D5.4: Final Report on Multi-Connectivity and Software Defined Network Control

Revision: v.1.5

Work package	WP 5
Task	Tasks 5.1 and 5.3
Due date	31/03/2026
Submission date	30/03/2026
Deliverable lead	Benjamin Barth (DLR)
Version	1.5
Authors	Benjamin Barth, Roshith Sebastian, (DLR), Marius Corici, Hemant Zope, Hauke Buhr, Manar Zaboub (FHG), Nuria Candelaria Trujillo Quijada (HSP), Nouhaila Ajdor, Laurent Reynaud, Emile Stephan (ORA), Daniele Tarchi, Roberto Picchi (CNIT)
Reviewers	Justin Tallon (SRS), Eddi Gonzalez (HSP)
Abstract	In order to integrated TN and NTNs for an Quality of Service improvement, in this deliverable Multi-Connectivity on transport layer is investigated following an adaptation of 3GPP Access Traffic Steering, Splitting and Switching (ATSSS). Architectural enablers are identified for the 5G-STARDUST direct and indirect scenario. Four complementary optimization strategies are researched: 1) AI/ML for scheduling; 2) scheduler hot swapping; 3) hand-over predictions; and 4) a joint MPTCP-MPQUIC controller. Furthermore, routing and hand-over aspects in such networks have been elaborated based of SDN.
Keywords	Multi-Connectivity, Software Defined Network Control, ATSSS, MPTCP, MPQUIC

Co-funded by the European Union 6GSNS
Grant Agreement No.: 101096573
Call: HORIZON-JU-SNS-2022
Topic: HORIZON-JU-SNS-2022-STREAM-A-01-02
Type of action: HORIZON-JU-RIA

Document Revision History

Version	Date	Description of change	List of contributor(s)
V0.1	29/08/2024	1st edit based on D5.2, draft ToC update	Benjamin Barth
V0.2	05/09/2024	First input AI section	Laurent Reynaud
V0.3	28/11/2024	Updated Structure	Benjamin Barth
V0.4	27/03/2025	First version of transport protocols	Emile Stephan
V0.5	08/05/2025	Update of predictive scheduler	Benjamin Barth
V0.6	15/09/2025	Contribution on AI/ML section	Laurent Reynaud, Roshith Sebastian
V0.7	25/09/2025	Update SDN section, update transport protocols, update AI section	Marius Corici, Laurent Reynaud, Benjamin Barth, Roshith Sebastian, Nouhaila Ajdor
V1.0.F	25/09/2025	QA ready version	Benjamin Barth
V1.1	13/03/2026	CNIT contribution to section 3	Daniele Tarchi, Roberto Picchi
V1.2	18/09/2026	ORA update to section 3	Nouhaila Ajdor, Laurent Reynaud
V1.3	23/03/2026	DLR update section 3 and editing	Roshith Sebastian, Benjamin Barth
V1.4	24/03/2026	HSA QA review	Eddi Gonzalez
V1.5	26/03/2026	Final revision according to the received comments	Benjamin Barth
V1.5.F	30/03/2026	Approved for the EC portal upload	Tomaso de Cola

DISCLAIMER

Co-funded by the European Union 6GSNS

5G-STARDUST (*Satellite and Terrestrial Access for Distributed, Ubiquitous, and Smart Telecommunications*) project has received funding from the Smart Networks and Services Joint Undertaking (SNS JU) under the European Union's Horizon Europe research and innovation programme under Grant Agreement No 101096573.

This work has received funding from the Swiss State Secretariat for Education, Research and Innovation (SERI).

Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.

 © 2023-2025 5G-STARDUST

COPYRIGHT NOTICE

Creative Commons Attribution-ShareAlike 4.0 International

Project co-funded by the European Commission in the Horizon Europe Programme		
Nature of the deliverable:	**R**	
Dissemination Level		
PU	*Public, fully open, e.g. web (Deliverables flagged as public will be automatically published in CORDIS project's page)*	✓
SEN	*Sensitive, limited under the conditions of the Grant Agreement*	
Classified R-UE/ EU-R	*EU RESTRICTED under the Commission Decision No2015/ 444*	
Classified C-UE/ EU-C	*EU CONFIDENTIAL under the Commission Decision No2015/ 444*	
Classified S-UE/ EU-S	*EU SECRET under the Commission Decision No2015/ 444*	

* R: Document, report (excluding the periodic and final reports)

DEM: Demonstrator, pilot, prototype, plan designs

DEC: Websites, patents filing, press & media actions, videos, etc.

DATA: Data sets, microdata, etc.

DMP: Data management plan

ETHICS: Deliverables related to ethics issues.

SECURITY: Deliverables related to security issues

OTHER: Software, technical diagram, algorithms, models, etc.

© 2023-2025 5G-STARDUST

EXECUTIVE SUMMARY

This deliverable is the final issue of the previous version of D5.2 which gave the preliminary findings and presents the final results from the 5G-STARDUST project with respect to multi-connectivity Terrestrial and Non-Terrestrial networks (TN/NTN) within an SDN-based architecture, which focuses on seamless integration of TN and NTN. The main highlights are then reported below.

3GPP Access Traffic Steering, Splitting and Switching (ATSSS) has been taken as the baseline to establish multi-connectivity to Terrestrial and Non-terrestrial Networks. Although it is possible to connect to multiple TNs or to NTNs, the main focus here is to connect User Equipment (UEs) via TN and NTN simultaneously. The legacy ATSSS approach supports dual connectivity by connecting to a 3GPP access and a non 3GPP access. Hence, in this work the current definition has been extended to realise connectivity to a single common core from multiple 3GPP access networks. The control plane aspects to realize this have been evaluated and the necessary enhancements have been suggested. The architectures considering direct and indirect access, via IAB-like nodes have been discussed with attention given also to the protocol stack adaptations. The advantages that the use of AI/ML can bring into the optimization process have been considered in the architectural design and two scheduling approaches are presented for higher layer connectivity. First a predictive scheduler is used that builds on WP4 federated Kolmogorov Arnold networks (KANs) to forecast the system load for a specific point in time. This information has been added to the base of knowledge of a scheduler selecting between TN and NTN. Second, the hand over timing between LEO satellites is predicted and a scheduling framework is presented that can make use of this information. Furthermore, a hot swapping has been designed and tested, which makes use of AI/ML that selects for the current network conditions the best set of scheduling strategy and congestion control algorithm. Lastly, a common MPTCP-MPQUIC controller is introduced. The deliverable presents the final developments of simulations and emulation testbeds to verify the suggested solutions for multi-connectivity.

For enabling the control of data path across the transport network an SDN based management is proposed which enables the transmission of routing related information prior to their usage, with this being able to immediately adapt to topology changes. This has been extended by a framework for simulating and performing conditional handover for mobility management in 5G NTN considering the predictable yet continuously evolving connectivity conditions of large LEO mega-constellations. In such environments, multiple satellites may simultaneously provide feasible access while the selected attachment point determines the subsequent end-to-end transport path towards gateway infrastructure. Consequently, mobility decisions driven solely by signal-strength indicators do not necessarily ensure either long connection persistence or transport-efficient routing, since they do not account for constellation-level dynamics. Predictive indicators are derived from a constellation digital twin implemented using the Fraunhofer FOKUS OpenLANES toolkit[104], providing awareness of routing evolution beyond instantaneous visibility conditions. Evaluation on a large LEO mega-constellation shows that transport-aware attachment improves latency consistency, while persistence-aware selection reduces unnecessary handover activity. The results highlight the benefit of integrating network-level performance indicators into conditional hand over design for future NTN deployments.

As an update of D5.2, this document includes the previous content. Newly introduced in this issue are:

- In section 2 we added a paragraph about Wireless Access Backhaul (WAB). Section 2.3 has been added dealing with transport protocols for Multi-Connectivity (MC).

 © 2023-2025 5G-STARDUST

- Section 3 has been restructured and is dealing with AI/ML for multi-connectivity. Four approaches, hand-over prediction, system load prediction and hot swapping have been added together with a joint MPTCP-MPQUIC controller. Previous part of the Mininet testbed has been extended with the indirect scenario. The section provides implementation detail of the four approaches and provides the test-bed results.

- Section 4 on software defined networking control has been fully reworked and is providing details on the simulation environment and presents the results. Furthermore, a full section of conditional hand-over has been included.

- Section 5, the conclusion has been updated.

© 2023-2025 5G-STARDUST

TABLE OF CONTENTS

© 2023-2025 5G-STARDUST

LIST OF FIGURES

© 2023-2025 5G-STARDUST

 © 2023-2025 5G-STARDUST

 © 2023-2025 5G-STARDUST

© 2023-2025 5G-STARDUST

LIST OF TABLES

© 2023-2025 5G-STARDUST

ABBREVIATIONS

3GPP	3rd Generation Partnership Project
5G NR	5G New Radio
5GC	5G Core
AI	Artificial Intelligence
AMF	Access and Mobility Management Function
AnLF	Analytics Logical Function
ATSSS	Access Traffic Steering, Splitting and Switching
BAP	Backhaul Adaptation Protocol
CA	Congestion Control Algorithm
CC	Congestion Control
CHO	Conditional Handover
CLI	Command Line Interface
CN	Core Network
CoMP	Coordinated Multi-Point
COTS	Commercial off-the-shelf
CU	Central Unit
DN	Data Network
DU	Distributed Unit
DVB	Digital Video Broadcasting
eBPF	Extended Berkeley Packet Filter
FR	Frequency Range
FRR	Fast Reroute
gNB	5G Node B
GEO	Geostationary Earth Orbit
GS	Ground Stations
GTP	General Packet Radio Service (GPRS) Tunnelling Protocol
HoL	Head of line Blocking

© 2023-2025 5G-STARDUST

HTTP	Hypertext Transfer Protocol
IAB	Integrated Access Backhaul
IE	Information Element
IETF	Internet Engineering Task Force
IP	Internet Protocol
ISL	Inter-Satellite Link
LDP	Label Distribution Protocol
LEO	Low Earth Orbit
MA	Multi-Access
MAI	Measurement Assistance Information
MAR	Multi-Access Rule
MC	Multi-Connectivity
MP	Multi-Path
MPLS	Multiprotocol Label Switching
MPTCP	Multi-Path Transmission Control Protocol
MPQUIC	Multi-Path Quick User Datagram Protocol Internet Connection
ML	Machine Learning
MTLF	Model Training logical function
N3IWF	Non-3GPP Inter-Working Function
NAS	Non-Access Stratum
NOC	Networks on Chip
NF	Network Function
NTN	Non-Terrestrial Network
NWDAF	Network Data Analytics Function
OSPF	Open Shortest Path First
PCF	Policy Control Function
PDU	Protocol Data Unit
PLMN	Public Land Mobile Network

© 2023-2025 5G-STARDUST

PM	Path Manager
PMF	Performance Measurement Function
PoC	Proof of Concept
PS	Predictive Scheduler
PSA	PDU Session Anchor
QoS	Quality of Service
RAN	Radio Access Network
RD	Redundant
RFC	Request for Comments
RRC	Radio Resource Control
RR	Round Robin
RTT	Round-trip time
SDN	Software Defined Networks
SIM	Subscriber Identity Module
SLA	Service Level Agreement
SMF	Session Management Function
SNMP	Simple Network Management Protocol
STIN	Satellite-Terrestrial Integrated Networks
TCP	Transmission Control Protocol
TCO	Total Cost of Ownership
TLS	Transport Layer Security
TM/TC	Telemetry/Telecommand
TNGF	Trusted Non-3GPP Gateway Function
TN	Terrestrial Network
UDM	Unified Data Management
UDP	User Datagram Protocol
UDR	Unified Data Repository
UE	User Equipment

© 2023-2025 5G-STARDUST

UPF	User Plane Function
URSP	UE route selection policy rules
UTs	User Terminals
QoS	Quality of Service

© 2023-2025 5G-STARDUST

1 INTRODUCTION

This deliverable reports the final outcomes of two tasks of the 5G-STARDUST-project: Task 5.1 – "Multi Connectivity Solutions" and Task 5.3 – "Software Defined Network Control". As the other task from WP5 both are dealing with network and transport layer aspects. The work presented is building on the architecture and baseline solutions elaborated in WP3 in Deliverable D3.1 and D3.2 [1], [2]. The architecture consists of a terrestrial 5G system and a 5G-Non-Terrestrial Network (NTN) on a Low Earth Orbit (LEO) constellation which are either offering connectivity in transparent or regenerative mode, i.e. hosting a gNB on-ground or on-board satellites, respectively. Both are complemented by a Geostationary Earth orbit (GEO) satellite. User Equipment's (UEs) can connect either directly to terrestrial or space-based base stations or indirectly using integrated access backhauling (IAB).

For multi-connectivity Access Traffic Steering, Splitting and Switching (ATSSS) as specified by 3GPP is the baseline architecture. Generally, multi-connectivity means the usage of at least two parallel connections to increase the Quality of Service (QoS). Usually, the main aim is either to increase throughput or resilience. One link can be used as back-up in case the connection with the main access is lost, also called switching, or they can be used complementary. In this case, traffic can either be split over the two links (usually based on a defined threshold) or, depending on the QoS flow, steered towards a specific link full filling the QoS demand. The benefits and selection of the setup depend on the characteristics of the available links. For instance, in a typical GEO NTN-TN parallel topology the NTN link is considered as back-up of the TN due to its higher availability and worse latency. If the links are similar in throughput, latency and jitter, they can be used in parallel increasing the throughput, while if one link has higher latency, arriving packets must be reordered slowing down the overall connection.

In the second part of this deliverable, we use the Software Defined Networks (SDN) concept to develop a mechanism enabling transport level routing solution for mega-constellations to significantly cut routing convergence time and reduce processing demands on space nodes, by providing centralized computed routing information to the space nodes prior to its usage and by enabling an automatic switching of the routing tables without inter-node communication.

The document is organised as follows:

- Section 2 discusses the architectural options for multi-connectivity following the baseline scenarios of the project. Furthermore, this section provides an exhaustive presentation of transport protocol options.
- Section 3 presents technologies for the optimization of multi-connectivity.
- In Section 4 Software Defined Networking-based routing and conditional hand over mechanism are presented.
- Section 5 concludes the document and provides an outlook.

© 2023-2025 5G-STARDUST

2 MULTI-CONNECTIVITY ARCHITECTURES

Two baseline use cases for 5G-STARDUST's proof of concept have been selected [3], one of them, connecting UEs on-board of an airplane (or ship), is used in the following to discuss the options for multi-connectivity (MC). Nevertheless, the presented architectural options and investigations can be generalized and apply also to other use cases.

In the above-mentioned use case, MC is established by connecting to TN and NTN simultaneously. There are two radio frequency bands for 5G NR namely, FR1 and FR2. FR1 ranges from 410 MHz to 7125 MHz [4], while FR2 offers a wider range from 24.25 GHz to 71 GHz [5]. FR1 and FR2 have frequencies that can be used by both TN and NTN, allowing devices to connect to either network [6]. On connecting to both frequencies UE can use FR1 for TN connectivity and FR2 for NTN or vice versa. This will help in reducing interference making the MC solution more efficient and the system more robust.

In Deliverable D3.1 the 5G-STARDUST system architecture was introduced. UEs can connect to TN or NTN for ubiquitous coverage. NTNs are either transparent in line with 3GPP Release 17 and 18 standard or regenerative following work in progress of Release 19. Furthermore, UEs can connect directly to the terrestrial and non-terrestrial networks accessing the gNBs or indirectly, by an gNB which is connected via IAB. Several options for MC on different layers of the communication stack have been discussed in Deliverable D3.2 [2]. As baseline ATSSS was selected, which is using Multi-Path TCP (MPTCP) or MPQUIC to connect to a non-3GPP access technology and to a 3GPP access with a common 5G Core (5GC). It is basically an adaptation of MPTCP and MPQUIC, compliant with the 5GC which allows to terminate MC at the PDU Session Anchor (PSA) User Plane Function (UPF), i.e. the user serving UPF. The specification includes also an ATSSS lower layer part for Ethernet traffic. Using MPTCP and MPQUIC to directly connect to data network (DN) on top of the 5G system can be considered as baseline for multi-core setups.

For a better overview the available links are shown again in Figure 2-1 for direct access and in Figure 2-2 for indirect access, respectively. UEs or IAB[1] nodes can connect to a GEO satellite, potentially multiple LEO satellites, potentially multiple terrestrial links, or eventually only a specific sub-set of these. The UE illustrated here can also be a group of locally close UEs, e.g. on board an airplane or train. An example following the airplane scenarios is: a group of users on board of an airplane connect either directly or indirectly using an IAB-node on board the plane. Close to ground, terrestrial links can be used in parallel to NTN links. While after take-off at high altitudes and above the oceans, only NTNs are available. Each of these links has different characteristics in terms of QoS parameters such as throughput, latency, or jitter that need to be considered and might affect the performance.

[1] IAB nodes are taken as reference here mostly for acting as relay nodes. However, the full protocol architecture of IAB nodes is not considered in this report, not will it be in the forthcoming phases of the 5G-STARDUST project. As such, it is about IAB-like nodes. Under this assumption, the attributes 'IAB' and 'IAB-like' will be used interchangeably throughout the entire document.

© 2023-2025 5G-STARDUST

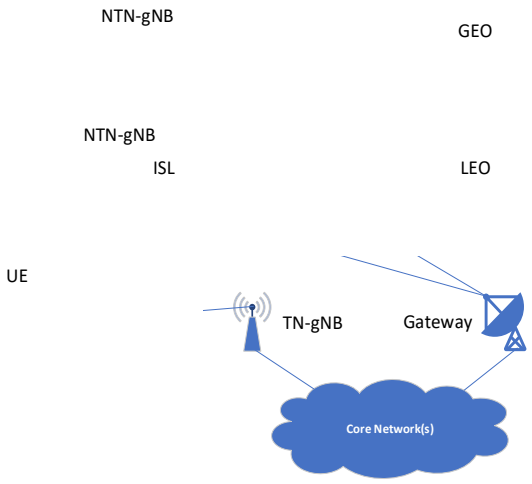

Figure 2-1: MC scenario direct access [2]

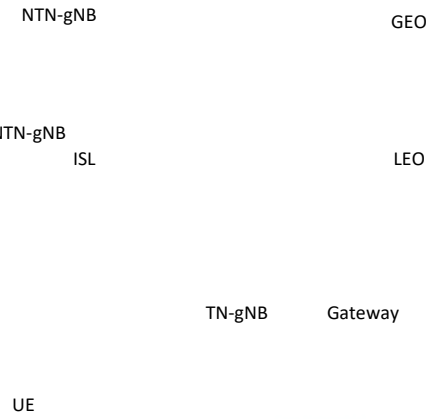

Figure 2-2: MC scenario indirect access [2]

With direct connectivity the MC can be started and controlled by the UE or even the user application, while with indirect connectivity, the UE connects to a gNB that is backhauled by multiple links (TN and NTN) which is not possible so far by standardized system components. We discuss in the following sections several options and extensions required to enable this beyond the current standard.

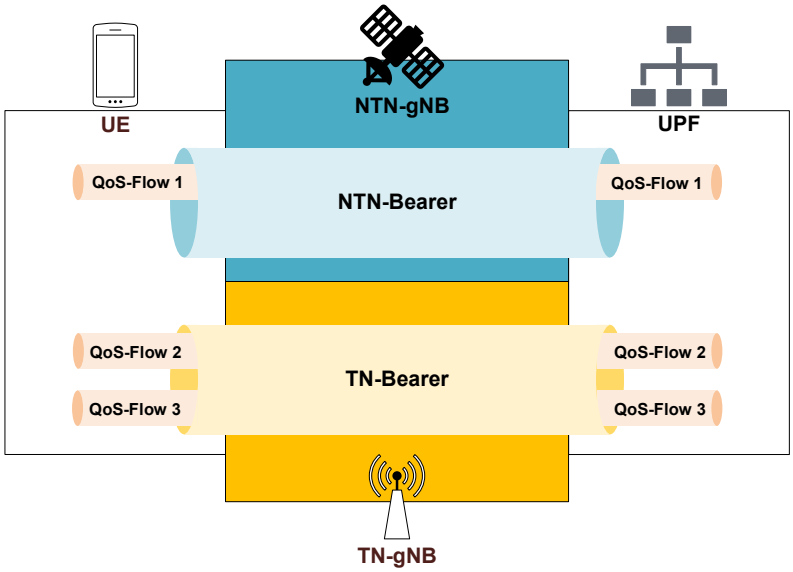

Figure 2-3: MC bearers direct connection

Figure 2-3 shows the bearers for the baseline case. The UE has access to the NTN and TN with a dedicated bearer. The UE can open various QoS-Flows for applications at each bearer and distribute the traffic accordingly using the multi-path (MP) protocol. As mentioned, the traffic can be split, steered or switched. The figure exemplary shows three QoS-Flows but it is not limited to this. The QoS flows reach via the NTN- and TN-gNBs the UPF which terminates the MC and reorders packets, if needed. With information about the network performance, QoS flows could be scheduled according to QoS profiles from applications. Characteristics of NTN and TN bearers might be different in terms of delay, packet loss, or jitter. Additionally, a user

© 2023-2025 5G-STARDUST

might prefer one of the bearers due to resource consumption and costs, e.g., there might be different costs for NTN or TN services or different service level agreements (SLA) with the providers. In principle, QoS flows can be established based on application-level profiles such that flows with low requirements on latency can select a GEO path while those having a more demanding requirements can use LEO or TNs.

Figure 2-4 and Figure 2-5 show two different possibilities for indirect connection from bearer perspective. In Figure 2-4 the first option is shown, the IAB-node transparently forwards the NTN and the TN bearer to the UE. The UE can establish and control QoS-flows as in direct communication case. From implementation perspective the IAB-node must be able to handle and forward different bearers, or simply the IAB functionality is duplicated, i.e., an IAB-node for each connection supported. But this scales with the number of bearers and providers that, e.g., at the airplane use case, might be multiple ones hosted at the NTN or TN system. In the second case presented in Figure 2-5, the IAB-node provides a dedicated IAB bearer to the UE and is backhauled by the NTN and TN bearer. Two options can be considered: (i) the UE handles the MP-connection as in ATSSS, (ii) the IAB nodes handles the MP connection. In first case, UE and IAB node must coordinate the different bearers and how the QoS-flows should be mapped to them at the IAB. In the second this can be optional, since the IAB could establish MC and provide its benefits transparently to the UE.

It must be noted that the NTN/TN bearers have been selected as example to simplify the description. However, the principle can be applied generally for other types of connections and for all link combinations, for instance two TNs or two NTNs (e.g., two LEOs or a LEO and a GEO).

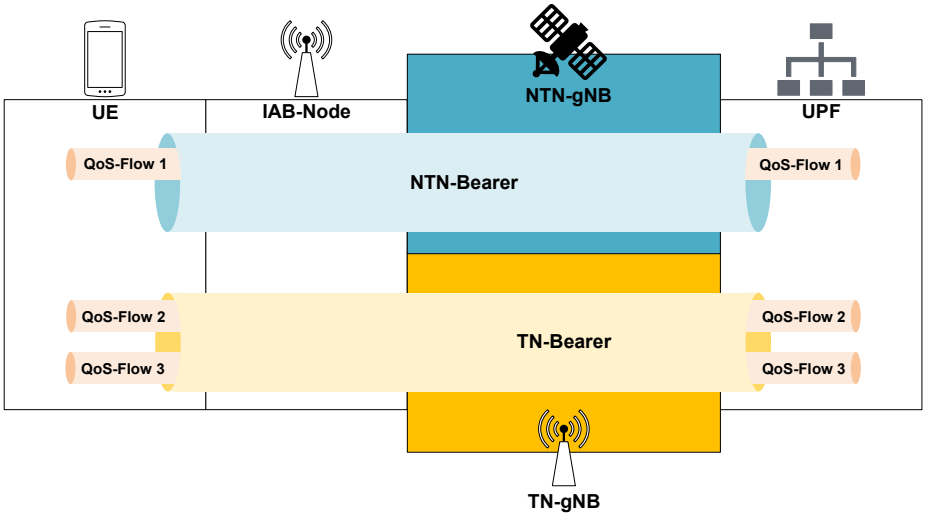

Figure 2-4: MC bearers indirect connection with transparent IAB

© 2023-2025 5G-STARDUST

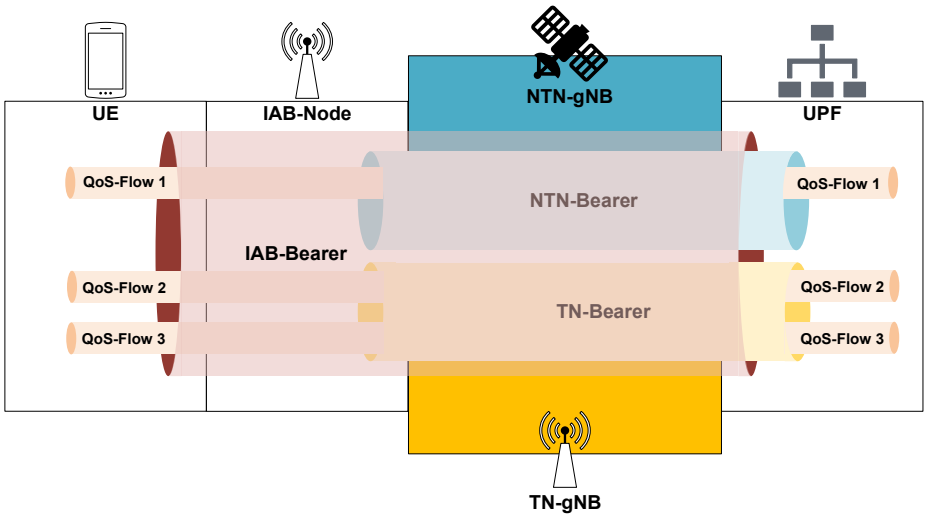

Figure 2-5: MC bearers indirect connection with IAB bearer

In all cases the core must be able to handle the different QoS flows, which are, as mentioned, parallel 5G connections. To be able to control these data bearers and to allocate the appropriate resources a comprehensive 5G control plane must be deployed. The control plane is described in section 2.2 of this deliverable.

As noted in [7], 3GPP shifted from IAB to Wireless Access Backhaul (WAB) nodes. In contrast to an IAB node, a WAB node the full gNB together with the MT for the user side. The WAB relay nodes are aimed at allowing Mobile Edge Computing on-board and to support Vehicle Mounted Relay (VMR) services. From architecture perspective and for higher layer MC an UPF can be introduced afterwards compatible with ATSSS. Therefore, the architectural considerations considered of the previous work for IAB still hold for WAB, but, in TR 38.799 WAB for NTN can only be used for backhauling.

2.1 ACCESS TRAFFIC STEERING, SWITCHING AND SPLITTING

ATSSS is an optional part of the 3GPP standards to connect a non 3GPP technology and a 3GPP technology to the 5GC, described in 3GPP TS 24.193 [8] and TS 23.501 [9]. The Non-3GPP Interworking Function (N3IWF) and the Trusted Non-3GPP Gateway Function (TNGF) are used for this. Since 5G-STARDUST is considering the 5G NTN in parallel to TN, the ATSSS functionality must be extended to allow two or even multiple 5G-bearers connected to one UPF. The functionality of N3IWF or TNGF are not needed in this case.

ATSSS establishes a Multi-Access (MA) PDU from the UE to the UPF. It supports ATSSS-Lower Layer, MPTCP or MPQUIC. MPTCP and MPQUIC steer traffic above TCP/IP and UDP/IP, respectively, while ATSSS-LL is below IP layer supporting Ethernet traffic. The UPF is including a MPTCP or MPQUIC proxy functionality. It shall be noted that MPQUIC was added to the standard in Release 18 and is referring to the current draft version of MPQUIC, since a standardized version by the IETF does not exist at the moment of writing this deliverable. On the user plane, there can be traffic on both connections at the same time, only on one of them, or on none of them keeping the connection open. MA-PDU session are established based on UE route selection policy rules (URSP) specified in 3GPP TS 24.526 [10]. As explained in the following section on the control plane process the setup of the connection depends on the capabilities of the UE, in principle plain MPTCP or MPQUIC are also supported.

© 2023-2025 5G-STARDUST

MPTCP is standardised by the IETF and is an extension to TCP. Basically, it builds on top of it and opens several TCP sub-flows. One design goal was that it can be used without changes to legacy applications [11]. The standards comprise of IETF RFC 6182 presenting the MPTCP architecture, RFC 6356 dealing the congestion control, RFC 6897 with application considerations, and finally RFC 8684 gives the necessary TCP extensions.

Similar to MPTCP, MPQUIC is the multi-path extension to QUIC, currently only available as draft version at IETF in which it is also referred in ATSSS [12]. QUIC is a transport protocol specified by the IETF in May 2021 as RFC 8999 – RFC 9002. It is the specified transport protocol for HTTP/3. Implementations of the core QUIC specs are available and QUIC is already widely deployed to provide web content across the Internet. QUIC is connection oriented and runs on top of UDP, it handles the same functionality that is performed usually by TCP (e.g., congestion control). According to [13], QUIC provides the following features:

- Multiplexing of separate objects using streams identified by a connection ID.
- Mandatory security establishment via TLS1.3 extensions embedded within QUIC, decreasing the initial hand-shake procedure to 1-RTT and 0-RTT.
- Avoidance of Head-of-blocking effect by managing retransmissions.
- Enforcement of congestion control, reordering and error recovery at application level (this aspect allows the possibility of protocol updates independently from operating system through an update of the web clients.
- Definition of different priorities among objects and pushing of objects.
- Connection migration to a new network with new IP address and/or port while the connection remains up and running.

2.1.1 Control Plane Process

To support MA PDU sessions, the AMF, the SMF, and the PCF must be extended to be able to handle the MA-requests. It's then on the AMF to inform the SMF about the two connections available, the SMF sets up the UPF, accordingly, using the N4 interface and a Multi-Access Rule (MAR). To introduce parallel 5G connections the specification must be extended on this regard.

The MA-PDU session establishment is a two-way handshake: the UE is sending the MA-PDU session request, from the AMF it receives the PDU SESSION ESTABLISHMENT ACCEPT message including the ATSSS container information element of the Non-Access Stratum (NAS). The connection and the dedicated user plane resources can then be considered established. ATSSS allows the UE to register in the same Public Land Mobile Network (PLMN) or even via two different PLMNs, consequently it covers already cases in which the NTN-Provider would offer its own PLMN in parallel to the TN-Provider, or in which both share a PLMN. In case both connections use the same PLMN, the MA-PDU session request can be sent via either one of these links, which is implementation specific. If the UE is connected to two PLMNs, it shall send the MA-PDU session request sequentially; the link selected first is implementation specific. In a third, special case, if only one connection is available at the point in time the UE wants to establish the MA session, then it sends the request via the network available and via the second if it becomes available.

For transmission, it is on the UE to decide how the traffic must be distributed, but during the MA PDU session establishment it can receive ATSSS rules from the core that must be applied.

© 2023-2025 5G-STARDUST

In general, the decision can be based on performance measures that can vary based on the UEs capabilities. One option is that the SMF provides the UE with Measurement Assistance Information (MAI) which can be used for default-QoS rules. If the UE is capable, it can also use measures available at MPTCP or MPQUIC layer applying the non-default QoS rules. Alternatively, feedback on the current link status can be provided by a performance measurement function (PMF). A protocol for message exchange between UE and UPF is specified. It should be noted that in TS 24.193 section 8, timer values are specified for PMF measurements at UE and UPF after which the measurement might be cancelled, most of the timers are set to 1s which might lead to unwanted cancellations since especially in GEO higher values can be expected, not on average, but in extremes cases. ATSSS rules and MAI can be updated by the SMF. On the other hand, the link conditions, especially in LEO, can vary a lot due to hand-overs and the change of elevation of the satellites. A frequent update might be necessary to make efficient use of the available links.

In ATSSS, steering refers to selecting for the user traffic, the best service for a flow with linked QoS-type. Switching means a handover to the second link in case of interruptions. Splitting allows to distribute the traffic between both links for load balancing. According to the standard [9], five modes are specified for the MAR and the ATSSS rule:

- Active-standby: one access is defined as main access, the other as active-standby. In case of unavailability, traffic is steered towards stand-by access.
- Smallest delay: the access network with the smallest round-trip time is used.
- Load balancing: the flow is steered across both the 3GPP access and the non-3GPP access, with a given percentage. It supports autonomous or UE-assisted operation.
- Priority based: the UE uses the access with high priority unless it is congested or unavailable, then the traffic is split over both the access networks.
- Redundant: traffic is duplicated on both access networks. Optionally a primary access can be selected in which case the primary must be used for all packets, the secondary may be used to send the duplicate packets.

If the UE changes the default steering parameters to adjust its steering based on its own decision it can use:

- Autonomous load-balance indicator: the UE applies its own distribution in percentages for the load-balancing case to maximize bandwidth.
- UE-assistance indicator: the UE applies its own distribution in percentages for the load-balancing, considering its internal states (e.g., battery state).

2.1.2 Controlling Multi-Connectivity by Artificial Intelligence and Machine Learning

The ATSSS baseline allows for introducing the use of artificial intelligence (AI) and Machine Learning (ML) for traffic control in two options: first is to make use of the fact that the UEs control the uplink traffic while it is on the PSA UPF, to control the downlink. An AI/ML controller can use the congestion state available at these nodes and traffic predictions to control the end-to-end flows from and to the UE to satisfy the QoS needs. Second option is to make use of an AI/ML controller to define and update MAR and ATSSS rules directly from the core. This provides a possibility to centrally control UE behaviour and traffic distribution within the network. Here the optimization goal is on general network traffic distribution since multiple UE

© 2023-2025 5G-STARDUST

connections can be considered. At this point it shall be noted that the use of AI/ML also for MC is investigated in Task 5.2 and preliminary results are presented in D5.3 [14]. Within this task, findings from these investigations are considering in the MC testbed presented in section 3.1.1. This testbed is following the first option to optimize single UE traffic distribution.

It should be noted that another option to make use of AI/ML for optimization is to place it in the UPF functionality that terminates the MP-traffic, e.g., this could be a satellite of the constellation which has at the moment a good link to the ground network, or a UPF located in the ground network.

2.1.3 Architecture Adaptations and Protocol Stacks

As mentioned already, in D3.2 [2] high-level options for the architecture have been presented and discussed preliminary based on the 5G-STARDUST architecture and use cases. Taking ATSSS as reference architecture, the technological principle is to introduce a MC-layer into the protocol stack which is enabling the benefits of MP. In the following, we separated the layer into MP-High representing MPTCP or MPQUIC and MP-Low since both use one or more TCP or QUIC connections, respectively. At 3GPP, MC functionality is not considered so far for indirect connections, i.e., backhauling a base station from different paths. Nevertheless, the presented architectural options and topologies provide different possibilities to place the MC-layer where we expect different benefits. In the Annex we present the complete protocol stacks for each of these topologies and, for a better overview again, the high-level architectures and the single connectivity cases. The stacks show only the user plane part since (as presented in the previous section) the control plane stacks stay unchanged; registrations are done sequentially. The topologies can be classified as direct and indirect connection (i.e., IAB), for both we considered the UPF either on-ground or on-board the satellite. For indirect case, four cases can be distinguished, the MC-layer can be at UE, at the IAB-node, or at an additional UPF which is included specifically for this layer.

- Direct Connection
    - With MC-layer at UE: this is the baseline ATSSS having the MPTCP/MPQUIC at UE and UPF (see Annex).
    - With MC-layer at UE, MP-termination at UPF in space: 5G-STARDUST is looking into regenerative and transparent payload and as discussed in D3.2, if the MC-layer is implemented as in reference ATSSS UE-UPF PSA connection there is no effect of this option. On the other hand, if not only the gNB but also a UPF is provided as payload of a satellite, MP can be terminated at this layer, providing single path connectivity from there on. This only makes sense if multiple paths are opened towards the NTN, connecting two or more LEOs. At the same time the feeder link can be off-loaded. This approach is deviating from the standard, since in principle relaying MC traffic is specified but MP traffic is terminated at the PSA-UPF, so at the end-to-end path of the 3GPP system. The corresponding protocol stack and topology can be found in the Appendix.
- Indirect Connection
    - With MC-layer at UE: the baseline ATSSS case considering IAB, combining both options. This case was not introduced on D3.2, for completeness we present it here. The MC-layer is implemented at UE and core side as in ATSSS case; however, the difference is that the UE is connected to one IAB-node (as introduced in Figure 2-4). As seen in Figure 5-10 in the Annex, the UE and UPF PSA can directly communicate on top of the available infrastructure. But either the infrastructure is duplicated (two

© 2023-2025 5G-STARDUST

IAB-nodes), the IAB-node is extended to transparently forward multiple gNB connections (e.g., the TN and NTN-gNB), or the IAB-node is extended to communicate with the UE to configure the MP links. This is needed to efficiently organize the communication for which the UE should be aware that there are multi-paths it can make use of, and the IAB node must be aware of how to distribute the traffic among its backhaul links.

- With MC-layer at UE and MP-termination at UPF in space: this case is similar to 1b) with the difference of the indirect connection having an IAB node in-between. From protocol stack, it can be seen as a merge of 1b) and 2a) as presented in Annex 0. As for both cases the IAB-node must be extended to enable the MP connection and for an efficient use of it to allow coordination between the UE or the core side with the IAB-node to control the traffic flows and it must be allowed that the MP traffic is terminated at a relaying UPF.

- With MC-layer at IAB-Node: enabling the bearer split as presented in Figure 2-5. This is a special case moving the MC-layer from UE-UPF to IAB level. This approach has been selected for the 5G-STARDUST proof of concept (PoC) initially introduced in D6.1 [15]. We are describing the stack and some related considerations in the following in more detail.

- With MC-layer at Forwarding UPF function: in this topology, an UPF is introduced right after the IAB-node. From use case perspective, this would mean that UEs, the IAB-Node and the UPF are hosted all on-board of an airplane or a ship. This is making the satellite network being part of the Transport Network, following classical backhauling setups. For the SatCom links, 5G-NR or any other access technology such as DVB-SX could be used. The actual UE 5G communication runs on top of it. Consequently, this topology allows the backhauling of indirect communication but is not in line with the 5G-STARDUST architecture having the NTN as part of the access network. For the sake of completeness, we are still presenting it here.

As mentioned, in the Appendix the protocol stacks for all considered Multi-Connectivity (MC) topologies are presented that have been derived from the 5G-STARDUST baseline use cases and architecture, D3.1 and D3.2. The illustration is following the colour code presented in Figure 2-6.

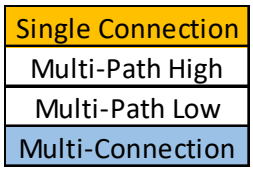

Figure 2-6: Protocol stacks, color-coding

It means that orange is single connections which is split, switched or steered by the MP-High (MPH) and MP-Low (MPL) layer coloured in white. The blue connection is the MC which means that these parts of the protocol stacks must be available at least twice, e.g., on TN and NTN path. The topologies show always an NTN and a TN path, but the stacks generally apply to other combinations of paths (e.g., NTN-NTN). The general illustration approach is that if moved down in layer the upper layer is encapsulated within the lower layer. Finally, at physical (PHY) layer the actual transmission is done and the stack is decapsulated up to a certain point. To further forward traffic it needs to be encapsulated again for the next hop which might follow another stack. Communication links are illustrated as lines between the protocol blocks. A layer can communicate with its corresponding part on sender or receiver side; it must be decapsulated properly to this point. Figure 2-7 shows exemplary the stack for the baseline

© 2023-2025 5G-STARDUST

case 1a). UE connects to the gNB via the 5G New Radio (NR) stack either via TN or NTN. The blue color-coding indicates that there can be two gNBs. The gNB is connected on the transport side to the UPF-PSA which connects to the data network. The UE opens the MA-PDU session and opens the MPH connections which are either MPTCP or MPQUIC. From there on, its two TCP/QUIC connections building on the 5G-NR stacks below it.

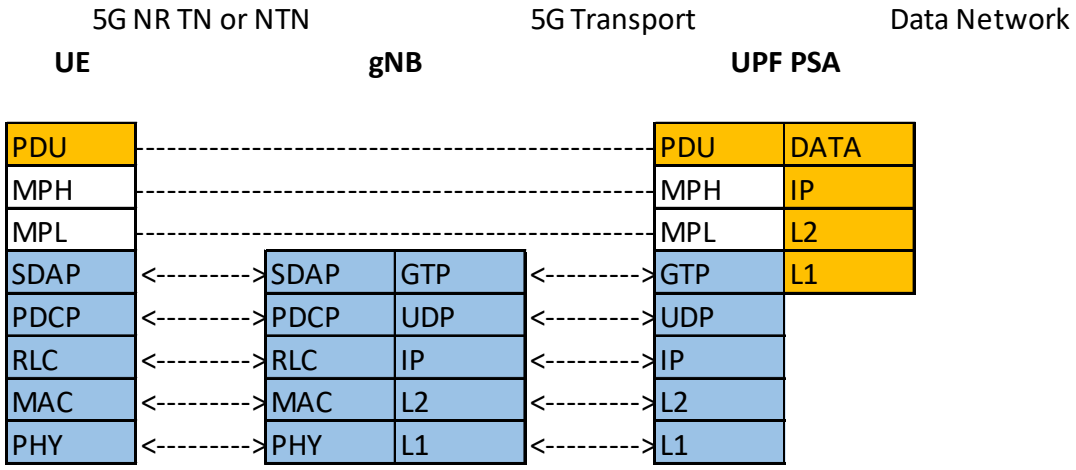

Figure 2-7: Protocol stack, direct connection with MC-layer at UE

In this example the MC-layer is at the UE. In Figure 2-8 an example for the second option, having the MC-layer directly at the IAB-node is presented. This is one possible option identified from the single connection protocol stack (cf. Annex 0, Figure 5-3) in which a UDP connection is established between the Distributed Unit (DU) of the IAB-Node and the Central Unit (CU) of the IAB-Donor. This connection is established below the GTP connection in the stack. This is a first intuitive solution aiming to minimize changes of the single connection stack to simplify potential implementation updates and keeping most part of the original stack unaffected. However, it must be pointed out that this approach is turning a connectionless link into a connection-oriented one, provided by features of TCP and QUIC which might introduce unwanted latencies due to re-transmissions and flow-control. Other options are shown in Annex 0. A way to circumvent this is to introduce extensions such as specified in IETF RFC 9221 [16] which enables unreliable data-flows of QUIC. However, with MPQUIC not specified is not clear if such extension could be included.

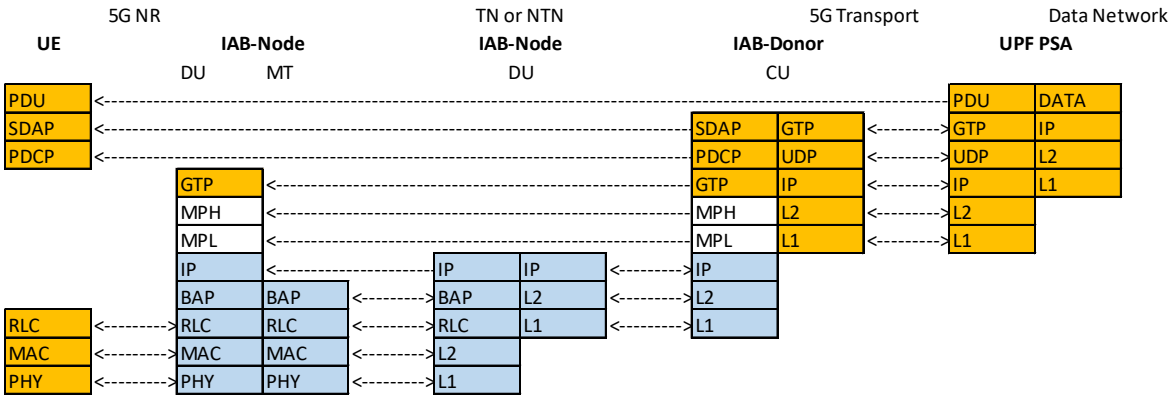

Figure 2-8: Protocol stack, Indirect Connection with MC-layer at IAB-node

The presented stack is the direct result from taking the standard IAB stack. From stack side there is no difference if the IAB-Donor is considered to be at the NTN gateway following e.g.,

© 2023-2025 5G-STARDUST

a transparent approach or if it is part of the satellite payload in a regenerative approach. However, for actual implementation having an IAB-Donor on-board a satellite opening multi-paths via NTN and TN might introduce additional challenges, it might be preferable for NTN-NTN paths.

It is important to note that if the MC-layer is introduced in this way, at the IAB-layer, it cannot be terminated at UPF, neither in space nor on ground, since the encapsulation order will be broken. This can be seen in Figure 2-8, there is no option of introducing a MC-layer that is able to communicate with the UPF, all protocols communicate with the IAB-donor. Layers above the GTP of the IAB-node's DU are from the UE preventing an introduction. Consequently, if a MC-layer needs to be terminated at a UPF it must be initiated by the UE or another UPF as shown in the other topologies. MC-layer that has been introduced beyond the standard at IAB must be terminated at this level.

A similar approach has been followed in the 5G-STARDUST PoC design presented in D6.1 [15] and presented in Appendix in Figure 5-15. It is on the IAB-node to establish and control the MP-connection. Also, optionally communication between the UE and the IAB node would be beneficial to optimize the use of both links. QoS parameters of traffic flows could be exchanged, but also completely switching between two links makes the end-to-end connection more robust. The stack was presented in D6.1 and is shown in the Annex in Figure 5-16. The stack includes two UPFs which are introduced due to a general problem which can be observed on introducing a MC-layer at an indirect node. IAB is basically an introduction of a gNB before a gNB, i.e., chaining base stations. If the MC-layer is introduced at the first, UE-serving gNB, it breaks the GTP connections making it necessary to introduce a second GTP layer at the UPF to ensure proper decapsulation (The Problem is illustrated in Figure 5-17 in the Appendix). This additional GTP block is breaking the standard 3GPP UPF stack, but it can be circumvented by introducing another UPF before the PSA-UPF. From use case side, this UPF could be the one from an NTN/TN provider while the UPF PSA is the UPF contracting the UE.

The protocol stack shown in Figure 2-8 introduces MPQUIC or MPTCP below GTP, i.e., encapsulating the GTP traffic. It is important to point out that MPQUIC is by default encrypting the traffic and headers making it impossible to access the GTP information before decapsulating at the MP-termination. This means that if GTP information is needed along the path, e.g., for UPF chaining, it is better to introduce the MPH and MPL protocols before GTP. In this way, each path would have its own GTP tunnel.

2.2 CONTROL PLANE FOR MULTI-CONNECTIVITY

As mentioned, Multi-access in systems up to 5G have been limited to support a 3GPP and a non-3GPP connection, multi-5G connectivity being implemented in the RAN through solutions like 5G Coordinated Multi-Point (CoMP) or secondary RAN. However, with the advent of multiple frequency ranges and 5G NTN-TN convergence, multi-access across two independent accesses of the same 3GPP technology type has become a new high capacity and reliable communication opportunity where RAN solutions like CoMP cannot be realized as the two RANs may pertain to different subnetworks requiring signalling and data path aggregation at core network level. To answer to this opportunity, we present a new connectivity concept and 5G architectural advancements required to achieve this type of multi-access, with focus on TN-NTN, although the principles and architecture would apply to other scenarios briefly mentioned as well. Following a background analysis of the multi-connectivity concept in 5G system, we propose the necessary architectural additions to support multi-access across two independent 5G connections. We assess the feasibility of the solution by considering the additional functionality to be added. This study demonstrates that despite its complexity, such

© 2023-2025 5G-STARDUST

a solution is feasible for 5G deployments and can be easily implemented in the beyond 5G and the foreseen 6G architecture.

Enabling multi-connectivity on devices represents an effective way to improve data rates and improve communication reliability. Whether having two connections to base stations of the same 3GPP technology or by using a secondary more cost-effective but less reliable non-3GPP access, multi-connectivity enables balanced network resource management. Through multiple connections, user traffic can be distributed across various paths, optimizing resource utilization based on applications' requirements. This dual connectivity ensures that even if the secondary connection is less reliable, it can handle less critical traffic, reserving the primary, more robust connection for essential data.

Seizing these advantages, several options for multi-connectivity on various layers have been summarized in [17], while 3GPP has standardized three mechanisms for multi-connectivity. The first is deeply embedded in the 3GPP RAN systems, enabling connections to multiple base stations of the same type, where the base stations collaborate to schedule the device's data traffic – 5G Coordinated Multi-point (CoMP). Second, dual connectivity/multi-connectivity is defined at 3GPP bearer level ([18]) with one gNB used as master the other as secondary solution. The master gNB is the entity communicating with the core handling the control plane and aggregating the data bearers, looking like a single gNB towards the core network. Last, 3GPP has also standardized a multi-radio system that enables connections across a 3GPP access network and a non-3GPP one, balancing data traffic between a high-reliability link and a cost-effective one.

However, with the adoption of multiple frequency ranges, the potential use of 5G in unlicensed or privately allocated spectrum, the losing of network reliability requirements for specific frequencies and the integration of the 5G NTN networks, the same 5G radio technology is deployed in systems with very different communication characteristics and requirements. As a result, multi-connectivity using two heterogeneous and independent 5G access networks should be also considered. These systems being separately administrated, will not enable direct interfacing between the RANs, making direct RAN collaboration like CoMP, secondary RAN or the new 6G user-centric architecture ([19]) unfeasible. Additionally, coordination may be impractical due to significantly different RAN coverage areas, ranging from square meters for high-frequency radio to hundreds of square kilometres with frequent handover for NTN.

To enable multi-3GPP connectivity without deep RAN coordination, there is a need to extend the standard architecture. We address this challenge by introducing a new concept for multi-3GPP connectivity and analysing the architecture to provide specific enhancement details. Since interoperability is not feasible at the RAN level, our proposed multi-connectivity relies on the device and core network support. Our proposed solution is based on the core network Access Traffic Steering, Switching and Splitting (ATSSS) feature ([20]), further adapted for the specific target access networks. Although this level of convergence presumes independent RAN operations, it significantly minimizes the interactions between different administrative domains, making it effective and more likely to be deployed in the beyond 5G and 6G networks.

