ETSI TR 104 006 V1.1.1 (2025-01)

TECHNICAL REPORT

Rail Telecommunications (RT);
Future Railway Mobile Communication System (FRMCS);
Study on Onboard Radio Interface (OB$_{RAD}$)

Reference

DTR/RT-0081

Keywords

FRMCS, interface, protocol, radio, railways

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - APE 7112B
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° w061004871

Important notice

The present document can be downloaded from the
ETSI Search & Browse Standards application.

The present document may be made available in electronic versions and/or in print. The content of any electronic and/or print versions of the present document shall not be modified without the prior written authorization of ETSI. In case of any existing or perceived difference in contents between such versions and/or in print, the prevailing version of an ETSI deliverable is the one made publicly available in PDF format on ETSI deliver repository.

Users should be aware that the present document may be revised or have its status changed, this information is available in the Milestones listing.

If you find errors in the present document, please send your comments to the relevant service listed under Committee Support Staff.

If you find a security vulnerability in the present document, please report it through our Coordinated Vulnerability Disclosure (CVD) program.

Notice of disclaimer & limitation of liability

The information provided in the present deliverable is directed solely to professionals who have the appropriate degree of experience to understand and interpret its content in accordance with generally accepted engineering or other professional standard and applicable regulations.
No recommendation as to products and services or vendors is made or should be implied.
No representation or warranty is made that this deliverable is technically accurate or sufficient or conforms to any law and/or governmental rule and/or regulation and further, no representation or warranty is made of merchantability or fitness for any particular purpose or against infringement of intellectual property rights.
In no event shall ETSI be held liable for loss of profits or any other incidental or consequential damages.

Any software contained in this deliverable is provided "AS IS" with no warranties, express or implied, including but not limited to, the warranties of merchantability, fitness for a particular purpose and non-infringement of intellectual property rights and ETSI shall not be held liable in any event for any damages whatsoever (including, without limitation, damages for loss of profits, business interruption, loss of information, or any other pecuniary loss) arising out of or related to the use of or inability to use the software.

Copyright Notification

Contents

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables may have been declared to ETSI. The declarations pertaining to these essential IPRs, if any, are publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI IPR online database.

Pursuant to the ETSI Directives including the ETSI IPR Policy, no investigation regarding the essentiality of IPRs, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

DECT™, **PLUGTESTS™**, **UMTS™** and the ETSI logo are trademarks of ETSI registered for the benefit of its Members. **3GPP™**, **LTE™** and **5G™** logo are trademarks of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners. **oneM2M™** logo is a trademark of ETSI registered for the benefit of its Members and of the oneM2M Partners. **GSM**® and the GSM logo are trademarks registered and owned by the GSM Association.

Foreword

This Technical Report (TR) has been produced by ETSI Technical Committee Railway Telecommunications (RT).

Modal verbs terminology

In the present document "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

Executive summary

The goal of the present document is to study and analyse requirements on OB_{RAD} interface captured in UIC FRMCS TOBA FRS [i.1], UIC FRMCS SRS [i.3] and other relevant UIC specifications to propose potential solution(s) and possible technical realization(s), covering the physical and functional OB_{RAD} interface as well as to analyse and identify available protocols, suitable for OB_{RAD} Data Transport protocol and OB_{RAD} Management and Control protocol.

The resulting study contains an analysis of the On-Board FRMCS functional architecture consisting of the FRMCS On-Board Gateway Function and FRMCS Radio Function architecture derived from UIC requirements specifications with regards to the OB_{RAD} functional interface.

Developing on the analysis, different solutions for physical OB_{RAD} architectures/installations/configurations are taken into consideration, and their technical characteristics are analysed and compared to explore the suitability of these solutions for the technical implementation of the OB_{RAD} physical interface as well as OB_{RAD} functional interface.

The present document notably provides an assessment of existing standardized protocols summarized in a table with pros and cons (Table 7.2), and a recommendation resulting from the consensus of a protocol for management and control points to NETCONF/RESTCONF/YANG as the most preferred protocol.

Introduction

As the needs of the railways are constantly evolving, in particular in the context of the digitalization of rail operation that is pursued in many countries and considering the upcoming obsolescence of GSM-R technology, UIC launched in 2012 the first studies for a successor to GSM-R, named Future Railway Mobile Communication System (FRMCS). The UIC published in 2023 a set of specifications for FRMCS version 1 and in 2024 for FRMCS version 2:

- UIC FRMCS TOBA FRS [i.1];
- UIC FRMCS FRS [i.2];
- UIC FRMCS SRS [i.3];
- UIC FRMCS FIS [i.4];
- UIC FRMCS FFFIS [i.5].

Within this set of specifications several interface reference points have been defined, including an interface reference point OB_{RAD} (On-Board Radio).

The present document is a study on the OB_{RAD} interface that identifies potential solutions and elaborates on possible technical realizations of the interface, as a follow-up of the need for further study mentioned in ETSI TR 103 459 [i.50], clause 6.3.2.

1 Scope

The present document is a study of the Onboard Radio Interface (OB$_{RAD}$). The following is covered:

- An analysis of the requirements on OB$_{RAD}$ captured in UIC FRMCS TOBA FRS [i.1], UIC FRMCS SRS [i.3] and other relevant UIC specifications.
- An analysis and identification of available protocols, suitable for OB$_{RAD}$ Data Transport protocol and OB$_{RAD}$ Management and Control protocol.
- A proposal on potential solution(s) and possible technical realization(s), covering the physical and functional OB$_{RAD}$ interface as well as physical implementations of the OB$_{RAD}$ interface.
- An analysis of the impact of the proposed OB$_{RAD}$ solution/realization to chipset, On-Board FRMCS architecture (Gateway Function, Radio Function, Operation and Maintenance) and migration aspects (existing versus new installations).
- An analysis of the capability of the proposed OB$_{RAD}$ solution/realization for performance aspects like responsiveness of the interface, latency, timing, and for availability (redundancy) aspects.

2 References

2.1 Normative references

Normative references are not applicable in the present document.

2.2 Informative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

[i.1] UIC FRMCS TOBA FRS TOBA-7510 (Version 2.0.0) (December 2024): "On-Board FRMCS - Functional Requirements Specification".

[i.2] UIC FRMCS FRS FU-7120 (Version 2.0.0) (December 2024): "Functional Requirement Specification".

[i.3] UIC FRMCS SRS AT-7800 (Version 2.0.0) (December 2024): "System Requirements Specification".

[i.4] UIC FRMCS FIS-7970 (Version 2.0.0) (December 2024): "Functional Interface Specification".

[i.5] UIC FRMCS FFFIS-7950 (Version 2.0.0) (December 2024): "Form Fit Functional Interface Specification".

[i.6] UNISIG SUBSET-147 (Version 0.1.10) (30.06.2022): "ERTMS Data Applications; FFFIS part: CCS Consist Network Communication Layers".

[i.7] USB/IP PROJECT (retrieved 20.10.2023).

[i.8] Takahiro Hirofuchi, Eiji Kawai, Kazutoshi Fujikawa, and Hideki Sunahara: "USB/IP - a Peripheral Bus Extension for Device Sharing over IP Network". In the Proceedings of the FREENIX Track: USENIX Annual Technical Conference, pp. 47-60, April 2005.

[i.9] Takahiro Hirofuchi, Eiji Kawai, Kazutoshi Fujikawa, and Hideki Sunahara: "USB/IP: A Transparent Device Sharing Technology over IP Network". IPSJ Transactions on Advanced Computing Systems, Vol. 46, No. SIG11(ACS11), pp. 349-361, August 2005.

[i.10] ITxPT Information Technology for Public Transport: "ITxPT TR3-003 MQTT v1.0.1".

[i.11] IETF RFC 2003 (October 1996): "IP Encapsulation within IP".

[i.12] IETF RFC 791 (September 1981): "Internet Protocol DARPA Internet Program Protocol Specification".

[i.13] IETF RFC 3410 (December 2002): "Introduction and Applicability Statements for Internet Standard Management Framework".

[i.14] IETF RFC 3411 (December 2002): "An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks".

[i.15] IETF RFC 3412 (December 2002): "Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)".

[i.16] IETF RFC 3413 (December 2002): "Simple Network Management Protocol (SNMP) Applications".

[i.17] IETF RFC 3414 (December 2002): "User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)".

[i.18] IETF RFC 3415 (December 2002): "View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)".

[i.19] IETF RFC 3416 (December 2002): "Version 2 of the Protocol Operations for the Simple Network Management Protocol (SNMP)".

[i.20] IETF RFC 3417 (December 2002): "Transport Mappings for the Simple Network Management Protocol (SNMP)".

[i.21] IETF RFC 3418 (December 2002): "Management Information Base (MIB) for the Simple Network Management Protocol (SNMP)".

[i.22] IETF RFC 2578 (April 1999): "Structure of Management Information Version 2 (SMIv2)".

[i.23] ETSI TS 138 415 (V17.0.0): "5G; NG-RAN; PDU session user plane protocol (3GPP TS 38.415 Release 17)".

[i.24] ETSI TS 129 281 (V17.4.0): "Universal Mobile Telecommunications System (UMTS); LTE; 5G; General Packet Radio System (GPRS) Tunnelling Protocol User Plane (GTPv1-U) (3GPP TS 29.281 Release 17)".

[i.25] ETSI TS 129 274 (V17.9.0): "Universal Mobile Telecommunications System (UMTS); LTE; 5G; 3GPP Evolved Packet System (EPS); Evolved General Packet Radio Service (GPRS) Tunnelling Protocol for Control plane (GTPv2-C); Stage 3 (3GPP TS 29.274 Release 17)".

[i.26] ETSI TS 129 244 (V17.9.0): "LTE; 5G; Interface between the Control Plane and the User Plane nodes (3GPP TS 29.244 Release 17)".

[i.27] IEEE Std 802.1™AB-2016 (Revision of IEEE Std 802.1™AB-2009): "IEEE Standard for Local and metropolitan area networks - Station and Media Access Control Connectivity Discovery".

[i.28] TIA-1057 (April 2006): "Telecommunications IP Telephony Infrastructure Link Layer Discovery Protocol for Media Endpoint Devices".

[i.29] Andrew Banks, Ed Briggs, Ken Borgendale, and Rahul Gupta: "MQTT Version 5.0". 07 March 2019. OASIS Standard.

[i.30] IETF RFC 2784 (March 2000): "Generic Routing Encapsulation (GRE)".

[i.31] IETF RFC 2890 (September 2000): "Key and Sequence Number Extensions to GRE".

[i.32] IETF RFC 8086 (March 2017): "GRE-in-UDP Encapsulation".

[i.33] IETF RFC 4741 (December 2006): "NETCONF Configuration Protocol" (obsoleted by IETF RFC 6241).

[i.34] IETF RFC 4742 (December 2006): "Using the NETCONF Configuration Protocol over Secure SHell (SSH)" (obsoleted by IETF RFC 6242).

[i.35] IETF RFC 5246 (August 2008): "The Transport Layer Security (TLS) Protocol Version 1.2" (obsoleted by IETF RFC 8446).

[i.36] IETF RFC 6020 (October 2010): "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)".

[i.37] IETF RFC 6241 (June 2011): "Network Configuration Protocol (NETCONF)".

[i.38] IETF RFC 6242 (June 2011): "Using the NETCONF Protocol over Secure Shell (SSH)".

[i.39] IETF RFC 7525 (May 2015): "Recommendations for Secure Use of Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)" (obsoleted by IETF RFC 9325).

[i.40] IETF RFC 7950 (August 2016): "The YANG 1.1 Data Modeling Language".

[i.41] IETF RFC 8040 (January 2017): "RESTCONF Protocol".

[i.42] IETF RFC 8446 (August 2018): "The Transport Layer Security (TLS) Protocol Version 1.3".

[i.43] IETF RFC 9325 (November 2022): "Recommendations for Secure Use of Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)".

[i.44] IETF RFC 7231 (June 2014): "Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content".

[i.45] C(2024)2466: "Commission Implementing Decision of 22.4.2024 on a standardisation request to the European Telecommunications Standards Institute as regards the definition of system specification requirements for the Future Railway Mobile Communication System in support of Directive (EU) 2016/797 of the European Parliament and of the Council".

[i.46] Directive (EU) 2016/797 of the European Parliament and of the Council of 11 May 2016 on the interoperability of the rail system within the European Union.

[i.47] Commission Implementing Regulation (EU) 2023/1695 of 10 August 2023 on the technical specification for interoperability relating to the control-command and signalling subsystems of the rail system in the European Union and repealing Regulation (EU) 2016/919.

[i.48] ISO/IEC 20922:2016: "Information technology - Message Queuing Telemetry Transport (MQTT) v3.1.1".

[i.49] "Eclipse Foundation, Eclipse Mosquitto, An open source MQTT broker" (retrieved 29.10.2024).

[i.50] ETSI TR 103 459 (V1.2.1): "Rail Telecommunications (RT); Future Rail Mobile Communication System (FRMCS); Study on system architecture".

[i.51] ETSI TS 127 007: "Digital cellular telecommunications system (Phase 2+) (GSM); Universal Mobile Telecommunications System (UMTS); LTE; 5G; AT command set for User Equipment (UE) (3GPP TS 27.007)".

3 Definition of terms, symbols and abbreviations

3.1 Terms

For the purposes of the present document, the terms given in UIC FRMCS TOBA FRS [i.1], UIC FRMCS FRS [i.2], UIC FRMCS SRS [i.3], UIC FRMCS FIS [i.4], UIC FRMCS FFFIS [i.5] and the following apply:

Application Plane: interaction plane providing the data exchange between endpoint applications

FRMCS Service Control Plane: interaction plane providing the signalling for session establishment and teardown via MCX/SIP

FRMCS Service User Plane: interaction plane providing for Loose-Couple Applications through MC clients the tunnelling for the data exchanged between endpoint applications

NOTE: This is equivalent to the Application Plane for Tight-Coupled Applications.

3.2 Symbols

Void.

3.3 Abbreviations

For the purposes of the present document, the following abbreviations apply:

3GPP	3rd Generation Partnership Project
5G	5th Generation of cellular telecommunications technologies standardized by 3GPP
5QI	5G QoS Identifier
ACL	Access Control List
ANSI	American National Standards Institute
APCO	Additional Protocol Configuration Options
API	Application Programming Interface
APN	Access Point Name
AT	Attention
CCS TSI	Control Command and Signalling Technical Specification for Interoperability
CCS	Control Command and Signalling
COTS	Commercial Off-The-Shelf
CPU	Central Processing Unit
CRUD	Create, Read, Update, Delete
DDP	Datagram Delivery Protocol
DDS	Data Distribution Service
E2E	End-to-End
ETSI	European Telecommunication Standards Institute
FFFIS	Form Fit Functional Interface Specification
FRMCS	Future Railway Mobile Communications System
FS_{MPM}	FRMCS System Multipath Management reference point/interface
FS_{OMR}	FRMCS System OM Remote reference point/interface
FS_{ONI}	FRMCS System Other Network reference point/interface
GNSS	Global Navigation Satellite Systems
G-PDU	GTP encapsulated user Plane Data Unit
GRE	Generic Routing Encapsulation
GSM-R	Global System for Mobile Communications - Railway
GTP	GPRS Tunnelling Protocol
GTP-C	GTP Control
GTP-U	GTP User
GW	Gateway
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
HW	Hardware

I/O	Input/Output
ID	Identity
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IMS	IP Multimedia Subsystem
IP	Internet Protocol
IPX	Internetwork Packet Exchange
JSON	JavaScript Object Notation
KPI	Key Performance Indicator
LAN	Local Area Network
LC	Loose-Coupled
LLDP	Link Layer Discovery Protocol
LLDP-MED	Link Layer Discovery Protocol - Media Endpoint Discovery
LTE	Long-Term Evolution
MAC	Medium Access Control, Media Access Control
MACsec	MAC Security
MBIM	Mobile Broadband Interface Model
MC	Mission Critical
MCX	Mission Critical Services
MIB	Management Information Base
MPF	Multipath Function
MP-QUIC	Multipath QUIC
MPTCP	Multipath TCP
MQTT	Message Queue Telemetry Transport
MSM	Mobile Station Modem
MTU	Maximum Transmission Unit
NETCONF	Network Configuration (Protocol)
O&M	Operation and Maintenance
OASIS	Organization for the Advancement of Structured Information Standards
OB	On-Board
$\mathrm{OB}_{\mathrm{ANT}}$	On-Board Antenna system reference point/interface
$\mathrm{OB}_{\mathrm{APP}}$	On-Board Application reference point/interface
OBGW	On-Board Gateway
$\mathrm{OB}_{\mathrm{OM}}$	On-Board Operation & Maintenance reference point/interface
$\mathrm{OB}_{\mathrm{RAD}}$	On-Board Radio Module reference point/interface
OM	Operation and Maintenance
OS	Operating System
OSI	Open Systems Interconnection
PDN	Packet Data Network
PDU	Packet Data Unit
PFCP	Packet Forwarding Control Protocol
PGW	PDN Gateway
QMI	Qualcomm MSM Interface
QoS	Quality of Service
RAT	Radio Access Technology
REST	Representational State Transfer
RESTCONF	Representational State Transfer Configuration (Protocol)
RF	Radio Function
RFC	Request for Comments
RFMF	Radio Function Management Function
RM	Radio Module
RPC	Remote Procedure Call
RTP	Real-Time Transport Protocol
SCTP	Stream Control Transmission Protocol
SDO	Standards Developing Organization
SDP	Session Description Protocol
SGW	Serving Gateway
SIP	Session Initiation Protocol
SNMP	Simple Network Management Protocol
SSE	Server-Sent Events
SSH	Secure Shell protocol
SW	Software

TC	Tight-Coupled
TCP	Transmission Control Protocol
TIA	Telecommunications Industry Association
TLS	Transport Layer Security
TOBA	Telecom On-Board Architecture
T-PDU	Transport PDU
TR	Technical Report
TS	Technical Specification
UART	Universal Asynchronous Receiver and Transmitter
UDP	User Datagram Protocol
UE	User Equipment
UIC	Union Internationale des Chemins de Fer
UPF	User Plane Function
USB	Universal Serial Bus
VETH	Virtual Ethernet
VHCI	Virtual Host Controller Interface
VLAN	Virtual Local Area Network
VRF	Virtual Routing and Forwarding
Wi-Fi®	Wireless Fidelity
XML	Extensible Markup Language
YANG	Yet Another Next Generation

4 OB$_{RAD}$ within On-Board FRMCS architecture

4.1 On-Board FRMCS v2 functional architecture

The On-Board FRMCS v2 functional architecture is depicted in below Figure 4.1 based on UIC FRMCS SRS [i.3], clause 7.1.3.1 and indicates the location of the OB$_{RAD}$ interface. It is a functional view and does not assume any physical deployment.

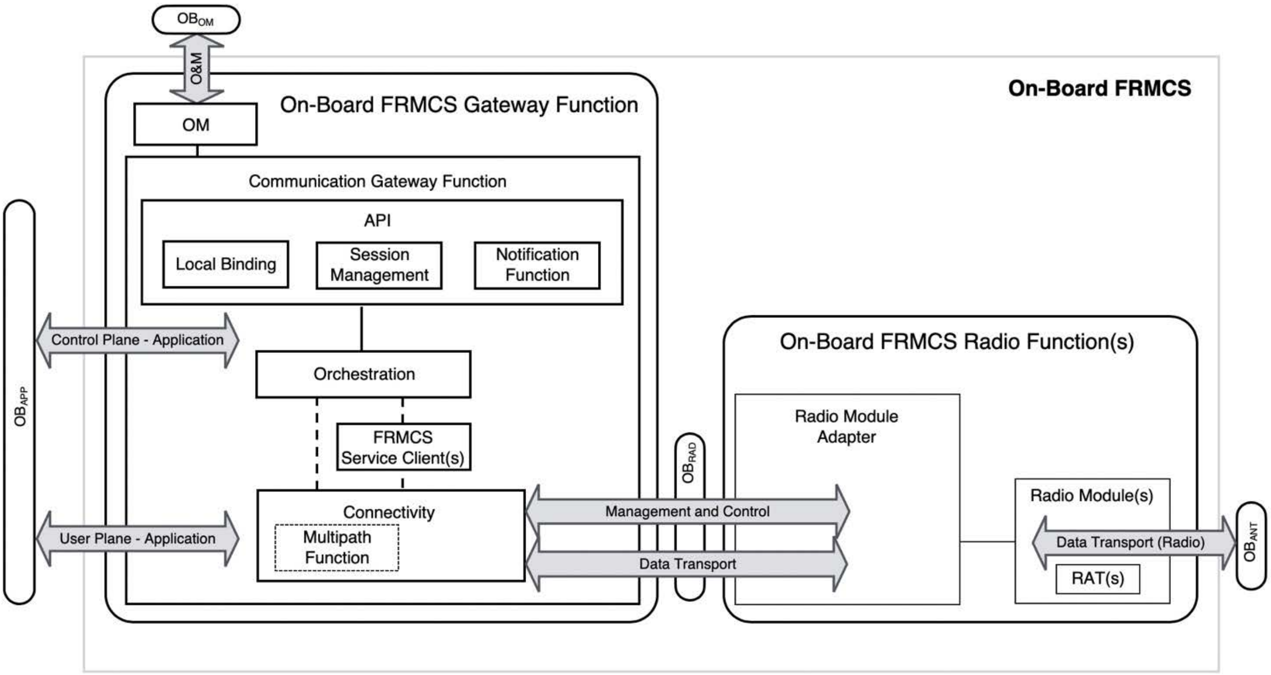


Figure 4.1: On-Board FRMCS v2 architecture

The On-Board FRMCS functional architecture as shown in Figure 4.1 is intended to support (at least) (UIC FRMCS SRS [i.3], clause 7.1.3.1.2 and clause 7.1.4.2.1.1.3):

- Integrated architecture;
- Integrated architecture providing interchangeability;
- Distributed architecture providing interchangeability.

These architecture options are analysed and described in clause 4.3 with regard to their impact on OB_{RAD}.

One or more Radio Function(s) is/are connected via the OB_{RAD} interface with one Gateway Function ("System mode 1:n", UIC FRMCS SRS [i.3], clause 17.3.1.2).

NOTE 1: "System mode m:n" is *"out of scope for FRMCS V2"* (UIC FRMCS SRS [i.3], clause 17.3.1.3).

