Ref. Ares(2022)3331856 - 29/04/2022

Grant Agreement Number: 101014517
Project Acronym: AB4Rail
Project title: Alternative Bearers for Rail

DELIVERABLE D [3.4]

[Identification of transport protocol for railway applications]

Project acronym:	AB4Rail
Starting date:	01-01-2021
Duration (in months):	24
Call (part) identifier:	S2R-OC-IP2-02-2020
Grant agreement no:	Number 101014517 – IP/ITD/CCA - IP2
Grant Amendments:	N/A
Due date of deliverable:	31-01-2022
Actual submission date:	29-04-2022
Coordinator:	Franco Mazzenga (Radiolabs)
Lead Beneficiary:	Romeo Giuliano (USGM)
Version:	0.1
Type:	Report
Sensitivity or Dissemination level[1]:	PU
Contribution to S2R TDs or WAs[2]	TD2.1
Taxonomy/keywords:	Adaptable Communication System; ACS; IP emulator; IP impairment models; lightweight virtualization; transport protocols; application protocols;

This project has received funding from the Shift2Rail Joint Undertaking (JU) (now Europe's Rail Joint Undertaking, EU-RAIL) under grant agreement No. 101014517. The JU receives support from the European Union's Horizon 2020 research and innovation programme and the Shift2Rail JU members other than the Union.

[1] PU: Public; CO: Confidential, only for members of the consortium (including Commission Services)

[2] https://projects.shift2rail.org/s2r_matrixtd.aspx

Authors Table

Name	Affiliation	Contribution
Romeo Giuliano	Università degli Studi Guglielmo Marconi (USGM)	Main contributor
Francesco Vatalaro	University of Rome Tor Vergata	Main contributor
Franco Mazzenga	Radiolabs (RDL)	Main contributor
Alessandro Vizzarri	Radiolabs (RDL)	Support to contributors

The document history table provides a summary of all the changes in reverse chronological order (latest version first).

Document history

Date	Name	Affiliation	Position/Project Role	Action/ Short Description
31 Jan. 2022	Romeo Giuliano	Università degli Studi Guglielmo Marconi (USGM)	WP leader	Identification of the transport protocols
29 April 2022	Romeo Giuliano	Università degli Studi Guglielmo Marconi (USGM)	WP leader	The updated document includes all the revisions provided by PO

Disclaimer

The information in this document is provided "as is", and no guarantee or warranty is given that the information is fit for any particular purpose. The content of this document reflects only the author's view – the Shift2Rail Joint Undertaking (now Europe's Rail Joint Undertaking, EU-RAIL) is not responsible for any use that may be made of the information it contains. The users use the information at their sole risk and liability.

Table of Contents

Table of Contents 3
Executive Summary 5
List of abbreviations, acronyms, and definitions 7
List of Figures 10
List of Tables 12
1. Introduction 13
1.1 Purpose and scope of the document 13
1.2 Document organization 13
1.3 Reference Documents 14
2. High level organization of activities for transport protocol evaluation 15
2.1 Notes 17
3. Description of the emulator 18
3.1 Architecture and Setup 18
3.2 Generation of IP impairments 20
3.3 Evaluation of available transmission capacity for the reference train in different scenarios 20
4. Description of the radio environment 29
4.1 Positioning of LTE cells along the railway line 29
4.2 Evaluation of the number of trains in the same cell at the same time 31
4.3 Calculation of train transmission capacity 33
4.4 LTE/5G latency and packet loss 34
4.4.1 On packet loss on the radio access network 35
4.5 Positioning of Wi-Fi stations for urban-metro 36
4.6 Example of generated file for changing values of IP impairments during emulation 36
5. Emulation test plan and performance parameters 39
5.1 Introduction 39
5.2 Traffic sources and their characteristics 40
5.3 Performance metrics 42
6. Performance Results 43
6.1 Introductory remarks 43
6.2 Constant Bit Rate 45
6.2.1 High-Speed rail line 45
6.2.2 Regional rail line 48

6.2.3 Metropolitan line 50
6.2.4 Average Throughput 51
6.2.5 Larger bandwidths for High-Speed line 52
6.3 File Transfer 54
6.3.1 High-Speed rail line 54
6.3.2 Regional rail line 56
6.3.3 Metro rail line 58
6.4 Variable Bit Rate 59
6.4.1 High-Speed rail line 59
6.4.2 Regional rail line 62
6.4.3 Metropolitan rail line 64
6.4.4 Average Throughput 67
6.5 Short conclusions on TCP-Cubic, TCP-BBR and SCTP 68
6.6 Coexistence of TCP Cubic, TCP BBR and SCTP 69
6.6.1 High-Speed rail line 69
6.6.1.1 Results on TCP BBR and TCP cubic 69
6.6.1.2 Results on TCP BBR and SCTP 70
7. Conclusions 72
8. References 75
Appendix 77

Executive Summary

The task 3.4 of AB4Rail project is dedicated to the evaluation of the communication transport protocols for the selected traffic sources in different rail scenarios including mainline, regional/freight and metro/urban.

Evaluation at transport level requires the definition of the models of traffic sources to be used in performance evaluation. In general, these models do not refer to any particular application protocol but are based on simple classification distinguishing between Constant Bit Rate (CBR) and Variable Bit Rate (VBR) traffic sources. These two generic models account for the characteristics of traffic generated by any application i.e., video streaming, file transfer, applications typically generate CBR traffic as seen at transport level while messaging, web-based etc. applications generate variable rate traffic.

The evaluation of transport protocols considered in this work consists in the analysis of the Transport Control Protocol (TCP) in the two versions cubic and bottleneck bandwidth and round-trip propagation time (BBR), the User Datagram Protocol (UDP) and Stream Control Transmission Protocol (SCTP).

To assess transport protocol performances, we have used the software emulator developed in the Task 3.3, [1]. This emulator can reproduce the behavior of one or more communication bearers as seen at IP protocol level and allows to use the transport protocol stacks implemented in the operating system Kernel (i.e., Linux OS in our case) to emulate the data transport between source and the destination. Furthermore, the emulator allows to model the variations with time of the typical packet impairments characterizing the IP layer link such as: bandwidth, latency and packet loss rate. In order to emulate the realistic time variation of packet impairments at IP level we have first identified three realistic rail scenarios and we have assumed the entire rail line is covered by LTE radio technology. The three railway lines we have considered are:

- The Roma – Firenze high-speed line, to evaluate the generic mainline environment
- The Roma – Firenze regional line, to evaluate the regional/freight line type
- The metro of Rome, to evaluate the metro/urban line

To account for the variations with time of the available transmission capacity along the track in our emulation we have also considered the variations of the available LTE modulation coding scheme (MCS) that is selected by the on-board terminal in accordance with its distance from the eNB.
Transport protocol performance have been evaluated in terms of the statistics (i.e., cumulative distribution function, mean, standard deviation etc.) of:

1. One-way transmission latency i.e., the time required from one packet enqueued in the transmission buffer to reach the receiver;
2. The receiver data rate which is referred as the receiver throughput or more simply throughput;
3. The download time for CBR traffic sources generating a finite number of packets.

In the first phase, performance evaluation has started by considering each transport protocol separately i.e., only one active transport protocol on the link for the three different rail scenarios listed above.
In the case of small or zero packet loss both protocols TCP cubic and TCP BBR are able to track the available transmission channel capacity in all the considered scenarios, showing in practice the same behavior, at the expense of increased packet latency.
In case of lossy channel, results show that in every scenario the TCP BBR is able to track the available transmission channel capacity, since the CDF of the TH at PL=1% is very similar to that obtained at PL=0%. As expected, this behavior is practically independent of the considered rail scenario even though TH can be slightly higher in the mainline due to reduced percentage of time the train remains in the area characterized by MCS with reduced modulation efficiency. But this result does not depend on the features of the considered transport protocol.
Results on Linux OS SCTP implementation (single stream) show that this protocol tries to save latency, due to the adopted congestion control algorithm. This may render SCTP particularly interesting for signaling services characterized by low data rates but requirements on latency.
It is worth noting that the behavior of the three protocols is almost invariant passing from CBR to VBR, with the considered high bit rates traffic sources.

In a second phase we have analyzed the coexistence between pairs of transport protocols sharing the same transmission tunnel. Performance comparison in this important scenario has been carried out by comparing the latency and the achievable throughput of each one of the two protocols. In case of coexistence of TCP BBR and TCP cubic, both perform similarly in terms of average download time and experienced throughput for lossless channels, while TCP BBR acquires more bandwidth in lossy channels. In case of TCP BBR and SCTP, TCP BBR outperforms SCTP both for lossless and lossy channels.

The results presented in this Deliverable will be used for the activities of task 3.5 concerning the analysis of: QUIC protocol, the application and transport protocols and secure protocols.

List of abbreviations, acronyms, and definitions

Acronym	Definition
3G	3rd Generation
3GPP	Third Generation Partnership Project
4G	4th Generation
5G	5th Generation
AB	Alternative Bearer
ACS	Adaptable Communication System
AI	Artificial Intelligent
API	Application Programming Interface
BS	Base Station
CDF	Cumulative Distribution Function
CTA	Communication Traffic Analysis
DS	downstream
EIA	Electronic Industries Alliance
eNB	evolved Node B
GEO	Geographical Earth Orbit
GRE	Generic Routing Encapsulation
GW	Gateway
HAPS	High Altitude Platform Station
ICT	Information and Communications Technology
KPI	Key Performance Indicators
IP	Internet Protocol
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
LAN	Local Area Network
LEO	Low Earth Orbit
MA	Movement Authority
MCS	Modulation and Coding Scheme

MEO	Medium Earth Orbit
MSS	Maximum Segment Size
MTU	Maximum Transmission Unit
NA	Network Application
NAT	Network Address Translation
NG	Network Gateway
NIC	Network Interface Card
NSA	Non-Stand Alone
OA	On Board Application
OFDM	Orthogonal Frequency-Division Multiplexing
OG	On-Board Gateway
OS	Operating System
PDCP	Packet Data Convergence Protocol
P-GW	Packet Gateway
PL	packet loss probability
PLMN	Public Land Mobile Network
PMTUD	Path MTU Discovery
PR	Position Report
PRB	Physical Resource Block
PSTN	Public Switched Telephone Network
QoS	Quality of Service
RAN	Radio Access Network
RB	Resource Block
RLC	Radio Link Control
RRC	Radio Resource Control
RTT	Round Trip Time
SCTP	Stream Control Transmission Protocol
SINR	Signal-to-Interference plus Noise Ratio
SIP	Session Initiation Protocol
SNR	Signal-to-Noise Ratio
SLA	Service Level Agreement

TBF	token bucket First-In First-Out
TDMA	Time Division Multiple Access
TIA	Telecommunications Industry Association
TCP	Transmission Control Protocol
VoIP	Voice over IP
UDP	User Datagram Protocol
UE	User Equipment
US	upstream
VM	Virtual Machine
Wi-Fi	Wireless Fidelity

List of Figures

Figure 1: High-level view of the emulator setup for the transport protocol analysis. 15
Figure 2: Architecture of the Linux OS emulator used to assess transport protocol performance. 18
Figure 3: Stops and stations: a. high-speed; b. regional 22
Figure 4: considered velocity vs time train profile for mainline scenario. 25
Figure 5: considered velocity vs time train profile for Regional/Freight scenario. 25
Figure 6: considered velocity vs time train profile for Metro/Urban scenario. 26
Figure 7. Distance profile as a function of the time for a train departing from Termini or from Firenze stations: mainline case 27
Figure 8. Distance profile as a function of the time for a train departing from Termini or from Firenze: Regional/freight line 27
Figure 9. Distance profile as a function of the time for a train departing from one starting station of the line: Urban/metro 28
Figure 10. Positioning of eNBs along the railway line. 29
Figure 11. Number of additional trains in the same LTE cell with time for the mainline. 31
Figure 12. Number of additional trains in the same LTE cell with time for the regional/freight line. 32
Figure 13. Number of additional trains in the cell – inter-departure time interval D=5 min. 33
Figure 14. Command example for generating IP impairments during emulation 37
Figure 15. Packet traces of transport protocols (example). 43
Figure 16. Packet traces for UDP protocol: three examples. 44
Figure 17. CDF for TCP cubic in terms of latency (a) and throughput (b) for High-Speed train line 46
Figure 18. CDF for BBR in terms of latency (a) and throughput (b) for High-Speed train line 47
Figure 19. CDF for TCP cubic, BBR, UDP and SCTP in terms of latency (a) (c) and throughput (b) (d) and for different PL values for High-Speed train line. 48
Figure 20. CDF for TCP cubic in terms of latency (a) and throughput (b) for Regional line 49
Figure 21. CDF for BBR in terms of latency (a) and throughput (b) for Regional line 49
Figure 22. CDF for TCP cubic in terms of latency (a) and throughput (b) for Metropolitan line 50
Figure 23. CDF for BBR in terms of latency (a) and throughput (b) for Metropolitan line 51
Figure 24. Average Throughput vs channel latency 51
Figure 25. CDF of throughput for TCP cubic (a) and BBR (b) for N_c = 300 (5 MHz) for High-Speed line 53
Figure 26. CDF of throughput for TCP cubic (a) and BBR (b) for N_c = 1200 (20 MHz) for High-Speed line 53
Figure 27. Average Throughput vs channel latency for larger LTE bandwidths for High-Speed line: (a) N_c = 300; (b) N_c = 1200. 54
Figure 28. Average Download Time for TCP cubic, BBR and SCTP vs channel latency for High-Speed train line. File dimensions: (a) 1 MB and (b) 10 MB. 55
Figure 29. CDF of Download Time for TCP cubic, BBR and SCTP for High-Speed train line (channel latency = 150 ms). File dimensions: (a) 1 MB and (b) 10 MB 56
Figure 30. Average Download Time for TCP cubic, BBR and SCTP vs channel latency for Regional train line. File dimensions: (a) 1 MB and (b) 10 MB. 57
Figure 31. CDF of Download Time for TCP cubic, BBR and SCTP for Regional line (channel latency = 150 ms). File dimensions: (a) 1 MB and (b) 10 MB 57

Figure 32. Average Download Time for TCP cubic, BBR and SCTP vs channel latency for Metropolitan train line. File dimensions: (a) 1 MB and (b) 10 MB. 58
Figure 33. CDF of Download Time for TCP cubic, BBR and SCTP for Metropolitan train line (channel latency = 150 ms). File dimensions: (a) 1 MB and (b) 10 MB 59
Figure 34. CDF for TCP cubic in terms of latency (a), (c) and throughput (b) (d) at Low-Rate traffic and High-Rate traffic for High-Speed train line 60
Figure 35. CDF for BBR in terms of latency (a), (c) and throughput (b) (d) at Low-Rate traffic and High-Rate traffic for High-Speed train line 61
Figure 36. CDF for TCP cubic in terms of latency (a), (c) and throughput (b) (d) at Low-Rate traffic and High-Rate traffic for Regional train line 62
Figure 37. CDF for BBR in terms of latency (a), (c) and throughput (b) (d) at Low-Rate traffic and High-Rate traffic for Regional train line 63
Figure 38. CDF for TCP cubic in terms of latency (a), (c) and throughput (b) (d) at Low-Rate traffic and High-Rate traffic for Metropolitan train line 65
Figure 39. CDF for BBR in terms of latency (a), (c) and throughput (b) (d) at Low-Rate traffic and High-Rate traffic for Metropolitan train line 66
Figure 40. Average Throughput vs channel latency for VBR 67
Figure 41. Coexistence between BBR (solid line) and TCP cubic (dashed) vs channel latency for High-Speed train line: (a) Average Download Time; (b) Average Throughput. 69
Figure 42. Coexistence between BBR (solid line) and SCTP (dashed) vs channel latency for High-Speed train line: (a) Average Download Time; (b) Average Throughput. 70
Figure 43. example of transmitted packets of TCP cubic and TCP BBR during their coexistence. 71

List of Tables

Table 1: Reference Documents. 14
Table 2. Table maximum speeds that can be reached. 21
Table 3. Distances in the Metro A in Rome. 24
Table 4. MCS for the LTE and associated coverage distances (in km) from the eNB – Urban and sub-urban area types. 30
Table 5. Inter-distance between two consecutive eNBs for different types of areas – handover threshold of -115 dBm. 31
Table 6. Performance values for a good network [5] 35
Table 7. Achievable bit rate data for Wi-Fi IEEE 802.11g and its corresponding coverage in outdoor environment. 36
Table 8: Examples for evaluating the overflow occurrence for R_{source} = 5.2 Mbit/s 45
Table 9: Summary of transport protocols for ACS application classes 73

1. Introduction

This document constitutes the Deliverable D3.4 “Identification of transport protocol for railway applications” according to Shift2Rail Joint Undertaking (now Europe's Rail Joint Undertaking, EU-RAIL) programme of the project titled “Alternative Bearer for Rail” (Project Acronym: AB4Rail, Grant Agreement No 101014517 — IP/ITD/CCA — IP2). On 22nd July 2020, the European Commission awarded a grant to the AB4Rail consortium of the Shift2Rail / Horizon 2020 call (S2R-OC-IP2-02-2020). AB4Rail is a project connected to the development of a new Communication System planned within the Technical Demonstrator TD2.1 of the 2nd Innovation Programme (IP2) of Shift2Rail JU: Advanced Traffic Management & Control Systems.

The IP2 “Advanced Traffic Management & Control Systems” is one of the five asset-specific Innovation Programmes (IPs), covering all the different structural (technical) and functional (process) sub-systems related to control, command, and communication of railway systems.

1.1 **Purpose and scope of the document**

The aim of this document is to investigate the behavior of the communication transport protocols on realistic railway scenarios. The models of traffic sources that have been used in performance evaluation have been classified as Constant Bit Rate (CBR) and Variable Bit Rate (VBR). These two generic models account for the characteristics of traffic generated by any application such as those indicated in Table 7 in [3] i.e., applications for video, file transfer (file download/upload), typically generate CBR traffic as seen at transport level while voice (critical or not), messaging, web-based etc. applications generate variable rate traffic and can be categorized as VBR.

The transport protocols, we have considered in this analysis are:

1. the Transport Control Protocol (TCP) in its versions including Cubic and bottleneck bandwidth and round-trip propagation time (BBR) congestion control algorithms;
2. the User Datagram Protocol (UDP);
3. the Stream Control Transmission Protocol (SCTP).

The analysis of transport protocol performance considering the specific characteristics of the railway application belonging to a specific ACS traffic class will be carried out in the next Task 3.5.

1.2 **Document organization**

The document is organized according to AB4Rail Grant Agreement Number 101014517 (RD-1) and AB4Rail Consortium Agreement (RD-2). The document structure is the following.

In Section 2, we introduce the organization of the activities for the transport protocol evaluation at high-level.

In Section 3, we describe the emulator developed in the Task 3.3 and used for the transport protocols assessment.

In Section4, it is detailed the modeling of the radio environment and the IP impairments in the case the LTE radio technology is used for covering the mainline and the regional/freight rail tracks and Wi-Fi for covering the urban/metro environment. The considered model allows to account for the MCS and for the number of trains in the same cell at every time instant sharing the available transmission capacity of the LTE cell. Emulator parameters can be easily changed to account for 5G characteristics which are similar to those of LTE for outdoor transmissions.

In Section 5, we describe the emulation test plan by detailing the characteristics of the considered traffic sources and parameters, the data to be extracted and processed to obtain the statistics.

In Section 6 we report the results of the tests. We consider four larger sub-sections including CBR, Transfer of a file, VBR and coexistence between two transport protocols sharing the same ACS transmission tunnel. Finally, conclusions are drawn in Section 7.

1.3 **Reference Documents**

Table 1: Reference Documents.

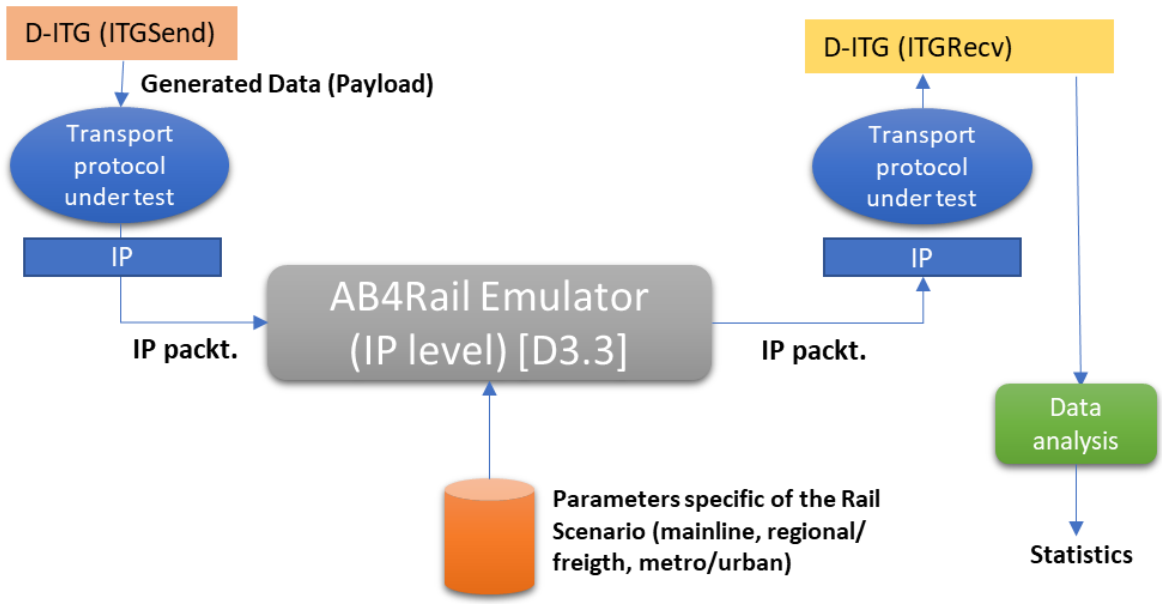
Document Number	**Document Description**
RD-1	AB4Rail Grant Agreement Number 101014517 – IP/ITD/CCA – IP2
RD-2	AB4Rail Consortium Agreement

2. High level organization of activities for transport protocol evaluation

For the evaluation of transport protocol evaluation, we have used the emulator designed and developed in the Task 3.3 [1]. The emulator can reproduce the behavior of the transmission bearers on the communication network at IP level. The Linux operating system implements the three transport protocols natively in its Kernel and reliable and well tested software for proper generation of data traffic at transport level is freely available from the open-source community. For performance evaluation we have considered the distributed Internet traffic generator (D-ITG) freely available [2]. The downloadable source code of D-ITG available on the network has been modified by us to include two important options that can be set at run-time i.e., the possibility of setting the congestion control of the TCP protocol (the new *-K <congestion>* option) and the possibility of setting the dimension of the transmitter window for TCP and also UDP (the new *-M <no bytes windows size>* option).
The organization of the test of transport protocols based on the IP emulator/simulator in [1] is schematized in Figure 1 and it consists of:

1. Generating data traffic with specific characteristics of the considered traffic source (CBR or VBR).
2. Inject application traffic in the emulator/simulator so to analyze the end-to-end achievable performance in each one of the considered network scenarios.
3. Collect statistics of the considered performance parameters i.e.
   a. One-way transmission latency i.e., the time required from one packet enqueued in the transmission buffer to reach the receiver;
   b. The receiver data rate which is referred as the receiver throughput or more simply throughput;
   c. The download time for CBR traffic sources generating a finite number of packets.
4. Analysis of results and comments

Figure 1: High-level view of the emulator setup for the transport protocol analysis.



Data source (ITGSend) and sink (ITGRecv) in Figure 1 exchange IP packets whose payload is the transport information unit (e.g., the datagram for UDP, the segment for TCP, etc.).

Based on the emulator developed in [1] we are able to model the IP packet transmission channel by varying some link-level parameters such as the bandwidth, the latency and the packet loss rate.

As shown in Figure 1 the emulator parameters related to the generation of the IP packet impairments are set in accordance with the three considered rail scenarios: mainline, regional/freight and metro/urban. To this purpose we have identified three realistic rail environments so to reproduce the characteristics of a real rail track. Then, we have considered the following three real rail lines:

- The Roma – Firenze high-speed line, able to evaluate the generic mainline environment
- The Roma – Firenze regional/freight line, able to evaluate the regional line type
- The metro of Rome, able to evaluate the metro/urban line

We have assumed the high-speed and regional/freight lines are covered with LTE radio transmission technology. In order to render the scenario more realistic we have assumed the LTE band is about 1.4 MHz and LTE transmits in the GSM-R band. This scenario has been considered in [4] and can be easily adapted to 5G even though as pointed out in this document the achievable communication capacity is not sufficient to accommodate many of the future railway services (e.g., ATC and ATO with video from Cabin). For these new services the bandwidth of the LTE or 5G should be increased provided new spectrum will be made available from regulators to the railway community.

To account for the variations with time of the available transmission capacity along the rail track in our emulation we have also included the possibility of changing the LTE modulation and coding scheme (MCS) while train moves along the track. In general, the MCS is selected by the on-board terminal in accordance with its distance from the eNB. The available transmission capacity for a train inside the LTE cell depends on:

- The MCS to be selected from the train;
- The number of trains in the same LTE cell sharing the available cell transmission capacity

The rate of variation of the capacity when the train moves in the cell thus changing the MCS depends on the train speed and then differs when passing from mainline to regional/freight and metro rail lines.