To evaluate the feasibility of the proposed solution, we assess the additional features required as software additions needed to be developed on top of the Fraunhofer FOKUS Open5GCore toolkit ([21]), a reference software implementation of the 3GPP 5G core network standards able to handle both 3GPP and non-3GPP connectivity. This assessment provides insight into the necessary additions and implications of adopting such a solution. Based on this evaluation, we introduce a set of considerations for the foreseen 6G network architecture, enabling an overall architectural and signalling simplification.

© 2023-2025 5G-STARDUST

2.2.1 Use Cases and Requirements

The multi-3GPP connectivity addresses the use cases that cannot be covered by centralized RAN solutions or deep RAN coordination such as CoMP. These scenarios involve multiple 3GPP RAN deployments with heterogeneous characteristics, covering the same device area, able to provide complementary communication services to user devices. Below, we present an initial list of such use cases, highlighting the importance of this feature.

1) TN-NTN Interoperability: to have the most gains from 5G NTN deployments, a seamless handover mechanism to terrestrial networks is required. As 5G NTN is highly dependent on the line of sight which can be suddenly interrupted and the handover procedure through the core network incurs significant delays due to the NTN, maintaining dual connectivity is the only effective solution. This approach ensures no interruptions or significant jitter during handovers by splitting the user traffic based on link availability.

2) Public-Private 5G Interoperability: like the NTN use case, private 5G deployments in enterprises, hospitals, malls, or stadiums may offer additional 5G connectivity to devices complementing the public network. However, this connectivity may be limited in capacity and reliability. Therefore, a multi-connectivity solution for these multi-administrated domains should be considered to enhance device available bandwidth and overall service reliability. For example, this feature would enable private 5G networks not to deploy emergency services, relying on the public network one, reducing their Total Cost of Ownership (TCO).

3) Sub-Networks: often referred to as “network of networks”, the dividing of an end-to-end network into smaller, logically separated segments can enhance the network security by isolating administrative domains and limiting the attack spreading. Sub-networks can have their own QoS policies and access control specific to their administrative domains, optimizing local operations. In the sub-network’s environment, the devices can improve their connectivity by connecting to multiple co-located 3GPP-based sub-networks. In this situation, only a high-level coordination of the multi-connectivity would be possible.

4) Unlicensed/low reliability 5G networks: the network operators may deploy 5G networks in either unlicensed or in spectrum where only low reliability SLAs would be required, to complement the existing high reliability network with additional capacity. This approach is similar to the current 3GPP-non-3GPP multi-connectivity use case but involves two 3GPP access networks. This enables operators to offload a large amount of the user traffic within their own network, drastically reducing the TCO by deploying more cost-effective networks. Devices maintain their high reliability services over the existing 5G network while connecting to the less reliable 5G network in parallel, resulting in multi-3GPP connectivity.

To realize these use cases, the network should be able to coordinate the user identity, its access control and the resource allocation across two parallel 3GPP connections. Additionally, the system should support separately coordinated handovers within each of the 3GPP connections and handle situations where one of the connections is unavailable. Furthermore, dynamic user traffic splitting depending on the momentary established sessions and on the link conditions should be considered to optimize communication across the two links.

2.2.2 Standardisation Background

3GPP CoMP enhances transmission and reception proactively managing the interference among the users increasing the spectral efficiency and the throughput especially at the cell edges. It is based on the coordination of the different base stations through direct exchanges using a direct backhaul interface between base stations through which Channel State Information (CSI) is exchanged [22] enabling joint transmissions, coordinated scheduling and coordinated beamforming.

CoMP demands high-capacity, low-latency backhaul links between the base stations and increased synchronization, making it impractical for the scenarios proposed.

© 2023-2025 5G-STARDUST

Dual connectivity is a simpler method than CoMP, the latter requires an interface between the two base stations. It splits user traffic into two independent RAN connections that are transparent to the core network. In this setup, the master RAN communicates with the core network, while the secondary RAN only handles the second data path connection, merged through the interface between the base stations at the master RAN.

To support multi-connectivity between 3GPP and non-3GPP accesses, a third mechanism, named ATSSS was standardized by 3GPP ([20]), illustrated in Figure 2-9. Non-3GPP technologies are converging in the core network through a Trusted Non-3GPP Gateway Function (TNGF) or a Non-3GPP Inter-Working Function (N3IWF), both adapting the specific signalling to the 3GPP stack. Based on the type of access, the Access and Mobility Function (AMF) can determine if the connection of the device is 3GPP or non-3GPP and to permit both of them to be separately established and separately executing horizontal handovers when needed. The Session Management Function (SMF) is in charge of establishing data bearers over the two connections while the Policy Control Function (PCF) is selecting the appropriate policies to be enforced for the specific user. All operations are executed using the subscription profile in the User Data Management (UDM), the same network function where also information on the current connectivity of the User Equipment (UE) is maintained.

To be able to split the user traffic between the two the UE and the User Plane Function receive a set of ATSSS policies respectively Multi-Access Rule (MAR) transmitted from the PCF. Based on the momentary status of the two connections, the user traffic is split either at the lower layers: ATSSS-LL with an explicit split based on the data bearers or at the Multipath TCP (MPTCP) [23] or Multipath Quick UDP Internet Connections (MPQUIC) [24] where the user traffic is split at the higher levels.

The user traffic can use both connections at the same time, one connection or none while keeping them both open. The split between the two links is executed based on the link performance measurements where the SMF could signal the UE with additional Measurement Assistance Information (MAI), or the UE could interact directly at the data path level with the Performance Measurement Function (PMF) in the UPF. Specifically, for the ATSSS, the non-3GPP link is considered less reliable and more cost-effective to carry the user traffic.

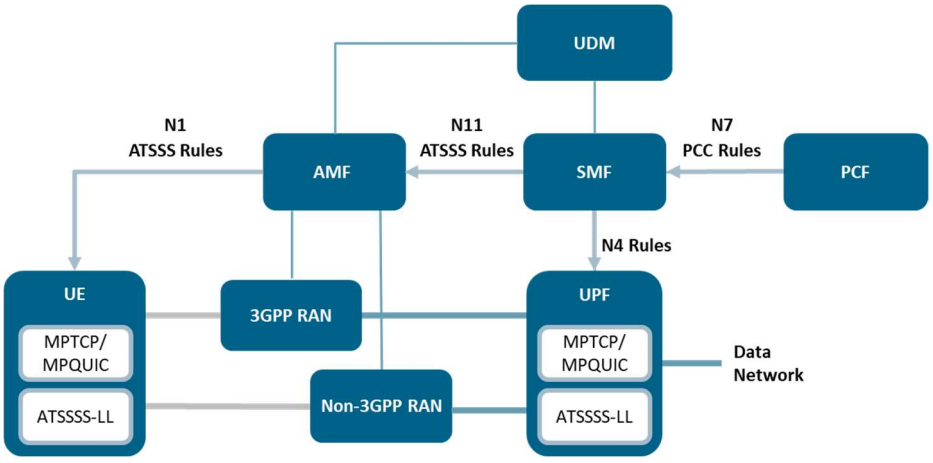

Figure 2-9 : ATSSS Architecture

© 2023-2025 5G-STARDUST

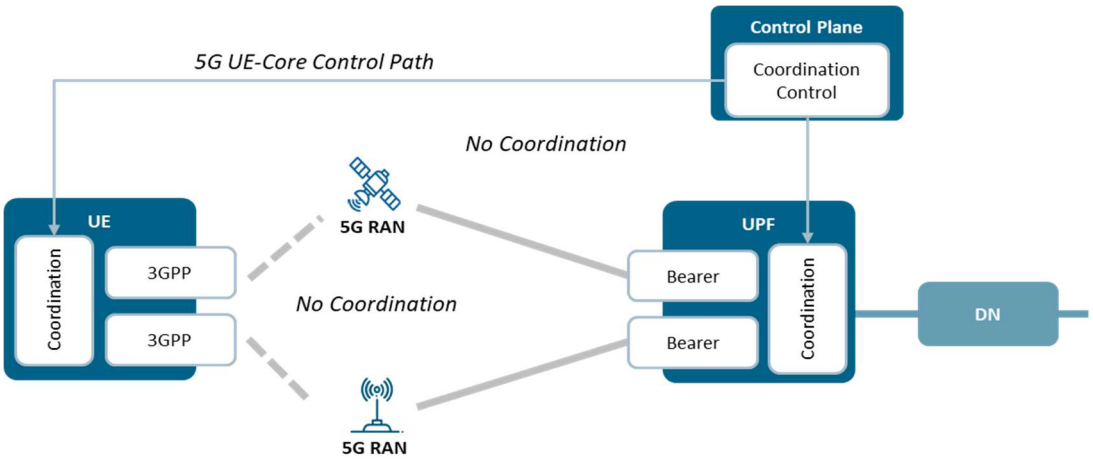

Figure 2-10: Multi-3GPP Connectivity Concept

2.2.3 Multi-3GPP Connectivity Concept

The multi-3GPP connectivity concept, as the name says, presumes that the User Equipment (UE) will be able to connect to two 3GPP accesses at the same time and establish bearers across them. Although the implementation seems straightforward, as illustrated in Figure 2-10, it is difficult due to the usage of the same identity across both links having the same subscription profile.

Considering that there is no coordination between the two RAN networks, the UE will have to register to both networks consecutively and gain access. Similarly, the sessions over the two links must be established independently like the ATSSS.

In the current 3GPP situation, the initiation of a new connection over a 5G RAN will result in taking over the connectivity of the previous connection, executing a hard handover. For this not to happen in the core network, there should be a second differentiator of the established connection included in all the messages from the registration on.

Using the second differentiator, the core network is able to signal to the UE policies how to treat each of the access networks, similar to the current ATSSS policies, however not split on technology.

However, this differentiator does not have a direct impact on the UPF where the information received is not about access networks but about data bearers. As the bearers are access independent, we assume that the same functionality as in the case of ATSSS will be used to signal the policies.

A critical element for the functioning of the multi-3GPP is the possibility to split and to gather the user traffic of the two links. This operation should be executed in the UPF where the two bearers will be associated with a single IP address towards the data network as well as in the UE. Here, the splitting of the user traffic is highly dependent on how many 3GPP stacks are implemented i.e., if the UE includes 2 separate UEs, one for each link, acting as single devices and an overlay split and switch layer.

2.2.4 Multi-3GPP Architecture

2.2.4.1 Required Architecture Extensions

To support a multi-3GPP connectivity it is essential that a new connection will not replace an existing connection as in the case of a handover. Specifically, a differentiator is required between the two connections to be able to split the user traffic, differentiator which must be

© 2023-2025 5G-STARDUST

known by the UE, AMF, SMF and PCF to be able to appropriately allocate the bearers according to the UE policies.

The same connection discriminator should be used for the ATSSS level to be able to define the appropriate user traffic split policies. However, the actual ATSSS functionality does not require other modifications and the policies distributed remains the same, the only open question being which connection is the main one and which will act as secondary for the signalling.

It should be noted that timer values specified for PMF measurements might be needed to be updated especially in GEO higher values can be expected in extremes cases. On the other hand, the link conditions, especially in LEO, can vary a lot due to handovers and the change of elevation of the satellites. A frequent update might be necessary to make efficient use of the available links.

Furthermore, a new mechanism for handling the idle mode is required as well as for service activations and de-activations, as the UE may use two, a single connection or none at specific moments of time.

Additionally, one of the connections may be lost quickly or for very short amounts of time, specifically for the satellite link when losing the line of sight to the currently used satellite or in case of terrestrial networks in areas with low number of base stations and using of higher frequencies such as roads through the forests. In these situations, in order to be able not to lose user traffic and to adapt fast to the situation, notifications should be sent from the link management towards the ATSSS in the user device as well as from the RAN towards the UPF to be able to appropriately redirect the user traffic for the very short duration of such interruptions.

2.2.4.2 Architectural Options

While most of the architectural requirements can be solved with an adapted parametrization of the existing ATSSS solution, the critical element in implementing the architectural requirements is the capability of the user device to handle the UE functionality for two parallel 3GPP connections and the coordination between them. As such, there are two major architectural options for implementing the user device: using two different UEs coordinated between them, as illustrated in Figure 2-11 or using a single integrated UE as illustrated in Figure 2-12.

In the case of two UEs, the derivation of the keys and the establishment of the connections remain unmodified. Practically the UE is seen as a dual-SIM device connected to the same network with both of the UEs. In this situation, the problem is not the key derivation ([25]) as

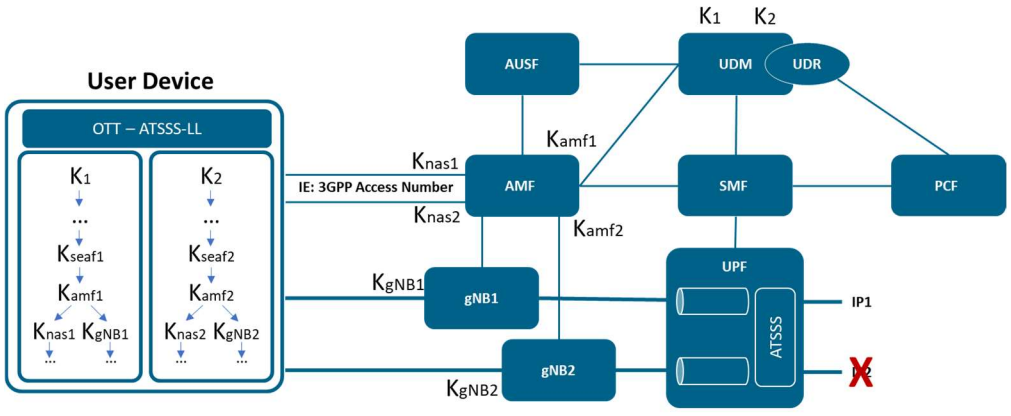

Figure 2-11: Dual UE User Device

© 2023-2025 5G-STARDUST

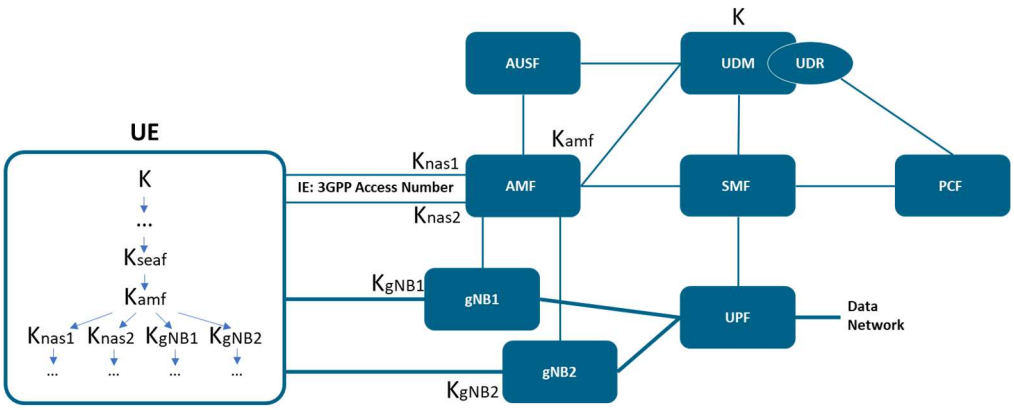

Figure 2-12: Single UE User Device

this is done independently by both of the UEs, but the matching of the user traffic as to enable the splitting across the two UEs. The key derivation follows the standard ([25]) from the main key down to K_{amf} followed by the derivations for control and data plane for each gNB (K_{gNB} and K_{nas}. In this situation, the ATSSS and the MAR policies transmitted to the UE have to consider a primary connection and a secondary one. For such policies the PCF should be aware that the two UEs are co-located into the same device, information which should be introduced within the UDR. Based on this, the PCF may generate specific ATSSS policies and the afferent MAR policies. As the ATSSS are transmitted transparently to the UE, there is no major issue. However, the MAR policies are more problematic as both of the UEs receive their own IP addresses (IP1 and IP2 in Figure 2-11). In this situation a single IP address should be used towards the data network and the user traffic should be split across the bearers of the two UEs. For these, the same MAR policies should be used. The two UEs option is simpler to implement considering the current system as it does not require major changes to the system. However, it will have independent tracking area updates and idle mode management, deemed unnecessary. And especially problematic is the dual SIM requirement for the devices which also implies the modification of the current phones.

In case of a single UE, the key derivation will remain the same down to the K_{AMF} as the authentication and authorization can freely flow between the two 3GPP accesses with having a single index increased based on the operations across both connections. This is in line with the current single UE which executes handovers between different 3GPP access networks. However, as the device is connected to 2 RANs, two distinct K_{gNB} must be generated. In order not to take over the existing second connection, a new Information Element (IE) must be introduced in the NAS messages to enable the network to know this is a second connection and not a takeover of a previous one. If the new IE has more than one bits, virtually the number of 3GPP connections can be increased to more than two through this giving a large advantage to the current ATSSS solution which is strictly bound to two connections. Based on the distinct K_{gNB}s all the RRC and user plane encryption keys can be generated. In this situation, the UE should be able to maintain two K_{gNB} keys and to use them appropriately for the two connections.

However, for a solution to function, it should work also for the current UEs with no modifications. In this case, the network is taking over the functionality of determining which access it is used and create the two bearers as to be ready in case of a handover with a secondary end-to-end bearer. This is a special case of a single UE connectivity where the network is keeping an idle mode connection over the unused 3GPP link. In this situation, the UE cannot differentiate between the different access networks. The differentiator should come from the access itself, for example NTN and TN through this making the AMF aware that the two 3GPP access networks should be run in parallel and not handed over. When the second

© 2023-2025 5G-STARDUST

connection is established, through a handover, the first connection is not destroyed but kept in idle mode. In this situation, all the specific bearer information is maintained, however without any resources reserved over the wireless link. When the UE wants to make a handover to the second network after establishing the RRC connection, the user traffic can be immediately forwarded without requiring additional operations. Similarly, the downlink user traffic will be forwarded to the other link based on notifications from the access. As the notifications will not come from the PMF, the UE not having one, such notification should be transmitted through the handover commands. This way the handovers do not require a specific preparation phase being instead replaced by a fast takeover. A conditional handover mechanism can be employed to reduce the option that a handover command will not be received.

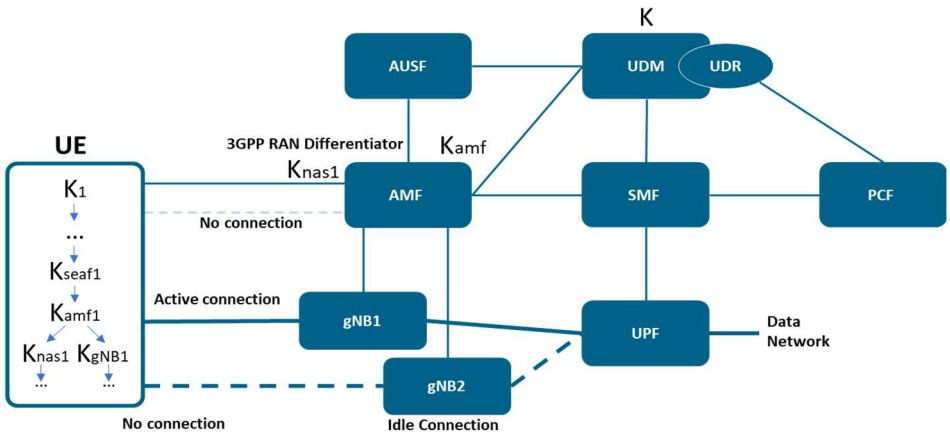

Figure 2-13: COTS UE

2.2.5 Implementation, Feasibility and Complexity Assessment

In this section we assess the implementation feasibility of both the 2 UEs, 1 UE and not modified options and compare how complex it would be to become standard behaviour and what are the implications on the user traffic. First, all the solutions require extensions of the core network, specifically for the policies transmitted to the UE and for the bindings between them. This is a complex feature which mostly impacts the PCF as the policy generator as well as the SMF and UPF for being able to properly treat the user traffic in the network. From this perspective, the easiest solution is to implement the single UE solution and to adapt the MAR policies that are now used for the 3GPP and non-3GPP accesses for the adaptation to the two 3GPP technologies.

However, the single UE solution proves to be highly complex to be implemented in the UE, requiring all the devices to support ATSSS as well as to be able to coordinate the authentication and cryptography across the two accesses, this being a limiting factor for the adoption of such solution.

More realistic would be to have an unmodified UE. In this situation, the communication service will be able to handover quickly between accesses as the second access will maintain an idle mode state for the UE. However, this will still require a handover time between the two accesses even though smaller than the current solution which may induce a longer delay in forwarding the downlink data due to the need to notify the UPF of the change as in the case of handovers. Also, such a solution maintains two states for the same device, and although one is idle, it still consumes more resources than a single connection with a hand over. More evaluation and practical implementation are required to assess the opportunity of such complexity increase.

© 2023-2025 5G-STARDUST

2.3 MULTI-CONNECTIVITY TRANSPORT PROTOCOLS

With the focus of WP5 on networking aspects, the focus for MC is on transport protocol level. Options on other layers considering 3GPP solutions have been presented in D3.2 which we complement in the following Table 2-1 summarizing the existing multipath solutions over IP.

Table 2-1: Multipath solutions over TCP

Protocol	RFC/Draft	Purpose	Benefits	Use Cases	Deployments
Multipath TCP (MPTCP)	RFC 8684	Enables a single TCP connection to use multiple paths.	Higher bandwidth, fault tolerance, seamless handovers.	Mobile networks, data centres, high-availability applications.	Apple iOS (Siri, Wi-Fi Assist), Linux Kernel, 5G networks.
Multipath QUIC (MP-QUIC)	Draft-ietf-quic-multipath	Extends QUIC to support multiple network paths.	Lower latency, improved resilience, better congestion control.	Web services, real-time applications, 5G.	Experimental (Google, Cloudflare testing).
Multipath DCCP (MP-DCCP)	RFC 6773	Extends DCCP (Datagram Congestion Control Protocol) to support multipath transport.	Better support for real-time traffic, improved congestion control.	Video streaming, VoIP, gaming.	Limited adoption, research networks.
SCTP Multihoming & CMT-SCTP	RFC 9260	SCTP supports multiple IP addresses per connection; CMT-SCTP enables concurrent multipath transfer.	Reliability, seamless failover, load balancing.	Telecom, financial systems, industrial networks.	Telecom (3GPP, Diameter protocol), FreeBSD, Linux Kernel.

By far, the solution widely deployed is MPTCP. SCTP deployment is limited but it is the functional reference of modern transport protocols like QUIC. Also, since for ATSSS higher layer solutions MPTCP and MPQUIC are specified, we deeper investigated on them in the following.

2.3.1 MPTCP

MPTCP is specified by a set of IETF RFCs with the main document being RFC 8684 “TCP Extensions for Multipath Operation with Multiple Addresses”. As mentioned in Table 2-1, it is already available as product, e.g. Linux operating system. As the name indicates, it builds on top of TCP connections for multipath transport transparently to the application, therefore, it creates TCP sub-flows each linked to a dedicated TCP socket. Each socket is assigned to a

© 2023-2025 5G-STARDUST

TCP 4-tuple (set of source and destination IPs and ports) that is used to identify the connection and IP handling. The related protocol stack is presented in Figure 2-14. The number of sub-flows is not limited to two, multiple connections are possible.

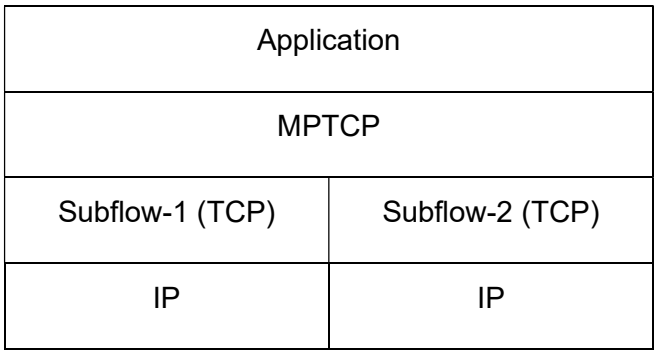

Figure 2-14: MPTCP protocol stack [27]

The MPTCP concept is to open an initial path on the first TCP connection and create the other sub-flows in addition. Consequently, the MPTCP connection setup includes two-times a TCP-handshake. IETF RFC 6182 [28] specifies the informal architecture for MPTCP which consist of a path management and a packet scheduling function, TCP sub-flows and congestion control.

- Path management detects and uses different paths based on multiple IP addresses and signals alternative IP addresses between connected hosts.
- The path scheduling segments the steam to transmit and distributes among the sub-flows. For re-ordering it adds connection-level sequence numbering.
- Sub-flows make use of single-path TCP for data delivery, hence, on sub-flow level congestion windows and packet losses are different for each path.
- Congestion control which coordinates the congestion control across the sub-flows. This function is part of the path scheduling deciding which segments should be sent at which rate on the flows.

Furthermore, as guidelines, IETF RFC 6182 describes the problem of middleboxes on transport layer that are part of today's internet. Middleboxes are for instance: firewalls, network address translation and performance enhancement proxies, all which might even terminate transport layer connections and reopen. These can break the multi-path connection, add TCP extension or modify in any way. The problem can be mapped to the indirect connection scenario where the forwarding node would also be a middlebox for the UE creating the MP connection. The authors ask for any future TCP connection level extension to consider the coexistence with MPTCP.

2.3.2 MPQUIC

MPQUIC is an IETF draft [24] describing the multipath extension for QUIC. This needs to be distinguished from the QUIC connection migration feature that allows to a peer switching, MPQUIC is introducing a path ID to manage underlying QUIC connection IDs and packet numbers per path, enabling the simultaneous use of multiple paths. Also, MPQUIC introduces a path manager and path initialization. In contrast to MPTCP, MPQUIC leaves the address discovery and management to the application. Traffic scheduling is needed but not specified how it is done. MPQUIC is recommending a coupled congestion control algorithm, additionally

© 2023-2025 5G-STARDUST

to the link level ones, and refers to the same algorithms that can be applied to MPTCP. It shall be noticed that MPQUIC is not changing the security mechanisms of the single link QUIC and is keeping the authentication and encryption. The protocol stack of MPQUIC is shown in Figure 2-15. In contrast to MPTCP that builds on sub-flow level on TCP, MPQUIC builds (as QUIC) on UDP. With this MPQUIC is an extension of QUIC and not a layer on top.

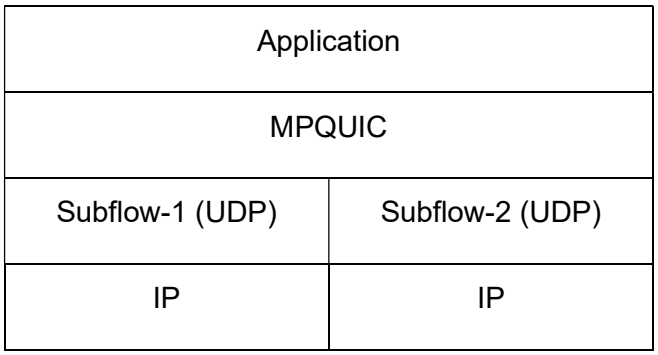

Figure 2-15: MPQUIC protocol stack [29]

Still the main features are inherited from QUIC and as such it differs from (MP)TCP by design it requires e2e authentication e2e encryption, it does not expose CC parameters to the networks and interleave pieces of applications data (compared to TCP, QUIC does not retransmit packet per see, but blog of frames). A more extension of comparison of TCP and QUIC is presented in the following since these aspects, as mentioned, are inherited by the MP variant and are therefore still valid.

2.3.3 Transport Evolution

Most of the Web traffic moved to encryption since 2014. Encryption is gaining the transport header due to the deployment of QUIC by the main internet stakeholder. QUIC represents already 30% of the traffic. It means that there are 2 kinds of end-to-end traffic with regard to the readability of the transport header:

- Part of the traffic will support end-to-end transport observation and adaptation on-the-path (TLS).
- The rest of the traffic will be fully opaque (QUIC).

Authenticated: for QUIC, HTTPS, MPQUIC the applications data exchanges are end-to-end authenticated. By consequence, any data optimization like data compression or transcoding is detected by the end points who break the connection.

Encrypted: in all cases the applications data are end-to-end encrypted. By consequence on-the-path traffic classification is very difficult.

Interleaved: In addition, as transport headers are encrypted, on-the-path priority enforcement inside an end-to-end connection becomes impossible (e.g. Google meet audio, Visio and decks multiplexed in a single QUIC connection).

Regarding the multiplexing, it is important to recall that the flows multiplexed inside a single connection might belong to separate domain names hosted in the same facility (aka Cloudflare).

© 2023-2025 5G-STARDUST

Caching: End-to-end authenticated encryption prevents existing on-the-path shared caches (enterprise proxies, transparent caching…) to work.

Proxies: on-the-path network level optimization functions (similar to PEP) remain applicable on the part of the traffic which has, like TCP, the transport layer in the clear. Nevertheless, there are cases on top of TCP (TCPcrypt [RFC8548]) where the transport layer is in the clear and its integrity is end-to-end protected.

The IETF Masque WG (https://datatracker.ietf.org/wg/masque/about/) is specifying tunneling over QUIC and QUIC HTTP proxy.

3GPP ATSSS Proxy (Application Traffic Steering, Switching, and Splitting): It's based on a proxy which implements MPTCP. There are works to support non TCP traffic like QUIC and Ethernet (ref TS23501, rfc8803, https://tools.ietf.org/html/draft-bonaventure-quic-atsss-overview-00).

2.3.4 TCP vs QUIC transport features

Highlight partially reliable, connection migration

Raw transport and CC features mechanisms are detailed in WG TAPS document.

Table 2-2: transport elementary functions [30]

	Feature name
1	Unicast
2	Mcast/IPv4Bcast
3	Port Mux
4	Connected
5	Data bundling
6	Feature Nego
7	Data priority
8	Data bundling
9	Reliability
10	Ordered deliv
11	Corruption Tol.
12	Flow Control
13	PMTU/PLPMTU
14	Cong Control
15	ECN Support
16	NAT support

Raw transport security features are detailed in WG TAPS document.

Table 2-3: security transport elementary functions [31]

id	**TAPS feature**
1	Forward-secure segment encryption and authentication
2	Private key interface or injection
3	Mutual authentication
4	Endpoint authentication
5	Identity Validation
6	Source Address Validation
7	Pre-Shared Key Export
8	Pre-Shared Key Import
9	Encrypt application data
10	Decrypt application data

© 2023-2025 5G-STARDUST

11	Application-layer feature negotiation
12	Configuration extensions
13	Session caching and management
14	Connection mobility
15	Key Update
16	Key Expiration
17	Session Cache
18	Authentication Delegation
19	Supported Algorithms (Key Exchange, Signatures, and Ciphersuites)
20	Identity and Private Keys
21	Send Handshake Messages
22	Receive Handshake Messages

2.3.5 Ossification and QUIC invariant

TCP evolution was slowed down by the inability of middle boxes to understand and to adapt to TCP extensions. QUIC per design supports an invariant part to enable version negotiation and prevent ossification. The invariant is specified in In RFC 8999 "Version-Independent Properties of QUIC": It defines the immutable characteristics of the QUIC transport protocol that remain consistent across all versions. It ensures that new versions can be deployed without disrupting existing network operations or causing protocol ossification. The invariants outline properties of QUIC that are intended to remain stable as new versions are developed and deployed. By documenting these invariants, the goal is to preserve the ability for QUIC endpoints to negotiate changes to any other aspect of the protocol.

As to hears, QUIC packets are encapsulated within UDP datagrams and come in two forms:

- Long Header: Identified by the most significant bit of the first byte being set to 1. These packets include version information and are typically used during connection establishment.

- Short Header: Identified by the most significant bit of the first byte being set to 0. These packets are more compact and used after the connection is established.

The packet header includes a connection ID, which is an opaque field of arbitrary length used to identify the QUIC connection. The connection ID is chosen by each endpoint using version-specific methods.

QUIC allows for version negotiation to enable protocol evolution. If a server receives a packet with an unsupported version, it may respond with a Version Negotiation packet. This packet has a version field set to 0x00000000 and includes a list of supported versions. Endpoints use this mechanism to agree on a mutually supported version.

While QUIC aims to minimize information exposed to on-path observers, certain invariant properties are visible to facilitate basic network operations. The Version Negotiation packet is not integrity protected, so endpoints must authenticate before acting upon the version information.

The core aspects of QUIC remain consistent while allowing for innovation and evolution in the protocol's design. By adhering to these invariants, the QUIC protocol can adapt to new requirements without compromising interoperability.

© 2023-2025 5G-STARDUST

2.3.6 QUIC proxy, Masque WG specifications

The impacts of QUIC encryption and authentication are deeply developed in RFC 9065, titled "Considerations around Transport Header Confidentiality, Network Operations, and the Evolution of Internet Transport Protocols," explores the implications of encrypting transport protocol headers—a practice gaining traction with protocols like QUIC. While payload encryption has long been standard, extending encryption to transport headers presents both benefits and challenges.

Motivations for Transport Header Encryption

- **Preventing Protocol Ossification**: Middleboxes that inspect and rely on specific transport header fields can hinder the deployment of new protocols or features. Encrypting headers mitigates this by reducing dependency on fixed header structures.
- **Enhancing Privacy**: Encrypting headers protects metadata, such as sequence numbers and flow identifiers, from on-path observers, thereby improving user privacy.
- **Improving Security**: Encrypted headers prevent certain attacks that exploit visible header information.

Impacts on Network Operations

- **Reduced Visibility**: Network operators lose access to header information traditionally used for traffic management, performance monitoring, and security functions like intrusion detection.
- **Challenges in Troubleshooting**: Encrypted headers complicate diagnosing network issues, as operators have limited insight into transport-layer behaviours.
- **Adaptation Requirements**: Operators may need to develop new tools or rely on endpoints to provide necessary diagnostic information.

Design Considerations for Protocol Developers

- **Selective Exposure**: Deciding which header fields, if any, to expose can balance the need for network manageability with privacy concerns.
- **Authentication of Exposed Fields**: Ensuring that any visible header information is authenticated can prevent tampering by malicious entities.
- **Avoiding Ossification**: Even exposed fields should be designed to prevent ossification, possibly through mechanisms like "greasing," which introduces variability to discourage hardcoded assumptions.

Balancing Stakeholder Needs

- **User Privacy vs. Network Manageability**: While encryption enhances privacy, it can impede network operations. A balanced approach requires careful consideration of which information is essential for network functions and which can remain confidential.
- **Collaborative Solutions**: Engagement between protocol designers, network operators, and other stakeholders is crucial to develop standards that address both privacy and operational needs.

© 2023-2025 5G-STARDUST

RFC 9065 does not prescribe specific solutions but rather highlights the trade-offs and considerations involved in encrypting transport headers. It serves as a guide for stakeholders to navigate the evolving landscape of Internet transport protocols, emphasizing the importance of deliberate design choices that account for both privacy and operational requirements.

2.3.7 QUIC proxy using Masque WG specification

MASQUE (Multiplexed Application Substrate over QUIC Encryption) is an IETF Working Group developing methods to tunnel UDP and IP traffic over HTTP/3 using QUIC. This approach enables secure, efficient, multiplexed proxying, improving support for VPNs, real-time apps (like gaming or VoIP), and privacy-enhanced network routing.

"The primary goal of this working group is to develop mechanism(s) that allow configuring and concurrently running multiple proxied stream- and datagram-based flows inside an HTTP connection. MASQUE leverages the HTTP request/response semantics, multiplexes flow over streams, uses a unified congestion controller, encrypts flow metadata, and enables unreliable delivery suitable for UDP and IP-based applications."

It offers a set of proxy modes that might be applicable to multiplex, tunnel traffic. They might combine with MPQUIC to host the server side in a TN or NTN entities without breaking the HTTPS authentication of the target. It is developing standards to enable privacy-preserving tunnelling protocols over QUIC that are presented in Table 2-4.

Table 2-4: tunneling options Masque WG

Use Case	Description	Reference
VPN over HTTP/3	Full IP tunneling via CONNECT-IP, enabling VPN-like functionality.	RFC 9476
UDP Proxying (VoIP, WebRTC)	Tunnels real-time UDP traffic using CONNECT-UDP.	RFC 9298
Oblivious HTTP	Privacy Use Case	https://datatracker.ietf.org/doc/html/draft-kazuho-masque-ohttp

CONNECT-UDP (same for CONNECT-IP):

A client establishes a QUIC HTTP/3 connection to a proxy: It connects and authenticates in QUIC to the proxy. It requests the proxying of UDP packets to a remote server by sending an Extended CONNECT (RFC 9220) HTTP request. If the proxy accepts the proxying request, it opens a UDP socket to the target and forwards UDP packets between the client and the target. Between the client and the proxy, UDP datagrams are sent on the QUIC connection using HTTP Datagrams [32].

Like in HTTP1, the client uses the CONNECT header. But it first authenticates the proxy and then indicates the target to connect too:

:method = CONNECT

© 2023-2025 5G-STARDUST

:protocol = udp

:authority = target.example.com:443

Then the proxy opens an UDP connection to the target (using the given UDP port) and strictly relays the end-to-end UDP packets. They can be UDP packets of any application protocols like HTTP3/QUIC, RTP… It is up to the client to know the right target and port.

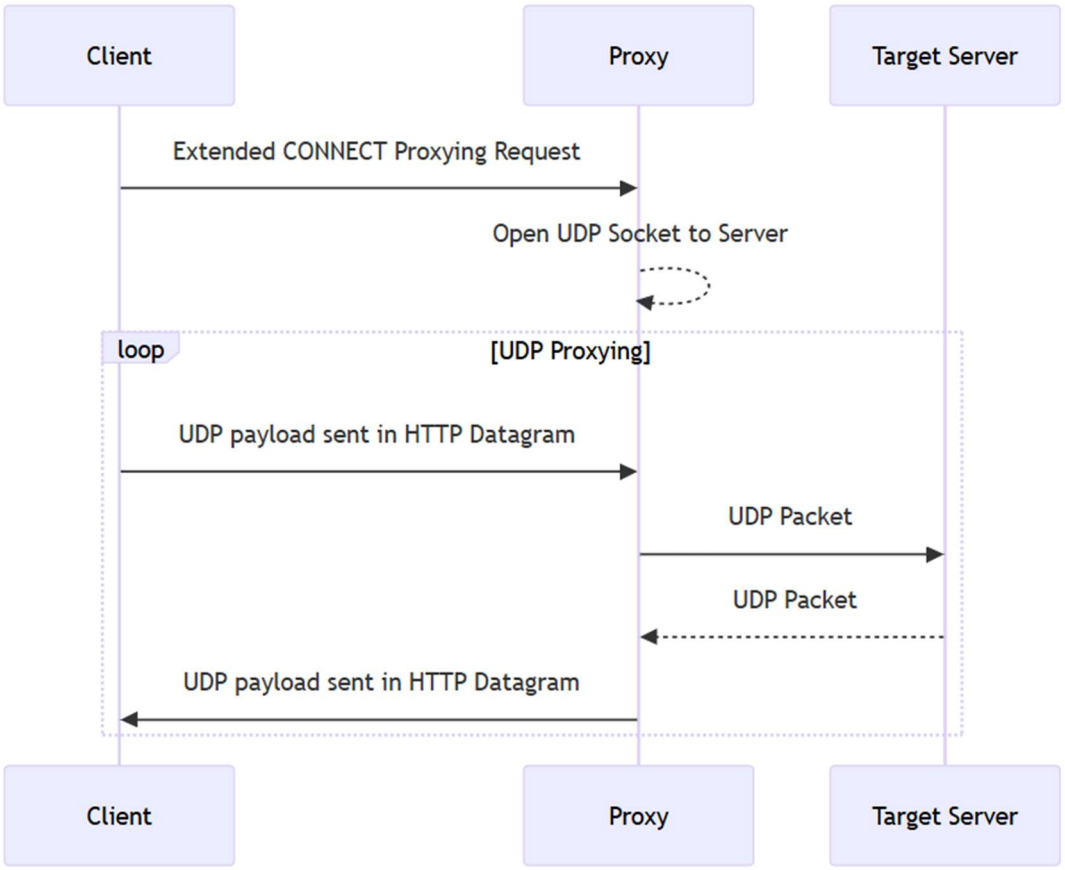

Figure 2-16: Proxying UDP in HTTP [33]

Table 2-5 present the MASQUE Implementation & Interoperability Status.

Table 2-5: MASQUE Implementation & Interoperability Status

Method	Implementations	Interoperability	References
CONNECT-UDP	Google QUICHE (masque_server & masque_client) quic-go/masque-go gmh5225/masque-go-new PreResearch-Labs/masque-go-http3-udp-proxy jromwu/masquerade	Multiple interop tests completed at IETF 110 Hackathon	https://github.com/ietf-wg-masque/draft-ietf-masque-connect-udp/wiki/Implementations https://github.com/ietf-wg-masque/draft-ietf-masque-connect-udp/wiki/Interop

© 2023-2025 5G-STARDUST

CONNECT-IP	quic-go/connect-ip-go jromwu/masquerade	Interop nearly complete	https://github.com/quic-go/connect-ip-go https://github.com/jromwu/masquerade
CONNECT-ETHERNET	Google QUICHE prototype Ericsson client prototype	ARP request/reply interop at IETF 121 Hackathon	https://datatracker.ietf.org/doc/html/draft-ietf-masque-connect-ethernet https://github.com/ietf-wg-masque/wg-materials

2.3.7.1 Evolution of QUIC security

QUIC version 1 [RFC 9000] security layer, TLS1.3 [RFC9001], is fully isolated to avoid ossification of QUIC in the future: TLS1.3 [RFC9001] is not part of the QUIC invariants [RFC 8999]. This might allow the usage of frameworks closed to TLS to secure QUIC when strongly required by the architectures or the use cases. As an example, the individual proposal “Key Exchange Customization for TIPTOP”, [35] proposes the use of Message Layer Security protocol (MLS) [rfc9750] in QUIC.

2.3.7.2 TVR, NTN and multipath

TVR (Time-Variant Routing) is an IETF WG focused on developing routing methods and models for networks where the topology or connectivity changes predictably over time. Time vary routing occurs when a node is placed on a mobile platform and the mobility of the platform causes changes to the topology of the network over time. Satellite and multipath situations are systematically referred to in the WG documents like requirements, use cases, [36] drafts but always for routing and never for end-to-end transport.

2.3.8 Disadvantages of MPQUIC in NTN

There are several limitations to use MPQUIC with intermediary agents.

QUIC security by itself, without QUIC proxying architecture specified by the Masque WG, cannot be implemented in a network function because the network function cannot be authenticated as belonging to the application server realm.

Another limitation of Masque proxying usage is the poor number of implementations compared to QUIC Masque implementations [38] versus IETF QUIC Interop Matrix [37].

Masque architecture is easier to explain with Apple iCloud Private Relay use case. It uses Oblivious HTTP/3 relaying techniques, closely aligned with MASQUE WG outputs. It routes user internet traffic through two separate relays, so using 2 QUIC connections, one operated by Apple and one by a third-party partner like Cloudflare: the first QUIC connection securely terminates in the Apple Ingress relay, hiding the client IP address from the rest of the path. The second connection securely terminates in the egress proxy, hiding the end-to-end exchanges (it includes the furniture site domain name in clear, aka QUIC CHO of the handshake) between the customer and the furniture site.

© 2023-2025 5G-STARDUST

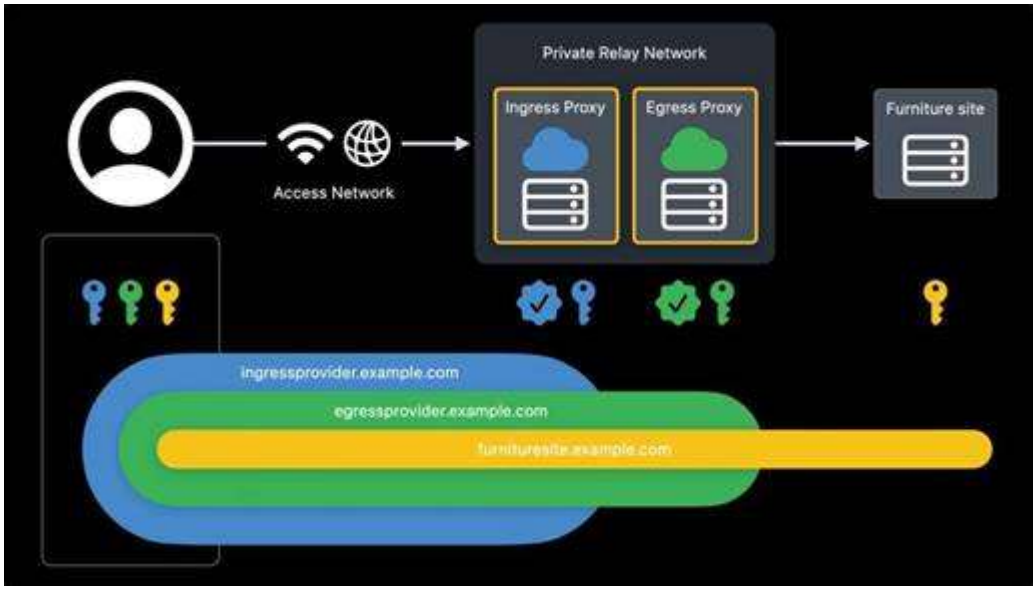

Figure 2-17: architecture Apple iCloud

Apple icloud is a very specific architecture. It works because of the trust relation between Apple devices, Apple device owners (aka consent to use Apple icloud) and Apple ingress proxy, and because of the agreement between both ingress and egress proxies (to use QUIC Client Hello encryption). Figure 2-17 illustrates the sort of architecture that can be built on top of trusted intermediaries which may enforce multi-connectivity based on QUIC.

Finally, per design QUIC and MPQUIC run in user space to let the fine grain control of the elementary transport functions to the application level (like avoiding useless retransmission, see section about transport feature above). While running applications in user space opens innovation space, it is a drawback when implementing transport layer functions in mobile network nodes, especially when deployed in cloud architecture like O-RAN where hardware accelerations mechanisms are at the core of the virtualization: TCP and MPTCP benefit directly from COTS HW acceleration functions not yet for QUIC neither MPQUIC.

© 2023-2025 5G-STARDUST

3 MULTI-CONNECTIVITY OPTIMIZATION

In this section we present four different approaches for the optimization of link utilization in an NTN-TN set up. The common underlying architecture builds on the previous sections, either in indirect or direct mode, one or multiple UEs connect via a TN and an NTN link. Multi-Connectivity is enabled by transport protocols assuming an adapted ATSSS setup. The main problem in TN–NTN integration, it is not static: the terrestrial segment typically provides relatively stable latency and capacity, whereas the satellite segment, especially in LEO constellations, exhibits a much more dynamic behaviour due to changing propagation conditions, time-varying capacity profiles, and interruptions induced by satellite handovers. The four approaches each has their own goal of optimization; they could be used stand-alone as investigated in the following sections but could also considered complementary. First approach is investigating on AI/ML for scheduling traffic on TN and NTN links. Second is introducing a hot swapping solution; for complexity issues the assumption is that an AI is selecting out of a fixed set of scheduler and congestion control algorithms a fitting set for the current network conditions. Third approach is to predict the hand-over occasion in an NTN link to be able to re-route during these since the performance during these strongly decrease. Last, we are looking at an extension to have a joint controller for MPTCP and MPQUIC to support both protocols in an optimized way for the direct and indirect setup. Therefore, optimizing the TN-NTN path usage is happening on two scales, one that is using the traffic load on the TN and NTN systems from the provided data in D4.1 and D5.1 which is rather at a longer time-scale. And a shorter time scale where we look at the link performance, e.g. the hand-over prediction.

As previously introduced, we are building mainly on MPTCP and MPQUIC, which are designed to enhance data transmission throughput and resilience by aggregating bandwidth across multiple network paths, but its efficacy is critically dependent on three core functionalities: Path management, Packet scheduling and Congestion control. The common problem that is addressed by the four approaches is that standard MPTCP schedulers, such as the default minRTT algorithm, are somehow effective in terrestrial networks where the characteristics of available paths are relatively homogeneous. However, this paradigm fails in Satellite-Terrestrial Integrated Networks (STIN). The core issue stems from the inherent heterogeneity in STIN, Default scheduler will negate the advantages of MPTCP by leaving the satellite path chronically underutilized. Which leads to known problems such as Head of line Blocking (HoL) [55], [56] which may happen very often when there is high difference of properties. In addition, congestion control (CC) may not work as intended in the context of link disruptions where the CC misinterprets a sudden RTT spike on a satellite link during handover and takes the wrong decisions and reduces performance even on uncongested links.

Various works tried to address these issues in the literature, including the reactive schedulers like BLEST (Blocking Estimation based MPTCP) [55] and ECF (Earliest Completion first) [56], using more sophisticated metrics to avoid HoL blocking by estimating packet arrival time and ensuring in-order arrival. Despite this advancement, they are fundamentally constrained by their reliance on past measurements. Which remains inadequate for the dynamics of LEO satellite environments.

The limitations of reactive scheduling in LEO Satellite environment led research towards a proactive paradigm. The core concept is to provide the transport layer with predictable lower-layer network events. ALCS (Adaptive Latency Compensation Scheduler) [57] is designed specifically for STINs. It refines its transmission latency estimates by incorporating traditional transport-layer metrics (RTT, congestion windows, queue lengths) and cross-layer information, specifically ephemeris.

© 2023-2025 5G-STARDUST

Common tool for the investigations of the four approaches is Mininet that has been used to implement the TN and NTN network. Mininet is a network emulator which creates a network of virtual hosts, switches, controllers, and links that can all be implemented in a python script. In Mininet, it is possible to easily interact with the network using the Mininet CLI (and API), customize it, share it with others, or deploy it on real hardware which makes it useful for research, development, learning, prototyping, testing, debugging, and any other tasks that could benefit from having a complete experimental network on a laptop or other PC.