The Radio Function enables access for the Communication Gateway to the FRMCS Transport Stratum and enables the transmission of control and user plane related data (derived from UIC FRMCS SRS [i.3], clause 7.1.6.1.2i and clause 7.1.6.1.2ii).

The boundaries of the Radio Function are identified by the reference points OB_{RAD} and OB_{ANT} (derived from UIC FRMCS SRS [i.3], clause 7.1.6.1.2iii).

The Data-Transport (Protocol) of the OB_{RAD} interface enables Control Plane (Session) and User Plane (Media) communication between Gateway Function and Radio Function(s) (derived from UIC FRMCS SRS [i.3], clause 7.1.4.2.1.2.1 and clause 7.1.4.2.1.2.2).

The Management and Control (Protocol) of the OB_{RAD} interface enables the Gateway Function to establish communication session(s) within a Radio Function, to relocate established communication session(s) between Radio Functions or Radio Modules, to select and use the Radio Function(s) and to select and use the Radio Module(s) hosted by Radio Function(s) (derived from UIC FRMCS SRS [i.3], clause 7.1.4.2.1.2.3, clause 7.1.4.2.1.2.4 and clause 7.1.4.2.1.2.5).

The Management and Control (Protocol) of the OB_{RAD} interface enables the OM Function (of the Gateway Function) to retrieve status information, log and performance data from the Radio Functions(s) and to transfer the data necessary for software/firmware updates and configuration changes/updates to the Radio Function(s) (derived from UIC FRMCS TOBA FRS [i.1], clause 7.11.2.7 and clause 7.7; UIC FRMCS SRS [i.3], clause 7.1.5.11).

NOTE 2: The configuration changes/updates include changes/updates to the UE capability settings of the Radio Module(s).

4.2 FRMCS Radio Function architecture

Figure 4.2 shows as an example the On-Board architecture having two Radio Functions connected via the OB_{RAD} interface with the Gateway Function with a total of three Radio Modules. The two Radio Functions could be of different vendors. One Radio Function could be located close to the Gateway Function while the other is placed remotely elsewhere on-board (e.g. close to the antennas), i.e. used in a distributed architecture. The three Radio Modules could, for example, implement the same or different RATs (e.g. 3GPP LTE/4G, 3GPP FRMCS/5G, non-3GPP, Wi-Fi®, etc.), while being from the same or from different equipment vendors. One Radio Module could be dedicated to railways FRMCS/5G and another to public 5G, or (if they implement the same RAT) one Radio Module could be active while another is set to act as stand-by/spare unit (redundancy). Many other configurations of the system are possible.

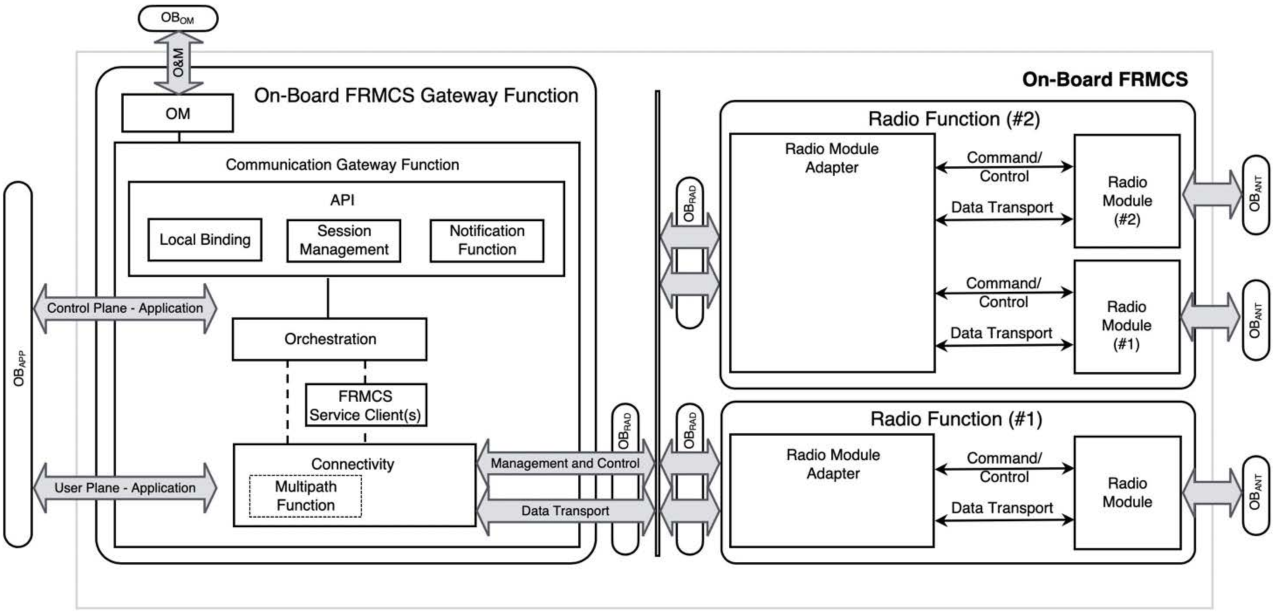


Figure 4.2: On-Board FRMCS example with two Radio Functions and three Radio Modules

The desired co-existence of Radio Functions from different vendors and the option of installing the Radio Function(s) and the Gateway Function at different locations implies a standardized OB_{RAD} interface (UIC FRMCS TOBA FRS [i.1], clause 7.11.1).

Each Radio Function includes one Radio Module Adapter functional block (that may be further divided into several adapter instances) and one or more Radio Module(s) (UIC FRMCS SRS [i.3], clause 7.1.3.2.4).

A Radio Module may be a COTS (commercial off-the-shelf) wireless modem, supporting one or more RATs (3GPP and/or non-3GPP). It offers one or more interfaces for Command/Control and Data Transport. These interfaces (physical and functional) may differ from vendor to vendor and model to model. The Radio Module is connected via OB_{ANT} to the antenna(s), as shown in Figure 4.2 (UIC FRMCS SRS [i.3], clause 7.1.6.3.4).

The Radio Module Adapter functional block is in charge of mapping the manufacturer specific Command/Control and Data Transport interface(s) of each installed Radio Module to the future standardized OB_{RAD} interface (UIC FRMCS SRS [i.3], clause 7.1.6.2.4). The Radio Module Adapter is designated to support Radio Function and optional Radio Module interchangeability (UIC FRMCS SRS [i.3], clause 7.1.6.2.5). By following this concept, the standardized OB_{RAD} does not add specific constraints or requirements to the chosen Radio Module(s), thus does not per se exclude the use of neither any specific Radio Module(s) nor any (Radio Module) chipset(s).

Radio Function(s) might be added to or removed from the OB_{RAD} interface while the On-Board FRMCS is operational (in-service replacement, UIC FRMCS SRS [i.3], clause 7.1.4.2.1.2.6), requiring no other task on the Gateway Function than a software configuration.

Radio Function Interchangeability, i.e. the *"on-board addition or replacement of On-Board FRMCS Radio Functions without impact on the On-Board FRMCS interfaces"*, is achieved by the introduction of the "On-Board FRMCS Radio Function configuration(s)" (see clause 4.3).
Radio Module Interchangeability, i.e. the *"on-board addition or replacement of Radio Modules without impact on the On-Board FRMCS interfaces"*, is achieved by the introduction of the "On-Board FRMCS Radio Function configuration(s)" (see clause 4.3).

NOTE: For the definition of "FRMCS Radio Function Interchangeability" and "FRMCS Radio Module Interchangeability", see definition section of UIC FRMCS TOBA FRS [i.1] and UIC FRMCS SRS [i.3].

4.3 Physical OB_{RAD} architectures/installations/configurations

Two possible configurations of a Radio Function are defined (UIC FRMCS TOBA FRS [i.1], clause 7.11.1.1 and definition of "On-Board FRMCS Radio Function configuration" in definition section of UIC FRMCS SRS [i.3]):

- Radio Function **Detachable** configuration: the Radio Function can be disconnected and re-attached on-board without intervention at the manufacturer's factory; or
- Radio Function **Attached** configuration: a permanent HW connection exists between the Gateway Function and the Radio Function, i.e. the Radio Function cannot be replaced without factory intervention.

For each of the above configurations, the Radio Module(s) as part of each Radio Function can be **Attached** or **Detachable** (UIC FRMCS TOBA FRS [i.1], clause 7.11.1.2).

In Radio Function **Detachable** configuration, the Gateway Function and the Radio Function(s) need a physical port to connect to each other through OB_{RAD} (UIC FRMCS TOBA FRS [i.1], clause 7.11.1.3).

The Radio Function **Detachable** configuration, connected to a "standardized OB_{RAD}" interface, allows different physical architectures (installations) within the On-Board FRMCS as shown in Figure 4.3:

a) Local/centralized installation of FRMCS Gateway Function and FRMCS Radio Function(s) (i.e. the "Integrated architecture providing interchangeability", UIC FRMCS SRS [i.3], clause 7.1.3.1.2); or

b) Remote/distributed installation of FRMCS Gateway Function and FRMCS Radio Function(s) (i.e. the "Distributed architecture providing interchangeability", UIC FRMCS SRS [i.3], clause 7.1.3.1.2); or

c) A mixture of both local/centralized and remote/distributed installation.

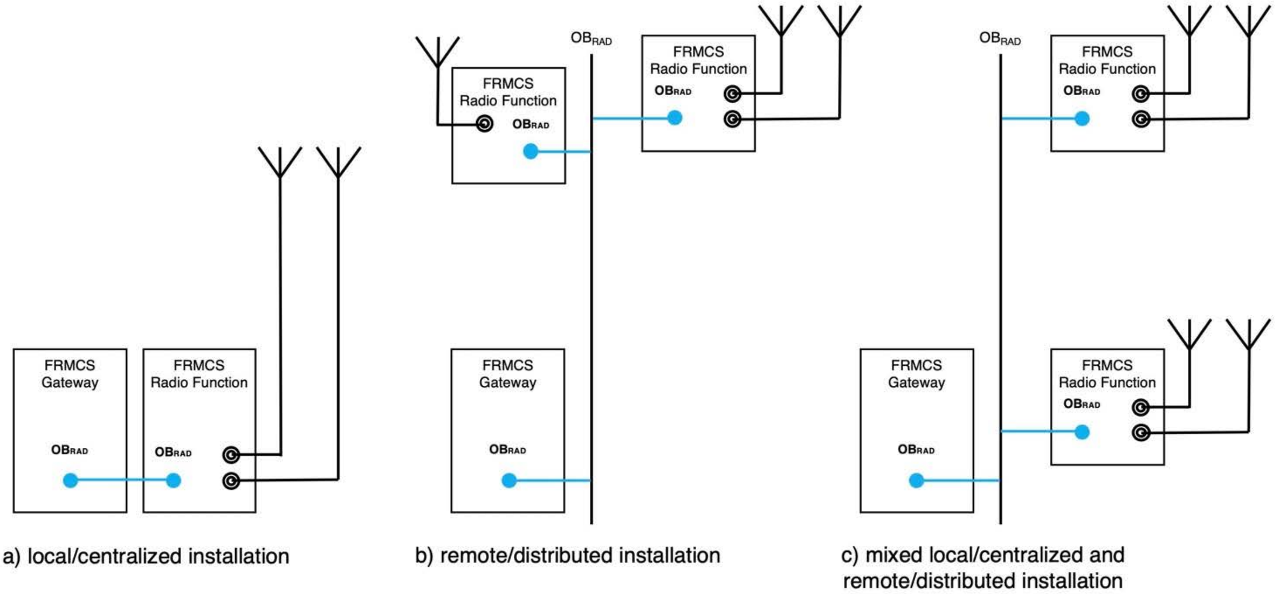


Figure 4.3: Physical OB_{RAD} architectures/installations with Detachable Radio Function(s)

For an "Integrated On-Board FRMCS" (i.e. the Gateway Function and one or more Radio Function(s) are integrated in one device, i.e. the "Integrated architecture", UIC FRMCS SRS [i.3], clause 7.1.3.1.2), the Radio Function(s) may be used either in Radio Function **Detachable** configuration (similar as architecture/installation "a" in Figure 4.3, but mounted in one mechanical housing) or in Radio Function **Attached** configuration as shown below in Figure 4.4.

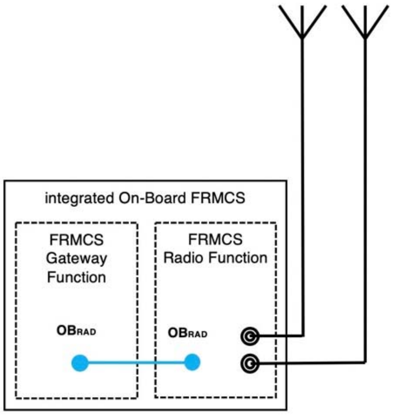


Figure 4.4: Integrated On-Board FRMCS with Attached Radio Function

In any of the above configurations, the OB_{RAD} interface needs to support the same protocols, (list of) parameters, triggered actions and procedures.

4.4 Requirements from UIC FRMCS TOBA FRS related to OB_{RAD}

Table 4.1 lists those functional requirements from UIC FRMCS TOBA FRS [i.1] which are related to OB_{RAD}.

Table 4.1: Functional requirements related to OB_{RAD} from UIC FRMCS TOBA FRS

Reference to UIC FRMCS TOBA FRS	Comments
Clause 7.11.2.1	
Clause 7.11.2.1i	
Clause 7.11.2.2	
Clause 7.11.2.2i	
Clause 7.11.2.3	
Clause 7.11.2.4	
Clause 7.11.2.5	
Clause 7.11.2.7	Referenced clause 7.7 is about Operations and Maintenance requirements.

4.5 Requirements from UIC FRMCS SRS related to OB_{RAD}

Table 4.2 lists those system requirements from UIC FRMCS SRS [i.3] which are related to OB_{RAD}.

Table 4.2: System requirements related to OB_{RAD} from UIC FRMCS SRS

Reference to UIC FRMCS SRS	Comments
Clause 7.1.4.2.1.1.1	
Clause 7.1.4.2.1.1.2	
Clause 7.1.4.2.1.1.3	Referenced clause 7.1.3.1.2 is about possible architectures: • Integrated architecture; • Integrated architecture providing interchangeability; • Distributed architecture providing interchangeability.
Clause 7.1.4.2.1.1.4	
Clause 7.1.4.2.1.2.1	
Clause 7.1.4.2.1.2.2	
Clause 7.1.4.2.1.2.3	
Clause 7.1.4.2.1.2.4	

Reference to UIC FRMCS SRS	Comments
Clause 7.1.4.2.1.2.5	
Clause 7.1.4.2.1.2.6	
Clause 7.1.4.2.1.2.7	
Clause 7.1.4.2.1.3.1.1	
Clause 7.1.4.2.1.3.1.2	
Clause 7.1.4.2.1.3.1.3	
Clause 7.1.4.2.1.3.1.4	
Clause 7.1.5.10.2.4	
Clause 7.1.5.10.3.2	
Clause 7.1.5.11.2.1.5	
Clause 7.1.5.11.2.6.2	
Clause 7.1.5.11.2.6.4	

5 OB$_{RAD}$ physical and functional interface

5.1 OB$_{RAD}$ physical interface definition

The OB$_{RAD}$ physical interface may differ depending on whether the installed FRMCS Radio Function(s) is/are in a FRMCS Radio Function **Attached** or **Detachable** configuration (see clause 4.3).

In case of an integrated FRMCS Gateway/Radio Function architecture with FRMCS Radio Function **Attached** configuration, the physical internal OB$_{RAD}$ interface will be an implementation-specific interface.

In case of FRMCS Radio Function **Detachable** configurations, the physical OB$_{RAD}$ interface is recommended to be Ethernet (IEEE 802.3), and recommended to use an already existing (present) On-Board network infrastructure based on Ethernet, e.g. "Ethernet CCS Consist Network" as defined in SUBSET-147 [i.6] (UIC FRMCS SRS [i.3], clause 7.1.4.2.1.3.1.4). The physical OB$_{RAD}$ interface can either be shared or separated from OB$_{APP}$ physical interface.

The choice of Ethernet as the physical OB$_{RAD}$ interface may enable sufficient performance (bandwidth, latency) to carry both the OB$_{RAD}$ Management and Control (plane) and the OB$_{RAD}$ Data Transport (plane) traffic (see clause 5.2).

NOTE: For performance requirements related to OB$_{RAD}$, see UIC FRMCS SRS [i.3], clause 7.1.4.2.1.3.1.2; for QoS requirement values see UIC FRMCS SRS [i.3], Annex A and additional requirements are identified in UIC FRMCS SRS [i.3], clause 14.

Performance analysis in the present document is only related to the identification of potential requirements, since some functionalities are still unknown e.g. the performance of chipsets.

5.2 OB$_{RAD}$ functional interface

The OB$_{RAD}$ functional interface can be divided into the two planes for:

- Management and Control; and
- Data Transport.

The OB$_{RAD}$ Management and Control (plane) provides functionalities for:

- Control and management of Radio Function(s)
(set and retrieve configuration parameters, retrieve status and communication session information, retrieve operation and maintenance information, retrieve performance and diagnostic information);
- Establishment, relocation and release of communication sessions;
- Control and provision of GNSS positioning information from Radio Module(s), if a GNSS receiver is integrated on a Radio Module (UIC FRMCS TOBA FRS [i.1], clauses 7.8.2 and 7.8.7 and *"3GPP UE incl. GNSS"* within UIC FRMCS SRS [i.3], clause 16.4.1/Figure 16-3);

NOTE: This is a proposal, because in current versions of UIC FRMCS SRS [i.3] and UIC FRMCS TOBA FRS [i.1] there is no explicit written requirement about GNSS positioning information for OB_{RAD}.

- Transfer and control of SW updates and configuration changes (from Gateway Function (OM) to Radio Function(s) and their Radio Module(s));
- Retrieval of log data (by Gateway Function (OM) from Radio Function(s) and their Radio Module(s)).

The OB_{RAD} Data Transport (plane) provides functionalities for:

- Application Plane, the FRMCS Service User Plane and the FRMCS Service Control Plane data transfer between Gateway Function (Connectivity) and Radio Function(s) for one or more communication session(s).

The status and communication session information retrieved from Radio Function(s) and their Radio Module(s) enables the Gateway Function (Connectivity) to perform the Data Path routing, i.e. to select which communication session on which Radio Module on which Radio Function is to be used for Data Transport.

The On-Board FRMCS Multipath is part of the Gateway Function (Connectivity) as shown in Figure 4.1 and called FRMCS Multipath Function (MPF) according to UIC FRMCS SRS [i.3], clause 12.3.6.

According to UIC FRMCS SRS [i.3], clause 7.1.5.10.3.2, the *"On-Board FRMCS Multipath is a function that manages and controls concurrent user plane data flow distribution over* OB_{RAD}*"*.

UIC FRMCS TOBA FRS [i.1] clause 7.2.2 and UIC FRMCS SRS [i.3] clause 12.3 list the requirements applicable to FRMCS Multipath. Multipath use cases are listed in UIC FRMCS SRS [i.3], clause 12.3.5.

According UIC FRMCS TOBA FRS [i.1], clause 7.2.2.1, the On-Board FRMCS *"shall enable communication concurrently over multiple transport domains"*.

The On-Board FRMCS MPF *"should, whenever active, be able to contribute with information supporting the evaluation of data paths per data flow (e.g. availability, QoS)"* (UIC FRMCS TOBA FRS [i.1], clause 7.2.2.3) and based on the results *"shall be able to switch any data flow from one data path to another"* (UIC FRMCS TOBA FRS [i.1], clause 7.2.2.5).

FRMCS Multipath is a functionality within the Gateway Function, and as far as OB_{RAD} is concerned, it provides the capability to select given Radio Module(s) and route specific data flows to/from the associated Radio Module(s) independently.

6 Analysis of existing standardized protocols

6.1 Introduction

The following clauses contain the analysis of existing standardized protocols, which might be suitable to be used within the OB_{RAD} as functional/logical interface for:

- Data Transport
- Management and Control

The following protocols have been analysed:

- USB over IP (Management and Control, Data Transport)
- SNMP (Management and Control)
- IP-in-IP encapsulation (Data Transport)
- GTP-U (Data Transport)
- HTTP API / MQTT (Management and Control)
- MQTT (Management and Control)

- NETCONF/RESTCONF/YANG (Management and Control)

The analysis of the proposed protocols are given in the subsequent clauses. The order of appearance does not follow any order, ranking or assessment.

NOTE: Pros and cons for all proposals are listed in a comparison table (see Table 7.2).

6.2 Proposal A: USB over IP

This proposal is about the use of USB over IP protocol (USB-IP, USB/IP) ([i.7], [i.8] and [i.9]) for Management and Control protocol as well as for Data Transport protocol. It has been implemented in FRMCS prototype for interfacing an On-Board Gateway Function with a remote FRMCS Radio Module (embedded in a Radio Function).

Table 6.1 shows a simplified comparison of the protocol stacks used when interfacing with an internal Radio Module and when interfacing with an external remote Radio Module.

Table 6.1: Comparison of stacks when interfacing an internal or a remote Radio Module (simplified)

OSI Layers	Radio Module is inside the OB GW (see Figures 4.3a/4.4)	Radio Module is outside the OB GW (inside the remote Radio Function) (see Figures 4.3b/4.3c)
Application	GW software OBRAD API	
Presentation	USB Layers	USB
Session		
Transport		UDP, TCP
Network		IP
Link		IEEE 802.3
Physical	USB3.x / PCIe* M.2 interface	Ethernet M12 connector

Based on Table 6.1, it appears that the application layer is identical when interfacing an internal Radio Module and a remote Radio Module embedded in a Radio Function. Thus, a common protocol to drive the Radio Module in both configurations could be one or more of the widely used AT commands, QMI, MBIM or debug interface over USB links. A local USB link is sufficient for an internal Radio Module, but an Adapter is needed for a remote Radio Module which would be reached through the IP network of the train.

USB-IP allows a remote access via IP to the USB interface of the remote Radio Module from the FRMCS On-Board Gateway. The remote Radio Module will be seen by the Gateway software as if the Radio Module is inside the On-Board Gateway via a virtual internal USB interface.

