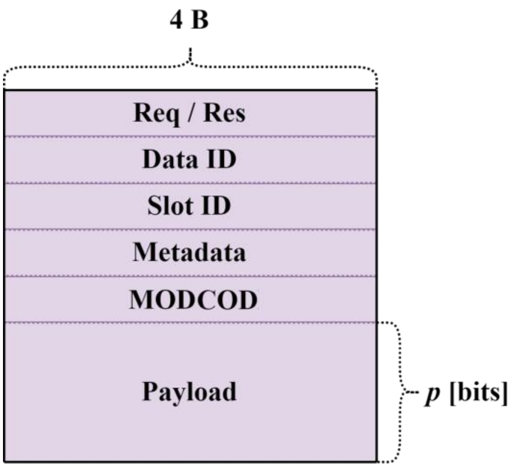

Figure 2. Data message format.

Frame Messages are generated within the T/R module, where they encapsulate the content of Data Messages in the appropriate terrestrial or satellite physical layer format. Once created, these messages are transmitted from the T/R module to the Channel module via a UDP link, where they undergo further processing within the corresponding channel model.

Control Messages are implemented using ASCII string format. These messages are generated on the HUT side and transmitted to the other modules using the TCP protocol. For every Control Message received and processed by the modules, a corresponding response is generated to indicate the execution status. For example, the module supports the "hello" control message, which serves to verify the module's presence at a specific IP address. The command for this check is "device hello", and if the module software is running on the platform, the expected response is "OK".

SECTION 4 – MODULES

The current simulation framework includes the following software modules:

- Hybrid User Terminal (HUT),
- Transmitter/Receiver (T/R),
- Channel

These modules are instances of simulator components HUT Software Component and Device Software Component which are defined in [19].

The **HUT module** is an instance of HUT Software Component and is deployed on a PC. It is basically a GUI application responsible for generating user data and selecting the active communication channel. Additionally, it configures other simulation framework components such as the T/R and Channel modules.

The **T/R and Channel modules** are instances of Device Software Component and are deployed onto dedicated embedded platform that can enable processing acceleration.

The **T/R module** adds specific frame header to Data Messages, which contain user data, and forms Frame Messages that contain coded data. This module is also responsible for decoding process e.g. extracting Data Messages from received Frame Messages.

The **Channel module** is responsible for modeling the behavior of terrestrial and satellite channels.

4.1. HYBRID USER TERMINAL (HUT)

The HUT module is implemented as a Graphical User Interface (GUI) application written in C++ language, developed using the Qt framework [46], which ensures portability across different operating systems. Overall, HUT architecture is presented in Figure 3.

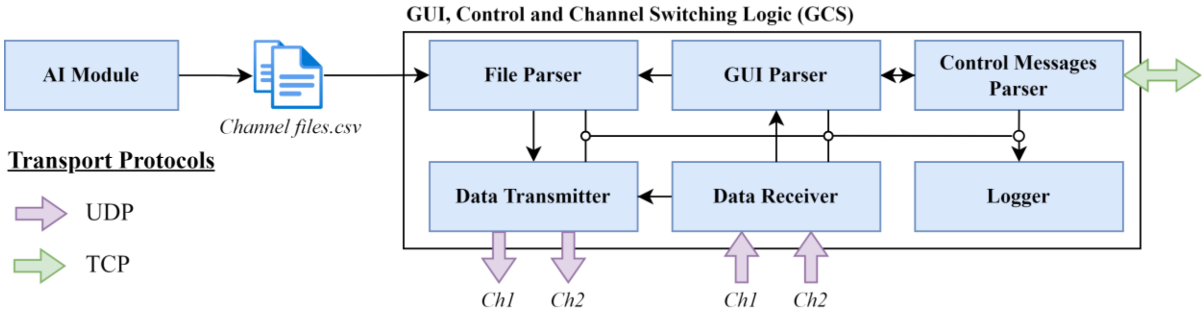

Figure 3. Architecture of the HUT module.

HUT consists of two main components:

- AI Module (AIM),
- GUI, Control and Channel Switching Logic (GCS).