Mininet hosts run standard Linux network software, and its switches support OpenFlow for highly flexible custom routing and Software-Defined Networking. Mininet provides a simple and inexpensive network testbed for developing OpenFlow applications. It enables complex topology testing, without the need to wire up a physical network. The CLI is topology-aware and OpenFlow-aware for debugging or running network-wide tests. It supports arbitrary custom topologies and includes a basic set of parametrized topologies usable out of the box without programming. It also provides a straightforward and extensible Python API for network creation and experimentation. Mininet provides an easy way to get correct system behaviour (and, to the extent supported by the hardware, performance) and to experiment with topologies. Mininet networks run real code including standard Unix/Linux network applications as well as the real Linux kernel and network stack. Due to these advantages, the code developed and tested on Mininet, for an OpenFlow controller, modified switch, or host, can move to a real system with minimal changes, for real-world testing, performance evaluation, and deployment. Importantly this means that a design that works in Mininet can usually move directly to hardware switches for line-rate packet forwarding.

Nearly every operating system virtualizes computing resources using a process abstraction. Mininet uses process-based virtualization to run many hosts and switches on a single OS kernel. Since version 2.2.26, Linux has supported network namespaces, a lightweight virtualization feature that provides individual processes with separate network interfaces, routing tables, and ARP tables. The full Linux container architecture adds chroot() jails, process and user namespaces, and CPU and memory limits to provide full OS-level virtualization, but Mininet does not require these additional features. Mininet can create kernel or user-space OpenFlow switches, controllers to control the switches, and hosts to communicate over the simulated network. Mininet connects switches and hosts using virtual ethernet pairs.

The limitation of Mininet is that it cannot exceed the CPU or bandwidth available on a single server but the tested cases in the following have been designed appropriately.

3.1 AI/ML SCHEDULER

A strategy implementing predictive AI is to observe the overall system load and predict parts in which an overload is expected by the own transmission. Figure 3-1 shows the block diagram of a scheduler and the possible inputs and outputs. The scheduler takes as basis the traffic coming from a user and get at the same time a traffic prediction for a path, i.e. either the traffic load on the NTN, the TN or both. Assuming MPTCP, there are two sockets, one for each path, and these TCP sockets provide feedback on the current state of transmission. In this approach the AI logic outside the scheduler feeds the predicted traffic for scheduling logic. Another approach would be to have the AI logic within the scheduler, where the scheduler takes decision on the logic developed by itself.

© 2023-2025 5G-STARDUST

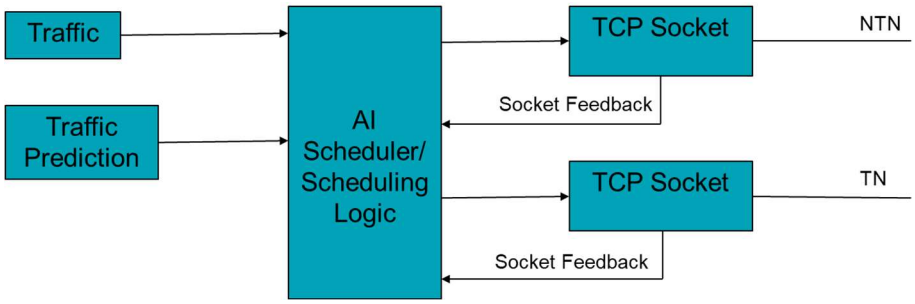

Figure 3-1: Block diagram predictive traffic scheduler

In case an AI/ML is used for scheduling for each mode of ATSSS the goal and according to rewards for the ML-algorithm can vary. For active standby the goal is to improve hand-over performance by optimizing (lowering) the switch-over time by predicting traffic loads that could lead to a switch over. The scheduler should re-direct the user traffic before it would lead to packet losses. For smallest delay the goal is to improve latency by optimizing RTT. For load balancing it is to try to be close as possible to a given threshold while optimizing data throughput and packet loss. Same hold for priority based: prioritize one while optimizing data throughput and packet loss. For the redundant mode the scheduler transmits data on both paths while optimizing reliability and resource consumption, i.e. if possible, and the load and socket conditions allows use only one link, saving energy on the second.

In standard Linux system there is a struct form the socket that can be used to retrieve the information such as the connection state, the congestion window, sent packets, lost packets, retransmits, round trip time etc. Having this information as baseline there could be multiple strategies to implement a scheduling logic/scheduler. One of them could be to train an AI model. For the prediction of the traffic, we base our work on the previous work from WP4 and WP5, i.e. for the data we use the data sets defined in the work packages and for the prediction we use the predictive model based on federated Kolmogorov Arnold networks (KANs) from WP4. Both the approaches will be further discussed in section 3.1.2.

The TN link characteristics are based on the traffic pattern of a moving train. Hence, the TN link characteristics illustrate the typical traffic pattern of train passengers. The bandwidth available for the file transfer varies based on the traffic demand that changes based on the time of the day according to D4.1. During peak hours the file transfer occurs at a reduced data rate. In the case of NTN link, the link characteristics change over time based on the relative movement of the satellite with respect to the ground node. The NTN characteristics have been obtained from 5G-system simulations using the channel models presented in [25]. The throughput of a LEO constellation with altitude of 600km has been modelled and simulated and the throughput over time has been determined. This is used to tune the NTN link performance in Mininet. Additionally, as explained the system load is considered, i.e. we see how much capacity in percentage is consumed at a given point in time and add this limitation to the link for the user.

3.1.1 Multi-Connectivity Testbed

The testbed for Multi-Connectivity (MC) scenario has been developed in Mininet[2]. The block diagram for multi-connectivity setups with Machine Learning (ML) controllers are shown in Figure 3-2 and Figure 3-3. The direct scenario in which the data transfer happens between node1 and node2.

[2] https://mininet.org/

© 2023-2025 5G-STARDUST

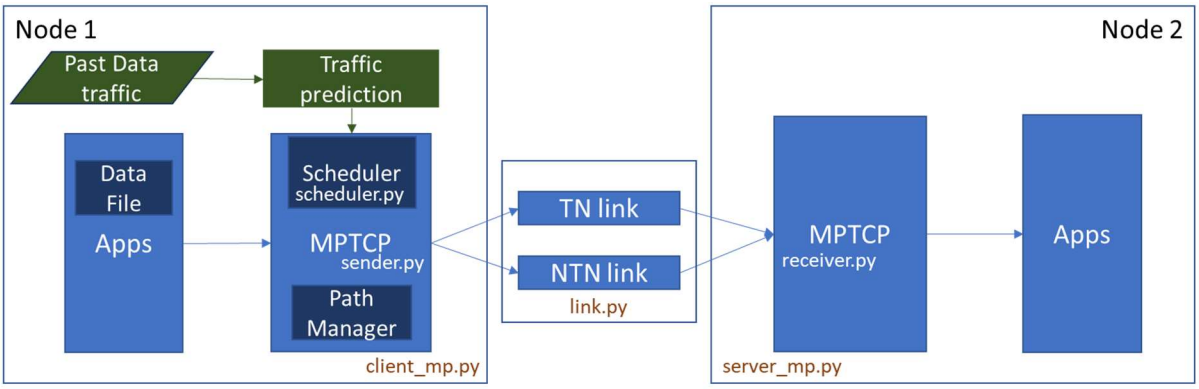

Figure 3-2 : Direct scenario - Building Blocks in testbed design

The nodes are modelled as Ubuntu hosts with MPTCP module, which inherently supports multi-connectivity. It is possible in Mininet to realize the implementation of MPQUIC for further investigation on the Multi-Connectivity setup. After establishing a TCP connection, if it is possible to create additional sockets between these hosts a new sub flow is created. For the hosts to use MPTCP, the newly added TCP option field of the underlying sub flow should be acknowledged by the remote host. If not, the connection will be downgraded to normal TCP and it will continue with a single path.

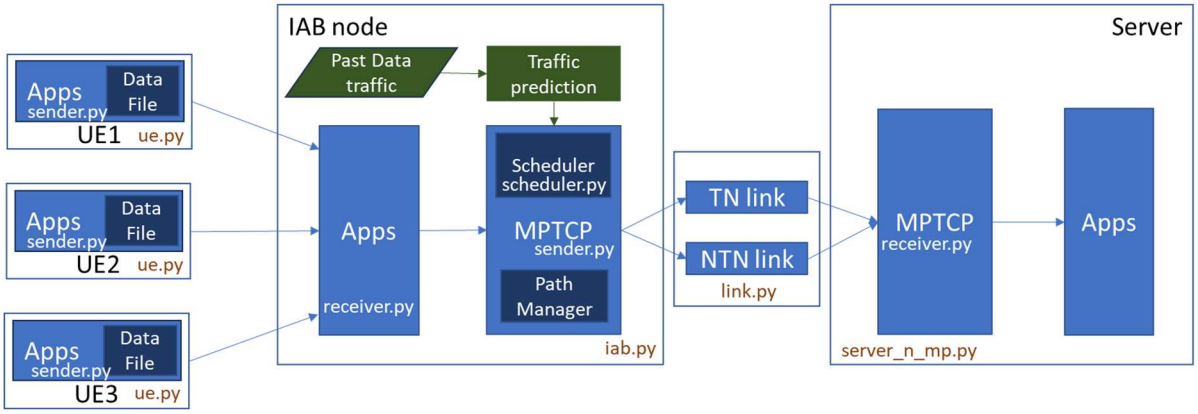

Figure 3-3: Indirect scenario - Building Blocks in testbed design

The indirect scenario in which a backhaul node is used is shown in Figure 3-3. In this case, the UEs which transfers data to the server is not MPTCP capable. To transfer data with higher rate than what is offered by a single link, the UEs make use of the IAB node. The UEs connects to IAB node with a TCP link. The IAB nodes receives the data from UE and opens MPTCP connections to the server using both TN and NTN link. The predicted traffic data helps the IAB node in the scheduling process.

For the implementation of the Path Manager from Linux versions 5.19, there are two path managers: in kernel and in user space[3]. If the PM in kernel is used, same rules are applied for all the connections. The application of different rules to each connection is possible only if the PM in user space is used. It is possible to switch between these PMs by setting the net.mptcp.pm_type sysctl as “0” (kernel) or “1” (user space).

[3] https://www.mptcp.dev/

© 2023-2025 5G-STARDUST

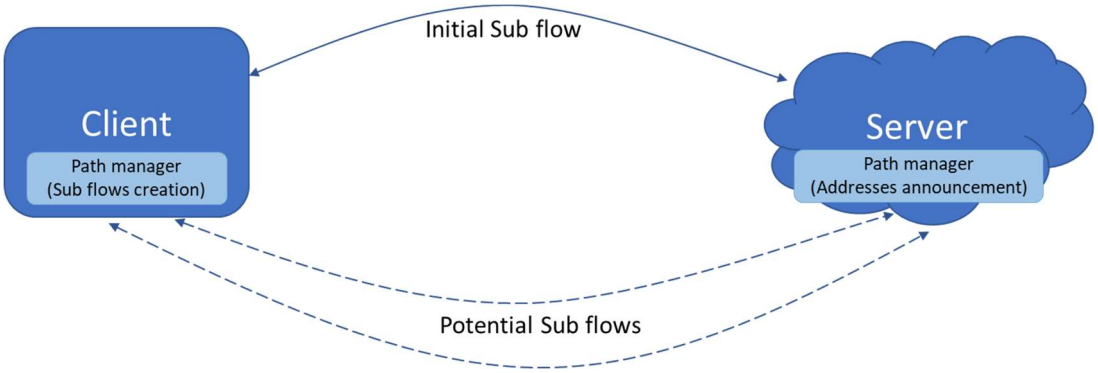

Figure 3-4 : Path manager

For the path scheduler decides on which available sub-flow to send the next available data packet as shown in Figure 3 5. To impose different rules for each sub-flow, it is essential to use the PM in user space. Once it is set to use the PM in user space, a daemon (i.e. MPTCP) can be used to decide the rules for each connection. The decision can be based on available bandwidth, latency, and any other parameter defined by the scheduling policy. Since these characteristics are different from TN to NTN, the Path Scheduler plays a vital role in the efficiency of the MC setup. The ML controller helps the Path Scheduler in this decision-making process. The overall throughput depends on the ML algorithm employed for optimization.

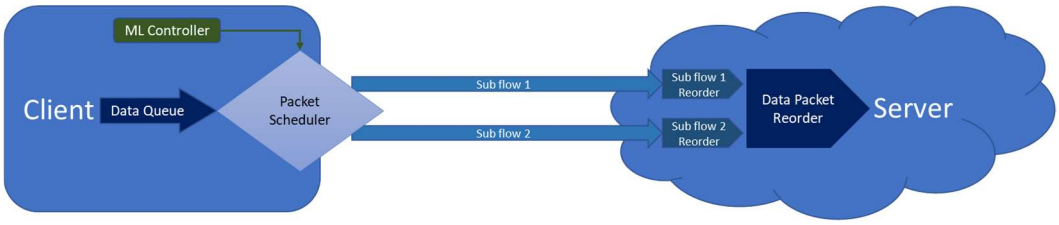

Figure 3-5 : Packet Scheduler and data packet reordering at receiver side

At the receiver end, the data packets are handled in two steps. The first step is to receive data at the sub-flow level and reorder it according to the sub-flow sequence numbers. The second step is to reorder the data at the connection level by using the data sequence numbers and finally deliver it to the application [52]. An end-to-end ML controller can be used to collect feedback from the receiver. This periodic feedback can also be used as input to the scheduler to decide on the sub-flow to send the subsequent packets.

NWDAF [54] also plays a major role in the optimization process. It is part of the 5G architecture specified in TS 23.501 [9] and has the interfaces to interact with various NFs to collect data for analytics and for ML inclusion. NWDAF contains the following logical functions:

- Analytics logical function (AnLF): Performs inference, derives analytics information (i.e. derives statistics and/or predictions based on Analytics Consumer request) and exposes analytics service.

- Model Training logical function (MTLF): Trains ML models and exposes new training services (e.g. providing trained ML model).

© 2023-2025 5G-STARDUST

The ML controllers continuously interact with NWDAF to fetch the latest analytical data and ML models to improve the decision process. NWDAF also receives information from the ML controllers regarding the performance of the MC setup which can be used to update the analytics/statistics and the ML models.

The Mininet testbed is developed in a modularised way so that it is easily customisable. The main functional blocks are explained for better understanding of the implementation.

To perform the different functionalities carried out by the hosts, the following apps have been created in python:

- sender.py - this app is capable of creating TCP sockets to connect to the destination. It is capable of creating MPTCP connection as well. To create MPTCP connections, it creates 2 sockets simultaneously and the selected scheduler divides the traffic between the sub-flows.

- receiver.py - this app can receive data in both TCP and MPTCP modes. A TCP socket is created and the app listens on that socket to accept connection and receive data. In the case of MPTCP connections, the app listens on 2 simultaneously created TCP sockets.

- link.py - This app is used to modify the link characteristics based on delay, bandwidth and PLR. Both the TN and NTN links are modified using this app.

- scheduler.py – This app helps the MPTCP module to schedule packets on each sub-flow. It has the implementations for benchmark schedulers such as Round Robin, Redundant and default (minRTT) schedulers. The proposed predictive schedulers are also available here.

These app are run on the hosts using different threads so that the hosts can fulfil the role in the testbed. To realise the different type of hosts in the network, the node scripts have been developed. These are the scripts that are run on each host. It contains the code that run apps on the nodes in different threads to realise functionalities. The different types on node scripts are:

- client_mp.py - This node is used to realise a MPTCP client sending data to the receiver continuously. It runs the sender app capable of creating an MPTCP connection with the server. The scheduler app helps in the scheduling process to send data through both the TN and NTN links.

- server_mp.py - This node can receive data sent by a MPTCP capable host. It runs the receiver app capable of creating an MPTCP connection.

- ue.py - This node creates a TCP connection with another host to send data to it. This node cannot create a MP connection. It runs the sender app capable of creating a TCP connection with another host.

- iab.py - This node acts as a backhaul node capable of creating MP connections. It receives data from UEs which are not MP capable and opens MPTCP connections to the server corresponding to each UE. It runs the receiver app which accepts TCP connections and the sender apps capable of creating MPTCP connections. To run the MPTCP sender, it also makes use of the scheduler to divide the traffic between the links.

© 2023-2025 5G-STARDUST

- server_n_mp - This node can create n MPTCP receivers, hence it can accept multiple MPTCP connections simultaneously. Runs the receiver apps which can accept MPTCP connections.

The scripts to start the simulation are:

- direct.py - This is the main script to start simulation in the case of direct scenario. It created two nodes which act as client and server. The client_mp.py script is run on the client node and server_mp.py is run on the server node. The links are also modified based on the link characteristics of TN and NTN.

- indirect.py - This script is used to start the simulation for indirect scenario. It will start 3 UEs which sends data to a backhaul node. The backhaul node receives the data via TCP sockets from the UEs and opens MPTCP connections corresponding to each UE to the server. The backhaul node run the iab.py script and the server node runs the server_n_mp.p script.

3.1.2 Predictive Scheduler

Different scheduling strategies have been implemented for comparison. The scheduler directs packets to the sub flows. The most common algorithms are Round Robin (RR), Redundant scheduling (RS) and default (minRTT) schedulers. Round Robin divides the packets equally while redundant strategy duplicates the packets on both links. The default scheduler sends packets on the link which has minimum RTT between both the sub flows. The ideal approach depends on the link characteristics. Round Robin algorithm is suitable for scenarios in which both the links have similar characteristics (i.e., homogeneous networks), while redundant scheduler is mostly used in scenarios with high Packet Loss Rate (PLR). The default scheduler is an option where the link characteristics vary over time.

We are considering two approaches for predictive scheduler. In the first approach, the predicted traffic demand is given as an input to the scheduler. For the prediction of traffic, as mentioned we build on previous work from WP4 using the KAN prediction and use the data models presented in D5.1 and D4.1.

The predictive scheduler receives the current TN link capacity and the forecasted NTN link capacity at a given instant. Then it calculates the share of the total traffic that each link can support. The incoming traffic data packets is scheduled on each sub flow based of the fraction of the total traffic each link can support. Hence, the scheduling mechanism considers the capacity in the immediate future which can change over the whole duration of the file transfer. This approach is based on the load balancing mode of ATSSS described earlier.

The scenario considered is a rural area where both TN and NTN can provide connectivity based on the traffic demand. The usual traffic demand over the NTN link has been used to predict the future traffic demand. The goal of the predictive scheduler is to maximize the throughput so as to minimize the file transfer time between the nodes. In the test bed, we have used the predicted bandwidth available for non-terrestrial links using the federated KAN approach mentioned.

3.1.3 Q-Learning

The second approach is based on the minRTT (or default) mode of ATSSS. Here, a RL algorithm is used for the scheduling logic. In the default scheduler, the current values of RTT from both the sub flows are used for deciding on which sub flow the next packet must be scheduled. The proposed RL algorithm based on Q-Learning is also based on RTT of the sub flows. The algorithm tries to reward the decisions which will reduce the RTT.

© 2023-2025 5G-STARDUST

The basic building blocks of a Q-Learning algorithm are State Space (SS), Action Space (AS) and reward [26]. The SS indicates the states the system can be in. The AS is a set of all possible actions that can be taken in a particular state. A reward is received by taking an action in a particular state. After taking an action, the system may move to another state or can continue in the same state – this depends entirely on the action was taken in that particular state. So, the algorithm computes the reward for each state-action pair and updates the values of all possible pairs over time. To map these rewards to a numerical table, it uses a Q-table. Once these values converge, the actions are guided by the values in the Q-table. The actions are taken so as to maximise the Q-values. To prevent the convergence to a local minimum value, random actions are taken in each state based on an exploration value. This exploration value to what extend random decisions are taken in each state to explore new possible outcomes in each state, which might not maximise the next Q-Value. The values in the Q-table are calculated based on the Bellman's equation as given below:

$$Q_{(s(t),a(t))} \leftarrow (1 - \delta)\, Q_{(s(t),a(t))} + \delta\, [R(t+1) + \gamma \max_{at+1} Q_{(a(t+1),a(t+1))}]$$

Where:

- s(t) – current state
- a(t) – action in the current state
- s(t+1) – next state due to action a(t)
- a(t+1) – action from the next state, s(t+1)
- R(t+1) – reward moving to the next state due to action a(t)
- δ – learning rate, determines how much new values overrides the old value influencing how quick the agent learns from its environment
- γ – discount factor, determines how much future rewards are valued compared to immediate rewards. A value close to 0 focuses on short term reward and a value close to 1 encourages long term rewards

To come up with the Q-learning model for the scheduling logic SS, AS and reward must be defined. For MP scheduling optimisation, the network parameters of both the sub flows must be considered to define the state of the system. But considering many different parameters would make the Q-table complex and require longer training time. Hence, a parameter which gives a good indication of the performance of the sub flow is essential for defining the state spaces. The send window value (swnd) is a good indicator of the state in which a sub flow is currently. The swnd is the total number of bytes that have been transmitted but not yet acknowledged in addition to the bytes that have not been sent but are ready to be received by the receiver. Hence, the state can be represented by the tuple $\{swnd_1, swnd_2\}$. The AS for the scheduler consists of the selection of the sub flows. So, there are two actions – select sub flow 1, select sub flow 2. The scheduler is based on the RTT values of the sub flows hence the reward is defined as: - RTT[action], where action is the currently chosen sub flow. The negative of the current RTT value is taken so that the algorithm tries to minimise the RTT values.

In traditional AI algorithm approach the agent is trained and the model is saved before starting the inference phase. Here, we have approached the problem such that both training and inference phases happen together. The agent learns while the scheduler selects the sub flow and the Q-Matrix is updated. During the exploration, the sub flow with the minimum RTT is selected. Hence, during the operational phase, the algorithm improves and the updated Q-matrix is used for future decision-making process.

Compared to the previous discussed algorithm, this approach does not require the predicted traffic as input. Hence, it is computationally more efficient, and the decision can be made locally without relying on the inputs from an external AI agent which makes inference on the traffic

© 2023-2025 5G-STARDUST

demand. This simplifies the algorithm implementation and reduces the control plane signalling involved in the scheduling process.

3.1.4 Predictive Scheduler and Q-Learning Results

We compare the performance of different MPTCP schedulers. The link characteristics used for all the different algorithms remain the same. The schedulers compared here are Round Robin, redundant, default minRTT and predictive schedulers. Data files of different sizes have been used to compare the performance of each algorithm.

3.1.4.1 Direct scenario

The data file is packetized and sent continuously by the sender app present on node 1 to the receiver app in node 2. The packets are scheduled on the sub flows created, as per the different algorithms taken into consideration. The receiver app present on node 2 listens on both sub flows and reorders the packets on both the sub flows to receive the entire data file.

The time to transfer the data file for different file sizes is shown in Table 3-1 and plotted in Figure 3-6. The redundant scheduler (RD) consumes more time than all the other scheduling algorithms. This was expected as the packets are duplicated on both the links and in effect doubles the number of packets needed to send the file. Round Robin scheduler consumes less time compared to redundant approach as the packets are split between the sub flows. But this split is performed by the scheduler without any knowledge of the channel characteristics. The packet flow is completed on the sub flow with higher data rate ahead of the sub flow with lower data rate. The default minRTT scheduler performs better than RD and RR schedulers. The AI based Predictive Scheduler (PS) and Q-Learning based schedulers take lesser amount of time to transfer the packet as the scheduling process takes into consideration not only the current channel characteristics but also the predicted available channel capacity over both the links and the RTT values. Each sub flow is assigned packets based on the available bandwidth for transmission. This results in non-proportional splitting of traffic between the sub flows utilisation the difference in data rate between the sub flows. For smaller file sizes, the Q-Learning approach performs better than PS. But, when the file size increases, the predicted traffic demand comes into play, and the PS can perform better that the Q-Learning based algorithm.

Table 3-1: Direct scenario - File transfer time in seconds

File Size (MB)	RD	RR	minRTT	PS	Q-Learning
100	169.296	87.504	75.466	72.508	69.862
250	409.219	209.126	177.4	173.881	170.671
500	815.874	412.884	350.272	346.674	345.496
750	1219.128	609.675	527.446	516.236	519.548
1000	1625.014	817.773	721.46	684.92	708.334

© 2023-2025 5G-STARDUST

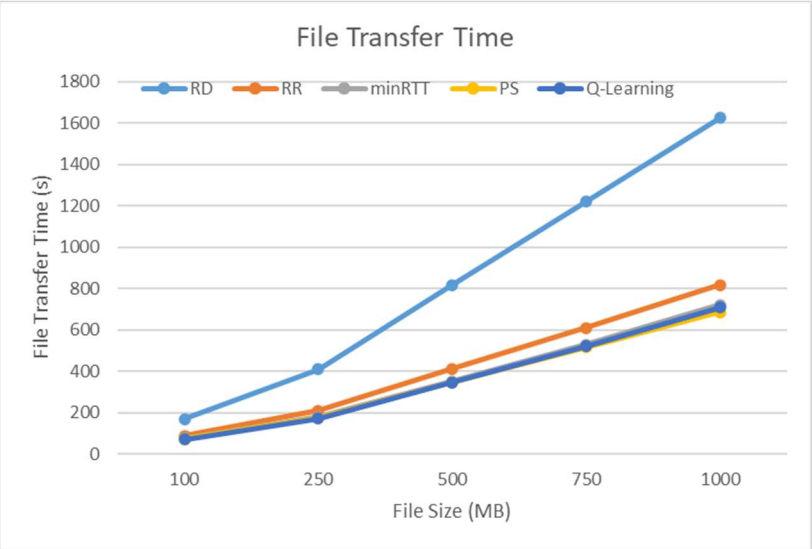

Figure 3-6: Direct scenario - File transfer time

Table 3-2: Throughput in Mbps

File Size (MB)	RD	RR	minRTT	PS	Q-Learning
100	4.725451	9.142439	10.600800	11.033265	11.451146
250	4.887359	9.563612	11.273957	11.502119	11.718452
500	4.902718	9.687951	11.419696	11.538217	11.577558
750	4.92155	9.841309	11.375572	11.622591	11.548499
1000	4.923035	9.782666	11.088625	11.680196	11.294107

© 2023-2025 5G-STARDUST

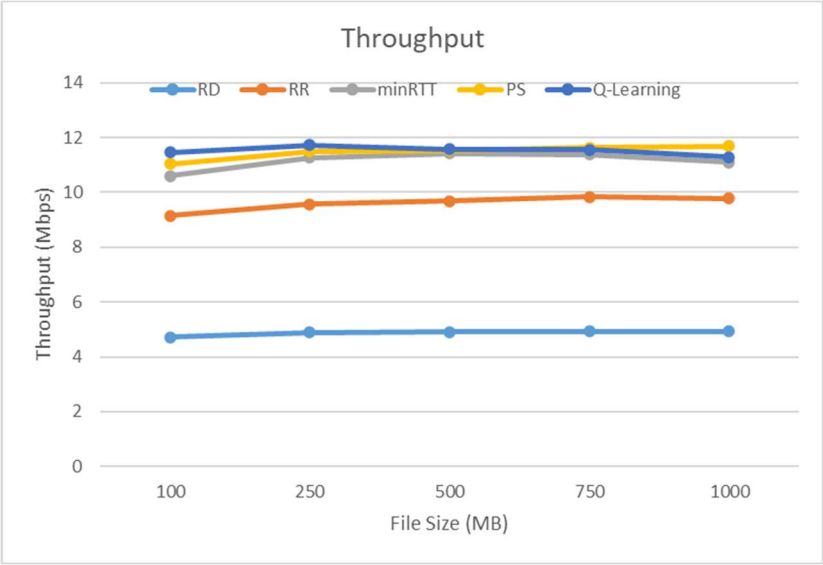

Figure 3-7: Direct scenario - Average throughput

The average throughput achieved by the different algorithms is shown Table 3-2 and plotted in Figure 3-7. The throughput achieved by different algorithms stays fairly constant for different file sizes. This is because the average capacity over a period of time remains the same. The AI based algorithms have the highest throughput due to proportional scheduling that utilises the maximum available capacity on both the links. In the case of redundant scheduler, the multi-path throughput cannot be higher than the individual sub flow data rates. This is because the throughput is calculated based on the size of the data file being transmitted. Since redundant scheduler duplicates the data packets, the throughput cannot be higher than the data rate of the slower sub flow.

3.1.4.2 Indirect scenario

In the case of indirect scenario, the UEs send the data file using the sender app to the receiver app present in IAB node. The IAB nodes uses the sender app to create MPTCP connections to the server node.

Table 3-3: Indirect scenario - File Transfer time in seconds

File Size (MB)	RD	RR	minRTT	PS	Q-Learning
100	480.859	240.246	195.956	203.618	203.908
250	1190.48	597.202	476.857	501.556	507.719
500	2361.786	1187.762	1013.787	997.86	1046.52
750	3532.803	1775.022	1600.519	1498.836	1556.671
1000	4713.283	2353.344	1977.9	1986.156	2054.926

© 2023-2025 5G-STARDUST

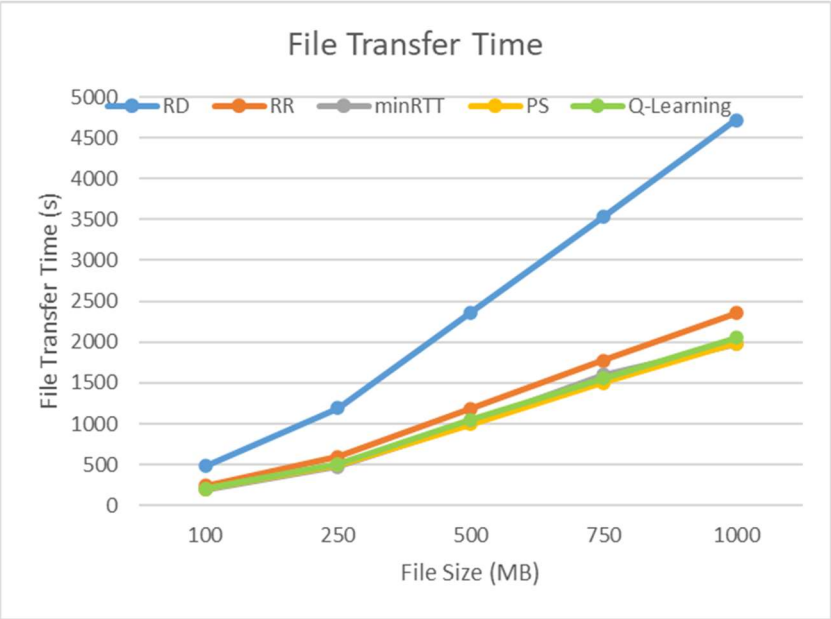

Figure 3-8: Indirect scenario - File transfer time

Table 3-4: Indirect scenario - Throughput in Mbps

File Size (MB)	RD	RR	minRTT	PS	Q-Learning
100	4.9910680	9.9897604	12.247647	11.786777	11.770013
250	5.0399838	10.046851	12.582388	11.962771	11.817560
500	5.0809006	10.103034	11.836805	12.025735	11.466574
750	5.0951043	10.140719	11.246351	12.009319	11.563136
1000	5.0919921	10.198254	12.134081	12.083642	11.679252

© 2023-2025 5G-STARDUST

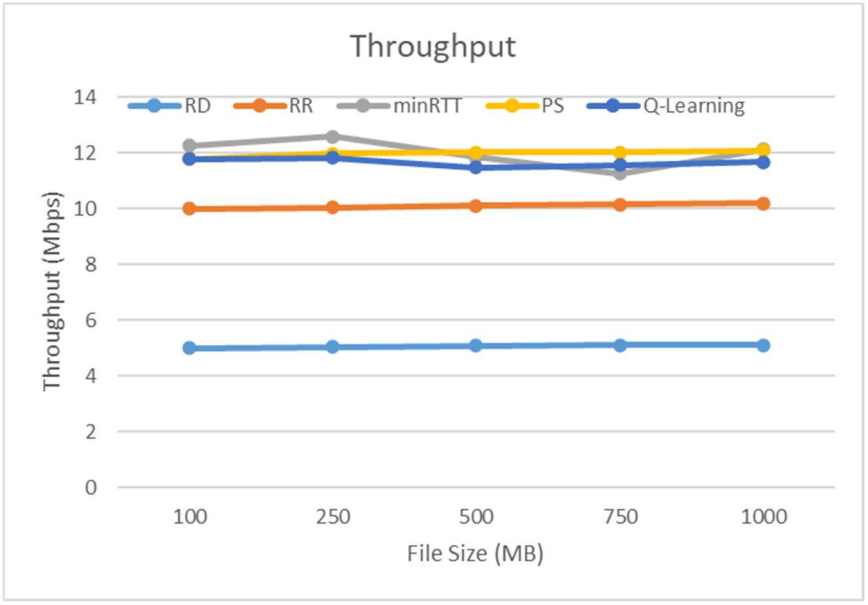

Figure 3-9: Indirect scenario - Average throughput

The time to transfer the data file for different file sizes is shown in Figure 3-8. As in the case of direct scenario, RD consumes more time than all the other algorithms.

The average throughput achieved by the different algorithms is shown in Figure 3-9. The behaviour is the same as in the direct scenario. The throughput achieved by different algorithms stays fairly constant for different file sizes. It musted be noted that the performance of the minRTT the PS and Q-Learning is in a similar range. For smaller file sizes minRTT achieved the highest rate that dropped for 500MB and 750MB, respectively, and increased again for 1000MB. The reason for this must be investigated, one assumption could be that the varying link conditions lead to environment parameters that were less good for the minRTT, for longer period min RTT could catch up again. Deeper investigations are required to understand the behaviour of the different schedulers.

3.2 RL-BASED HOT SWAPPING

3.2.1 Objectives

This work intends to design, implement, and evaluate an intelligent, real-time MPTCP scheduler hot-swapping mechanism. This mechanism leverages Reinforcement Learning (RL) and Deep Reinforcement Learning (DRL) to dynamically select the optimal scheduling strategy by learning from observed network conditions within the Mininet network emulation environment. A general representation of how an Agent interacts with its observed environment and with a MPTCP stack is shown in Figure 3-10. The rationale behind this design is the increasing prevalence of heterogeneous link environments in multi-orbit NTN communications. This design involves several key steps. First, a low-overhead data collection pipeline using command-based parsing is developed to extract a rich set of per-sub flow MPTCP performance metrics from kernel structures in real-time. Second, these metrics (e.g., smoothed RTT, delivery bandwidth, congestion window utilization, and packet loss ratios) are synthesized into state vectors to provide the learning agent with a comprehensive view of the environment. Two distinct RL-based decision agents are implemented for a comparative study: a conventional RL agent that relies on binning for state-space discretization and a more advanced DRL agent designed to handle high-dimensional, continuous state spaces natively. The action space for these agents involves switching between different MPTCP scheduling

© 2023-2025 5G-STARDUST

behaviours, achieved either by modifying sub flow flags (e.g., 'backup') on standard kernels or by swapping between distinct scheduler modules on customized kernels. Ultimately, this work aims to conduct a systematic, comparative analysis of using two types of RL techniques in terms of state representation complexity, of computational cost of the decision model, and of resulting gains in network performance.

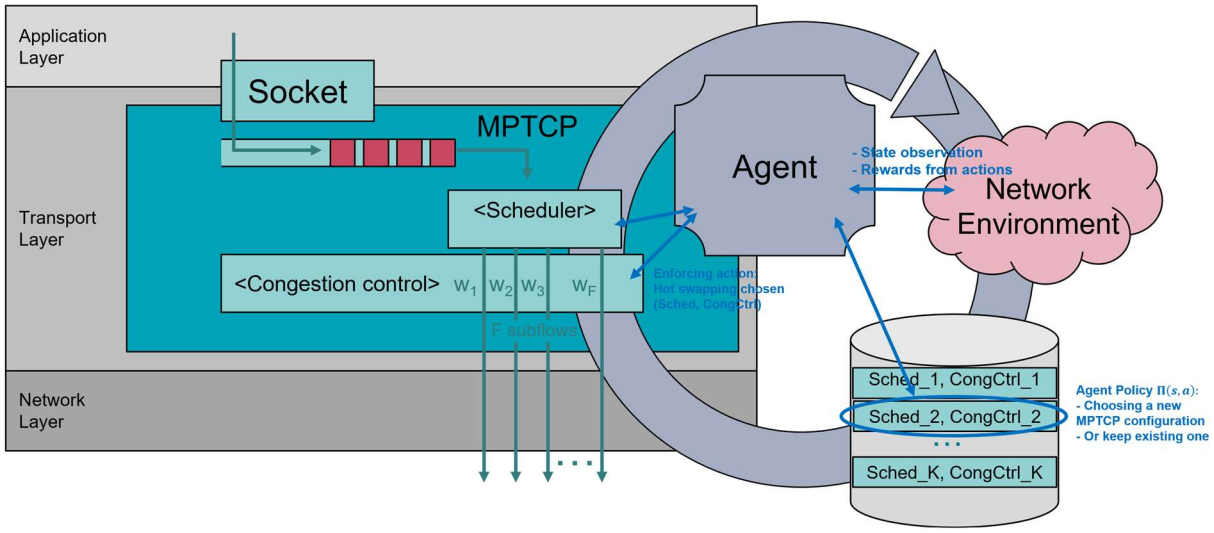

Figure 3-10 - High-level view of a RL Agent interacting with a MPTCP/Network Environment

3.2.2 Architectural breakdown

To facilitate the design, implementation, and evaluation of the MPTCP scheduler hot-swapping mechanism, a component-based architecture is defined that breaks down the system into modular, interacting elements aligned with the standard RL techniques [49],[50]. This architectural breakdown is illustrated by Figure 3-11. In this structure, the "Environment" encompasses key components responsible for simulating the world, including the Environment Controller (for setup and core simulation), Metrics Collector (raw data sensing), State Processor (observation synthesis), Action Executor (action application), and Traffic Generator (to drive workloads and influence rewards, making the environment dynamic and non-static). The Decision Agent acts as the pure RL/DRL learner and selector, interacting with the Environment via states, actions, and rewards, while the Logger/Evaluator serves as an external observer for post-hoc analysis outside the core RL loop. This RL-native grouping enhances clarity on data flows—from network emulation and metric collection to intelligent decision-making and action execution—while supporting scalability, real-time operations, and comparative analysis across RL and DRL configurations. It also simplifies integration with frameworks like Gymnasium, where a custom Gym-like environment class can wrap these components for seamless experimentation in dynamic network environments. The following subsections detail each component's primary function, drawing from the aforementioned objectives and integrating tools like Mininet, command-line utilities for metrics collection, and PyTorch. The next paragraphs give a more in-depth description of the seven components defined for this architecture.

© 2023-2025 5G-STARDUST

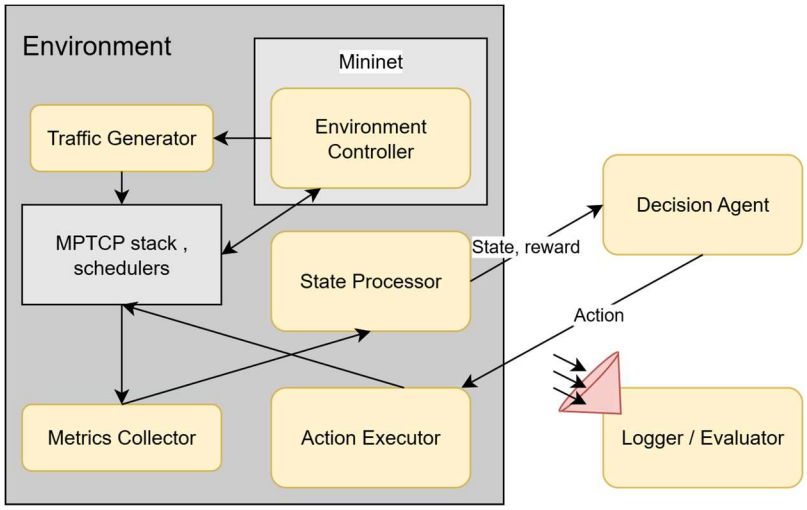

Figure 3-11 - Architectural breakdown of the MPTCP hot-swapping mechanism

2.5.1.2.1. Environment Controller

The Environment Controller component is responsible for setting up and managing the emulated network environment using Mininet, including the creation of network topologies (e.g., hosts, switches, links with configurable properties like delay, loss, and bandwidth), enabling MPTCP on hosts via sysctl configurations, and handling both initial and dynamic MPTCP endpoint/sub-flow setups. It manages sub-flow flags (e.g., backup/nobackup for scheduling Strategies A and B, as will be seen in the rest of this section). Moreover, it simulates heterogeneous network scenarios (e.g., asymmetric paths mimicking NTN/LEO satellite dynamics), serves as the core simulation platform for reproducible experiments, and handles the emulation lifecycle (start, run, stop) while ensuring MPTCP configurations support scheduler variations without disrupting ongoing flows. This component is implemented as the EnvironmentController class, which orchestrates the Mininet topology, MPTCP configurations, and interactions with other components to ensure a scalable emulation platform.

3.2.2.1 Metrics Collector

The Metrics Collector performs the real-time collection of per-sub-flow MPTCP performance metrics using a low-overhead, command-based parsing method. It queries kernel structures by executing the ss -tiM command to capture raw data points like smoothed RTT, delivery rate, and packet counters. This component is implemented as the MetricsCollector class, which provides the foundational data for the agent's environmental observations.

3.2.2.2 State Processor

The State Processor component synthesizes the raw metrics gathered by the MetricsCollector class into structured, normalized state vectors and calculates the reward signal for the learning agent. It is responsible for constructing the 4n-dimensional state vector and implementing the multi-objective reward function that balances throughput with penalties for latency, loss, and jitter. This component is implemented as the StateProcessor class, which is executed on the Mininet host to ensure real-time processing during active data transfers.

3.2.2.3 Decision Agent

This component implements the core decision-making logic, learning a policy that maps observed network states to optimal scheduling strategies. The design supports two distinct RL methods:

© 2023-2025 5G-STARDUST

- A simple Q-Learning agent, implemented in the TabularQLearning class, which uses a discretized state space and a Q-table to store state-action values.

- A DRL agent, implemented in the DRLAgent class. This agent uses a DQN, which employs a neural network to approximate the optimal action-value function from high-dimensional, continuous state spaces. The DQN agent learns from past experiences stored in a replay buffer.

3.2.2.4 Action Executor

Action Executor acts as the bridge between the Decision Agent and the live network environment. It is responsible for applying the agent's chosen actions in real-time by modifying MPTCP sub flow properties to enforce different scheduling behaviours. It ensures that these changes are applied with minimal disruption to ongoing data flows, verifying success and logging the changes for traceability. This component is implemented as the ActionExecutor class, which handles the execution of scheduler variation strategies.

3.2.2.5 Traffic Generator

This component is implemented as the TrafficGenerator class, which programmatically generates echo server and client scripts to create network workloads between hosts. A key aspect of its design is that the generated client script is responsible for more than just data transmission; it directly imports and executes the MetricsCollector and StateProcessor modules on the Mininet host. This allows for an integrated, real-time loop where per-sub flow metrics are collected and processed into state vectors during the active data transfer, providing immediate input for the decision-making process.

3.2.2.6 Logger/Evaluator

This component is responsible for comprehensive data logging and performance evaluation. It is implemented through two collaborating classes: LoggerEvaluator, which performs detailed, step-by-step logging of episode events (actions, rewards, states) for post-experiment analysis. A second class is TrainingMetrics, which tracks high-level training progress in real-time. TrainingMetrics monitors rolling reward averages, detects performance threshold achievements, and exports aggregated data for subsequent visualization purposes.

3.2.3 Reference Scenario: Asymmetric path learning with dynamic conditions

This evaluation scenario serves as a preliminary experimental framework designed to establish baseline performance metrics and validate core algorithmic behaviour under controlled conditions. The streamlined configuration enables systematic collection of initial results that will support the development of more elaborate and comprehensive scenarios incorporating additional NTN complexities. The scenario employs a single Mininet topology designed to test RL-based scheduler adaptation under representative NTN conditions. Two hosts (h1, h2) connect via asymmetric direct links simulating heterogeneous satellite paths:

- Path 1: 80ms delay, 5 Mbps bandwidth

- Path 2: 25ms delay, 15 Mbps bandwidth

Dynamic Network Behaviour: During each episode, random delay variations occur with 20% probability per decision step, replacing the original delay with values uniformly distributed between 20-200ms. This stochastic behaviour models satellite handover events and orbital dynamics without requiring deterministic scenario scripting.

© 2023-2025 5G-STARDUST

Traffic Pattern: Episodes consist of bidirectional echo traffic using 1024-byte packets with 100ms inter-packet intervals, lasting 8-40 packets per episode.

Learning Objectives: The values given for paths 1 and 2 allow the design of an asymmetric topology, able to create a fundamental and measurable trade-off, providing in turn a learning challenge for the RL Agents. This trade-off is rooted in the distinct, explainable behaviours of the two scheduling strategies (detailed in Section 3.1.5.2), which are implemented by manipulating the backup flag on the built-in minRTT scheduler:

- Scheduling Strategy B (prioritized) provides predictable latency stability. It confines traffic primarily to the superior path by forcing the minRTT scheduler to exclusively use Path 2 for its significantly lower 25ms latency. This results in a stable, low-latency connection but caps the maximum throughput at 15 Mbps, as the bandwidth of Path 1 is left entirely unused.
- Scheduling Strategy A (aggregating) aims to maximize throughput by combining both paths. This approach unlocks a higher potential data rate (up to 20 Mbps) but introduces the significantly higher latency of Path 1 (80ms) into the connection, making overall performance less stable and predictable.

The agent's primary objective is to learn context-dependent switching. It must discern from the network metrics when to favour the higher throughput of Strategy A during stable conditions, and when to adapt by switching to the reliability of Strategy B as the stochastic delay variations begin to degrade path performance. This purposefully designed difference in outcomes is key, as it provides the needed causal signals for the RL Agents to build a valuable policy.

3.2.4 Impact of development environment

3.2.4.1 Target environment

This environment uses Linux Fedora 42 with Kernel 6.16, providing a stable baseline with an updated MPTCP implementation that avoids known path manager bugs in older kernels. In effect, reliable sub flow creation and management is performed using the kernel path manager (pm_type=0) [41]. MPTCP is enabled via sysctl configurations, supporting multipath testing with default schedulers like minRTT, ideal for reproducible experiments in Mininet with updated tools [42]. It facilitates RL-based swapping through observable metrics from kernel structures, ensuring compatibility with modern distributions for broader applicability, with confirmed OpenVSwitch service and kernel modules for network emulation [43].

3.2.5 Emulation Implementation

The network environment is built by the EnvironmentController class, which automates the setup in Mininet. The emulated topology consists of two hosts, h1 and h2, connected by two direct, asymmetric links designed to simulate path diversity. MPTCP is enabled on both hosts using sysctl commands to activate the kernel path manager (net.mptcp.pm_type=0), and initial MPTCP endpoints are configured for each network interface using the 'ip mptcp endpoint add' command to ensure both paths are available for the connection.

3.2.5.1 Data collection

The solution collects per-sub flow MPTCP metrics (e.g., smoothed RTT and delivery rate) to form a state vector for RL-based scheduler swapping decisions, enabling the RL agent to evaluate network conditions and switch strategies dynamically in the target environment, with minRTT and variants using backup flags [43]. In addition, the solution focuses on efficient extraction using kernel queries to capture real-time sub flow performance, providing the RL agent with low-latency feedback on throughput and latency to identify states like congestion on the primary path (high smoothed RTT, low rate in Strategy B) or aggregation inefficiencies

© 2023-2025 5G-STARDUST

(variable smoothed RTT in Strategy A), facilitating optimal swaps for performance improvement [44]. This approach ensures scalability by aggregating metrics and producing periodic summaries, supporting RL state-action pairs and scheduler changes to balance throughput and latency [44].

Real-time data collection is performed by the MetricsCollector class, which executes the 'ss -tiM' command via a Python subprocess call to retrieve comprehensive tcp_info structures for each sub flow directly from the kernel. This command-based parsing approach ensures access to critical metrics such as smoothed RTT, delivery bandwidth, congestion window and packet counters (e.g., lost_out, retrans_out, segs_out). Data is captured and aggregated during monitoring loops initiated by the client script to observe dynamic network changes in real-time.

3.2.5.2 Scheduler Variation Strategies

This work simulates two distinct MPTCP scheduling behaviours by manipulating sub-flow properties rather than by loading different scheduler modules. This approach allows for dynamic adaptation using the single, built-in minRTT scheduler. The two resulting strategies create a clear and measurable trade-off between throughput maximization and latency stability, providing the reinforcement learning agent with distinct choices for optimizing network performance based on observed conditions.

- **Scheduling strategy A, the "aggregating" variant**, is configured to prioritize maximum throughput. It enables the scheduler to perform active load balancing across all available sub flows, effectively combining their bandwidth.

- **Scheduling strategy B, the "prioritized" variant**, is configured to prioritize stable, low latency. It confines traffic primarily to a single, optimal path, using secondary paths only for failovers, which results in more predictable round-trip times.

The scheduler variation strategies are applied at runtime by the ActionExecutor class. Given the constraints of the standard Linux kernel environment where new schedulers cannot be dynamically loaded, these strategies are implemented by programmatically modifying sub flow properties. Specifically, the 'ip mptcp endpoint change id backup/nobackup' command is used to toggle the backup flag on MPTCP endpoints. For scheduling strategy A, all sub flows are set to 'nobackup', allowing the minRTT scheduler to aggregate traffic for maximum throughput. For scheduling strategy B, the secondary sub flow is marked as 'backup', restricting traffic to the primary path to ensure stable latency. This method allows for dynamic strategy switching without requiring kernel modifications and produces distinct, observable changes in performance metrics like delivery rate and round-trip time.

3.2.6 Design of the Decision Point

The decision point employs RL and DRL mechanisms to enable dynamic scheduler swapping based on observed network states, allowing for an evaluation of the trade-offs between computational cost and decision performance [46], [47]. The experimental design facilitates a direct comparison between two distinct agent types: a traditional RL agent using state discretization and a DRL agent designed to handle continuous states natively.