USB-IP is a protocol to encapsulate USB connections over an TCP/IP link. USB/IP PROJECT [i.7] gives an overview of the USB-IP Design.

In [i.7] there are also links to further documents/articles ([i.8] and [i.9]) as well as an USB-IP Linux® implementation.

NOTE 1: Linux® is the registered trademark of Linus Torvalds in the U.S. and other countries.

Performance tests within an LTE infrastructure have been performed, demonstrating good results in terms of bandwidth and latency. In these tests, the USB/IP client was the FRMCS OB GW, and the USB/IP server was a Linux CPU and a 5G Radio Module connected to the CPU through local USB.

Conclusion

Pros:

- In performance tests an external 4G Radio Module was well detected and remotely managed over a TCP/IP connection.
- "Opens the door" to physical distributed architecture inside the train.
- Several remote Radio Modules can be handled by the On-Board Gateway (not tested).

- The USB-IP implementation under Linux is available and documented [i.7].
- Good global performance: low added latency, low CPU usage (well split between CPU cores), limited overhead.
- All USB connections of remote Radio Module are available locally: AT link, Debug link, etc.
- USB3.x is well supported.
- Many Radio Functions may be remotely connected via Ethernet to one Gateway, thereby providing, for example, hardware redundancy and bearer flexibility.

Cons:

- Software part to encapsulate USB inside IP (not an RFC standard) is available as Linux implementation, but may not be available for other operating systems.
- The Gateway needs to manage the remote Radio Modules at low level (drivers for remote Radio Modules have to be included).
- USB-IP is exclusive, the remote Radio Module can only be coupled to one On-Board Gateway at a time.
- USB-IP does not cover all requirements to drive a Radio Module. Needs to be completed by specific remote procedure calls (via HTTP RESTful API, SNMP, etc.), to drive physical electronic signals: Radio Module switch on/off, LEDs, Radio Module reset, thermal aspect, etc.
- Potential 100/1 000 base-TX throughput limitation versus USB 3 maximum transmission speed (5 Gbit/s).

The analysis leads to the following conclusive statements and questions:

- USB-IP makes both internal and remote Radio Modules visible at the same driver level (high coupling between Gateway software and standalone Radio Module software). Performance is mainly impacted by the addition of an IP header for traffic towards and from the remote Radio Module.
- USB-IP does not cover all the requirements on its own, an additional API needs to be defined to drive remotely some physical functions of the remote Radio Module.
- This solution should be compared and challenged against using a remote Radio Module as an IP wireless router.
- Should USB-IP be an optional feature of OB_{RAD} definition?

NOTE 2: A simpler solution to implement would be to use directly a USB-C interface for OB_{RAD}. It would avoid requiring an Adapter attached to the Radio Module to perform the USB/IP "server" function.

6.3 Proposal B: SNMP and IP-in-IP encapsulation

6.3.1 Introduction

Since the Management and Control protocol and the Data Transport protocol may have interdependent prerequisites, proposals for both protocols are first presented to allow for independent assessment before concluding.

6.3.2 IP-in-IP encapsulation (Data Transport protocol)

This proposal is about the use of IP-in-IP encapsulation for Data Transport protocol.

As a pre-requisite, before an Application or the Communication Gateway is able to send User Plane or Control Plane data via OB_{APP} -> Gateway Connectivity -> OB_{RAD} -> Radio Function/Radio Module, at least one **Communication Session** needs to be established by the Radio Module (e.g. a PDU Session in 3GPP networks). There can be one or more Communication Sessions be activated at the same time, in 3GPP each PDU Session having one or more QoS flows as shown in Figure 6.2.

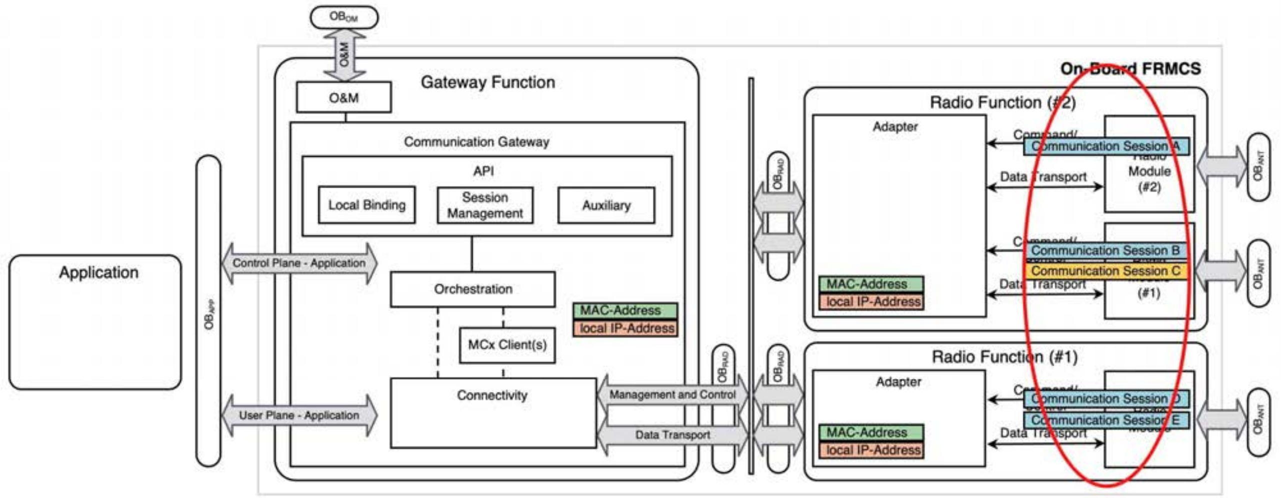


Figure 6.2: PDU Sessions on Radio Module level

A Communication Session might be "pre-configured" and established automatically as part of the initialization/start-up (i.e. in 3GPP after PS attach) of the On-Board FRMCS, or established on demand as required by the Gateway Connectivity or O&M, or re-established either by the Radio Function/Radio Module, by the O&M or by the Gateway Connectivity.

The parameters of all Communication Sessions of all Radio Modules within a Radio Function need to be available/known in the Adapter, and the parameters of all Communication Sessions of all Radio Functions relevant for selecting an appropriate Data Path (Communication Session and Radio Function) needs to be available/known in the Gateway Connectivity as shown in Figure 6.3. The Gateway Connectivity gets the parameters via the OB_{RAD} Management and Control Protocol. The parameter set of one Communication Session contains at least (internal) Communication Session ID, status and assigned IP-Address, and may contain (e.g. in 3GPP) APN, 5QI and other parameters. The Gateway Connectivity needs to know the local IP-Address of every Radio Function; every Radio Function needs to know the local IP-Address of the Gateway Function.

For selecting the appropriate Data Path not only the Communication Session parameters might be needed, but also parameters like e.g. RAT (3GPP 4G, 3GPP 5G, non-3GPP Wi-Fi®, etc.), network (public, FRMCS Rail, etc.), coverage status etc. More than one Data Path might be selected e.g. for multipath use cases or in case of redundancy configurations (e.g. one "active" and one "stand-by" Data Path), according to the FRMCS Multipath use cases in UIC FRMCS SRS [i.3], clause 12.3.5.

The exchange of the parameters between the Gateway Connectivity and Radio Function(s) is bi-directional, i.e. the Gateway Connectivity may request to set/configure/change parameter(s) in the Radio Function(s) (e.g. in 3GPP to specify the PDU Session parameters for a PDU Session establishment or modification, etc.), while the Radio Function(s) need to inform the Gateway Connectivity about any parameter or status change.

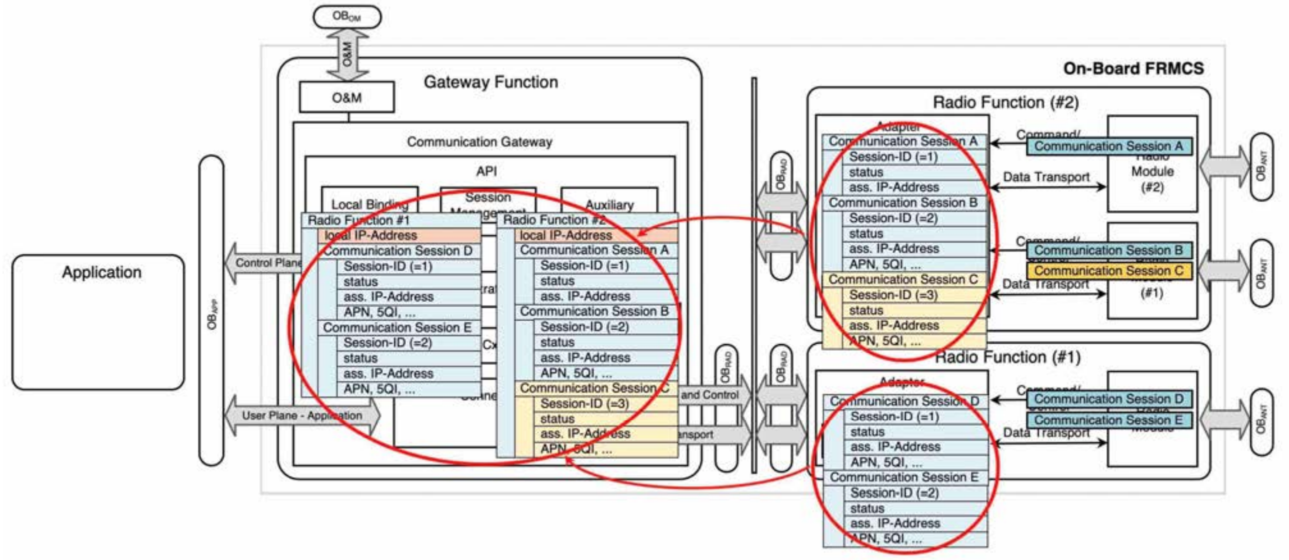


Figure 6.3: Communication Session parameters on Radio Function and Gateway Connectivity level

The knowledge/availability of all Communication Sessions and their relevant parameters in the Gateway Connectivity is a pre-requisite for routing a Data Transport IP-packet for a specific/selected Communication Session via OB$_{RAD}$ to the specific/selected Radio Function (and then subsequently route it to the specific/selected Radio Module maintaining that Communication Session) as shown in Figure 6.4.

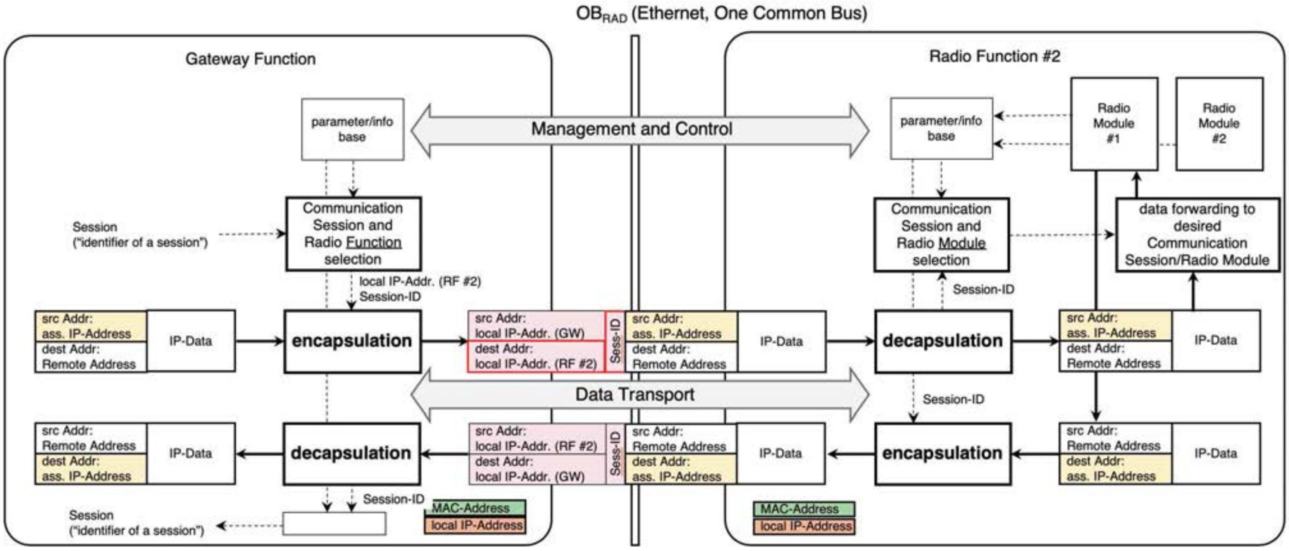


Figure 6.4: IP-in-IP encapsulation and routing functionality

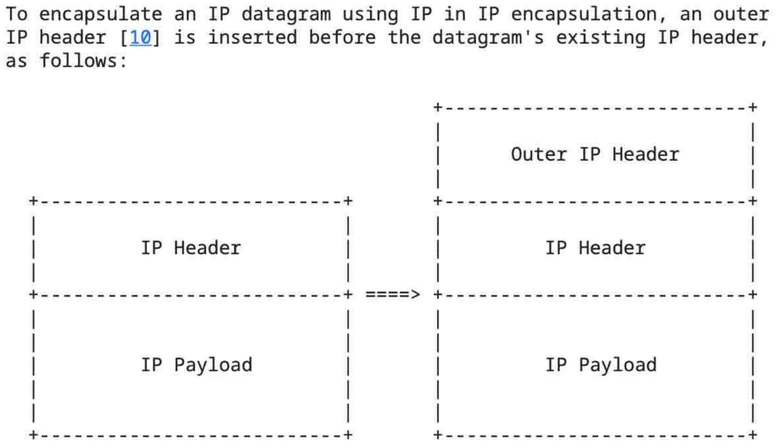
As the focus of the current proposal is on Data Transport protocol, it is assumed that the Management and Control protocol ensures the Communication Session parameters contained in the "parameter/info base" are always "up-to-date" between the Gateway Function and the involved Radio Function (as it is described in detail in Management and Control protocol, clause 6.3.3).

For each and every Data Transport IP-packet to be sent to the mobile network, the Gateway Function needs to perform a "Communication Session and Radio Function selection" by which it determines the local IP-Address of the Radio Function (in this example Radio Function #2) and the "(internal) Communication Session ID", based on the Session ("identifier of a session") to which this IP-packet belongs and the Communication Session parameters in the "parameter/info base".

Prior to the transmission of the "original" Data Transport IP-packet via the OB_{RAD} Ethernet network, the Gateway Function encapsulates that IP-packet by adding another IP-header ("outer IP-header") in front of it. This "IP-in-IP encapsulation" is further defined in IETF RFC 2003 [i.11]. In the "outer IP-header" the source (src) Address is set to the local IP-Address of the Gateway Function, the destination (dest) Address is set to the local IP-Address of the Radio Function (#2) and, the "(internal) Session and QoS Flow ID" needs to be set in the "Options" field (the "Options" field is optional and may consist of 0, 1 or more "TLV"-coded information elements; see IETF RFC 791 [i.12], clause 3.1). Other parameters in the "outer IP-header" need to be set accordingly, e.g. the "Protocol" field is set to indicate "4: IP in IP (encapsulation)" (IETF RFC 2003 [i.11], clause 3.1).

Upon reception of an IP-packet via OB_{RAD}, the Radio Function decapsulates the "original" Data Transport IP-packet (if, and only if, the "Protocol" field in the "outer IP-header" indicates "4: IP in IP (encapsulation)"). Based on the included "(internal) Communication Session ID" and on the Communication Session parameters in its "parameter/info base", the Radio Functions needs to determine the Communication Session (in 3GPP: PDU Session/QoS Flow) and the related Radio Module, to which it then routes that "original" Data Transport IP-packet.

The reception of Data Transport IP-packet from the network and its encapsulation and routing work in a similar manner.



```
To encapsulate an IP datagram using IP in IP encapsulation, an outer
IP header [10] is inserted before the datagram's existing IP header,
as follows:

                                     +---------------------------+
                                     |                           |
                                     |      Outer IP Header      |
                                     |                           |
 +---------------------------+       +---------------------------+
 |                           |       |                           |
 |         IP Header         |       |         IP Header         |
 |                           |       |                           |
 +---------------------------+ ====> +---------------------------+
 |                           |       |                           |
 |                           |       |                           |
 |         IP Payload        |       |         IP Payload        |
 |                           |       |                           |
 |                           |       |                           |
 +---------------------------+       +---------------------------+
```

Figure 6.5: IP-in-IP encapsulation with "Outer IP Header" (Source IETF RFC 2003 [i.11])

Summary

- Proposal to use IP-in-IP encapsulation for Data Transport protocol according to IETF RFC 2003 [i.11].
- IP-in-IP encapsulation does not modify the original IP-Packet.
- Routing/forwarding information is based on parameter set (parameter/information base) exchanged via OB_{RAD} Management and Control protocol (i.e. (internal) Communication Session ID, local IP-Address of Radio Function).
- The (internal) Communication Session ID is indicated in "outer IP-header" (as part of the optional "Options" field).
- In 3GPP, the (internal) Communication Session ID identifies not only the PDU Session, but also the QoS-Flow.
- The "(internal) Communication Session ID" is not the same as the "identifier of a session" on OB_{APP}.
- The Gateway is the controlling entity.

6.3.3 SNMP (Management and Control protocol)

This proposal is about the use of Simple Network Management Protocol (SNMP) for Management and Control protocol.

As shown in the previous clause, the bi-directional exchange of the PDU Session/QoS Flow parameters contained in the "parameter/info base" between the Gateway Function and the Radio Function(s) (keeping them "up-to-date") is the essential pre-requisite for the Data Transport protocol, which needs to be ensured by the Management and Control protocol as highlighted in Figure 6.6.

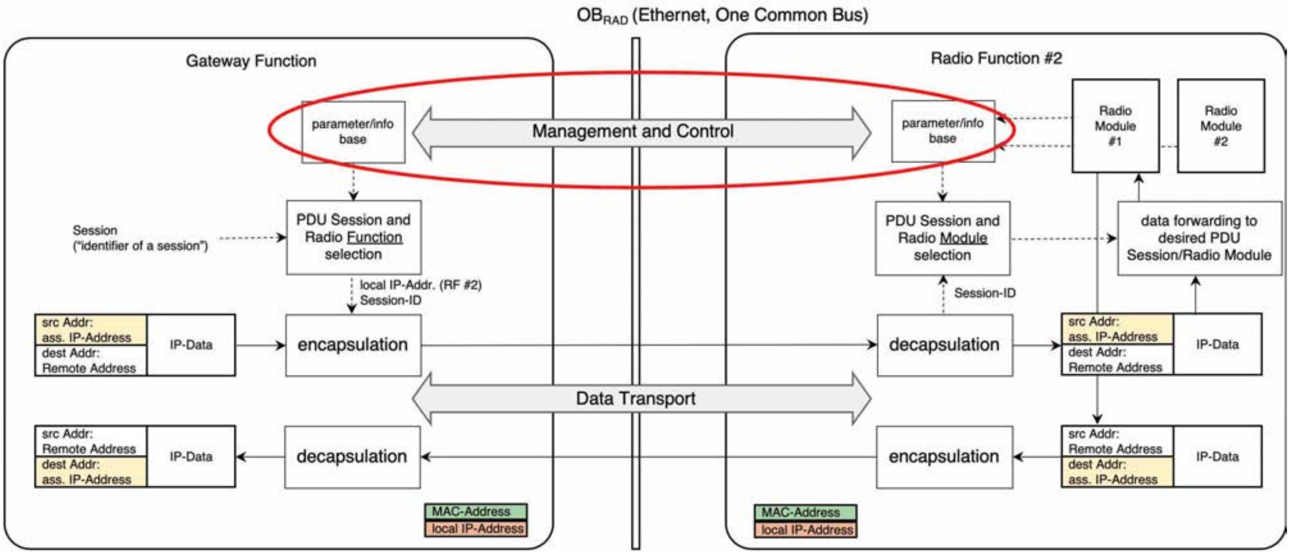

OBRAD (Ethernet, One Common Bus)
Gateway Function
Radio Function #2
parameter/info base
Management and Control
parameter/info base
Radio Module #1
Radio Module #2
Session ("identifier of a session")
PDU Session and Radio Function selection
PDU Session and Radio Module selection
data forwarding to desired PDU Session/Radio Module
local IP-Addr. (RF #2) Session-ID
Session-ID
src Addr: ass. IP-Address
dest Addr: Remote Address
IP-Data
encapsulation
decapsulation
Data Transport
src Addr: Remote Address
dest Addr: ass. IP-Address
decapsulation
encapsulation
MAC-Address
local IP-Address

Figure 6.6: Management and Control to Data Transport protocol relationship

From a generic perspective, the Management and Control protocol needs to provide messages to enable the Gateway Function to:

- Command Set/Configure parameter(s) in a Radio Function.
- Command Get/Enquire parameter(s) from a Radio Function.
- Receive Change Notifications (of parameter(s)) from a Radio Function.

as well as to provide a set of defined parameters, as shown in Figure 6.7.

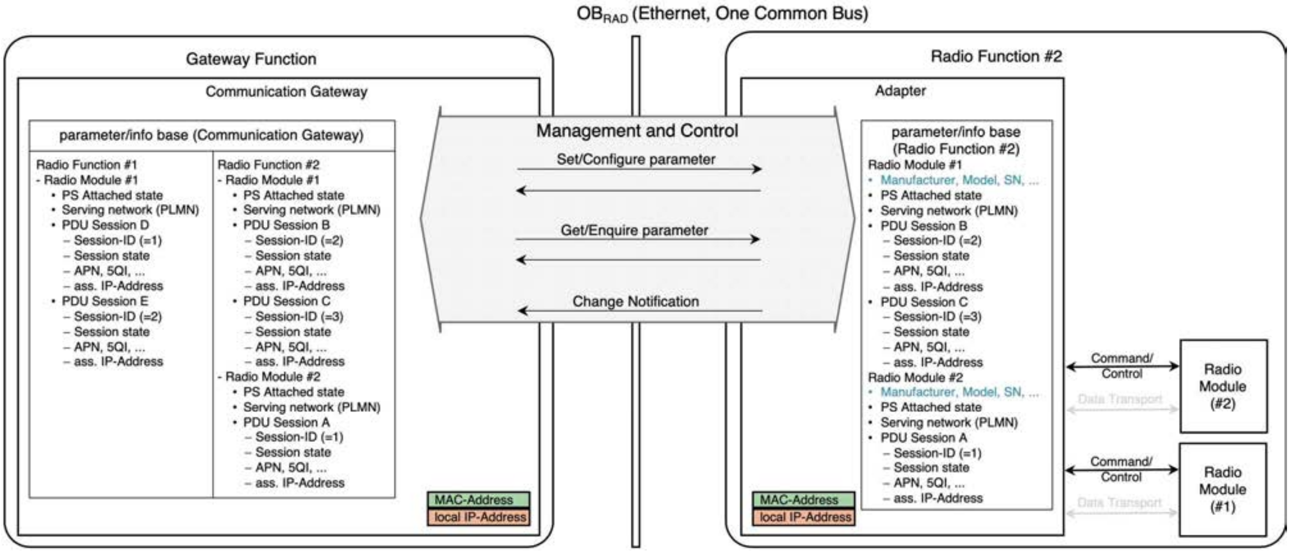


Figure 6.7: Management and Control protocol elements

The example in Figure 6.7 shows the "parameter/info base (Communication Gateway)" containing parameters of two Radio Functions #1 (hosting Radio Module #1) and #2 (hosting Radio Modules #1 and #2), while the Radio Function #1 itself is not shown in the Figure. It should be mentioned that the Communication Gateway might be interested only in a subset of the Radio Functions parameter list (in this example the light blue coloured parameters are (currently) not of interest for the Communication Gateway).