The detailed description of the functionalities integrated within the **AI module** will be described in detail alongside the channel simulation description in Deliverable 6.3. These functionalities primarily involve the decision-making process that takes several samples of the measured SNR sequence for a given channel and tries to predict an optimal MODCOD that should be used in the following timeslot. This algorithm is developed to keep the transmission error rate under a certain margin, while providing the highest possible spectral efficiency, hence enabling the fastest data transmission rate without errors under the given channel conditions. Based on predicted MODCOD, maximal number of frames that can be transmitted over each channel in the following timeslot is calculated.

This value is used by the GCS on the HUT side to select the best channel for frame transmission. In the simulation environment, this channel is called Active channel. Since MODCOD, used for channel switching, is predicted based on earlier SNR values it should be compared to the actual MODCOD calculated using the channel model in the Channel module block.

The **GCS** is a component of the HUT Module responsible for configuring all modules, supervising communication, and managing data frame routing. This component consists of the following software submodules: GUI Parser, File Parser, Data Transmitter, Data Receiver, and Control Message Parser.

The **GUI Parser** is responsible for generating GUI control elements and processing user-defined configuration parameters through the corresponding control inputs. Once a configuration is received, it is forwarded to the **Control Message Parser** submodule, which generates control message commands from configuration parameters. Besides configuration of simulation framework, the GUI Parser also load files containing user data. Once loaded, this file is sent to the **File Parser** submodule. Here user data is divided into code words whose number depends on used MODCOD as shown in Fig. 4. This data represents a payload which is transferred inside Data message frame as shown in Fig. 2.

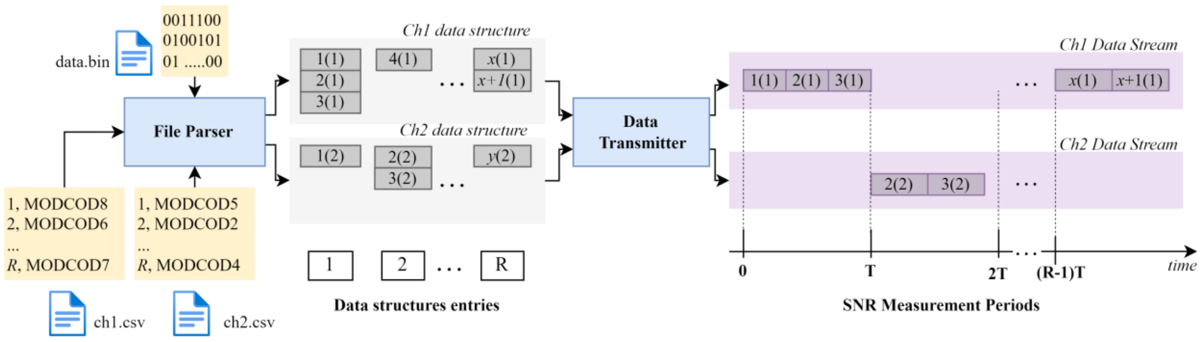

Figure 4. File parsing and Data Transmission mechanism.

MODCOD that is used for current data transmission is provided by the AI module. Since different channels have different characteristics and instantaneous conditions, different sets of MODCOD parameters are provided for each channel. The MODCOD file for each channel contains R rows, representing R timeslots. The **File Parser** submodule generates a data structure for each channel that contains information for each of R timeslots. Each element of this structure contains N substructures, representing the data messages that should be sent during a single timeslot if the corresponding channel is selected. The value of N, as well as the data message transmission period within a single timeslot, is determined based on the MODCOD and the preconfigured bandwidth for each channel.

Once these structures are generated, they are forwarded to the Data Transmitter submodule. A key function of this submodule is to determine which channel should be active in the given timeslot. The selection algorithm is triggered periodically at the end of each timeslot, and the active channel is chosen with a goal of maximizing number of frames that can be transmitted in the given timeslot. In addition to selecting the active channel, Data Transmitter submodule formats data from channel data structure and transmits it over the selected active channel using UDP transport protocol. This data is sent to the T/R module. Figure 4 illustrates the file parsing, data splitting into packets and data transmission process on the HUT side. In the demonstrated example the duration of timeslot is set to T, and the entire simulation contains R timeslots.