Both agents are tasked with learning an optimal policy from a single, comprehensive state vector that incorporates a rich set of metrics, including latency, throughput, utilization, and loss ratios. This framework is designed to measure how the two learning paradigms differ in terms of training efficiency, decision accuracy, and resource consumption when operating on the same complex state representation, particularly in emulated heterogeneous NTN scenarios involving challenges like LEO satellite handovers [48]. Through evaluations in Mininet, the goal

© 2023-2025 5G-STARDUST

is to quantify the pros and cons of each configuration, thereby informing optimal designs for MPTCP scheduler hot swapping.

3.2.6.1 State vector design

To provide the learning agent with a comprehensive, real-time view of network conditions, the state vector S_t is constructed as a 4*n*-dimensional vector for a connection with *n* sub flows. This vector incorporates four key metrics for each sub flow to capture latency, throughput, utilization, and reliability.

The state vector is defined as $S_t = \left(A_{t,1}, B_{t,1}, C_{t,1}, D_{t,1}, \cdots, A_{t,n}, B_{t,n}, C_{t,n}, D_{t,n}\right)$ where the components for each subflow *i* are:

- $A_{t,i}$ represents smoothed round-trip time, derived from srtt_us metrics and aggregated via histograms over configurable intervals (e.g., 100ms), to balance responsiveness and computational efficiency.
- $B_{t,i}$ represents effective delivery bandwidth, derived from rate_delivered combined with rate_interval_us and MSS in tcp_sock, converted to bps and aggregated in maps.
- $C_{t,i}$ represents normalized $\mathrm{CWND/RTT}$ proportion, computed as $\frac{\mathrm{snd\_cwnd}_i/\mathrm{srtt\_us}_i}{\sum_{k=1}^{n}\mathrm{snd\_cwnd}_k/\mathrm{srtt\_us}_k}$, with snd_cwnd and srtt_us from tcp_sock, extracted and aggregated.
- $D_{t,i}$ represents recent packet loss ratio, computed as $\frac{\mathrm{EWMA}(\Delta(\mathrm{lost\_out+retrans\_out}))}{\mathrm{EWMA}(\Delta\mathrm{segs\_out})}$, with lost_out, retrans_out, and segs_out from tcp_sock, deltas calculated over intervals and aggregated [48].

3.2.6.2 Reward function

Defining an appropriate reward function is a significant challenge in the context of NTN. In heterogeneous, multi-orbit architectures, MPTCP sub-flows exhibit different characteristics, e.g., in terms of latency, throughput, and stability. A primary difficulty lies in defining a utility function that enables the learning agent to distinguish between network performance degradation caused by genuine congestion and that caused by physical link phenomena, such as the frequent, transient disruptions inherent to LEO satellite handovers. A naive reward signal focused solely on throughput would fail to capture this nuance, potentially teaching the agent counterproductive behaviours. The reward function must therefore provide a comprehensive and robust signal that incentivizes not just speed, but also resilience, stability, and the proactive mitigation of predictable link disruptions.

Formulation

To address the complexities of heterogeneous and dynamic networks, the reward R_t at each timestep *t* is formulated as a multi-objective utility function. This function balances the goal of achieving highly efficient throughput with penalties for poor latency, reliability, and stability.

The reward is calculated as follows:

$$R_t = \alpha \cdot \mathrm{goodput\_norm} - \beta \cdot \mathrm{latency\_norm} - \gamma \cdot \mathrm{loss\_ratio} - \delta \cdot \mathrm{jitter\_norm}$$

With:

© 2023-2025 5G-STARDUST

- Goodput_norm: Represents the normalized effective data rate, scaled by a transmission efficiency factor that penalizes packet retransmissions. This encourages efficient, successful data transfer over raw transmission speed.

- Latency_norm: The normalized smoothed RTT of the primary sub flow, representing the connection's responsiveness.

- Loss_ratio: A direct measure of the connection's reliability, calculated from the ratio of lost and retransmitted packets to total packets sent.

- Jitter_norm: The normalized RTT variance, which captures the stability and predictability of the path.

- $\alpha, \beta, \gamma, \delta$: A set of weighting coefficients where $\alpha + \beta + \gamma + \delta = 1$, which collectively define the agent's policy by prioritizing these competing objectives.

Implementation

From an implementation perspective, the reward function's components are calculated within the State Processor component. This component derives the necessary values from the per-sub flow metrics previously extracted from the kernel's tcp_info structure by the MetricsCollector. Specifically, Goodput and Loss_ratio are calculated using the delta values of packet counters (including segs_out, lost_out, and retrans_out) combined with the rate_delivered metric. The connection's Latency is based on srtt_us, while Jitter is derived from rttvar. To ensure a consistent scale, these metrics are normalized using min-max scaling against the environment's theoretical bounds. Additionally, to promote policy stability and prevent rapid oscillations (known as "flapping"), a small negative penalty is applied to the reward whenever the agent's action changes from the previous timestep.

Weight Tuning Methodology

The use of a four-component reward function introduces the challenge of selecting appropriate weights $(\alpha, \beta, \gamma, \delta)$ that align with the specific characteristics of the network scenario. For the highly dynamic LEO satellite handover context, the weights are chosen to provide meaningful reward signals that distinguish between different network conditions without requiring optimization. A fixed configuration of $(\alpha = 0.4, \beta = 0.3, \gamma = 0.25, \delta = 0.05)$ is used for the experiments to ensure consistent reward scaling for both RL and DRL methods.

This weighting configuration makes the reward function sensitive to the sharp spikes in packet loss and latency that indicate handover disruptions. This configuration is at the same time able to maintain sufficient goodput emphasis to encourage effective bandwidth utilization. The fixed-weight approach allows skipping hyperparameter tuning steps in the computational complexity comparison. In addition, the effectiveness of this reward configuration is validated by measuring whether both methods can reach the established performance threshold within the allocated training time, rather than through iterative weight refinement. This methodology prioritizes fair comparison of computational trade-offs over reward function optimization, aligning with the research objective of quantifying, in a practical emulation framework, efficiency differences between RL and DRL approaches.

3.2.7 RL-based implementation

The RL-based implementation uses a custom tabular Q-learning approach rather than neural network-based methods to provide a learning baseline for comparison. This implementation uses discrete state spaces created through manual binning of the continuous state vectors,

© 2023-2025 5G-STARDUST

storing exact Q-values in lookup tables rather than function approximation. The tabular approach enables precise value tracking for each state-action pair while avoiding the computational overhead and generalization characteristics of neural networks. This design choice ensures a clear methodological distinction from the DRL implementation, easing the meaningful comparison of both methods in terms e.g., of computational complexity, memory usage, and convergence behaviour. The implementation integrates directly with the existing Mininet infrastructure for real-time metric collection and MPTCP scheduler swapping without requiring external RL framework.

3.2.7.1 State space discretization approach

As previously mentioned, discretization is achieved by partitioning each dimension of the state vector into a fixed number of uniform bins. For the 8-dimensional state vector of the reward function in the context of the reference scenario, this discretization uses 5 bins per dimension, yielding a total discrete state space of 5^8 = 390625 possible states. The binning operation achieves $O(n \cdot \log(b))$ computational complexity, where *n* represents the state dimensionality and *b* the number of bins per dimension, ensuring minimal overhead during real-time decision-making [11]. It is worth highlighting that the value of 5 bins represents a conservative approach, which balances state resolution with computational complexity, since higher bin counts are susceptible to lead to exponential state space growth. Such growth would become computationally prohibitive, especially in the context of virtualized network function deployments. Practical analysis suggests that 6-7 bins per dimension constitute the upper bounds for reasonable memory consumption and convergence time in resource-constrained environments, with Q-table memory requirements scaling from approximately 27 MB (6 bins) to 92 MB (7 bins), while 8+ bins approach gigabyte-scale memory usage. This consideration becomes particularly relevant for dynamic scenarios such as LEO satellite handovers, where finer state granularity might be desirable to capture rapid network condition fluctuations, yet must be balanced against the computational overhead of larger state spaces.

3.2.7.2 DRL-based implementation

For the DRL-based implementation, we use a DQN agent. Our implementation, built entirely in PyTorch, consists of three core components:

- **Experience Replay Buffer**: This structure stores transitions of "(state, action, reward, next_state, done)" collected from the environment. By sampling mini batches from this buffer, the agent can learn from a diverse set of uncorrelated experiences, which stabilizes the training process.

- **Q-Network**: A feed-forward neural network with two hidden layers of 64 neurons each serves as the function approximator. It takes the continuous 8-dimensional state vector as input and outputs the estimated Q-value for each of the two possible actions (i.e., scheduling strategy A or B).

- **Target Network**: To prevent training instability, a separate target network with the same architecture is used. Its weights are periodically updated with those from the main Q-network, providing a stable target for the loss calculation during gradient descent updates.

The agent follows an ε-greedy policy for exploration, and the entire training process is managed by a unified script, which feeds experiences into the Decision Agent and triggers learning updates.

© 2023-2025 5G-STARDUST

3.2.7.3 RL-DRL comparison objectives

In the context of this first reference scenario, the objective is to experiment with this scheduler hot swapping design, in our practical Mininet testbed, with computational trade-offs in mind for each RL and DRL method. To that end, the comparison setup evaluates three aspects through data collection and visualization:

- Learning efficiency of each RL/DRL method through reward curves. This approach captures convergence rates and relative learning speeds rather than focusing solely on final performance metrics.
- Threshold achievement timing by tracking multiple performance levels. This multi-threshold analysis maps which approach achieves different performance requirements more efficiently.
- Resource consumption, including memory usage during training and inference, CPU utilization, training duration, and storage requirements. This analysis addresses practical deployment considerations by measuring the computational cost of achieving comparable performance levels, in the context of the reference scenario.

This experiment design establishes a baseline performance threshold (average reward above zero) and measures efficiency in reaching this target rather than pursuing optimal performance. A reason is the observation that the heterogeneous nature of both chosen RL/DRL approaches makes it difficult to fairly value their respective core parameters for fair comparative analysis. In effect, these parameter sets govern different mathematical processes and lack systematic equivalence, making harmonization attempts counterproductive. Rather than pursuing harmonization that risks artificially constraining superior methods, tests were passed with both sets of hyperparameters configured with their default recommended values from authoritative sources [49][50] and [51]:

- For tabular Q-learning, parameters are default-valued according to [49].
- For DQN, parameters adhere to specifications from [51] and implementation guidance from [50].

This approach preserves each algorithm's characteristic behaviour while ensuring both methods operate at established baseline effectiveness levels.

3.2.8 Hot-Swapping Results

The comparative evaluation of tabular Q-learning and DQN-based decision agents was conducted over identical 1-hour training sessions, each comprising 201 episodes with 8 packets per episode on the same asymmetric network topology.

© 2023-2025 5G-STARDUST

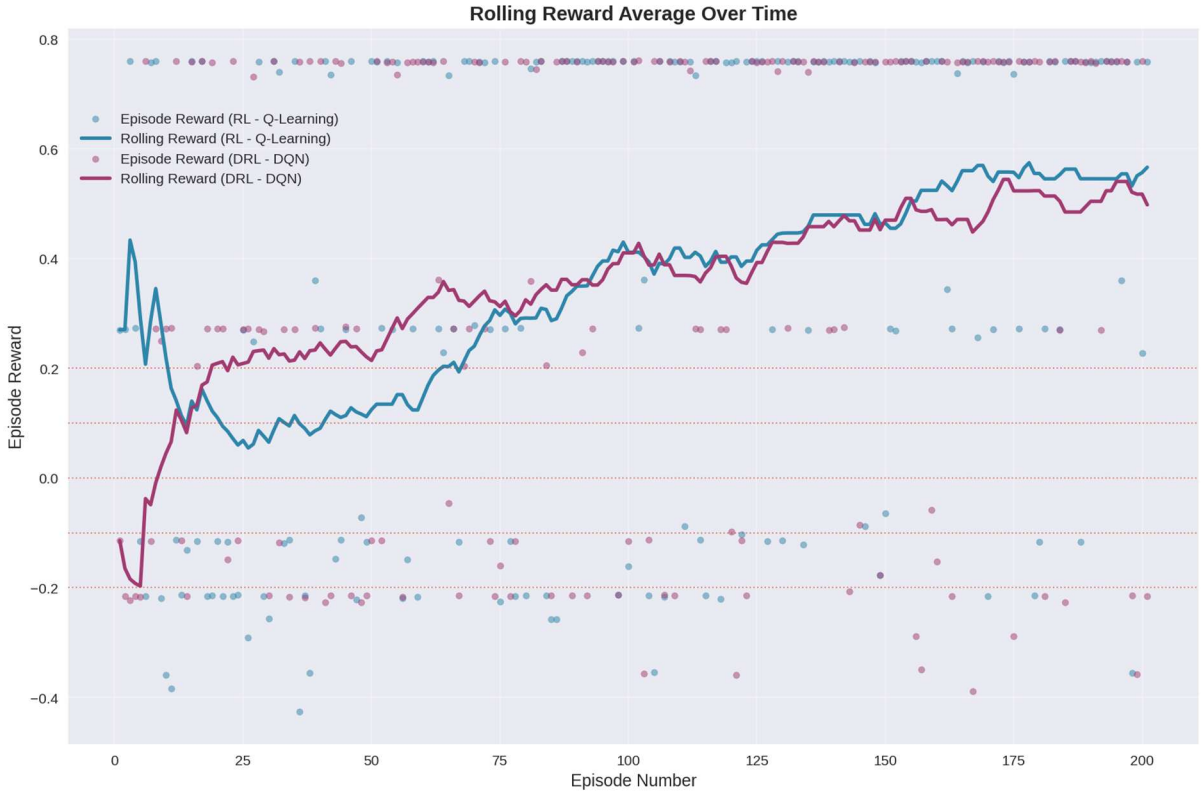

Figure 3-12 – Rolling reward average for Q-learning and DQN methods

Learning Performance and Convergence

Both tabular Q-learning and DQN approaches achieved positive learning outcomes, as illustrated by Figure 3-12, with Q-learning reaching a final rolling average of +0.567 and DQN achieving +0.498. Both methods satisfy all five performance thresholds (-0.2, -0.1, 0.0, +0.1, +0.2) within the training timeframe. The tabular method exhibited pronounced early exploration followed by stable exploitation, while DQN showed gradual improvement with continued exploration throughout training.

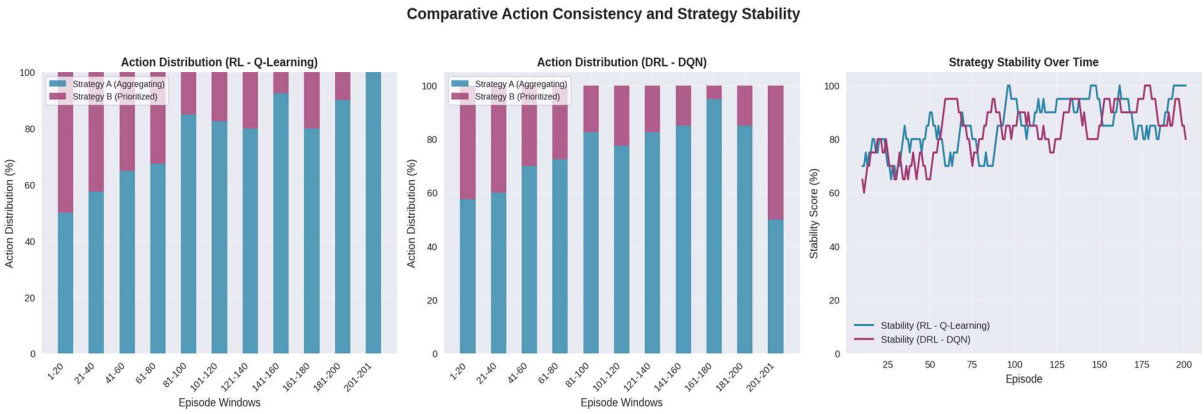

Figure 3-13 - Action distribution and Strategy Stability for Q-learning and DQN methods

Strategy Consistency and Decision Stability

© 2023-2025 5G-STARDUST

Figure 3-13 reveals distinct behaviours across training progression. The tabular agent demonstrated increasing preference for Strategy A (aggregating) throughout most episode windows, while DQN maintained more balanced action selection with Strategy A comprising approximately 60-70% of decisions across the majority of training windows. However, interpretations of the final episode windows must be approached cautiously, as they represent extremely small sample sizes (the rightmost windows contain only 2 actions from a single episode, making percentage calculations statistically unreliable). The strategy stability curves show that the tabular method achieved generally higher within-episode consistency, indicating more decisive policy convergence, while DQN exhibited greater variability throughout training.

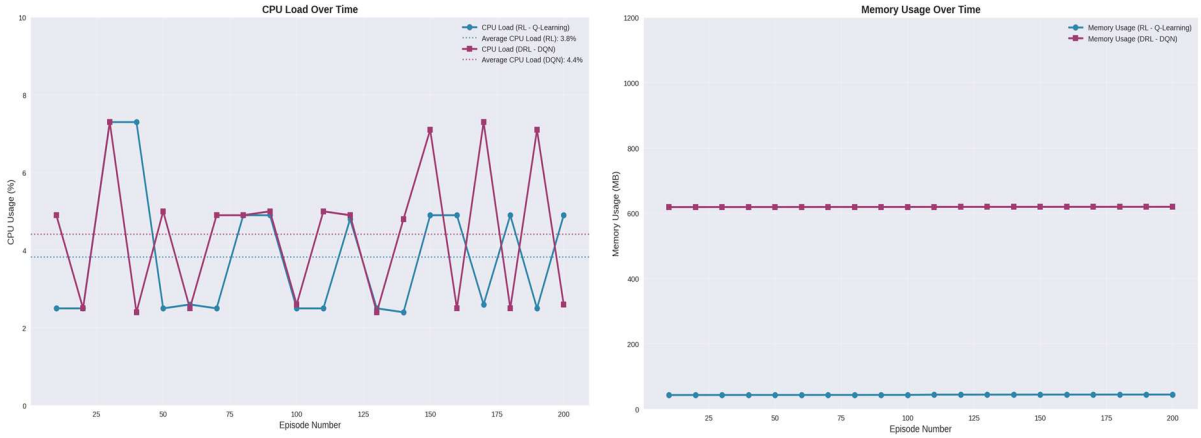

Figure 3-14 - CPU load and memory usage for Q-learning and DQN methods

Computational Resource Requirements

Memory consumption differed significantly as show in Figure 3-14, right plot: tabular Q-learning maintained stable usage at 44 MB while DQN required 620 MB, naturally reflecting design differences between discrete lookup tables and neural network architectures. CPU utilization proved similar between methods, averaging 3.8-4.4% (Figure 3-14, left plot), indicating comparable computational intensity during training.

Training Efficiency and Scalability

Both methods completed 201 episodes within the 1-hour window, yielding equivalent throughput rates of 200 episodes per hour. The computational bottleneck resided primarily in network emulation rather than algorithmic processing. These results must be interpreted within the simple reference scenario. The asymmetric dual-path topology with extremely short episodes (8 packets being in this case equivalent to 2 decision steps) naturally favours tabular approaches with small binning, enabling exact value functions with only 4^8 = 65,536 discrete states and modest memory requirements of 0.5 MB.

However, it is very important to highlight that increasing scenario complexity along multiple dimensions would rapidly favour DQN. Beyond finer state discretization requirements (scaling Q-table memory from 0.5 MB to 13.4 MB at 6 bins, reaching 134 MB at 8 bins), longer episodes with extended decision sequences would challenge tabular methods' ability to learn effective temporal policies. More realistic scenarios involving dozens or hundreds of packets per episode, multiple sub-flows, and dynamic network conditions would require the sequential learning capabilities and generalization properties of neural network approaches. DQN's continuous state handling, fixed memory footprint, and capacity for temporal credit assignment provide core scalability advantages for heterogeneous NTN environments with extended

© 2023-2025 5G-STARDUST

decision timescales. Those early results based on a simple reference scenarios must therefore be relativized by the observation that while traditional Q-learning behaves well in highly constrained scenarios, DRL provides, with its scalable design, the means to address the complexity challenges of more elaborate network scenarios, and practical MPTCP deployment contexts.

3.2.9 Next Steps Hot Swapping

In the context of the test of this AI/ML-based hot-swapping scheduler, we adopted several parametric assumptions to facilitate testbed validation and enable clear interpretation of the RL/DRL comparison framework. Particularly, an inter-packet timing of 100ms was chosen to ensure observable state transitions in MPTCP sub-flow metrics, providing stable statistical backgrounds for agent decision-making. While this approach successfully validates the core infrastructure and enables systematic comparison between traditional RL and DRL methods, it introduces limitations on the interpretations of the results. For instance, this packet pacing artificially stabilizes the dynamics of the modelled traffic, potentially masking the MPTCP scheduler limitations this experimental setup can contribute to quantify. Similarly, the current metrics collection approach using command parsing, while robust and easily implementable, introduces measurement overhead that constrains traffic generation rates. This subprocess-based approach may become a bottleneck when transitioning to more realistic, high-frequency traffic patterns. Additionally, Mininet's network emulation capabilities at high packet rates remain to be characterized for this specific application context. Therefore, future revisions of this scheduler hot-swapping experiment should address these limitations through the following complementary approaches:

- Implementing more detailed traffic (e.g., burst-based) would model realistic application behaviours (web requests, video streaming) while maintaining measurement observability through structured inter-burst pauses.
- Explicitly considering more link disruption causes such as handover events in reference scenarios could introduce mid-episode topology changes that more accurately reflect LEO constellation dynamics.
- Direct measurement of scheduler response times would quantify MPTCP adaptation delays to rapid network changes.
- The support of Berkeley Packet Filtering (BPF) modular MPTCP schedulers [45] would incur more scheduling behaviour variety and enriched action spaces, allowing more realistic and comprehensive tests.
- Transitioning to lower-overhead metrics collection methods (such as Extended Berkeley Packet Filter - eBPF-based kernel probes, direct `getsockopt()` calls with `SOL_MPTCP`, or MPTCP netlink interfaces) could support realistic traffic rates without measurement interference.

3.3 HAND OVER PREDICTION FRAMEWORK

As mentioned at the beginning of this section, the switch from reactive to proactive scheduling marks a significant improvement in MPTCP's performance, but the feature of multi-orbit architecture introduces levels of dynamic complexity. Managing handover between orbital layers will require more than just scheduling; but proactive scheduling based on predictive HO information related to satellite trajectory to enhance Default Scheduler before HO occurs.

© 2023-2025 5G-STARDUST

The objective of this approach is first to improve the scheduler's behaviour in the context of satellite plus multi-orbit architecture and to leverage the productive approaches compared to reactive approaches mentioned before. Even more than that we intend to have a predictive ability to mention to the scheduler when it will have to finetune its behaviour in preparation of a handover.

3.3.1 Proposed Framework

To address the inherent limitations of reactive schedulers in dynamic, multi-orbit satellite network environments, we defined a predictive and proactive framework. The core design principle consists of a clear separation between non-real-time control and prediction actions and the real time actions, executed by the MPTCP stack. This modular approach consists of four interacting components: Constellation Simulator, Prediction Engine, Policy Engine and Schedular Actuator, as is illustrated in Figure 3-15.

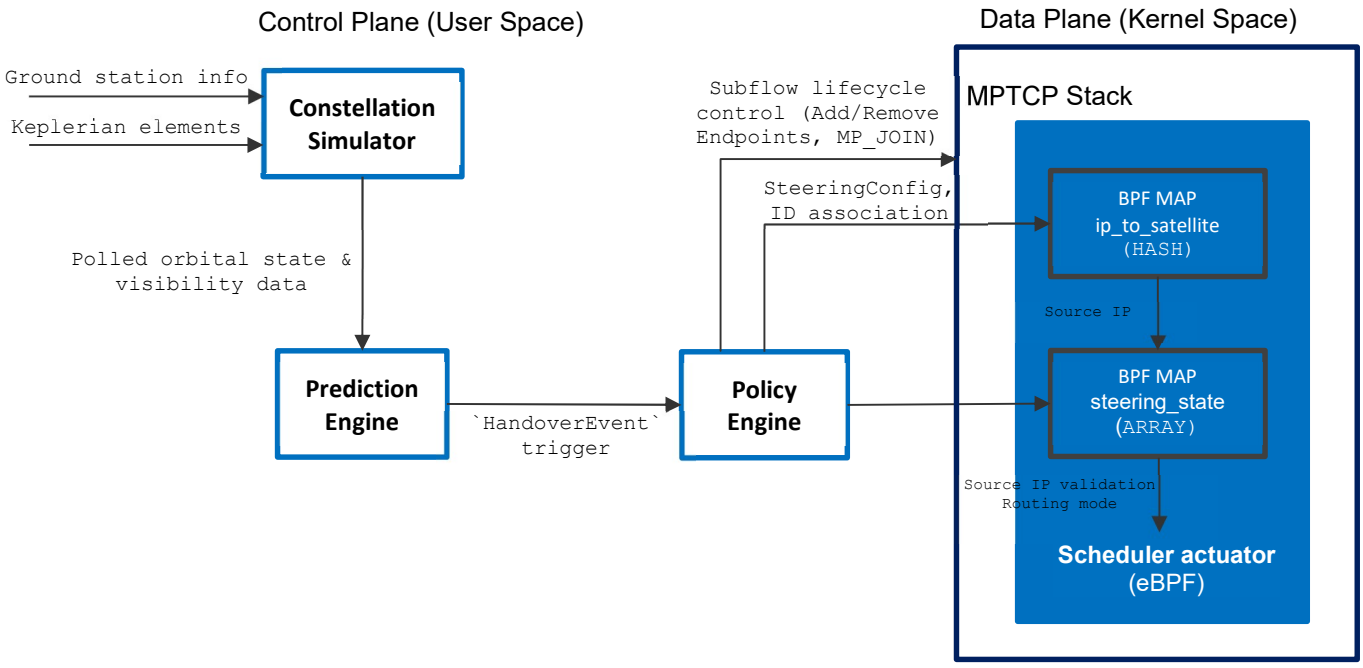

Figure 3-15 Conceptual view of the MPTCP handover predictive framework

3.3.1.1 Constellation Simulator

We developed the Constellation Simulator component as a Python-based orbital mechanics simulator that tracks satellite positions, elevations, and beam steering angles relative to a configured ground station. This module models deterministic LEO orbits with zero propagation error using the orbit-predictor [81] library and its KeplerianPredictor module.

Rather than relying simply only on geometric visibility (e.g., solely on angle of elevation), This component Constellation Simulator operates on beam steering geometry, computing the angle from satellite nadir to the ground station, which drives realistic handover frequencies. The simulator identifies the optimal serving satellite as the one with minimum steering angle (closest to nadir) and returns visible satellite lists filtered by dual constraints (steering within `max_beam_steering_angle`, and elevation above `min_elevation`), sorted by steering angle. A SatelliteLinkModel sub-component maps these geometric parameters to physical link characteristics (delay, bandwidth, loss), which are forwarded to a specific DirectLinkUpdater subcomponent which oversees dynamically shaping the satellite link for better realism. (The

© 2023-2025 5G-STARDUST

shaping model is largely configurable. We chose a simple throughput-based shaping depending on several simple parameters like `beam_steering_angle` and `elevation_angle`). Finally, another subcomponent named TimeProvider was developed with the objective to decouple time queries from system clock, ensuring fully reproducible orbital geometry across all test iterations. As the sole source of orbital truth, Constellation Simulator provides the foundation for the Prediction Engine's handover forecasting. In effect, the predictive framework, thanks to this mechanism, can replay any scenario set at a specific date in the past or the future, with a credibly reproducible constellation simulation.

3.3.1.2 Prediction Engine

Prediction Engine is a Python monitoring service that continuously tracks the serving satellite's beam-steering angle to predict handover event before it occurs by tracking steering limit. Its core loop polls Constellation Simulator at a configurable interval and maintains a sliding window of (timestamp, angle) samples.

HO prediction relies on two thresholds:

- A configurable trigger threshold that arms handover preparation.
- And a lower reset threshold derived by subtracting a fixed hysteresis margin, which prevents oscillation alerts near the boundary.

When armed, the engine fits the rate model adaptively, linear regression in steady-state conditions or a quadratic model when acceleration is detected with sufficient fit quality, then refines the predicted limit-crossing time via binary search on Constellation Simulator future queries, converging to sub-second precision despite non-linear steering dynamics. The target satellite is selected as the candidate with the smallest steering angle at the predicted handover time, queried via a specific method `GetAllVisibleSatellites()`, ensuring both beam and elevation constraints are satisfied while excluding the current serving satellite.

A `handover_in_progress` flag prevents duplicate alerts for the same event. Upon detection, the engine forwards a handover event to the Policy Engine component through a registered callback, containing critical fields such as predicted handover time, source and target satellite IDs, handover type ("`BEAM_STEERING_LIMIT`"), uncertainty window, current steering angle, and the predicted maximum steering angle being approached.

3.3.1.3 The Policy Engine

The Policy Engine is a Python orchestration service that runs in all experimental configurations and manages the MPTCP sub flow lifecycle in response to handover event notifications from the Prediction Engine component. Its responsibilities are divided into two layers:

- **Layer 1 Scheduler-agnostic sub flow management:** on receiving a handover event, Policy Engine resolves the source and target satellites IDs to specific IP addresses and interfaces via an internal registry. It immediately establishes a target MPTCP sub flow by bringing up client and server interfaces, registering endpoints, and executing a targeted ping to trigger the kernel MPTCP Path Manager's MP_JOIN handshakes. The authoritative internal state is maintained in a `known_ips` dictionary. System commands requiring namespace isolation are handled by an internal helper. Failed operations are retried via exponential backoff and queued to a background cleanup thread. A dedicated method continuously cross-audit `know_ips` against live kernel endpoints and BPF maps.

© 2023-2025 5G-STARDUST

- **Layer 2 BPF map management:** When the custom Scheduler Actuator (i.e., our handover predictive BPF scheduler) is active, the Policy Engine component drives the three-phase handover transitions by writing into a `steering_state` BPF map at precise timestamps:
  - Phase 1 (`REDUNDANT`, T-30s): mode 1 written to `steering_state` with both satellite IDs,
  - Phase 2 (`GRACEFUL_EXIT`, T-150ms): mode 2 written to `steering_state` with both satellite IDs,
  - Phase 3 (`DEFAULT`, T+5s): source sub flow teardown executes, then mode 0 written to `steering_state`.

When any other scheduler is registered (in our case: BLEST, BPF redundant, BPF round-robin or BPF minRTT), Policy Engine's BPF map writes execute without error but have no effect on packet steering decisions, which are made entirely by the active scheduler's own logic. Sub flow lifecycle operations remain active and identical across all configurations, ensuring comparable MPTCP path topology management throughout experimental runs.

3.3.1.4 The Scheduler Actuator

Operating entirely in the kernel space, we developed this component as a `BPF_STRUCT_OPS` program, written in c, and registered as the predictive MPTCP scheduler. It is invoked by the kernel on every per-connection scheduling decision, and we carefully designed this component to satisfy the BPF verifier's strict constraints on loop complexity and instruction count. The central gating mechanism is the `ip_to_satellite` map. On each invocation, the scheduler reads the sub flow's source IP and performs a map lookup; a miss (e.g., satellite removed by Policy Engine on Loss of Signal (LOS) event) causes sub flow to be skipped regardless of its TCP state. This is the binding point between the Python control plane and the BPF data plane: a sub flow can be TCP-ESTABLISHED long after its satellite has gone into LOS, and the map gate is what prevents data from queuing onto dead links.

Scheduling behaviour is driven by the value of the aforementioned `steering_state` map:

- `DEFAULT`: In our implementation, the scheduler behaves consistently like a minRTT scheduler during this state: it selects the single mapped sub flow with the lowest RTT. The `ip_to_satellite` map gate applies unconditionally: sub flows whose source IP is absent from the map are skipped regardless of their TCP state. If no mapped sub flow exists (e.g., if the constellation is imperfectly designed, and a transient inter-satellite gap exists at a given time), then the scheduler returns -1 and data waits in the meta-socket buffer. There is no fallback to unmapped sub flows.

- `REDUNDANT`: Scheduling is constrained to the source and target satellite IDs in a dedicated structure found in `ip_to_satellite`, up to a compile-time cap (this cap being one of the stringent limits which is set, by principle, by the BPF verifier, so that the BPF scheduler can be allowed to execute in kernel space).

- `GRACEFUL_EXIT`: Same ID-constrained path as `REDUNDANT`, capped at 2. Both source and target remain schedulable. Moreover, the source queue drains naturally as its link degrades toward the steering limit and TCP's congestion window shrinks accordingly. The source IP is not removed from the map until Policy Engine's endpoint teardown, performed in our implementation at T+5s.

© 2023-2025 5G-STARDUST

Furthermore, the Scheduler Actuator's behaviour can be summarized with the following algorithm:

```
For packet p, select subflow based on steering mode m:

    If m is default:

        Return sub flow with lowest RTT.

    If m is pre-handover redundant:

        If source sub flow f_s has window space, send p over f_s.

        If target sub flow f_r  has window space, send p over f_r.

        Return null (packet duplicated).

    If m is graceful exit:

        List available sub flows F = all subflows excluding source.

        If F empty, return null.

        Else return sub flow in F with lowest RTT.
```

3.3.1.5 Deployment scenarios and topologies

To validate the predictive handover framework, as previously described in section 3.3.1, and to accurately quantify the performance of the BPF-based scheduler, we designed different deployment topologies for the system, which we called Scenario A, B and C. Each scenario isolates specific aspects of the handover orchestration pipeline while maintaining reproducible orbital geometry and network conditions.

Common Principles:

All deployment scenarios A, B and C share the following principles, regardless of the underlying link implementation or fleet scale:

- **Symmetric Addressing Verification**: Each satellite link is assigned to a completely isolated /24 subnet (e.g., 10.N.x.0/24), enforcing strict physical path separation between client and server. This prevents routing ambiguities and ensures that each sub flow operates over a distinct network path.

- **Stable Server Endpoint**: A satellite-independent server endpoint (10.255.255.2/32) is bound to a loopback interface on the server host. All satellite sub flows execute their MP_JOIN handshakes against this loopback address rather than a transient per-satellite IP. This design guarantees that the overarching MPTCP meta-socket survives the physical UP/DOWN cycles of individual satellite links, maintaining connection continuity throughout handover transitions.

- **Dark Mode State Management:** Inactive satellite links are administratively disabled (interfaces marked DOWN) and dynamically shaped via TC netem to block traffic. This physically prevents the kernel from attempting to queue data on non-visible paths.

- **BPF map gate:** the `ip_to_satellite` BPF map, as mentioned in section 3.3.1, is the binding point between control plane and data plane, regardless of topology.

© 2023-2025 5G-STARDUST

Scenario A: Scalable Constellation

This main scenario establishes the baseline dense-topology architecture, modelling a 576-satellite Walker-Delta constellation (24 orbital planes with 24 satellites per plane at an altitude of ~600km). Operationally, 0 to 2 satellites are simultaneously visible within the defined beam steering and elevation constraints, though the BPF scheduler is compiled to theoretically support up to 8 active satellites. The remaining constellation (i.e., 568–576 satellites under our assumptions) remains continuously in Dark Mode. Brief zero-visibility gaps are accommodated by the BPF scheduler returning -1, safely buffering data in the MPTCP meta-socket.

To manage this 576-satellite scale, Scenario A needed several specific infrastructure mechanisms:

- **Scalability:**
    - **Process split:** We purposefully separated the execution of non-real time and real-time components. As a result, Constellation Simulator and Prediction Engine components run in a dedicated child process. We justify this implementation choice by observing that, with 576 satellites, O(N) orbital propagation and Python's Global Interpreter Lock (GIL) principles [82] resulted in 600+ ms jitter in main process, breaking the intended Policy Engine's phase-transition timing precision. With that process separation, we fully addressed the detrimental effect of this computational jitter and made sure that the Policy Engine actions are unaffected by this jitter.
    - **Path Manager & Concurrency:** The configuration utilizes Kernel Path Manager with the `subflow` flag exclusively. The `fullmesh` is prohibited at this scale. Endpoint management is therefore manual and explicitly executed during Acquisition of Signal (AOS) / LOS events.
- **Per-satellite policy routing:** Each satellite has dedicated IP policy routing table with single scope link route restricting routing to its own /24 subnet. Per-source-IP IP rule entries select correct table at send time.
- **Sub flow model:** One satellite serves as bootstrap master, i.e., this satellite will anchor the initial MPTCP connection. It is important to note that this master TCP socket is kept alive during the whole execution of the predictive framework, to preserve the MPTCP connection. In effect, destroying the initial sub flow would risk incurring MPTCP connection teardown.
- **Concurrency discipline:** It is worth noting that all Mininet calls are serialized under a single 'shell lock'. This mechanism prevents inter-locking effects from concurrent threads, which would be likely to occur with the considered satellite number.

Scenario B Direct Access

This scenario models Direct-to-Cell (D2C) or Direct-to-Device (D2D) access topologies using a significantly reduced fleet of 2 to 9 satellites in close orbital proximity. It is similar to Scenario A, but it greatly simplifies the implementation by assuming the removal of all scalability mechanisms. This specific scenario was not implemented during this study.

Scenario C Indirect Access

© 2023-2025 5G-STARDUST

This scenario evaluates the traditional bent-pipe or ground relay architecture, modelling a gateway-based indirect satellite access topology using 2 to 9 satellites. The ground terminal connects to a ground gateway, which subsequently provides access to the satellite link.

This scenario employs a Switch-based structure. Hosts are connected by independent paths, each routed through a dedicated Open vSwitch (i.e., Mininet's OVSBridge). Each switch logically represents the complete satellite infrastructure for a single link (ground gateway to satellite to ground gateway).

3.3.2 Hand Over Prediction Results

3.3.2.1 Handover predictive framework validation

The validation of the predictive handover framework was performed under Scenario A conditions: a full 576-satellite Walker-Delta constellation with the built-in BLEST scheduler as the active MPTCP packet scheduler. The objective of this validation was twofold:

- First, to confirm that the framework's non-real-time components (i.e., Constellation Simulator, Prediction Engine and Policy Engine) correctly orchestrate handover detection, sub flow lifecycle management, and dynamic link shaping at full constellation scale.

- Second, to establish BLEST as a first reference baseline against which the custom predictive scheduler's performance will ultimately be measured (the second reference baseline being the BPF minRTT scheduler we developed for that purpose).

We tested the framework under many parameter values, in particular, the virtual time at which we ran the tests (to isolate different constellation configurations and different handover patterns), the test duration, and the user traffic patterns. In that context, we use the following Figure 3-16, as a representative illustration of the multiple runs we performed under various conditions. This figure, produced directly by the framework's instrumentation pipeline, shows a 300-second run with a custom 2 Mbps paced MPTCP traffic flow over the emulated constellation, with the ground station arbitrarily located in Paris. In this typical graphical output, the top panel shows the dynamically shaped link capacity for each visible satellite, the middle panel shows the per-sub flow throughput as measured by BLEST, and the bottom panel shows the round-trip time of the active sub flow. Handover predictions are overlaid as vertical markers (dashed grey for the alert time, solid red for the predicted handover time, with red shading covering the [T-30s, T+5s] transition window).

© 2023-2025 5G-STARDUST

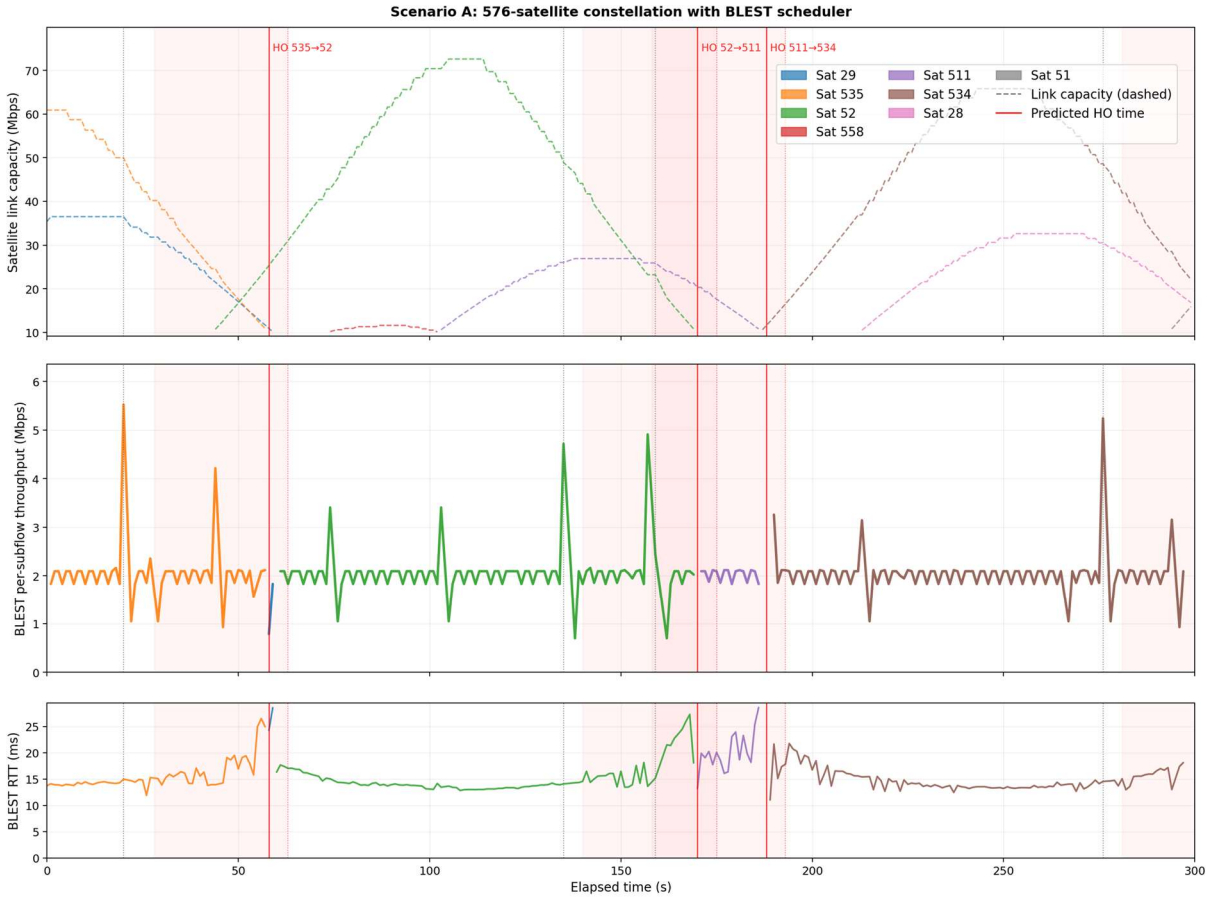

Figure 3-16 Representative composite output of the handover predictive framework with BLEST

Over the exemplary 300-second run illustrated by Figure 3-16, the Prediction Engine autonomously detects four handover events (3 fully displayed), involving satellites from different orbital planes. The framework correctly anticipates each transition; with handover alerts issued at predetermined 30 to 35 seconds ahead of the predicted beam steering limit crossing. The satellite chain progresses through five active carriers (Sat 535 → Sat 29 → Sat 52 → Sat 511 → Sat 534), with BLEST maintaining a steady throughput of approximately 2 Mbps, consistent with the chosen application pacing rate and well within the available link capacity, which ranged from 10 to 73 Mbps depending on each satellite's orbital geometry.

Several observations from this illustrative run are worth noting:

- The top panel of Figure 3-16 illustrates how the framework dynamically shapes each satellite link as a function of the satellite's current beam steering angle and elevation. Link capacity is not static: it follows the satellite's pass geometry, rising as the satellite approaches its closest point to Nadir and declining as it moves toward its beam steering limit. This dynamic shaping has a measurable effect on BLEST's scheduling behaviour. In the bottom panel, the RTT of the active sub flow exhibits a characteristic U-shaped profile within each satellite pass: RTT reaches its minimum when link capacity is at its peak (satellite near nadir) and increases as the link degrades toward the edges of the pass. This coupling between shaped link capacity and observed RTT is a step toward more realistic emulation of LEO satellite conditions, and it ensures that multipath schedulers are exercised in an environment where path characteristics evolve continuously rather than remaining artificially constant. Naturally, the level of realism on the MPTCP stack depends on the model defined in our SatelliteLinkModel

© 2023-2025 5G-STARDUST

subcomponent of the Constellation Simulator. This level of abstraction fits our purpose for this study, but could easily be adapted towards more simplicity, or more realism to trigger some specific behaviour (or limitations) of existing schedulers.

- The handover transitions visible in Figure 3-16 reveal that the overlap between successive satellites is not always perfectly aligned. In an ideal configuration, Sat 52 would have handed over directly to Sat 534, which had sufficient coverage duration to carry traffic for the remainder of the run. Instead, the orbital parameters of this particular configuration required two intermediate handovers (Sat 52 → Sat 511, then Sat 511 → 534), with Sat 511 carrying traffic for only 17 seconds. This observation demonstrates that the framework can serve as a diagnostic tool for constellation design: by replaying different orbital configurations and epoch times, one can identify parameter settings that produce cleaner handover chains with longer overlap windows, or conversely, characterize worst-case scenarios where rapid successive handovers are unavoidable.

- BLEST's scheduling decisions, while robust, are not optimal from a handover-aware perspective. Immediately after the first handover from Sat 535, BLEST briefly selected Sat 29 as the active carrier (at about T=54 to 59s) based on its instantaneous RTT and congestion window metrics. However, Sat 29 was already approaching its beam steering limit and offered only a few seconds of viable service. The more resilient choice would have been Sat 52, which BLEST eventually selected at T=60s and which remained the active carrier for over 100 seconds. This suboptimal transient selection is inherent to reactive schedulers: they optimize based on current path measurements without knowledge of satellite trajectories or predicted handover times. More broadly, packet losses during handover transitions are observable in the throughput panel as brief dips. These losses are inescapable with a reactive scheduler that has no anticipation of the upcoming link disruption. These are precisely the opportunities that the predictive handover scheduler is designed to exploit: by entering REDUNDANT mode before a predicted handover and maintaining dual-sub flow transmission during the transition window, our custom handover predictive scheduler is designed to eliminate or significantly reduce the packet losses that BLEST incurs during these transitions.

3.3.2.2 BPF Scheduler Characterisation and issues detection

In parallel with Scenario A framework validation, we performed a characterization of BPF MPTCP schedulers, comparing our custom BPF minRTT implementation and the kernel's built-in BPF redundant scheduler (bpf_red) against BLEST as a reference baseline. This investigation was motivated by a critical requirement of the predictive framework. In effect, during REDUNDANT and GRACEFUL_EXIT phases, the custom scheduler must be able to steer packets simultaneously across two sub flows. This is a capability that depends on the correct functioning of the BPF interface with the kernel MPTCP stack. Through systematic testing across multiple traffic patterns and transfer durations, we identified three distinct failure modes that are exclusive to BPF schedulers. These failure modes are absent with BLEST under all tested conditions. They do not originate from bugs in any specific scheduler's logic, but rather from interactions between the BPF framework and the kernel's MPTCP protocol machinery.

- Failure Mode #1: Unsent data amplification. Under paced application traffic, BPF schedulers accumulate large volumes of kernel-generated queued data that far exceed the application's actual writes. This amplification is unbounded and grows linearly with time. BLEST is entirely immune to this failure mode, exhibiting zero-queued unsent data at all measurement points.

© 2023-2025 5G-STARDUST

- Failure Mode #2: Sub flow freeze. A secondary sub flow establishes successfully and transfers data for a brief initial period, then all TCP state metrics freeze at constant values indefinitely. The sub flow is not torn down, but becomes inert, and the scheduler falls back to single-path operation.

- Failure Mode #3: Secondary sub flow establishment failure. The MPTCP secondary sub flow either fails to establish or dies immediately after establishment. This was consistently observed with BPF redundant (bpf_red) across all tested configurations.

Table 3-5 summarizes the performance comparison across three representative test configurations. The test topology consisted of two direct paths with asymmetric characteristics (Path 1: 5 ms delay and 100 Mbps; Path 2: 40 ms delay and 50 Mbps).

Table 3-5 - Performance comparison of MPTCP schedulers across traffic patterns.

	BLEST	**BPF minRTT**	**BPF redundant**
50 MB transfer			
Goodput[4]	118.8 Mbps	84.3 Mbps	86.4 Mbps
Path split	2.51:1	2.31:1	Single-path
Peak unsent queue	55 KB	3.42 MB	183 KB
Failure mode	None	None	#3 (MP_JOIN failure)
500 MB transfer			
Goodput	132.8 Mbps	93.4 Mbps	88.3 Mbps
Path split	1.98:1	Single-path	Single-path
Peak unsent queue	1.4 KB	1.91 MB	45.6 KB
Failure mode	None	#2 (subflow freeze)	#3 (MP_JOIN failure)
10s traffic paced at 10 Mbps			
Goodput	9.9 Mbps	1.9 Mbps	1.0 Mbps
Path split	Single-path	Degraded	Single-path

[4] Note that goodput is measured at NIC level for bulk transfers, at application level for paced traffic.

© 2023-2025 5G-STARDUST

Peak unsent queue	0 bytes	82 MB	103 MB
Failure mode	None	#1 and #2	#1 and #3

The most critical issue for the predictive handover framework is Failure Mode #3. Our custom predictive scheduler relies on a functioning redundant scheduling mechanism during the REDUNDANT and GRACEFUL_EXIT handover phases. The BPF redundant scheduler's systematic inability to maintain a working secondary sub flow means that this dual-sub flow steering cannot be exercised under current kernel conditions. In addition, with failure Modes #1 and #2, even when dual-sub flow operation is briefly achieved, the resulting queue amplification and sub flow freezing degrade performance far below what BLEST achieves natively.

Consequently, while the predictive handover framework is functionally complete and its orchestration pipeline has been validated at full constellation scale with BLEST, the comparative performance evaluation between the predictive scheduler and reference schedulers could not be carried out during this study's lifetime. The observed BPF limitations represent a kernel-level constraint that is outside the scope of the framework's design and would require upstream kernel modifications to resolve.

3.3.3 Next steps of the HO prediction framework

The BPF limitations identified in the previous section motivate a future development shift of the predictive handover framework toward MPQUIC as the transport layer. This transition is planned as the primary short-term development direction, while longer-term monitoring of upstream kernel fixes for the MPTCP development branch remains an active parallel track.