For the realization of such a "parameter centric" Management and Control protocol, the use of Simple Network Management Protocol, version 3 (SNMPv3) has been further analysed and its protocol elements are shown in Figure 6.8.

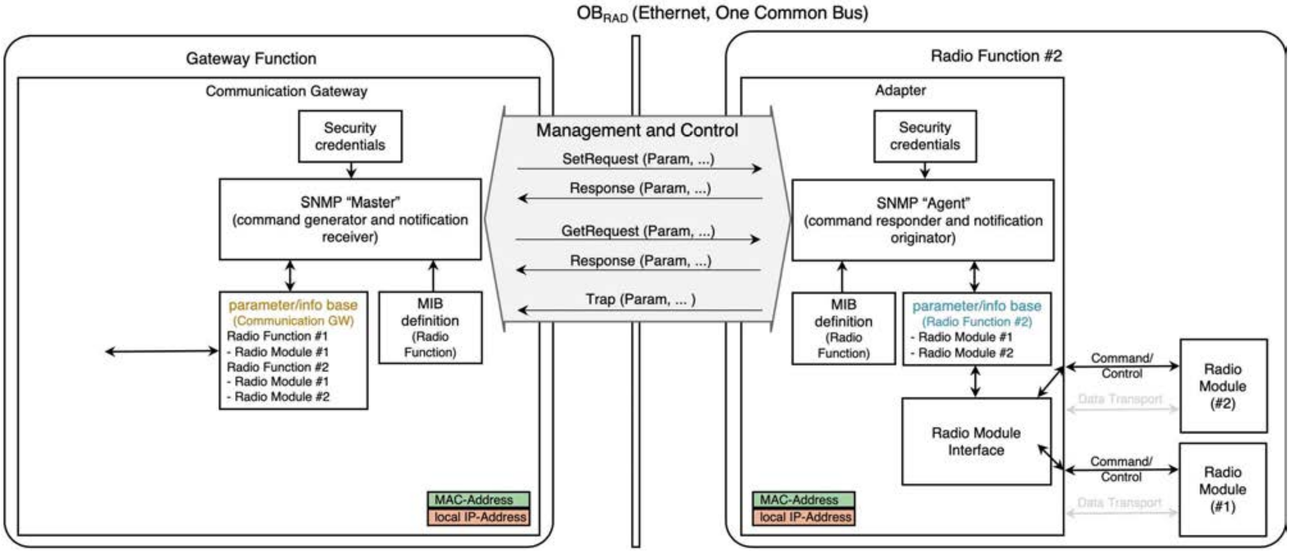


Figure 6.8: Use of SNMPv3 as Management and Control protocol

The SNMP-Entity (e.g. available under Linux) of the Communication Gateway needs to be configured as the SNMP "Master" (in old SNMP terminology) or as the "command generator and notification receiver" (in newer/current terminology), while the SNMP-Entity of the Radio Function(s) needs to be configured as the SNMP "Agent" or "command responder and notification originator".

Both SNMP-entities have access to their specific parameter/info base which stores the relevant parameters.
On the Communication Gateway (SNMP "Master") side, one or more parameter(s) are stored/updated in its "parameter/info base (Communication GW)" when the "SetRequest" or "GetRequest" is used, or upon reception of a "Trap" (Change Notification).

On the Radio Function (SNMP "Agent") side, one or more parameter(s) in the "parameter/info base (Radio Function)" are get/enquired by the Communication Gateway (SNMP "Master") via "GetRequest"; and one or more parameter(s) are set/configured by the Communication Gateway (SNMP "Master") via "SetRequest". A change/update of specific parameters (via "SetRequest") may lead the Radio Function to trigger/execute an appropriate action towards a Radio Module (e.g. PDU Session Establishment, Initial Registration, De-registration, etc.).

The parameters in the "parameter/info base (Radio Function)" may also be changed/updated by the Radio Module(s) via the "Radio Module Interface". Upon detection of a parameter change/update, the SNMP "Agent" indicates the changes by sending a "Trap" (Change Notification) to the SNMP "Master", which then updates its parameter/info base and may perform appropriate actions.

EXAMPLE: A Radio Module indicates "out of coverage" to the "Radio Module Interface", the "Radio Module Interface" updates the coverage status in the parameter/info base. Upon a parameter change (here: from "in coverage" to "out of coverage") the SNMP "Agent" sends a "Trap" (Change Notification) including the changed/updated parameter(s) and its new value(s). The SNMP "Master" updates that coverage status in its parameter/info base, which leads the Communication Gateway to perform an action to check (in its parameter/info base) whether at least one of the Radio Modules shows "FRMCS availability", and - in case there is no longer FRMCS available - indicate this to the registered Application(s).

It should be emphasized that in Figure 6.8, "parameter/info base (Radio Function #2)" (coloured in light blue) may contain a different set than in "parameter/info base (Communication GW)" (coloured in light gold): the Radio Function contains all the parameters of that Radio Function #2, while the one for Communication Gateway contains the collection of all Radio Functions, and it may contain only a subset of their parameters (e.g. without Manufacturer/Model info, without SW/HW Revision info, without Diagnostics info, etc.).

Both SNMP-entities need to be "fed" with a Management Information Base (MIB) "Radio Function" (including Traps/Notifications) to provide both SNMP-Entities the information on how to access the parameters in their parameter/info base (parameter name, access conditions, type of parameter, value range, etc.) upon GetRequest, SetRequest and Trap. This MIB "Radio Function" is the same used by both SNMP-Entities. This MIB "Radio Function" is needed in addition to the standard system "MIB-2" (not shown in Figure 6.8), which is used for initial device detection, SNMP system/diagnostics, etc. (see IETF RFC 3418 [i.21]).

Both SNMP-entities need to be "fed" with "Security credentials" for the security features available with SNMPv3. This has not yet been analysed in detail (see Summary at the end of this clause).

Using SNMP as Management and Control protocol would also enable the Operation and Maintenance (OM) entity of the Gateway Function to manage, control and monitor the Radio Function(s). The OM entity would need to implement similar SNMP protocol entities as the Communication Gateway, shown in Figure 6.9.

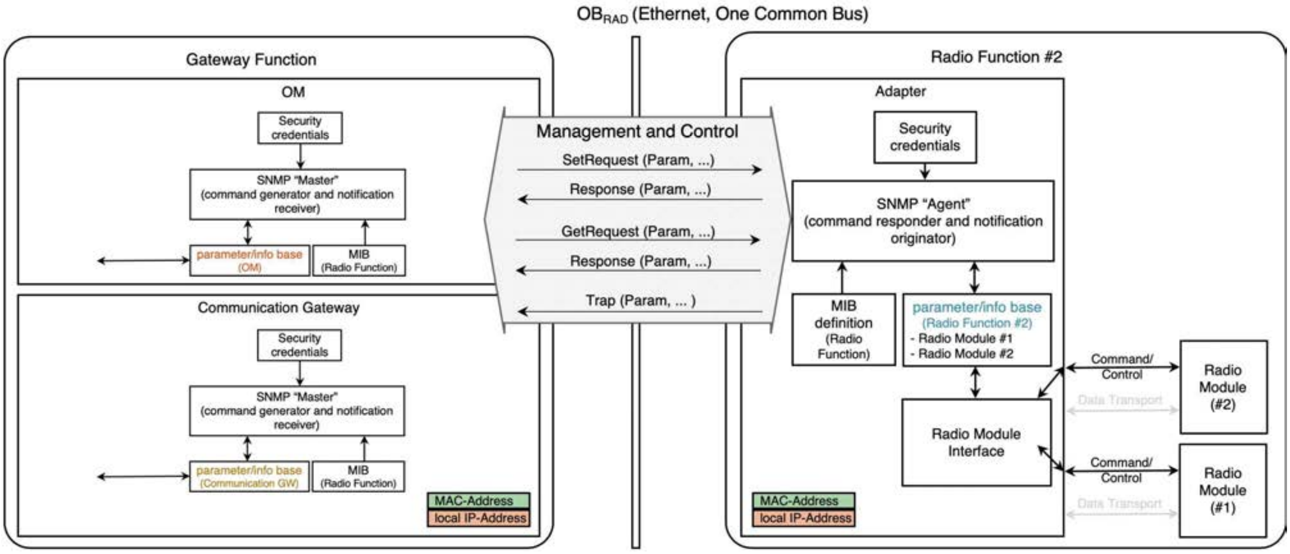


Figure 6.9: Use of SNMP as Management and Control protocol for OM

It should be emphasized that in Figure 6.9, all three parameter/info bases (for Radio Function #2 (coloured in light blue), for Communication Gateway (coloured in light gold) and for OM (coloured in light red)) may/will contain a different set of parameters: the Radio Function contains all the parameters of that Radio Function #2, while the one for Communication Gateway contains the collection of all Radio Functions but only a subset which are required for connectivity management and control, while the one for OM may contain all parameters of all Radio Functions (e.g. including Manufacturer/Model info, SW/HW Revision info, Diagnostics info, etc.).

The MIB "Radio Function" is the same used by all three SNMP-Entities.

Summary

- Proposal to use SNMP (SNMPv3) as Management and Control protocol, standardized by IETF in a set of RFCs [i.13], [i.14], [i.15], [i.16], [i.17], [i.18], [i.19], [i.20], [i.21] and [i.22].
  There are several other RFCs about SNMP available, but many of them are obsoleted by the above documents. Even if the title of some of above RFCs mentions "Version 2", these are valid for SNMPv3 as well, as SNMPv3 is SNMPv2 plus security.

- SNMP uses UDP as transport IP (IETF RFC 3417 [i.20]).

- Control Messages (SetRequest, GetRequest, Response) are using port 161, while Control Message "Trap" (Notifications) is using port 162 (IETF RFC 3417 [i.20]).

- In addition, there are the following Control Messages defined:

  - GetNextRequest;

  - GetBulkRequest;

  - InformRequest (same as Trap, but acknowledged by response "Report");

  - Report.

- The parameter/info base (parameter set) for a Radio Function needs to be defined/customized.
- Based on that parameter set, a MIB (Management Information Base) "Radio Function" (including Traps/Notifications) needs to be defined/customized.
- Control Messages are generic (SetRequest, GetRequest, Response, Trap), the parameters are selected out of what is provided by the MIB.
- The same protocol/MIB could be (re)used by OM to manage, control and monitor the Radio Function(s) (and their hosted Radio Modules);
  this requires the Radio Functions parameter set to be the (mathematical) "set union" of parameters for Gateway Connectivity and OM, i.e. to cover parameters of interest for Gateway Connectivity as well as for OM.

Further items to be analysed/studied:

- Security functionality of SNMPv3.
- Protocol for SW update of Radio Function(s) (their hosted Radio Modules and the Adapter) by OM; this is not covered by SNMP.

6.4 Proposal C: I/O streams with GTP

6.4.1 Introduction

It is here proposed a solution based on an underlay network topology using GTP for connectivity between a Gateway Function and one or more Radio Function(s) as part of On-Board FRMCS. This means and should be understood that any set of other protocols (e.g. SNMP, MQTT, REST/HTTP(s), SIP, RTP, TLS, UDP, TCP, SCTP, MPTCP, MP-QUIC, etc.) can transparently be managed as either overlay or as parallel networking protocol(s) for communication service(s) interfacing with any railway application with various needs for network service(s).

To enable interchangeability, a set of protocols would need to be selected. If such selected set of protocols contains more than one protocol, a mechanism for discovery of protocols and capabilities between entities would be necessary. A discovery mechanism could be based on e.g. Link Layer Discovery Protocol (LLDP, IEEE 802.1AB [i.27]) potentially with media endpoint discovery extension (LLDP-MED, ANSI/TIA-1057 [i.28]).

A discovery mechanism could also increase maintainability if initially considered. A discovery mechanism could be difficult to retrofit.

A mechanism based on e.g. LLDP or LLDP-MED may also become useful for maintenance purposes, e.g. inventory management and/or automated verification with acceptance and configuration of attaching HW components, i.e. to provide "plug and play" capabilities.

GTP-U as part of GTP has a potential capability to support FMCS Multipath (to reuse it as part of FS_{MPM}). GTP-U is a widely deployed and used protocol starting from 2,5G and is still used in 5G for encapsulation and transmission of user data as payload. One example is e.g. UPF to UPF communication via reference point N9.

GTP has the potential to accommodate the requirements so far identified in:

- Table 4.1: Functional requirements related to OB_{RAD} from UIC FRMCS TOBA FRS [i.1].
- Table 4.2: System requirements related to OB_{RAD} from UIC FRMCS SRS [i.3].

The referenced existing specifications and protocol(s) are defined and specified in ETSI TS 138 415 [i.23], ETSI TS 129 281 [i.24] and ETSI TS 129 274 [i.25].

The proposed commonly available technology is based on I/O streams. I/O streams are in this context to be understood as a generic stream of unstructured data (set of octets) without semantic meaning.

Practically, I/O streams can be implemented by e.g. epoll(), poll() or select() in conjunction with e.g. socket() and packet(). Epoll(), poll(), select(), socket() and packet() are system calls available in many platforms as part of an operating system (OS). This approach could allow the support of an OB_{RAD} implementation using any type of OSI layer 2 connection regardless of physical implementation(s) in the sense of whether it is embedded in a single physical box or not. The limiting factor would be given by the capabilities in terms of KPIs that an OSI layer 2 connection is capable of.

NOTE 1: OSI layer 2 connection is within this clause to be understood as one of:

- a peer-to-peer connection between 2 communicating entities;
- a peer-to-multi-peer connection, e.g.:
    - Ethernet controllers transmitting and receiving Ethernet frame(s) that encompasses unstructured user data;
    - Universal Asynchronous Receiver and Transmitters (UARTs) transmitting and receiving unstructured user data.

NOTE 2: Unstructured user data is within this context to be understood as a set of octets with no explicit and predefined definition of what the octets actually represent and means in a specific context. Context and meaning are created only by applying a certain communication protocol for encoding and/or decoding.

Based on the unstructured nature of any data transmission(s), I/O streams are transparent with regards to any type of protocol above OSI layer 3 (e.g. SIP/SDP for MC service(s), IMS, SNMP, etc.), any type of coupling mode (e.g. LC or TC), any type of Protocol, any type of E2E addressing scheme, any type of Physical media and any type of Radio Access Technology embedded within a Radio Function or Radio Module.

I/O streams as a concept would potentially have the capability to enable protocol maintainability providing upgrade or change capabilities for version(s) of e.g. GTP-C or GTP-U. This would require some mechanism for the correct and unique detection, configuration and activation of protocol or protocol version applied to I/O streams. Careful consideration on conditions for any protocol maintenance activity would be needed regarding railway operation in terms of safety, security, reliability and railway interoperability.

I/O streams as a concept would potentially also have the capability to support other protocol(s) from other domain(s) outside of the railway system, e.g. using part(s) of ITS framework in conjunction with Data Distribution Service (DDS). It would require further effort with preparations and considerations on feasibility and security requirements prior to any study on such interworking between system domains. A potential placeholder object for such study could be the OB_{RAD} reference point in conjunction with the FS_{ONI} reference point as identified within UIC FRMCS SRS [i.3].

Another potentially interesting study item at some future point in time could be to support Packet Forwarding Control Protocol (PFCP) as specified in ETSI TS 129 244 [i.26].

Figure 6.10 illustrates I/O streams with GTP as a concept study architecture.

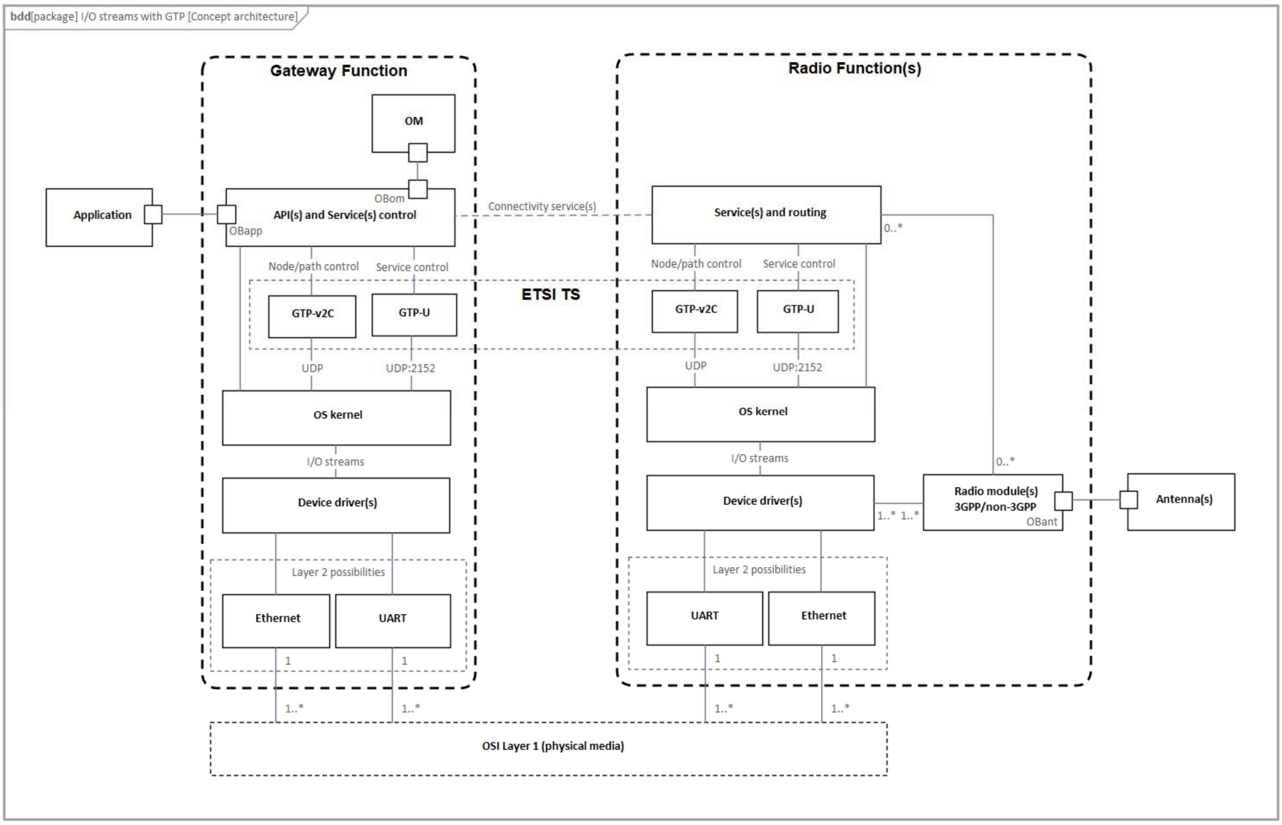


Figure 6.10: Concept study architecture

NOTE 3: The indicated connectivity between the conceptual blocks "Radio Module(s)" and "Service(s) and routing" is to be understood as conditional depending on whether a Radio Module:

1) Is to be considered as managed by GTP or not.

2) Need the capability of specific Application protocol routing by the Connectivity service(s) as conceptual and functional interface between a Gateway Function and one or more Radio Function(s).

3) Combination of 1) and 2).

6.4.2 OB$_{RAD}$ versus I/O streams with GTP relation

6.4.2.1 General

Figure 6.11 visually describes the relations between OB$_{RAD}$ and GTP in their relevant specification domains.

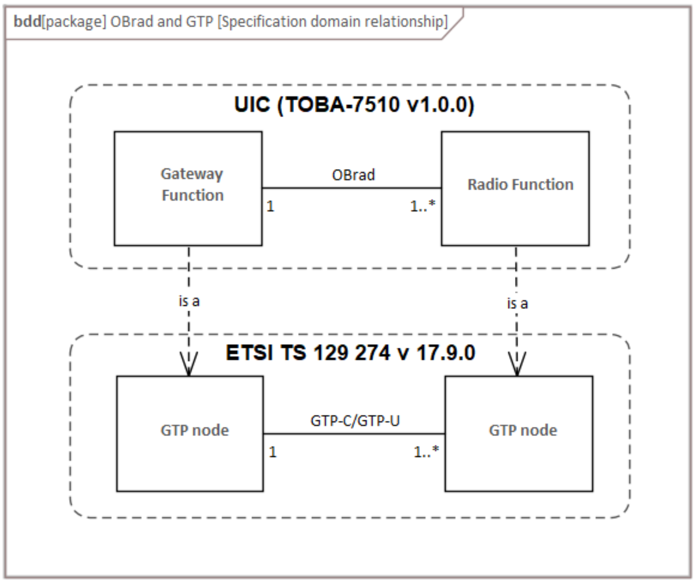


Figure 6.11

Figure 6.12 intends to visually describe the most relevant definitions in ETSI TS 129 274 [i.25] in relation to Figure 6.11.