Data Receiver submodule receives Data Messages from the T/R module, extracts all relevant information and updates the associated channel data structure. Update of channel data structure triggers the Logger submodule that records all Data Receiver activities into a log file. This log file is later analyzed to extract relevant information

4.2. TRANSMITTER/RECEIVER (T/R) AND CHANNEL

The Transmitter/Receiver (T/R) and Channel modules are responsible for transmitting, receiving, processing, and routing data and frame messages. Both modules are instances of Device Software Component (Device) illustrated in Figure 5 that is introduced in [19]. Device Software Component is written in C and is designed as a multi-threaded application for the Linux operating system.

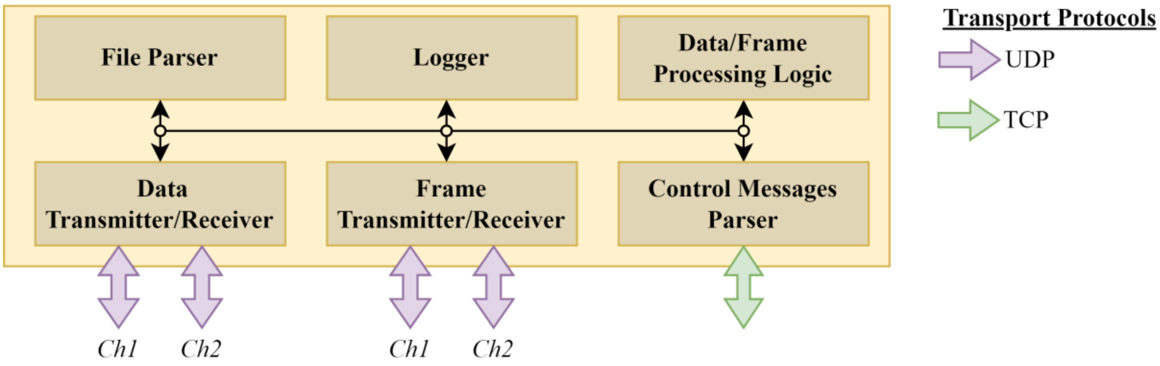

Figure 5. Architecture of the Device Software Component.

The Device Software Component is composed of several software submodules: File Parser, Logger, Data/Frame Processing Logic, Data Transmitter/Receiver, Frame Transmitter/Receiver and Control Messages Parser. It can be configured to operate as Transmitter/Receiver module or the Channel module. The configuration process is done from the HUT module via control interface during simulation initialization phase. Figure 6 illustrates the configuration of T/R and Channel Module, as well as connection between them.

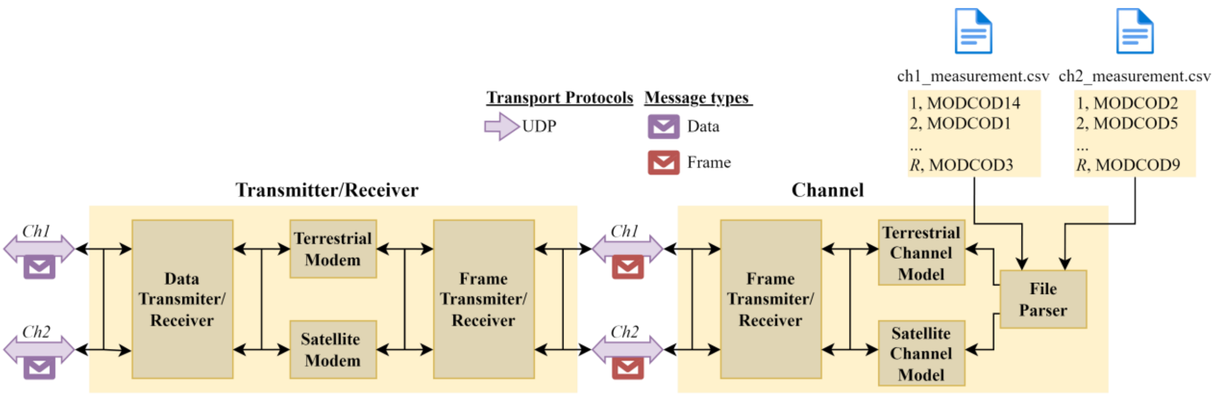

Figure 6. Configuration of the T/R and Channel Modules as well as a connection between them.