Starting from previous considerations, one of the main goals of the activities carried out in this Task 3.4 of AB4Rail project is to assess the ability/effectiveness of the transport protocol to adapt to the changing of the available transmission capacity in the considered rail scenarios. Current implementations of transport protocols such as TCP and SCTP are the results of several years of research consisting in tuning the protocol parameters to account for the different transmission characteristics of the underlying communication network [5]. Main parameters that can be set in typical transport protocols are (in the TCP case) the transmitter and receiver windows size, the dimension of the transmitter and receiving buffers, the frequency of the ACK messages and the number of segments that are acknowledge for each ACK and so on. Many of these parameters can be automatically changed by the protocol to (optimally) adapt to the actual status of the transmission channel (e.g., current available transmission capacity) so to minimize latency and improve the

receiver throughput. As an example, in Linux OS the windows size of the TCP protocol (i.e., the transmission buffer) can assume one of three values depending on the available transmission rate experienced at transport level and the measured RTT.

In general, the modern implementation of the TCP protocol could automatically adapt the transmission buffer size so to guarantee the number of bytes on the fly is close to the (measured) “bandwidth-delay” product (the delay is the RTT). In general, trying to fix one or more values of protocol parameters, such as the windows size, may lead to undesirable and pejorative performance of the transport protocol especially in challenging rail scenarios considered in this work where transmission capacity can change with time (see before). As an example, transmission capacity can increase but since the windows size has been fixed the achievable receiver rate is limited by the transmission window size especially in the case of small packet loss on the link. Furthermore, current transport protocols such as TCP may operate with or without Nagle’s algorithm which is a means of improving the efficiency of TCP/IP networks by reducing the number of packets that are needed to be sent over the network so to solve the "small-packet problem". The presence of Nagle’s algorithm in TCP can be deleterious for applications generating real-time or quasi real time traffic. As shown in the next Sections of this document, greedy CBR traffic sources, considered in this work, can provide useful information about transport protocol adaptability and suitability on the considered rail scenarios.

2.1 Notes

QUIC protocol will be analyzed in the next Task 3.5 activities and results will be provided in the next deliverable concerning the secure versions of transport protocols. In fact, the QUIC protocol is by definition a secure-by-design protocol integrating by default the TLS features. The TLS that cannot be disabled in any way. Thus, performance of QUIC Protocol will be compared with those of secure versions of the transport protocols i.e., TLS over TCP and SCTP. Furthermore, up to now QUIC protocol can be interfaced only with the HTTP protocol with the HTTP3 shim layer. Thus, QUIC is optimized to transport HTTP data and only very recently the possibility of interfacing QUIC with the RTP protocol to transport multimedia data has been considered for further study and possibly standardization.

An explanation of the performance of RaSTA transport protocol has been added in Appendix. It includes the description of the analysis approach and overall trends observed in protocol throughput. The RTP is not a communication transport protocol as TCP or UDP. RTP is stacked on UDP protocol (which is the true communication transport protocol) and it completes UDP by adding some functionalities. In fact, RTP provides facilities for jitter compensation and detection of packet loss and out-of-order delivery, which are common especially for UDP transmissions over an IP network. RTP with UDP is typically used for transferring data associated to specific applications such as VoIP and video streaming that will be studied in the next task.

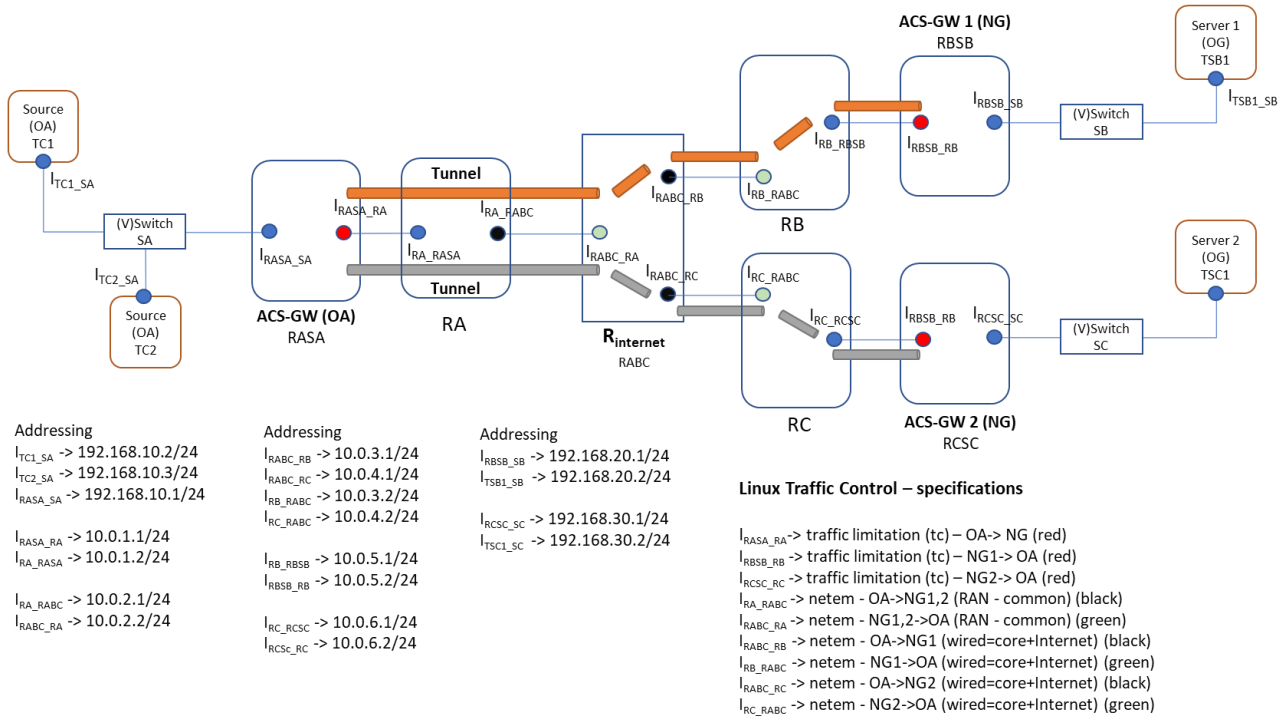
The design of RTP is based on the architectural principle known as application-layer framing where protocol functions are implemented in the application as opposed to the communication protocol stack including UDP and TCP and SCTP. Thus, RTP should be considered as an application protocol rather than a communication transport protocol and it will be discussed in the next Task 3.5 in the case of VoIP transmission.

3. Description of the emulator

3.1 Architecture and Setup

Assessment of transport protocol performance is carried out by computer emulation based on Linux operating system. The architecture of the emulator we have considered for performance evaluation is depicted in Figure 2.

Figure 2: Architecture of the Linux OS emulator used to assess transport protocol performance.



The realization of the emulator in Figure 2 is based on the software library developed in AB4Rail Task 3.3, whose operations and APIs are detailed in [1].

Using the scheme in Figure 2 we can easily emulate the behavior of the on-board network by adding more hosts generating traffic. Hosts are connected to the SA virtual switch. The applications generating IP traffic (i.e., the D-ITG traffic generator) runs inside the network namespace associated to each host indicated with TC1 (or TC2) in the scheme in Figure 2. The virtual Ethernet (veth) interfaces connecting the network namespaces TC1 and TC2 to the virtual switch are also indicated in the same scheme. More hosts can be created and then connected to the SA switch by instantiating the corresponding network namespaces and by creating the veth links.

The SA switch is connected to the router virtual machine emulating the high-level behavior of the on-board ACS gateway i.e., it only emulates the routing capabilities of the ACS GW to transmit/receive IP packets over the (already) established Generic Routing Encapsulation (GRE) based (logical) communication tunnels evidenced in Figure 2 connecting the on-board ACS GW to the ACS Network gateways. For emulation purposes explained later, two network side ACS GWs

have been considered in the emulator implementation namely ACS-GW1 (RBSB) and ACS-GW2 (RCSC). Both ACS-GWs route IP packets to/from the two local server networks in Figure 2 Servers TSB1 and TSC1 are connected to the virtual switches SB and SC respectively and then to the corresponding ACS-GW (RBSB or RCSC). Servers are created similarly to clients i.e., server applications run inside the network namespaces that in turn are connected to the switches SB or SC by veths.

From the scheme in Figure 2 it can be observed that the two tunnels share a common path from RASA to $R_{internet}$ routers. In fact, we assume that the two tunnels connecting the on-board ACS to the network ACS GWs are created over the same radio access network. As detailed later this allows to evaluate the performance of the transport protocols considering the sharing of the available transmission capacity and the effects of the same IP impairments over the same radio access channel. However, the two tunnels are connected to different destinations i.e., the paths from $R_{internet}$ to RBSB and to RCSC are different and, in general, are characterized by different impairments on the transmitted IP packets.

The task of the intermediate routers RA, RB and RC in Figure 2 is to add impairments to transmitted packets over the GRE tunnels i.e., the delay, jitter, packet loss and limitation on the available transmission bandwidth on the forward link (i.e., from on-board network to the server) and on the return link. In particular, looking at the scheme in Figure 2 we have assumed that:

In the forward link,

1. Transmission bandwidth limitation is set using token bucket (TB) strategy on the IRASA_RA veth interface (in red); the well tested transmission control (TC) traffic software integrated into Linux OS is used for this purpose;
2. Latency, jitter, packet loss on the common radio path are added in the veth IRA_RABC (black) interface connecting the RA and the $R_{internet}$ routers. The Linux netem tool is used for this purpose. The same veth interface could also be used to modify the order of transmission of packets so to require IP packet re-ordering in the receiving ACS-GW.
3. Latency, jitter, packet loss which are specific of the link connecting the radio access network to the destination ACS GW are added in the IRABC_RB and IRABC_RC (black) interfaces. The possibility of changing the order of reception of packets (thus requiring packet re-ordering at transport protocol level when available) is implemented in IRB_RBSB and IRC_RCSC (blue) interfaces from on-board to trackside transmission direction only.

In the return link,

1. Transmission bandwidth limitation is set in IRBSB_RB veth interface (red) using the TB strategy;
2. Latency, jitter, packet loss which are specific of the link connecting the radio access network to the destination ACS GW are added in the IRB_RABC and IRC_RABC (green) interfaces.
3. Latency, jitter, and packet loss on the common radio path are added in the veth I_{RABC_RA} (green) interface connecting the RA and the $R_{internet}$ routers.

Finally, in Figure 2 we have also indicated the IP addressing scheme used to transmit IP packets in the emulator. The two logical GRE tunnels are created between the on-board ACS GW and the network side ACS-GWs. The routing table in the on-board ACS-GW forwards traffic directed to

TSB1 or TSC1 on the GRE tunnel. Similarly, the network side ACS-GWs route traffic on the corresponding GRE tunnel to reach the on-board GW.

3.2 Generation of IP impairments

The procedure used to generate IP impairments concerning latency, jitter packet loss and available transmission bandwidth has been outlined in [1] and is now detailed.

Since emulation should consider a realistic railway scenario including mainline, regional and freight in the following we detail the assumptions used to evaluate some the available transmission bandwidth for the reference train in each scenario. Latency in the mobile network, jitter and packet loss are generated according to the models indicated in [1] and will be shortly summarized in the next of this Section.

To consider a realistic railway scenario in the following evaluation we consider an LTE system with bandwidth 1.4 MHz operating in the 5MHz band already allocated to GSM-R transmission. This scenario has been investigated in [4] where it was concluded that two LTE channels at 1.4 MHz and two GSM-R channels can coexist in this band. However, due to reduced bandwidth the LTE is unable to support the future railway services indicated in the CTA, documents [4], [6] conclude that more bandwidth is required in order to support new services. The LTE case is taken into consideration in this deliverable but results can be easily translated to the 5G case since LTE and 5G NSA (non-stand alone) solution are fully compatible when 15 kHz of sub-carrier spacing is considered. This spacing is typical even for 5G outdoor applications in the case of cells providing very large coverage as in the case of railway.

3.3 Evaluation of available transmission capacity for the reference train in different scenarios

We have based our evaluations by considering two realistic Italian railway lines connecting Rome with Florence, namely the Rome-Florence or Roma-Firenze, High-Speed and Regional train lines and the Metro A underground in Rome. In the following, we assume freight scenario is similar to the considered regional one.

The main characteristics of the Roma-Firenze railway lines (High-Speed and Regional) are summarized below and we have been reported in Figure 3a [7] and Figure 3b [8], respectively.

The Table 2 summarizes the allowed train speed on the Roma-Firenze regional line.

Table 2. Table maximum speeds that can be reached.

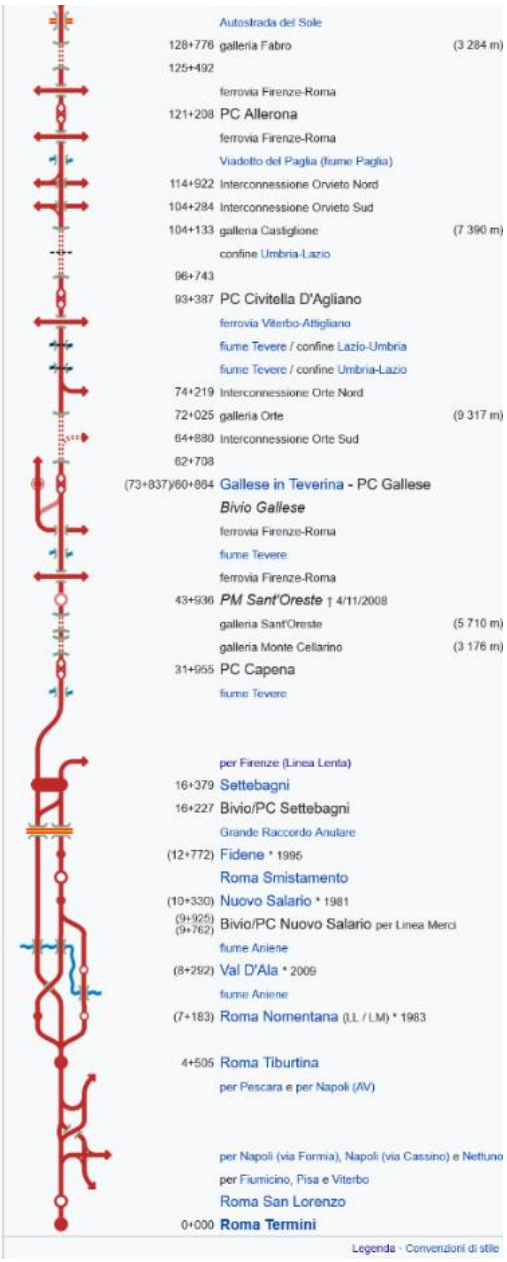
sezione / ranghi	A	B	C	P	Progressione km
Firenze S.M.N.-Cavalcavia Statuto	60	60	60	60	314-312
Cavalcavia Statuto - Firenze Campo Marte	80	85	90	105	312-311
Firenze Campo Marte - Deviatoio ultimo Rovezzano	100	105	110	130	311-305
Deviatoio ultimo Rovezzano - Incisa	90	95	100	-	305-278
Incisa - Figline	140	150	155	105	278-274
Figline - Montevarchi	140	160	180	180	274-261
Montevarchi - 2° Bivio Valdarno Sud	100	105	110	130	261-258
2° Bivio Valdarno Sud - Imbocco galleria "Croce"	95	100	105	120	258-256
Imbocco galleria "Croce" - Imbocco galleria "Ambra"	85	90	95	110	258-253
Imbocco galleria "Ambra" - Laterina	95	100	105	125	253-248
Laterina - Cippo 245+000	95	105	110	130	248-245
Cippo 245+000 - Deviatoio ultimo Ponticino	90	95	100	125	245-243
Deviatoio ultimo Ponticino - Indicatore	100	110	115	135	243-233
Indicatore - Deviatoio 1 Arezzo	115	125	130	150	233-227
Deviatoio I Arezzo - Punto medio "Olmo"	90	95	100	115	227-221
Punto medio "Olmo" - Punto medio "Rigutino"	140	155	160	180	221-215
Punto medio "Rigutino" - Deviatoio 1 Castiglion Fiorentino	140	150	150	180	215-210
Deviatoio 1 Castiglion Fiorentino - Cippo	95	105	110	85	210-208
Cippo 208+000 - Punto medio "Panicale S."	140	150	150	180	208-175
Punto medio "Panicale S." - Cippo 166+000	125	135	140	180	175-166
Cippo 166+000 - Chiusi	90	95	100	180	166-164
Chiusi - Collelungo	140	150	150	-	164-142
Collelungo - Cippo 138+000	95	100	105	-	142-138
Cippo 138+000 - Cippo 85+000	100	105	110	-	138-86
Cippo 85+000 - Orte	100	105	110	-	84-86
Orte - Poggio Mirteto	80	95	100	-	82-48
Poggio Mirteto - Cippo 46+000	80	105	110	-	48-46
Cippo 46+000 - Cippo 25+000	80	100	105	-	46-25
Cippo 25+000 - Settebagni	80	110	115	-	25-16
Settebagni - Nomentana	95	100	105	-	16-7
Nomentana - Roma Termini	60	70	75	-	7-0

Data in Table 2 have been used to set up the train speed profile which is necessary to evaluate the distance-vs-time graph of the single train in accordance with the procedure illustrated in [1].

As shown in Figure 3 the regional and the high-speed line share some segments the first one starting from Rome station to the entrance of the high-speed railway line at Settebagni. For what concern the train speed moving on the high-speed line it has been set at 250km/h from Settebagni to Firenze Rovezzano.

The selected train velocity-vs-time profile on the mainline is reported in Figure 4.

Figure 3: Stops and stations: a. high-speed; b. regional



(a)


Stazioni e fermate
per Bologna, per Viareggio
per Bologna (AV)
5+372 Firenze Castello
per Pisa
2+767 (4+840) 0+000 314+077 Firenze Rifredi
Firenze Santa Maria Novella
(2+814) Firenze Statuto
(2+380) Dev. Estr. Firenze Statuto
311+762 per Faenza
variante * 1896
(0+000) 310+117 Firenze Porta alla Croce / Firenze Campo di Marte
307+047 Firenze Rovezzano * 1996[9]
306+429 P.M. Rovezzano
ferrovia Direttissima per Roma
302+089 Compiobbi
298+636 Sieci
294+479 Pontassieve
per Borgo San Lorenzo
289+394 Sant'Ellero
per Saltino † 1922
Ponte di Sant'Ellero – fiume Arno
286+278 Rignano sull'Arno
Autostrada del Sole
ferrovia Direttissima
variante * 1935
fiume Arno
fiume Arno
276+726 Incisa (vecchia / nuova)
275+531 interconnessione Valdarno Nord con la Direttissima
274+378 Figline Valdarno
266+844 San Giovanni Valdarno
Tramvia Valdarnese
261+416 Montevarchi-Terranuova
Tramvia Valdarnese
258+219 2° Bivio Valdarno Sud interconnessione con la Direttissima
255+303 P.C. Campitello
253+695 Bucine
248+888 Laterina
243+520 Ponticino
Autostrada del Sole
233+976 P.M. Indicatore
ferrovia Direttissima
232+062 2° Bivio Arezzo Nord interconnessione con la Direttissima
per Sinalunga / Siena
227+370 Arezzo
per Fossato di Vico † 1945
221+382 P.M. Olmo
219+277 2° Bivio Arezzo Sud interconnessione con la Direttissima
215+328 P.M. Rigutino-Frassineto
209+614 Castiglion Fiorentino
199+093 Camucia-Cortona
192+738 Terontola-Cortona
per Foligno
Raccordo autostradale 6
confine Toscana-Umbria
182+725 Castiglion del Lago
175+838 P.M. Panicale-Sanfatucchio
confine Umbria-Toscana
per Empoli e ferrovia Direttissima per Firenze
interconnessione Chiusi Nord con la Direttissima (lato DD)
P.M. Montallese
interconnessione Chiusi Nord con la Direttissima (lato LL)
ferrovia Direttissima per Roma
per Ellera (non completata)
164+209 Chiusi-Chianciano Terme
confine Toscana-Umbria
156+717 P.M. Città della Pieve
interconnessione Chiusi Sud con la Direttissima
147+316 Fabro-Ficulle
Autostrada del Sole
ferrovia Direttissima
ferrovia Direttissima
135+462 Allerona-Castel Viscardo
fiume Paglia
130+453 2° Bivio Orvieto Nord interconnessione con la Direttissima
124+876 Orvieto
122+933 2° Bivio Orvieto Sud interconnessione con la Direttissima
117+236 Baschi † 2011
Autostrada A1
111+974 P.C. Castiglione in Teverina
fiume Tevere
105+108 Alviano
per Viterbo
96+093 Attigliano-Bomarzo
Autostrada del Sole
fiume Tevere / confine Umbria-Lazio
90+929 Bassano in Teverina † 2011
88+378 2° Bivio Orte Nord interconnessione con la Direttissima
per Ancona
82+503 Orte
per Capranica (senza traffico)
interconnessione Orte Sud con la Direttissima
Ferrovia Direttissima
fiume Tevere
73+837 Gallese in Teverina
fiume Tevere
69+737 Civita Castellana-Magliano
Ferrovia Direttissima
fiume Tevere
Autostrada del Sole
60+810 Collevecchio-Poggio Sommavilla
56+784 Stimigliano
52+835 Gavignano Sabino
48+521 Poggio Mirteto
ferrovia Salaria (mai realizzata)
37+130 Fara Sabina-Montelibretti
30+994 Piana Bella di Montelibretti * 1981
25+582 Monterotondo-Mentana
Autostrada del Sole
ferrovia Direttissima per Firenze
16+379 Settebagni
16+227 Bivio/PC Settebagni
Grande Raccordo Anulare
12+772 Fidene * 1995
Roma Smistamento
10+330 Nuovo Salario * 1981
9+925 9+762 Bivio/PC Nuovo Salario per Linea Merci
fiume Aniene
8+292 Val D'Ala * 2009
fiume Aniene
7+183 Roma Nomentana (L.L. / L.M.) * 1983
4+505 Roma Tiburtina
per Pescara e Napoli (AV)
per Napoli (via Formia), Napoli (via Cassino) e Nettuno
per Fiumicino, Pisa e Viterbo
Roma San Lorenzo DL Roma San Lorenzo
0+000 Roma Termini
Legenda - Convenzioni di stile

(b)

Finally, for the Metro A underground scenario in Table 3 we have reported the distances between stations of the Metro A line in Rome.

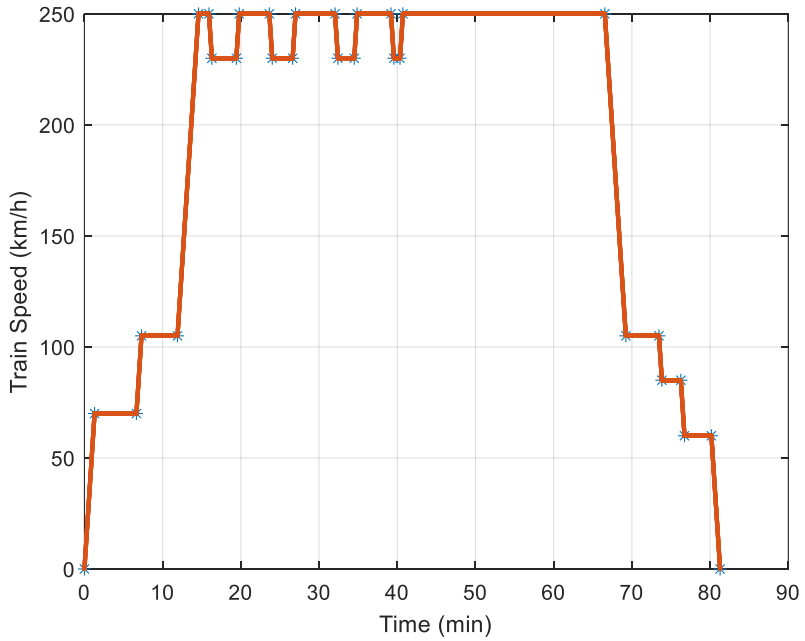
Table 3. Distances in the Metro A in Rome.

Metro Line A – Rome (Italy)	
Start-Destination	**Distance (km)**
Anagnina – Cinecitta	1.50
Cinecitta – Subaugusta	0.72
Subaugusta – Giulio Agricola	0.53
Giulio Agricola – Lucio Sestio	0.51
Lucio Sestio – Numidio Quadrato	0.58
Numidio Quadrato – Porta Furba	0.43
Porta Furba – Arco di Travertino	1.20
Arco di Travertino - Colli Albani	0.58
Colli Albani – Furio Camillo	0.79
Furio Camillo – Ponte Lungo	0.48
Ponte Lungo – Re di Roma	0.63
Re di Roma – San Giovanni	0.45
San Giovanni – Manzoni	0.66
Manzoni – Vittorio Emanuele	0.47
Vittorio Emanuele – Termini	0.85
Termini – Repubblica	0.28
Repubblica – Barberini	0.65
Barberini – Spagna	0.38
Spagna – Flaminio	0.93
Flaminio – Lepanto	0.86
Lepanto – Ottaviano	0.73
Ottaviano – Cipro	0.91
Cipro – Valle Aurelia	0.86
Valle Aurelia – Baldo degli Ubaldi	0.76
Baldo Degli Ubaldi – Cornelia	0.79
Cornelia – Battistini	1.10

Note: the distances between stations have been evaluated with Google Maps and the total length of the line is overestimated by about 1.5 km.