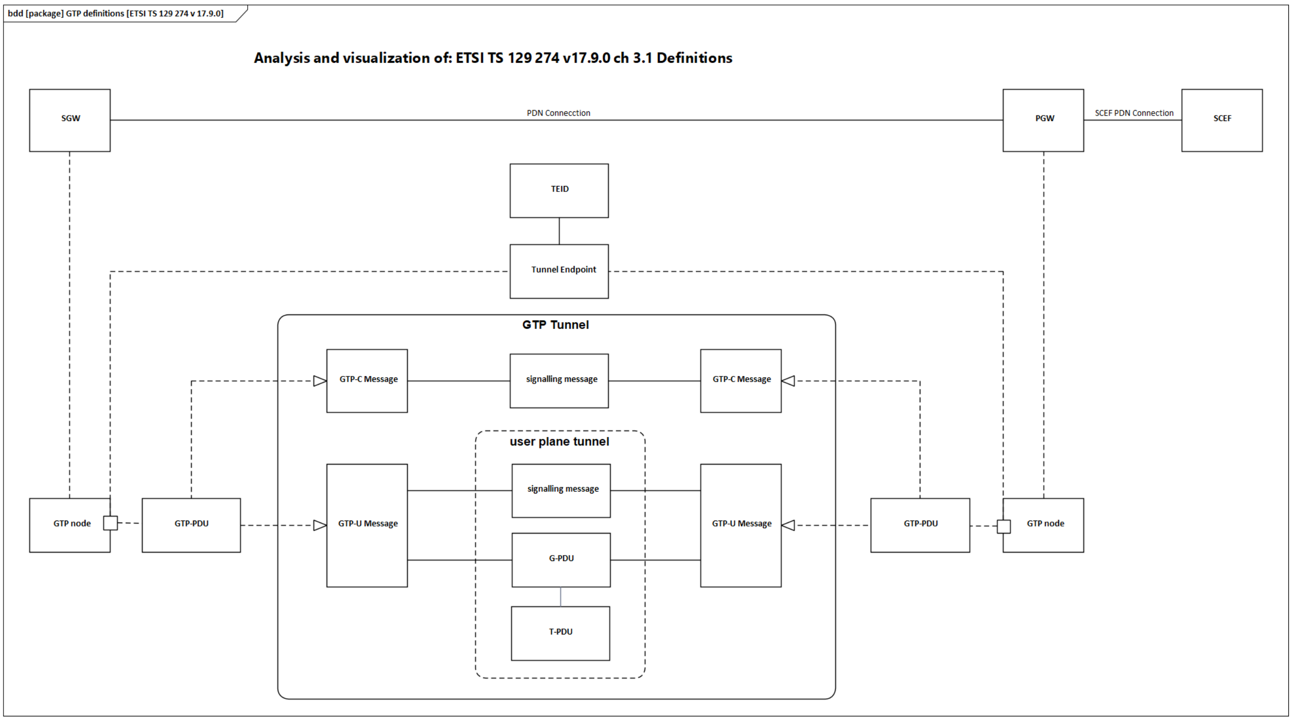


Figure 6.12

In GTP, unstructured user data is represented by T-PDU(s). T-PDU(s) are normally used to transmit and receive IP packet(s) related to a GTP tunnel.

Since OB$_{RAD}$ is a defined part of an FRMCS E2E system with initial support for a set of service(s) and feature(s) for a well-defined set of railway applications, and with the need for expansion capabilities by e.g. FRMCS version(s) or release(s), GTP appears to be a valid protocol candidate. It would rely on existing and already available ETSI technical specifications that are publicly available and used to exchange communication encompassing voice-, data- and video service(s).

GTP has the potential of supporting any kind of protocol on top of it, which may also indicate the need for isolation between GTP and other protocol(s). This could be achieved by using e.g. network namespaces in conjunction with other networking facilities available in modern operating system(s). A non-exhaustive list of networking facilities is Virtual Routing and Forwarding (VRF), Virtual LAN (VLAN), Virtual Ethernet (VETH) and Media Access Control security (MACsec).

6.4.2.2 PDU session

PDU session(s) can be supported by Additional Protocol Configuration Options (APCO) (see ETSI TS 129 274 [i.25]).

6.4.2.3 GTP-U

GTP-U and GTP-U Messages can potentially be used as a basic protocol for OB_{RAD} Data Transport. This would lead to the following relationships between OB_{RAD} and GTP:

- Gateway Function as defined in UIC FRMCS TOBA FRS [i.1] would be analogous with a GTP node as defined in ETSI TS 129 274 [i.25] as an instance of SGW.

- Radio Function(s) as defined in UIC FRMCS TOBA FRS [i.1] would be analogous with one or more GTP node(s) as defined in ETSI TS 129 274 [i.25] as instance(s) of PGW.

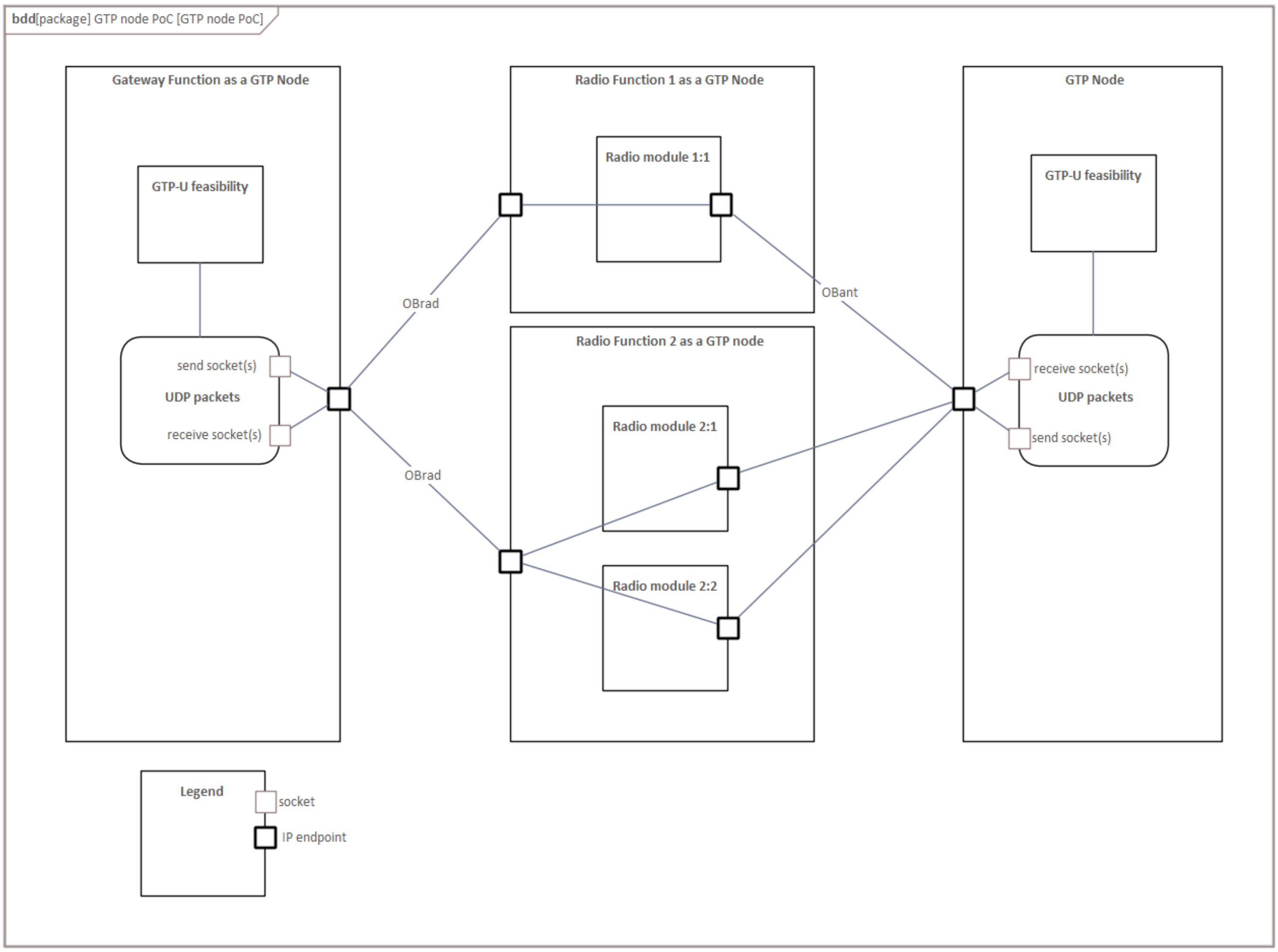
GTP-U and GTP-U Messages can be used to transmit and receive:

- T-PDU(s) with user data as payload (encapsulated in G-PDU(s)) for any OB_{RAD} Management and Control protocol proposed by the present document.

- T-PDU(s) with user data as payload (encapsulated in G-PDU(s)) for the Application Plane.

- GTP-U tunnel management for user plane tunnel(s) and control plane messaging.

A simple concept architecture to study feasibility of GTP with focus on UDP packet(s) (as an example) for GTP-U could be illustrated by Figure 6.13.



NOTE: This figure reuses the configuration as previously given by Figure 4.2 in the present document.

Figure 6.13: GTP and UDP packet(s) feasibility

A non-exhaustive list of potential issues to address are:

- IP address change(s) for radio module(s) due to mobility.
- General robustness and quality of UDP traffic in uplink and downlink direction using OB_{ANT} (sending versus receiving) with respect to e.g. packet error rate, packet loss rate, etc.

Figure 6.14 attempts to address the issue of IP address change(s) for Radio Module(s), by using e.g. Virtual Routing and Forwarding, which could potentially decouple the process of IP address management between the Gateway Function and connected Radio Functions.

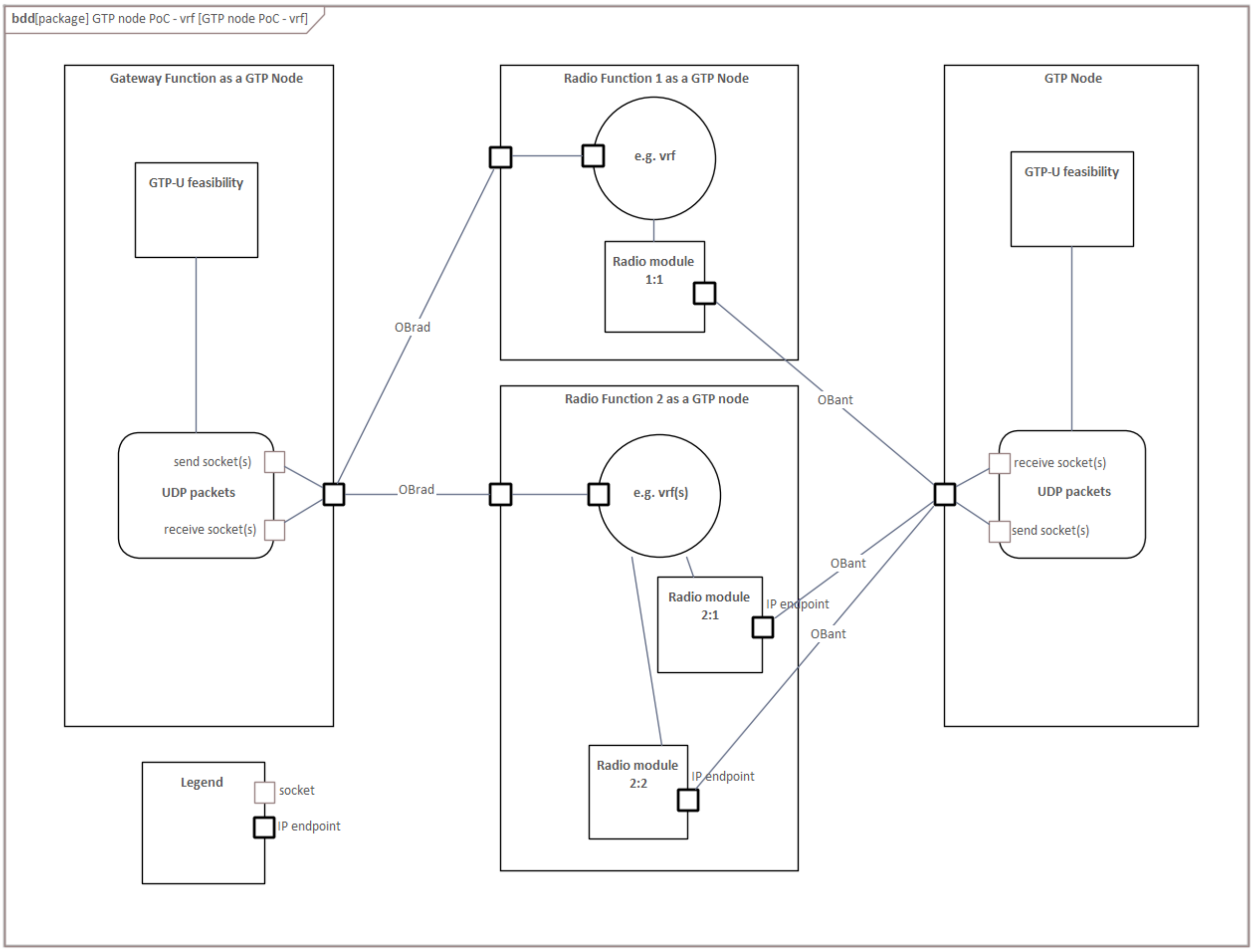


Figure 6.14: GTP and UDP packet(s) feasibility and IP addressing issue for Radio Module(s)

6.4.2.4 GTP-C

GTP-v2C (see ETSI TS 129 274 [i.25]) seems applicable for e.g. GTP node management. Within the scope of the present document and within this clause, Gateway Function and Radio Function as specified in UIC FRMCS SRS [i.3] could be considered as instance(s) of GTP node(s).

6.5 Proposal D: HTTP / MQTT API

6.5.1 Protocols presentation

This proposal is a combination of MQTT (e.g. for notifications) and HTTP (e.g. for actions/requests).

MQ Telemetry Transport or Message Queue Telemetry Transport (MQTT) is an ISO standard (ISO/IEC PRF 20922 [i.48]) publish-subscribe-based messaging protocol. It works on top of the TCP/IP protocol. Even though it may work on top of UDP, this mode of operation may have some drawbacks in terms of reliability. MQTT is designed for connections with remote locations where a "small code footprint" is required or the network bandwidth is limited. The publish-subscribe messaging pattern requires a message broker.

MQTT has been primarily developed for monitoring and status reporting of different equipment and applications.

On the other hand, HTTP [i.44] is a well-known protocol. Use of version 1.1 is proposed in the present document.

6.5.2 Implementation of OB$_{RAD}$ using HTTP and MQTT API

In this proposition, a Radio Function (RF) is composed of at least one Modem/Radio Module (RM) and a processing unit on which the drivers of the modems are installed as well as a webserver and an MQTT client (publisher).

On the On-Board FRMCS Gateway (GW) side, an MQTT broker is installed. On the Radio Function Management Function (RFMF) of the GW, there are an MQTT client (subscriber) and a HTTP client.

NOTE: The Radio Function Management Function (RFMF) is not appearing in UIC FRMCS specifications.

The MQTT publisher of each Radio Function publishes for each Radio Module of the Radio Function the following, non-exhaustive, list of information:

- its identifier (so that the GW can route packets through individual Radio Modules);
- its capacities:
    - 3GPP / non 3GPP;
    - supported control commands (or list of control command);
    - user plane IP address.
- status information:
    - list of established bearers;
    - connection quality;
    - Cell Id;
    - Tracking Area.

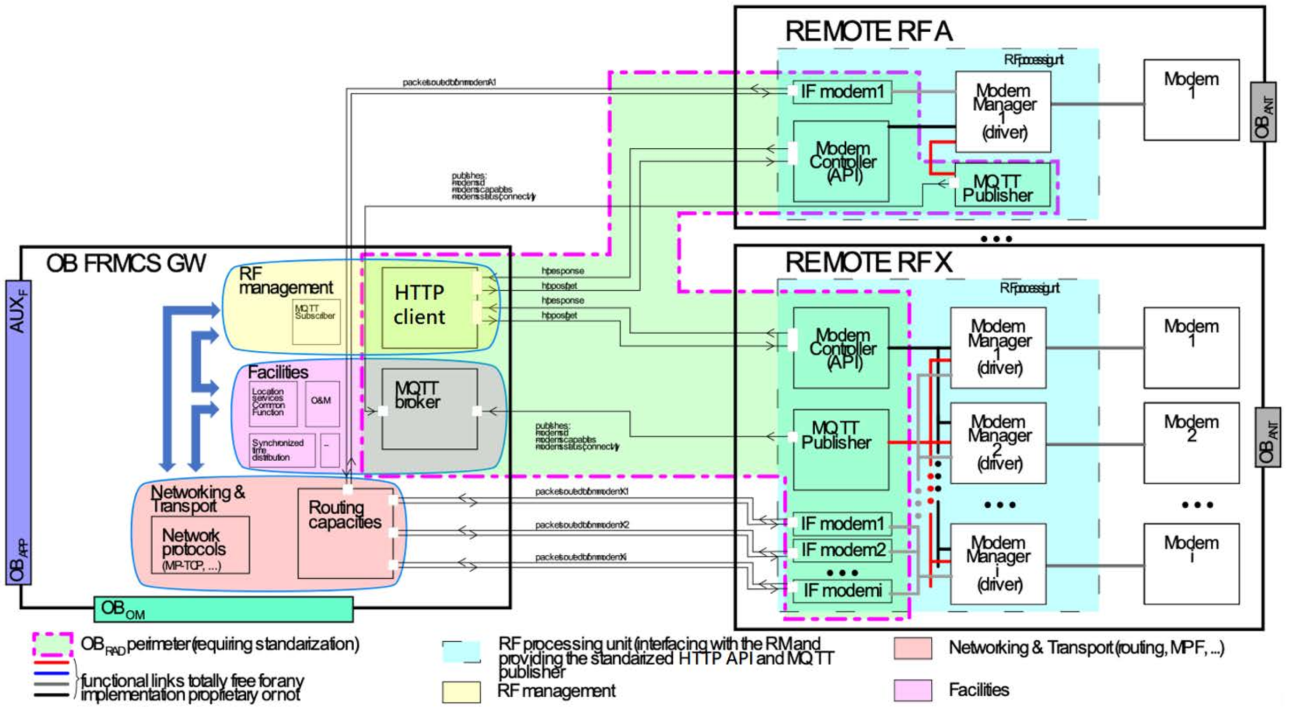


Figure 6.14a

The HTTP API allows to control the modems independently converting standardized call to the API to the right commands via the Modem Manager to the modem.

The GW RFMF directly routes the packets to the Radio Module according to its own algorithms/choices. It gets the states and capacities of each Radio Module via its MQTT subscriber. In this aspect, the Radio Function needs to act as a basic router.

Each Radio Module being identified by an IP address, the routing choices and capacities remain fully on the Gateway Function side as it would be with local physical modems.

Having a standardized API that allows the GW to control each Radio Module might also be convenient as it could as well be used for non-remote Radio Function.

6.5.3 Definition of the control commands

For the RFMF to be able to handle the Radio Module correctly, the most efficient way is to define a standardized set of commands accessible via the API and statuses reported by the MQTT publisher. This set may include some optional commands (whose existence may be notified by the MQTT publisher).

The Radio Module or the Radio Function vendor would have to make an adapter, mapping the standardized functions to its own control commands.

The HTTP API, the MQTT broker and the standardized set of commands jointly define a kind of standardized adapter for the Radio Module so that the On-Board FRMCS Gateway can manage any Radio Module in a standardized way.

6.5.4 O&M of the Radio Function

The HTTP API should also implement O&M functions for the Radio Function (allowing to control the configuration and updates or upgrades of the Radio Function - e.g. modem drivers and/or modem firmware - via the processing unit of the Radio Function). These Radio Function O&M functions should be accessible via the GW O&M function (and thus via OB_{OM} and FS_{OMR}). The Radio Function O&M capabilities exposed by the OB_{RAD} HTTP API should include:

- Software upgrades of the Radio Function processing unit (including Radio Module drivers updates/upgrades).
- Firmware updates/upgrades of the Radio Module.
- Configuration of the MQTT publisher, HTTP server and other pieces of software hosted by the Radio Function.

6.5.5 Key points of this proposed solution

6.5.5.1 Impacts on the GW architecture

On the GW side, the main impact is the implementation of an MQTT broker and client. Open sources lightweight brokers and clients, like Eclipse Mosquitto [i.49], are available. The other impact on the GW is the fact that the RFMF has to use a HTTP API and to subscribe to the broker in order to manage the Radio Function.

6.5.5.2 Identifier to use for the virtual interfaces representing the Radio Module

Use of an IP address or a port on the Radio Function IP address as identifier can be adequate. The transport between the GW and the Radio Function is an IP network, so it seems useless to define a lower OSI level identifier that would need to be encapsulated in an IP packet to be handled by the Radio Function processing unit.

Use of a lower-level identifier (such as MAC or any layer 2 identifier) is possible, but might lead to a more complicated implementation. The added value of such a lower-level identifier might need to be explained in detail.

6.5.5.3 Network to be used between the GW and the Radio Function

Lot of trains that will need remote Radio Function for migration from GSM-R to FRMCS do not have an existing secure IP network. The use of any existing IP network (even comfort networks) should be eligible as candidate transport medium between the GW and the Radio Function, provided that it meets security requirements which would need to be further specified. Support of flow separation (e.g. through VLAN) and some priority mechanisms might help in demonstrating security compliance. In case of absence of such network, installation of a dedicated IP link between the GW and the Radio Function might be used; in this kind of integration, the GW might have multiple ethernet ports for that purpose or be connected to an intermediate switch or router in order to allow connecting multiple remote Radio Function.

6.5.5.4 Splitting of the control plane in two protocols (HTTP and MQTT)

The proposal foresees the usage of HTTP/1.1 for the HTTP API, and there was no reverse communication implemented from the server (i.e. the Radio Function) to the client (i.e. the Gateway Function). So, the proposition was made to use the well suited MQTT instead. Another possibility is to use http2 or http3 that allow reverse communication. Nevertheless, the use of MQTT allows to give a simple existing framework consistent with ITxPT specification TR3-003 [i.10].

NOTE: ITxPT TR3-003 [i.10], clause 9.4.2 identifies a possible alternative widely used in the IoT world whereby MQTT is used both for the status part and for the command part.

6.6 Proposal E: MQTT (Management and Control protocol)

6.6.1 Introduction

This proposal is about the use of the MQTT protocol for Management and Control protocol.