A key advantage of this pivot is that the framework's modular architecture limits the scope of the required modifications. In effect, the Constellation Simulator, Prediction Engine, and the network-agnostic sub flow lifecycle logic within the Policy Engine are transport-independent by design. These components are expected to require only limited adaptation, primarily at their interfaces with the transport-specific layer. The components requiring substantial revision are the BPF map management layer of the Policy Engine and the Scheduler Actuator itself, which will be replaced by a MPQUIC scheduler implementation.

This rewrite will benefit from a mature and active MPQUIC research and experimental ecosystem. Several well-documented open-source MPQUIC implementations exist, providing both reference architectures and reusable building blocks for scheduler development. In contrast to the BPF approach, MPQUIC schedulers execute entirely in user space. This architectural difference eliminates the class of kernel-level interaction issues documented in the previous section. User space execution also provides significantly more flexibility for debugging, instrumentation, and rapid iteration during development.

Within the MPQUIC ecosystem, the LowRTT scheduler [79] provides a natural performance reference point, analogous to the role that BLEST fulfilled in the MPTCP context. LowRTT's single-path, lowest-RTT selection logic is also well suited to serve as the DEFAULT mode of the MPQUIC predictive scheduler, maintaining efficient single-sub flow operation between handover events. For the redundant and graceful exit phases, the MPQUIC multipath extension [80] natively supports per-path congestion control and cross-path retransmission, providing the building blocks for the dual-sub flow steering logic that the predictive framework requires. Unlike the BPF redundant mechanism, which exhibited systematic secondary sub flow establishment failures, the CODE scheduler's user space execution model is expected to avoid these constraints entirely.

© 2023-2025 5G-STARDUST

In addition, we will continue to monitor the MPTCP upstream research branch (mptcp_net-next [45]) for fixes addressing the three BPF failure modes identified in Subsection 3.3.2.2. When these kernel-level issues are resolved in a future kernel release, the MPTCP-based predictive scheduler remains a viable path that can be revisited with the existing BPF codebase.

Finally, the Scenario A validation presented in the Results section provides an immediately usable asset for the MPQUIC development phase. The BLEST validated framework including the 576-satellite constellation, dynamic link shaping pipeline, and instrumented data collection, establishes a concrete performance baseline. Future MPQUIC predictive scheduler runs can be directly compared against this BLEST reference to quantify the benefit of predictive handover management over reactive scheduling, which remains the central objective of this work.

3.4 JOINT MPTCP-MPQUIC CONTROLLER

Following the ongoing work at ATSSS in which both MPTCP and MPQUIC are considered, it is realistic to assume that both TCP and QUIC traffic will coexist on the same IAB-based backhaul as defined by the previous architecture. This coexistence creates a control challenge: even if MPTCP can be tuned to steer TCP traffic across TN/NTN, it does not provide any direct governance over QUIC/UDP traffic, which will compete for the same backhaul resources and may indirectly influence the observed performance of MPTCP.

The activity reported in this section therefore addresses a practical and project-relevant problem: how to ensure predictable and controllable utilization of TN and NTN backhaul capacity when multipath TCP and multipath QUIC traffic coexist.

The activity pursued three operational objectives:

- Characterize baseline behaviour of existing in-kernel MPTCP schedulers under a TN–NTN backhaul with realistic asymmetry and evaluate whether such schedulers can ensure meaningful NTN exploitation in practice.
- Design and implement a joint coordination mechanism that can simultaneously manage (i) MPTCP traffic decisions and (ii) MPQUIC traffic competition on the same interfaces.
- Enable deterministic, operator-driven policy enforcement, i.e., the ability to define *per-link* and *per-protocol* shares (slicing) across TN and NTN resources and maintain these shares despite dynamics on the satellite segment.

It is worth to mention that this activity provides a concrete building block for AI/ML-based control: it delivers a closed-loop telemetry-driven controller that can later be upgraded from rule/threshold adaptation to predictive or learning-based policy optimization.

Thereby, a key challenge in TN–NTN backhaul is that protocols interact indirectly through shared resources. Even if MPTCP is tuned to use both backhaul links, MPQUIC may still absorb a large share of capacity on one interface, affecting what remains available for MPTCP. Therefore, a joint control approach is required to provide stable, operator-defined behaviour across protocol families.

© 2023-2025 5G-STARDUST

3.4.1 Evaluation scenarios

Following the reference scenarios of 5G-STARDUST, for evaluation of the joint MPTCP-MPQUIC controller an IAB-like aggregation node forwarding traffic toward the core via two parallel backhaul links:

- A TN backhaul: relatively stable capacity, low RTT.
- An NTN backhaul (LEO satellite): higher RTT, time-varying capacity, and possible outage intervals aligned with handover/visibility transitions.

Multiple UEs generate traffic mapped to two protocol families:

- TCP/MPTCP flows (representative of legacy TCP services and long-lived sessions).
- QUIC/MPQUIC flows (representative of modern application traffic).

The IAB node becomes the natural control point for:

- MPTCP sub flow scheduling (in-kernel), and
- per-interface, per-protocol slicing (Linux traffic control) to coordinate MPTCP and MPQUIC coexistence.

Consequently, with this activity we extend the concept further by considering not only MPTCP but also MPQUIC, reflecting realistic mixed-protocol environments.

Figure 3-17 illustrates the end-to-end scenario used in this activity. UEs send traffic that is aggregated by an IAB node and forwarded toward the core network across two upstream backhaul segments: TN and NTN (LEO). The figure also highlights the main architectural contribution of this work: instead of relying solely on MPTCP's sub flow scheduler, a joint controller is introduced at the IAB node to enforce protocol-aware slicing across the two interfaces, allowing independent control of TCP (MPTCP) and UDP (MPQUIC) shares on TN and on NTN.

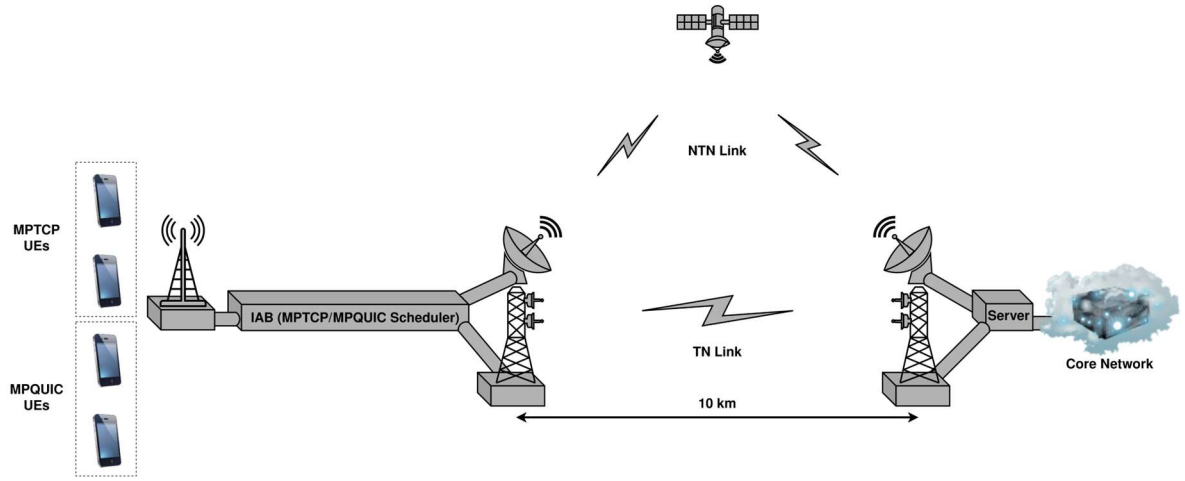

Figure 3-17: System architecture: IAB node aggregating traffic over dual TN/NTN backhaul; joint control across MPTCP and MPQUIC with protocol-aware slicing.

© 2023-2025 5G-STARDUST

3.4.2 Experimental framework and implementation choices

The evaluation is performed using a Mininet-based experimental framework, chosen to preserve realistic protocol behaviour while allowing repeatable control of link conditions. Linux network namespaces emulate endpoints and routers, while per-link shaping parameters emulate TN and NTN characteristics. Mininet is widely used for network experimentation and provides a practical approach to emulate such topologies [87].

MPTCP implementation: Linux in-kernel MPTCP is used, enabling direct evaluation of production-grade schedulers available in the kernel (including eBPF struct_ops schedulers where applicable). This is important from a project standpoint: it demonstrates what can be achieved with standard OS capabilities, without requiring custom protocol stacks.

MPQUIC implementation: MPQUIC is implemented using picoquic stack [86]. This ensures that QUIC behaviour is realistic: congestion control, packetization, acknowledgment behaviour, and multipath support reflect actual stack behaviour rather than a simplified UDP generator. As MPQUIC is not yet standardized this can be considered a prototype implementation but not a full reference.

NTN dynamics: The satellite backhaul is modelled using realistic LEO dynamics in the form of time-varying capacity envelopes and, in relevant scenarios, handover-like interruptions. This is essential because the ability of transport protocols (and controllers) to adapt is highly dependent on whether link properties remain stationary or not.

In the following TN link capacity of 501.7 Mbit/s and NTN capacity ranging between 286.1 Mbit/s and 362.0 Mbit/s are considered in the evaluated scenario

Figure 3-18 establishes a baseline reference for the terrestrial backhaul. The throughput time series is relatively stable, reflecting the lower variability expected from the TN segment. This baseline is used later to interpret whether multipath operation and joint control maintain stable TN utilization while enabling additional capacity contribution from the NTN segment.

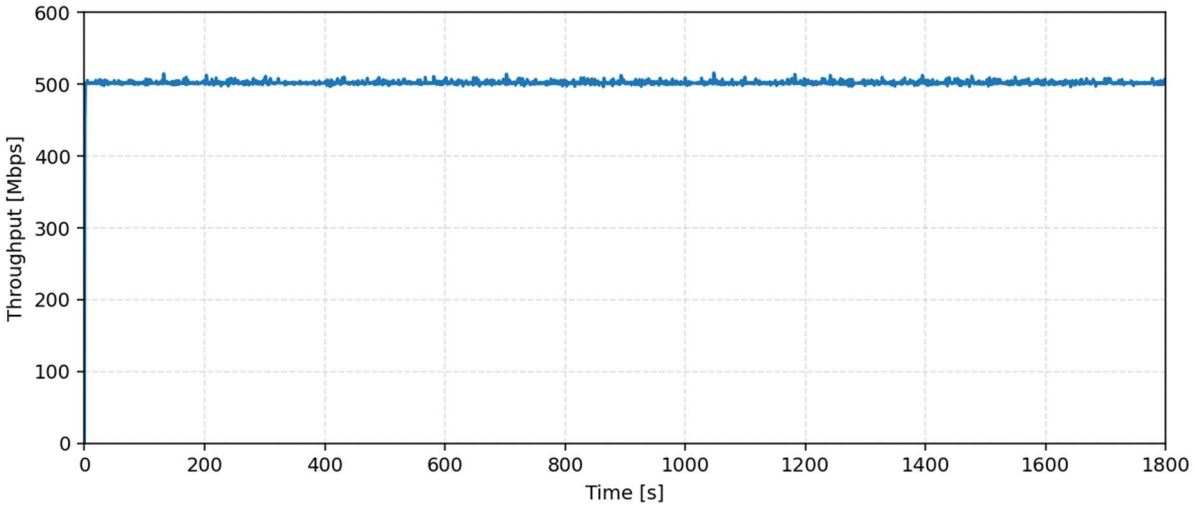

Figure 3-18: Baseline single-path throughput over the TN backhaul

Figure 3-19 provides the corresponding baseline for the satellite backhaul. Unlike TN, the NTN throughput follows a time-varying profile that reflects LEO dynamics, including capacity variations and possible interruption intervals. This figure motivates the need for control

© 2023-2025 5G-STARDUST

mechanisms that remain robust under non-stationary link behaviour: schedulers and controllers must accommodate a path whose available capacity is not constant.

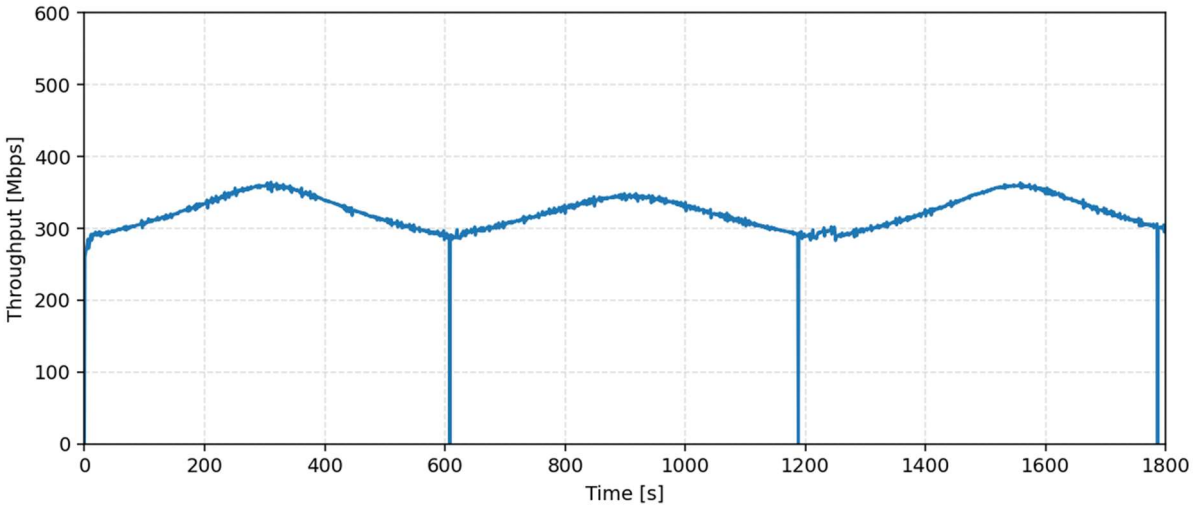

Figure 3-19: Baseline single-path throughput over the NTN (LEO) backhaul

3.4.3 Baseline assessment

This subsection provides a baseline assessment of representative Linux in-kernel MPTCP schedulers in the considered hybrid TN–NTN backhaul scenario. Importantly, the baseline is not a TCP-only experiment: it explicitly considers mixed TCP/QUIC coexistence, where MPTCP flows and MPQUIC flows are active simultaneously and compete for the same TN and NTN backhaul resources. MPQUIC traffic (UDP/QUIC) remains outside the direct control of the MPTCP scheduler, and therefore cross-protocol contention can override scheduler intentions and prevent deterministic, operator-defined TN/NTN sharing.

In Linux, the evaluated schedulers are implemented as eBPF struct_ops schedulers (where applicable) and expose hooks that the MPTCP core invokes at run time. Operationally, all schedulers share the same actuation mechanism: they select one or more candidate sub flows and mark them as eligible for transmission via `mptcp_subflow_set_scheduled()`, after which the MPTCP core transmits on sub flows marked as scheduled. The active policy is selected at runtime through the namespace sysctl `net.mptcp.scheduler`.

In the following, for each scheduler, the performance evaluation shows the joint outcome produced by:

- The scheduler's *MPTCP sub flow selection logic* (which impacts how MPTCP uses TN vs NTN), and
- MPQUIC's concurrent utilization of TN and NTN (which competes for the same bottlenecks but is *not scheduled by MPTCP*).

Therefore, in the interpretation below we explicitly describe both components:

- MPTCP behaviour induced by the scheduler, and
- The observable coexistence effect with MPQUIC on each backhaul link.

© 2023-2025 5G-STARDUST

3.4.3.1 Default (MinRTT)

The in-kernel Default scheduler implements a latency-oriented policy commonly referred to as MinRTT. It prioritizes the sub flow with the lowest estimated RTT whenever that sub flow has immediate sending opportunity (i.e., not blocked by congestion-control dynamics) and shifts transmissions to other sub flows when the preferred one cannot carry more data. This is often summarized as: *"fill the minimum-RTT sub flow first, then move to the next."* The kernel also includes robustness knobs (e.g., staleness tracking via `subflow->stale_count`, with thresholds such as `net.mptcp.stale_loss_cnt`) to de-prioritize persistently non-progressing paths, which is relevant on time-varying links. Since TN typically has much lower RTT than NTN, Default/MinRTT tends to bias MPTCP toward TN, leaving MPQUIC to opportunistically occupy residual capacity on TN and/or exploit NTN capacity independently.

Figure 3-20 shows the throughput evolution on TN and NTN under Default/MinRTT with MPTCP and MPQUIC coexisting. In the TN–NTN setting, MinRTT naturally prioritizes the TN sub flow for MPTCP due to lower RTT, so MPTCP's contribution concentrates on TN. MPQUIC, not controlled by the MPTCP scheduler, continues to utilize TN and/or exploit NTN capacity independently, shaping the residual capacity available to MPTCP on each link.

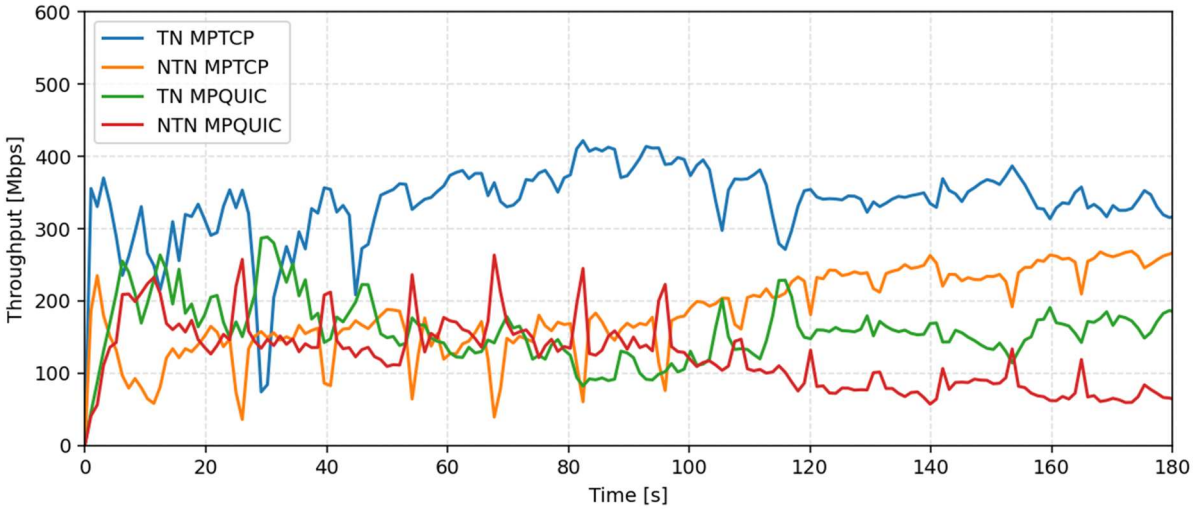

Figure 3-20: Default/MinRTT scheduler: TN/NTN throughput time series for MPTCP and MPQUIC.

Figure 3-21 reinforces the same conclusion statistically: MPTCP throughput distribution is TN-heavy, while MPQUIC distributions reflect its independent allocation across TN/NTN. This figure supports the statement that scheduler-only control cannot ensure deterministic TN/NTN

© 2023-2025 5G-STARDUST

utilization under mixed traffic, because QUIC contention can “mask” or override scheduler intent.

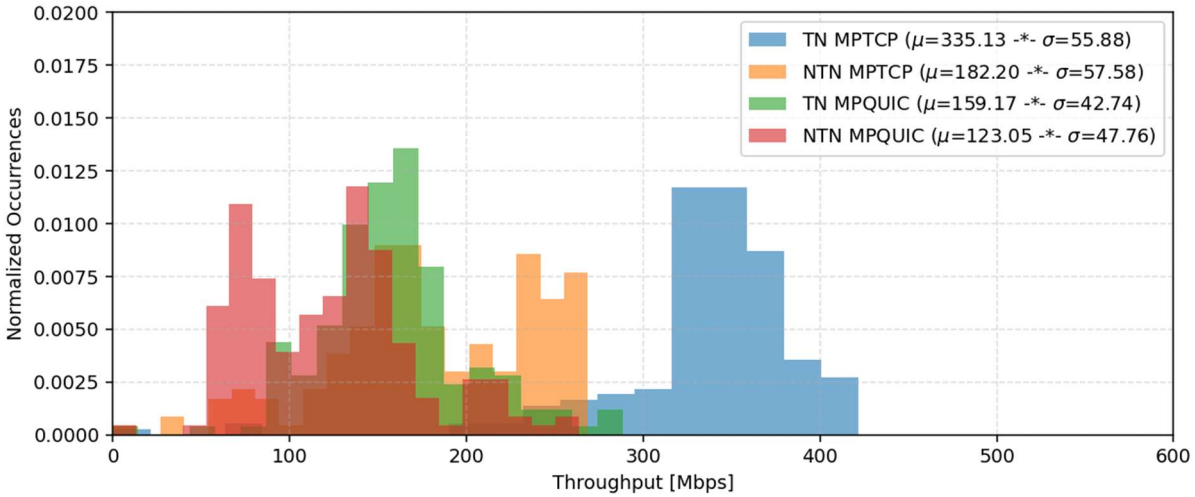

Figure 3-21: Default/MinRTT scheduler: TN/NTN throughput distributions for MPTCP and MPQUIC.

3.4.3.2 Backup

The Backup scheduler implements an explicit active/backup rule. At each scheduling decision, it scans sub flows and selects the first sub flow that is not marked as pure backup, based on the `backup` and `request_bkup` flags. It schedules that sub flow and stops scanning. It does not inspect RTT, congestion, queue occupancy, or memory availability; it is a lightweight preference mechanism where the chosen path depends on kernel-maintained sub flow ordering. Backup tends to keep MPTCP pinned to the active (typically TN) sub flow, making NTN largely irrelevant for TCP aggregation in steady state—while MPQUIC can still use NTN independently and thereby “inherit” most of the satellite share.

Figure 3-22 reports the Backup scheduler outcome under coexistence. Backup semantics keep MPTCP largely on the non-backup (typically TN) sub flow, limiting MPTCP exploitation

© 2023-2025 5G-STARDUST

of NTN in steady state. In contrast, MPQUIC can continue to use NTN, which can result in a scenario where the satellite path is active primarily due to QUIC rather than TCP aggregation.

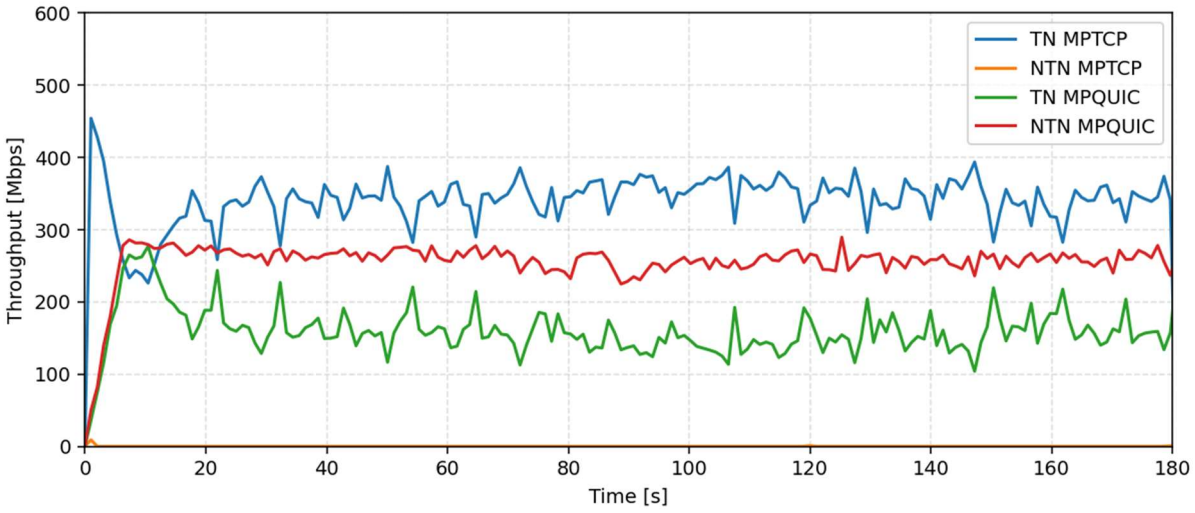

Figure 3-22: Backup scheduler: TN/NTN throughput time series for MPTCP and MPQUIC.

Figure 3-23 shows the associated distributions: MPTCP is concentrated on TN, while MPQUIC retains non-negligible usage of NTN (depending on offered load and MPQUIC path selection). This helps articulate an operational takeaway: Backup for MPTCP does not imply "protecting" NTN resources, it mainly decides that MPTCP will not compete for them, leaving them to other protocols.

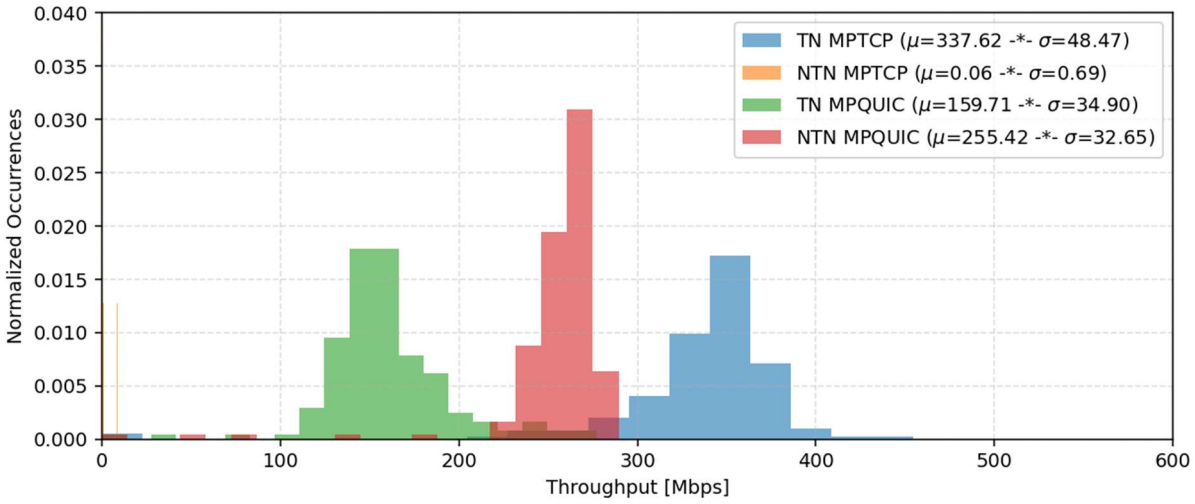

Figure 3-23: Backup scheduler: TN/NTN throughput distributions for MPTCP and MPQUIC.

3.4.3.3 First

The First scheduler enforces a deterministic "always use the primary sub flow" behaviour, effectively emulating single-path preference inside an MPTCP connection. It targets `msk->first` (primary TCP sub flow pointer), converts it into the MPTCP sub flow context, marks it scheduled, and returns. It does not consider backup flags, congestion/pacing, queue or memory signals. This makes it a controlled baseline to isolate the impact of *having multipath available* from *using multipath for scheduling*. MPTCP behaves as "TN-only" (if TN is primary),

© 2023-2025 5G-STARDUST

while MPQUIC continues to compete across links, meaning overall TN/NTN utilization becomes largely dictated by the QUIC side.

Figure 3-24 illustrates that First enforces “primary-only” behaviour for MPTCP (typically TN), effectively acting like single-path TCP within an MPTCP connection. Under coexistence, MPQUIC continues to utilize TN/NTN independently; thus, any observed NTN utilization may mainly reflect QUIC activity rather than multipath TCP aggregation.

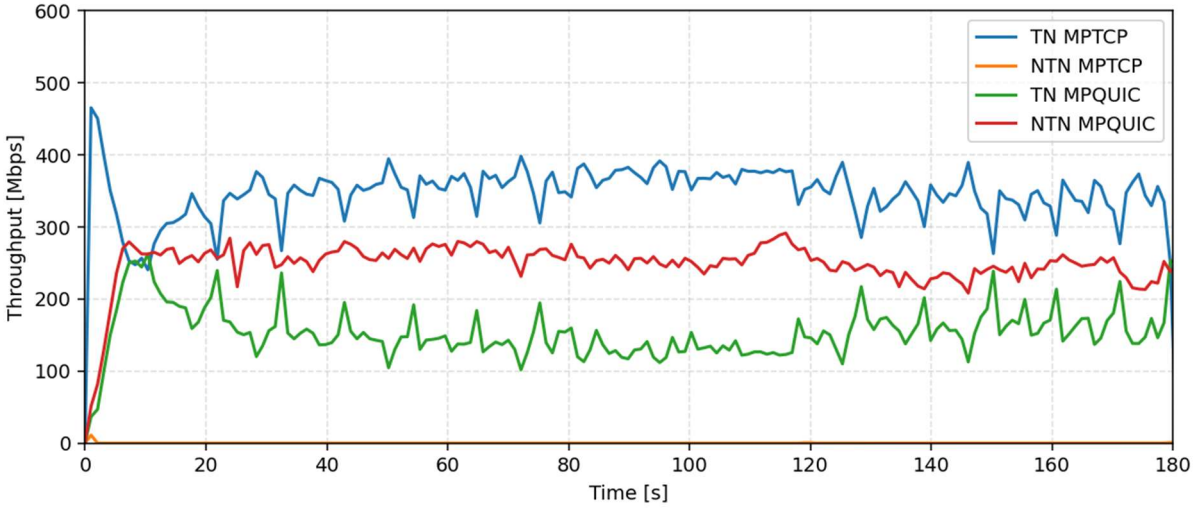

Figure 3-24: First scheduler: TN/NTN throughput time series for MPTCP and MPQUIC

Figure 3 25 confirms this behaviour in distribution form: MPTCP samples are tightly TN-focused, while MPQUIC shows its own distribution across the two links. This figure is useful to underline that First is a “control baseline” to isolate the effect of multipath scheduling decisions from the mere presence of multiple sub flows.

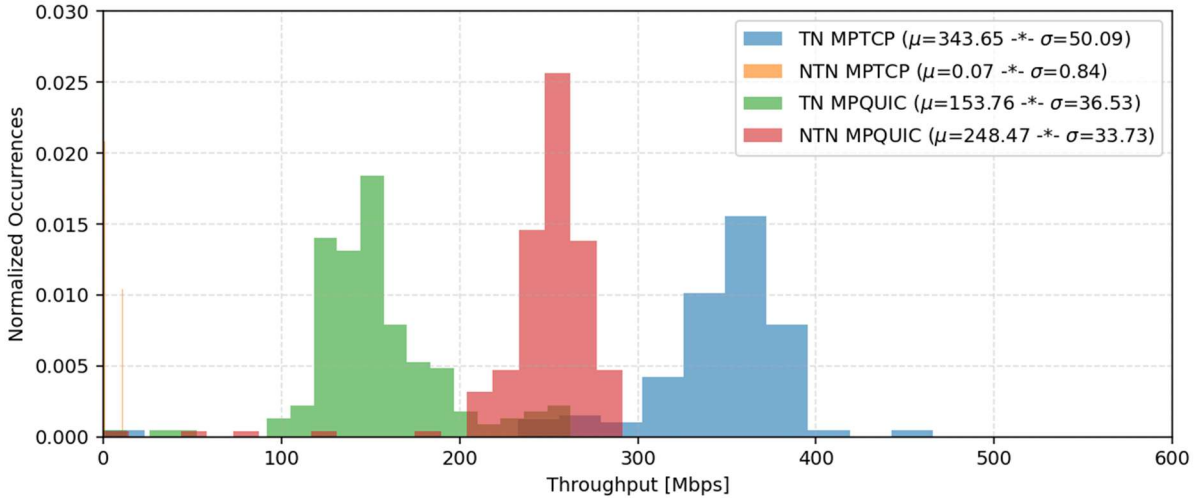

Figure 3-25: First scheduler: TN/NTN throughput distributions for MPTCP and MPQUIC.

© 2023-2025 5G-STARDUST

3.4.3.4 Redundant

The Redundant scheduler implements full replication by scheduling all sub flows at every transmission decision. It iterates over the sub flow list and marks each sub flow as scheduled, with no filtering on backup flags or path conditions. This can improve robustness when paths may intermittently fail, but it trades off bandwidth efficiency and may increase reordering pressure. Redundant TCP transmission can increase resource pressure on TN and/or NTN and therefore reduce residual capacity available for MPQUIC. In coexistence, redundancy does not only affect MPTCP; it changes the contention landscape for QUIC as well.

Figure 3-26 shows redundant scheduling under coexistence. Although the policy schedules all sub flows in each decision considering a replicate-everywhere rule with no filtering on backup flags or link conditions, the MPTCP component is still largely carried by the terrestrial path. TN MPTCP fluctuates around its operating point with recurrent short-term drops, whereas NTN MPTCP throughput stays close to zero for most of the experiment. A different trend is observed for MPQUIC, which is due to the fact that the scheduler is not designed to manage MPQUIC flows. The NTN MPQUIC throughput stays tightly around the nominal satellite operating point, while the TN MPQUIC exhibits moderate variability with occasional bursts.

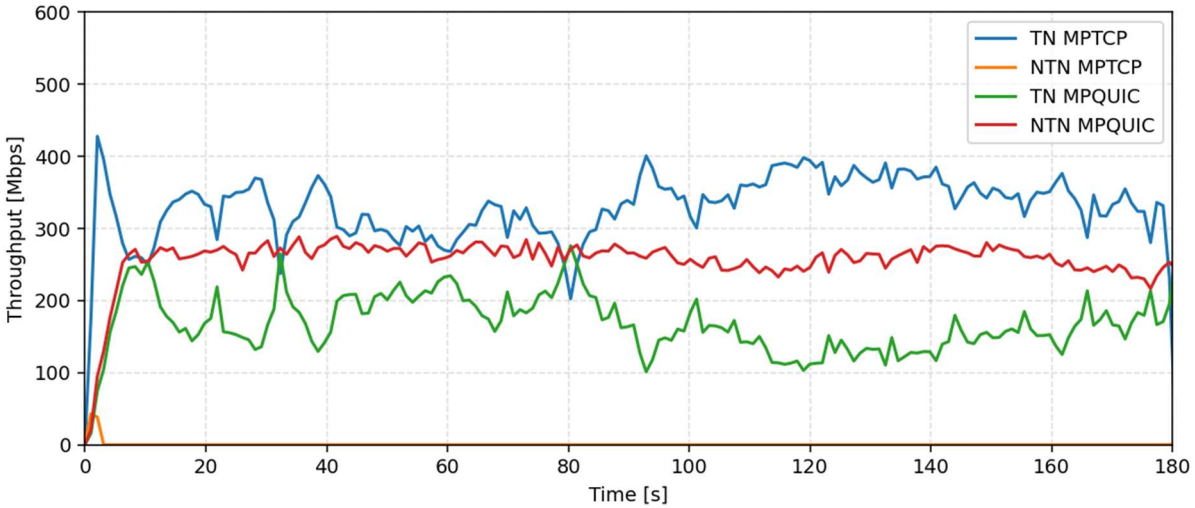

Figure 3-26: Redundant scheduler: TN/NTN throughput time series for MPTCP and MPQUIC.

Figure 3-27 provides the distribution perspective, highlighting that redundancy is not designed for aggregation efficiency. The normalized histogram for NTN MPTCP collapses near zero, while TN MPTCP samples spread over a comparatively wide range, reflecting the variability of the terrestrial backhaul. The TN and NTN MPQUIC histograms suggest a stable operating point in the satellite segment.

© 2023-2025 5G-STARDUST

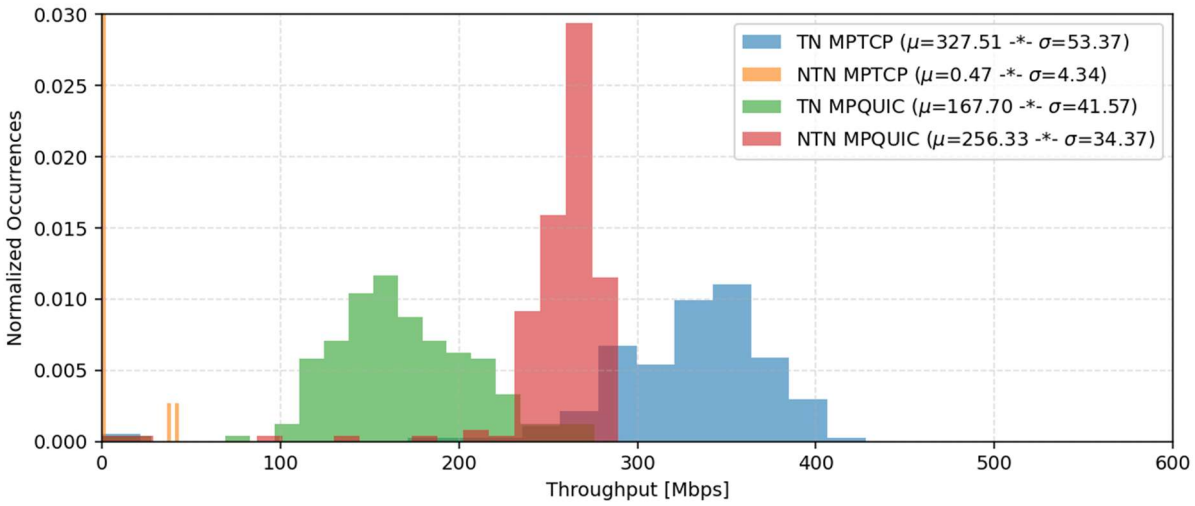

Figure 3-27: Redundant scheduler: TN/NTN throughput distributions for MPTCP and MPQUIC.

3.4.3.5 Round-Robin

The Round-Robin scheduler rotates transmissions across sub flows and maintains explicit per-connection state (e.g., `last_snd` pointer to last-used TCP sub flow) stored in `BPF_MAP_TYPE_SK_STORAGE`. At each decision, it selects the successor sub flow in the kernel ordering (wrapping to the first if needed), schedules it, and updates `last_snd`. It does not treat backup flags specially and does not use congestion or pacing signals; fairness is achieved purely by rotation over the current ordering. Round-Robin expresses an *intent* to balance MPTCP across paths, but in TN–NTN asymmetry the achieved split can still be unstable due to RTT/BDP differences and satellite capacity variability, and it remains sensitive to MPQUIC's concurrent consumption of each interface.

Figure 3-28 presents Round-Robin scheduling with MPQUIC coexistence. Although the scheduler rotates MPTCP transmissions across sub flows, the measured behaviour remains strongly asymmetric across the two backhaul segments. Over the TN link, MPTCP maintains a stable operating region around, with moderate short-term variability and occasional dips that reflect transient transport dynamics under load. However, the NTN MPTCP component is essentially suppressed for most of the experiment, with throughput values concentrated close to zero and only sporadic activity. MPQUIC exhibits sustained utilization on both links, as the scheduler acts exclusively on MPTCP flows. The NTN MPQUIC throughput remains clustered around its nominal operating point, whereas the TN MPQUIC fluctuates around a slightly lower mean with moderate variability and occasional bursts.

© 2023-2025 5G-STARDUST

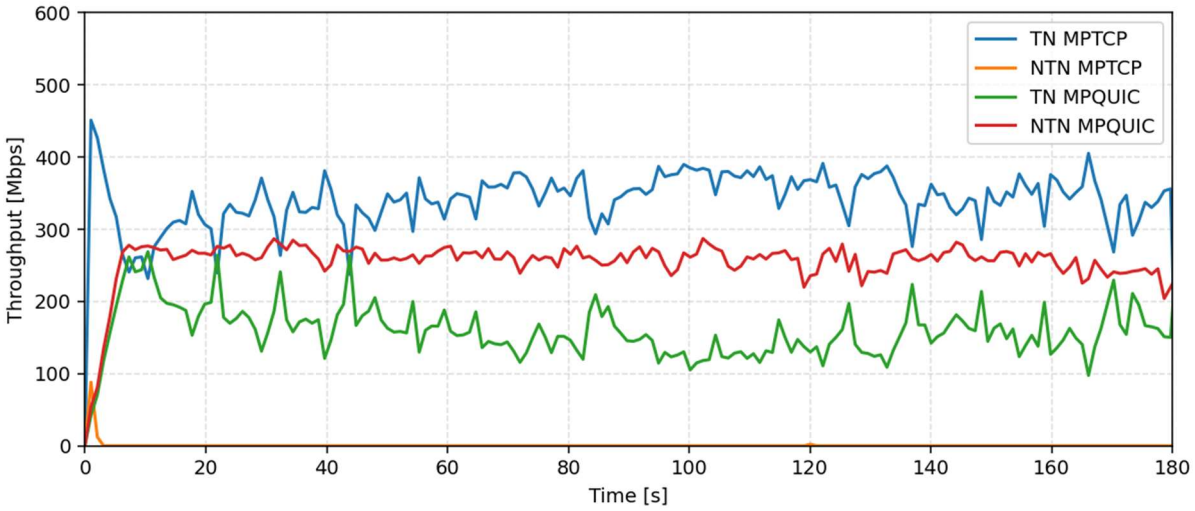

Figure 3-28: Round-Robin scheduler: TN/NTN throughput time series for MPTCP and MPQUIC.

Figure 3-29 further corroborates these observations. The normalized histogram for NTN MPTCP collapses near zero, whereas the TN MPTCP samples concentrate around their mean with moderate dispersion. In contrast, the MPQUIC histograms suggest a relatively stable operating point on the satellite segment.

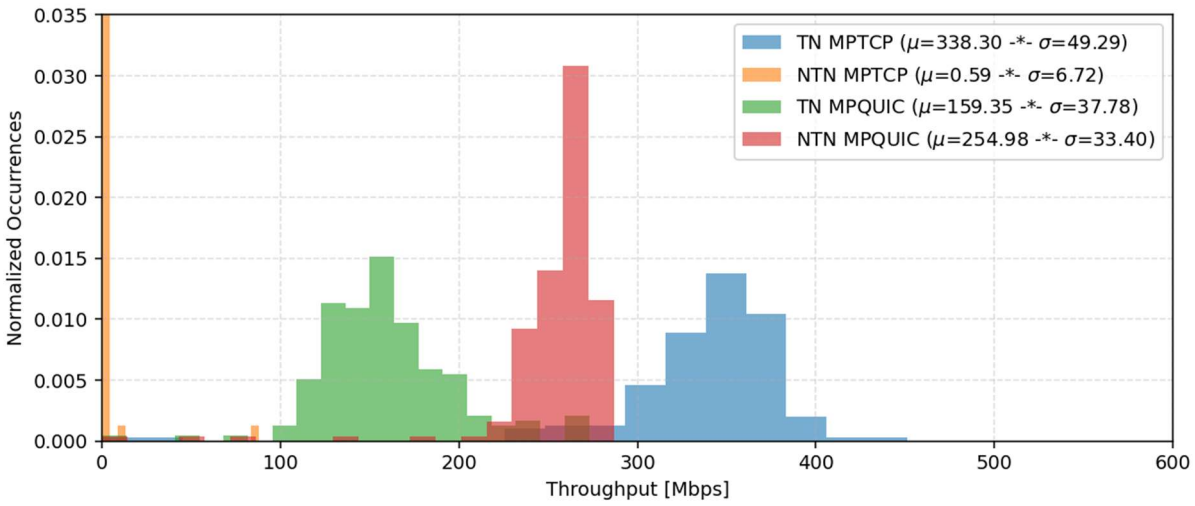

Figure 3-29: Round-Robin scheduler: TN/NTN throughput distributions for MPTCP and MPQUIC.

3.4.3.6 Burst

The Burst scheduler is the most heterogeneity-aware among the provided BPF struct_ops schedulers. It combines a queue/pacing-driven decision for new data with a cautious rule for retransmissions and exports a bounded send quantum to the MPTCP core (`msk->snd_burst`). For new transmissions, it classifies sub flows as active vs backup (based on `backup||request_bkup`), skips inactive sub flows, seeds and smooths pacing estimates, and selects the minimum "linger time" candidate based on queued bytes normalized by pacing rate. It also checks memory availability and bounds burst size. For retransmissions, it avoids sub flows with outstanding TCP data, leverages staleness tracking, and only uses backup paths under specific conditions. Burst can activate the NTN sub flow more than MinRTT in some conditions, but the realized benefit is intertwined with MPQUIC competition on the same

© 2023-2025 5G-STARDUST

satellite bottleneck; therefore, the achieved TN/NTN and TCP/QUIC shares remain an emergent outcome without joint governance.

Figure 3-30 evaluates Burst, which uses queue/pacing signals and exports an explicit send quantum (`snd_burst`) to the MPTCP core. Burst can increase the likelihood of using NTN compared to MinRTT, but in mixed traffic the visible gain depends on how much NTN capacity is simultaneously consumed by MPQUIC. Therefore, the achieved sharing can vary over time and across conditions.

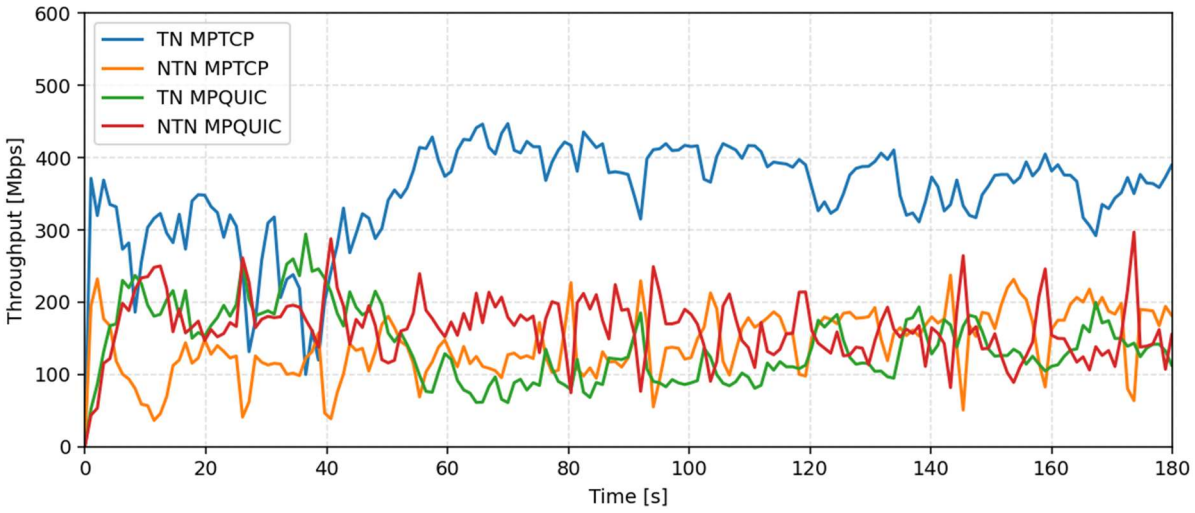

Figure 3-30: Burst scheduler: TN/NTN throughput time series for MPTCP and MPQUIC.

Figure 3-31 confirms the more variable outcomes through distributions. The dispersion reflects both (i) MPTCP's heterogeneity-aware scheduling behaviour and (ii) uncontrolled cross-protocol contention with MPQUIC. This sets up the motivation for the next subsection: a joint controller that replaces emergent sharing with explicit protocol-aware slicing on each interface.

© 2023-2025 5G-STARDUST

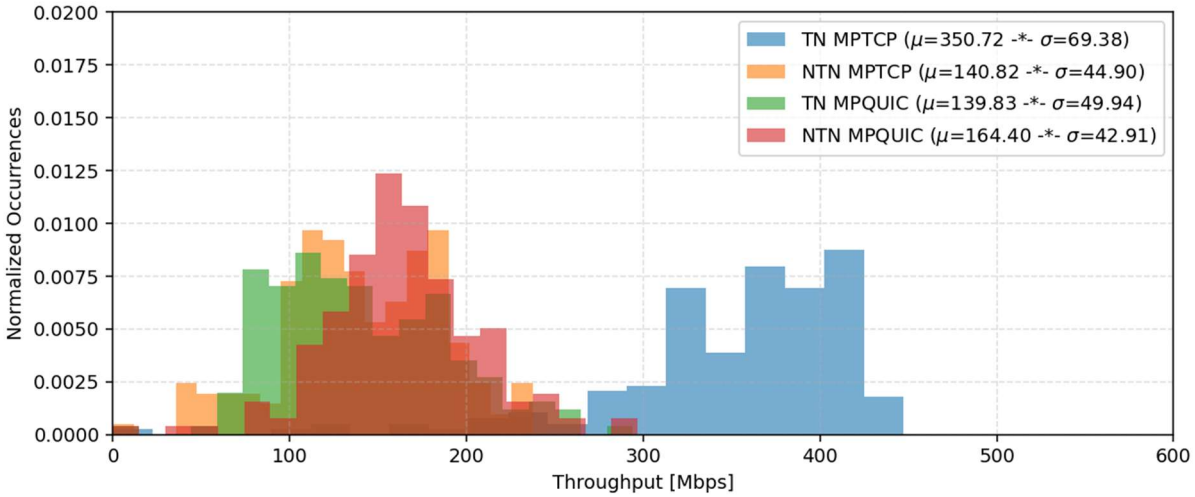

Figure 3-31: Burst scheduler: TN/NTN throughput distributions for MPTCP and MPQUIC.

Baseline conclusion

The baseline study shows that Linux MPTCP schedulers can significantly alter how MPTCP uses TN vs NTN, but the overall observed behaviour in realistic deployments is strongly affected by MPQUIC's concurrent, unmanaged use of the same TN/NTN backhaul resources. Consequently, scheduler selection alone cannot guarantee stable TN/NTN utilization nor stable TCP/QUIC coexistence. This motivates the telemetry-driven, per-link protocol slicing controller introduced in the next subsection.

3.4.4 Telemetry-Driven Per-Link Protocol Slicing Controller

The baseline analysis highlights that relying only on transport-layer scheduling is not sufficient to obtain a predictable and configurable use of a hybrid TN–NTN backhaul when modern traffic mixes are considered. Particularly, Linux MPTCP schedulers can only influence how MPTCP traffic is distributed across TCP sub flows, while MPQUIC traffic follows its own end-to-end decisions and competes for the same bottleneck resources. Consequently, the observed utilization of TN and NTN links, and the effective sharing between TCP- and QUIC-based services, emerge from cross-protocol contention rather than from a controllable operator policy.