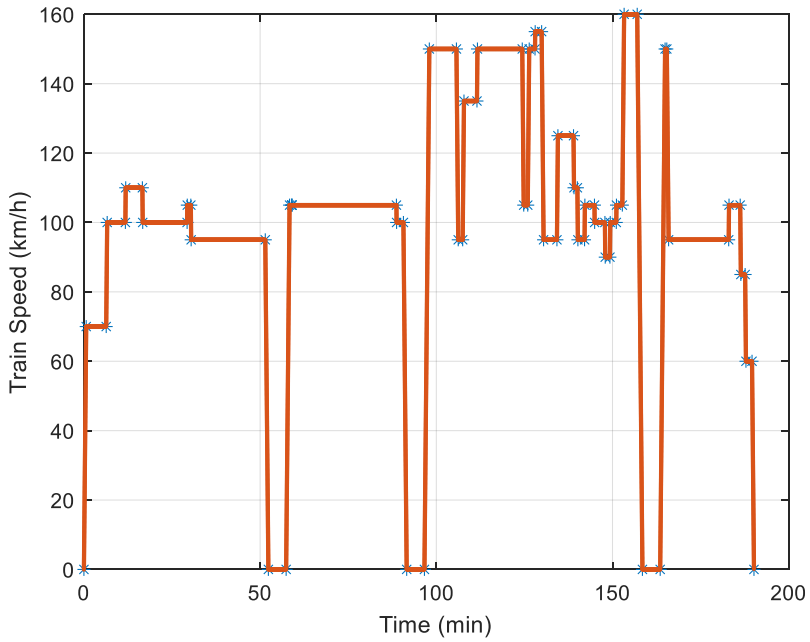
In Figure 4 we illustrate the train velocity-vs-time profile in the mainline scenario using the high-speed Rome-Florence line.

Figure 4: considered velocity vs time train profile for mainline scenario.



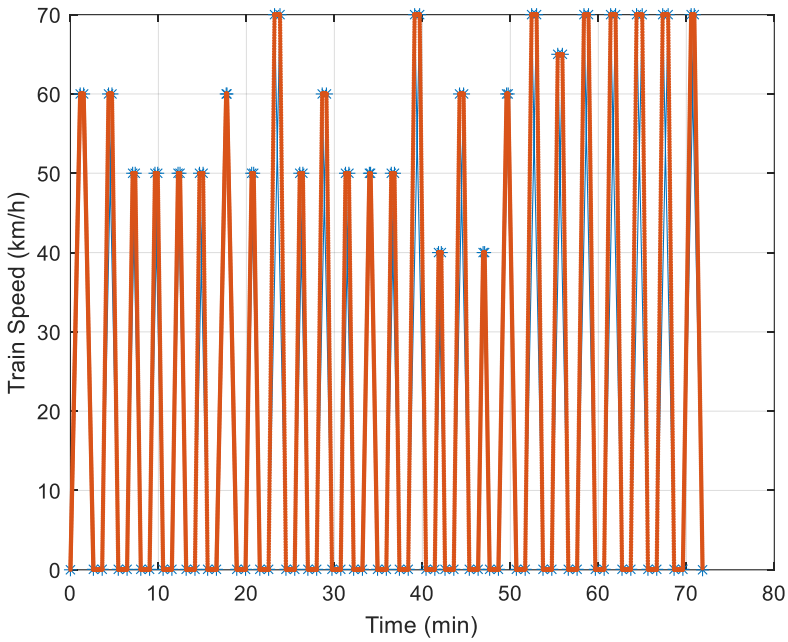
In Figure 5 we illustrate the train velocity-vs-time profile in the Regional/Freight case.

Figure 5: considered velocity vs time train profile for Regional/Freight scenario.



In Figure 6 we report the profile of the speed as a function of the time in the Metro line for the urban scenario.

Figure 6: considered velocity vs time train profile for Metro/Urban scenario.



As shown in Figure 5 we have also considered the possibility for the train to stop for up to 10 min in some intermediate stations (Orte, Terontola-Chiusi and Arezzo), while as shown in Figure 6 the assume that the Metro A train stops for 1 min at each station before restarting.

Using the train time-speed profiles in Figure 4 – Figure 6 we have determined the distance-vs-time profile for the single train as shown in [1]. In order to determine the number of trains in the same radio cell that are served by the same base station we have assumed that trains departing from Rome or from Firenze or from Anagnina in the metro/urban case, always follow the same velocity-vs-time profile thus obtaining the distance-vs-time profile for all the trains moving on the mainline or regional/freight lines. The train starting at time of zero is the reference train.

Figure 7. Distance profile as a function of the time for a train departing from Termini or from Firenze stations: mainline case

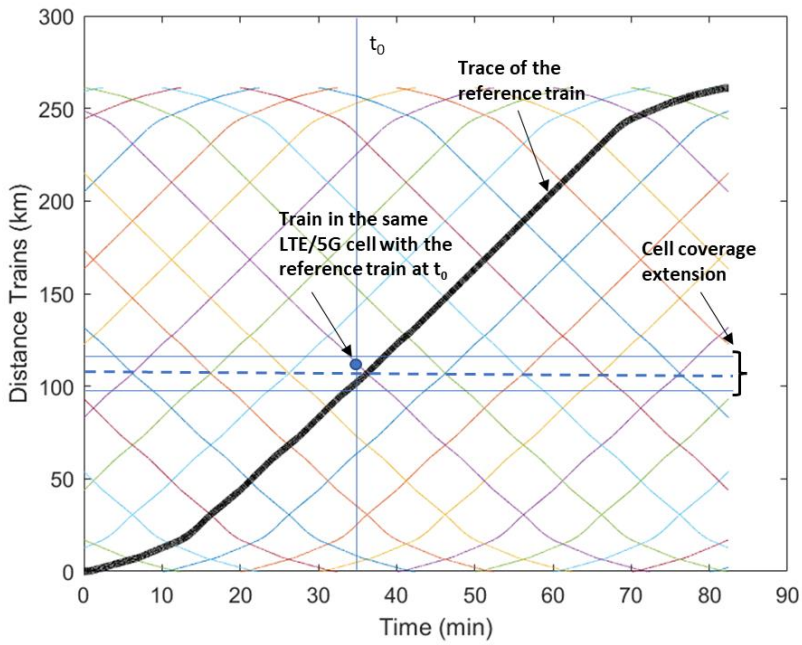
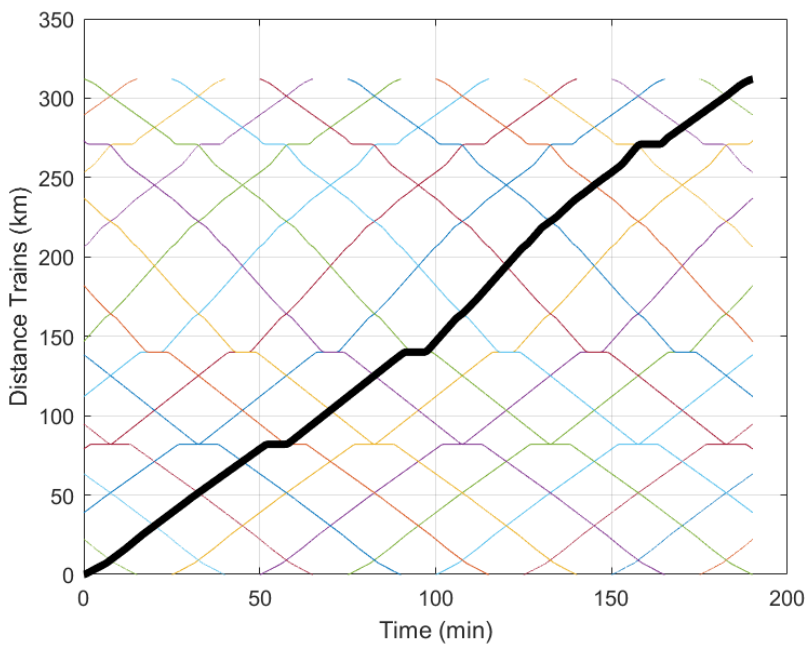
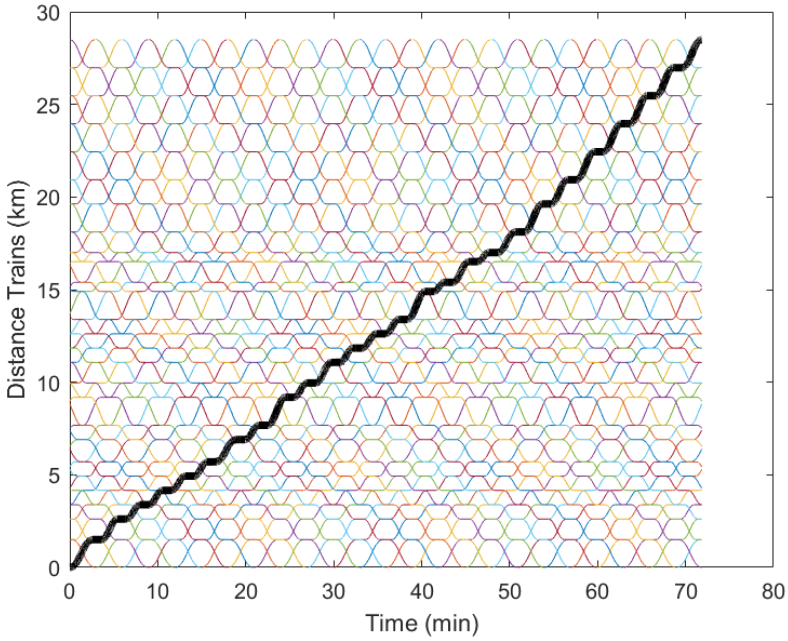


Figure 8. Distance profile as a function of the time for a train departing from Termini or from Firenze: Regional/freight line



To obtain the results in Figure 7 and Figure 8 we have assumed the time interval between two consecutive trains departures from Termini or from Firenze is about 10 min for the mainline and 25 min for the regional/freight line.

Figure 9. Distance profile as a function of the time for a train departing from one starting station of the line: Urban/metro



Results for the urban/metro case in Figure 9 have been obtained assuming the inter-departure time interval between two consecutive trains is 2 min.

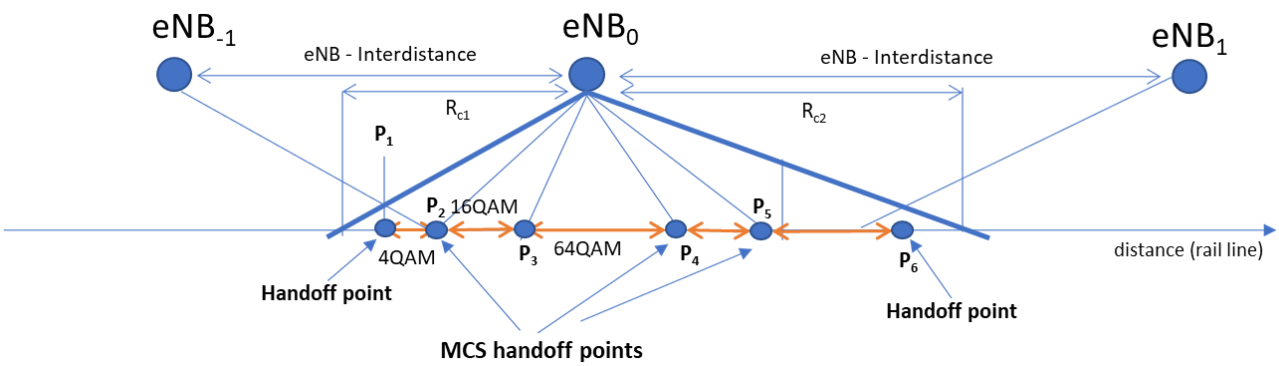
4. Description of the radio environment

4.1 Positioning of LTE cells along the railway line

We assume that LTE eNBs are positioned along the railway line and that each one is equipped with two separate sets of radiating elements one pointing in one direction and the other one at the opposite. The antenna beamwidth is assumed to be narrow enough so to follow the rail line profile. Parameters of each antenna set in the same cell, such as the transmission power, the tilt, the height and so on, can be set independently i.e., the single cell can have different coverage radii in the two opposite directions. This situation may happen when the eNB serving the cell covers two different types of area in the two directions i.e., for example, urban area on one side and sub-urban on the other side. Furthermore, we assume the first eNB is located at the starting and ending stations (i.e., Roma Termini and Firenze Santa Maria Noella).

In Figure 10 we indicate an example of the radio coverage provided by the eNB_0, which is at different distances from the adjacent eNBs, i.e. it has different inter-distances from eNB_{-1} and eNB_1.

Figure 10. Positioning of eNBs along the railway line.



In the same Figure 10 we have evidenced the handoff points between eNBs. In our setting we determine the handoff distance from the eNB by referring to a handoff power threshold of -115 dBm.

In order to evaluate the variation of the available transmission capacity inside the cell, we also evidenced the points inside the cell where the modulation-coding scheme (MCS) to be selected for radio transmission changes. The MCSs specified for LTE are summarized in Table 4 for the Urban and sub-urban area types, respectively and assuming the transmission frequencies of GSM-R for both DL and UL.

Table 4. MCS for the LTE and associated coverage distances (in km) from the eNB – Urban and sub-urban area types.

MCS	rDL_ 1.4MHz	rDL_ 5MHz	rDL_ 10MHz	rDL_ 20MHz	rUL
{'QPSK 1/3'}	2.4852	2.4872	2.4875	2.4871	2.5679
{'QPSK 1/2'}	2.0485	2.0501	2.0503	2.05	2.1166
{'QPSK 2/3'}	1.7664	1.7678	1.768	1.7677	1.8251
{'QPSK 3/4'}	1.635	1.6363	1.6365	1.6362	1.6894
{'QPSK 4/5'}	1.5629	1.5641	1.5643	1.5641	1.6149
{'16QAM 1/2'}	1.4008	1.4019	1.402	1.4018	1.4474
{'16QAM 2/3'}	1.1253	1.1261	1.1263	1.1261	1.1627
{'16QAM 3/4'}	1.0619	1.0627	1.0628	1.0627	1.0972
{'16QAM 4/5'}	1.0216	1.0224	1.0225	1.0224	1.0556
{'64QAM 2/3'}	0.86965	0.87032	0.87042	0.87029	0.89856
{'64QAM 3/4'}	0.75473	0.75532	0.7554	0.75529	0.77982
{'64QAM 4/5'}	0.7031	0.70365	0.70373	0.70362	0.72648

Urban area

MCS	rDL_ 1.4MHz	rDL_ 5MHz	rDL_ 10MHz	rDL_ 20MHz	rUL
{'QPSK 1/3' }	6.7382	6.7435	6.7442	6.7432	6.8764
{'QPSK 1/2' }	5.5541	5.5584	5.559	5.5582	5.668
{'QPSK 2/3' }	4.7892	4.793	4.7935	4.7928	4.8875
{'QPSK 3/4' }	4.4329	4.4364	4.4369	4.4362	4.5239
{'QPSK 4/5' }	4.2375	4.2408	4.2413	4.2406	4.3244
{'16QAM 1/2'}	3.7979	3.8009	3.8013	3.8007	3.8758
{'16QAM 2/3'}	3.0509	3.0533	3.0536	3.0531	3.1135
{'16QAM 3/4'}	2.879	2.8813	2.8816	2.8812	2.9381
{'16QAM 4/5'}	2.7699	2.772	2.7723	2.7719	2.8267
{'64QAM 2/3'}	2.3579	2.3597	2.36	2.3596	2.4062
{'64QAM 3/4'}	2.0463	2.0479	2.0481	2.0478	2.0883
{'64QAM 4/5'}	1.9063	1.9078	1.908	1.9077	1.9454

Sub-urban area

In the same table we have also indicated the downlink (DL) and the uplink (UL) coverage radius achievable with each one of the considered MCSs when considering different bandwidths for LTE (from 1.4 MHz to 20 MHz). To obtain data in Table 4, we selected eNB transmission power in order to balance DL and UL coverage radius. The coverage radius corresponding to the single MCS has been obtained by inverting the Okomura-Hata models including the correction terms each specific for the different area type.

Considering the hand-off power threshold of -115 dBm, the calculated inter-distance between two adjacent eNBs in the different types of areas is reported in Table 5.

Table 5. Inter-distance between two consecutive eNBs for different types of areas – handover threshold of -115 dBm.

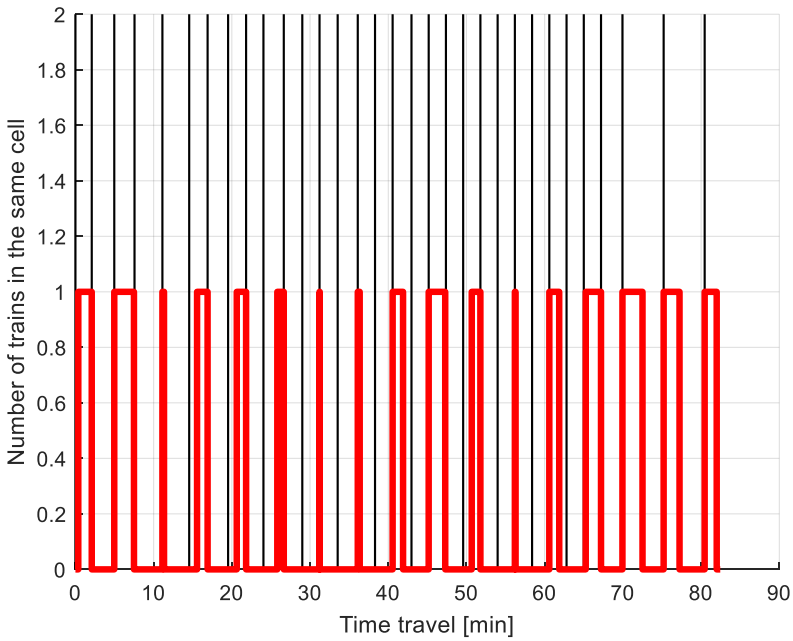
Distance between elements	Value
d_eNB – Urban	3.36 km
d_eNB – Suburban	9.12 km
d_eNB – Rural	9.6 km

The MCS at the handoff point (located at the half of the inter-distance among adjacent eNBs) is about QPSK 3/4. In our evaluation we have assumed that the maximum LTE cell radio coverage in rural area is only slightly larger than that achievable in the sub-urban area.
After classifying the different areas in the Roma-Firenze lines in urban, sub-urban and rural we have placed eNBs using the inter-distance values in Table 5. In the case of Roma-Firenze mainline the number of eNBs is about $N_{eNB} = 31$ (and 5 eNBs are in urban area in Rome and Firenze) while in the regional case we obtained $N_{eNB} = 41$ since the regional line is about 312 km and it is longer than the mainline of about 262 km.

4.2 Evaluation of the number of trains in the same cell at the same time

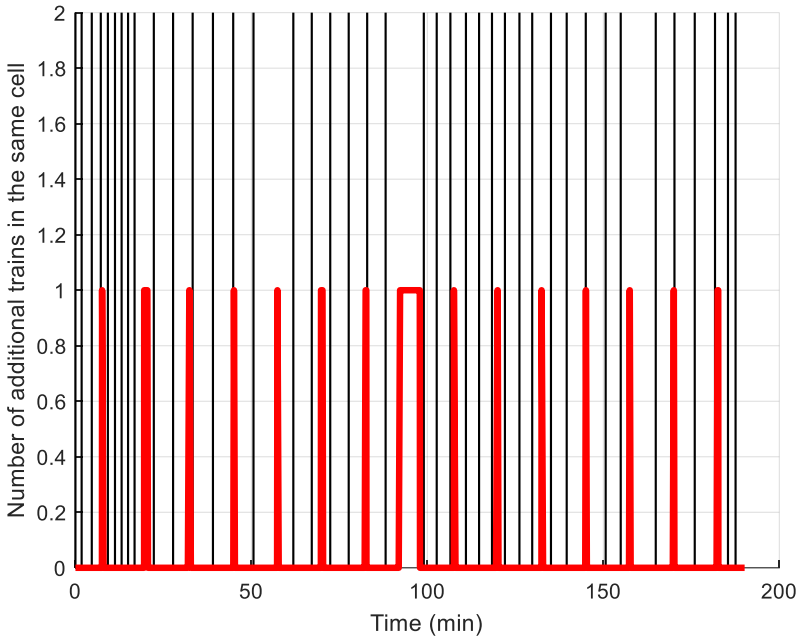
As exemplified in Figure 7 in the mainline case, to evaluate the additional number of trains in the same LTE radio cell of the reference train at each time (t_0 in Figure 7) we start by considering the position and the coverage extension of the LTE cell traversed by the reference train and we assess if the trace of one (or more) train is inside the cell at the same moment. In the case in Figure 7 we observe that at t_0 there is only one additional train is in the same cell travelling in the opposite direction. The number of additional trains in the same LTE cell with time for the mainline lines is reported in Figure 11.

Figure 11. Number of additional trains in the same LTE cell with time for the mainline.



The number of additional trains in the same LTE cell with time for the regional/freight lines is reported in Figure 12.

Figure 12. Number of additional trains in the same LTE cell with time for the regional/freight line.

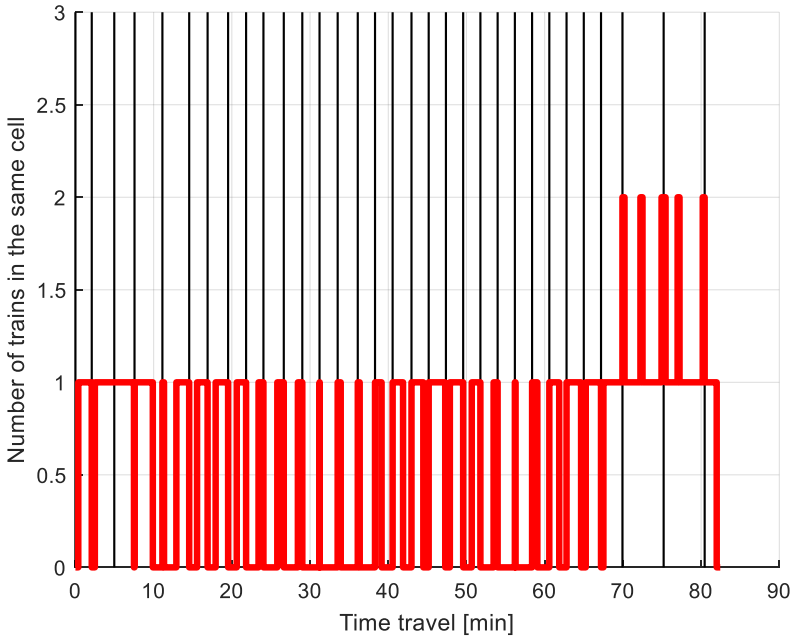


The vertical lines indicate the time interval the reference trains remain inside one LTE cell. The red line indicates the time interval the additional train remains in the same LTE cell.

From further simulations we observed that the number of trains in the cell is inversely proportional on the inter-departure time interval D between two successive trains i.e., reducing D leads to an increase in the number of additional trains in the same cell especially when the train speed is reduced such as in the case the train is approaching the destination.

As an (un-realistic) example in Figure 13 we plot the number of additional trains for the mainline assuming D=5 min.

Figure 13. Number of additional trains in the cell – inter-departure time interval D=5 min.



As expected, the duration of the time the reference train to share the radio cell capacity with another train in the same cell is increased.
When the reference train approaches the Firenze station the number of additional trains can increase to 2 thus leading to a further reduction of the available transmission capacity for each train as shown in the next Section.

4.3 Calculation of train transmission capacity

The transmission capacity available for the reference train, $C_{RefTrain}$, is evaluated at every time instant by equally dividing the available bandwidth (i.e., the number of data sub-carriers) among the trains in the same cell and then applying the MCS to evaluate the achievable bit rate:

$$C_{RefTrain} = \frac{N_{sc}}{N_{TrainsCell}} \cdot N_{bitsSymbol} \cdot C_{rate} \quad (1)$$

where N_{sc} is the number of LTE data carriers (e.g., N_{sc} =78 subcarriers for LTE band of 1.4 MHz). The $N_{TrainsCell}$ is the number of trains in the same cell in the considered time interval and we have $N_{TrainsCell} = N_T + 1$ where N_T is taken from the graphs in Figure 11.

Figure 11 for the mainline and Figure 12 for the regional/freight line. The $N_{bitsSymbol}$ is the number of bits per symbol (i.e., $N_{bitsSymbol}$ = 2, 4, 6 for QPSK, 16QAM and 64 QAM, respectively) and C_{rate} is the channel coding rate which depends on the selected MCS and can vary between 1/3 to 4/5.

To simplify emulation, we have assumed the radio link is always available (i.e., no interruption). This assumption can be acceptable when only terrestrial links are considered, and the LTE radio

coverage is properly designed to guarantee high link availability. Furthermore, in our setting the handoff time interval is not considered i.e., we assume handoff from one eNB to another one is instantaneous. As indicated in [9] this assumption is reasonable if the LTE system is designed to keep the handoff time below 100 ms which is a typical value.