As shown in clause 6.3.2 "IP-in-IP encapsulation (Data Transport protocol)", the bi-directional exchange of the Communication Session parameters contained in the "parameter/info base" between the Gateway Function and the Radio Function(s) (keeping them "up-to-date") is an essential pre-requisite for the Data Transport protocol (Data path routing) which needs to be ensured by the Management and Control protocol as highlighted in Figure 6.15.

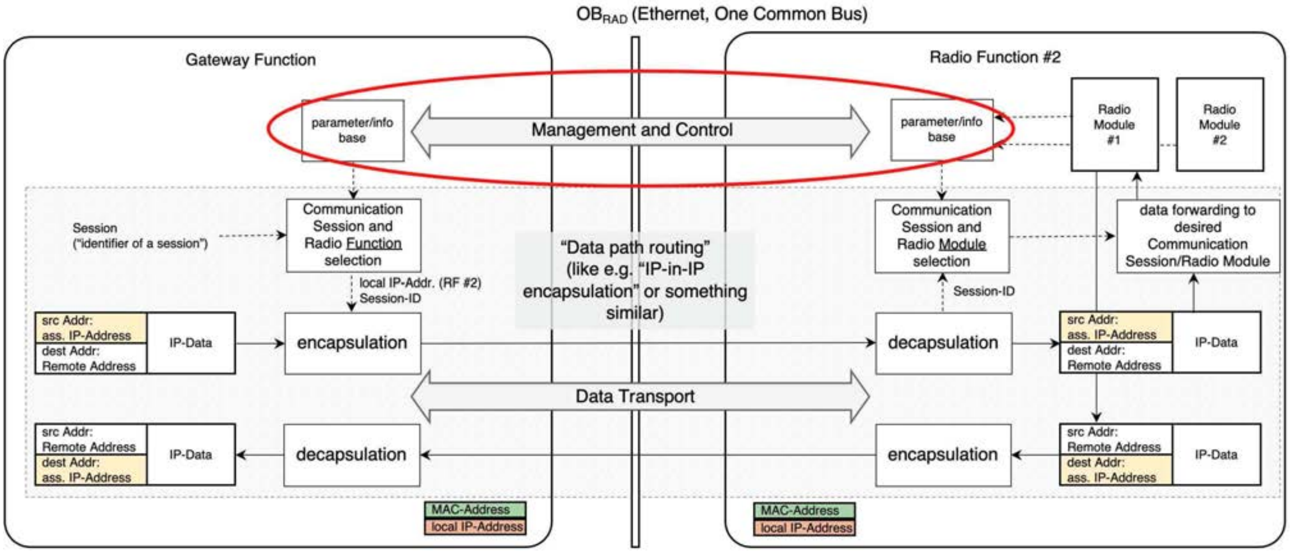

OB$_{RAD}$ (Ethernet, One Common Bus)
Gateway Function
Radio Function #2
parameter/info base
Management and Control
parameter/info base
Radio Module #1
Radio Module #2
Session ("identifier of a session")
Communication Session and Radio Function selection
local IP-Addr. (RF #2)
Session-ID
"Data path routing" (like e.g. "IP-in-IP encapsulation" or something similar)
Communication Session and Radio Module selection
data forwarding to desired Communication Session/Radio Module
Session-ID
src Addr: ass. IP-Address
dest Addr: Remote Address
IP-Data
encapsulation
decapsulation
src Addr: ass. IP-Address
dest Addr: Remote Address
IP-Data
Data Transport
src Addr: Remote Address
dest Addr: ass. IP-Address
IP-Data
decapsulation
encapsulation
src Addr: Remote Address
dest Addr: ass. IP-Address
IP-Data
MAC-Address
local IP-Address
MAC-Address
local IP-Address

Figure 6.15: Management and Control to Data Transport protocol relationship

6.6.2 A generic "parameter centric" approach

From a generic perspective, the Management and Control protocol needs to provide functionalities to enable the Gateway Function to:

- Command Set/Configure parameter(s) in a Radio Function.

- Command Get/Enquire parameter(s) from a Radio Function.
- Receive Change Notifications (of parameter(s)) from a Radio Function;

as well as to provide a set or subset of defined parameters, as shown in Figure 6.16.

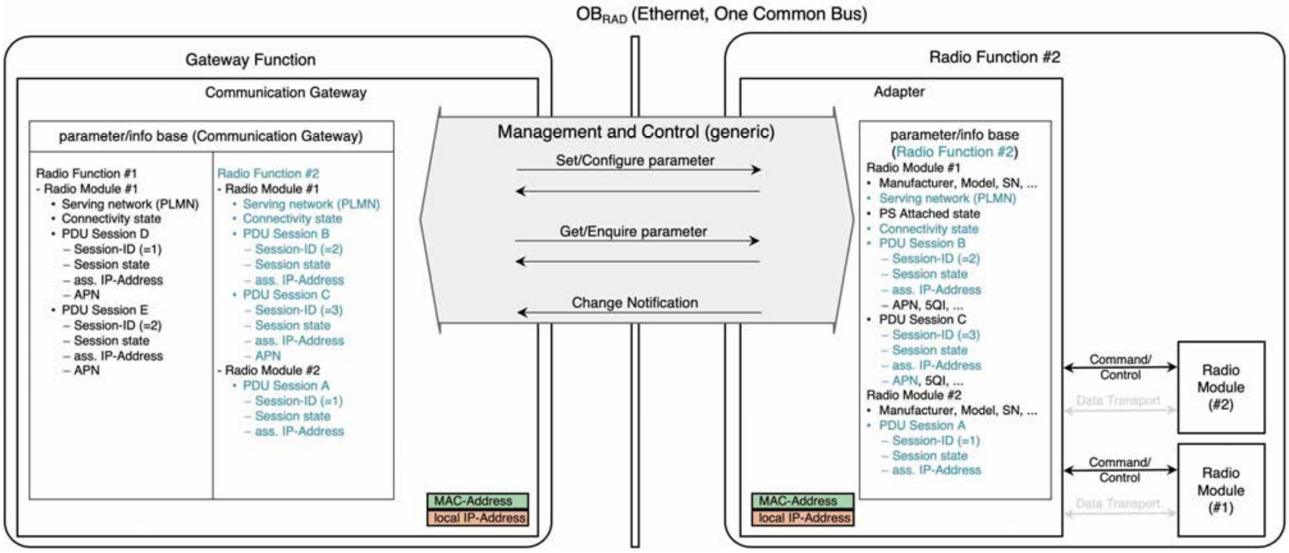


Figure 6.16: Management and Control protocol elements

The example in Figure 6.16 shows the "parameter/info base (Communication Gateway)" containing parameters of two Radio Functions #1 (hosting Radio Module #1) and #2 (hosting Radio Modules #1 and #2), while the Radio Function #1 itself is not shown in the figure. It should be mentioned that the Communication Gateway might be interested only in a subset of parameters of a Radio Functions parameter list (in this example the blue coloured parameters).

6.6.3 MQTT protocol entities

For the realization of such a "parameter centric" Management and Control protocol, the use of MQTT (MQTT v5.0) [i.29] has been further analysed. MQTT is a Client Server publish/subscribe messaging transport protocol.

NOTE: The MQTT specification uses the elements "client" and "server", while in many articles about implementation, use cases etc. the terminology "broker" is used instead of "server".

A MQTT client can be configured as:

- a publisher; or
- a subscriber; or
- both publisher and subscriber.

In this proposal E, the MQTT clients are both publisher and subscriber, as shown in Figure 6.17.

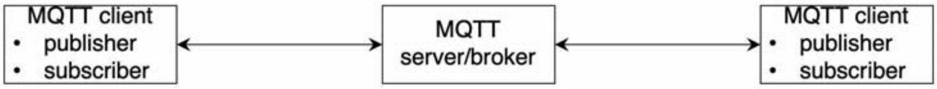


Figure 6.17: MQTT protocol entities

6.6.4 MQTT messages ("MQTT Control Packets")

The MQTT client(s) and the MQTT server/broker communicate via a defined set of messages ("MQTT Control Packets"), as shown in Figure 6.18.

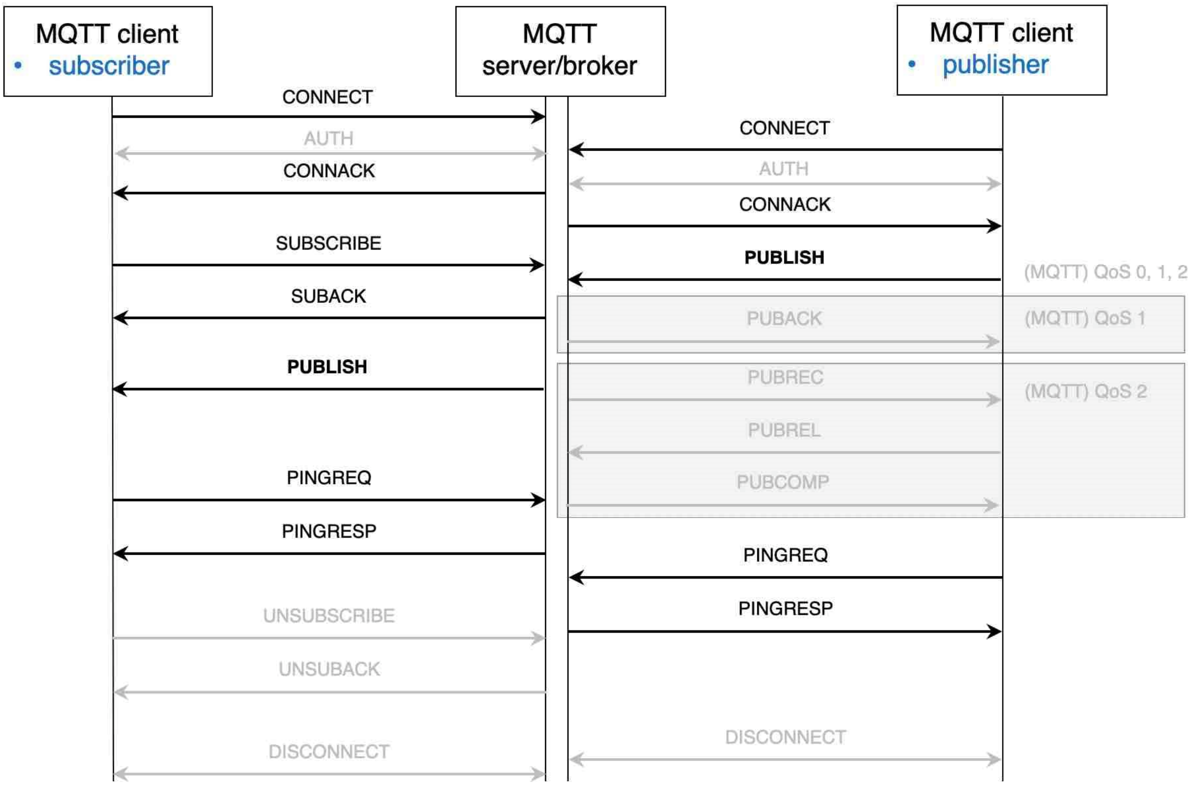


Figure 6.18: MQTT protocol messages ("MQTT Control Packets")

An MQTT client, configured as a subscriber, subscribes to one or more "topics" (i.e. parameter) by using the SUBSCRIBE/SUBACK messages.

An MQTT client, configured as a publisher, publishes one or more "topics" (i.e. parameter) and their values by using the PUBLISH message. Depending on the chosen (MQTT) QoS (QoS 0, 1 or 2), additional messages (PUBACK, PUBREC, PUBREL, PUBCOMP) may be exchanged between the MQTT server/broker and the MQTT client(s) within the Publish Procedure.

NOTE: The relationship server/broker to subscriber for the additional messages (PUBACK, PUBREC, PUBREL, PUBCOMP) is not shown in Figure 6.18.

6.6.5 MQTT for OB$_{RAD}$ Management and Control protocol

The proposed solution for the OB$_{RAD}$ Management and Control protocol for both Connectivity Gateway and OM using MQTT and its protocol entities is shown in Figure 6.19.

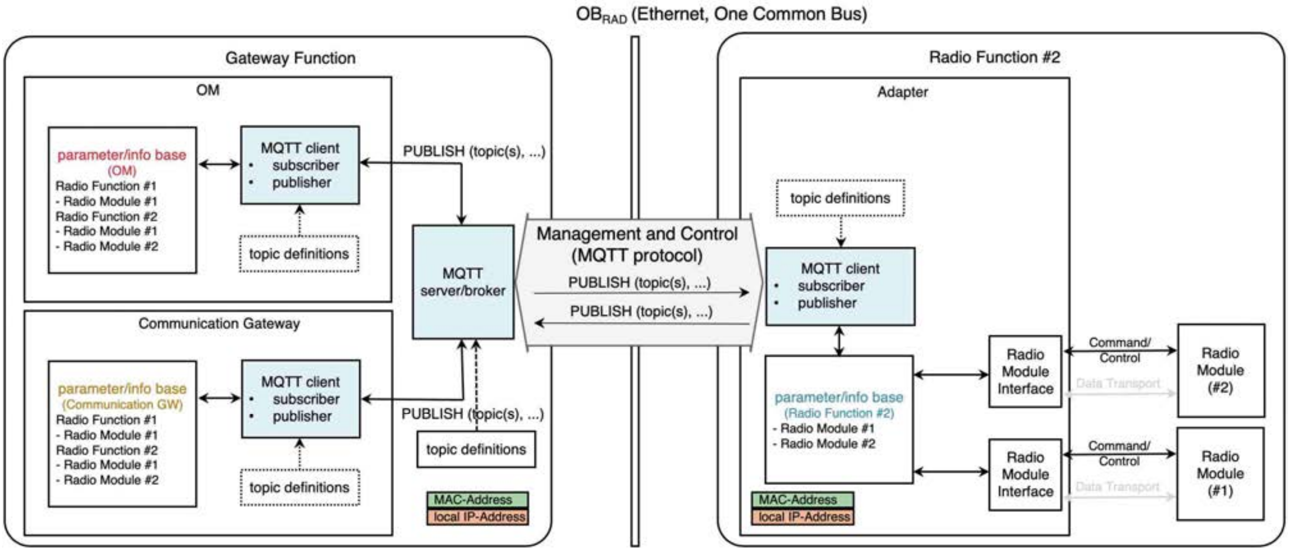


Figure 6.19: Use of MQTT as OB$_{RAD}$ Management and Control protocol

There is one MQTT server/broker in the Gateway Function, one MQTT client for the Connectivity Gateway, one MQTT client for the OM of the Gateway Function, and one MQTT client in every Radio Function Adapter. As an implementation option within each Radio Function, multiple Radio Modules may interact with a shared MQTT client (of the Adapter), or each Radio Module may use its own MQTT client. Every MQTT client has access to its individual parameter/info base, which stores and manages the relevant parameters of interest.

For each and every parameter an entity (Connectivity Gateway, OM or Radio Function) is interested in receiving updates about, its MQTT client subscribes to the corresponding topic at the MQTT server/broker.

The Radio Functions parameter/info base interacts via the Radio Module Interface(s) with the hosted Radio Module(s). Its parameters may be changed/updated by the Radio Module(s) via the Radio Module Interface. Upon detection of a parameter change/update, the (publisher) MQTT client (of the Radio Functions Adapter) "publishes" the changes by sending the updated topics via PUBLISH messages to the MQTT server/broker, which then "publishes" the updated topics to the (subscriber) MQTT clients which are subscribed to that topic. The (subscriber) MQTT clients update their parameter/info base which may perform/trigger appropriate actions.

EXAMPLE 1: A Radio Module indicates "out of coverage" to the "Radio Module Interface", the "Radio Module Interface" updates the coverage status in the Radio Functions parameter/info base. Upon a parameter change (here: from "in coverage" to "out of coverage") the MQTT client (of the Radio Function Adapter) sends a PUBLISH message including the changed/updated topic(s) and its new value(s) to the MQTT server/broker; the MQTT server/broker then "publishes" the updated topics to the subscribed MQTT clients (e.g. of Connectivity Gateway, OM). The MQTT client of Connectivity Gateway updates that coverage status in its parameter/info base, which leads the Communication Gateway to perform an action to check (in its parameter/info base) whether at least one of the Radio Modules shows "FRMCS availability", and - in case there is no longer FRMCS available - indicate this to the registered Application(s). The MQTT client of OM updates that coverage status in its parameter/info base e.g. for monitoring purposes.

It should be emphasized that in Figure 6.19 the three parameter/info bases may/will contain different sets of parameters:

- the "parameter/info base (Radio Function #2)" will contain all parameters a Radio Function provides (for the Adapter and its Radio Module(s));
- the "parameter/info base (OM)" may contain all parameters of all Radio Functions;
- and the "parameter/info base (Communication GW)" contains only those parameters of all Radio Functions, which are required for connectivity management and control (e.g. for determining and selecting the Data Path route(s) to the desired Communication Session of a Radio Module on a Radio Function).

The Radio Functions parameter/info base may also be changed/updated via their (subscriber) MQTT clients when requested by the Connectivity Gateway or OM. The Connectivity Gateway or OM may update/change parameter settings of a Radio Function by sending the updated topics via PUBLISH messages to the MQTT server/broker, which then "publishes" the updated topics to the (subscriber) MQTT client (of the Radio Function), which is subscribed to that topic(s). The (subscriber) MQTT client of the Radio Function updates its parameter/info base which may perform/trigger appropriate actions.

EXAMPLE 2: The OM wants to request the establishment of a Communication Session (e.g. a 3GPP PDU Session) on Radio Module #1 of Radio Function #2. In a first step it performs the desired parameter settings (e.g. APN etc.) in the OM parameter/info base, which then triggers the MQTT client (of the OM) to send a PUBLISH message including the changed/updated topic(s) and its new value(s) to the MQTT server/broker; the MQTT server/broker then "publishes" the updated topics to the subscribed MQTT client of the specific Radio Function #2, which updates the topic(s) in its parameter/info base. In a second step (as shown in Figure 6.20), the OM requests the establishment of the Communication Session (with the previously set/changed parameters) by "publishing" a topic "RequestSession = EstablishReq" via the MQTT server/broker to the (subscriber) MQTT client of the specific Radio Function #2. The MQTT client of (Radio Function #2) requests the Radio Module #1 via its dedicated/individual "Radio Module Interface" (e.g. AT-Command, QMI etc.) to establish the Communication Session. Once the Radio Module has accepted the requests, the MQTT client of the Radio Function "publishes" the topic "SessionState = EstOngoing" to both the OM and the Connectivity Gateway. After the Communication Session has been established, the MQTT client of the Radio Function "publishes" the topic "SessionState = Established" to both the OM and the Connectivity Gateway.

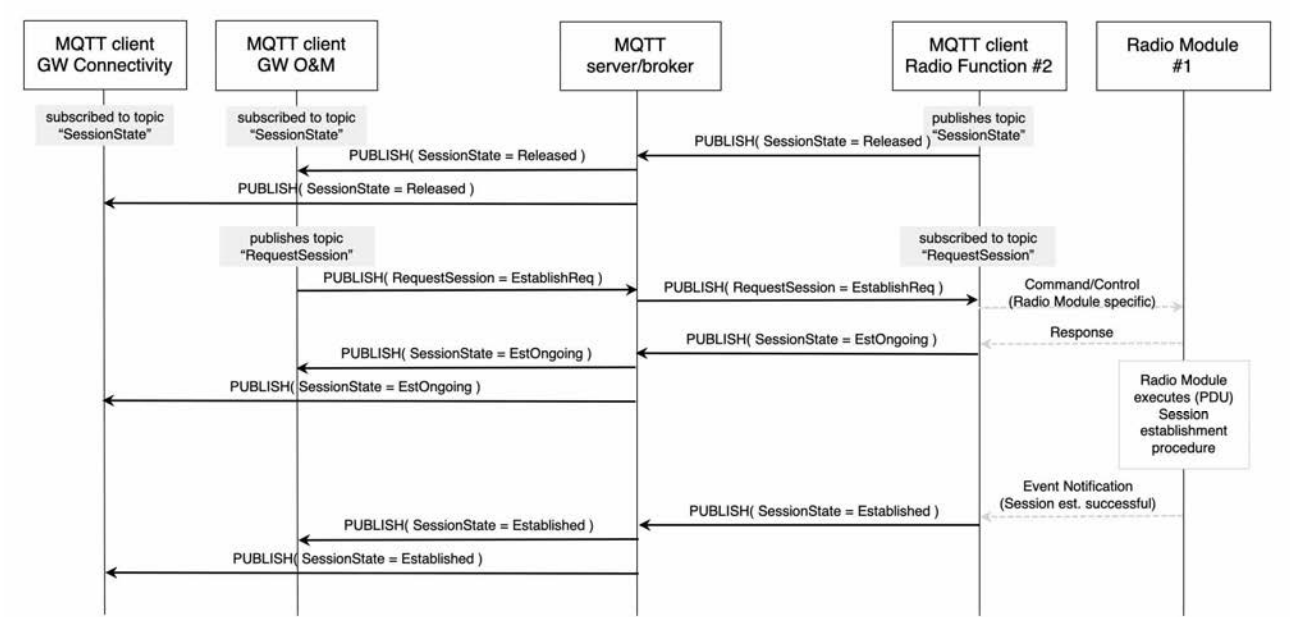
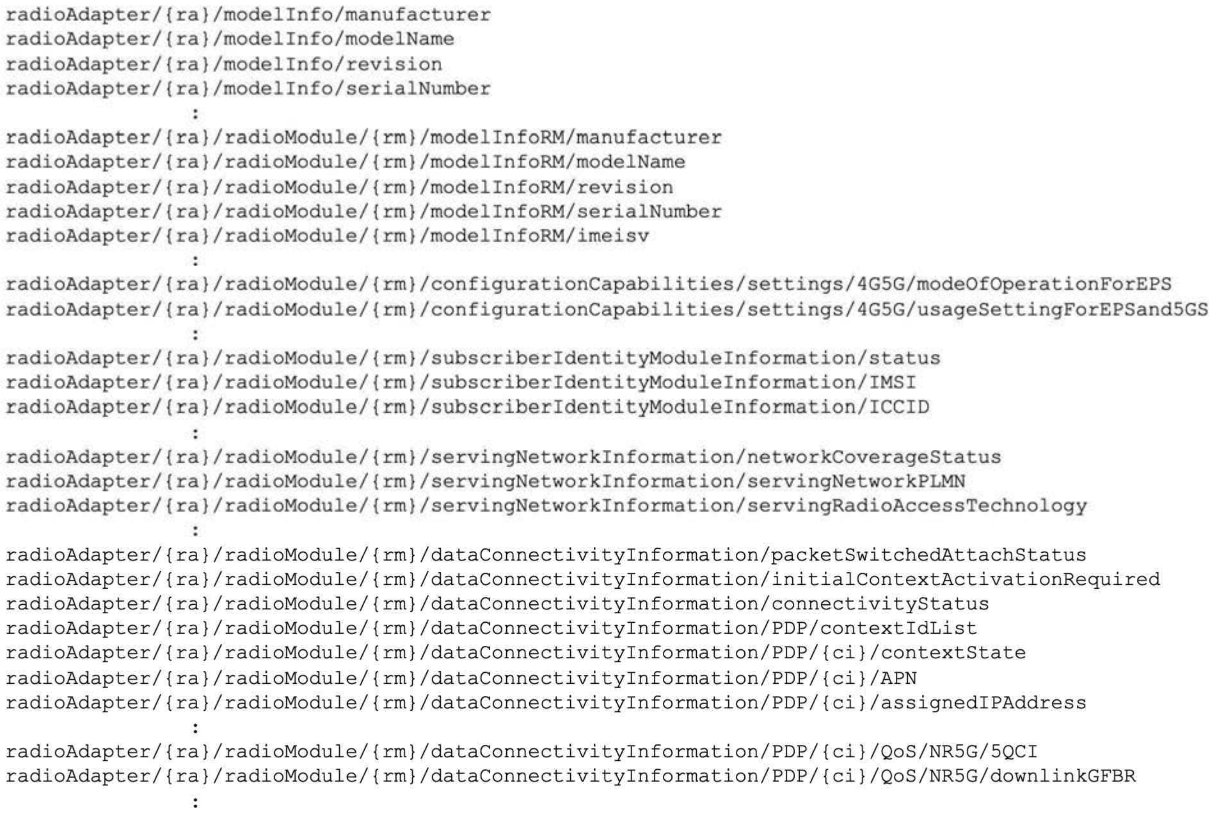