When the Device Software Component is configured to operate in Transmitter/Receiver (T/R) mode, all submodules shown in Figure 5 are utilized, except for the File Parser submodule. The primary function of the T/R module is to process Data Messages received from the HUT module at the uplink as well as Frame Messages received from the Channel module at the downlink over the corresponding channel. At the uplink it encodes Data Messages using appropriate channel modems, encapsulate the payload and header of these messages into terrestrial or satellite physical layer format thus producing Frame Messages that are sent to the Channel module. At the downlink it decodes Frame Messages received from the Channel module and extracts Data Messages that are sent back to the HUT module. When simulation is started, Logger submodule sends corresponding messages over UART interface which can be used for debugging.

When the Device Software Component is configured to operate in Channel mode, all submodules shown in Figure 5 are utilized, except for the Data T/R submodule. The primary function of the Channel module is to process Frame Messages received over the uplink, from the T/R module, by applying appropriate channel models thus simulating real-life scenarios. One of the key functions of the channel processing is to determine whether a Frame Message should pass or should be dropped. If the MODCOD, used in the current Frame Message, is not strong enough to handle channel SNR, then the frame should be dropped since it cannot be decoded successfully at the receiver's side. Otherwise, the Frame Message should pass to the downlink and almost certainly will be de-coded successfully in the T/R module. Dropping mechanism is implemented by comparing MODCOD used for the current Frame Message encoding with optimal MODCOD that should be used considering current channel conditions. The first MODCOD, that is used for Frame Message coding, is prediction provided by the HUT module, while latter MODCOD is value produced by Channel model and represents optimal code considering channel conditions. If the first MODCOD, used for encoding, has a larger SNR threshold value than the optimal MODCOD, Frame Message will be dropped, otherwise it will pass to downlink.

SECTION 5 CONCLUSION

In this Deliverable we have proposed a simulation framework for performance evaluation of hybrid satellite-terrestrial network access. This framework integrates all previously developed components such as Hybrid User Terminal GUI, Transmitter/Receiver modules as well as accurate Channel model for both satellite and terrestrial links. Framework also contains AI part that enables intelligent selection of better network access. Framework is successfully demonstrated on one use case scenario and the results obtained with this framework will be presented in Deliverable 6.3. This demonstration shows the potential of our proposed framework. By simply changing experimental setup, one can easily and very fast gain insight in relation between network accesses, and which access would be used more in a hybrid session. It is important to emphasize that, although this framework was used for specific proof-of-concept, presented architecture is more general and can support many different use cases.

REFERENCES

[1] Jorswieck, E.A. Next-Generation Multiple Access: From Basic Principles to Modern Architectures. *Proc. IEEE* **2024**, *112*, 1149–1178, doi:10.1109/JPROC.2024.3412423.

[2] Sultan, M.T.; Sayed, H. El QoE-Aware Analysis and Management of Multimedia Services in 5G and Beyond Heterogeneous Networks. *IEEE Access* **2023**, *11*, 77679–77688, doi:10.1109/ACCESS.2023.3298556.

[3] Soleymani, M.; Santamaria, I.; Jorswieck, E.; Rezvani, S. NOMA-Based Improper Signaling for Multicell MISO RIS-Assisted Broadcast Channels. *IEEE Trans. Signal Process.* **2023**, *71*, 963–978, doi:10.1109/TSP.2023.3259145.

[4] Wu, H.; Ferlin, S.; Caso, G.; Alay, Ö.; Brunstrom, A. A Survey on Multipath Transport Protocols Towards 5G Access Traffic Steering, Switching and Splitting. *IEEE Access* **2021**, *9*, 164417–164439, doi:10.1109/ACCESS.2021.3134261.

[5] Mohan, N.; Ferguson, A.E.; Cech, H.; Bose, R.; Renatin, P.R.; Marina, M.K.; Ott, J. A Multifaceted Look at Starlink Performance. In Proceedings of the Proceedings of the ACM Web Conference 2024; Association for Computing Machinery: New York, NY, USA, 2024; pp. 2723–2734.

[6] Moon, T.; Chu, H.; Lee, W.; Kim, J.-H. Performance Analysis of SpaceX and OneWeb Satellites in South Korea. In Proceedings of the 2024 15th International Conference on Information and Communication Technology Convergence (ICTC); 2024; pp. 1802–1803.