We have also evaluated the LTE average transmission bit rate for the single train moving inside the cell is about C_{avg} = 2.2 Mbps. The C_{avg} has been evaluated as:

$$C_{avg} = \sum_{over\ MCS\ set} \pi_i \cdot C_i \tag{2}$$

where π_i is the percentage of cell area the train is forced to adopt a specific MCS and its value is reported in the following equation:

$$\pi_i = \frac{\Delta R_i}{\Delta R_c}$$

where ΔR_i is the extension of the interval of distances from the eNB where the train is forced to use the i-th MCS for transmission. For example, as illustrated in Figure 10 ΔR_i is the length of the interval between points P4 and P5. Finally, C_i in (2) is given in (1) with $N_{TrainsCell}$ = 1.

4.4 LTE/5G latency and packet loss

Delay must be considered due to the core network and in the eNB processing and the possibility of one re-transmission on the radio interface and scheduling request. The latency for starting transmission on the LTE radio interface on the user plane is calculated in [10] [11]. From [1] we obtain the LTE latency on the user plane varies between 10 and 20 ms.

In fact, LTE transmission is not instantaneous. When packets to be transmitted are inserted in the transmission buffer, transmission need to be activated and this requires time. This means packets are delayed. This transmission latency (network latency) calculation is reported in [3] and results are used in our emulator to set up the IP packet latency at radio link level. In particular in the following, we assume LTE latency is randomly updated between 10 and 20 ms every time the MCS is changed, while train moving inside the cell. Instead, for 5G we should consider latency randomly varying between 5 and 10 ms.

For what concerns the latency to be added on the link connecting the mobile network exit point to the destination server we assume it can vary between 8 ms (ping of google.it, RTT = 14 ms on average and standard deviation of 1 ms) to 10 ms (i.e., RTT about 20 ms on average). In [5] anything under 20ms is generally considered to be very good. In our evaluation we will assume internal network connecting the mobile network exit point with the ACS network side server is a good network (as it should be – hopefully).
In accordance with [5] a good network is characterized by the performance figures reported in Table 6.

Table 6. Performance values for a good network [5]

Metric	Target
Latency (one way)	< 50 ms
Latency (RTT or Round-trip Time)	< 100 ms
Burst packet loss	<10% during any 200 ms interval
Packet loss	<1% during any 15 s interval
Packet inter-arrival Jitter	<30 ms during any 15 s interval
Packet reorder	<0.05% out-of-order packets

As suggested in [5] for analysis purposes the following parameters are considered: the thresholds of < 1% for Packet loss, < 20ms of Jitter and <300ms RTT (i.e., about 150 ms latency which is an almost bad situation).

4.4.1 On packet loss on the radio access network

In the following we assume the main contribution to packet loss is given in the radio access network. We can safely assume that packet loss introduced by the "dedicated" backhaul and core networks, typically based on fiber optic, is negligible with respect to the packet loss in the radio access network. Focusing on 4G and 5G technologies, typical packet loss values achievable on the corresponding radio interfaces are difficult to assess. Several results have been presented in the literature but they refer to very different operating conditions and propagation model and cell scenarios and many times they do not specific of any typical rail scenarios such as mainline, regional or freight. Furthermore, it has been clearly evidenced that the achievable packet loss performance strongly depends on the selected MIMO radio access technology i.e., SISO or 2x2, 4x4 or 8x8.

In [12] some operators have indicated a measured packet loss ratio on the LTE radio interface of about 0.03% when MIMO 8x8 is adopted and increases by reducing the number of antennas reaching higher values in the SISO case. In other cases, packet loss of 1% could be typical.

In this work, in order to assess the transport protocol performance by emulation in the case of packet loss we prefer to consider packet loss as one parameter to be set in the emulator that can be 0 (i.e., no packet loss) 0.1% and 1% [13]. Our choice is also motivated by the fact that, in general, the importance of packet loss effects also depends on the type of service we are considering. As an

example, with ordinary voice and video calls, 3% up to 5% packet loss could be considered "acceptable", while this can be un-acceptable for critical voice/video and data services.

4.5 Positioning of Wi-Fi stations for urban-metro

For urban/metro scenario we have considered the Wi-Fi transmission technology based on IEEE 802.11g (i.e., this is a conservative assumption) which provides bit rates from 1 to 54 Mbps. From [14] [15] we report in Table 7 the coverage versus achievable bit rate data considering outdoor range.

Table 7. Achievable bit rate data for Wi-Fi IEEE 802.11g and its corresponding coverage in outdoor environment.

Mode	Modulation	Outdoor range (m)
1 Mbps	DSSS	550
2 Mbps	DSSS	388
6 Mbps	OFDM	300
12 Mbps	OFDM	211
18 Mbps	OFDM	155
24 Mbps	OFDM	103
36 Mbps	OFDM	72
48 Mbps	OFDM	45
54 Mbps	OFDM	36

In the case of tunnels, covered distances could be larger than that indicated in Table 7.
Even for the metro/urban case in order to assess the available capacity for the reference train, assuming fair access (i.e., the access time to the channel is equally divided among trains inside the same cell), we have first estimated the number of trains inside the same cell and we have divided the available bandwidth by the number of trains and then we have evaluated the available capacity for the reference train taking into account for its position inside the cell.
Furthermore, we have considered a worst-case scenario by assuming only one gallery in the entire track where trains can move in both directions.

4.6 Example of generated file for changing values of IP impairments during emulation

In this Section we report and comment one example of file generated by the software used to change the IP impairments when the emulator is running.
In Figure 14 we have reported one chunk of this file generated with the parameters corresponding to the LTE case that have been illustrated in the previous Section.

Figure 14. Command example for generating IP impairments during emulation

```
#!/bin/bash
#
ip netns exec RASA tc qdisc change dev irasa_ra root tbf rate 1.73mbit burst 1580b limit 128kb
ip netns exec RBSB tc qdisc change dev irbsb_rb root tbf rate 1.73mbit burst 1580b limit 128kb
ip netns exec RCSC tc qdisc change dev ircsc_rc root tbf rate 1.73mbit burst 1580b limit 128kb
ip netns exec RA tc qdisc change dev ira_rabc root netem delay 14.36ms loss 0%
ip netns exec RABC tc qdisc change dev irabc_rb root netem delay 18.7ms loss 1.1%
ip netns exec RABC tc qdisc change dev irabc_rc root netem delay 133.16ms loss 1%
ip netns exec RABC tc qdisc change dev irabc_ra root netem delay 15.43ms loss 0%
ip netns exec RB tc qdisc change dev irb_rabc root netem delay 0.1ms loss 1.1%
ip netns exec RC tc qdisc change dev irc_rabc root netem delay 134.26ms loss 1.5%
echo "Sleeping for 10.8s elapsed time 10.8s"
sleep 10.8s
ip netns exec RASA tc qdisc change dev irasa_ra root tbf rate 2.3mbit burst 1580b limit 128kb
ip netns exec RBSB tc qdisc change dev irbsb_rb root tbf rate 2.3mbit burst 1580b limit 128kb
ip netns exec RCSC tc qdisc change dev ircsc_rc root tbf rate 2.3mbit burst 1580b limit 128kb
ip netns exec RA tc qdisc change dev ira_rabc root netem delay 13.11ms loss 0.1%
ip netns exec RABC tc qdisc change dev irabc_rb root netem delay 26.64ms loss 1.3%
ip netns exec RABC tc qdisc change dev irabc_rc root netem delay 134.24ms loss 1.3%
ip netns exec RABC tc qdisc change dev irabc_ra root netem delay 18.61ms loss 0%
ip netns exec RB tc qdisc change dev irb_rabc root netem delay 0.01ms loss 1.3%
ip netns exec RC tc qdisc change dev irc_rabc root netem delay 127.59ms loss 1.2%
echo "Sleeping for 2.4s elapsed time 13.2s"
sleep 2.4s
ip netns exec RASA tc qdisc change dev irasa_ra root tbf rate 2.59mbit burst 1580b limit 128kb
ip netns exec RBSB tc qdisc change dev irbsb_rb root tbf rate 2.59mbit burst 1580b limit 128kb
ip netns exec RCSC tc qdisc change dev ircsc_rc root tbf rate 2.59mbit burst 1580b limit 128kb
ip netns exec RA tc qdisc change dev ira_rabc root netem delay 19.23ms loss 0%
ip netns exec RABC tc qdisc change dev irabc_rb root netem delay 22.35ms loss 1.3%
ip netns exec RABC tc qdisc change dev irabc_rc root netem delay 141.53ms loss 1.2%
ip netns exec RABC tc qdisc change dev irabc_ra root netem delay 19.09ms loss 0.1%
ip netns exec RB tc qdisc change dev irb_rabc root netem delay 0.11ms loss 1.3%
ip netns exec RC tc qdisc change dev irc_rabc root netem delay 135.99ms loss 1.4%
echo "Sleeping for 1.6s elapsed time 14.8s"
sleep 1.6s
ip netns exec RASA tc qdisc change dev irasa_ra root tbf rate 2.76mbit burst 1580b limit 128kb
ip netns exec RBSB tc qdisc change dev irbsb_rb root tbf rate 2.76mbit burst 1580b limit 128kb
ip netns exec RCSC tc qdisc change dev ircsc_rc root tbf rate 2.76mbit burst 1580b limit 128kb
ip netns exec RA tc qdisc change dev ira_rabc root netem delay 14.3ms loss 0.1%
ip netns exec RABC tc qdisc change dev irabc_rb root netem delay 20.4ms loss 1.5%
ip netns exec RABC tc qdisc change dev irabc_rc root netem delay 144.82ms loss 1.1%
ip netns exec RABC tc qdisc change dev irabc_ra root netem delay 18.45ms loss 0%
ip netns exec RB tc qdisc change dev irb_rabc root netem delay 0.07ms loss 1.1%
ip netns exec RC tc qdisc change dev irc_rabc root netem delay 144.39ms loss 1.4%
echo "Sleeping for 6s elapsed time 20.8s"
sleep 6s
……
```

Figure 14 extract is from the file for changing IP impairments values when the emulator is running. In the file in Figure 14 we omitted netem commands used to include jitter on delay.

As shown in Figure 14, the file is divided into several Sections. Each section is included between two sleep commands, which instruct the process to be idle for the indicated number of seconds. The duration of the sleep time interval is evaluated by software by calculating the time interval required for the train to pass from the cell region characterized by one MCS to the next one. Obviously, this time interval depends on the train speed at the moment which, depending on the selected rail scenario, can be calculated from the train speed vs time profiles in Figure 4 – Figure 6.

Looking at the single Section of the file in Figure 13 we observe that:

1. The first three command lines set the available transmission capacity at IP level calculated for each MCS as indicated in the previous Section after scaling transmission rate to the IP level [1] bandwidth limitations are applied to the *irasa_ra*, *irbsb_rb* and *ircsc_rc* as detailed in previous Section 3. The Linux tc command implementing token bucket functionality is used to limit transmission capacity at IP level. The buffer size of the virtual network interfaces has been set to 512 kbyte, which is a standard value in current network interface cards. The 1,580 byte is the token bucket size, which is slightly higher than the length of a typical Ethernet frame e.g., 1,518 bytes.

2. The next six command lines set the IP impairments (delay, packet loss) on the remaining interfaces:
   a. *ira_rabc* for the common path on the radio access network (uplink and forward path) and *irabc_ra* on downlink;
   b. *irabc_rb* and *irabc_rc* for the forward link toward the ACS-GW B and C, respectively (see Figure 2) and *irb_rabc*, *irc_rabc* on the corresponding return links.

To set the delay and loss parameters on the virtual ethernet interfaces we have used the netem tool embedded in the Linux OS.

5. Emulation test plan and performance parameters

5.1 Introduction

This deliverable responds to the objective a) in workstream 2, [16] and it is devoted to identification of the appropriate transport protocols ensuring the required communication and characteristics capabilities in the application development stage. In this case the application developer could be interested in testing data transfer (e.g., stream data transfer) only across the ACS network without considering any application protocol. Analysis is conducted considering the typical traffic patterns generated by ACS services that are grouped in the classes identified in [13] numbered from 0 to 7 (see Table 4 in [13]). These application classes include all the railway applications.

As indicated in Section 2, the main goal is to test the capability of the considered transport protocol to operate in railway environment where the most important transmission parameters such as the available bandwidth can vary with time as a function of the MCS and of the train speed. It is useful to remark that it is not of great interest to analyze the impact of the different protocol parameters on the transport performance because actual transport protocols implementation is conceived for the transport protocol to adapt its parameters in real time so to achieve better performance. It should be remarked that this aspect also provides an important motivation for preferring emulation to simulation. In fact, in the emulation case we can use the true transport protocol stack used to communicate over real networks which is also optimized for achieving performance under very different transmission situations i.e., network links with large/small round-trip time (RTT), small/large transmission bandwidth etc.

The following transport protocols will be considered in the analysis:

- TCP including two different congestion control strategies namely: cubic and the Bottleneck Bandwidth and Round-trip propagation time (BBR).
- User Datagram Protocol (UDP) and
- Stream Control Transmission Protocol (SCTP) with single stream,

The Multipath-TCP (MP-TCP) protocol could be difficult to be applied in the ACS case and it is not of interest in AB4Rail evaluation. In fact, if the MP-TCP is implemented inside the ACS-GW we need to take into account for some important aspects that may influence the design and operations of the ACS-GW control plane such as:

- GRE tunnels are usually defined over the UDP transport protocol so that flow/congestion control of using TCP is unique and managed end-to-end on the application side; if GRE tunnels would be defined over TCP or MP-TCP protocols the flow/congestion control would be replicated twice i.e. the first flow/congestion control would act between the TCP/MP-TCP layers below the GRE tunnels while the second flow/congestion control would be managed at application level (end-to-end); in this situation achievable performance could be very hard to control since the low level TCP could seriously affect the behavior of high level TCP (e.g. timeout occurrence etc.) especially in the case of packet loss.

- The MP-TCP may have problems in establishing connection when firewalls are encountered on the path connecting the on-board ACS to the network ACS through the underlying IP network(s);

- The MP-TCP already implements a scheduling algorithm for the selection of the individual TCP flows to be used for transmission; this algorithm can be configured in some way from outside the MP-TCP socket. However, the availability of an existing scheduling algorithm inside the socket for selecting and controlling flows may impact/influence the ACS design choices finalized to the best usage of available communication resources; in fact, the MP-TCP operations in selecting the flow could be out of control by the ACS-GW intelligence and this could possibly lead to un-predictable behavior in resource management.

- Finally, the SCTP seems to be a good and desirable alternative to MP-TCP, due to its possibility to manage more than one flow simultaneously. However current commercial implementation of SCTP only implements the single stream option i.e., it is not possible to activate multiple-streams and even multi-homing. The SCTP is currently under study by the Internet community to introduce improvements allowing the SCTP performance to be comparable with those obtainable with the (optimized) TCP protocol.

5.2 Traffic sources and their characteristics

As previously outlined, in order to characterize the work of the transport protocols, it is useful to analyze in a railway scenario concerns the behavior of the considered transport protocols with respect to variations over time in the transmission capacity of the channel currently available at any time along the rail line.

To study this aspect well, we decided to consider greedy CBR sources that always transmit at the maximum available capacity (theoretically) from the selected radio technology. It is useful to remark that we have considered the case of LTE at 1.4 MHz because this is a realistic case that can also be deployed in a short time since the GSM-R bands could be easily re-assigned to the LTE technology.

However, for many services indicated by the CTA [6] (excluding passengers communications) the channel capacity guaranteed by the LTE system in the GSM-R bands may be sufficient. However, as shown in the Section of results the parameters of the emulator can be easily modified to account for other situations, such as LTE with 5 MHz and 20 MHz bands. Finally, it is worth noting that the results in the next Section can also apply to the 5G case as 5G and LTE are compatible at the level of transmission capacity in the case of numerology = 0.

At transport protocol level traffic sources can be broadly classified as constant bit rate (CBR) and variable bit rate (VBR). The main parameters characterizing these two types of traffic sources are:

- The transmitting bit rate for CBR sources;
- The file dimension for a CBR source aiming at transferring a file from a client to server or vice versa;
- The peak rate (that can be variable in time) and the duty cycle (variable in time) for the VBR sources.

The CBR traffic sources may model audio and video streaming (critical or non-critical), file transfer, while the VBR traffic sources may include real time voice call, generic exchange of messages, web browsing, email, ERTMS/ETCS services based on the exchange of position report and movement authorization messages at regular time intervals.

For the purpose of this task, we considered the following settings:

- The CBR sources are greedy. They always transmit at its maximum available capacity C_{max} based on the rail line described in Section 4. This setting is suitable for the capability of adaptation of the transport protocol to bandwidth variations so to evaluate the maximum (receiver) throughput of each protocol even when the physical bit rate is lower than the data rate required for the application.

- For the file transfer, we considered two cases: 1. A small file of 1 Mbyte (e.g., a report transmitted by the train to the remote-control center or high-quality photos from the cabin), and 2. A larger file of 10 Mbyte (e.g., a software download or an on-board recorded video). In this case, it is useful to evaluate the download time experienced by each transport protocol in the different rail line and on different channel conditions.

- The VBR sources alternate two ON-OFF periods. We considered that the ON period is exponentially generated with an average of 1.2 s, while the OFF period is exponentially generated with an average of 1.8 s as in [17] [18]. Moreover, concerning the peak data rate we selected two VBR cases:

    - The peak data rate is low. In this case, we transmit packets of 200 bytes every 50 ms, thus obtaining a peak data rate of 80 kbit/s. Example of applications are the skype call, the periodic generation of messages such as ERTMS/ETCS or web browsing. The aim is not to saturate the channel bandwidth, then to evaluate the behavior of the considered transport protocols with the VBR sources according to the different rail lines and the different channel conditions in terms of channel latencies and packet loss; i.e., no effects due to packet enqueuing in the transmission buffer (see after)
    - The peak data rate is high. In this case, the peak data rate is equal to the maximum capacity of the rail line as in Section 4. The aim is to evaluate if the transport protocols behave similarly to CBR sources.

Another analysis whose results are in the next Section is the Coexistence analysis between different transport protocols. The main purpose of this analysis is to assess possible coexistence issues between different transport protocols delivering:

- Traffic flows on the same tunnel and
- Traffic flows transmitted on two different tunnels but sharing the mobile radio path inside the same radio access network

The transport protocol scenarios detailed in the following coexistence matrix will be considered for evaluation:

- TCP BBR vs TCP cubic
- TCP BBR vs SCTP

Even in this case, the performance of each single protocol is evaluated in terms of the same metrics used for assessing each transport protocol. Performance comparison will be achieved considering the mainline railway scenario which is the more challenging in terms of bandwidth variations with time.

5.3 Performance metrics

The rail scenarios are generated as indicated in the previous paragraphs and their parameters are used to set the behavior of emulator at IP layer. For each scenario (mainline or regional/freight or metro/urban) we extract chunks lasting 10 min each. The initial position along the entire line of each chunk is selected randomly. Thus, the single chunk corresponds to a different section of the considered rail line. Up to 10 different chunks were considered for the performance evaluation. For each chunk we generate one configuration file for the emulator parameters such as that indicated in Figure 14. As previously outline we do not consider handoff events i.e., we assume handovers can be performed within approximately 100 ms over all the rail line. Thus, we neglect the effects on session disconnection. See for example the paper [9]. For each chunk data transmission lasts 2 min. Data are then recorded and stored for offline evaluation of performances.

Performances are expressed in terms of the statistics concerning:

- Packet latency, calculate as difference between the reception time of the packet and the time of reception and delivery to the higher layers at the receiver side;
- Achievable Receiver Throughput (in the following indicated with TH for short);
- Download Time (DT) with file dimensions of 1 MB and 10 MB, which is a parameter useful to evaluate the performance of the file transfer service

Unlike many papers in the literature, which typically found their discussion and conclusions on the main moments of the considered statistics such as mean and variance, we have opted to express performance in terms of the Cumulative Distribution Function (CDF) of the performance parameters, because in environments where the transmission conditions (i.e., the channel capacity) vary, averages such as mean and variance may provide limited information.

6. Performance Results

In this section we discuss and analyze the results obtained from emulation for the considered transport protocol.

The presentation of results is organized in accordance with the considered type of source. For each type of source we *evaluate*, *analyze* and *comment* results concerning the statistics listed in the previous section 5.2 for each one of the considered transport protocols.

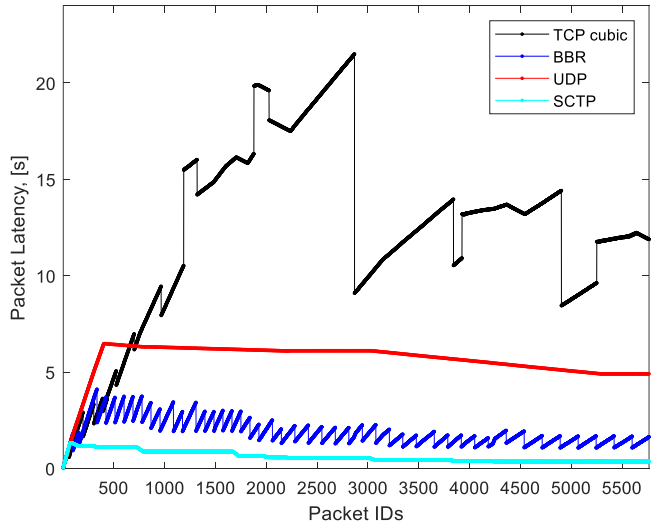
In the following the type of sources are indicated as: CBR, FILE TRANSFER and VBR. The corresponding results are reported in sub sections 6.2, 6.3, and 6.4, respectively. Coexistence is investigated in sub-section 6.6.

6.1 Introductory remarks

Before analyzing the results for each transport protocol in detail, we start our discussion by illustrating some figures showing the variation of latency of packets along the time. Latency variations are due to change of transmission bandwidth with time. In the results in the following Figures, no packet loss has been considered.

In Figure 15 we reported an example of packet latency vs packet ID corresponding to the four considered transport protocols TCP cubic, BBR, UDP and SCTP. As shown in the picture, depending on their congestion control strategy, each of them experiences a different packet latency. As expected for TCP cubic, BBR and SCTP the packets are never lost, and they adapt their congestion window to the available channel capacity to achieve receiver TH.

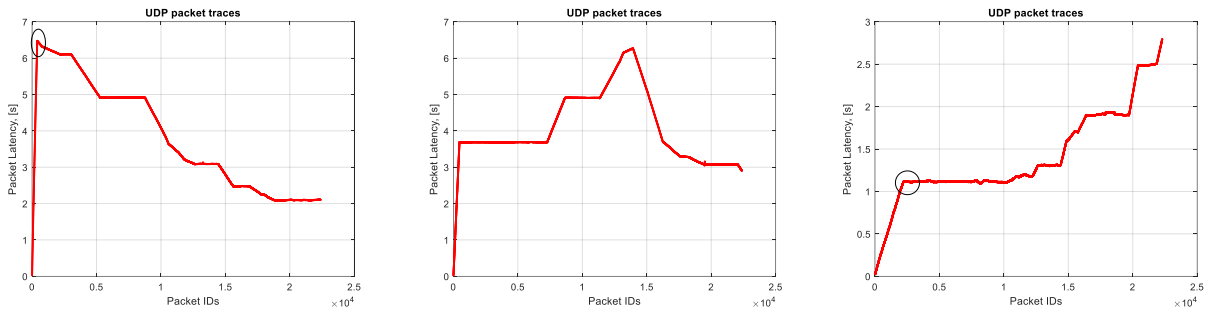
Figure 15. Packet traces of transport protocols (example).



For what concerns the UDP protocol we need to investigate ad discuss its behavior in more detail. UDP is a protocol without acknowledgments. Differently from TCP and SCTP, we noted that the UDP socket is not blocking for the application. It means that if the number of bytes delivered by the application to UDP socket exceeds the buffer size, the exceeding bytes are discarded at the transmitter.

In our test using CBR sources we assume a greedy source always transmitting at its maximum rate (e.g., 5.2 Mbps for LTE). However, in our scenario the transmission channel capacity can vary with time and the speed of change is a function of the train speed and therefore of the considered environment (e.g., High-Speed rail line). Starting from this observation, the Figure 16 shows three examples of the latency of packets transmitted using the UDP protocol in a High-Speed environment.

Figure 16. Packet traces for UDP protocol: three examples.



(a) (b) (c)

It can be observed that the packet latency grows significantly. This happens when the transmission channel capacity is lower than the data rate of the source. In this case the buffer of the network interface card (NIC) starts to fill thus causing a significant increase in latencies, which can be in the order of seconds. The values of latency obviously depend on:

- The size of the buffer (512 kB in this case),
- The difference between the channel capacity and the source transmission rate.

Below is reported a procedure for evaluating the buffer filling point (circles in Figure 16) at which the buffer overflow occurs for the first time (shown in the Figure 16). This procedure allows determining the number of packets correctly transmitted by UDP before starting to lose packets due to buffer overflow in the transmitter.

Step 1. Evaluate the bit rate difference between input (i.e., the source rate R_{source}) and output from the NIC buffer (i.e., the channel rate $R_{channel}$, which depends on the specific position in the cell of the cellular system), DR

$$DR = R_{source} - R_{channel}$$

Step 2. Evaluate the time DT to fill the buffer (with size B_S = 512 kB)

$$DT = \frac{B_S}{DR}$$

Step 3. Evaluate the number of correctly transmitted packets before starting losses based on the packet size P_S (1448 B in the considered case):

$$NL = \frac{R_{source}}{P_S} \cdot DT = tx\ packet\ per\ second \cdot DT$$

In Table 8 we reported some examples to verify the correspondence of theoretical and experimental data.