MQTT client GW Connectivity
MQTT client GW O&M
MQTT server/broker
MQTT client Radio Function #2
Radio Module #1
subscribed to topic "SessionState"
subscribed to topic "SessionState"
publishes topic "SessionState"
PUBLISH(SessionState = Released)
PUBLISH(SessionState = Released)
PUBLISH(SessionState = Released)
publishes topic "RequestSession"
subscribed to topic "RequestSession"
PUBLISH(RequestSession = EstablishReq)
PUBLISH(RequestSession = EstablishReq)
Command/Control (Radio Module specific)
Response
PUBLISH(SessionState = EstOngoing)
PUBLISH(SessionState = EstOngoing)
PUBLISH(SessionState = EstOngoing)
Radio Module executes (PDU) Session establishment procedure
Event Notification (Session est. successful)
PUBLISH(SessionState = Established)
PUBLISH(SessionState = Established)
PUBLISH(SessionState = Established)

Figure 6.20: Example for Request/Response procedure

6.6.6 MQTT topic structure/tree

As shown in Figure 6.19, all MQTT clients and the MQTT server/broker need to be "fed" with the "topic definitions". This is a configuration text file containing the defined "topic structure" (or "topic tree"), making it known to all MQTT entities. According to the MQTT specification this is not strictly required (a dynamic creation is supported). As stated in MQTT v5.0 [i.29], clause 4.7.3: *"The topic resource MAY be either predefined in the Server by an administrator or it MAY be dynamically created by the Server when it receives the first subscription or an Application Message with that Topic Name."* but it will be desired for a "standardized" OB_{RAD} (specification, implementation) and its operability to ensure which topics (name and location within the structure/tree) are available. Additionally, it should be noted, that to keep the hierarchical topic tree flexible, it is important to design the topic tree very carefully and leave room for future use cases. An example for a topic structure/tree (extract) is shown in Figure 6.21.

There are server/broker implementations available, which in addition to supporting access restriction to topics using an Access Control List (ACL), restrict topic access (publish, subscribe) to certain users or MQTT clients, even this is not part of the MQTT specification.



```
radioAdapter/{ra}/modelInfo/manufacturer
radioAdapter/{ra}/modelInfo/modelName
radioAdapter/{ra}/modelInfo/revision
radioAdapter/{ra}/modelInfo/serialNumber
                :
radioAdapter/{ra}/radioModule/{rm}/modelInfoRM/manufacturer
radioAdapter/{ra}/radioModule/{rm}/modelInfoRM/modelName
radioAdapter/{ra}/radioModule/{rm}/modelInfoRM/revision
radioAdapter/{ra}/radioModule/{rm}/modelInfoRM/serialNumber
radioAdapter/{ra}/radioModule/{rm}/modelInfoRM/imeisv
                :
radioAdapter/{ra}/radioModule/{rm}/configurationCapabilities/settings/4G5G/modeOfOperationForEPS
radioAdapter/{ra}/radioModule/{rm}/configurationCapabilities/settings/4G5G/usageSettingForEPSand5GS
                :
radioAdapter/{ra}/radioModule/{rm}/subscriberIdentityModuleInformation/status
radioAdapter/{ra}/radioModule/{rm}/subscriberIdentityModuleInformation/IMSI
radioAdapter/{ra}/radioModule/{rm}/subscriberIdentityModuleInformation/ICCID
                :
radioAdapter/{ra}/radioModule/{rm}/servingNetworkInformation/networkCoverageStatus
radioAdapter/{ra}/radioModule/{rm}/servingNetworkInformation/servingNetworkPLMN
radioAdapter/{ra}/radioModule/{rm}/servingNetworkInformation/servingRadioAccessTechnology
                :
radioAdapter/{ra}/radioModule/{rm}/dataConnectivityInformation/packetSwitchedAttachStatus
radioAdapter/{ra}/radioModule/{rm}/dataConnectivityInformation/initialContextActivationRequired
radioAdapter/{ra}/radioModule/{rm}/dataConnectivityInformation/connectivityStatus
radioAdapter/{ra}/radioModule/{rm}/dataConnectivityInformation/PDP/contextIdList
radioAdapter/{ra}/radioModule/{rm}/dataConnectivityInformation/PDP/{ci}/contextState
radioAdapter/{ra}/radioModule/{rm}/dataConnectivityInformation/PDP/{ci}/APN
radioAdapter/{ra}/radioModule/{rm}/dataConnectivityInformation/PDP/{ci}/assignedIPAddress
                :
radioAdapter/{ra}/radioModule/{rm}/dataConnectivityInformation/PDP/{ci}/QoS/NR5G/5QCI
radioAdapter/{ra}/radioModule/{rm}/dataConnectivityInformation/PDP/{ci}/QoS/NR5G/downlinkGFBR
                :
```

Figure 6.21: Example for a topic structure/tree (extract)

6.6.7 Summary

- Proposal to use MQTT v5.0 [i.29] as Management and Control protocol, standardized by OASIS (www.oasis-open.org).
- Among other features, the "Request/Response" has been "formalized" in v5.0.
- MQTT uses TCP/IP, TLS or WebSocket as a transport protocol.
- MQTT Control Messages are defined/generic (e.g. PUBLISH), the parameters are selected out of the "topic structure/tree".
- Same MQTT protocol could be (re)used by OM to manage, monitor and control the Radio Function(s) (and their hosted Radio Modules); this requires the Radio Functions parameter set to be the (mathematical) "set union" of parameters for GW Connectivity and OM.

Further items to be analysed/studied:

- Protocol for the SW update of Radio Function(s) (their hosted Radio Modules and the Adapter) by OM; i.e. a protocol to load/transfer a SW "image" file containing the SW Update for a Radio Function (Adapter) and/or a Radio Module to a Radio Function. It needs to be analysed whether the control and indication of status/progress etc. might be done via MQTT too, or via other dedicated signalling.
- Parameter/info base (parameter set) for a Radio Function needs to be defined/customized.
- Based on parameter set for a Radio Function, a "topic structure/tree" needs to be defined/customized, to be used by MQTT clients and MQTT server(s).

6.7 Proposal F: NETCONF, RESTCONF and YANG (Management and Control protocol)

6.7.1 Introduction

Network Configuration Protocol (NETCONF) (IETF RFC 4741 [i.33], IETF RFC 6241 [i.37]) is a network device management protocol, developed by the IETF to be the successor of SNMP. NETCONF provides a framework for users to add, modify, or delete network device configurations, or query configurations, status, and statistics.

Representational State Transfer Configuration (RESTCONF) protocol (IETF RFC 8040 [i.41]), is a stateless protocol that uses secure HTTP methods to provide CREATE, READ, UPDATE, and DELETE (CRUD) operations on a conceptual datastore containing YANG-defined data, which is compatible with a server that implements NETCONF datastores.

Both NETCONF and RESTCONF use Yet Another Next Generation (YANG) (IETF RFC 6020 [i.36], IETF RFC 7950 [i.40]) as a data modelling language to describe the interaction models between the NETCONF/RESTCONF client and server. YANG models are nowadays available from several Standards Developing Organizations (SDOs) covering any type of technology: optical transport, IP/MPLS networking, wireless and wired access, microwave, radio transmission.

6.7.2 Key characteristics of the protocols

The NETCONF architecture consists of two roles:

- A client manages network devices using NETCONF and provides the following functions:
    - Sends RPC requests to a NETCONF server to query or modify one or more parameter values.
    - Learns the status of a managed device based on the alarms and events sent by the NETCONF server of the managed device.
- A server maintains information about managed devices and responds to the client-initiated requests:
    - When receiving a request from a NETCONF client, the NETCONF server parses the request and sends a reply to the client.
    - If a fault or an event occurs, the NETCONF server reports an alarm or event to the client through the notification mechanism.

NETCONF uses a hierarchical protocol framework, suitable for handling automatic management tasks. The architecture is represented in Figure 6.22.

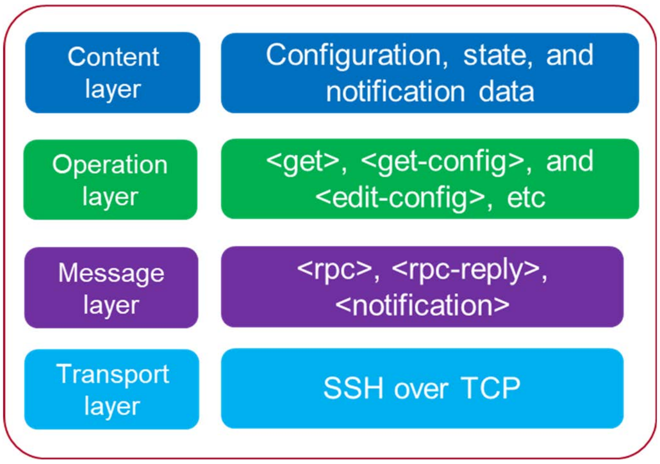

Content layer
Configuration, state, and notification data
Operation layer
<get>, <get-config>, and <edit-config>, etc
Message layer
<rpc>, <rpc-reply>, <notification>
Transport layer
SSH over TCP

Figure 6.22: NETCONF protocol architecture

The content layer contains a set of managed objects (configuration data, status data, and information notified by the server), modelled according to YANG. Extensible Markup Language (XML) data encoding is employed for the protocol messages and base operations, which are defined at the operation layer.

The operations are then invoked at the message layer as RPC methods with XML-encoded parameters. The protocol messages are exchanged on top of a secure transport protocol (SSH), defined at the transport layer.

Multiple operations are defined, expanding the capability previously defined by SNMP: in addition to the basic read/write operations, locking and transaction operations are supported. NETCONF supports extensions based on capability sets.

The prerequisite to establish a NETCONF session is that the SSH connection, authentication, and authorization are complete. Once the session is established, a client and a server immediately exchange with each other Hello messages, containing the set of capabilities supported locally. If both ends support a capability, they can implement special management functions based on the capability itself.

Depending on the operations, NETCONF can lock a specific datastore (set of configuration and operational data) to avoid conflicts. Then, the typical operations based on read, write, delete can take place, with the operation and relevant data encoded in XML. Once the management operations are completed, NETCONF commits the changes and unlocks the datastore. The session may or may not be kept open, depending on the local configuration. The server side can send, if instructed, notifications to keep the client side up-to-date about the state of the operational data.

RESTCONF client and server run HTTPS to establish a secure and connection-oriented session using the datastore concepts defined in the NETCONF. RESTCONF implements a subset of the NETCONF interaction capabilities:

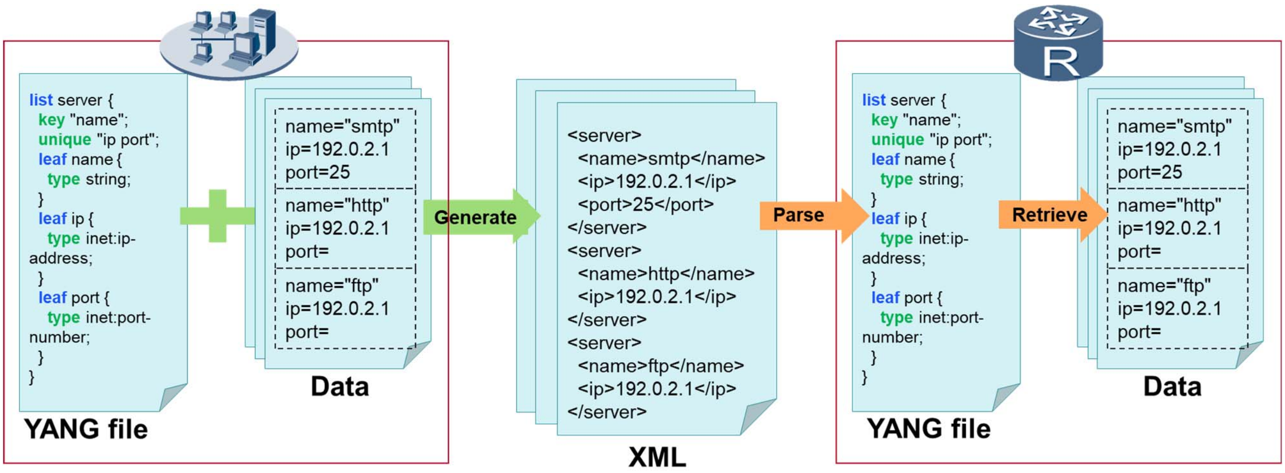
- The RESTCONF client can query the status data and configuration data but the client can only modify configuration data while it cannot modify status data.
- The RESTCONF server maintains managed network devices, responds to client requests, and reports management data to the client.

RESTCONF uses HTTP methods to identify the CRUD (Create, Read, Update, Delete) operations defined in NETCONF for accessing the YANG-defined data.

The data modelling language YANG is used to model both configuration and state data. It is also used to define the format of event notifications emitted by network elements and it allows data modelers to define the signature of RPCs that can be invoked on network elements via the NETCONF (RESTCONF) protocols.

The data modelling language comes with a number of built-in data types. Additional application specific data types can be derived from the built-in data types. More complex reusable data structures can be represented as groupings. YANG data definitions are contained in modules and provide a strong set of features for extensibility and reuse.

The peculiar characteristic of YANG is that it is a modular language representing data structures in an XML tree format. When a NETCONF (RESTCONF) operation is invoked, the application automatically parses the YANG model data and generates the corresponding XML encoding. The server side validates and parses the XML content to retrieve the data and applies the requested operation based on the corresponding YANG model. A simple example is shown in Figure 6.23, whereby a client configures port 25 of the server identified by the IPv4 address 192.0.2.1 for the SMTP service.



```
list server {
  key "name";
  unique "ip port";
  leaf name {
    type string;
  }
  leaf ip {
    type inet:ip-
address;
  }
  leaf port {
    type inet:port-
number;
  }
}
```

```
<server>
  <name>smtp</name>
  <ip>192.0.2.1</ip>
  <port>25</port>
</server>
<server>
  <name>http</name>
  <ip>192.0.2.1</ip>
</server>
<server>
  <name>ftp</name>
  <ip>192.0.2.1</ip>
</server>
```

```
list server {
  key "name";
  unique "ip port";
  leaf name {
    type string;
  }
  leaf ip {
    type inet:ip-
address;
  }
  leaf port {
    type inet:port-
number;
  }
}
```

Figure 6.23: Example of NETCONF (RESTCONF) operation (server configuration)

6.7.3 NETCONF/RESTCONF/YANG in the context of the OB$_{RAD}$ reference point

With reference to Figure 6.6 and Figure 6.15, NETCONF, RESTCONF and YANG all belong to the management and control area, the one highlighted in the red circle. The following analysis assumes to use NETCONF for OB$_{RAD}$, but RESTCONF may alternatively be used instead of NETCONF.

In terms of management tasks, NETCONF and YANG enable the Gateway Function to fully implement and support the commands to:

- Set/Configure the parameter(s) in a Radio Function;
- Get/Enquire parameter(s) from a Radio Function;
- Handle the Notifications (of parameter(s)) from a Radio Function.

Through these operations they handle the set of defined parameters, following the scheme shown in Figure 6.7 and Figure 6.16.

From a functional standpoint, a NETCONF session between the Gateway Function and a Radio Function can be thought as structured in a few steps:

- Step 0: this is the establishment of an SSH session, which is a mandatory prerequisite to enable the NETCONF session. The Gateway Function (client) and the Radio Function (server) reciprocally authenticate by means of their public keys. An SSH public/private key pair has to be configured beforehand.
- Step 1: once the NETCONF session is established, both the Gateway Function and the Radio Function immediately exchange Hello messages with each other to advertise optional capabilities supported locally. If both ends support a capability, they can implement special management functions based on that.
- Step 2: NETCONF operations are exchanged between the Gateway Function and the Radio Function as RPC messages. A NETCONF message consists of three parts (layer):

    1) Message: the message layer provides a simple and independent transmission frame mechanism for RPC messages. The Gateway Function encapsulates an RPC request into an <rpc> element. The Radio Function encapsulates the request processing result in the <rpc-reply> element and responds to the Gateway.

    2) Operations: the operations layer defines the specific NETCONF operation issued by the Gateway Function and the related attributes. The operation is invoked by RPC methods based on XML encoding.

    3) Content: the content layer defines the subset of the data model affected by the specific operation.

- Step 3: the Gateway Function may enable the Notification service through which the Radio Function can send asynchronous event notifications in case of alarms, state change, operational configuration change. The Gateway Function subscribes to the NETCONF notification service to receive alerts on the events it is interested to collect. The Radio Function acknowledges the notification service through an RPC Reply message. From that point on notifications are sent from the Radio Function to the Gateway Function.

- Step 4: the Gateway Function may close the session. In case a notification is sent, the session is opened on demand for the transfer of the requested information.

Figure 6.24 illustrates on a high-level basis the different steps of a NETCONF session.

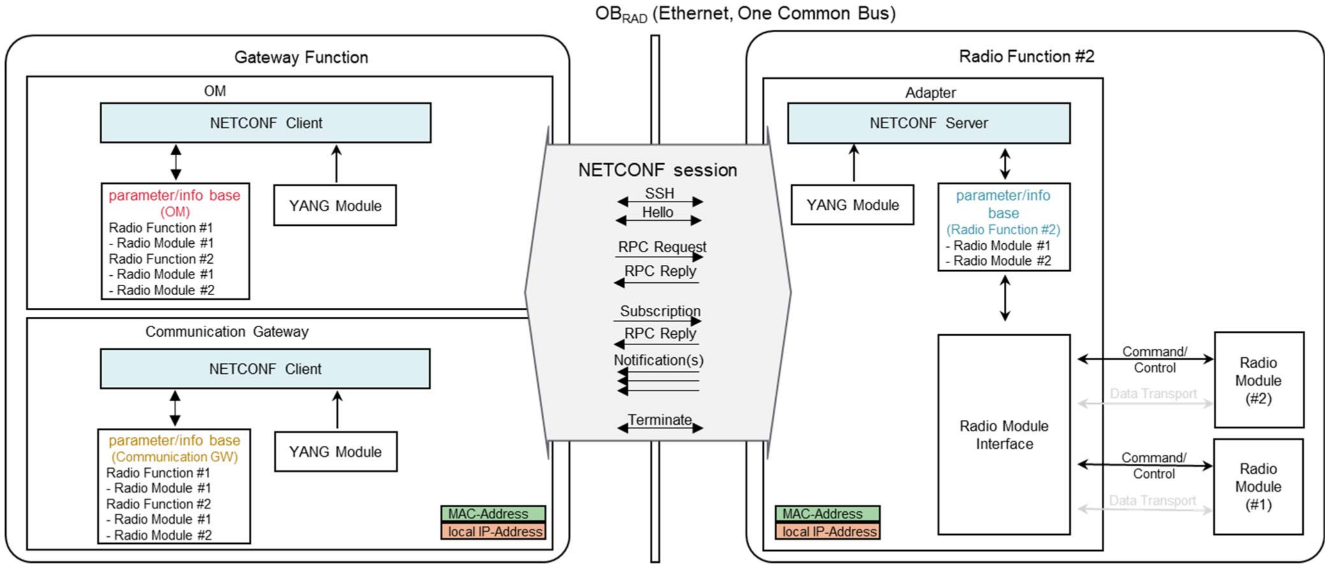


Figure 6.24: Use of NETCONF as OB_{RAD} Management and Control protocol

In Figure 6.24 above, the Operations and Maintenance (OM) function and the Communication Gateway function are represented as decoupled sub-systems. In such a case, both entities act as separated clients to the Radio Function. The mechanisms supported by NETCONF provide the necessary control to prevent simultaneous access to the same set of data (lock/unlock).

The same mechanisms applied to the management of the Radio Function(s) by the Gateway Function can also be enabled when a centralized control entity wants to access the configuration and operational data of a Gateway Function through the OB_{OM} interface as defined in UIC FRMCS TOBA FRS [i.1].

6.7.4 NETCONF/RESTCONF security aspects

NETCONF is connection-oriented, requiring a persistent connection between peers. According to IETF RFC 6241 [i.37], clauses 2.1 and 2.2, this connection is required to provide integrity, confidentiality, peer authentication, and reliable, sequenced data delivery.