To address this limitation, a joint controller is introduced at the aggregation point, i.e., at the node where traffic is collected before entering the dual backhaul. This position is strategic because it is the first place where the system simultaneously "sees" (i) the two physical upstream interfaces (TN and NTN) and (ii) the coexistence of the two protocol families. The resulting architecture enables multi-connectivity to be treated not only as a capability of individual transport stacks, but also as a controllable network function enforcing consistent backhaul usage and coexistence rules.

The controller design follows a simple operational principle: when multiple protocols share the same physical links, stable and reproducible behaviour requires control at the point where resources are actually shared. For this reason, the controller does not attempt to directly modify MPQUIC behaviour or replace MPTCP scheduling logic. Instead, it enforces protocol-aware resource partitioning on each backhaul interface, effectively creating deterministic "budgets" for TCP and UDP traffic on TN and on NTN. This moves the control problem from implicit sub flow choice to explicit resource governance, which is more aligned with SDN-style policy enforcement.

© 2023-2025 5G-STARDUST

Concretely, for each backhaul interface two traffic classes are defined:

- a **TCP class**, which carries the MPTCP traffic (and, more generally, TCP-based services), and
- a **UDP class**, which carries MPQUIC traffic (and, more generally, QUIC/UDP-based services).

For each interface, the operator policy specifies the target share assigned to each class. The policy can be uniform across interfaces (e.g., equal split on TN and NTN) or differentiated (e.g., reserving a larger fraction on NTN for one protocol family, while applying a different split on TN). This design choice is important in TN–NTN systems, where the two links are not equivalent: TN often serves as a stable anchor with low delay, while NTN provides additional capacity but with higher delay and time-varying characteristics.

Enforcement is implemented using standard Linux mechanisms for classification and shaping. Traffic is classified by protocol (TCP vs UDP) and by egress interface (TN vs NTN). Each class is then shaped to a configured rate so that the two classes form a stable partition of the interface capacity. The result is that neither protocol family can exceed its allocated share on a given interface, ensuring that coexistence remains predictable even when offered load changes. This is a key property: the system exposes an explicit control knob (the per-interface split) and provides deterministic enforcement without requiring modifications to transport stacks.

The controller operates as a closed loop driven by lightweight telemetry. At each control interval it observes the throughput achieved by TCP and UDP traffic on TN and on NTN and updates the shaping configuration accordingly. TN is treated as relatively stable and can be governed using a fixed or slowly varying capacity ceiling. NTN is time-varying, and therefore the controller aligns its shaping ceiling to the current capacity budget associated with the satellite segment. In this way, the controller can preserve the intended split even when the satellite capacity changes, and it can restore the nominal operating point after temporary degradations.

Beyond static slicing, the controller supports an adaptive mode intended for pronounced NTN variations. When a capacity reduction is detected or imposed on the NTN link, the controller can apply a rule-based adjustment of the split on NTN to preserve service priorities. When the NTN capacity recovers, the controller returns to the nominal split. This adaptive behaviour is intentionally lightweight, yet it demonstrates a key capability expected from software-defined multi-connectivity control: reacting to non-stationary link conditions while maintaining policy coherence. It also provides a direct pathway to AI/ML enhancements, where the decision logic could be replaced by predictive or learned policies while keeping the same telemetry and enforcement interfaces.

The interaction between the joint controller and the transport stacks is complementary. MPTCP continues to operate with any selected Linux scheduler, and the scheduler still determines which sub flow is preferred for TCP transmissions. MPQUIC continues to operate end-to-end as implemented in picoquic, selecting how to use its available paths. The controller does not alter these internal decisions; rather, it constrains their resource footprint through per-interface slicing, so that cross-protocol competition cannot produce uncontrolled outcomes. This combination preserves protocol correctness and implementation realism while enabling operator-defined behaviour at the network edge.

Overall, the joint controller transforms hybrid TN–NTN multi-connectivity into a governed capability: transport-layer multipath provides path diversity, while interface-level slicing

© 2023-2025 5G-STARDUST

provides deterministic coexistence and a policy handle suitable for SDN integration. The next subsection evaluates this approach quantitatively, including scaling with the number of UEs, equal-share and non-uniform slicing policies, and robustness under NTN capacity variations.

3.4.5 Controller evaluation results

This subsection reports the experimental results obtained with the proposed joint controller and highlights how it enables predictable, policy-driven coexistence of MPTCP (TCP) and MPQUIC (QUIC/UDP) over the hybrid TN–NTN backhaul, even in the presence of NTN time variability. The evaluation follows a progressive validation approach: first, scalability with the number of UEs; second, stable equal-share slicing; third, non-uniform slicing; and finally, robustness to NTN capacity variations with adaptive behaviour.

3.4.5.1 Scaling with the number of UEs

The first set of results evaluates how the system behaves as the number of active UEs increases. The purpose is to verify that: (i) the aggregate system does not collapse into a single-link usage pattern, and (ii) the controller maintains a consistent coexistence between MPTCP and MPQUIC as load grows.

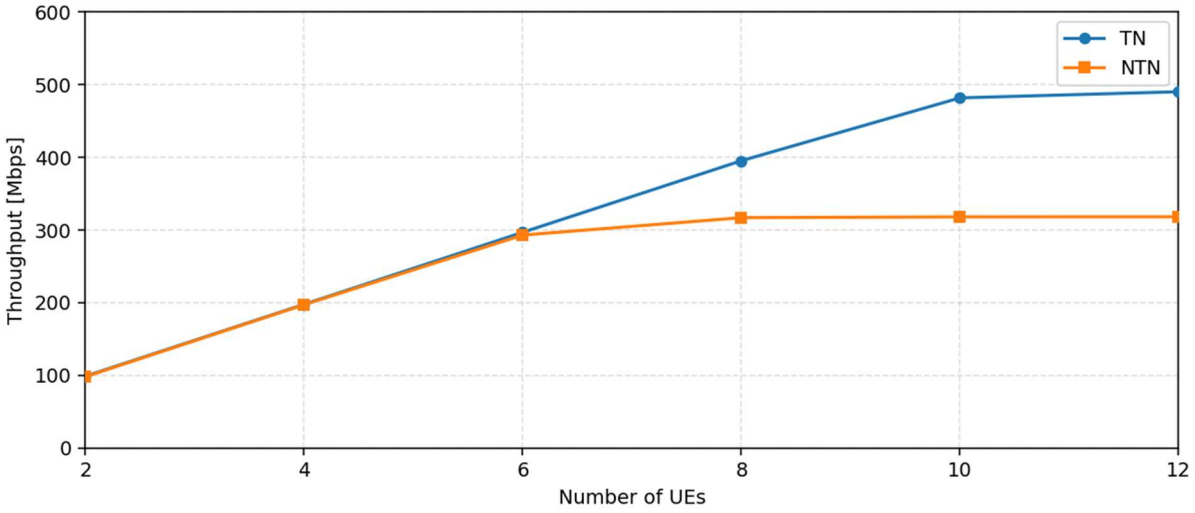

Figure 3-32: Average TN and NTN throughput vs number of UEs (MPTCP–MPQUIC coexistence).

As the number of UEs increases, Figure 3-32 shows how the aggregate throughput is distributed across the two backhaul links. The key interpretation in the TN–NTN setting is whether the NTN link remains an active contributor (when capacity is available), rather than becoming sporadically used only under overload. The figure also supports that the proposed control plane allows the system to scale offered load without losing the ability to exploit both paths, despite satellite variability.

© 2023-2025 5G-STARDUST

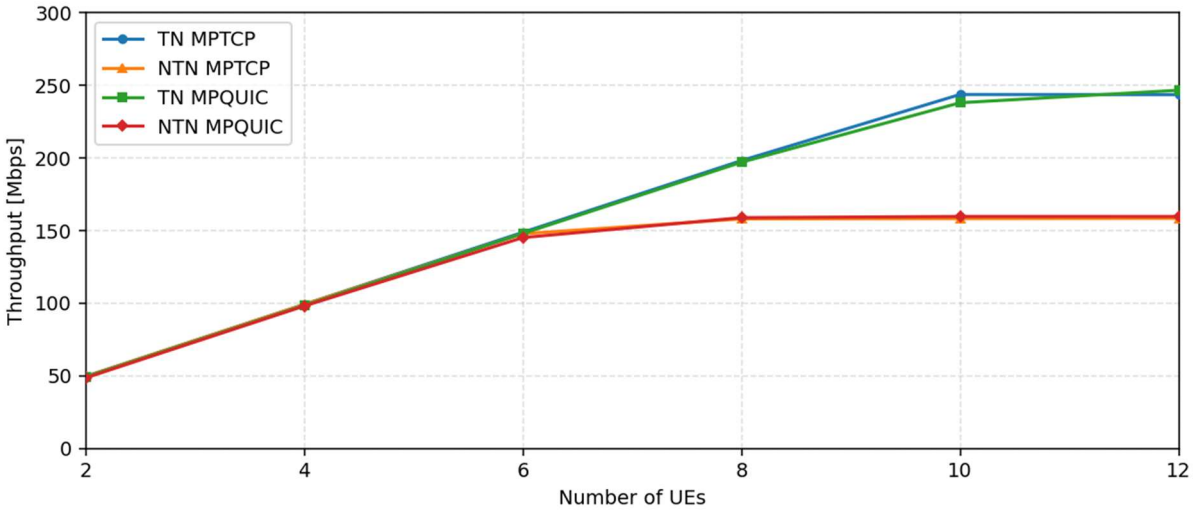

Figure 3-33: Average MPTCP and MPQUIC throughput vs number of UEs.

Figure 3-33 complements the link-centric view with a protocol-centric view. It shows how the total throughput is shared between MPTCP and MPQUIC as the number of UEs increases. This figure is used to demonstrate that coexistence is not left to uncontrolled competition: the joint control approach preserves a predictable balance between TCP-based and QUIC-based traffic families under increasing load.

3.4.5.2 Equal-share slicing

The second set of results validates the controller under a baseline policy where each interface is configured with an equal split between TCP and UDP classes. This is a fundamental proof point: under a fixed policy target, the controller must enforce stable and repeatable behaviour for both protocols on both backhaul links, while keeping each link efficiently utilized.

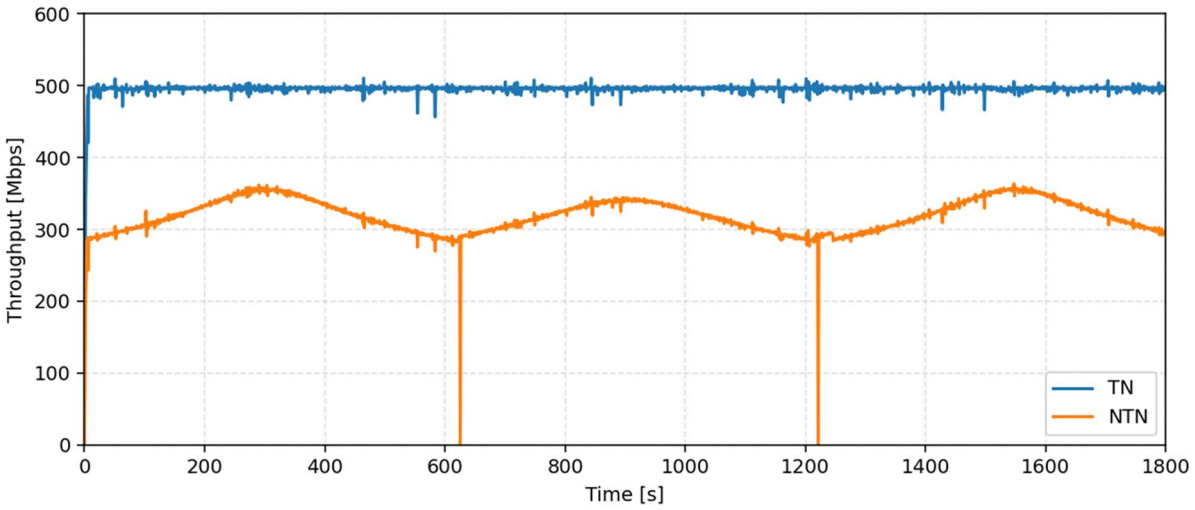

Figure 3-34: Controller enabled: TN/NTN throughput time series under equal-share slicing.

Figure 3-34 reports the throughput evolution on TN and NTN when the controller enforces equal-share slicing. The main expected behaviour is visible at the link level:

- the TN backhaul remains relatively stable around its configured/available rate, and

© 2023-2025 5G-STARDUST

- the NTN backhaul follows the time-varying satellite capacity envelope, including periods of reduction and potential disruptions.

A key message is that the controller enforces policy without wasting capacity: whenever demand exists, both links remain highly utilized, and the NTN contribution tracks the satellite envelope rather than being suppressed by protocol competition.

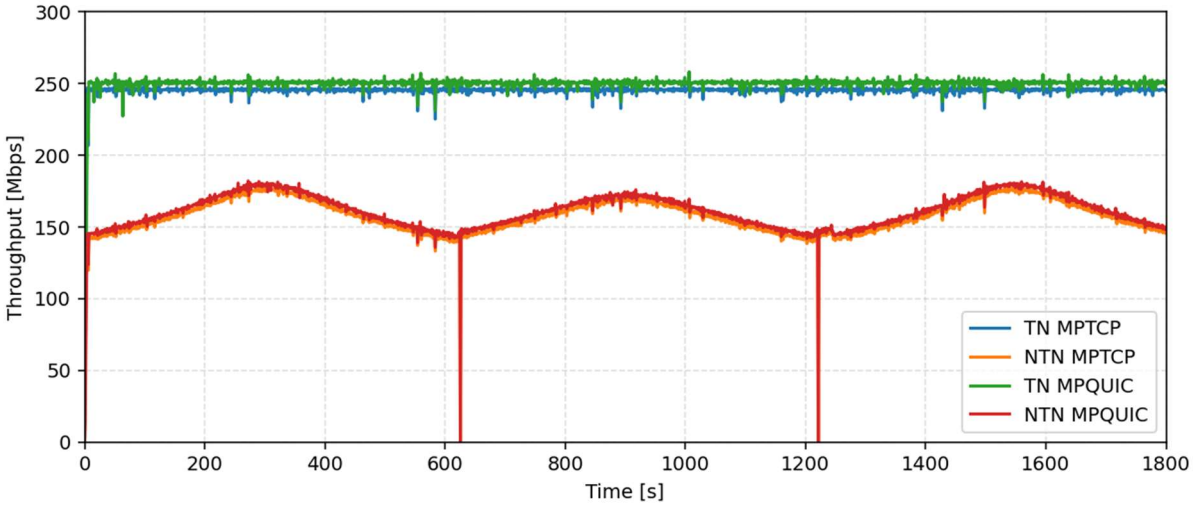

Figure 3-35: Controller enabled: MPTCP/MPQUIC throughput time series under equal-share slicing.

Figure 3-35 provides the protocol-level view of the same experiment. Under equal-share slicing, MPTCP and MPQUIC throughput remain close to the configured split over time. This figure is central because it demonstrates the main functional objective of the controller: the achieved TCP/QUIC coexistence becomes a policy-controlled outcome, rather than an emergent property of contention and scheduler interactions.

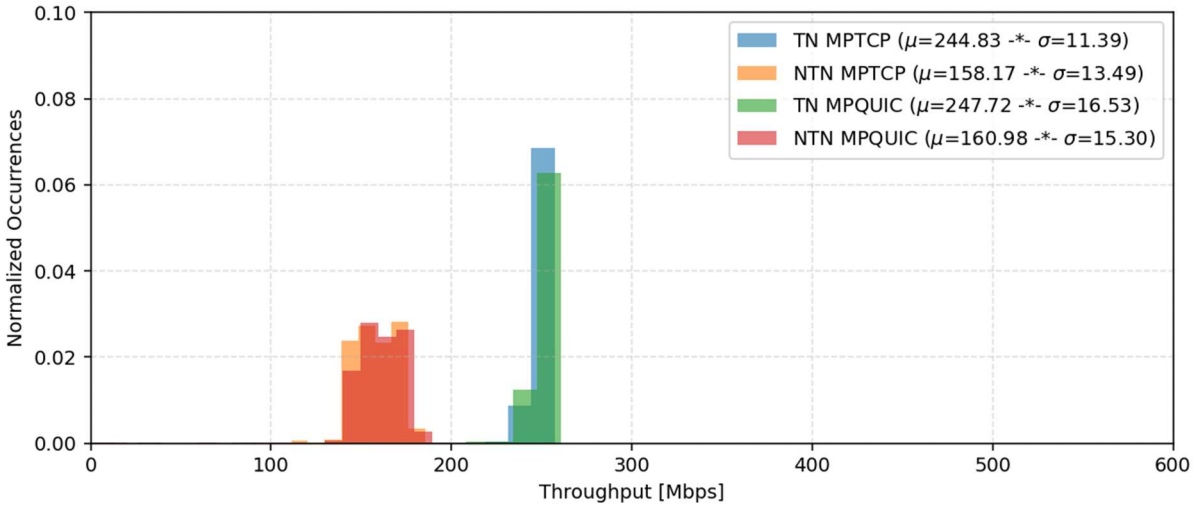

Figure 3-36 Controller enabled: throughput distributions under equal-share slicing (per link and per protocol).

Figure 3-36 confirms stability from a statistical perspective. While Figure 3-34 and Figure 3-35 show instantaneous evolution, the distributions in Figure 3-36 verify that the system repeatedly operates around the desired split rather than only achieving it transiently. In TN–NTN

© 2023-2025 5G-STARDUST

conditions, this is particularly relevant because satellite variability can induce fluctuations: the controller maintains coherent sharing across these variations, producing consistent distributions for both protocols across both links.

3.4.5.3 Non-uniform slicing

Operationally, the TN and NTN links may serve different roles. It is therefore important to demonstrate that the controller supports different TCP/UDP shares on TN and NTN, enabling operator-defined differentiation (e.g., reserving more satellite capacity for a selected protocol family or service type).

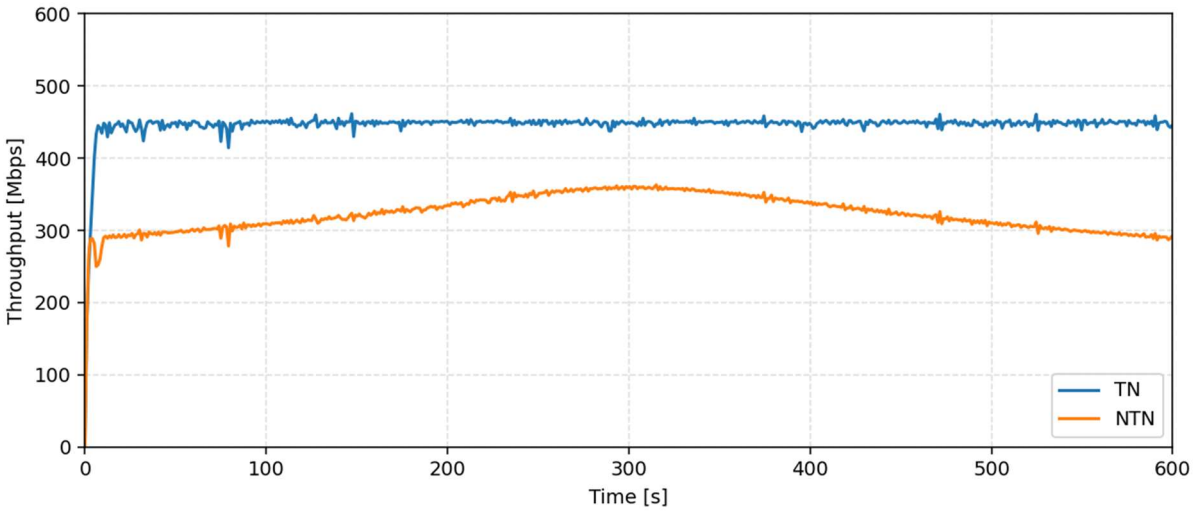

Figure 3-37: Controller enabled: TN/NTN throughput time series under non-uniform slicing policy.

Figure 3-37 shows the link-level throughput when the controller applies a non-uniform protocol share. The key point is that both links remain effectively utilized while the enforced partitioning is applied independently per interface. This confirms that the controller does not assume a single global split; instead, it implements per-interface policy decisions suitable for heterogeneous backhaul.

© 2023-2025 5G-STARDUST

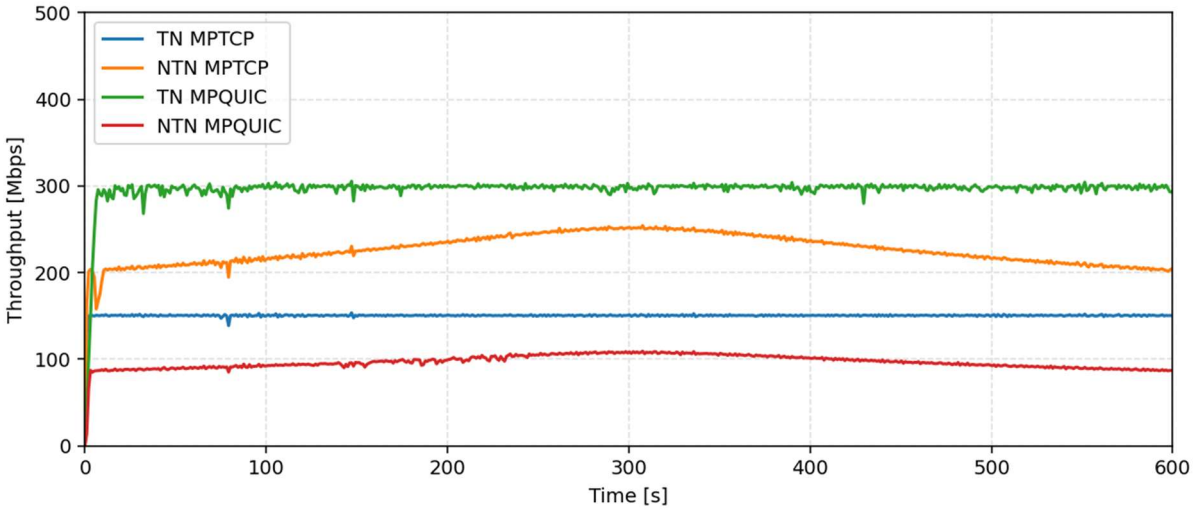

Figure 3-38: Controller enabled: MPTCP/MPQUIC throughput time series under non-uniform slicing policy.

Figure 3-38 reports the same configuration from the protocol perspective and verifies that each protocol tracks its interface-specific allocation. This figure demonstrates that the controller can translate policy intent into measurable outcomes: MPTCP and MPQUIC shares change as configured, and the system remains stable even when the NTN link is time-varying.

3.4.5.4 Robustness to NTN capacity variations

Finally, the evaluation considers a stress condition representative of NTN dynamics: a capacity reduction event (or generally a pronounced change in the NTN envelope). The goal is to show that the controller maintains predictable behaviour during the change and returns to the nominal policy when conditions recover. This is essential for LEO-based backhaul, where transitions can happen frequently.

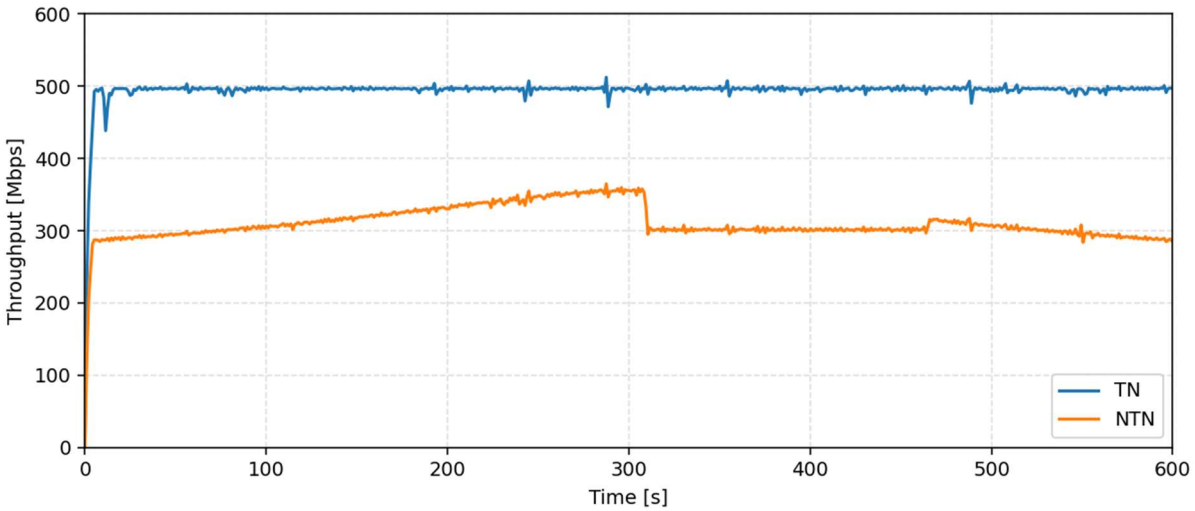

Figure 3-39: Controller enabled: TN/NTN throughput time series under NTN capacity variation.

Figure 3-39 shows the link throughput during the NTN capacity variation. The key result is that TN remains stable and continues to provide an anchor, while NTN throughput follows the degraded envelope. This demonstrates that joint control does not introduce instability during

© 2023-2025 5G-STARDUST

satellite degradation; instead, it continues to enforce a coherent policy under reduced resources.

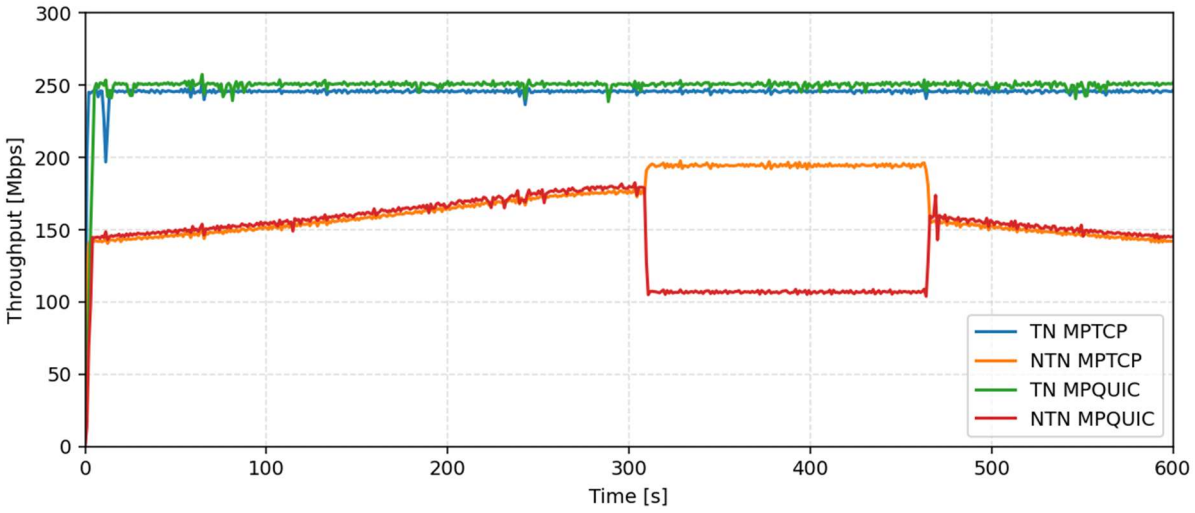

Figure 3-40: Controller enabled: MPTCP/MPQUIC throughput time series under NTN capacity variation (adaptive slicing)

Figure 3-40 provides the protocol-level view and highlights the controller's adaptive response. During the capacity reduction, the controller adjusts the enforced split on NTN according to the selected adaptive rule (e.g., temporarily prioritizing one protocol family on the reduced satellite resource), while maintaining the intended behaviour on TN. When the NTN capacity recovers, the controller restores the nominal split. This figure is used to underline the controller's role as a software-defined control function capable of reacting to non-stationary link conditions while maintaining policy coherence.

3.4.5.5 Summary of evaluation outcomes

Across previous numerical results, the evaluation confirms that the proposed controller:

- scales to increasing traffic demand (UE count) without collapsing into uncontrolled single-link behaviour.
- enforces stable equal-share slicing on both TN and NTN,
- supports non-uniform (differentiated) slicing per interface, and
- remains robust under NTN capacity variations, including an adaptive response that preserves operator intent during satellite degradation.

These results directly motivate the final part of the section on jitter characterization, which complements throughput-based validation with time-variation indicators relevant to QoS/QoE in TN–NTN settings.

3.4.6 Jitter characterization

In addition to throughput and resource-sharing stability, integrated TN–NTN backhaul introduces non-negligible delay variation due to the intrinsic properties of satellite connectivity (higher RTT, capacity fluctuations, scheduling and buffering effects, and handover-related transients). For 5G-STARDUST, characterizing this variability is important because it affects

© 2023-2025 5G-STARDUST

application experience and can interact with transport behaviour (e.g., congestion control dynamics, pacing, and recovery mechanisms). For this reason, the activity complements the controller evaluation with a jitter analysis comparing: (i) TN vs NTN, and (ii) MPTCP vs MPQUIC on each backhaul segment.

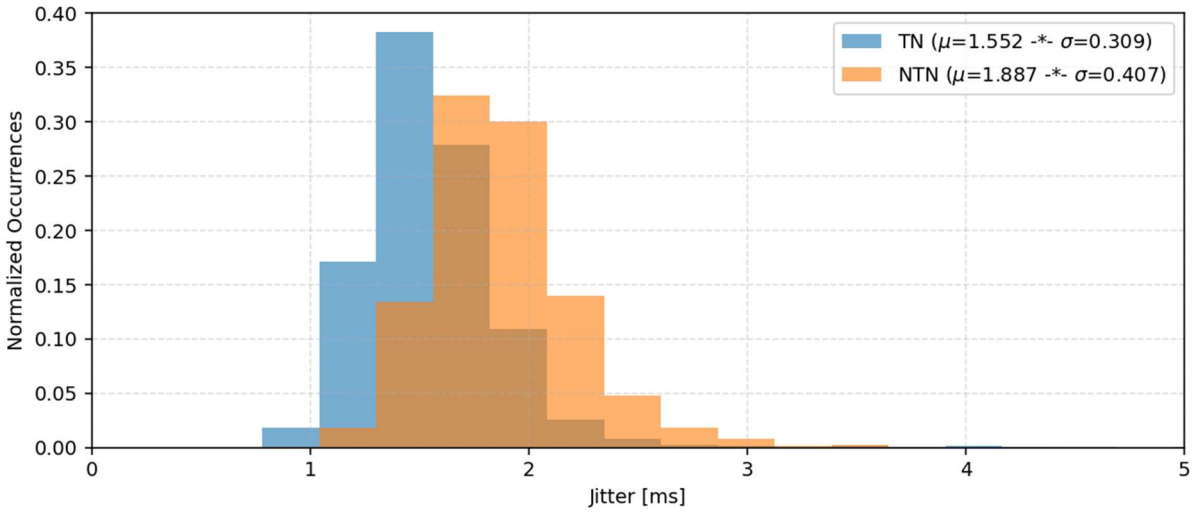

Figure 3-41: Jitter comparison: TN vs NTN.

Figure 3-41 compares the jitter observed on the terrestrial backhaul and, on the satellite backhaul. The expected and observed trend is that NTN exhibits higher jitter than TN. This reflects the non-stationary nature of LEO connectivity and the fact that the satellite path is more sensitive to variable capacity, buffering, and transient events. In this context, this figure supports two key points: (i) TN and NTN provide complementary characteristics (stability vs additional capacity), and (ii) multi-connectivity control must account not only for throughput aggregation but also for the temporal variability introduced by the satellite segment.

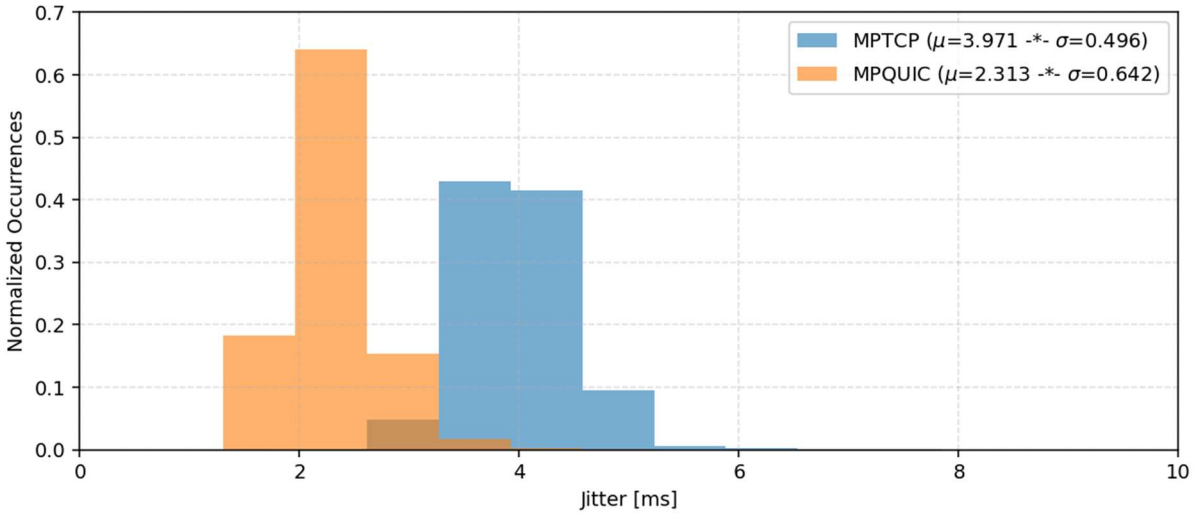

Figure 3-42: Jitter on TN: MPTCP vs MPQUIC

Figure 3-42 compares jitter for TCP-based and QUIC-based traffic on the terrestrial backhaul. Even on TN, where capacity is relatively stable, protocol implementation details (packetization, pacing, ACK behaviour, and congestion response) can lead to different jitter profiles. This

© 2023-2025 5G-STARDUST

figure is used to highlight that protocol coexistence is not only a question of bandwidth sharing different transport stacks may exhibit different time-domain behaviour under the same network conditions, which can be relevant when mapping traffic classes to slices or service objectives.

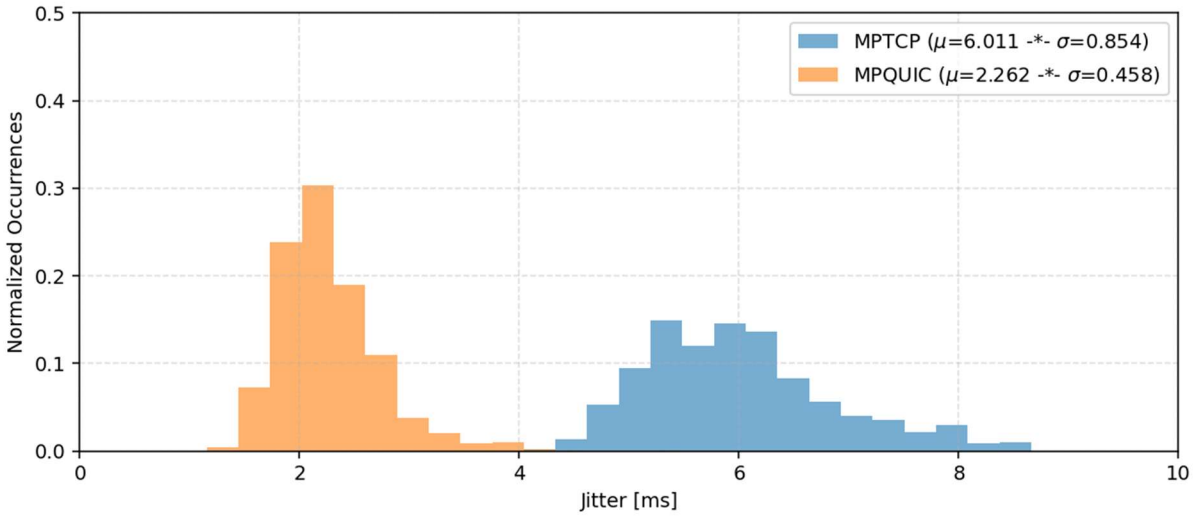

Figure 3-43: Jitter on NTN: MPTCP vs MPQUIC.

Figure 3-43 reports the same comparison on the satellite backhaul. On NTN, protocol differences are typically amplified by higher RTT and time-varying capacity. The figure highlights that both MPTCP and MPQUIC are affected by the satellite variability, but their jitter characteristics can differ due to their distinct transport mechanisms. In the context of this activity, this supports the rationale for protocol-aware control: when the backhaul includes NTN, enforcing predictable throughput shares is necessary but not always sufficient; understanding protocol jitter behaviour helps guide policy choices (e.g., which protocol family should be favoured on NTN under degradation or which services should be mapped to TN vs NTN resources).

The jitter results complement the throughput-based evaluation presented earlier by showing that TN–NTN integration introduces both capacity diversity and time-variation diversity:

- TN provides a stable, low-jitter anchor path.
- NTN provides additional capacity but with higher jitter and variability, requiring robust control and service-aware policy design.
- MPTCP and MPQUIC respond differently to these environments; therefore, protocol-aware slicing and adaptation (as introduced in the joint controller) are particularly relevant in hybrid backhaul deployments.

Overall, the jitter characterization confirms the importance of treating TN and NTN as complementary resources and supports the choice of a controller design that can enforce deterministic coexistence while leaving room for future enhancements where policy decisions may incorporate jitter-sensitive objectives.

3.4.7 Joint MPTCP-MPQUIC Conclusions

This activity demonstrates that transport-layer multi-connectivity over a hybrid TN–NTN backhaul can be made predictable and policy-driven even when heterogeneous protocol families coexist and the satellite segment exhibits strong time variability. The work starts from

© 2023-2025 5G-STARDUST

the practical observation that, in realistic deployments, traffic is not homogeneous: TCP-based services (leveraging MPTCP) and QUIC-based services (leveraging MPQUIC) share the same physical interfaces and therefore compete for the same bottleneck resources. In this context, controlling only one protocol stack is not sufficient to guarantee stable outcomes, because the overall TN/NTN utilization and the effective sharing between TCP and QUIC are influenced by cross-protocol contention.

The baseline analysis of Linux MPTCP schedulers confirms that scheduler choice can substantially change how MPTCP uses the terrestrial and satellite paths. Latency-oriented schedulers tend to favour the terrestrial backhaul; primary/backup-oriented schedulers can suppress steady-state satellite usage; balancing approaches can increase satellite activity but do not inherently provide deterministic splits under large RTT/BDP asymmetry and time-varying satellite capacity. These observations are important for the project because they show that “scheduler selection alone” does not provide a robust operator lever to control backhaul utilisation in integrated TN–NTN systems, especially when QUIC traffic is present.

The joint controller introduced in this activity addresses this gap by shifting the control point to where competition effectively occurs at the backhaul interfaces. By enforcing protocol-aware per-link slicing between TCP and UDP traffic classes on both terrestrial and satellite interfaces, the controller provides a deterministic policy handle that remains meaningful under mixed traffic. The evaluation confirms that this approach can maintain stable and configurable sharing between MPTCP and MPQUIC while preserving high utilisation of available capacity on both links. It also demonstrates that different policies can be applied independently on the terrestrial and satellite backhaul, supporting differentiated roles for TN and NTN resources in line with service objectives.

A further contribution is robustness to satellite dynamics. The satellite backhaul exhibits time-varying capacity and may undergo degradations or interruptions. The telemetry-driven control loop maintains policy coherence during such events and can apply adaptive behaviour when configured to do so, rebalancing allocations during capacity reductions and restoring the nominal policy upon recovery. This is particularly relevant to 5G-STARDUST, where the integration of NTN is motivated precisely by the need for continuity and additional capacity under heterogeneous and non-stationary conditions.

Finally, the activity provides an engineering-ready foundation for future intelligent control. The implemented control structure already includes the essential elements required for AI/ML-driven multi-connectivity: continuous telemetry, a policy decision step, and enforcement at the network edge. While the current controller uses lightweight rule-based adaptation, the same framework can be extended with predictive or learning-based policies, for example by anticipating satellite capacity evolution and proactively adjusting allocations to meet QoS/QoE targets. In summary, the activity contributes by delivering a practical mechanism for software-defined, protocol-aware multi-connectivity control over integrated TN–NTN backhaul, and by establishing a clear path toward more advanced AI/ML-assisted optimisation in future iterations.

© 2023-2025 5G-STARDUST

4 SOFTWARE DEFINED NETWORK CONTROL

With the emergence of large non-terrestrial constellations as an integral component of the fifth generation (5G) system and as an anticipated enabler of the sixth generation (6G) network evolution, the challenge of achieving efficient, scalable, and deterministic routing in highly dynamic topologies has become central to the architectural debate. The mega-constellation paradigm, characterized by hundreds or thousands of Low Earth Orbit (LEO) satellites interconnected through inter-satellite links and complemented by an extensive ground segment, introduces an environment in which topology changes occur with a periodicity defined by orbital motion and where the available paths between users, gateways, and data networks are continuously reconfigured. To achieve reliable and low-latency connectivity under such conditions, routing mechanisms must adapt at a speed and scale that exceeds the capabilities of classical distributed solutions. Traditional protocols such as Open Shortest Path First (OSPF) and Multiprotocol Label Switching (MPLS), while effective in terrestrial deployments, rely on frequent exchange of link state information, iterative convergence after failures or topology updates, and significant computational effort on each participating node. As such, their direct application to large-scale satellite infrastructures would introduce excessive overhead, prolonged convergence times, and a lack of determinism in the data path selection.

To address these limitations, Deliverable D5.2 of the 5G-STARDUST project introduced the concept of a Software Defined Networking (SDN) framework designed explicitly for satellite mega-constellations. The essence of this approach is to decouple the routing intelligence from the satellite nodes and to centralize it within a terrestrial controller capable of pre-computing the routing tables for all predictable topological states of the constellation. By proactively distributing these tables to satellites, user terminals, and ground stations prior to their usage, the SDN framework enables immediate switching of forwarding entries without the need for local route computation or inter-node signalling. Without such a shift of complexity towards the controller, the constellation nodes would remain burdened with state dissemination and recalculation, which scales poorly with constellation size. With this shift, on the other hand, routing convergence is effectively eliminated, node implementation is simplified to table look-up operations, and network determinism is enhanced. The main trade-off identified in this concept is the requirement for larger memory storage on spaceborne and terrestrial nodes to accommodate the set of pre-computed routing tables.

The present section builds upon this architectural proposition and provides a systematic validation of its feasibility. To this end, a simulation environment has been implemented in OmNET++, which models the orbital dynamics of a LEO constellation, integrates the SDN-based routing procedures, and establishes a reference baseline with distributed routing solutions. The simulator enables the measurement of routing convergence time, control plane overhead, and packet delivery performance under realistic traffic and mobility conditions. The evaluation confirms that the SDN-based approach delivers near-instantaneous adaptation to topology changes, reduces the control plane burden, and maintains robust data plane performance, thereby substantiating the claims initially formulated in Deliverable D5.2. Furthermore, the results underline that the memory overhead implied by the storage of multiple pre-computed tables is outweighed by the operational simplicity and responsiveness gained.

As such, the article not only provides the first quantitative performance evidence for the SDN framework in the context of non-terrestrial mega-constellations but also establishes a foundation for future extensions towards integrated transport-core 6G architecture. The remainder of the section is structured as follows: Section 4.2 summarizes the architectural concept of SDN-based routing as introduced in D5.2, Section 4.3 describes the OmNET++ simulation environment and the modelling assumptions, Section 4.4 presents the evaluation

© 2023-2025 5G-STARDUST

methodology and results, and Section 5 concludes with an analysis of implications and open research directions.

4.1 SDN ROUTING CONCEPT

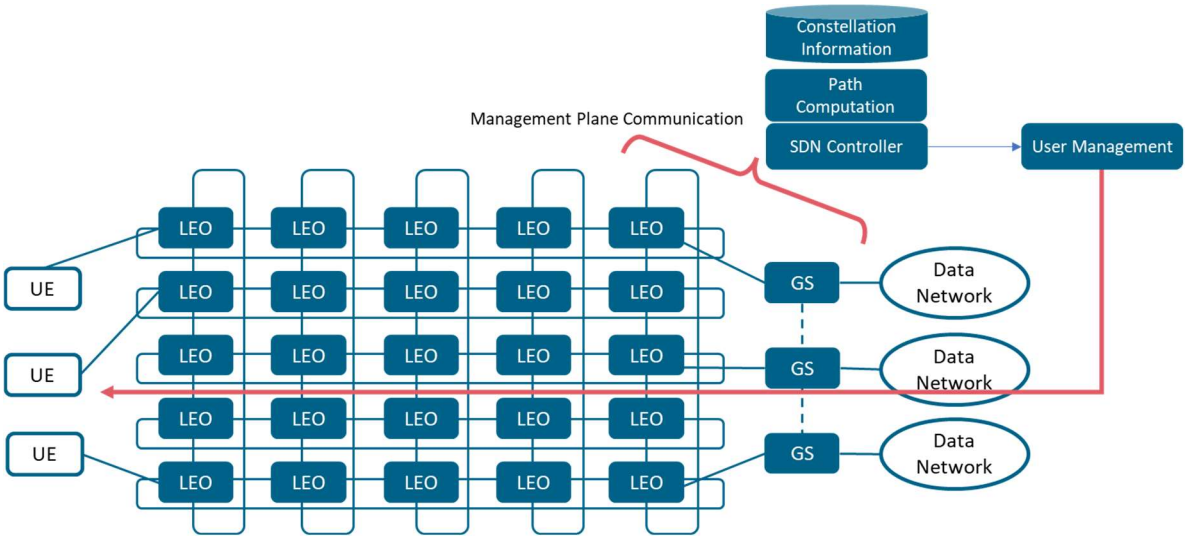

Figure 4-1: SDN Concept as introduced in D5.2

The SDN-based routing framework relies on the central principle of replacing reactive convergence with proactive configuration. Instead of engaging in distributed path computation, each satellite and ground node is equipped with a set of pre-computed routing tables corresponding to the constellation states that it will encounter over time. The terrestrial controller is responsible for generating these tables based on orbital predictability, for managing their distribution, and for ensuring their timely activation. In this way, the network avoids the uncertainty and delay associated with iterative reconvergence and provides deterministic adaptation to topological changes.

As illustrated in Figure 4-1, control architecture is characterized by a clear division of responsibilities. The terrestrial controller performs the global computation of routes, while the constellation nodes are reduced to executing table lookups and switching operations. This separation reduces the computational requirements of the satellites and minimizes in-orbit signalling, since nodes do not exchange routing updates but simply switch to the table that corresponds to the active topology state. The design thus leverages the predictability of orbital dynamics to ensure that routing adaptation can be performed locally and instantaneously, without dependence on inter-node coordination.

To realize this functionality, several architectural requirements must be satisfied. A label-based forwarding mechanism is employed to decouple the routing process from the IP addressing scheme, thereby avoiding the need for address reallocation as links appear and disappear. Each node must allocate sufficient memory to store the subset of routing tables required for its operational lifetime, which represents the principal trade-off of the design. The controller–node interface must provide reliable synchronization to guarantee that the correct table is activated at the correct time. Furthermore, fallback procedures are needed to handle potential desynchronization events, ensuring that service continuity is maintained even in the presence of temporary control-plane disruptions.

The resulting architecture transforms routing from a distributed, reactive, and computationally intensive process into a centralized, deterministic, and storage-driven process. While the

© 2023-2025 5G-STARDUST

elimination of convergence delays and signalling overhead constitutes a decisive advantage, the reliance on pre-distributed tables and the dependency on the terrestrial controller highlight the importance of dimensioning memory and ensuring robust control connectivity. These aspects define the operational feasibility of the SDN concept and motivate the need for a simulation-based evaluation, which is presented in the next section.

4.1.1 Simulation Environment

To evaluate the feasibility of the SDN-based routing framework introduced in Section 4.2, a discrete event simulation (DES) environment has been developed. The DES approach allows the complex behaviour of a satellite constellation to be represented as a sequence of scheduled events, executed in simulation time under deterministic control. Parameters defining the constellation, the physical environment, and the network configuration are provided through external configuration files. These are parsed at start-up and assigned to the corresponding models, ensuring that the simulation can be fully specified and reproduced. The simulator outputs structured result files, which contain the recorded measurements of routing performance, control overhead, and user-plane behaviour under the chosen conditions.

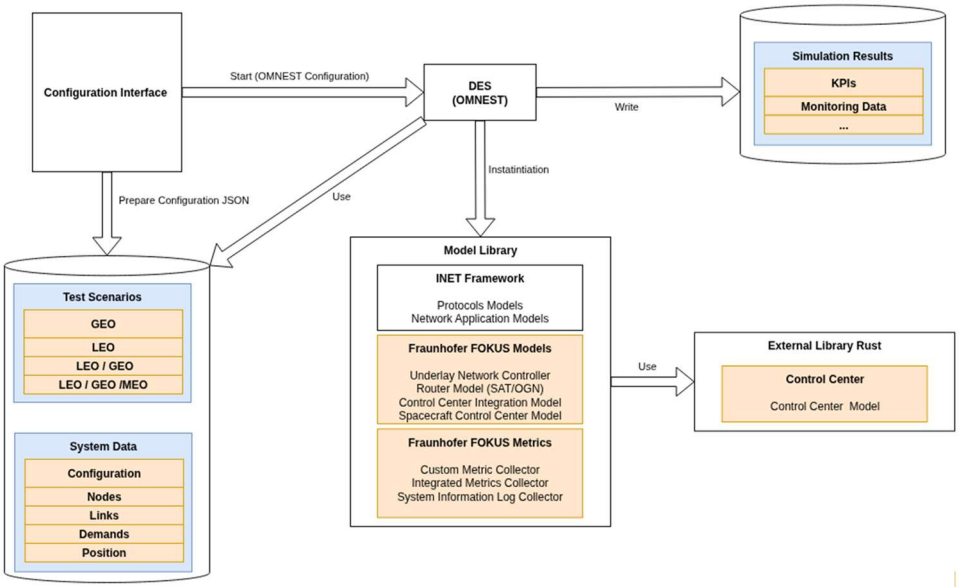

Figure 4-2: Discrete Event Simulator Architecture

As illustrated in Figure 4-2, the simulation environment is organized into a modular architecture that separates the simulation kernel, the modelling framework, and the configuration and result handling. At the core resides the discrete event kernel, which provides event scheduling, time management, and the execution of model interactions. This kernel forms the foundation of the execution environment and ensures that all interactions between models are executed in a temporally consistent manner. Above the kernel sits the configuration and user interface, provided through an Eclipse-based integrated development environment that supports the preparation of configuration files, the debugging of scenarios, and the verification of execution. A model library complements the kernel and interface by providing protocol and application components, with the INET framework serving as the baseline collection of modules for standard Internet protocols. These generic models can be extended by domain-specific modules that capture the characteristics of satellite links, orbital dynamics, and routing procedures. Test scenarios combine topology descriptions, traffic configurations, and protocol settings, enabling controlled and repeatable experiments under a wide variety of conditions.