Table 8: Examples for evaluating the overflow occurrence for R_{source} = 5.2 Mbit/s

High-Speed line	$R_{channel}$	DR	DT	NL (Theoretical value)	NL (Value from emulator)
File UDP_1	0.58 Mbit/s	4.6 Mbit/s	0.91 s	408	400
File UDP_2	4.15 Mbit/s	0.95 Mbit/s	4.41 s	1982	2050
File UDP_3	1.15 Mbit/s	4.0 Mbit/s	1.04 s	468	466

After the buffer overflow the packets are discarded at the transmitter. In the cases in Figure 16, values of 71%, 75%, 27% respectively were observed. Note that in other cases, the packet loss rate was lower (e.g., about 10-15%). High percentages of packet loss are experienced when the transmission channel capacity remains below its maximum value for a relatively long time.

In the first case (Figure 16a), buffer saturation is reached very quickly with few datagrams because the available transmission channel capacity is low (i.e., 0.58 Mbit/s< 5.2 Mbps of source rate). As the channel capacity increases it is observed that packets are transferred faster. But it should be noted that once the buffer is filled it can no longer be emptied until the transmission rate becomes equal or higher to the source transmission rate is always equal to the maximum channel capacity (i.e., 5.2 Mbit/s). When the transmission channel capacity is of 4.15 Mbit/s (see example 3) the packet emission latency would be around 1 second (value highlighted in the figure example 3).

6.2 Constant Bit Rate

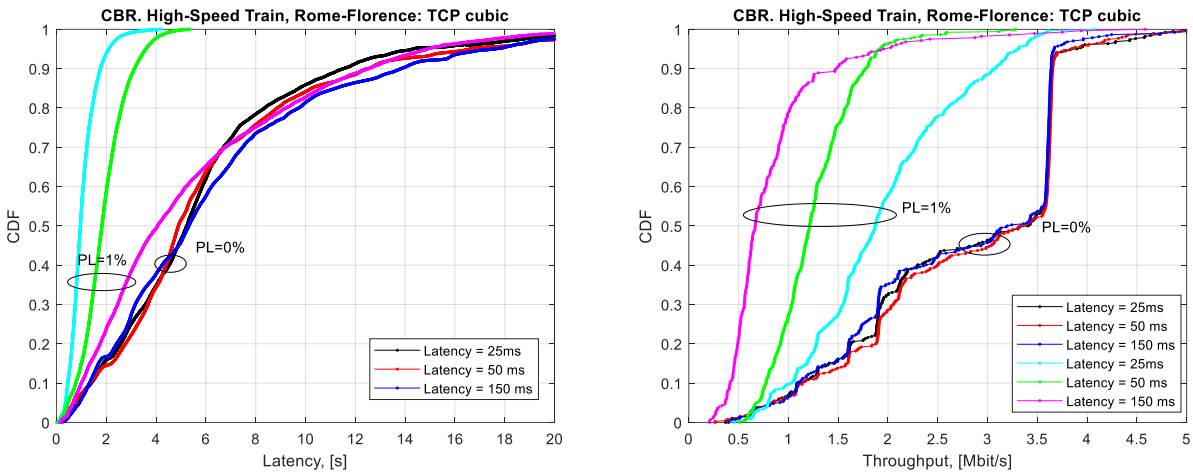
6.2.1 High-Speed rail line

In Figure 17, we report the CDF of packet latencies and receiver throughput. The TCP cubic case is considered and results have been reported for (average) channel delays of 25 ms, 50 ms and

150 ms and for a PL = 0% and 1%. The High-Speed line case has been considered.

Latencies increase due to the reduced transmission channel capacity in the wireless access (varying according to the number of trains in the cell and to the MCS in the cell) lower than the maximum data rate transmitted by the source. Data are buffered and delayed by the TCP entity in order to avoid packet loss. It should remined that the TCP socket is blocking for the application and that the TCP can adapt the buffer size.
Ideally when considering CBR sources we would expect that when packet loss is very small or zero the TH at the receiver should be able to use all the available transmission channel capacity at each time interval. In this sense the TH CDF curves obtained at PL=0% should be seen as reference, since they should follow the statistics of the transmission channel capacity as it varies along the entire rail track. However, as shown in the following, transport protocols may suffer in the case the PL is non-zero. This effect is also evidenced in Figure 17b, where the TH for PL=0% is practically independent by the channel latency and follows the bit rate of the channel capacity of the High-Speed rail line. In case of PL = 1%,
TCP cubic experience a reduction of TH due to the reduction of congestion window leading to reducing the data that can be transmitted.

Figure 17. CDF for TCP cubic in terms of latency (a) and throughput (b) for High-Speed train line

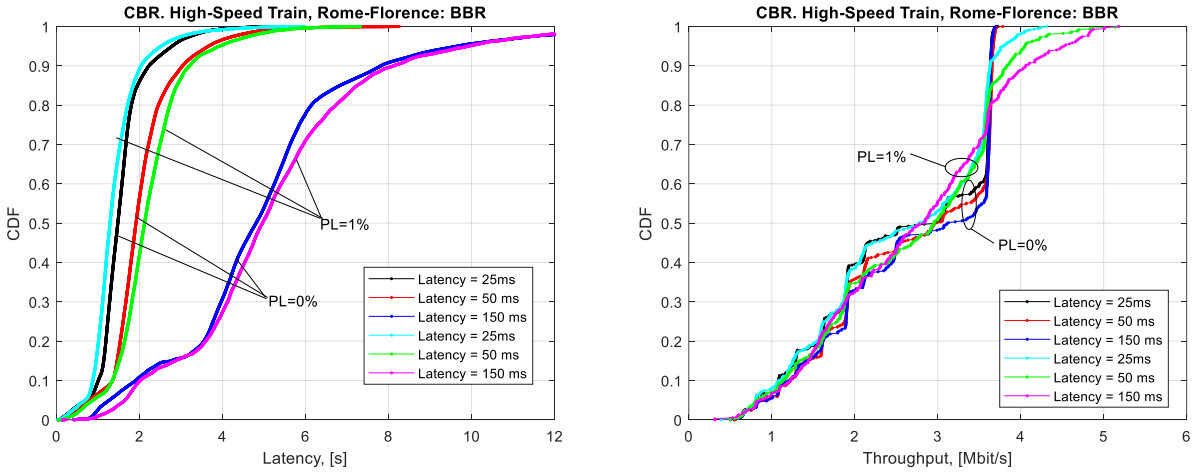


(a) (b)

In Figure 18, we reported the CDFs for the packet latencies and receiver throughput for the TCP BBR case in the same channel delays situations of 25 ms, 50 ms and 150 ms and for a PL = 0% and 1% and High-Speed line. Similarly to TCP cubic, even for TCP BBR latencies increase due to variations in the available transmission channel capacity. However, differently from TCP cubic, for BBR latency is less sensitive to the PL of the channel. In fact, as also showed in Figure 18b, the TH in the BBR case show a substantial insensitivity by PL. In fact, the receiver THs obtained for PL=0% and PL=1% are similar i.e., invariant in practice. This shows an interesting behavior of TCP BBR in presence of packet loss on the transmission channel that significantly differentiates it from other transport protocols.

This behavior is due to the particular operation of the congestion control implemented in BBR, which is very different from that of TCP cubic and it is based on different criteria. In fact, differently from other TCP flavors, the TCP BBR frequently monitors the RTT by measuring the received ACKs and periodically (e.g., every eight RTT) deliberately sends packets at a rate which is higher than the available channel capacity (e.g., more precisely at a multiple of the measured bandwidth delay product of the network path). This pushing data in the channel will favor to rapidly acquire any extra available channel bandwidth that can become available when the train is moving along the rail line. However, this leads to always fill the queues and this leads to an increase in packet latency. Subsequently, the sender will reduce the sending rate for a given time interval to compensate the extra sent data, allowing the bottleneck queue to drain. Moreover, note that the latencies of BBR are more sensitive to the channel latency due to how the BBR estimates the RTT (i.e., by marking sent packets and measuring the ACK delays).

Figure 18. CDF for BBR in terms of latency (a) and throughput (b) for High-Speed train line



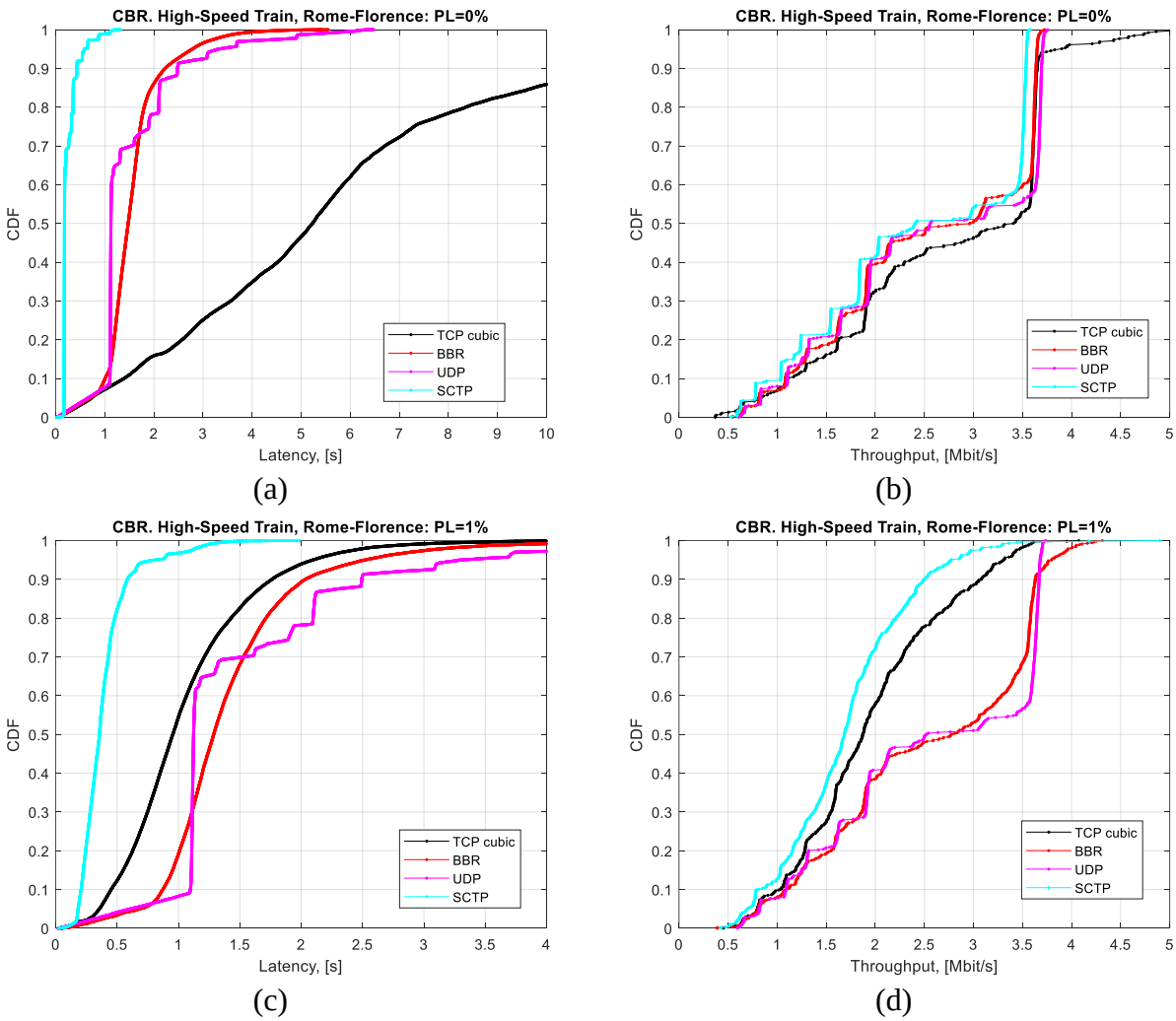
(a) (b)

In Figure 19 we report the CDFs for latency and throughput for the four analyzed transport protocols TCP cubic, BBR, UDP and SCTP. The PL = 0% and PL = 1% with channel latency of 25 ms for the High-Speed rail line are considered. The aim of results in Figure 18 is to highlight the behavior of the SCTP compared to the other two TCP protocols. In the same figure we reported also the UDP even if comparison of results needs to be considered with caution since it is hard (or has non-sense) to compare a lossy protocol with lossless protocols.

Although SCTP is not as performant in terms of TH as cubic and BBR, instead it tends to preserve latency. This is because SCTP uses a Reno-like congestion control algorithm that tends to reduce the transmission window/congestion window in the presence of anomalies (e.g., packet loss and buffer overflow due to channel capacity variation). This leads to a reduction of receiver TH for SCTP but latency is lower than that obtained with other transport protocols (i.e., TCP cubic and TCP BBR). In fact, these latter protocols are more aggressive from the point of view of receiver TH. SCTP instead tries to reduce the buffer queuing. This feature makes the use of SCTP particularly interesting for services with low data rates where latency is important such as low bit

rates services related to the transfer of signaling data (e.g., SCTP is used in the control plane in LTE).

Figure 19. CDF for TCP cubic, BBR, UDP and SCTP in terms of latency (a) (c) and throughput (b) (d) and for different PL values for High-Speed train line.

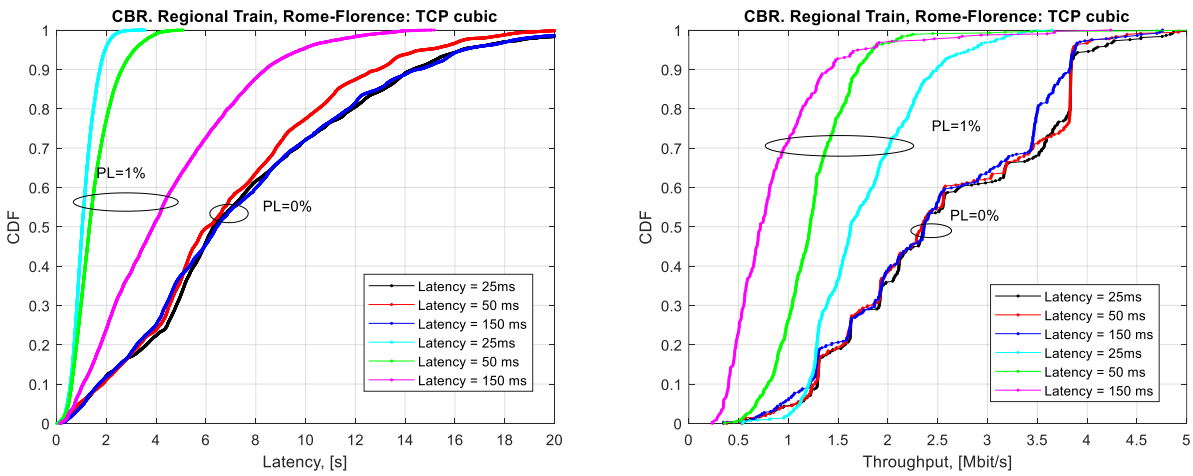


6.2.2 Regional rail line

In Figure 20 and in Figure 21, we reported the CDFs of the latency and receiver TH for the TCP cubic and of TCP BBR, respectively. Even in this case results are presented for channel latencies of 25 ms, 50 ms and 150 ms and for a PL = 0% and 1%. The Regional line is now considered. The behavior of TCP cubic and TCP BBR on the Regional line is similar to that of the High-Speed line. However, the receiver TH is slightly lower than the mainline due to reduced speed of variation of the transmission capacity which is related to the train speed. In fact, in Regional lines the train moves at lower speed so that it can remain in a zone in the cell with lower MCS for a larger

percentage of time with respect to the high-speed line.
As expected, the throughput of TCP cubic is high for PL = 0%, while it reduces when errors occur in the channel (case PL = 1%). Latencies are higher in case of lossless channel since packets are buffered and the congestion window remains large (see Figure 20a,b).

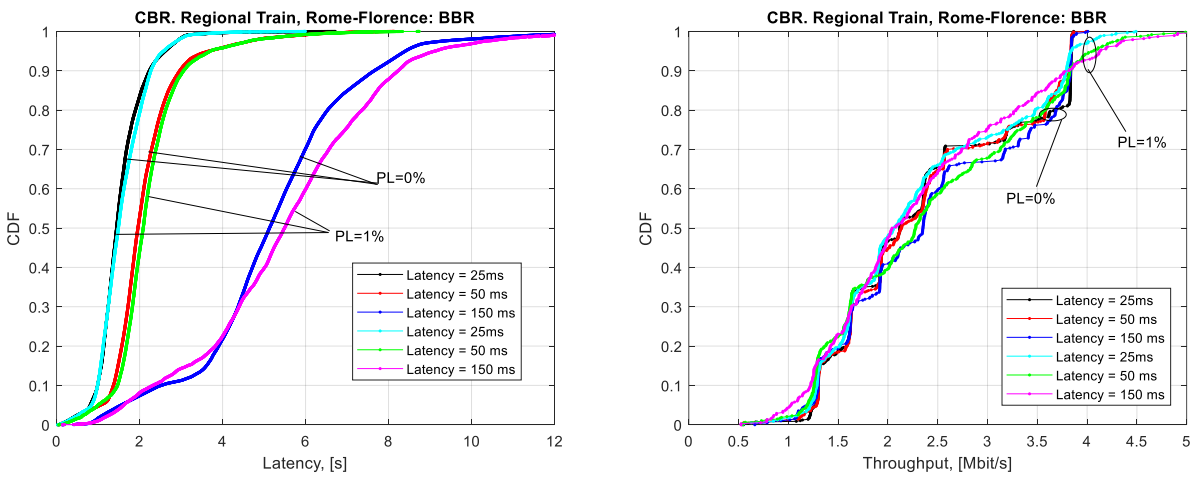
Figure 20. CDF for TCP cubic in terms of latency (a) and throughput (b) for Regional line



(a) (b)

As expected, even on Regional lines, the TCP BBR performs better than TCP cubic when the channel is lossy (PL = 1%), since it is able to track the channel capacity variations as the train moves along the line and to adapt its rate when losses occur. Even in this case due to reduced train speed the TCP BBR curves concerning the receiver TH with and without channel packet loss are similar. Even in this case the sensitiveness of TCP BBR to channel latency is similar to that observed for the High-Speed line (see Figure 21a,b).

Figure 21. CDF for BBR in terms of latency (a) and throughput (b) for Regional line



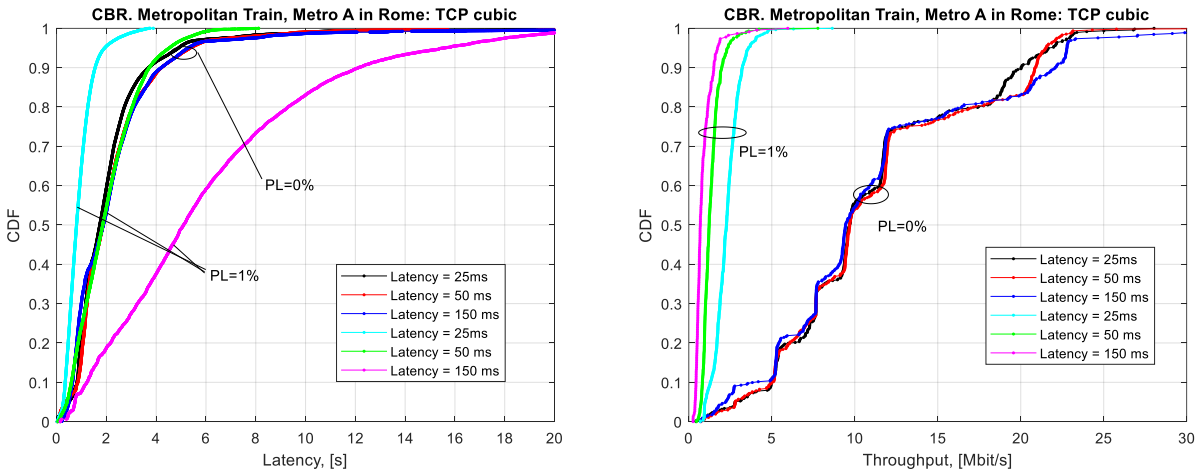
(a) (b)

6.2.3 Metropolitan line

In Figure 22 and in Figure 23, we reported the CDF of the TCP cubic and of BBR for the latencies and throughput. Even in this case channel delays are set to 25 ms, 50 ms and 150 ms and for a PL = 0% and 1%. The Metro case scenario is considered. The curves in Figure 22 and in Figure 23 further confirm the behaviors of TCP cubic and TCP BBR already observed on the other rail scenarios. The main differences are in terms of experienced throughput. In Metropolitan line the available bit rate in the wireless channel is higher than that for High-Speed/Regional lines, since the adopted technology is Wi-Fi.

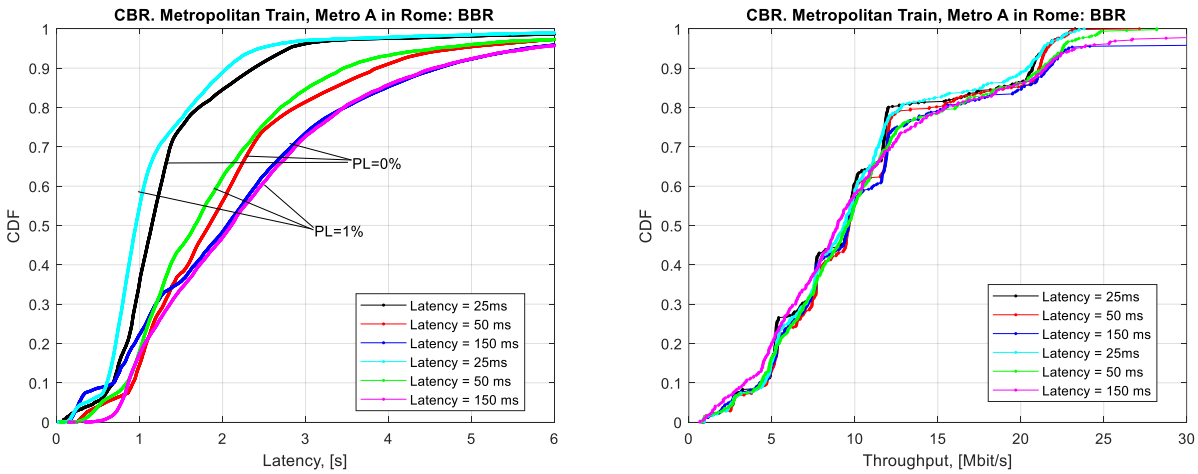
As for High-Speed/Regional lines, also in the Metropolitan case the TCP cubic suffers for non-zero PL.

Figure 22. CDF for TCP cubic in terms of latency (a) and throughput (b) for Metropolitan line



(a) (b)

Figure 23. CDF for BBR in terms of latency (a) and throughput (b) for Metropolitan line

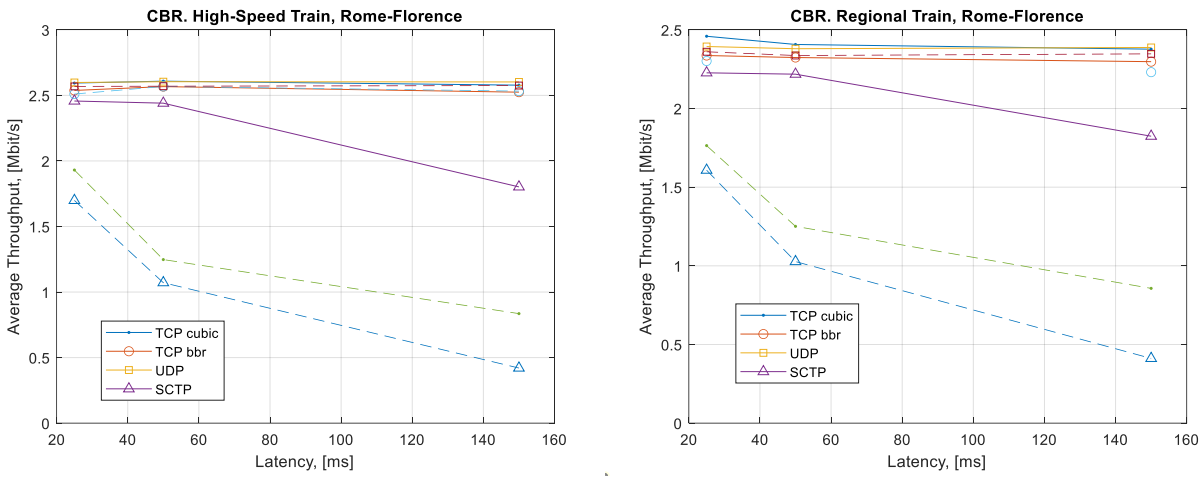


(a) (b)

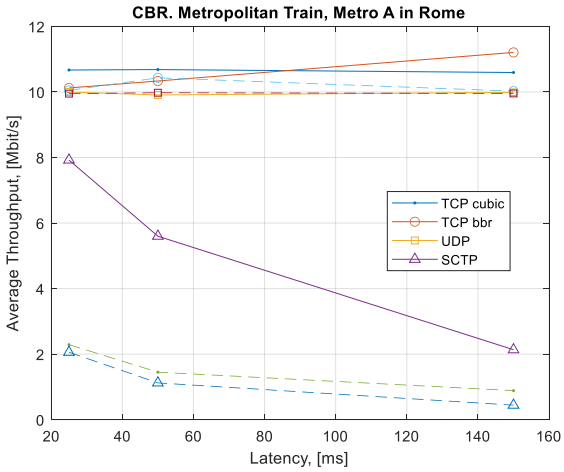
6.2.4 Average Throughput

In Figure 24 we indicate the average throughput as a function of the channel latency (e.g., 25ms, 50ms and 150ms) for all four transport protocols: TCP cubic, TCP BBR, UDP and SCTP. We reported TH values for the three considered scenarios: High-Speed in Figure 24(a), Regional in Figure 24(b) and Metropolitan in Figure 24(c). Moreover, we considered lossless case (PL = 0%) in the solid lines, while lossy case (PL = 1%) in the dashed lines.