This is achieved using Secure Shell (SSH), as described in the base NETCONF specification IETF RFC 4742 [i.34] and IETF RFC 6242 [i.38]. This requires having a private/public key pair on both connection end points. The keys need to be provided/configured before the actual establishment of the NETCONF session. Considering the controlled environment, the keys can be stored locally on both the client (Gateway Function) and the server (Radio Function) without relying on a third-party Certificate Authority.

As an alternative, local passwords can be configured and exchanged instead of the key pair. NETCONF can also rely on Transport Layer Security (TLS) (IETF RFC 5246 [i.35] and IETF RFC 8446 [i.42]), with support of mutual TLS certificate-based authentication. In this case as well the certificates necessary for the mutual authentication can be stored locally on the Gateway Function and the Radio Function.

RESTCONF is defined on top of HTTP and relies on HTTPS.

RESTCONF requires that the transport-layer protocol provide both data integrity and confidentiality. According to IETF RFC 8040 [i.41], clause 2.1, a RESTCONF server is required to support the TLS protocol IETF RFC 5246 [i.35] and IETF RFC 8446 [i.42] and "*should*" adhere to IETF RFC 7525 [i.39] and IETF RFC 9325 [i.43].

According to IETF RFC 8040 [i.41], clause 2.1, it is not allowed to use the RESTCONF protocol over HTTP without using the TLS protocol.

7 Assessment of protocol proposals

7.1 Management and Control protocol proposals

For the comparison and assessment of the Management and Control protocols proposed in clause 6 for OB_{RAD}, the criteria (capabilities and characteristics) as defined in Table 7.1 are used.

Table 7.1: Definition of OB_{RAD} Management and Control protocol capabilities and characteristics

Protocol capabilities/characteristics	Content/Purpose	Comments
Name of the protocol		
Standardization organization	The organization responsible for providing and maintaining the standard (specification) of the protocol (name and link).	
Available implementations:	Protocol implementations (SW) available:	
• operating system	• for which operating system(s)	
• open source	• as open source (yes/no)	
• commercial implementation	• as commercial implementation (yes/no)	
Applicability for OB_{RAD} functions:	Functional areas of OB_{RAD} the protocol is applicable/to be used:	
• Management and Control (Connectivity)	(yes/no)	
• Management, Control and Monitoring (OM)	(yes/no)	
• SW update (OM)	(yes/no)	
Assessment of the state of the art		Would someone use such protocol for new development or is more seen as outdated by the industry?
Synergies and reusability		Synergies and reusability of other OBGW protocol stacks (e.g. OB_{APP})
Complexity for Radio Function	(low/medium/high)	Implementation complexity on the Radio Function
Complexity for Gateway	(low/medium/high)	Implications and complexity of protocol for OBGW
Location of protocol adaption	(Radio Function/Gateway)	Where is the radio module (UE) protocol adaption taking place (OBGW or Radio Function)
Extensibility of protocol		Extensibility of the protocol with future radio module functionality

Protocol capabilities/characteristics	Content/Purpose	Comments
Protocol characteristics:	Characteristics of the protocol proposed:	
• command messages	• generic (i.e. defined by the standard) or to be defined	
• notification messages	• generic (i.e. defined by the standard) or to be defined	
• message content	• what is the content of the messages	
• Bidirectional	(yes/no)	
• Real-Time	(yes/no)	
• Monitoring of the connection	(yes/no)	
• Asynchronous	(yes/no)	
• Cybersecurity protocols		The means of providing some cybersecurity
• customization required for	for which parts of the protocol a customization is required	
• customization to be done via	how the customization is to be done	
Transport protocols supported:	List of transport protocols supported/recommended by the standard.	
Pros (+) and Cons (-) of the proposed solution:	List of Pros (+) and Cons (-)	

The capabilities and characteristics of the protocol proposals analysed and described in clause 6 are listed in Table 7.2.

As a result of finding an alignment/agreement on the different proposals, Proposal F (NETCONF/RESTCONF/YANG) has been selected as the recommended solution.

Table 7.2: Capabilities and characteristics of OB$_{RAD}$ Management and Control protocol proposals

Protocol capabilities/characteristics	Proposal A	Proposal B (note)	Proposal D		Proposal E	Proposal F
Name of the protocol	USB over IP	SNMP (SNMPv3)	HTTP	MQTT	MQTT (MQTT v5.0)	NETCONF/ RESTCONF/ YANG
Standardization organization	USB/IP PROJECT (usbip.sourceforge.net/)	IETF (www.ietf.org)	IETF (www.ietf.org)	OASIS	OASIS (www.oasis-open.org)	IETF (www.ietf.org)
Available implementations:						
• operating system	Linux®	Linux®	All/Any		Linux®	All major OSes support NETCONF/RESTCONF/YANG (e.g. Linux®, Windows)
• open source	yes	yes	yes		yes	yes
• commercial implementation	yes	yes	yes		yes	yes
Applicability for OB$_{RAD}$ functions:						
• Management and Control (Connectivity)	yes	yes	yes		yes	yes
• Management, Control and Monitoring (OM)	yes	yes	yes		yes	yes
• SW update (OM)	yes	partly (control of SW update)	yes		partly (control of SW update)	partly (SW may be or may be not updated depending on the device)
Assessment of the state of the art	Will be outdated when USB will be outdated		Fit for new developments	Fit for new developments	Fit for new developments	Fit for new developments
Synergies and reusability	Could synergies with OB$_{APP}$ for the Management and control API complement		Possible common stack with OB$_{APP}$	MQTT could possibly be used for the OB$_{APP}$ AUXF and thus provide very good consistency and reliability	MQTT could possibly be used for the OB$_{APP}$ AUXF and thus provide very good consistency and reliability	Common stack with OB$_{OM}$. Possible common stack with OB$_{APP}$
Complexity for Radio Function	Low (Easy server software to install)		Medium (hosts web server, MQTT client, and adapter)		Medium (MQTT client)	Low to Medium (Support of NETCONF/RESTCONF server)
Complexity for Gateway	Low under Linux®		Low (HTTP client already exists due to OB$_{APP}$, MQTT broker and client to be integrated)		Low (MQTT broker and client)	Low (Typically, already available in any Datacom implementation)
Location of protocol adaption	Radio Function and Gateway		Radio Function		Radio Function	Possibly the Radio Function

Protocol capabilities/characteristics	Proposal A	Proposal B (note)	Proposal D		Proposal E	Proposal F
Extensibility of protocol	Fully compliant with modem updates		Very high extensibility		Very high extensibility	Very high (YANG modules can be easily extended to introduce new capabilities)
Protocol characteristics:						
• command messages	USB Requests	generic: GetRequest, SetRequest	HTTP GET w/ resource	-	generic: PUBLISH	generic (e.g. edit, edit-config, CRUD, etc.)
• notification messages	USB Requests	generic: Trap, InformRequest	HTTP SSE if HTTP2 is used	generic: PUBLISH	generic: PUBLISH	generic (i.e. notifications)
• message content	AT commands, QMI, MBIM or debug interface	list of Object ID/value pairs	to be defined (XML or JSON)	list of topic/value pairs	list of topic/value pairs	Encoded as XML/JSON and embedded in RPC calls (NETCONF) or HTTP methods (RESTCONF)
• Bidirectional	yes		Yes (one protocol for each direction or HTTP2)		yes	Yes (RPC and RPC-Reply)
• Real-Time	yes		Yes		yes	Yes
• Monitoring of the connection	yes		Yes (at TCP level or HTTP and keep alive for MQTT)		yes	Yes (TCP, SSH, TLS)
• Asynchronous	yes		Yes for MQTT, HTTP can hold some kind of asynchronicity (esp. if v2 is used)		yes	Yes (notifications)
• Cybersecurity protocols	TLS		TLS (at least)		TLS	SSH, TLS, HTTPS (Native support of PKI or certificate exchange or passwords)
• customization required for	physical electronic signals: modem switch on/off, LEDs, modem reset, thermal aspect, etc.	parameter/info base	Define the commands from ETSI TS 127 007 [i.25]	List of status	parameter/info base	Possibly the YANG data models (Once defined, a model has a very high reusability)
• customization to be done via	WebSocket/JSONRPC API, SNMP, MQTT, etc.	MIB (Management Information Base)	XML/JSON	topic structure/tree	topic structure/tree	YANG tree/parameters of the data model
Transport protocols supported/suitable:	TCP/IP, UDP/IP	UDP (recommended/preferred), OSI, DDP, IPX, IEEE 802, TCP/IP ("experimental" RFC)	HTTP/TCP/IP	see MQTT (Proposal E)	TCP/IP, TLS, WebSocket	TCP/IP, Either SSH or TLS, HTTPS

Protocol capabilities/characteristics	Proposal A	Proposal B (note)	Proposal D		Proposal E	Proposal F
Pros (+) and Cons (-) of the proposed solution:	+ modem is detected as inside the GW by the GW, + low added latency, + low CPU usage (well split between CPU cores), + limited overhead, + USB 3 maximum transmission speed (5Gbit/s), - no decoupling of GW (Connectivity and OM) from (proprietary) Radio Module interfaces (GW needs to adapt to the specific Radio Modules)	+ decoupling of GW (Connectivity and OM) from (proprietary) Radio Module interfaces, + well defined/standardized, generic messages, + only the parameter list needs to be customized/defined, - UDP may not be robust enough to be used on trains (?), - support of TCP in SNMP implementations unclear (?)	+ decoupling of GW (Connectivity and OM) from (proprietary) Radio Module interfaces, - HTTP API (control commands) needs to be defined as for any other solution, + well defined HTTP methods (GET/PUT/POST, etc.), + full control of SW possible via HTTP API, - all the (pros) of Proposal E with the drawback of necessitating more resources because of the two protocols used	see MQTT (Proposal E)	+ decoupling of GW (Connectivity and OM) from (proprietary) Radio Module interfaces, + well defined/standardized, generic messages, - list of topics needs be defined	+ decoupling of GW (Connectivity and OM) from (proprietary) Radio Module interfaces, + for NETCONF, standardized RPC messages. For RESTCONF, standardized HTTP methods (GET/PUT/POST, etc.), + high reusability of the YANG data models, + only the parameters of the data model need to be customized/defined, + security embedded (SSH, TLS, HTTPS), + mainstream in the Datacom industry, - may have limitations in constrained implementations (IETF is working on a lightweight version of NETCONF/RESTCONF)
NOTE: The proposal B (SNMP) was no longer considered to be recommended for OB_{RAD}.						

7.2 Data Transport protocol proposals

7.2.1 Encapsulation protocol proposals

The following encapsulation protocols have been proposed as Data Transport protocols for $\mathrm{OB_{RAD}}$:

a) IP-in-IP encapsulation (IETF RFC 2003 [i.11], analysed within Proposal B in clause 6.3.2);

b) GTP-U (ETSI TS 129 281 [i.24], analysed within Proposal C in clause 6.4).

Figure 7.1 shows the encapsulation protocols and their additional required header sizes. The use of GRE (IETF RFC 2784 [i.30], IETF RFC 2890 [i.31], IETF RFC 8086 [i.32]) is not further analysed/described in the present document; it is shown here as it has been mentioned in one of the meeting discussions.

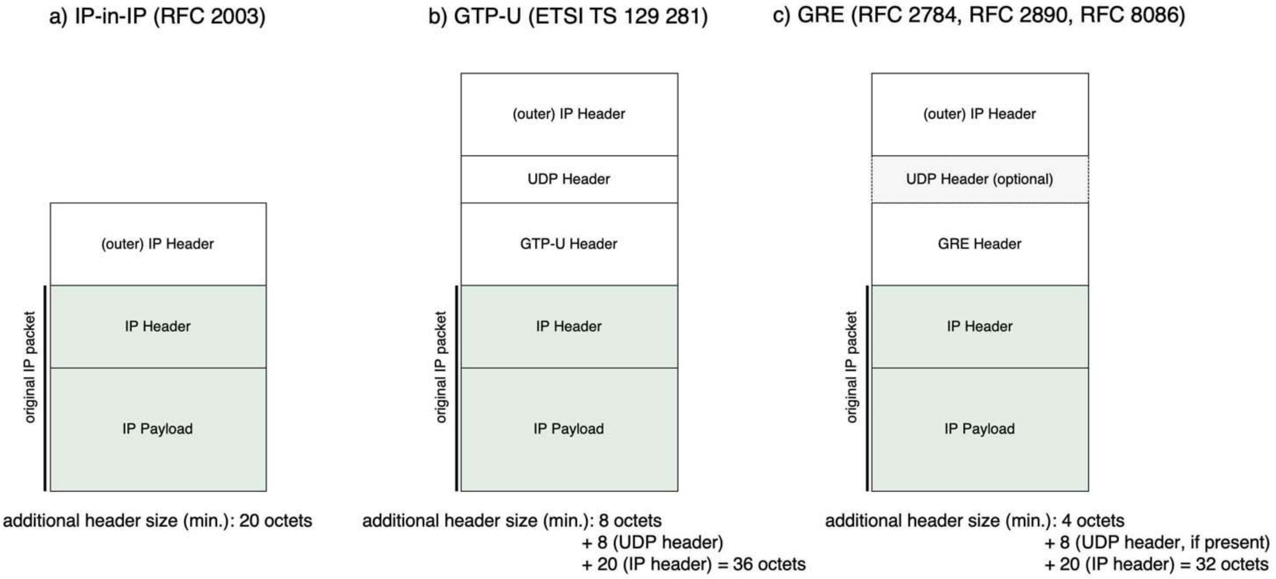

a) IP-in-IP (RFC 2003)
b) GTP-U (ETSI TS 129 281)
c) GRE (RFC 2784, RFC 2890, RFC 8086)
(outer) IP Header
IP Header
IP Payload
original IP packet
UDP Header
GTP-U Header
UDP Header (optional)
GRE Header
additional header size (min.): 20 octets
additional header size (min.): 8 octets + 8 (UDP header) + 20 (IP header) = 36 octets
additional header size (min.): 4 octets + 8 (UDP header, if present) + 20 (IP header) = 32 octets

Figure 7.1: Encapsulation protocols and additional header sizes

7.2.2 IP-in-IP encapsulation

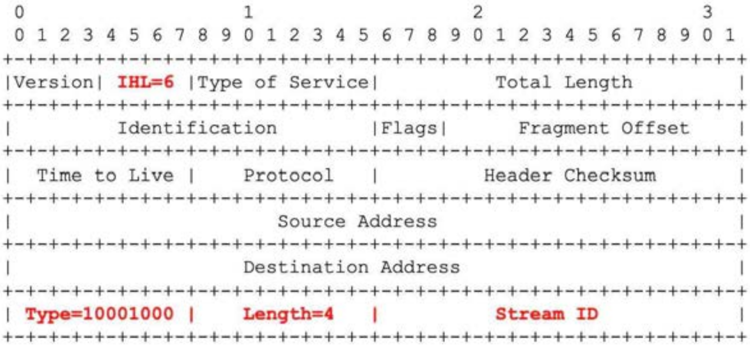
The comparison of the encapsulation protocols in Figure 7.1 shows that the "IP-in-IP encapsulation" seems to be the most effective approach in terms of additional header size (min. 20 octets) and number of encapsulation steps (one).

For the $\mathrm{OB_{RAD}}$ Data Transport protocol, it is recommended to use "IP-in-IP encapsulation".

When using IP-in-IP encapsulation, two addressing schemes for addressing the Communication Sessions are possible:

- by local IP-Address (one local IP-Address per Communication Session); or
- by optional "Stream Identifier" (one local IP-Address per Radio Function and one "(internal) Communication Session ID" per Communication Session, as analysed and described in clause 6.3.2).

Figure 7.2 shows the (outer) IP Header when the "Stream Identifier" of the "Options" filed is used to carry the "(internal) Communication Session ID".



```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Version| IHL=6 |Type of Service|          Total Length         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Identification        |Flags|      Fragment Offset    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Time to Live |    Protocol   |         Header Checksum       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Source Address                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Destination Address                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Type=10001000 |    Length=4   |            Stream ID          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 7.2: (Outer) IP Header with "Stream Identifier"

To allow a flexible use of the IP-in-IP encapsulation, it might be desired to support both addressing schemes by OB_{RAD} and offering a configuration option which one is to be used.

7.2.3 When exceeding the "standard" Ethernet payload size

Independently of the chosen encapsulation protocol, every additional encapsulation may lead to the situation where the payload size of the resulting, encapsulated IP packet exceeds the maximum "standard" Ethernet payload size of 1 500 octets.

To avoid a further reduction of the MTU size of the User Plane and Control Plane (media/data payload and signalling), the following mechanisms may be used:

- IP fragmentation; or
- "baby giant" (Ethernet) frames.

When allowing IP fragmentation, if the size of encapsulated IP-packet were to exceed 1 500 octet Ethernet payload size, the encapsulated IP-packet will be fragmented (on IP protocol level). This is a "standard" IP method (IETF RFC 791 [i.12], clause 2.3 and clause 3.2). IP fragmentation is used only for transferring the IP packets via the OB_{RAD} Ethernet network; they are reassembled immediately after reception and are never leaving the On-Board FRMCS.

"Baby giant" Ethernet frames are only slightly larger than a "standard" Ethernet frame. Frame sizes from 1 530 up to 2 000 octets might be supported (equipment specific). When using "baby giant" Ethernet frames, an IP fragmentation is not required in addition.

Because the support of "baby giant" frames cannot be guaranteed by all used Ethernet equipment, it might be desired to support both methods and offering a configuration option which one is to be used:

- use IP fragmentation (and "baby giants" Ethernet frames are never needed/used); or
- use "baby giants" Ethernet frames (and never use IP fragmentation).

8 Other items in scope

8.1 Migration aspects

Migration aspects (existing versus new installations) dealing with migrating from GSM-R to FRMCS are currently not in scope of UIC specifications and can therefore not be in scope and cannot be studied in the present document, since OB_{RAD} is an internal FRMCS interface and does not directly interact with GSM-R.

8.2 Regulatory considerations

There may be an effect on ETSI deliverables due to M/603 standardisation request. This standardization request ended up to Commission Implementing Decision C(2024)2466 [i.45], which is the (full title of document) *"Commission Implementing Decision on a standardization request to the European Telecommunications Standards Institute as regards the definition of system specification requirements for the Future Railway Mobile Communication System in support of Directive (EU) 2016/797 of the European Parliament and of the Council"*.

According to Commission Implementing Decision C(2024)2466 [i.45], clause (11), *"The standardisation deliverables drafted by ETSI should include detailed technical specifications of the essential requirements in accordance with Directive (EU) 2016/797. They should describe the technical solutions to ensure that* ***essential requirements*** *concerning technical compatibility are fulfilled and cover the requirements related to the* ***relevant basic parameters*** *described in the CCS TSI. They should also be based on risk assessment and risk reduction methodologies and reflect the generally acknowledged state of the art"*.

In detail, the technical compatibility is pointed out in Directive (EU) 2016/797 [i.46], clause 1.5 of Annex III, *"The technical characteristics of the infrastructure and fixed installations have to have compatibility with each other and with those of the trains in the rail system. This is a requirement for the safe integration of the subsystem of the vehicle with the infrastructure. If compliance with these characteristics proves difficult on certain sections of the network, temporary solutions, which ensure compatibility in the future, may be implemented"*.

As such the Directive (EU) 2016/797 [i.46] points to the CCS TSI [i.47]. In Article 2, clause 1 of the CCS TSI [i.47] is stated *"The TSI applies to new trackside CCS and on-board CCS subsystems of the rail system as defined in clauses 2.3 and 2.4 of Annex II to Directive (EU) 2016/797"*.

The **basic parameters** mentioned in Directive (EU) 2016/797 [i.46], Article 5, clause 2 (a), clause 2 (b) and clause 3, are referenced in CCS TSI [i.47], clause 4.1.1 of Annex I, which then point to the Tables A 1 (References between basic parameters and mandatory specifications), A 2 (List of mandatory specifications) and A 3 (List of standards), within CCS TSI [i.47], Appendix A, which ensures that **essential requirements** concerning technical compatibility.

For all Rail equipment regarding usage within CCS TSI [i.47], clause 4.1.1 of Annex I, these relevant basic parameters have to be fulfilled accordingly certain classifications regarding its purpose, usage and functional definition. This could be considered also for OB_{RAD} equipment and has to be kept in mind, when designing hardware for mobile equipment on Rolling Stock and of course implementation on train like OB_{RAD} is going to be.

9 Conclusion

The present document has analysed the requirements for the OB_{RAD} interface captured in relevant UIC specifications. Several available protocols (see Annex A), potentially serving as OB_{RAD} Management and Control protocol or OB_{RAD} Data Transport protocol have been analysed in detail and its usage within the technical realizations for OB_{RAD} has been described (see clause 6).

As a result of the study, it is recommended:

- For the OB_{RAD} Management and Control protocol to use NETCONF/RESTCONF/YANG;
- For the OB_{RAD} Data Transport protocol to use "IP-in-IP encapsulation".

Annex A: Investigation of available protocols ("possible candidates")

A.1 Introduction

This annex contains the list of possible "candidates" for OB$_{RAD}$ protocols (as Management and Control protocol as well as for Data Transport protocol), which has been used to select the one or other protocol to be further analysed/studied.

The list is not expected to be complete.

The protocols are listed in alphabetical order.

A.2 Management and Control protocol

- Data Distribution Service (DDS)
- Mobile Broadband Interface Model (MBIM)
- Modem Manager
- MQTT (see also proposal D in clause 6.5 and proposal E in clause 6.6)
- NETCONF/RESTCONF/YANG (see also proposal F in clause 6.7)
- Qualcomm MSM Interface (QMI)
- Representational State Transfer (REST) software architectural style
- REST, RESTful API
- Simple Network Management Protocol (SNMP) (see also proposal B in clause 6.3.3)
- Simple Object Access Protocol (SOAP)
- USB over IP, USB/IP (see also proposal A in clause 6.2)

A.3 Data Transport protocol

- GPRS Tunnelling Protocol, GTP-U (see also proposal C in clause 6.4.2.3)
- IP-in-IP encapsulation (see also proposal B in clause 6.3.2)
- Packet Forwarding Control Protocol (PFCP)

History

Document history		
V1.1.1	January 2025	Publication