© 2023-2025 5G-STARDUST

The model descriptions are themselves structured. Topologies are specified in NED files, which define the modules, ports, and interconnections. Message formats are introduced through dedicated .msg files, which generate C++ classes representing protocol data units with configurable fields. Functional behaviour is implemented in C++ sources, which are compiled and linked to form the simulation modules. This layered description allows scenarios to be assembled systematically, while retaining flexibility to introduce specialized behaviour where required.

The execution of the simulator follows a structured progression of states, each with a distinct role in ensuring reproducibility and robustness. The Start state contains only the kernel and its auxiliary services, such as the scheduler and logger. At this point no models are present, which allows the simulator to begin from a clean baseline without residual configurations. The transition from this state is automatic and error-checked, preventing execution if a configuration file is incomplete or corrupted.

The Instantiation and Initialization state creates the models defined in the topology files and assigns them their parameters. Instantiation generates the structural elements of the network, such as satellites, ground stations, and inter-satellite links. Initialization then parameterizes these elements with constellation coordinates, channel conditions, and routing configurations. Both steps are performed automatically by the kernel, and the simulator halts at the first blocking error to prevent execution under invalid conditions. This design ensures that every scenario begins from a consistent and validated baseline.

The Running state represents the main execution phase. Here simulation time advances continuously according to the scheduled events, while the modules interact as they would in a real network. Routing updates, packet transmissions, and control-plane exchanges are processed in order dictated by the global event queue. This state embodies the realism of the DES paradigm: each action is executed at the precise simulated time, maintaining strict causal order.

The Stepping state provides an alternative execution mode in which the simulation does not run continuously but instead advances through discrete time steps. Each step increases the simulated time as fast as the host machine allows, depending on the computational load of the scenario. Once the specified step is complete, the simulator returns to a paused condition, allowing inspection of intermediate results and detailed analysis of event sequences. This capability is essential for debugging and for fine-grained evaluation of protocol behaviour, since it exposes the micro-dynamics of the simulation that would otherwise be hidden in continuous execution.

The Stopped state is reached either after a complete execution or when the user interrupts the process. In this state, all results accumulated up to the stopping point are preserved, ensuring that partial experiments can still be analysed. The stopped state is not final: the simulation can be resumed or re-executed from this point, which makes it particularly useful for iterative testing.

Finally, the End state clears all instantiated models and releases the memory of the host machine. This guarantees that subsequent runs begin without interference from previous ones, preserving reproducibility across different experiments. The explicit transition to an end state also facilitates batch execution of scenarios, where multiple runs are conducted in sequence with controlled initialization and termination.

4.1.2 SDN Routing Evaluation and Results

The evaluation addresses the feasibility of deploying an SDN control plane over a LEO mega-constellation with three globally distributed feeder links, each connecting satellites to a

© 2023-2025 5G-STARDUST

terrestrial controller. This scenario represents the minimal diversity required to provide continuous access but also exposes the system to cold-start challenges and to the dependency of the control plane on feeder availability. The analysis focuses on three aspects: the impact of hop distance on routing stability, the propagation of control information across the constellation, and the resilience of the data plane in the presence of impairments.

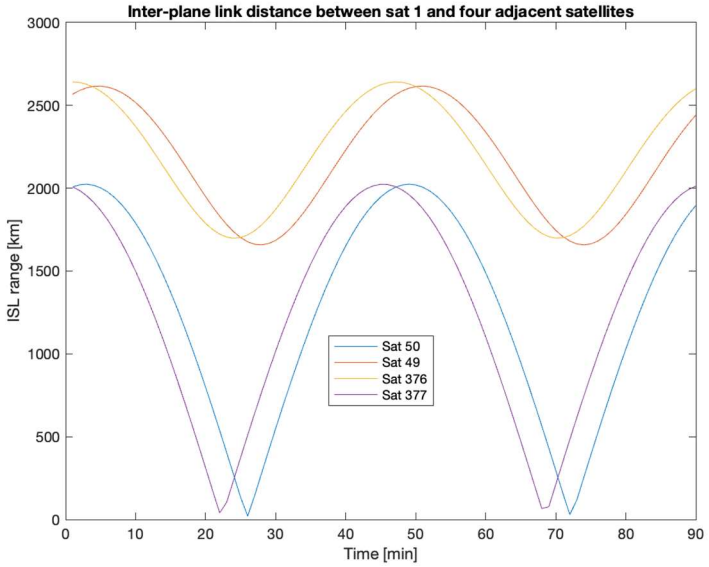

Figure 4-3: Distance between satellite and its neighbours (Matlab generated illustrative example)

The first aspect concerns the fundamental choice of metric for routing path selection. *Figure 4-3* shows an example of inter-plane link distances between one satellite and its four adjacent neighbours over a 90-minute orbital cycle. The results underline the oscillatory nature of orbital geometry: while some neighbours remain at relatively stable ranges, others approach and recede in periodic fashion, with link distances varying from almost 0 km at closest approach to more than 2500 km at maximum separation. If routing were to select paths purely on instantaneous physical distance, the preferred next hop would alternate repeatedly within each cycle, introducing instability into the forwarding process and unnecessary churn into the control plane. By abstracting to the number of hops rather than to raw distance, the SDN framework suppresses these oscillations, producing stable and predictable path selection despite the continuous variation of inter-satellite ranges. The importance of this choice extends beyond user-plane forwarding. It is equally critical for the management plane, since SDN control messages must traverse the constellation with reliability. If the path towards the controller were to change at each oscillation, the management plane itself would be destabilized, with control messages experiencing unnecessary detours or interruptions. The figure thus provides a representative example that explains why hop-count based routing is appropriate for deterministic operation of both data and SDN control planes in highly dynamic constellations.

The second aspect of the evaluation investigates the latency of control-plane communication between the terrestrial SDN controller and satellites across the constellation. *Figure 4-4* presents the one-way delay distribution measured at a representative set of nodes, while the constellation topology changes every 50 seconds. Each dot corresponds to a received control message, with its vertical position indicating the delay experienced. The figure clearly separates two regimes. During the interval from 0 to 50 seconds, when feeder links or gateway–satellite links are degraded, the observed delays are both larger and more dispersed, reflecting the reliance on longer detour paths and the reduced stability of the control channel.

© 2023-2025 5G-STARDUST

Once conditions improve after 50 seconds, the distribution contracts and the latencies converge towards lower, more predictable values.

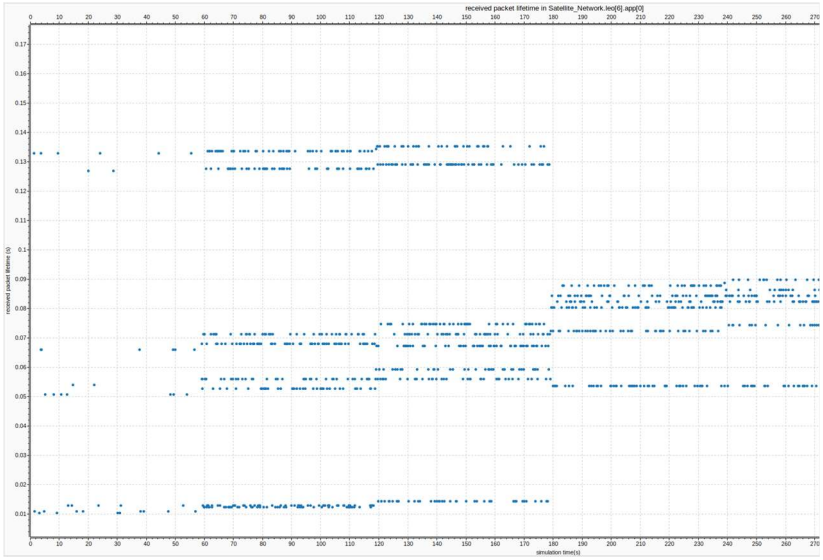

Figure 4-4: One way latency in a 600 satellite mega-constellation

This pattern demonstrates the dependency of the SDN management plane on feeder link quality. Even though inter-satellite forwarding can deliver control messages under degraded conditions, the delay penalty grows significantly, which can slow down the propagation of routing updates and synchronization commands. Conversely, when feeder links operate under favourable conditions, the controller–satellite communication becomes stable and bounded, enabling timely activation of pre-installed routing tables. The figure thus highlights both the strength and the weakness of the centralized control approach: the strength lies in its predictability when feeder connectivity is ensured; the weakness lies in its sensitivity to impairments of the feeder links, which directly affects the responsiveness of the management plane.

As already underlined in Deliverable D5.2, the weakest point of an SDN-based constellation is the dependability on continuous reachability of the terrestrial controller. This dependency creates two critical challenges. The first is how the management channel can be established initially when the constellation is cold-started and no satellite knows a path to the controller. The second is how the channel can be maintained under feeder impairments, where adverse weather or technical failures render one or more feeders unavailable. To address these challenges, the proposed mechanism relies on local neighbour exchanges. Each satellite queries its immediate neighbours for known routes towards a feeder, and once information is received, it updates its own state accordingly. Over successive exchanges, this information propagates through the constellation. The radius of this dissemination is referred to as the Horizon of Information (HoI): at each step, more satellites become aware of a valid path to the controller, and the management plane expands its reach.

The third challenge concerns the cold-start condition, where the constellation has just been activated and no satellite knows yet how to reach the SDN controller. In this case, the management plane must bootstrap itself by relying on neighbour exchanges that gradually extend knowledge of feasible paths. *Figure 4-5* illustrates this process, showing how satellites incrementally learn the number of hops required to reach a feeder link connected to the controller. At the beginning, only nodes in direct line of sight of a feeder link can establish communication. With each exchange, information spreads outward to adjacent nodes, which register their distance in hops to the controller. Over time, this learning process propagates

© 2023-2025 5G-STARDUST

across the constellation, until all satellites have a defined management path. The figure demonstrates that cold-start establishment is feasible without global flooding, but it requires controlled neighbour dissemination to prevent excessive signalling.

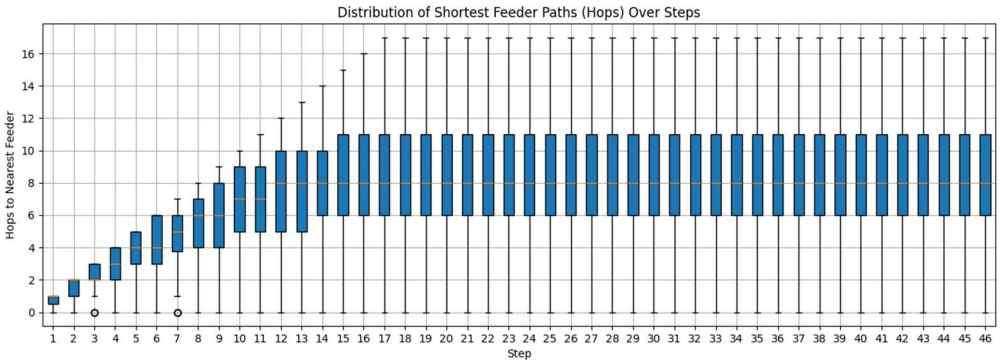

Figure 4-5: Cold Start Example: Satellites learn from neighbours where feeder links are available

For robustness, it is not sufficient that satellites know a single path to the controller; redundancy is required to sustain operation during feeder impairments. *Figure 4-5* reports how the number of satellites with at least one, two, or three feeder paths evolves as the neighbour-learning process continues. The results show that connectivity to at least one feeder is established relatively quickly across the constellation, whereas establishing multiple independent feeder paths takes considerably longer. In practice, this means that cold-start completion must be defined not only by full coverage of single paths but also by the time until all satellites have redundant routes to different feeders. The figure therefore highlights that redundancy does not come for free: it extends the duration of bootstrap but is indispensable for achieving a management plane that is both reliable and resilient.

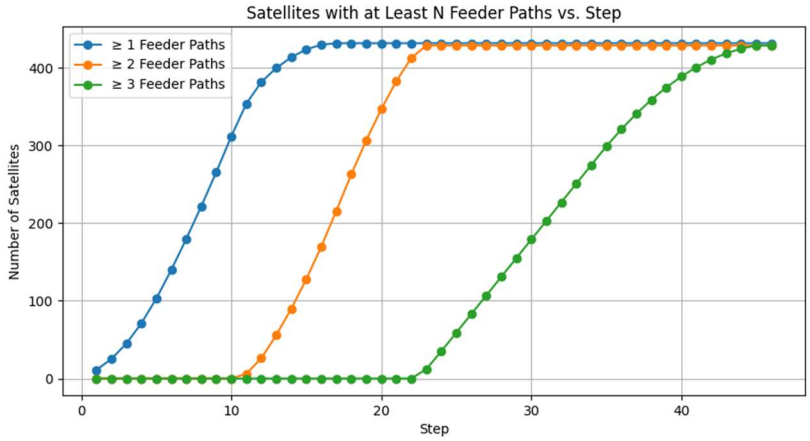

Figure 4-6: Number of available feeder links

The fourth challenge concerns the effect of feeder link outages on the established management plane. *Figure 4-7* shows the distribution of effective path lengths in hops during the propagation of a feeder failure. When the outage occurs, satellites that previously used the failed feeder must reroute their control traffic through alternative feeders. The boxplots reveal a sharp increase in path length, with median values rising from under 10 hops to nearly 30 hops at the peak of failure propagation. The spread of the distribution also widens, indicating that while some satellites experience only moderate additional delays, others are forced onto much longer detours. Outliers confirm that a small subset of nodes may be severely penalized, temporarily reaching the maximum feasible hop distance. Importantly, the figure also shows

© 2023-2025 5G-STARDUST

the qualitative shift in traffic behaviour: once the information about the failure reaches a satellite, its SDN traffic is no longer forwarded in vain towards the unavailable feeder but is redirected immediately towards an alternative feeder destination. This mechanism prevents unnecessary loops or packet losses and ensures that, even under degraded conditions, the management plane continues to operate, albeit at increased latency. The figure therefore demonstrates that a single feeder outage does not break the management plane, but it does stretch it significantly, reducing responsiveness and increasing heterogeneity across the constellation.

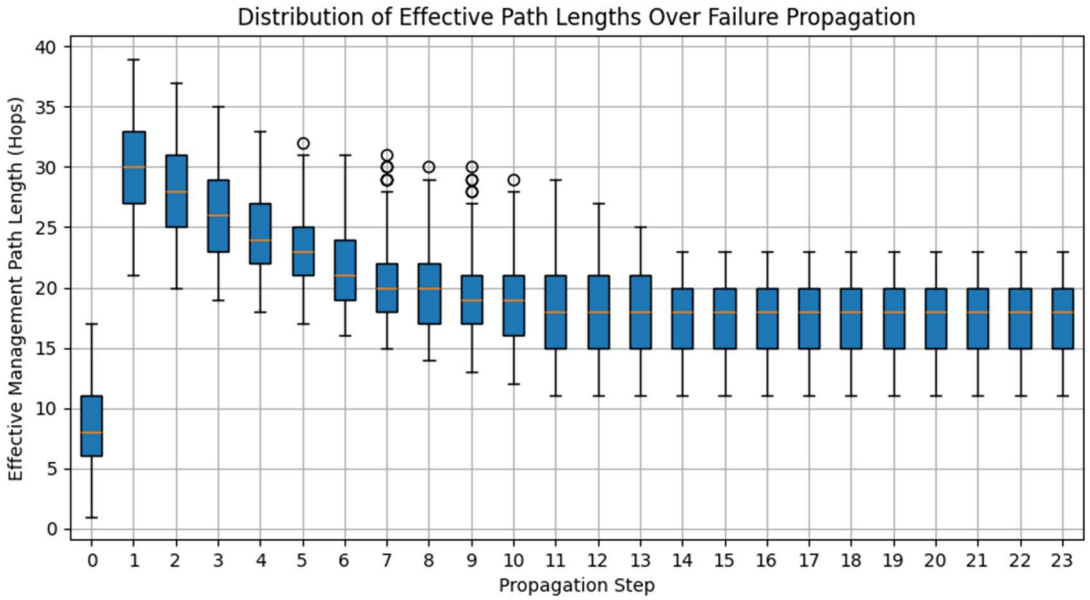

Figure 4-7: Impact of feeder link losing availability propagation (step 0 is before the failure)

Figure 4-8 presents the complementary recovery process once the failed feeder becomes available again. As the restoration information propagates step by step, satellites progressively redirect their SDN traffic away from long detours and back towards the re-established feeder destination. The distribution of effective path lengths contracts steadily towards shorter values, with the median hop count declining and the variance reducing at each propagation step. This confirms that the management plane does not merely fall back to normal operation by chance but actively restructures itself as nodes update their feeder selection and forward traffic directly to the restored entry point. Outliers, visible in the early steps, show that some nodes take longer to adjust, yet the overall distribution converges to the pre-failure baseline as the Horizon of Information extends across the constellation. The figure therefore demonstrates that the management plane is self-healing: once a feeder returns, routing efficiency is gradually restored, traffic is redirected without unnecessary delay, and the system stabilizes to its nominal operating state. The limitation is that the recovery is not instantaneous but bounded by the speed of propagation, which in large constellations translates to tens of steps before all nodes benefit from the restored feeder.

© 2023-2025 5G-STARDUST

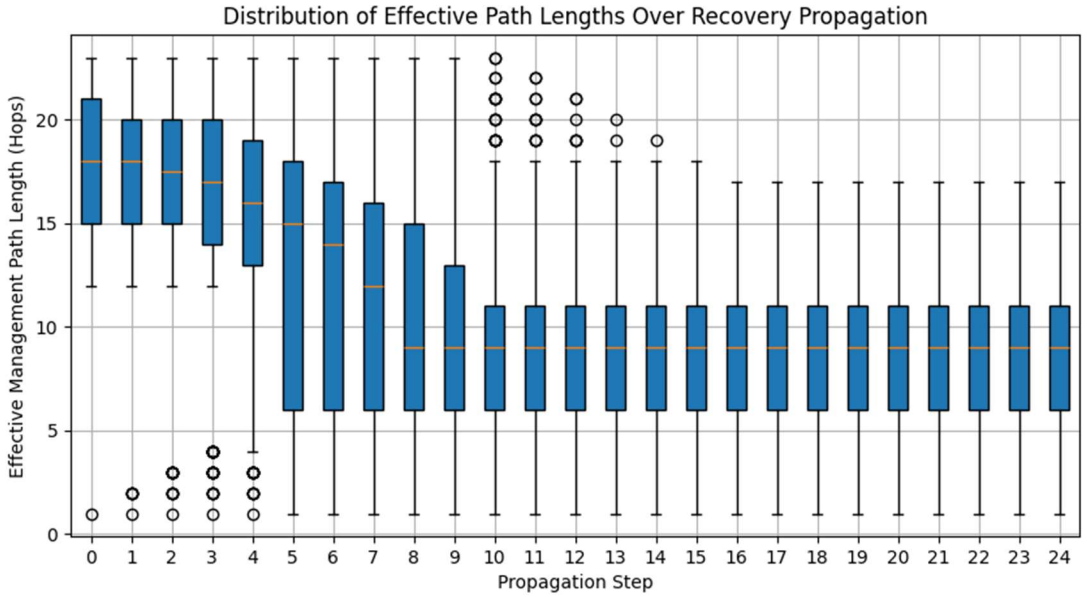

Figure 4-8: Impact of feeder link becoming again available (step 0 is when links becomes available)

The presented measurements provide an initial quantitative validation of the feasibility of deploying an SDN management plane in a large-scale LEO constellation. By combining hop-count abstraction, distributed neighbour learning, and feeder diversity, the evaluation has demonstrated that routing stability can be maintained despite oscillating inter-plane distances, that one-way controller latencies remain bound under favourable feeder conditions, and that management connectivity is preserved even in the presence of feeder outages. The experiments on cold start and recovery further confirm that the management plane can be bootstrapped and self-healed without requiring global flooding, with redundancy ensuring that satellites maintain access to at least one controller path even under degraded conditions.

It must be underlined, however, that these results are obtained in a controlled simulation environment. They reflect the behaviour of a single constellation topology and depend on the modelling abstractions chosen for inter-satellite links, feeder link dynamics, and event propagation. The discrete event simulation provides a faithful representation of protocol interactions, but it cannot capture all physical effects or the full variability of real orbital conditions. As such, the reported hop counts, propagation steps, and latency distributions are best interpreted as evidence of feasibility and relative trends, rather than as absolute guarantees.

From the viewpoint of architectural assessment, the measurements provide three further considerations. First, they confirm that SDN control can be sustained under realistic constellation dynamics, thereby proving the basic feasibility of the approach when feeder diversity is available. Second, they show that the data path from a satellite to the controller is inherently long, as already underlined in Deliverable D5.2, which means that management messages traverse multiple hops and may suffer from non-negligible delays. Third, they highlight that satellites cannot depend solely on immediate controller feedback for every decision; instead, they must implement local reactions that bridge the time until updated instructions arrive from the controller. In practice, this implies a hybrid model, where SDN provides global coordination, while nodes retain the ability to execute short-term adaptations based on local state.

Nevertheless, within these boundaries, the evaluation provides actionable insights. It shows that an SDN-based management plane can be constructed on top of realistic orbital dynamics,

© 2023-2025 5G-STARDUST

that its main vulnerability lies in feeder availability, and that redundancy mechanisms are essential to preserve stability. The results therefore serve as a foundation for refining the architecture and for guiding future experimental campaigns, whether through larger-scale simulations that incorporate multiple constellations or through hardware-in-the-loop emulation. In this sense, the measurements are not the final proof of viability, but they provide a necessary intermediate step: a controlled, reproducible confirmation that the proposed concepts hold under representative conditions and expose the design points that require further attention.

4.2 MULTI-CRITERIA CONDITIONAL HANDOVER

The integration of NTN into 5G systems introduces mobility conditions that differ fundamentally from terrestrial deployments. Large LEO mega-constellations form communication infrastructures with continuously evolving topology in which multiple satellites may simultaneously provide viable access connectivity. While satellite motion follows predictable orbital dynamics, both visibility relations and transport paths toward ground stations change over time, influencing the performance perceived by the UE.

In contrast to terrestrial networks, where mobility decisions are largely decoupled from backhaul routing conditions, satellite attachment directly determines the subsequent end-to-end transport path through the constellation. As a result, selecting a serving satellite based solely on instantaneous radio indicators does not necessarily ensure efficient service performance. Differences in inter-satellite routing distance and ground station feeder link reachability may dominate latency behaviour, particularly in large-scale deployments with distributed ground statement placement.

Conditional Handover (CHO) mechanisms standardized for 5G improve mobility robustness by enabling the network to pre-configure candidate targets and execution conditions at the UE. NTN enhancements further exploit satellite predictability through timing- or location-based triggers intended to approximate expected residence intervals. However, residence-time estimation derived from geometric visibility provides only an upper-bound indication of connectivity duration and does not capture environmental impairments that may interrupt the access link earlier than predicted.

These characteristics require mobility decision frameworks that extend beyond access feasibility and explicitly incorporate network-level performance. When multiple satellites satisfy minimum connectivity conditions, the attachment decision determines both access persistence and transport efficiency. At the same time, the access link may be interrupted abruptly due to loss of Line-of-Sight (LoS) caused by environmental obstruction or rapid geometric evolution. As such, mobility control must rely on predictive evaluation in order to stabilize end-to-end service performance across evolving NTN constellation states.

In this section we propose a multi-criteria CHO framework enabling predictive candidate ordering based on combined residence-time and transport-efficiency indicators derived from a constellation digital twin implemented on the Fraunhofer FOKUS OpenLANES platform. Evaluation on a large LEO mega-constellation demonstrates improved latency consistency and reduced unnecessary handover activity compared to single-metric selection approaches.

4.2.1 Background: Conditional Handover and NTN Mobility

Conditional Handover (CHO) was introduced in 5G mobility procedures to improve handover robustness by decoupling preparation and execution phases ([88], [89]). The network pre-configures the UE with candidate target cells and associated execution conditions while radio conditions remain favourable. The UE subsequently performs the handover autonomously

© 2023-2025 5G-STARDUST

once the configured criteria are fulfilled, thereby reducing dependence on real-time signalling and improving mobility stability under rapidly varying link conditions.

With the introduction of NTN support, CHO mechanisms were extended to exploit the deterministic characteristics of satellite motion. Timing and location based triggering conditions were introduced to approximate satellite residence intervals and enable predictive handover execution within visibility windows ([90], [91], [92]). Although such approaches improve access robustness in LEO constellations, mobility decisions driven primarily by instantaneous signal-strength indicators such as Reference Signal Received Power (RSRP) may lead to increased handover activity ([93]). Multi-attribute handover schemes therefore incorporate additional criteria such as channel conditions or predicted residence time ([94]). However, these approaches generally treat the serving satellite as a radio endpoint and do not explicitly consider the transport path traversed by user traffic through the constellation.

In large mega-constellation deployments, handover signalling may need to traverse the core network due to the absence of persistent inter-gNB connectivity, increasing latency and control overhead ([95]). At the same time, distributed routing protocols are challenged by the highly dynamic nature of inter-satellite link topologies ([96], [97], [98], [99]). Centralized routing architectures based on Software-Defined Networking (SDN) therefore provide a practical means of maintaining global awareness of constellation connectivity ([100], [101], [102], [103]). Under these conditions, once multiple satellites satisfy minimum access-feasibility constraints, the handover decision determines the resulting end-to-end routing path and its latency characteristics. This observation motivates predictive mobility frameworks in which candidate ordering reflects both residence-time dynamics and anticipated transport performance in evolving NTN topologies.

4.2.2 Multi-Criteria Conditional Handover Framework

Mobility control in large Low-Earth-Orbit mega-constellations differs fundamentally from terrestrial handover operation. In NTN deployments, the serving satellite determines not only access feasibility but also the subsequent end-to-end routing structure toward geographically constrained gateway infrastructure. As a result, mobility decisions must consider the expected evolution of both connectivity persistence and transport performance across successive constellation states.

Due to wide satellite coverage footprints and predictable orbital motion, multiple attachment opportunities frequently coexist. Once geometric LoS is available and the received signal strength exceeds the minimum level required for reliable communication, the access link primarily represents a feasibility constraint rather than a continuous optimization objective. Differences in instantaneous signal strength therefore have limited impact on perceived service performance compared to the transport characteristics induced by constellation topology. This observation motivates a predictive multi-criteria candidate evaluation process integrated into Conditional Handover preparation.

At constellation state t, the network first determines the set of feasible satellites satisfying minimum connectivity conditions,

$$C(t) = \{s\_i \mid LOS\_i(t) = 1 \wedge RSRP\_i(t) \geq \theta\}, \quad (1)$$

thereby ensuring that subsequent ordering operates exclusively on viable attachment alternatives. For each feasible satellite, the network derives predictive indicators describing both the remaining visibility duration and the anticipated end-to-end transport delay toward gateway infrastructure. The predicted residence time provides an upper bound on connection persistence based on orbital geometry, while the transport delay reflects the efficiency of the routing path induced by attachment to the respective satellite.

© 2023-2025 5G-STARDUST

Candidate ordering is obtained through a composite preference evaluation combining normalized delay and residence-time indicators together with an explicit switching penalty favouring attachment stability. This ordering reflects the fundamental NTN mobility trade-off between minimizing routing latency and avoiding excessive handover activity. The resulting ranked candidate sequence is conveyed to the UE as part of Conditional Handover configuration.

Given the deterministic short-term evolution of satellite motion and inter-satellite connectivity, predictive candidate ordering can be computed in advance using estimated UE position and routing objectives. The UE therefore maintains locally available fallback attachment options and can react immediately to sudden access degradation caused by LoS interruption, propagation impairment, or topology transitions without requiring real-time control-plane signalling.

In accordance with standardized CHO operation, handover execution remains governed by locally observed feasibility conditions. Periodic re-computation of predictive indicators enables proactive mobility preparation aligned with constellation dynamics. In this way, the proposed framework supports stabilization of end-to-end service performance across successive visibility intervals while preserving robust access continuity in large-scale NTN deployments.

4.2.3 Digital Twin-Based Generation of predictive CHO indicators

To enable the predictive multi-criteria mobility preparation process described in Subsection 4.2.2, the UE requires continuous awareness of the temporal evolution of satellite visibility and transport-routing conditions. This awareness is provided through a constellation-level digital twin that represents the dynamic geometric configuration of the considered LEO deployment and its implications for end-to-end connectivity performance. The digital twin therefore constitutes the functional basis for deriving the predictive residence-time and transport-delay indicators used in the proposed CHO framework.

The digital-twin environment models the temporal evolution of an operational LEO mega-constellation using publicly available TLE orbital descriptions. Satellite trajectories are obtained through ephemeris propagation rather than synthetic topology construction, thereby enabling reconstruction of the expected satellite motion and realistic inter-satellite geometry. Based on these time-varying orbital states, satellite nodes hosting digital gNB functionality are interconnected through dynamically evolving inter-satellite links that form a distributed transport fabric. Gateway connectivity is provided through geographically fixed ground stations anchoring user traffic toward a ground-based core-network instance. A centralized routing controller maintains global awareness of constellation connectivity derived from propagated ephemeris information and periodically computes transport paths toward the selected gateway objective.

© 2023-2025 5G-STARDUST

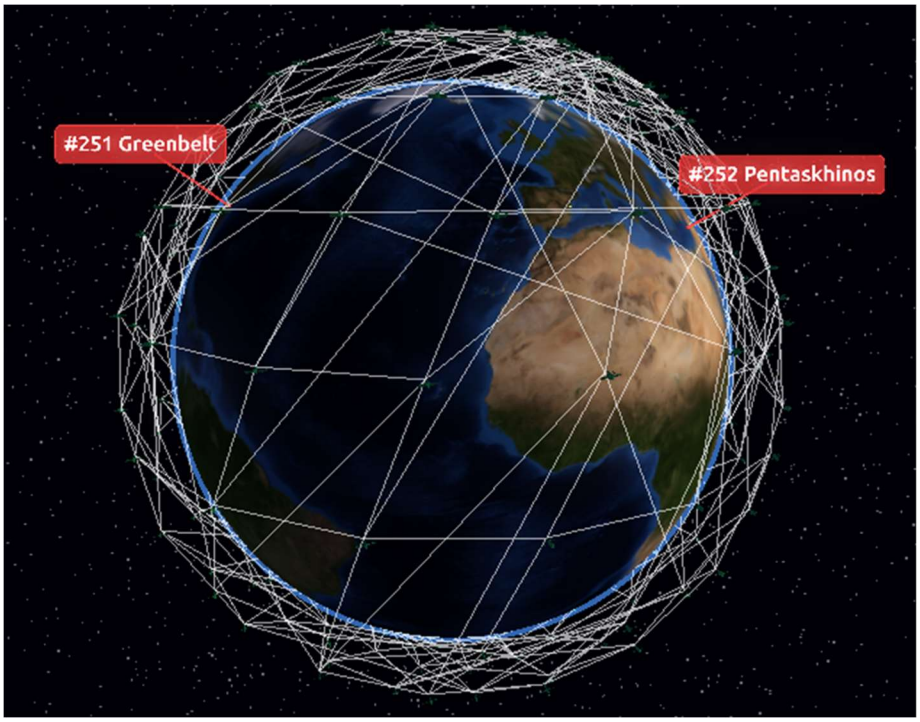

Figure 4-9: TLE-derived realization of the considered operational LEO mega-constellation used for predictive CHO indicator generation, including the gateway placement relevant for end-to-end routing evaluation.

The constellation realization used for predictive indicator generation is illustrated in Figure 4-9. The depicted topology highlights the simultaneous availability of multiple feasible attachment opportunities and the resulting routing diversity toward gateway infrastructure. As such, the figure underlines the fundamental coupling between access selection and transport efficiency that motivates the proposed multi-criteria handover preparation approach.

Constellation evolution is evaluated over a sequence of discrete time instants representing successive orbital states. Predictive indicators are recomputed at minute-scale temporal resolution for the entire constellation domain rather than continuously for each individual UE. This design reflects a practical trade-off between predictive fidelity and computational scalability. Instead, a continuous per-UE re-computation would introduce prohibitive processing overhead without significantly altering the relative ordering of feasible candidates, whereas periodic constellation-wide updates preserve the mobility-relevant topology dynamics governing attachment decisions.

At each evaluation instant, geometric LoS visibility between the UE and satellite nodes is determined using ephemeris information, enabling estimation of the remaining visibility duration associated with each feasible attachment alternative. This prediction supports computation of the maximal residence-time indicator defined in Subsection 4.2.2. In parallel, deterministic routing procedures are applied to derive end-to-end transport paths from each visible satellite toward the selected gateway. These computations yield latency-related performance indicators and inter-satellite hop-count characteristics that reflect the efficiency of the induced transport path across the evolving constellation topology.

While TLE-based ephemeris propagation provides physically grounded modelling of constellation dynamics, the resulting predictive indicators inherently represent an approximate estimate of future connectivity conditions. Orbital perturbations, atmospheric effects, or operational manoeuvres may introduce deviations between predicted and effective satellite

© 2023-2025 5G-STARDUST

positions. In addition, geometric line-of-sight availability derived from ephemeris information does not guarantee effective access connectivity in practical deployments. Local terrain, urban structures, vegetation, or transient blockage may interrupt the access link earlier than predicted visibility intervals suggest. Consequently, the candidate ordering derived from the digital twin should be interpreted as anticipatory mobility guidance rather than an exact representation of future attachment sequences. This predictive uncertainty reflects an inherent characteristic of long-horizon mobility preparation in large-scale NTN deployments.

The digital twin can operate using TLE datasets obtained from measurement traces or using ephemeris information adjusted to reflect orbital knowledge updated based on the live system telemetry. This flexibility enables both offline evaluation scenarios and operational mobility-preparation mechanisms in which constellation state information is periodically refreshed. The resulting time-indexed predictive indicator sets are subsequently used to construct the ranked CHO configurations described in Subsection 4.2.2. By providing anticipatory awareness of routing evolution and visibility persistence, the digital twin enables proactive mobility preparation processes that extend beyond reactive radio-centric decision strategies and contribute directly to the stabilization of end-to-end service performance across successive visibility intervals.

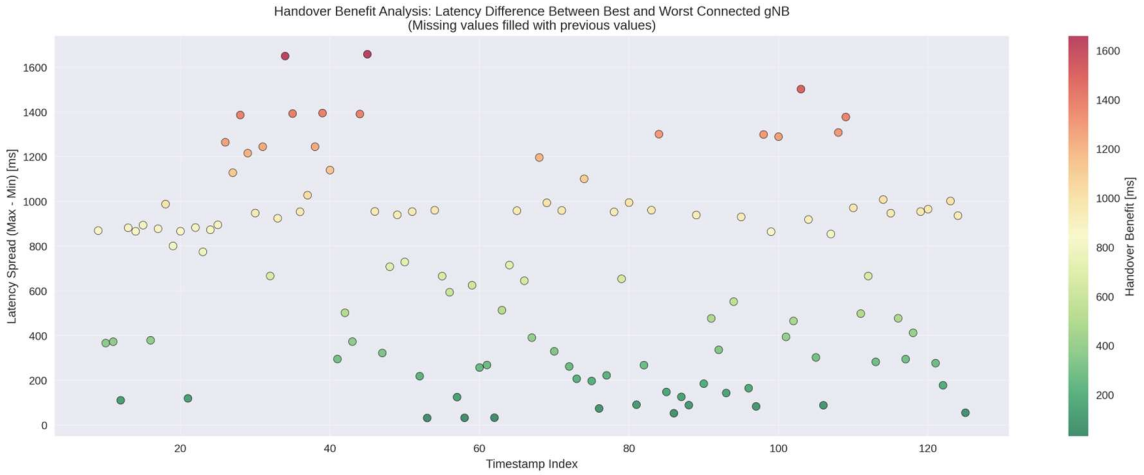

Figure 4-10: Latency spread across visible satellites for all evaluated constellation states.

4.2.4 Conditional Hand-Over Evaluation

Using the mega-constellation digital-twin scenario described in Subsection 4.2.3, this evaluation investigates how satellite selection during CHO influences end-to-end latency and connection stability. For each constellation state, predictive indicators are derived for all satellites simultaneously visible to the UE. These indicators include routing latency toward gateway infrastructure, ISL hop count, LoS-based access feasibility, and the predicted residence time obtained from orbital geometry. The resulting time-indexed dataset enables systematic comparison of attachment strategies derived from the predictive indicators defined in Subsection 4.2.2 and therefore allows observation of mobility behaviour across the evolving constellation topology.

Measurements are obtained by periodically evaluating constellation geometry and deterministic routing conditions in the digital twin. Satellite trajectories are derived from TLE-based ephemeris propagation, while transport paths toward the gateway are computed through deterministic routing across the inter-satellite link fabric. This methodology enables observation of latency variability across feasible attachment alternatives and supports analysis of the trade-off between transport efficiency and mobility stability. The evaluation therefore focuses on the temporal evolution of three key indicators governing NTN mobility decisions:

© 2023-2025 5G-STARDUST

predicted satellite residence time, routing latency toward gateway infrastructure, and the number of executed handovers.

Figure 4-11: Comparison of attachment strategies. Top: end-to-end latency evolution for latency-optimal and persistence-based selection. Middle: ISL hop-count evolution. Bottom: summary statistics including mean latency and total handover count

4.2.4.1 Comparison of Latency-Optimal and Residence-Time-Based Selection

To illustrate the mobility trade-offs introduced by transport-aware satellite selection, *Figure 4-11* compares two attachment strategies. The first strategy selects the satellite that minimizes end-to-end routing latency toward the gateway. The second strategy prioritizes satellites providing the longest predicted residence time, thereby maximizing expected connection persistence. These strategies represent two opposing mobility objectives: transport efficiency and connection stability.

The latency evolution plot shows that routing-aware attachment consistently maintains a lower delay envelope across most constellation states. The latency-optimal strategy tends to follow satellites positioned along shorter transport paths toward the gateway and therefore reduces the number of inter-satellite traversals required to reach ground infrastructure. In contrast, residence-time-based selection tends to remain associated with satellites offering extended visibility intervals, which results in fewer mobility transitions but may lead to longer transport paths.

© 2023-2025 5G-STARDUST

The ISL hop-count evolution further illustrates the structural relationship between latency and route length. Satellites selected by the latency-optimal strategy consistently lie along routes involving fewer inter-satellite hops, while persistence-based selection frequently follows satellites located deeper within the constellation topology. The summary statistics shown in the lower panel of Figure 4-11 confirm this trade-off. While routing-aware attachment reduces average latency, it also introduces increased handover activity due to transitions toward satellites offering improved routing conditions. These observations confirm that mobility control in large NTN mega-constellations requires coordinated evaluation of transport efficiency and connection persistence.

4.2.4.2 Path Diversity and Potential Handover Benefit

The impact of satellite selection becomes particularly visible during constellation states in which multiple feasible attachment opportunities lead to significantly different routes. Figure 4-12 illustrates two representative constellation configurations in which visible satellites provide distinct transport routes toward the gateway.

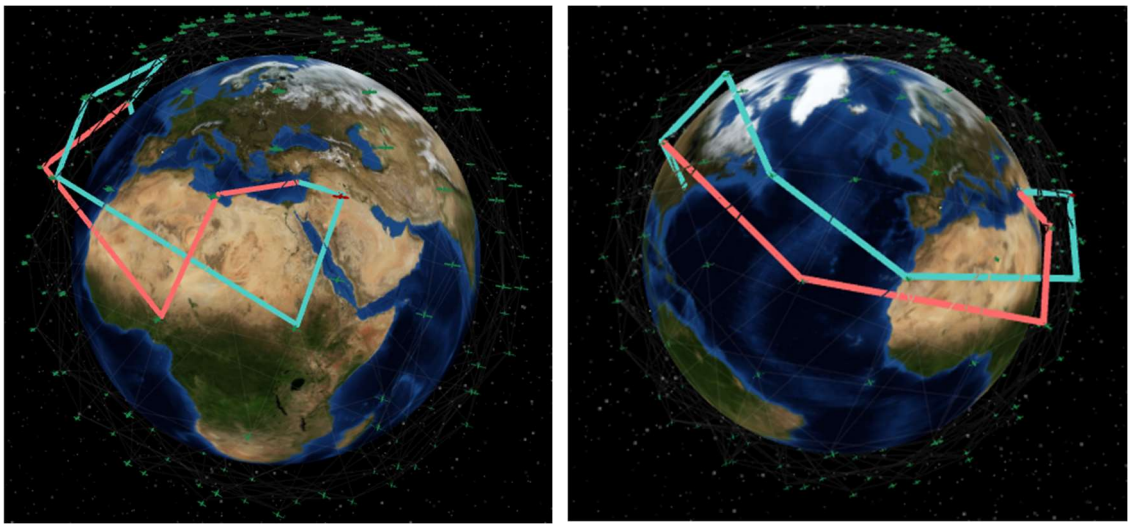

Figure 4-12: Two alternative routing paths toward the gateway at representative constellation states

The visualizations show that satellites located closer to the UE may require longer multi-hop transport paths across the inter-satellite network, while satellites positioned further away may provide shorter end-to-end routes toward the gateway. As such, access-link geometry alone does not determine the resulting service performance. Instead, the constellation routing structure plays a decisive role in shaping the achievable latency characteristics. This behaviour highlights the structural decoupling between access-link feasibility and transport efficiency in distributed satellite networks.

© 2023-2025 5G-STARDUST

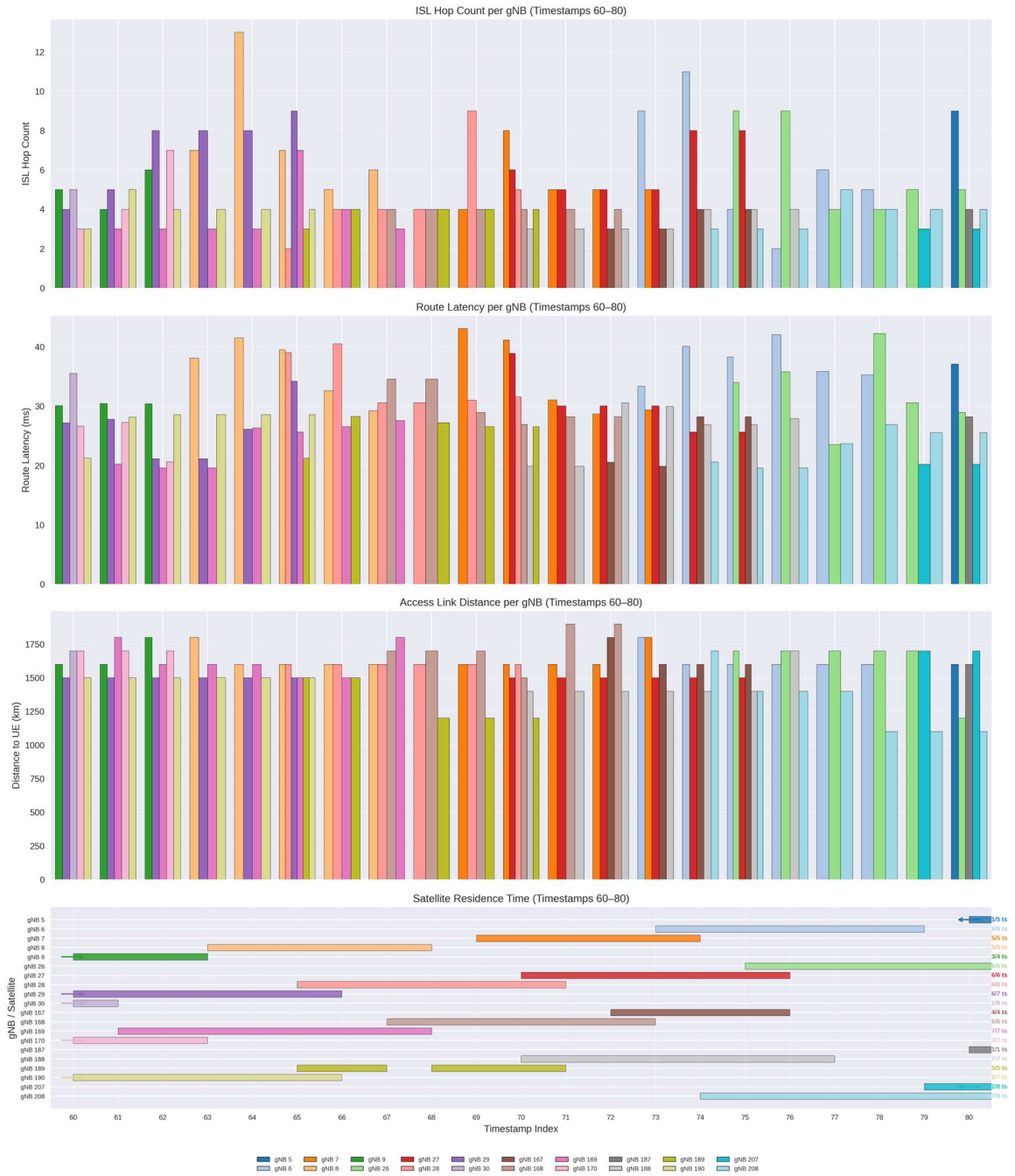

Figure 4-13: Detailed multi-metric comparison during a representative constellation interval including: ISL hop count, routing latency, predicted residence time, and candidate ordering dynamics

4.2.4.3 Detailed Multi-Metric Behaviour

A more detailed view of the mobility dynamics is presented in Figure 4-13, which illustrates the temporal evolution of routing latency, ISL hop count, and predicted residence time for all visible satellites during a representative constellation interval. The upper panel shows substantial variation in hop count across feasible candidates, which directly translates into the latency variability observed in the second panel. These variations arise from the evolving inter-satellite link topology and the relative position of satellites with respect to the gateway.

The predicted residence-time curves reveal that satellites offering extended visibility intervals may correspond to either favourable or unfavourable routing positions. As a result, satellites

© 2023-2025 5G-STARDUST

maximizing connection persistence do not necessarily coincide with latency-optimal transport paths. Furthermore, predicted residence time represents a geometric upper-bound estimate derived from ephemeris-based visibility calculations. In practical deployments, local environmental conditions such as terrain, urban structures, or transient blockage may reduce the effective connectivity interval and therefore introduce uncertainty in long-horizon mobility prediction.

Joint observation of these indicators shows that the satellites providing minimum routing latency, maximum predicted residence time, and minimal hop count frequently represent distinct attachment alternatives. The handover decision therefore reflects a structured balance between competing performance objectives rather than a single dominant criterion. Predictive evaluation of these metrics across successive constellation states enables mobility control strategies that improve latency consistency while preserving sustained connectivity intervals in large-scale NTN deployments.

4.2.5 Conditional Handover Conclusions and Further Work

In this section, we investigated the handover decision problem in 5G NTN mega-constellation networks using a constellation-level digital twin implemented on the Fraunhofer FOKUS OpenLANES platform. The analysis shows that satellite selection during mobility is not limited to maintaining access feasibility but determines the subsequent end-to-end transport path through the constellation and its associated latency characteristics.

The results demonstrate that attachment decisions based solely on access persistence do not necessarily achieve efficient service performance, while latency-optimal selection may increase handover activity. Instead, mobility control in large NTN deployments is inherently multi-criteria, requiring the joint consideration of access feasibility, predicted residence time, and end-to-end transport efficiency. These indicators frequently point to different candidate satellites at the same constellation state, making predictive candidate ordering essential.

Based on these observations, the paper introduced a multi-criteria CHO framework in which candidate ranking is derived from predictive indicators generated by a constellation digital twin. The evaluation shows that transport-aware attachment improves latency consistency, while persistence-aware selection reduces unnecessary handovers, enabling balanced mobility behaviour across evolving constellation conditions.

From a standardization perspective, the presented results indicate that future NTN mobility support may benefit from mechanisms enabling the distribution of predictive network-performance indicators together with conventional CHO configuration parameters. As satellite attachment directly influences end-to-end routing characteristics, mobility preparation procedures could evolve toward conveying ordered candidate sets reflecting both access feasibility and anticipated transport efficiency. Such an evolution would remain compatible with existing CHO principles while extending their applicability to large-scale distributed satellite systems in which routing dynamics represent a dominant performance factor.

Further work will focus on validating the proposed approach under realistic deployment conditions, with particular attention to deviations between predicted residence time and effective connectivity due to environmental obstruction and propagation effects. Additional studies will investigate different constellation configurations and gateway distributions, as well as the integration of predictive candidate ordering into standardized CHO signalling procedures.

© 2023-2025 5G-STARDUST

5 CONCLUSION

Summarizing the main finding of the architectural considerations based on ATSSS: ATSSS is only specified for 3GPP and another non-3GPP connection, or one 5G connection and one 4G. The standard would need to be extended to allow parallel 5G connections across independent 5G base stations beyond CoMP and Dual Connectivity. The control plane procedures specified fulfilling already all use cases identified considering several options of network operators of parallel flows, however, requiring extensions on the identities and identifiers of the connections as presented in Section 2.2. Next to this, for the data path also the PMF timers must be revised to cover the specific satellite delay. At the moment the MP anchor is specified to be at PSA UPF, in order to make efficient use of MP also in NTN-NTN setups it could be beneficial to terminate MP-connections also at UPFs hosted as payload of satellites and forward data from there toward the PSA UPF in single connections mode. However, this represents a secondary use case to the TN-NTN interoperability which represents the main current requirement for the adoption of 5G NTN. For indirect communications one new option introduced is to enable an MC-layer at IAB level so that the relaying node is backhauled by two or more connections that allow for steering, splitting and switching. The inclusion of MC in this implementation of indirect connection requires an update of standard. If it is introduced at IAB level, it needs to be terminated at the serving CU of the IAB-Donor, a termination at UPF is not possible to not break the stack. For indirect communication mechanism for MC handling must be developed to make UEs aware about the multiple paths available and for exchanging QoS information.