Figure 24. Average Throughput vs channel latency



(a) (b)



(c)

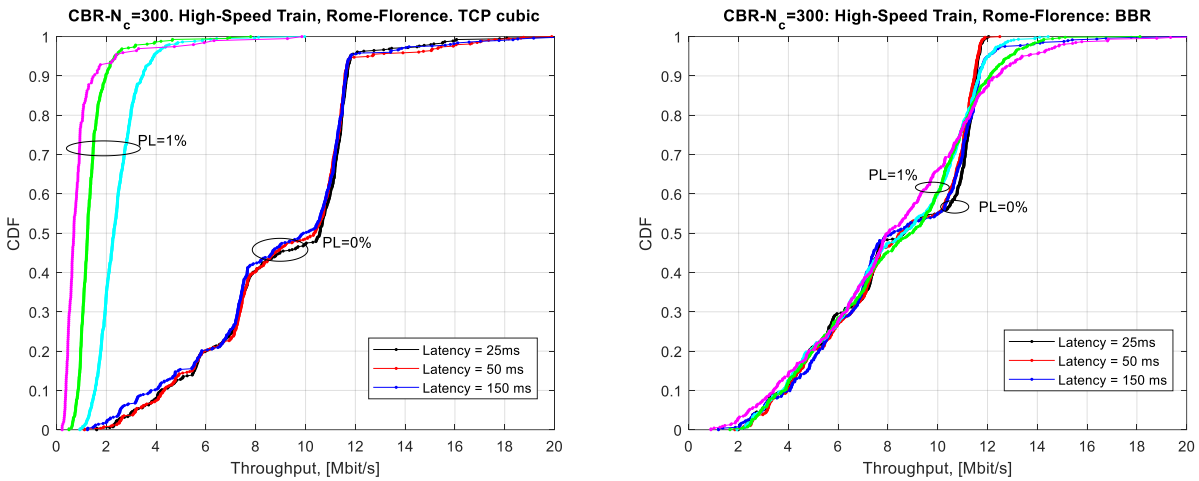
As expected and as explained in the previous Sections, TCP cubic performs better in case of lossless or very small loss channel, while degrades for lossy channel. Performance of TCP BBR are insensitive to PL. Instead SCTP experiences a high TH for lossless channels and low channel latency (e.g., lower than 50 ms), while for latency of 150 ms SCTP has a small TH even for PL = 0%. This is due to variations of transmission channel capacity that can lead the congestion control algorithm to reduce the transmission window. For the Metro line, SCTP has a lower TH (reduced from 20% to 80% w.r.t cubic and BBR) even for lossless channel. Even in this case, this behavior is due to how the SCTP manages its congestion window (as also explained in section 6.2.1). TCP BBR performs similarly in case of lossless and lossy channel. Also, UDP has been reported for comparison. UDP throughput is independent of the channel latency and packet loss rate but it has limited importance due to buffer overflow and the subsequent loss of data)

6.2.5 Larger bandwidths for High-Speed line

Before concluding this section on CBR source, we investigate the possible effects related to the usage of LTE with larger bandwidth. Analysis is restricted to TCP cubic and TCP BBR protocols on the achievable protocol throughput. For reasons that will be clear in the following only the mainline is considered.

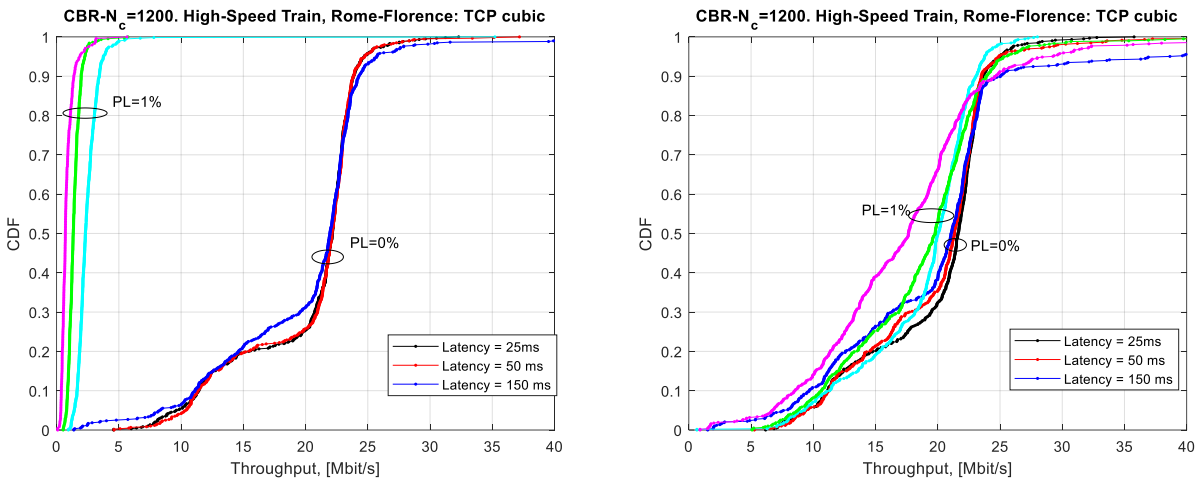
In Figure 25 and Figure 26 we reported the case of an LTE with 5 MHz (i.e., with a number of subcarriers of N_c = 300) and the case of an LTE with 20 MHz (i.e., with a number of subcarriers of N_c = 1200), respectively. Similarly to the case of the case of an LTE with 1.4 MHz (i.e., with N_c = 72 – see results in Figure 17 and Figure 18), TCP cubic performs better in case of lossless channel, while TCP BBR always performs at its higher TH both for lossless and lossy channels. Nevertheless, it is evident the increase in the experienced maximum throughput due to the larger bandwidth availability. Considerations on receiver TH and latency remain un-altered with respect to the LTE at 1.4 MHz bandwidth analyzed before.

Figure 25. CDF of throughput for TCP cubic (a) and BBR (b) for N_c = 300 (5 MHz) for High-Speed line



(a) (b)

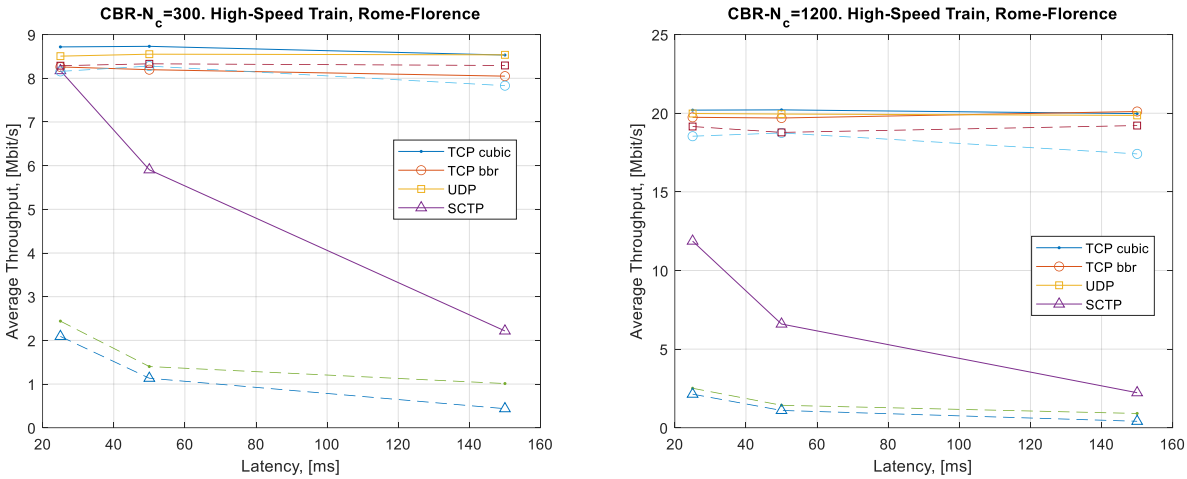
Figure 26. CDF of throughput for TCP cubic (a) and BBR (b) for N_c = 1200 (20 MHz) for High-Speed line



(a) (b)

To complete the analysis, we reported in Figure 27 the average Throughput as a function of the channel latency for larger LTE bandwidths for High-Speed line. Looking also to results in Figure 24, it can be noted that the maximum throughput increases to about 8.5 Mbit/s for N_c = 300 and to about 20 Mbit/s for N_c = 1200 w.r.t about 2.5 Mbit/s for N_c = 72. Finally, note that TCP cubic and SCTP perform similarly in case of N_c = 300, 1200 w.r.t N_c = 72 in case of lossy channel.

Figure 27. Average Throughput vs channel latency for larger LTE bandwidths for High-Speed line: (a) $N_c = 300$; (b) $N_c = 1200$.



(a) (b)

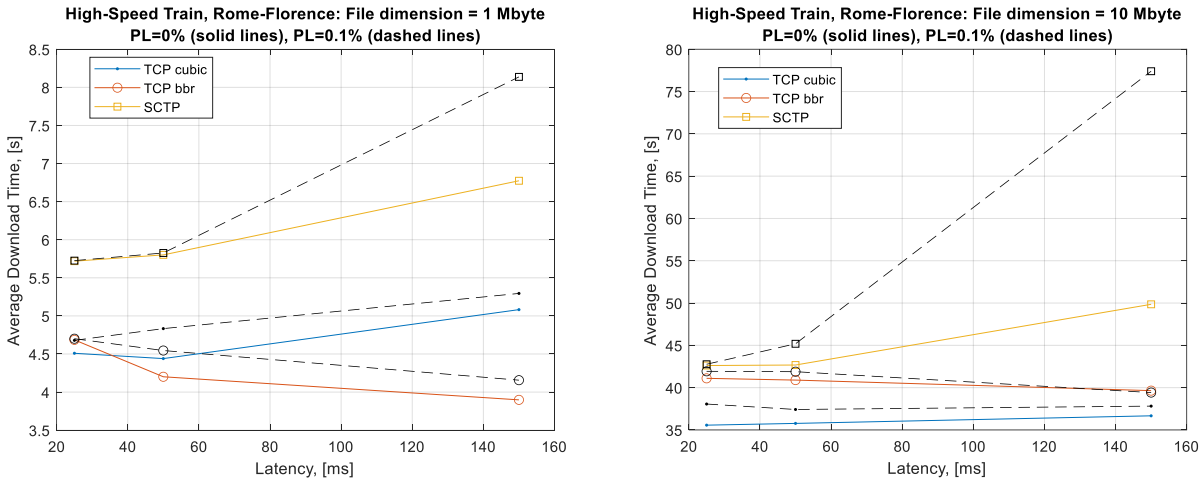
6.3 File Transfer

In this section we analyze the case of file transfer. We report the Download Time (DT) for TCP cubic, BBR and SCTP. We didn't consider UDP in this case, since UDP alone is not used for these types of services where file integrity needs to be guaranteed. We have considered two files with dimensions of 1 MB and 10 MB and all three rail lines: High-Speed (LTE of 1.4 MHz), Regional (LTE of 1.4 MHz) and Metropolitan (Wi-Fi).

6.3.1 High-Speed rail line

In Figure 28 we reported the average Download Time for TCP cubic, TCP BBR and SCTP protocols as a function of the channel latency varying from 25 ms to 150 ms in the High-Speed train line case. We considered both no packet loss (solid and colored curves) and PL = 0.1% and 1% (dashed and black curves).

Figure 28. Average Download Time for TCP cubic, BBR and SCTP vs channel latency for High-Speed train line. File dimensions: (a) 1 MB and (b) 10 MB.



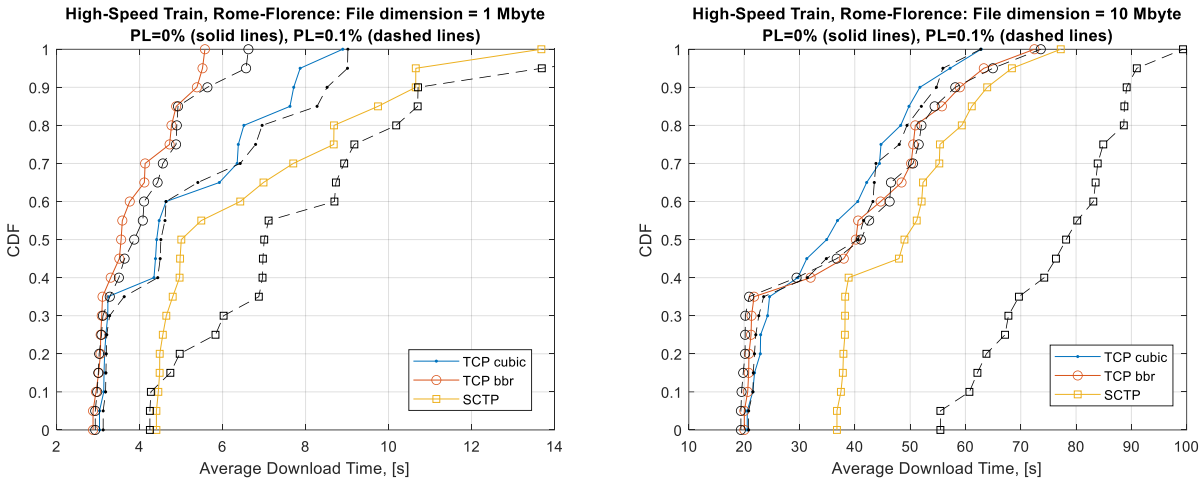
(a) File 1 Mbyte (b) File 10 Mbyte

For limited file dimension, all three transport protocols show comparable values for the DT: about 4-5 s for TCP cubic and BBR, about 6-7 s for SCTP. Performance experienced a limited degradation for PL = 0.1%, while the degradation is more marked for PL = 1% for SCTP (DT is about 20 s not reported in Figure 28).

For a file dimension of 10 MB, the DT of SCTP increases noticeably above all for PL = 0.1%, while for TCP cubic and BBR the DT is about 36-42 s both for PL = 0% and PL = 0.1%. Note that for a file of 10 MB, TCP cubic is more sensitive to the channel packet loss rate. DT for TCP cubic can be even more than 1 minute for PL = 1% (again not reported in Figure 28), while DT degradation for TCP BBR is smaller.

To complete the analysis, we reported in Figure 29 the CDF of the Download Time for TCP cubic, BBR and SCTP for channel latency of 150 ms. We considered High-Speed train line and the two file dimensions of 1 MB and 10 MB. In the figure we reported the case PL = 0% (solid and colored curves) and the case of PL = 0.1% (dashed and black curves).

Figure 29. CDF of Download Time for TCP cubic, BBR and SCTP for High-Speed train line (channel latency = 150 ms). File dimensions: (a) 1 MB and (b) 10 MB



(a) File 1 Mbyte (b) File 10 Mbyte

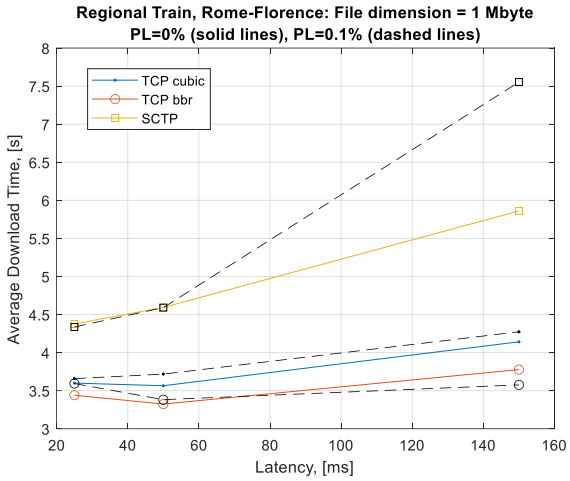
Obviously, CDF confirm averaged data of the previous figure. The most interesting case is the DT for 10 MB (Figure 29b) where it is evident the DT increase for the SCTP in case of PL = 0.1%.

6.3.2 Regional rail line

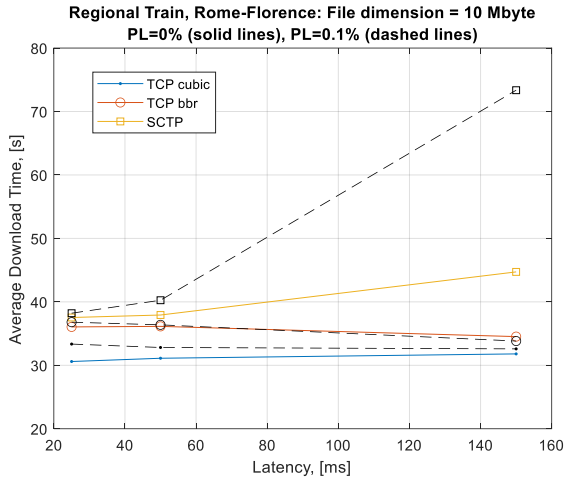
In Figure 30 we reported the average Download Time for TCP cubic, BBR and SCTP as a function of the channel latency (from 25 ms to 150 ms) for Regional train line. We considered both lossless case (solid and colored curves) and case of PL = 0.1% (dashed and black curves).

Regional line shows similar performance of those of High-Speed line. TCP cubic and BBR have the same DT independently of the channel latency of about 3.5-4.5 s (for 1 MB) and 32-35 s (for 10 MB) both for PL = 0% and PL = 0.1%. Due to its congestion window managing, SCTP is not able to follow the channel capacity variation due to the train movement within the cell.
This provides a DT for SCTP of about 4.5 s (for 1 MB) and 37-40 s (for 10 MB) for channel latency lower than 50 ms, values similar to TCP cubic and TCP BBR under the same operating conditions. As the channel latency increases (e.g., 150 ms) as well as the packet loss rate (e.g., 0.1%), SCTP performance rapidly degrades showing a DT of 7.5 s (for 1 MB) and about 75 s (for 10 MB) as worst case.

Figure 30. Average Download Time for TCP cubic, BBR and SCTP vs channel latency for Regional train line. File dimensions: (a) 1 MB and (b) 10 MB.



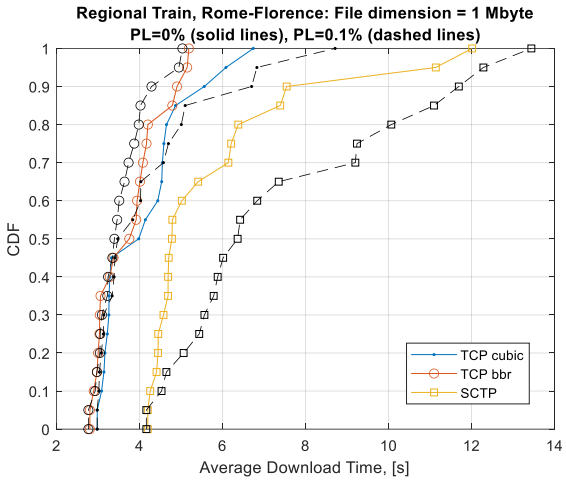
(a) File 1 Mbyte



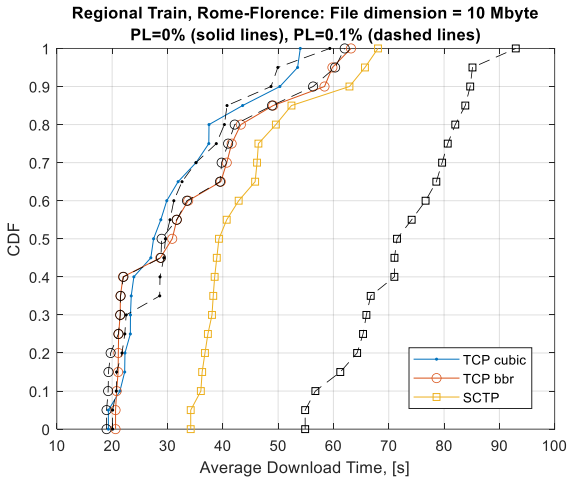
(b) File 10 Mbyte

To confirm the results in the previous figure, we reported in Figure 31 the CDF of the Download Time for TCP cubic, BBR and SCTP for channel latency of 150 ms also in the case of Regional line and the two file dimensions of 1 MB and 10 MB. In the figure we reported the case PL = 0% (solid and colored curves) and the case of PL = 0.1% (dashed and black curves).

Figure 31. CDF of Download Time for TCP cubic, BBR and SCTP for Regional line (channel latency = 150 ms). File dimensions: (a) 1 MB and (b) 10 MB



File 1 Mbyte

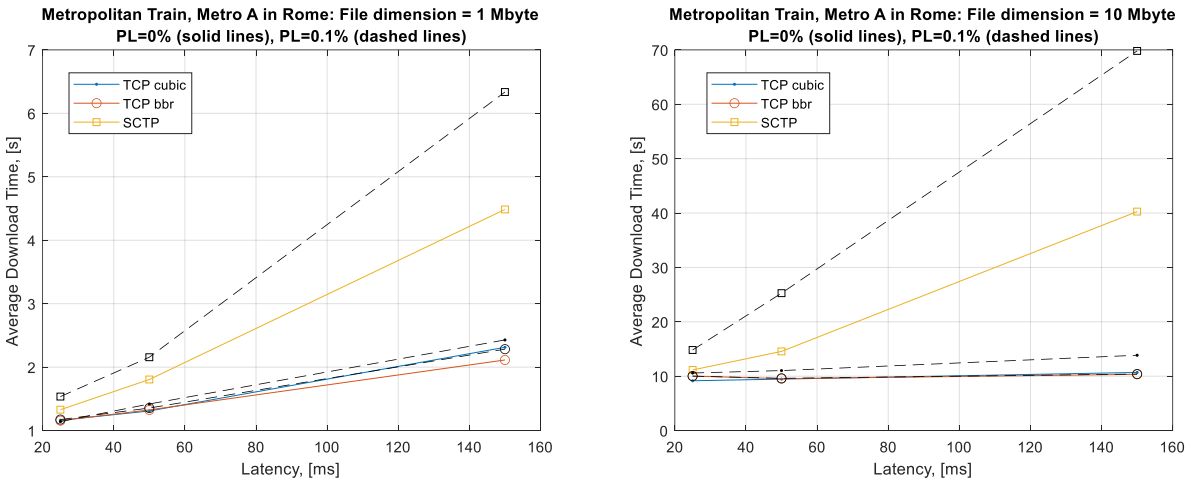


File 10 Mbyte

6.3.3 Metro rail line

In Figure 32 we reported the average Download Time for TCP cubic, BBR and SCTP as a function of the channel latency (from 25 ms to 150 ms) for Metro train line. We considered both lossless case (solid and colored curves) and case of PL = 0.1% (dashed and black curves).

Figure 32. Average Download Time for TCP cubic, BBR and SCTP vs channel latency for Metropolitan train line. File dimensions: (a) 1 MB and (b) 10 MB.



(a) File 1 Mbyte (b) File 10 Mbyte

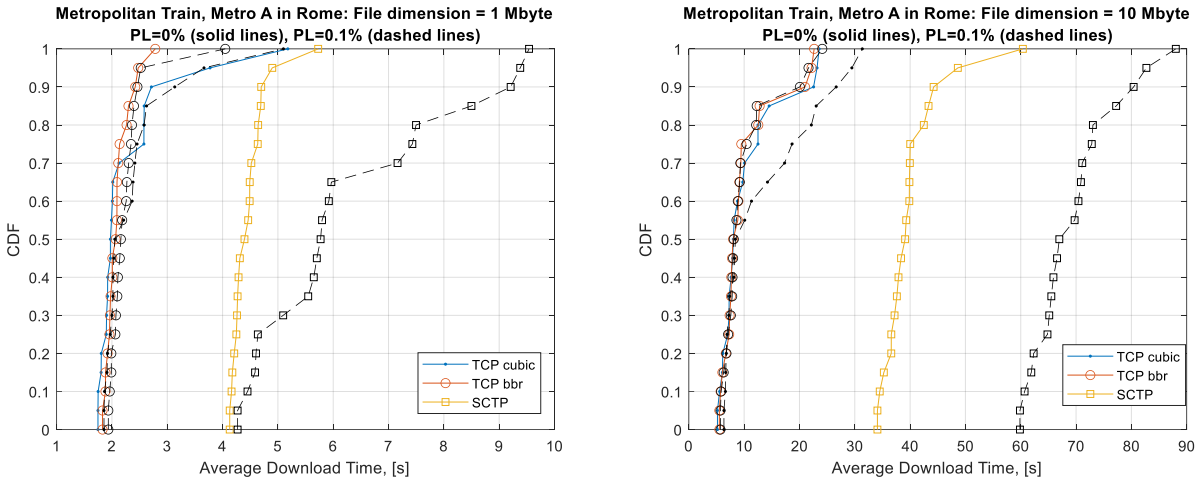
As expected, in case of limited file dimension (i.e., 1MB) DT for this rail line is shorter (of about 1-2 s for channel latency lower than 50 ms, both for PL = 0% and PL = 0.1%) w.r.t. the High-Speed line and Regional line due to higher bit rate available on this line. Better performance w.r.t. High-Speed/Reginal cases are experienced even for 150 ms channel latency for TCP cubic and BBR (about 2-3 s). Instead, SCTP degrades to 6.5 s for PL =0.1% which is similar to DT values of High-Speed/Reginal cases (i.e., about 7-8 s).