[7] Mohsan, S.A.H.; Khan, M.A.; Noor, F.; Ullah, I.; Alsharif, M.H. Towards the Unmanned Aerial Vehicles (UAVs): A Comprehensive Review. *Drones* **2022**, *6*, doi:10.3390/drones6060147.

[8] Tsipi, L.; Karavolos, M.; Vouyioukas, D. An Unsupervised Machine Learning Approach for UAV-Aided Offloading of 5G Cellular Networks. *Telecom* **2022**, *3*, 86–102, doi:10.3390/telecom3010005.

[9] Darwish, T.; Kurt, G.K.; Yanikomeroglu, H.; Bellemare, M.; Lamontagne, G. LEO Satellites in 5G and Beyond Networks: A Review From a Standardization Perspective. *IEEE Access* **2022**, *10*, 35040–35060, doi:10.1109/ACCESS.2022.3162243.

[10] Sylla, T.; Mendiboure, L.; Maaloul, S.; Aniss, H.; Chalouf, M.A.; Delbruel, S. Multi-Connectivity for 5G Networks and Beyond: A Survey. *Sensors* **2022**, *22*, doi:10.3390/s22197591.

[11] Zhu, X.; Jiang, C. Integrated Satellite-Terrestrial Networks Toward 6G: Architectures, Applications, and Challenges. *IEEE Internet Things J.* **2022**, *9*, 437–461, doi:10.1109/JIOT.2021.3126825.

[12] Zhu, X.; Jiang, C. Integrated Satellite-Terrestrial Networks Toward 6G: Architectures, Applications, and Challenges. *IEEE Internet Things J.* **2022**, *9*, 437–461, doi:10.1109/JIOT.2021.3126825.

[13] Shayea, I.; Dushi, P.; Banafaa, M.; Rashid, R.A.; Ali, S.; Sarijari, M.A.; Daradkeh, Y.I.; Mohamad, H. Handover Management for Drones in Future Mobile Networks—A Survey. *Sensors* **2022**, *22*, doi:10.3390/s22176424.

[14] Wu, H.; Alay, Ö.; Brunstrom, A.; Ferlin, S.; Caso, G. Peekaboo: Learning-Based Multipath Scheduling for Dynamic Heterogeneous Environments. *IEEE J. Sel. Areas Commun.* **2020**, *38*, 2295–2310, doi:10.1109/JSAC.2020.3000365

[15] Zhang, H.; Li, W.; Gao, S.; Wang, X.; Ye, B. ReLeS: A Neural Adaptive Multipath Scheduler Based

on Deep Reinforcement Learning. In Proceedings of the IEEE INFOCOM 2019 - IEEE Conference on Computer Communications; 2019; pp. 1648–1656.

[16] Broner, M.; Lee, S.; Jin, L.; Stoleru, R. 5G-MANTRA: Multi-Access Network Testbed for Research on ATSSS. In Proceedings of the Proceedings of the 3rd ACM Workshop on 5G and Beyond Network Measurements, Modeling, and Use Cases; Association for Computing Machinery: New York, NY, USA, 2023; pp. 8–13.

[17] Gomez, J.; Kfoury, E.F.; Crichigno, J.; Srivastava, G. A Survey on Network Simulators, Emulators, and Testbeds Used for Research and Education. *Comput. Networks* **2023**, *237*, 110054, doi:https://doi.org/10.1016/j.comnet.2023.110054.

[18] Ferraz, O.; Subramaniyan, S.; Chinthala, R.; Andrade, J.; Cavallaro, J.R.; Nandy, S.K.; Silva, V.; Zhang, X.; Purnaprajna, M.; Falcao, G. A Survey on High-Throughput Non-Binary LDPC Decoders: ASIC, FPGA, and GPU Architectures. *IEEE Commun. Surv. Tutorials* **2022**, *24*, 524–556, doi:10.1109/COMST.2021.3126127.

[19] Turkmanović, H.; El Mezeni, D.; Čiča, Z.; Saranovac, L. Distributed Simulation Framework for Assesing Multipath Transport Protocols. In Proceedings of the 2024 32nd Telecommunications Forum (TELFOR); 2024; pp. 1–4.