We have exhaustively discussed the transport protocols that could be used for MC and have detailed the MPTCP and MPQUIC structure and compared them. MPQUIC is introducing mandatory encryption and builds on UDP, it is an extension of QUIC. MPTCP, on the other hand, builds on top of TCP.

We have introduced four schemes for optimizing multi-connectivity in NTN-TN context: AI/ML scheduling; one that makes use of the predicted system load and tries to avoid predicted capacity gaps; hand-over predictions; hot-swapping that switches the CCA and scheduler out of a selected set; and last a controller for joint MPTCP-MPQUIC traffic. All approaches could present gains in performance and deeply investigated the performance in the selected direct and indirect scenarios in testbeds built using Mininet. Still, for all technologies, more investigations are required. Since MPQUIC is not yet fully specified, although referred to in ATSSS, it gives some freedom to investigate on implementation options for the future, but a clear test on the performance could not be performed. Next steps, besides a deeper investigation of MPQUIC could include additional scheduling algorithms, also based on AI/ML that might be able to further improve the performance.

Concluding the work of Software Defined Network Control, the evaluation has confirmed that an SDN-based management plane for LEO constellations is feasible when supported by hop-count abstraction, feeder diversity, and controlled propagation of management information. The measurements demonstrated that cold-start establishment, feeder failure handling, and recovery can all be realized through localized neighbour exchanges, with redundancy providing the necessary robustness to sustain operation under degraded conditions. At the same time, the results underlined the structural limitations of the approach: the dependency on feeders as gateways to the terrestrial controller which on their term are prone to weather effects, the propagation delays that lengthen control-plane responsiveness, and the consequent need for satellites to retain local reaction mechanisms in parallel with global coordination.

© 2023-2025 5G-STARDUST

Additionally, by simulating a mega-constellation using the Fraunhofer FOKUS OpenLANES toolkit, an evaluation of the conditional handover procedures as currently standardized by 3GPP was proposed including new handover criteria based on maximum potential residence time and end-to-end data path delay, solution which provides a better response to the service requirements. The results show that the such multi-metric solutions based on the specific mega-constellation characteristics are better suited for the further deployments than the current received signal strength-based ones.

As further work, the next phase will extend the simulations by embedding a management protocol such as NETCONF, thereby enabling explicit modelling of configuration and state synchronization across the constellation. The integration of the SDN management plane with a 5G/6G data plane will be studied to assess end-to-end service continuity under realistic traffic loads. Beyond simulation, an implementation of SDN switching elements on representative space payloads is planned, with the objective of validating feasibility in a hardware environment. Finally, the integration of these components into a laboratory breadboard testbed will provide experimental confirmation, complementing the simulation results with reproducible platform-level trials. Together, these steps will advance the concept from simulated feasibility towards deployable architecture within.

© 2023-2025 5G-STARDUST

REFERENCES

[1] 5G-STARDUST Deliverable 3.1, “System Requirements Analysis and Specifications”, v.1.3, 07.2023

[2] 5G-STARDUST Deliverable 3.2, “End-to-end ground and space integrated architecture”, v1.0, 02.2024

[3] 5G-STARDUST Deliverable 2.1, “Scenario, use cases and services”, Version: 3.0.1, 08.2023

[4] 3GPP TS 38.101-1 “User Equipment (UE) radio transmission and reception; Part 1: Range 1 Standalone” V18.5.0, 03.2024

[5] 3GPP TS 38.101-2 "User Equipment (UE) radio transmission and reception; Part 2: Range 2 Standalone", V18.5.0, 03.2024

[6] 3GPP TS 38.101-5 "User Equipment (UE) radio transmission and reception; Part 5: Satellite access Radio Frequency (RF) and performance requirements", V18.5.0, 03.2024

[7] B. Barth, R. Sebastian, T. De Cola and A. Guidotti, "NTN Architecture for PPDR in 5G-Advanced," 2024 IEEE Future Networks World Forum (FNWF), Dubai, United Arab Emirates, 2024, pp. 351-356, doi: 10.1109/FNWF63303.2024.11028781.

[8] 3GPP TS 24.193, “Access Traffic Steering, Switching and Splitting (ATSSS) Stage 3”, v18.5.0, 03.2024

[9] 3GPP TS 23.501, “System architecture for the 5G System (5GS), Stage 2”, v18.5.0, 03.2024

[10] 3GPP TS 24.526, “User Equipment (UE) policies for 5G System (5GS); Stage 3”, v18.6.0, 04.2024

[11] IETF RFC 8684, “TCP Extensions for Multipath Operation with Multiple Addresses”, March 2020

[12] IETF draft-ietf-quic-multipath-07, “Multipath Extension for QUIC”, 04.2024, expiring 10.2024

[13] M. Luglio, M. Quadrini, S. P. Romano, C. Roseti and F. Zampognaro, "Enabling an efficient satellite-terrestrial hybrid transport service through a QUIC-based proxy function," 2020 International Symposium on Networks, Computers and Communications (ISNCC), Montreal, QC, Canada, 2020, pp. 1-6, doi: 10.1109/ISNCC49221.2020.9297334.

[14] 5G-STARDUST Deliverable 5.3, “Preliminary report on AI-data driven networking and QoS management”, to be submitted in June 2024

[15] T. de Cola et al., 5G-STARDUST Deliverable 6.1: “PoC functional architecture document”, v. 1.0, 04.2024

[16] IETF RFC 9221, “An Unreliable Datagram Extension to QUIC”, 03.2022

[17] C. Pupiales, D. Laselva, Q. De Coninck, A. Jain and I. Demirkol, "Multi-Connectivity in Mobile Networks: Challenges and Benefits," in IEEE Communications Magazine, vol. 59, no. 11, pp. 116-122, November 2021, doi: 10.1109/MCOM.111.2100049;

[18] 3GPP TS 37.340. “Multi-connectivity; Overall description; Stage-2”, v18.1.0., March 2024, https://www.3gpp.org/;

[19] Pan, Cunhua, et al. "User-centric C-RAN architecture for ultra-dense 5G networks: Challenges and methodologies." IEEE Communications Magazine 56.6 (2018): 14-20;

[20] 3GPP TS 24.193 Technical Specification Group Core Network and Terminals. 5G

 © 2023-2025 5G-STARDUST

System Access Traffic Steering, Switching and Splitting (ATSSS); v.18.5.0, April 2024, https://www.3gpp.org/;

[21] Fraunhofer FOKUS Open5GCore toolkit, www.open5Gcore.org;

[22] T. Sylla, L. Mendiboure, S. Maaloul, H. Aniss, M. A. Chalouf, and S. Delbruel, “Multi-Connectivity for 5G Networks and Beyond: A Survey,” Sensors (Basel), vol. 22, no. 19, p. 7591, Oct. 2022, doi: 10.3390/s22197591;

[23] IETF RFC 8684, “TCP Extensions for Multipath Operation with Multiple Addresses”, March 2020, https://datatracker.ietf.org/doc/html/rfc8684;

[24] IETF, “Multipath Extension for QUIC”, draft-ietf-quic-multipath-014, https://datatracker.ietf.org/doc/draft-ietf-quic-multipath/, work in progress;

[25] 3GPP TS 33.501, “Security architecture and procedures for 5G System”, v18.5.0, April 2024, https://www.3gpp.org/;

[26] T. T. Nguyen, M. H. Vu, P. L. Nguyen, P. T. Do, and K. Nguyen, ‘A Q-learning-based Multipath Scheduler for Data Transmission Optimization in Heterogeneous Wireless Networks’, in 2023 IEEE 20th Consumer Communications & Networking Conference (CCNC), Las Vegas, NV, USA: IEEE, Jan. 2023, pp. 573–578. doi: 10.1109/CCNC51644.2023.10060683.

[27] IETF RFC 8684, “TCP Extensions for Multipath Operation with Multiple Addresses”, March 2020

[28] IETF RFC 6182, “Architectural Guidelines for Multipath TCP Development”, 12.2018

[29] T. Viernickel, A. Froemmgen, A. Rizk, B. Koldehofe and R. Steinmetz, "Multipath QUIC: A Deployable Multipath Transport Protocol," 2018 IEEE International Conference on Communications (ICC), Kansas City, MO, USA, 2018, pp. 1-7, doi: 10.1109/ICC.2018.8422951.

[30] IETF RFC 8095, “Services Provided by IETF Transport Protocols and Congestion Control Mechanisms”, 01.2020

[31] https://tools.ietf.org/html/draft-pauly-taps-transport-security-02#section-4

[32] IETF RFC 9279, “Internet Group Management Protocol Version 3 (IGMPv3) and Multicast Listener Discovery Version 2 (MLDv2) Message Extension”, 08.2022

[33] https://quic-go.net/docs/connect-udp/

[34] https://datatracker.ietf.org/doc/html/draft-huitema-quic-in-space-01, draft-ietf-tsvwg-careful-resume-17

[35] https://datatracker.ietf.org/meeting/122/materials/slides-122-tiptop-key-exchange-customization-for-tiptop-00,

[36] https://datatracker.ietf.org/doc/html/draft-zdm-tvr-applicability

[37] https://github.com/ietf-wg-masque/draft-ietf-masque-connect-udp/wiki/Implementations

[38] https://docs.google.com/spreadsheets/d/1D0tW89vOoaScs3IY9RGC0UesWGAwE6xyLk0l4JtvTVg/edit?pli=1&gid=2079541679#gid=2079541679

[39] Bonaventure, O., et al. "Improving Multipath TCP Backup Subflows." IETF draft-bonaventure-mptcp-backup-00, 2015. https://datatracker.ietf.org/doc/html/draft-bonaventure-mptcp-backup-00

[40] Viernickel, T., et al. "Efficient Continuous Latency Monitoring with eBPF." International Conference on Passive and Active Measurement (PAM), 2023. DOI: 10.1007/978-3-031-28486-1_9

 © 2023-2025 5G-STARDUST

[41] Ferlin, S., et al. "On the Benefits of Applying Experimental Design to Improve Multipath TCP Performance." ACM CoNEXT, 2013. https://conferences.sigcomm.org/co-next/2013/program/p393.pdf

[42] Lantz, B., et al. "A Network in a Laptop: Rapid Prototyping for Software-Defined Networks." ACM HotNets, 2010. https://conferences.sigcomm.org/hotnets/2010/papers/a19-lantz.pdf

[43] De Coninck, Q., & Bonaventure, O. "Multipath TCP." RFC 8684, IETF, 2020. https://datatracker.ietf.org/doc/html/rfc8684

[44] Jay, N., et al. "A Deep Reinforcement Learning Perspective on Internet Congestion Control." ICML, 2019. https://proceedings.mlr.press/v97/jay19a.html

[45] Multipath TCP Development Repository. GitHub - multipath-tcp/mptcp_net-next. https://github.com/multipath-tcp/mptcp_net-next

[46] Li, M., et al. "SmartCC: A Reinforcement Learning Approach for Multipath TCP Congestion Control in Heterogeneous Networks." IEEE Journal on Selected Areas in Communications 37.11 (2019): 2620-2633. DOI: 10.1109/JSAC.2019.2933761

[47] Tang, M., et al. "Experience-Driven Congestion Control: When Multi-Path TCP Meets Deep Reinforcement Learning." IEEE Journal on Selected Areas in Communications 37.6 (2019): 1325-1342. DOI: 10.1109/JSAC.2019.2907831

[48] Maliha, M., Habibi, G., & Atiquzzaman, M. "A Survey on Congestion Control and Scheduling for Multipath TCP: Machine Learning vs Classical Approaches." Annals of Computer Science and Information Systems 35 (2023): 983-992. DOI: 10.15439/2023F983

[49] Sutton, R. S., & Barto, A. G. (2018). Reinforcement learning: An introduction (2nd ed.). MIT Press. https://web.stanford.edu/class/psych209/Readings/SuttonBartoIPRLBook2ndEd.pdf

[50] Achiam, J. (2018). Spinning Up in Deep Reinforcement Learning. OpenAI. https://spinningup.openai.com/en/latest/spinningup/rl_intro.html

[51] Mnih, V., et al. "Human-level control through deep reinforcement learning." Nature 518.7540 (2015): 529-533. DOI: 10.1038/nature14236

[52] S. Barré, C. Paasch, and O. Bonaventure, “MultiPath TCP: From Theory to Practice,” in NETWORKING 2011, J. Domingo-Pascual, P. Manzoni, S. Palazzo, A. Pont, and C. Scoglio, Eds., Berlin, Heidelberg: Springer, 2011, pp. 444–457. doi: 10.1007/978-3-642-20757-0_35.

[53] “Study on New Radio (NR) to support non-terrestrial networks” 3GPP, Technical Specification Group Radio Access Network (Release 15), 3GPP TR 38.811 v15.4.0 (2020-09);

[54] 3GPP TS 23.288: "Architecture enhancements for 5G System (5GS) to support network data analytics services", V18.5.0, 03.2024.

[55] S. Ferlin, Ö. Alay, O. Mehani, et R. Boreli, « BLEST: Blocking estimation-based MPTCP scheduler for heterogeneous networks », in 2016 IFIP Networking Conference (IFIP Networking) and Workshops, mai 2016, p. 431-439. doi: 10.1109/IFIPNetworking.2016.7497206.

[56] Y. Lim, E. M. Nahum, D. Towsley, et R. J. Gibbens, « ECF: An MPTCP Path Scheduler to Manage Heterogeneous Paths », in Proceedings of the 13th International Conference on emerging Networking EXperiments and Technologies, in CoNEXT ’17. New York, NY, USA: Association for Computing Machinery, nov. 2017, p. 147-159. doi: 10.1145/3143361.3143376.

 © 2023-2025 5G-STARDUST

[57] L. Wang, Z. Wang, Z. Deng, J. Zhang, et Y. Gao, « ALCS: An Adaptive Latency Compensation Scheduler for Multipath TCP in Satellite-Terrestrial Integrated Networks », 11 mars 2025, arXiv: arXiv:2503.07973. doi: 10.48550/arXiv.2503.07973.

[58] M. Hoyhtya, M. Corici, S. Covaci, and M. Guta, “5G and beyond for new space: Vision and research challenges,” in Advances in Communications Satellite Systems. Proceedings of the 37th International Communications Satellite Systems Conference (ICSSC–2019). IET, pp. 1-16.

[59] A. Vanelli-Coralli, A. Guidotti, T. Foggi, G. Colavolpe, and G. Montorsi, “5g and beyond 5g non-terrestrial networks: trends and research challenges,” 2020 IEEE 3rd 5G World Forum (5GWF), pp. 163—169

[60] Louis J. Ippolito, "Transmission Impairments," in Satellite Communications Systems Engineering: Atmospheric Effects, Satellite Link Design and System Performance , Wiley, 2017, pp.87-137

[61] U. D. Black and U. N, Black. IP routing protocols: RIP, OSPF, BGP, PNNI, and Cisco routing protocols. Prentice Hall Professional 2000.

[62] M. N. Alslaim, et al, “A comparative study of manet routing protocols,” in The Third International Conference on e-Technologies and Networks for Development (ICeND2014). IEEE, pp. 178-182.

[63] A. Agarwal, C. Iskander, and R. Shankar, “Survey of network on chip (noc) architectures & contributions.” Journal of Engineering, Computing and Architecture 3, no. 1, pp. 21—27.

[64] Cao, Xiaoli, et al. "Dynamic routings in satellite networks: An overview." Sensors 22, no. 12 (2022): 4552.

[65] Previdi, S., E. Aries, and D. Afanasiev, IETF "RFC 9087: Segment Routing Centralized BGP Egress Peer Engineering." (2021).

[66] Jiang, Weiwei. "Software defined satellite networks: A survey." Digital Communications and Networks 9.6 (2023): 1243-1264.

[67] Köhler, Stefan, and Andreas Binzenhöfer. "MPLS traffic engineering in OSPF networks - a combined approach." Teletraffic Science and Engineering. Vol. 5. Elsevier, 2003. 21-30.

[68] Li, Tong, Jinqiang Chen, and Hongyong Fu. "Application scenarios based on SDN: an overview." Journal of Physics: Conference Series. Vol. 1187. No. 5. IOP Publishing, 2019.

[69] Raj, Alex, et. al, "A survey of IP and multiprotocol label switching fast reroute schemes." Computer Networks 51.8 (2007): 1882-1907.

[70] Dao, N.N. et al., 2024. A review on new technologies in 3GPP standards for 5G access and beyond. Computer Networks, p.110370.

[71] Maravić, Igor, and Aleksandra Smiljanić. "MPLS implementation for the Linux kernel." 2012 IEEE 13th International Conference on High Performance Switching and Routing. IEEE, 2012.

[72] Kriezis, Anargyros, and Whitney Lohmeyer. "An Overview and Assessment of Today's US Satellite Telemetry, Tracking and Telecommand (TT&C) Spectrum Allocation Needs." Tracking and Telecommand (TT&C) Spectrum Allocation Needs (April 10, 2023)

[73] Mellia, Marco, Nur Zincir-Heywood, and Yixin Diao. "Overview of Network and Service Management." Communication Networks and Service Management in the Era of Artificial Intelligence and Machine Learning (2021): 1-18.

[74] Fraunhofer FOKUS OpenLanes: Large Nework Emulator, high level toolkit description,

 © 2023-2025 5G-STARDUST

https://connectivity.esa.int/sites/default/files/ADS%20public%20ScyLight%202, last visited on 28.04.2023;

[75] 3GPP TS 24.526, https://www.3gpp.org/dynareport/24526.htm, last visited on 28.04.2024

[76] Roth, Manuel, Hartmut Brandt, and Hermann Bischl. "Implementation of a geographical routing scheme for low Earth orbiting satellite constellations using intersatellite links." International Journal of Satellite Communications and Networking 39.1 (2021): 92-107.

[77] Korçak, Ömer, and Fatih Alagöz. "Priority-based adaptive shortest path routing in IP over LEO satellite networks." 23rd AIAA International communications satelllite systems Conference. 2005.

[78] Z. Han, et. al, "Dynamic Routing for Software-Defined LEO Satellite Networks based on ISL Attributes," 2021 IEEE Global Communications Conference (GLOBECOM), Madrid, Spain, 2021, pp. 1-6

[79] Q. De Coninck and O. Bonaventure, "Multipath QUIC: Design and Evaluation," Proceedings of the 13th International Conference on Emerging Networking EXperiments and Technologies (CoNEXT '17), ACM, pp. 160–166, 2017. DOI: 10.1145/3143361.3143370

[80] Y. Liu, Y. Ma, Q. De Coninck, O. Bonaventure, C. Huitema, and M. Kühlewind, "Multipath Extension for QUIC," IETF Internet-Draft draft-ietf-quic-multipath-19, January 2026. Available: https://datatracker.ietf.org/doc/draft-ietf-quic-multipath/

[81] Satellogic SA, "orbit-predictor: Python library to propagate satellite orbits," v1.15.2, MIT License. Available: https://github.com/satellogic/orbit-predictor

[82] Python Software Foundation, "GlobalInterpreterLock," Python Wiki. Available: https://wiki.python.org/moin/GlobalInterpreterLock

[83] Multipath TCP Project. mptcp.dev: Multipath TCP (MPTCP) community website. Website. URL https://www.mptcp.dev/

[84] Jana Iyengar and Martin Thomson. QUIC: A UDP-Based Multiplexed and Secure Transport. RFC 9000, May 2021. URL https://www.rfc-editor.org/info/rfc9000

[85] Quentin De Coninck and Olivier Bonaventure. Multipath QUIC: Design and Evaluation. In Proceedings of the 13th International Conference on Emerging Networking EXperiments and Technologies, CoNEXT '17, page 160–166, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450354226. doi:10.1145/31 43361.3143370.

[86] Private Octopus Inc. picoquic: A Minimal Implementation of the QUIC Protocol. GitHub repository. URL https://github.com/private-octopus/picoquic

[87] Mininet Project. Mininet: Instant Virtual Networks on your Laptop (or other PC). Website. URL https://mininet.org

[88] 3GPP, "NR; Radio Resource Control (RRC); Protocol specification," Technical Specification TS 38.331, Release 16, 2021.

[89] 3GPP, "NR; NR and NG-RAN Overall Description; Stage 2," Technical Specification TS 38.300, Release 16, 2021.

[90] 3GPP, "Solutions for NR to support Non-Terrestrial Networks (NTN)," Technical Report TR 38.821, Release 16, 2021.

[91] 3GPP, "NR; Radio Resource Control (RRC); Protocol specification," Technical Specification TS 38.331, Release 17, 2022.

[92] 3GPP, "NR; NR and NG-RAN Overall Description; Stage 2," Technical Specification TS 38.300, Release 17, 2022.

 © 2023-2025 5G-STARDUST

[93] E. Juan, M. Lauridsen, J. Wigard, and P. Mogensen, "Performance Evaluation of the 5G NR Conditional Handover in LEO-based Non-Terrestrial Networks," in Proc. IEEE WCNC, 2022.

[94] J. Li, C. Wang, C. Wang, W. Wang, and J. Zhen, "Beam Handover Based on Multi-Attribute Decision in User-Centric LEO Satellite Networks," in Proc. IEEE/CIC ICCC, 2022, pp. 314-319.

[95] M. Hosseinian, J. P. Choi, S. H. Chang, and J. Lee, "Review of 5G NTN standards development and technical challenges for satellite integration with the 5G network," IEEE Aerosp. Electron. Syst. Mag., vol. 36, no. 8, pp. 22-31, 2021.

[96] J. Wang, F. Xu, and F. Sun, "Benchmarking of routing protocols for layered satellite networks," in Proc. Multiconference on Computational Engineering in Systems Applications, vol. 2, 2006, pp. 1087-1094.

[97] H. Han, Y. Zhao, J. Wei, and S. Cao, "GHRP: An Efficient Routing Protocol for Satellite Networks," in Proc. 11th Int. Conf. Wireless Communications and Signal Processing (WCSP), 2019, pp. 1-5.

[98] H. Xie, Y. Zhan, G. Zeng, and X. Pan, "LEO Mega-Constellations for 6G Global Coverage: Challenges and Opportunities," IEEE Access, vol. 9, pp. 164223-164244, 2021.

[99] M. Sheng, D. Zhou, R. Liu, Y. Wang, and J. Li, "Resource mobility in space information networks: Opportunities, challenges, and approaches," IEEE Network, vol. 33, no. 1, pp. 128-135, 2018.

[100] M. M. H. Roth, H. Brandt, and H. Bischl, "Distributed SDN-based Load-balanced Routing for Low Earth Orbit Satellite Constellation Networks," in Proc. ASMS/SPSC, 2022, pp. 1-8.

[101] N. Wang, C. Dong, L. Liu, M. Wang, and B. Liang, "An SDN based highly reliable in-band control framework for LEO mega-constellations," in Proc. IEEE ICCCS, 2021, pp. 970-975.

[102] D. Gopi, S. Cheng, and R. Huck, "Comparative analysis of SDN and conventional networks using routing protocols," in Proc. CITS, 2017, pp. 108-112.

[103] M. Jia, X. Wang, S. Zhang, B. Yi, and M. Huang, "Software defined network based fault tolerant routing mechanism for satellite networks," Journal of Computer Applications, vol. 39, no. 6, p. 1772, 2019.

[104] Fraunhofer FOKUS OpenLANES toolkit, www.openlanes.net

© 2023-2025 5G-STARDUST

APPENDIX A – MULTI-CONNECTIVITY PROTOCOL STACKS

In the following the protocol stacks for all considered Multi-Connectivity (MC) topologies are presented that have been derived from the 5G-STARDUST baseline use cases and architecture, D3.1 and D3.2. The illustration is following the color-code:

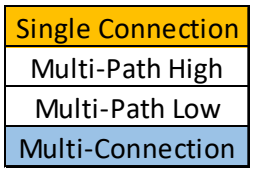

Figure 5-1: Protocol stacks, color-coding

Which means that orange is single connections which is split, switched or steered by the MP-High and MP-Low layer colored in white. The blue connection is the MC which means that these parts of the protocol stacks must be available at least twice, e.g., on TN and NTN path. The topologies show always an NTN and an TN path, but the stacks generally apply to other combination of paths (e.g., NTN-NTN).

Baseline Single Link Connection

For a better overview, we present in the following again the reference single connection stacks as presented in D3.1.

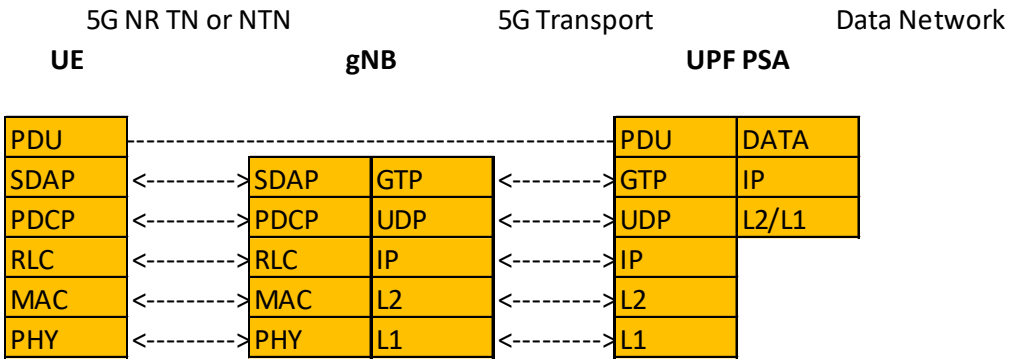

Figure 5-2: Protocol stack, baseline 5G-UP Stack single connectivity [1]

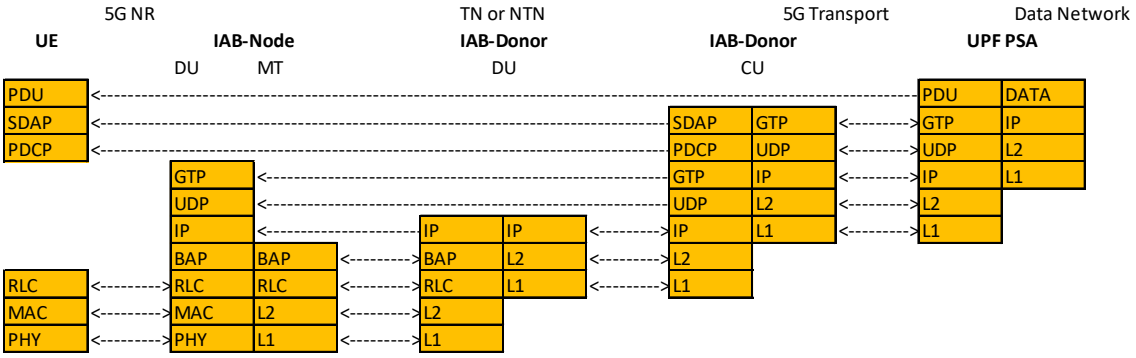

Figure 5-3: Protocol stack, baseline 5G-IAB Stack single connectivity [1]

© 2023-2025 5G-STARDUST

Direct Connection with Multi-Connectivity Layer at UE

This is the baseline ATSSS case enhanced to allow parallel 5G-connections.

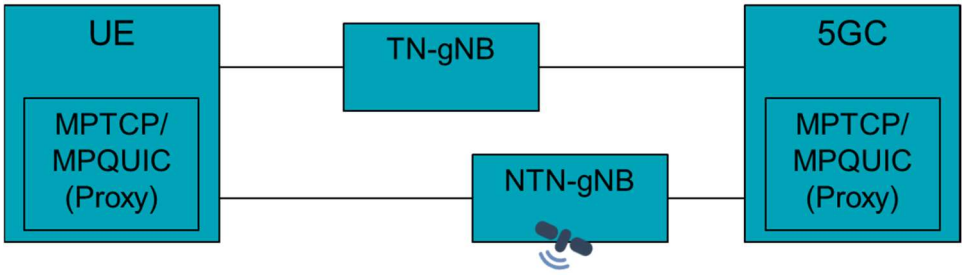

Figure 5-4: Direct connection with MC-layer at UE

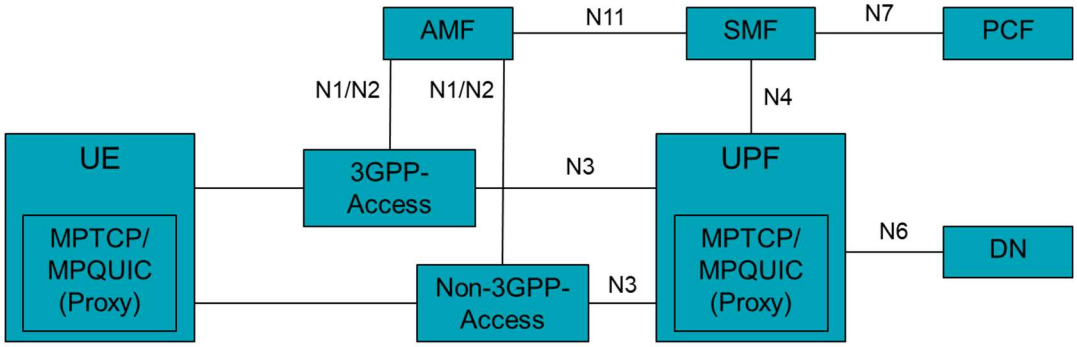

Figure 5-5: ATSSS architecture [9]

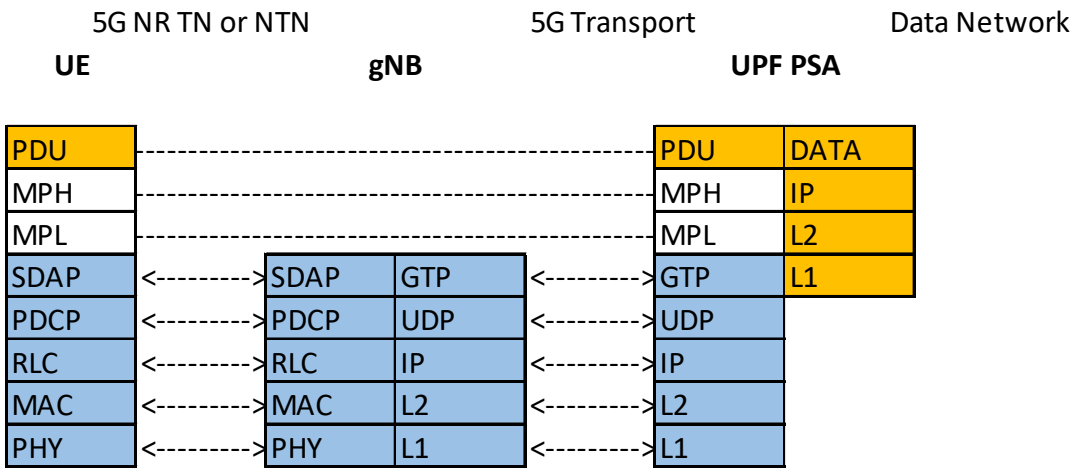

Figure 5-6: Protocol stack, direct connection with MC-layer at UE

© 2023-2025 5G-STARDUST

Direct Connection with Multi-Connectivity Layer at UE, UPF in Space

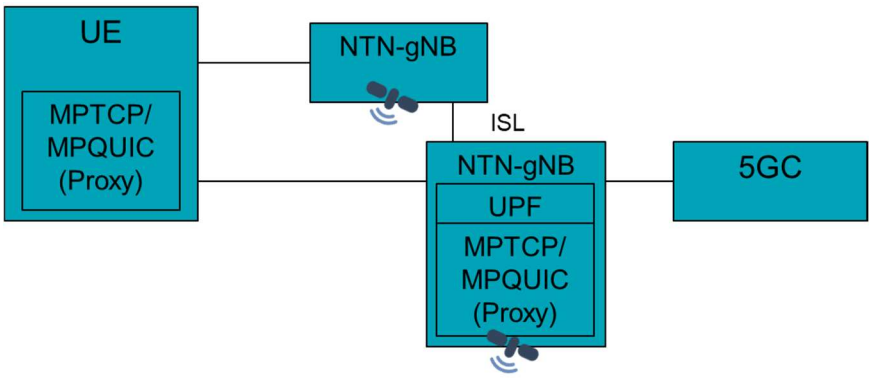

Figure 5-7: Direct Connection with MC-layer at UE, UPF in Space

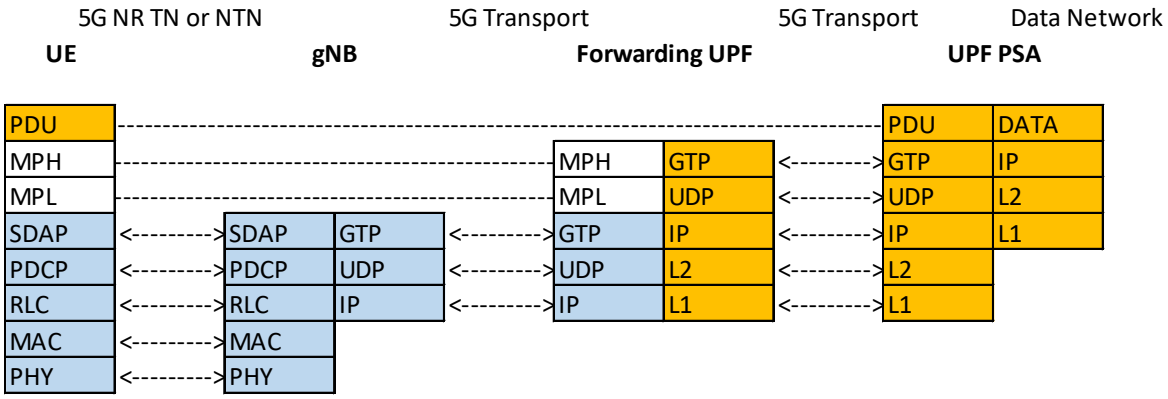

Figure 5-8: Protocol stack, direct Connection with MC-layer at UE, UPF in Space

Indirect Connection with Multi-Connectivity Layer at UE

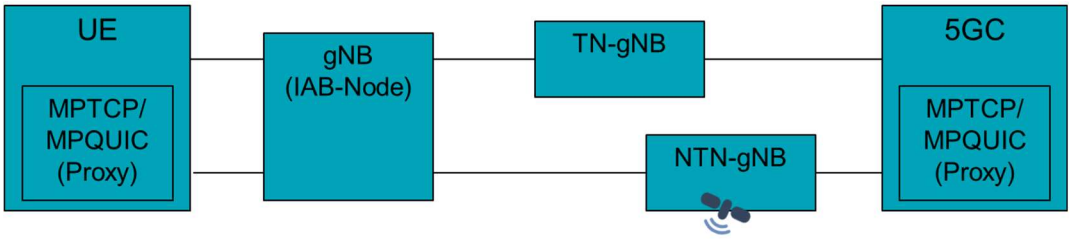

Figure 5-9: Indirect connection with MC-layer at UE

© 2023-2025 5G-STARDUST

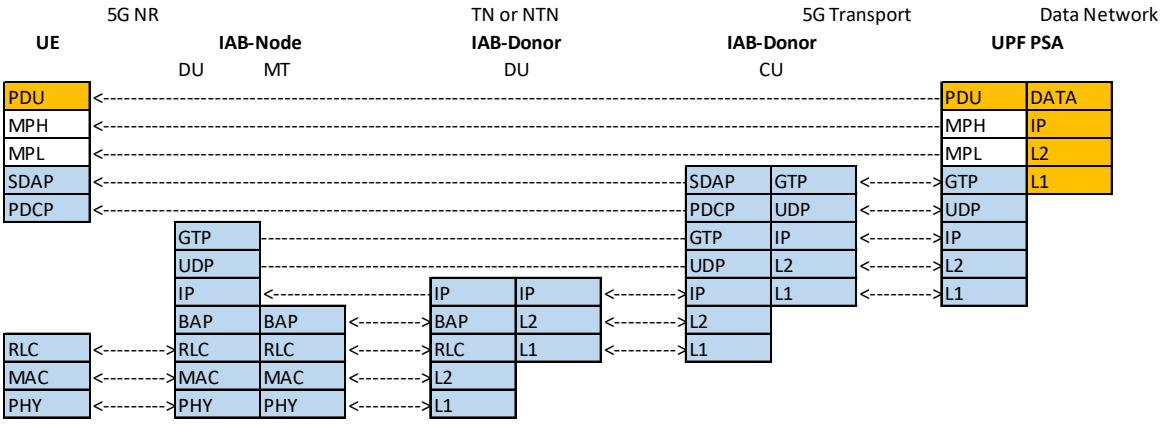

Figure 5-10: Protocol stack, indirect connection with MC Layer at UE

Indirect Connection with Multi-Connectivity Layer at UE and UPF in Space

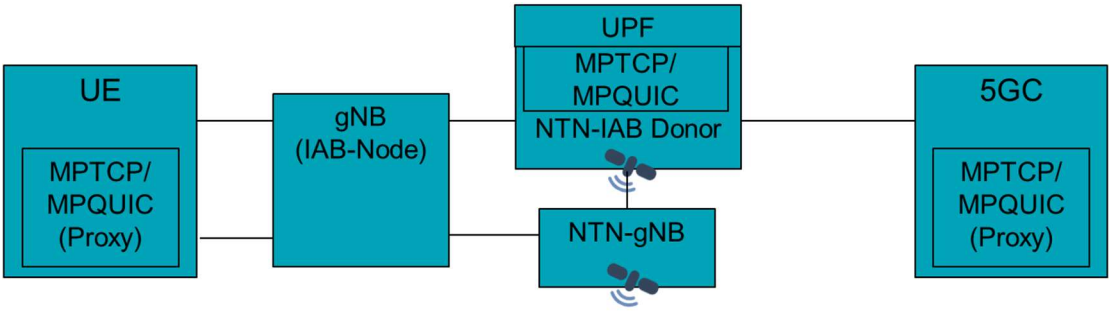

Figure 5-11: Indirect connection with MC-layer at UE, UPF in Space

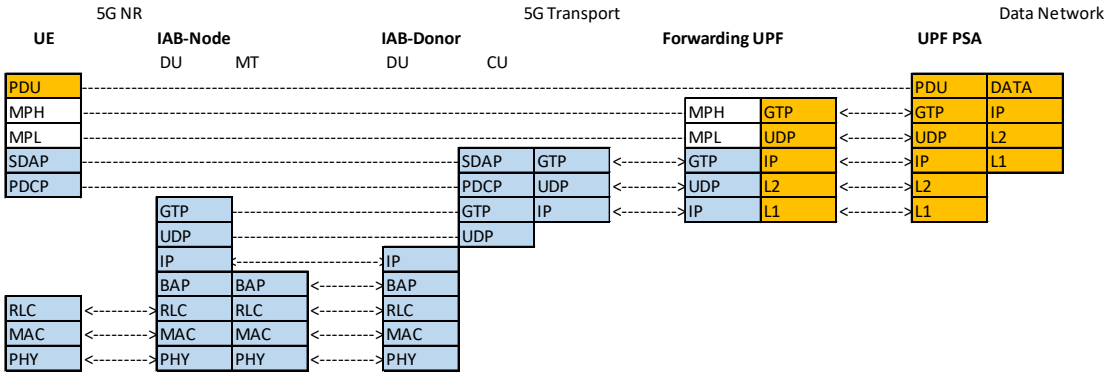

Figure 5-12: Protocol stack, indirect connection with MC-layer at UE, UPF in Space

© 2023-2025 5G-STARDUST

Indirect Connection with Multi-Connectivity Layer at IAB-Node

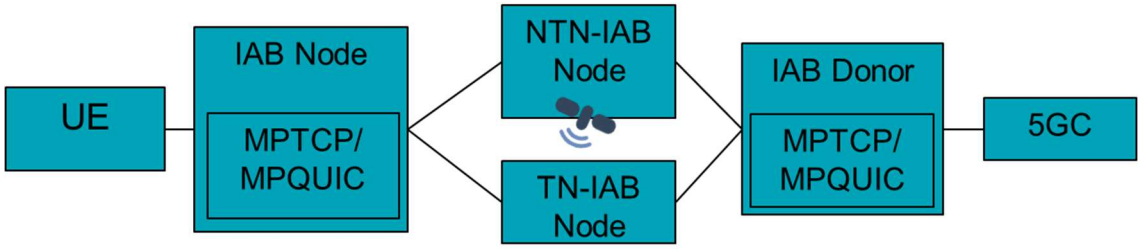

Figure 5-13: Indirect connection with MC-layer at IAB-node, donor on ground

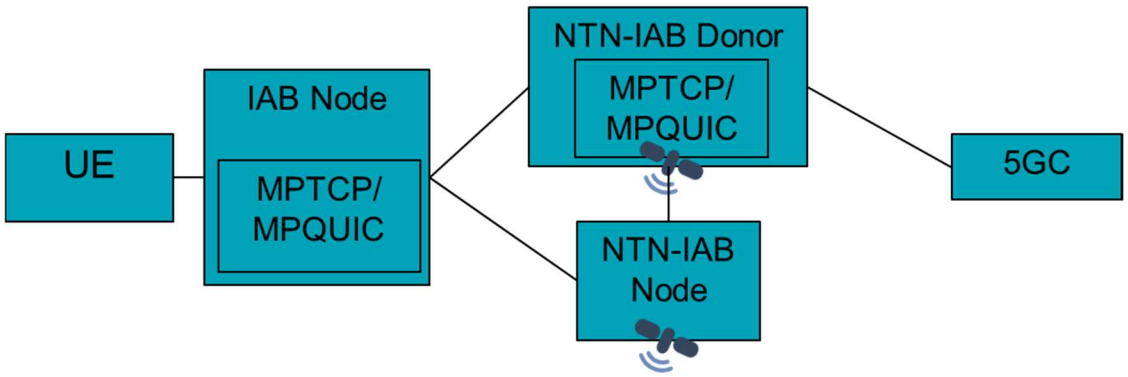

Figure 5-14: Indirect connection with MC-layer at IAB-node, donor in space

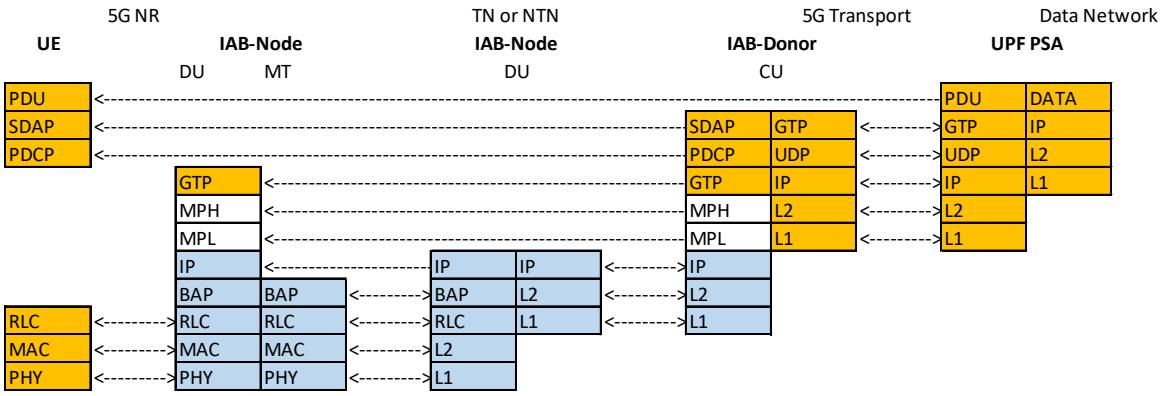

Figure 5-15: Protocol stack, indirect connection with MC-layer at IAB-node

© 2023-2025 5G-STARDUST

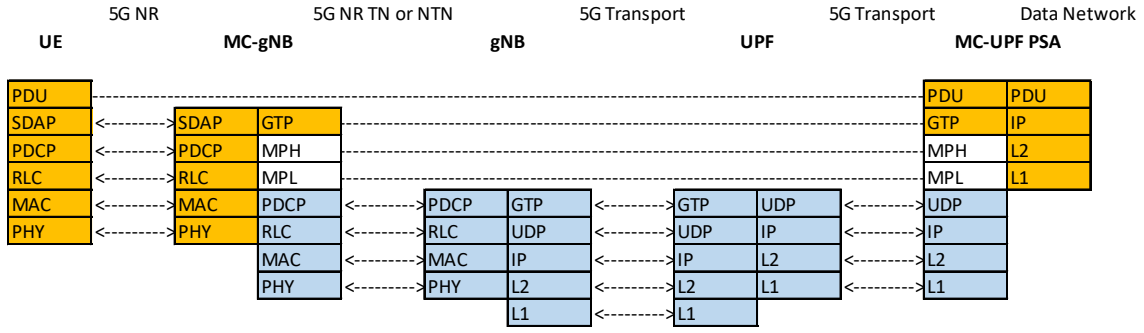

Figure 5-16: Protocol stack, indirect connection with MC-layer at IAB-node, PoC

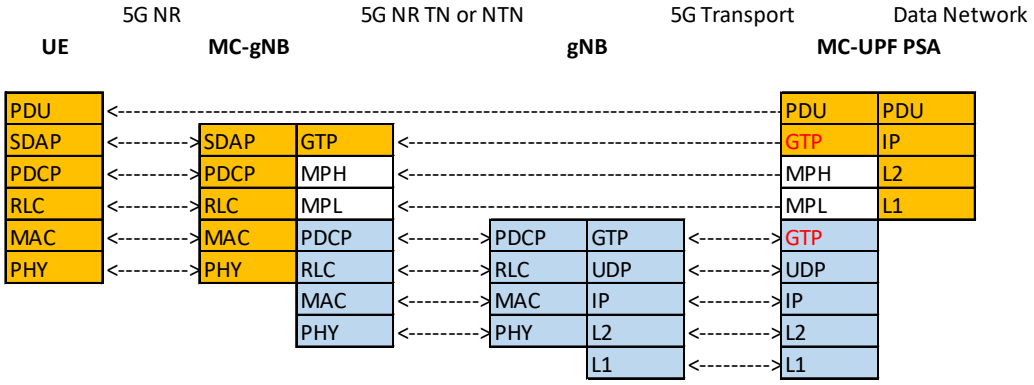

Figure 5-17: Protocol stack, indirect connection with MC-layer at IAB-node, GTP problem

Indirect Connection with Multi-Connectivity Layer at forwarding UPF

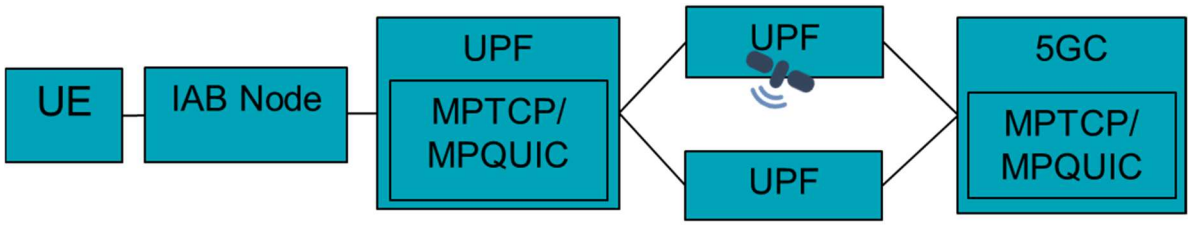

Figure 5-18 : Indirect connection with MC-layer at forwarding UPF, UPF PSA on-ground

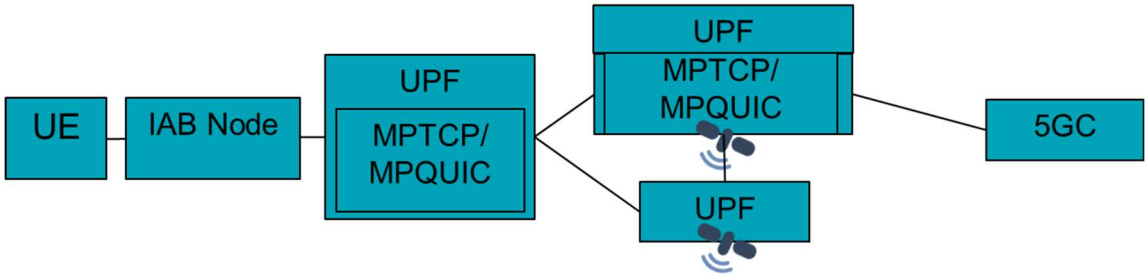

Figure 5-19 : Indirect connection with MC-layer at forwarding UPF, UPF PSA in space

© 2023-2025 5G-STARDUST

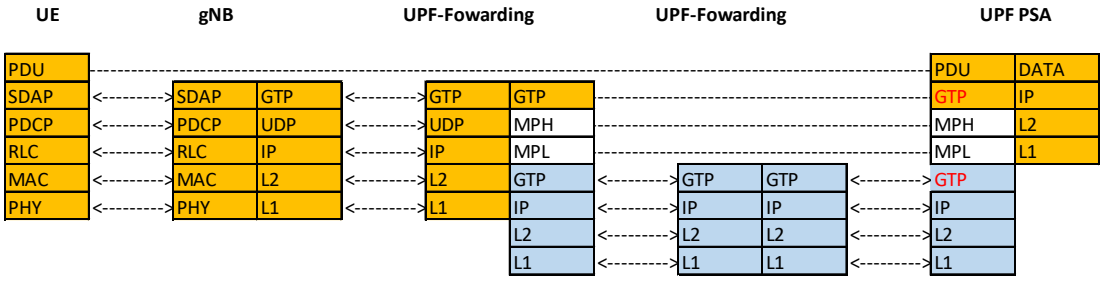

Figure 5-20 : Protocol stack, Indirect connection with MC-layer at forwarding UPF

© 2023-2025 5G-STARDUST