Concerning the transfer of large file (i.e., 10 MB), DT for TCP cubic and BBR is about 10 s (slightly degradation is for TCP cubic at PL = 0.1). On the contrary, SCTP has good performance only for channel latency of 25 ms, while it rapidly degrades at 150 ms, reaching values (i.e., 40 s for PL = 0% and 70 s for PL = 0.1%) not far from those of High-Speed/Reginal cases in the same channel conditions (45-50 s for PL = 0% and 73-77 s for PL = 0.1%).

Also in the Metropolitan case, for completeness of the analysis we reported in Figure 33 the CDF of the Download Time for TCP cubic, TCP BBR and SCTP for channel latency of 150 ms: two file dimensions of 1 MB and 10 MB. In the figure we reported the case PL = 0% (solid and colored curves) and the case of PL = 0.1% (dashed and black curves).

CDF better explain averaged data of Figure 32. It is evident that SCTP has higher DT both for small and large file dimension, above all when the channel is lossy.

Figure 33. CDF of Download Time for TCP cubic, BBR and SCTP for Metropolitan train line (channel latency = 150 ms). File dimensions: (a) 1 MB and (b) 10 MB



(a) File 1 Mbyte (b) File 10 Mbyte

6.4 Variable Bit Rate

In this section we analyze the performance of variable bit rate sources. Two VBR sources are considered: the first has limited peak bit rate of 80 kbit/s, while the second has a peak bit rate equal to the maximum available capacity in each considered rail line i.e., 5.2 Mbit/s for High-Speed and Regional lines and 54 Mbit/s for Metropolitan line. For VBR sources, we considered only the case of LTE with a band of 1.4 MHz.

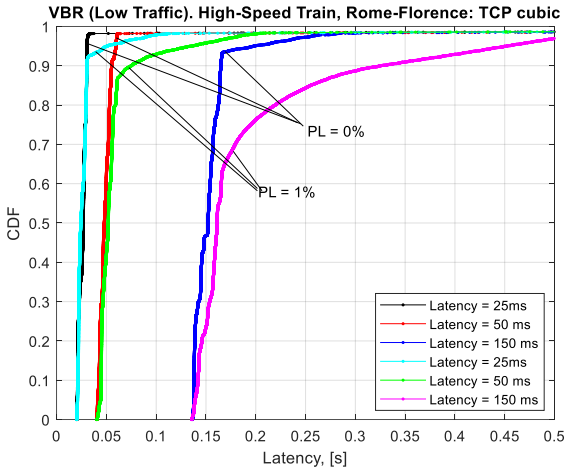
6.4.1 High-Speed rail line

In Figure 34, we report the CDFs for the packet latency and TH for the TCP cubic in the case of transmission channel with latency of 25 ms, 50 ms and 150 ms and for a PL = 0% and 1%. The high-speed line is considered. In Figure 34a (latency) and Figure 34b (throughput) we report the low-rate traffic source, while Figure 34c (latency) and Figure 34d (throughput) we report the high-rate traffic source.

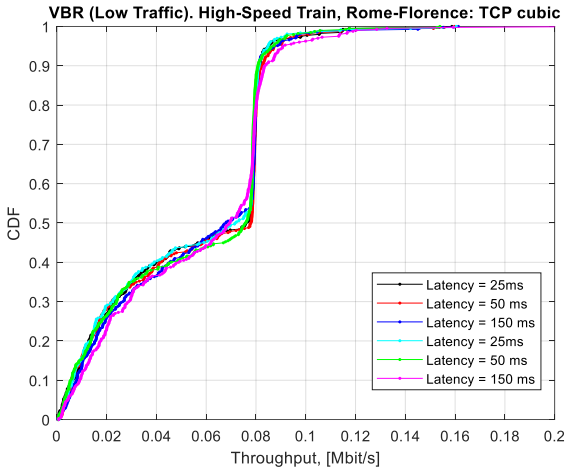
In case of low-rate VBR traffic (see Figure 34a), TCP cubic performance are limited by the transmission channel latency. A slight dependence on channel packet loss is observed. As expected, the TH is always close to 80 kbit/s and it is practically invariant to the channel latency and PL (see Figure 34b). This is due to the fact that transmission channel capacity, even if variable is always much higher that the VBR rate.

Figure 34. CDF for TCP cubic in terms of latency (a), (c) and throughput (b) (d) at Low-Rate traffic and High-Rate traffic for High-Speed train line

Low-Rate Traffic (80 kbit/s)

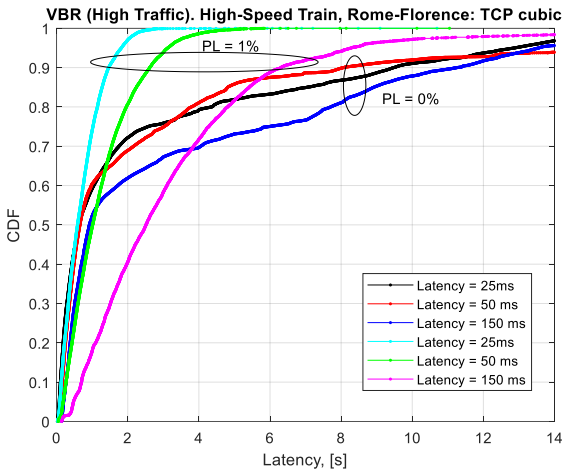


(a)

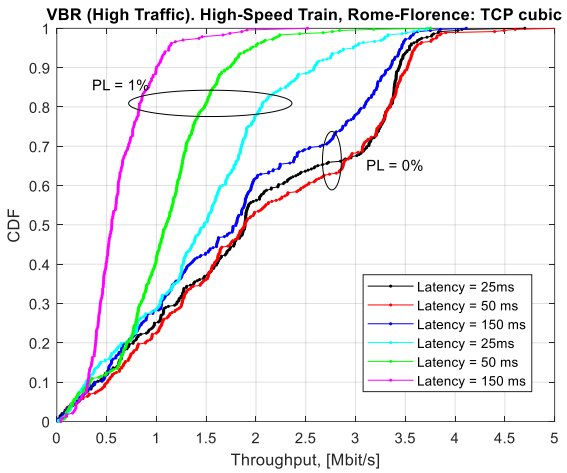


(b)

High-Rate Traffic (5.2 Mbit/s)



(c)



(d)

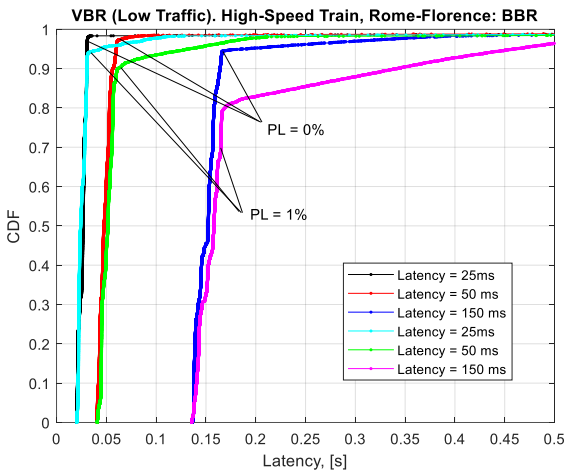
In case of high-rate traffic performance are similar to those obtained for the CBR source. From Figure 34c, the increase of latency is due to the variable transmission channel capacity (see previous paragraphs for CBR sources). Data are buffered and delayed by the TCP entity in order to avoid packet lost. In case of PL = 1%, experienced latencies are lower than those experienced for PL = 0% since it causes a reduced congestion window and lower traffic transmitted. This effect is clear also in Figure 34d, where TH for PL=0% is practically independent by the channel latency and equal to the average bit rate of the channel capacity of the High-Speed rail line. In case of PL = 1%, TH reduces due to the action of the congestion control algorithm which reduces the transmission window.

It should be noted that values of TH for VBR high-rate traffic are slightly lower than those for CBR (see Figure 17). In fact, the rate of the VBR source is, on average, lower than the rate of the CBR source, which transmits always at its maximum capacity and not only in the ON period.

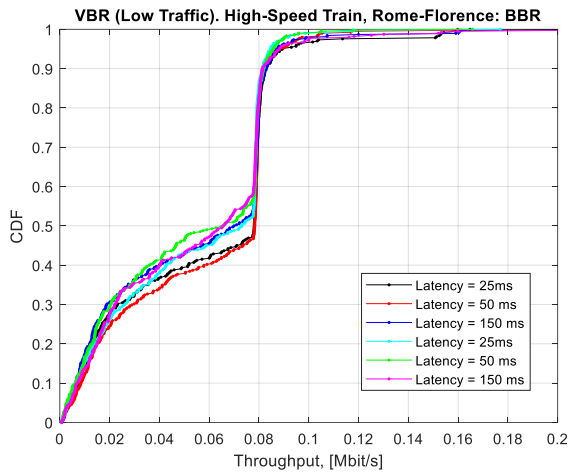
In Figure 35, we reported the CDF of the TCP BBR for the experienced latencies and throughput for latencies in the channel of 25 ms, 50 ms and 150 ms and for a PL = 0% and 1%, in case of High-Speed line. In Figure 35a (latency) and Figure 35b (throughput) we reported the low-rate traffic source, while Figure 35c (latency) and Figure 35d (throughput) we reported the high-rate VBR traffic source.

Figure 35. CDF for BBR in terms of latency (a), (c) and throughput (b) (d) at Low-Rate traffic and High-Rate traffic for High-Speed train line

Low-Rate Traffic (80 kbit/s)

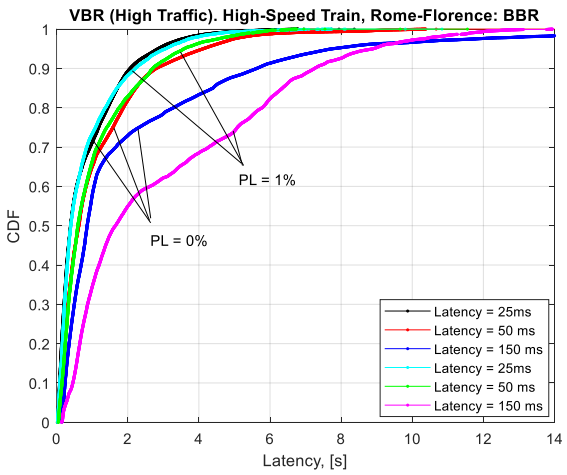


(a)

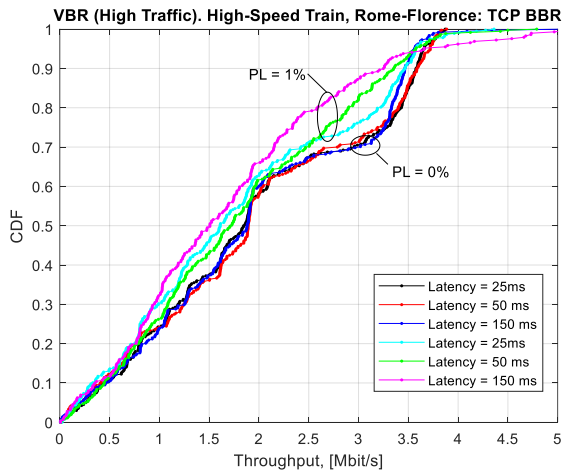


(b)

High-Rate Traffic (5.2 Mbit/s)



(c)



(d)

Similar values of TCP cubic are obtained for BBR in case of low-rate traffic source. Experienced latencies are 25 ms, 50 ms and 150 ms (see Figure 35a), while the TH is 80 kbit/s (see Figure 35b).

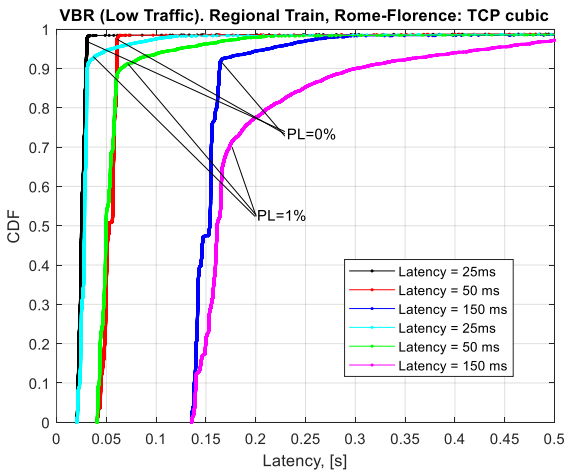
Concerning high-rate source, the behavior of BBR in case of VBR is similar to that of a CBR source. However, TH show some variations with PL and latency with respect to the reference curve at PL=0%. The TCP BBR is still able to rapidly adapt its congestion window to the available channel capacity variation due to the train movement along the rail line and to presence of lost packets in its flow. Differently from CBR, this could be due to the variability with time of data source in addition to the changing of the available channel capacity i.e., when the TCP BBR enlarges the transmission window, data from the source could not be available for transmission so to reach the maximum available receiver TH.

6.4.2 Regional rail line

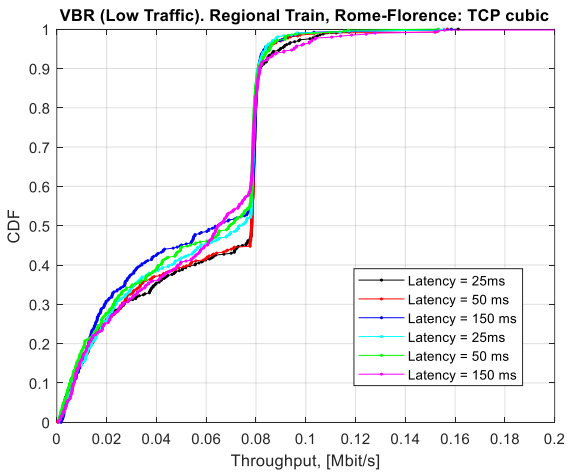
In Figure 36, we report the CDFs of the latencies and throughput for TCP cubic. Even in this case latencies channel latency has been set to 25 ms, 50 ms and 150 ms and for a PL = 0% and 1%, in case of High-Speed line. In Figure 36a (latency) and Figure 36b (throughput) we report the low-rate traffic source, while Figure 36c (latency) and Figure 36d (throughput) we report the high-rate traffic source.

Figure 36. CDF for TCP cubic in terms of latency (a), (c) and throughput (b) (d) at Low-Rate traffic and High-Rate traffic for Regional train line

Low-Rate Traffic (80 kbit/s)

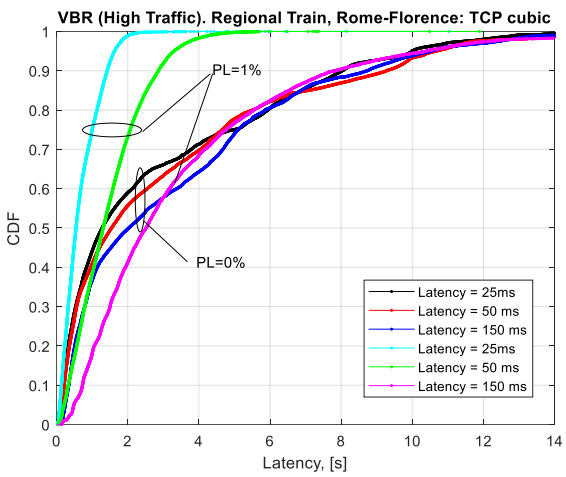


(a)

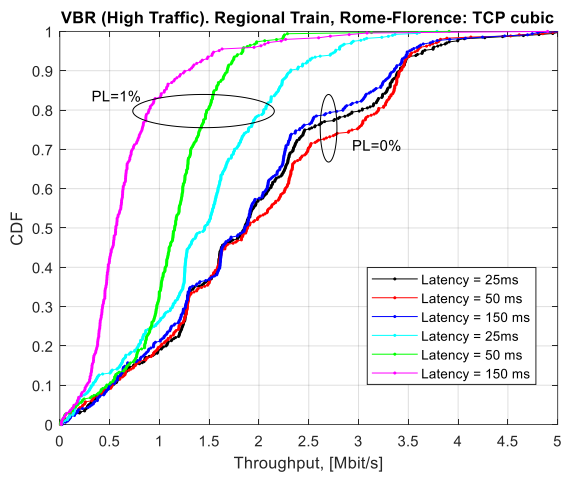


(b)

High-Rate Traffic (5.2 Mbit/s)



(c)

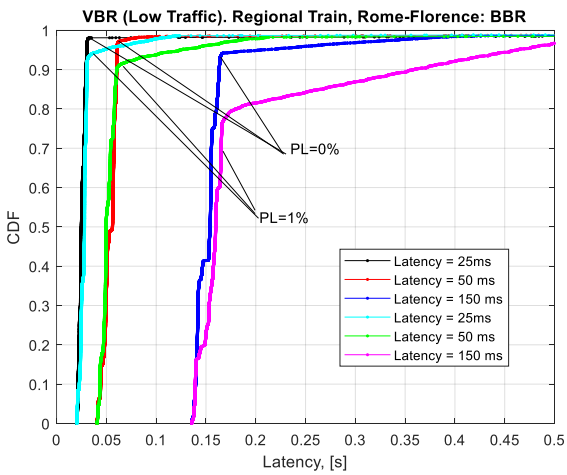


(d)

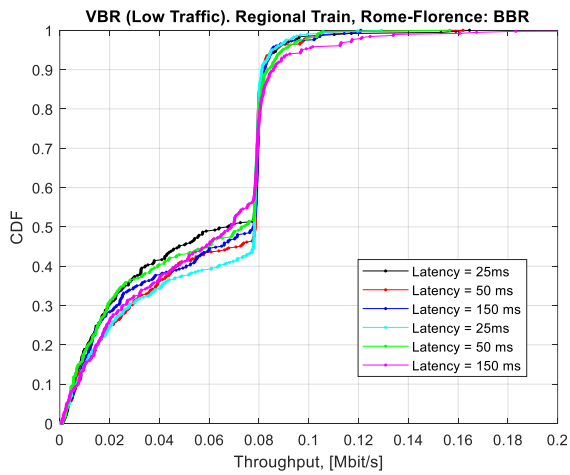
In Figure 37, we reported the CDF for latency and throughput for the TCP BBR in the same channel used for TCP cubic and for a PL = 0% and 1%, in case of High-Speed line. In Figure 37a (latency) and Figure 37b (throughput) we report the low-rate traffic source, while Figure 37c (latency) and Figure 37d (throughput) we report the high-rate traffic source.

Figure 37. CDF for BBR in terms of latency (a), (c) and throughput (b) (d) at Low-Rate traffic and High-Rate traffic for Regional train line

Low-Rate Traffic (80 kbit/s)

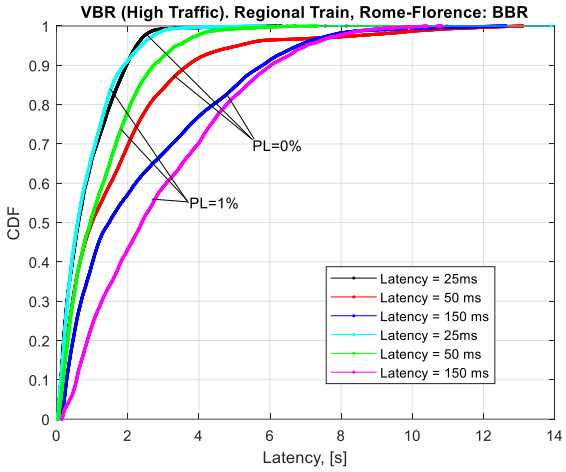


(a)

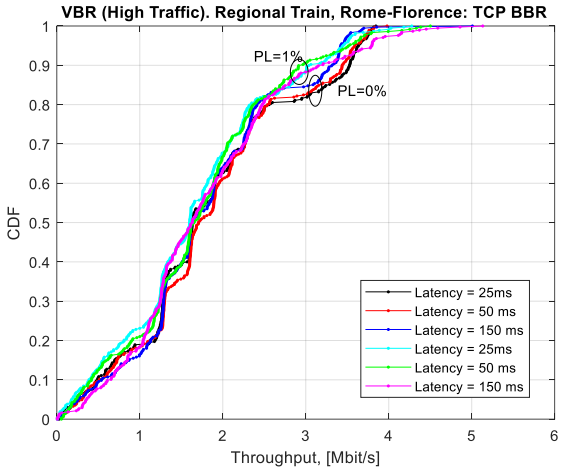


(b)

High-Rate Traffic (5.2 Mbit/s)



(c)



(d)

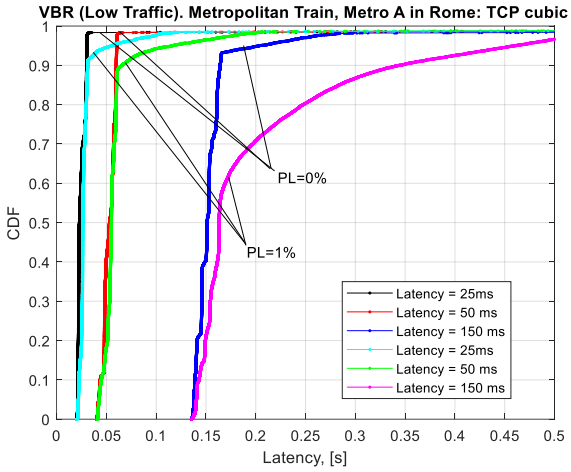
As expected from results in Figure 36 and Figure 37, comments on the Regional line results are quite similar to those of High-Speed line. Differences in latency and throughput are small and due to the different duration of the time intervals the train stays in the MCS regions.

6.4.3 Metropolitan rail line

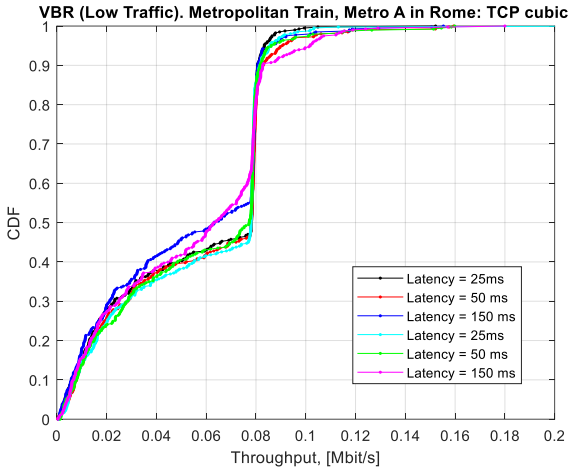
Following the organization of the results for VBR in High-Speed and Regional lines, we reported in Figure 38 results of TCP cubic and in Figure 39 results of BBR for Metropolitan line. For TCP cubic and BBR we reported the CDF latency and CDF throughput for low-rate traffic and high-rate traffic for latencies in the channel of 25 ms, 50 ms and 150 ms and for a PL = 0% and 1.

Figure 38. CDF for TCP cubic in terms of latency (a), (c) and throughput (b) (d) at Low-Rate traffic and High-Rate traffic for Metropolitan train line

Low-Rate Traffic (80 kbit/s)

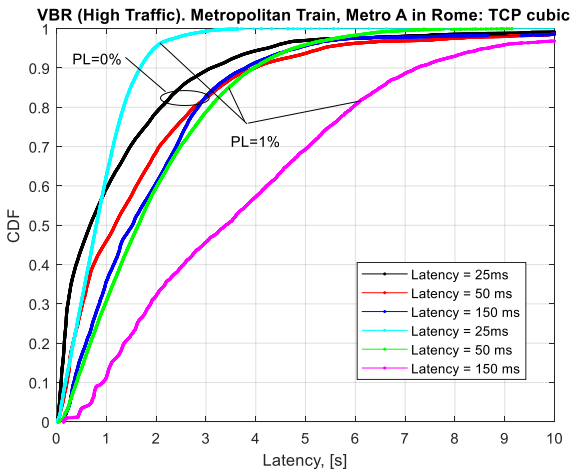


(a)

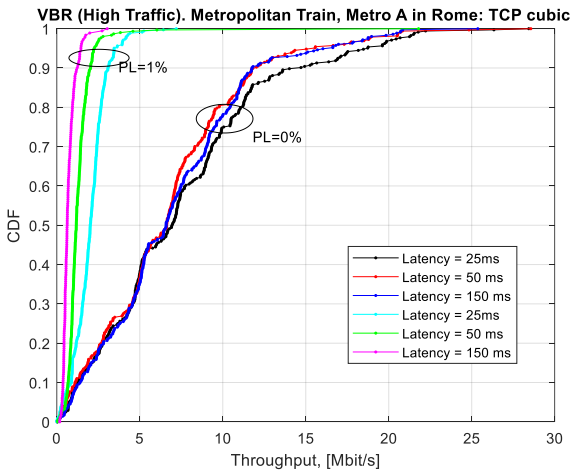


(b)

High-Rate Traffic (5.2 Mbit/s)



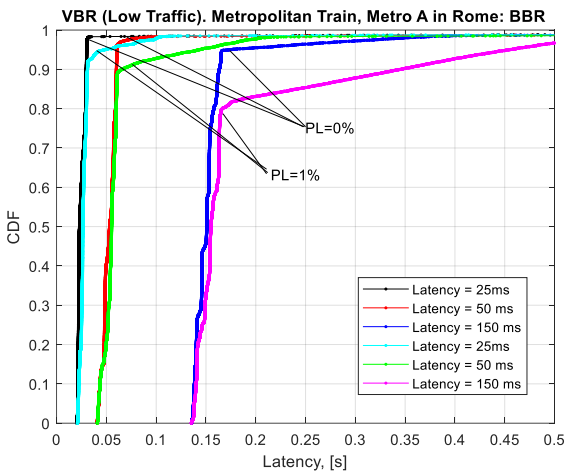
(c)



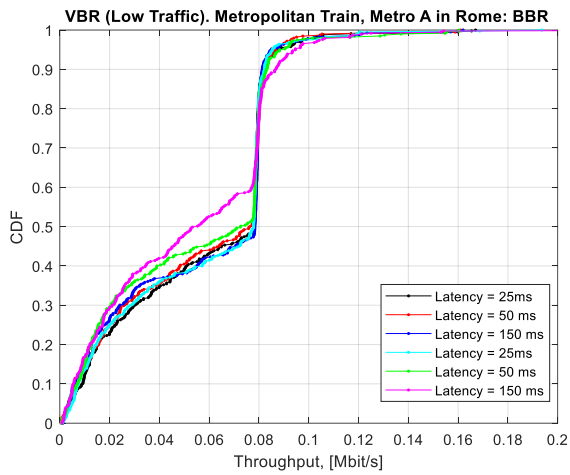
(d)

Figure 39. CDF for BBR in terms of latency (a), (c) and throughput (b) (d) at Low-Rate traffic and High-Rate traffic for Metropolitan train line

Low-Rate Traffic (80 kbit/s)

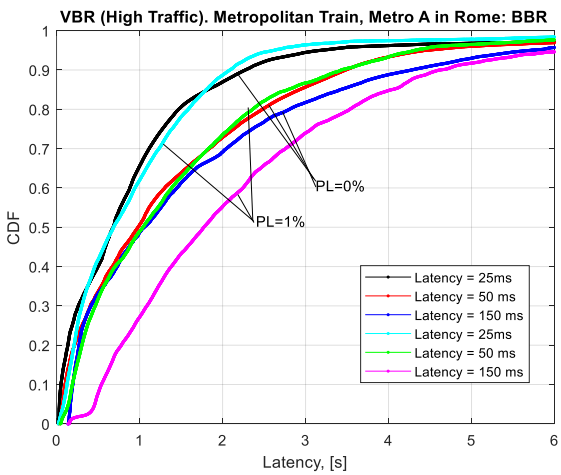


(a)

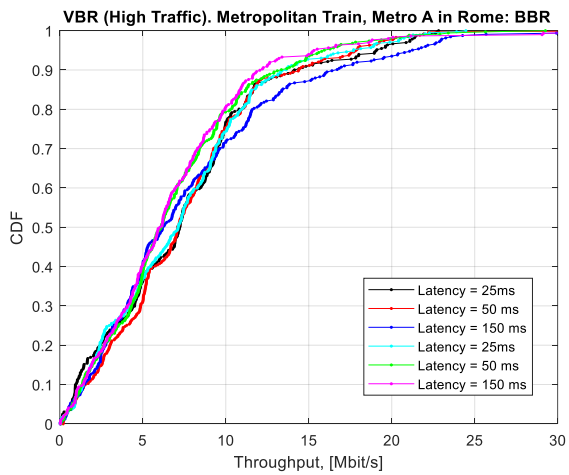


(b)

High-Rate Traffic (5.2 Mbit/s)



(c)



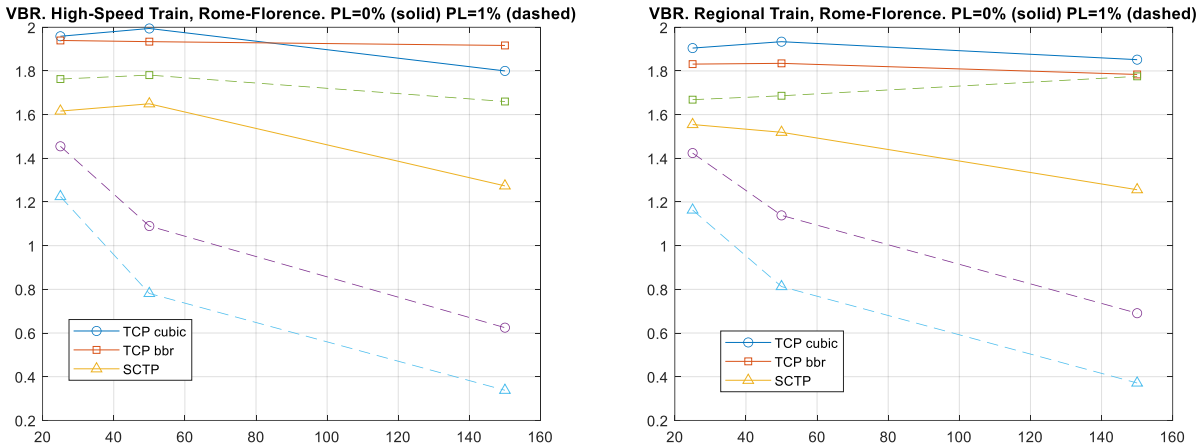
(d)

Results in Figure 38 and in Figure 39 show that the behavior of TCP cubic and BBR is similar also in the Metropolitan line as for Roma-Firenze lines. Obviously, the TH is higher in this case due to larger Wi-Fi bandwidth. Also for VBR, BBR works well in case in which the channel has a packet loss rate not null.

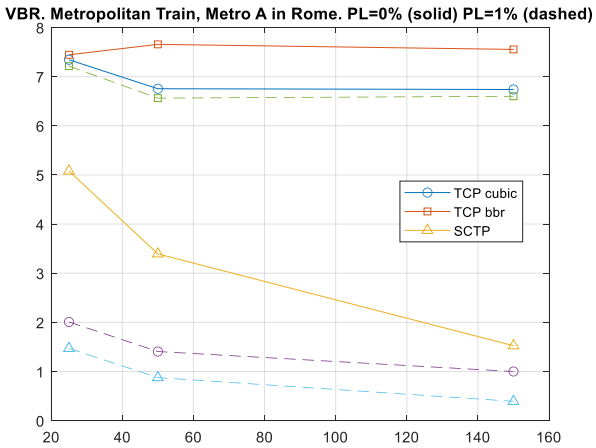
6.4.4 Average Throughput

To complete the analysis of VBR sources, in Figure 40 we report the average throughput as a function of the latency set in the channel in the emulator for TCP cubic, TCP BBR and SCTP. We report TH values in the three rail scenarios: High-Speed in Figure 40(a), Regional in Figure 40(b) and Metropolitan in Figure 40(c). Moreover, we considered lossless case (PL = 0%) in the solid lines, while lossy case (PL = 1%) in the dashed lines.

Figure 40. Average Throughput vs channel latency for VBR



(a) (b)



(c)

TCP cubic and BBR have a constant TH with respect to the channel latency for PL = 0%. It is equal to about 1.9 Mbit/s for High-Speed and Regional line, while for Metro line it is about 7.5 Mbit/s due to higher channel capacity provided by Wi-Fi.
SCTP has lower TH both when latency increases and when packet loss probability increases. Independently of the available channel capacity, as expected when the channel is lossy (i.e., PL = 1%) TCP cubic reduces its TH to about 0.5-1 Mbit/s for latency of 150 ms.

6.5 Short conclusions on TCP-Cubic, TCP-BBR and SCTP

From previous results it can be observed that the most important channel parameter strongly influencing the transport protocol behavior is packet loss. In the case of small or zero packet loss both protocols TCP cubic and TCP BBR are able to track the available transmission channel capacity in all the considered scenarios at the expense of increased packet latency. As shown for example in Figure 19b in the case of PL=0% the two protocols show, in practice, the same behavior.

The increase of PL to 1% show that in every scenario the TCP BBR is able to track the available transmission channel capacity since the CDF of the TH at PL=1% is very similar to that obtained at PL=0%. As expected, this behavior is practically independent of the considered rail scenario even though TH can be slightly higher in the mainline due to reduced percentage of time the train remains in the area characterized by MCS with reduced modulation efficiency. But this result does not depend on the features of the considered transport protocol.

Moreover, due to the congestion control algorithm adopted in the Linux OS SCTP implementation (single stream), we have observed that this protocol tries to save latency and this may render SCTP particularly interesting for signaling services characterized by low data rates but requirements on latency.

Finally, with the considered high bit rates traffic sources the behavior of the three protocols is almost invariant passing from CBR to VBR.

6.6 Coexistence of TCP Cubic, TCP BBR and SCTP

In this section we report the results concerning the performance analysis of one traffic source transmitting two separate flows:

a. The first flow always adopts TCP BBR
b. The second flow can use TCP Cubic or SCTP

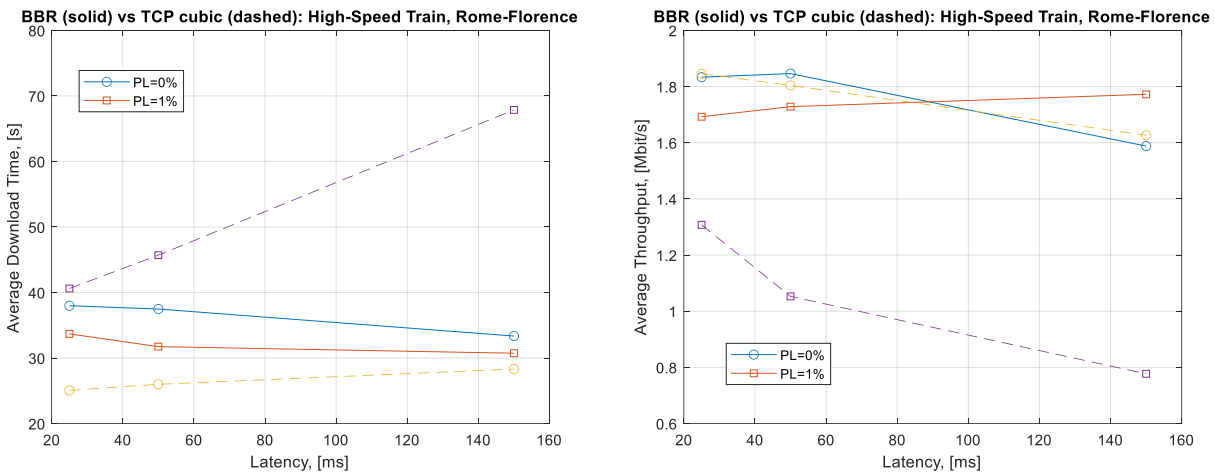
Both flows are directed to the same destination. The main goal of this analysis is to investigate on the presence of possible coexistence issues between the two flows.
Performance is analyzed in terms of Download Time and achievable receiver throughput. Considering the results presented in the previous Section we restrict our analysis to the High-Speed and Regional line.
In all cases results have been obtained considering two equal rates CBR sources such that the sum of the two bit rates is equal to the maximum available transmission capacity i.e. 5.2 Mb/s for the considered LTE radio access technology with band 1.4MHz.

6.6.1 High-Speed rail line

6.6.1.1 Results on TCP BBR and TCP cubic

In Figure 41a we report the average Download Time for BBR (solid line) and for TCP cubic (dashed line), as a function of the channel latency (from 25 ms to 150 ms) for High-Speed train line. In Figure 41b, we reported the average throughput for BBR (solid line) and TCP cubic (dashed line) as a function of the channel latency (from 25 ms to 150 ms) for High-Speed train line.

Figure 41. Coexistence between BBR (solid line) and TCP cubic (dashed) vs channel latency for High-Speed train line: (a) Average Download Time; (b) Average Throughput.



(a) (b)

The TCP BBR achieve similar TH performance for different channel latencies and PL = 0% and PL = 1%. A download time of about 32-38 s and a throughput of about 1.6-1.8 Mbit/s is achieved. TCP cubic has a lower download time w.r.t. TCP BBR for lossless channel of about 26-28 s and a similar TH w.r.t. TCP BBR. However, TCP cubic performance significantly degrades when the channel is

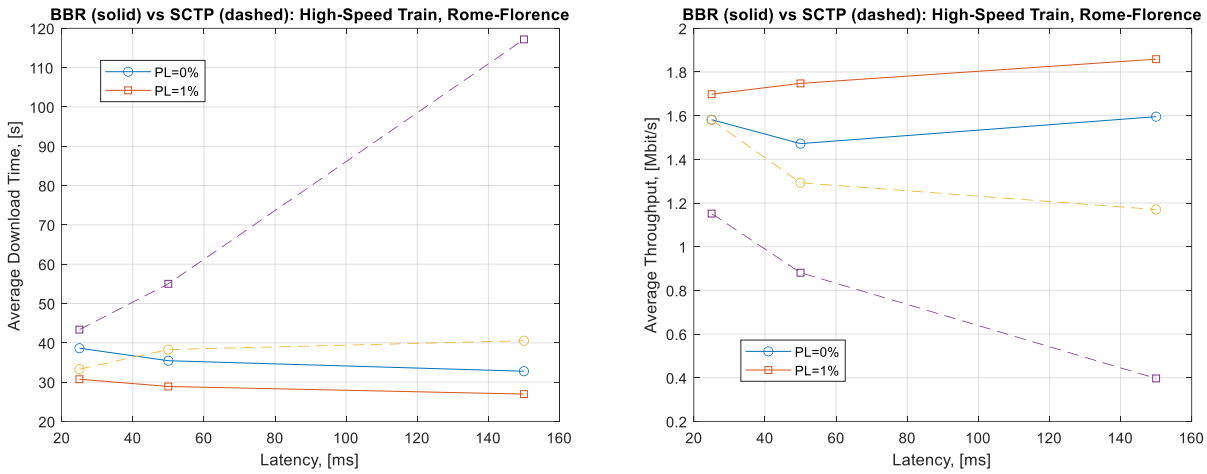
lossy showing a DT of about 70 s and TH of 0.8 Mbit/s.

This fact can be explained observing that the TCP BBR tries to use all the available capacity on the channel even at the expense of other flows. Assuming for example that at the start of transmission TCP Cubic and TCP BBR occupy a similar channel band, in the case of non-zero PL the TCP BBR congestion control is faster than the TCP Cubic to acquire transmission bandwidth and may lead the TCP Cubic to starve leading to a TH reduction for the TCP cubic.

6.6.1.2 Results on TCP BBR and SCTP

To assess coexistence between BBR and SCTP, in Figure 42a and in Figure 42b we report the average Download Time and the average throughput for the two transport protocols, as a function of the channel latency (from 25 ms to 150 ms) for High-Speed train line.

Figure 42. Coexistence between BBR (solid line) and SCTP (dashed) vs channel latency for High-Speed train line: (a) Average Download Time; (b) Average Throughput.

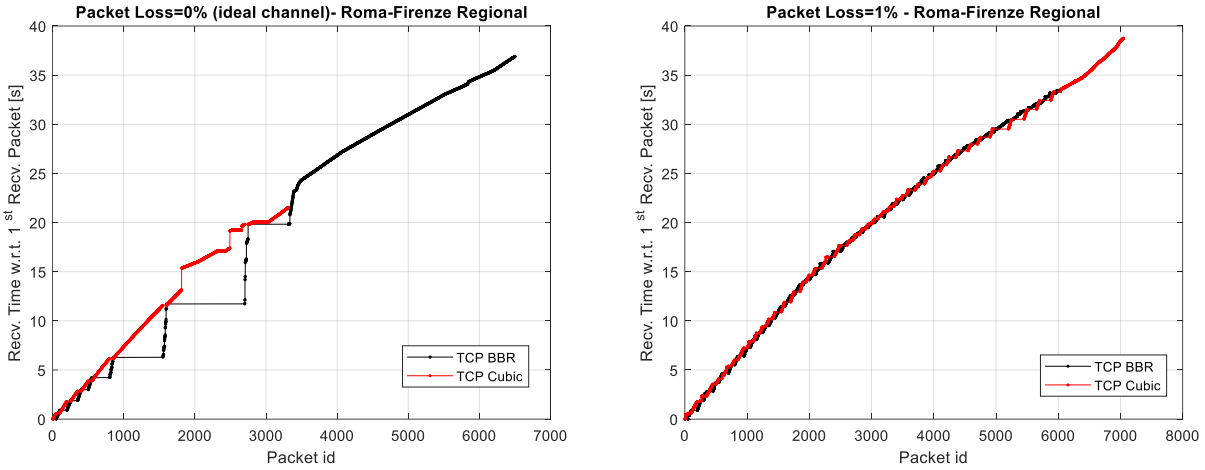


(a) (b)

Also with an SCTP flow, the TCP BBR has similar performance for different latencies and PL = 0% and PL = 1%, showing in this case a download time of about 28-38 s and a throughput of about 1.5-1.8 Mbit/s. In this case, SCTP has worse performance w.r.t. TCP BBR, showing a download time ranging from 32 s to 40 s for lossless channel and a TH of 1.2-1.6 Mbit/s. As for TCP cubic, SCTP performance significantly degrades when the channel is lossy showing a DT of about 120 s and TH of 0.4 Mbit/s. This is always due to the greedy characteristic of the congestion control algorithm of TCP BBR which in the case of PL tends to rapidly occupy all the available transmission channel bandwidth, thus starving the other non-TCP BBR flows.

From results in Figure 41a we observe that in the case of PL=0% the TCP cubic may work better that the TCP BBR.
This fact can be explained looking at the results in Figure 43, where we indicate the receiving time of packets from TCP BBR and TCP cubic.

Figure 43. example of transmitted packets of TCP cubic and TCP BBR during their coexistence.



From the results in Figure 43a (PL=0%), it seems that at the start the TCP Cubic fills the transmission buffer, since its congestion window is not reduced because no packet loss occurs. Then, the TCP BBR seems to be unable to insert packets in the common transmission buffer (due to coexistence) because the TCP Cubic continues its transmission and forcing the TCP BBR to delay packet transmissions. Only when the TCP cubic has concluded its file transmission, the TCP BBR can finalize its file transfer.

In the lossy case (PL=1%), the TCP Cubic may (sooner or later) need to reduce its congestion window (when a packet loss occurs) so that TCP BBR has a chance to occupy the transmission buffer. The situation is then depicted in Figure 43b.

7. Conclusions

This deliverable responds to the objective a in workstream 2, and it is devoted to identification of the appropriate transport protocols ensuring the required communication and characteristics capabilities in the application development stage.
The main objective of emulation activities carried out in Task 3.4 of the AB4Rail project consists in the performance assessment of the selected transport protocols for the two different types of sources in the considered scenarios, thus investigating which protocol performs better in the different environments (i.e., rail lines).

To assess transport protocol performances, we have used the software emulator developed in the Task 3.3 and which allows reproducing the behavior of the communication bearers at IP protocol level. Then, in this case we used it to evaluate the performance of the transport protocols: TCP cubic, TCP BBR, UDP and SCTP.

The emulator allows to model the variations with time of the typical packet impairments characterizing the IP layer link such as: bandwidth, latency and packet loss rate. In order to emulate the realistic time variation of packet impairments at IP level we have first identified three realistic rail scenarios and we have assumed the entire rail line is covered by LTE radio technology. To account for the variations with time of the available transmission capacity along the track in our emulation we have also considered the variations of the available LTE modulation coding scheme (MCS) that is selected by the on-board terminal in accordance with its distance from the eNB.

The three railway lines we have considered are:

- The Roma – Firenze high-speed line, to evaluate the generic mainline environment
- The Roma – Firenze regional line, to evaluate the regional/freight line type
- The metro of Rome, to evaluate the metro/urban line

Transport protocol performance have been evaluated in terms of the statistics (i.e., cumulative distribution function, mean, standard deviation etc.) of:

1. One-way transmission latency i.e., the time required from one packet enqueued in the transmission buffer to reach the receiver;
2. The receiver data rate which is referred as the receiver throughput or more simply throughput;
3. The download time for CBR traffic sources generating a finite number of packets.

In the result group, we evaluate the performance by considering each transport protocol separately i.e., only one active transport protocol on the link for the three different rail scenarios listed above.
In the case of small or zero packet loss both protocols TCP cubic and TCP BBR are able to track the available transmission channel capacity in all the considered scenarios, showing in practice the same behavior, at the expense of increased packet latency.
In case of lossy channel, results show that in every scenario the TCP BBR is able to track the available transmission channel capacity, since the CDF of the TH at PL=1% is very similar to that obtained at PL=0%. As expected, this behavior is practically independent of the considered rail

scenario even though TH can be slightly higher in the mainline due to reduced percentage of time the train remains in the area characterized by MCS with reduced modulation efficiency. But this result does not depend on the features of the considered transport protocol.
Results on Linux OS SCTP implementation (single stream) show that this protocol tries to save latency, due to the adopted congestion control algorithm. This may render SCTP particularly interesting for signaling services characterized by low data rates but requirements on latency.
It is worth noting that the behavior of the three protocols is almost invariant passing from CBR to VBR, with the considered high bit rates traffic sources.

In the second part of the result section, we have analyzed the coexistence between pairs of transport protocols sharing the same transmission tunnel. Performance comparison in this important scenario has been carried out by comparing the latency and the achievable throughput of each one of the two protocols. In case of coexistence of TCP BBR and TCP cubic, both perform similarly in terms of average download time and experienced throughput for lossless channels, while TCP BBR acquires more bandwidth in lossy channels. In case of TCP BBR and SCTP, TCP BBR outperforms SCTP both for lossless and lossy channels.

In Table 9 we summarize and comment the transport protocol(s) to be selected for each ACS application class. From results presented in previous Sections we observe that transport protocol performance are practically independent from the selected railway scenario (e.g., mainline and regional).

Table 9: Summary of transport protocols for ACS application classes

ACS Application class	Transport protocol	Note
Signaling	SCTP	SCTP experiences lower latency (even though throughput is also reduced). Reduced throughput is not a concern for signaling. An alternative to SCTP, TCP BBR can be used in case TH is a requirement.
Critical Voice	TCP BBR, UDP	Similar behavior when channel capacity is higher than application requirements (such as voice in wideband channels). In low PL conditions TCP cubic could also be used. Furthermore, VoIP services use TCP to establish an initial connection and then UDP for purposes of streaming audio as rapidly as possible to the destination with minimal overhead. If a few packets get lost (i.e., low PL condition) voice codecs are designed to be as tolerant as possible to this kind of occasional data loss.
Critical Data	TCP BBR	It experiences higher throughput above all in lossy communication channel
Critical Video	UDP, TCP BBR	Even though in lossy environments UDP should be avoided, typically for video

		transmissions UDP is not used alone; in fact, the typical protocol used for video transmissions used in conjunction with RTP/RTPC protocols. TCP BBR could be used because is less sensitive to loss in terms of latency.
Non-critical Data	TCP BBR, TCP cubic	Most of the traffic uses TCP and both TCP variants could be used provided coexistence issues are taken into account.
Internet Connectivity	TCP BBR, TCP cubic, UDP	Most of the traffic uses TCP. UDP is considered for some applications protocols that use it as default transport protocol.

The results presented in this Deliverable will be used for the next activities foreseen in task 3.5 focusing on the analysis of: QUIC protocol, the application and transport protocols and secure protocols.

8. References

[1] A. Vizzarri, F. Vatalaro, R. Giuliano, F. Mazzenga, "IP Emulator and scenarios definition", Deliverable D3.3 vers. 0.1, AB4RAIL project, 31 Jul. 21.

[2] [D-IGT] A. Botta, A. Dainotti, A. Pescapè, "A tool for the generation of realistic network workload for emerging networking scenarios", Computer Networks (Elsevier), 2012, Volume 56, Issue 15, pp 3531-3547.]

[3] Shift2Rail, X2Rail-1 D3.3 "Communication_system_Specification_and_Technology Guideline – *Annex Guidelines for Choice of Technology*", 2018.

[4] Electronic Communications Committee (ECC), "ECC Report 294: Assessment of the spectrum needs for future railway communications", 2019.

[5] T. Arbuthnot, "What are Thresholds for Good and Poor Network Packet Loss, Jitter and Round Trip Time for Unified Communications?", May 17, 2018. Available on line at: https://tomtalks.blog/what-are-thresholds-for-good-and-poor-network-packet-loss-jitter-and-round-trip-time-for-unified-communications/

[6] AB4Rail Project Deliverable 2.2, "Assessment of ABs benefits challenges and impact on infrastructure", 2021

[7] Wikipedia, "Ferrovia Firenze-Roma (direttissima)". Available on line at https://it.wikipedia.org/wiki/Ferrovia_Firenze-Roma_(direttissima),

[8] Wikipedia, Ferrovia Firenze-Roma, Available on line at https://it.wikipedia.org/wiki/Ferrovia_Firenze-Roma

[9] J. Wang, Y. Zheng., Et al., "An Active-Passive Measurement Study of TCP Performance over LTE on High-speed Rails", MobiCom '19: The 25th Annual International Conference on Mobile Computing and Networking, August 2019 Article No. 18, Pages 1–16, https://doi.org/10.1145/3300061.3300123

[10] H. Holma, A. Toskala, T. Nakamura, "5G Technology: 3GPP New Radio", First Edition, John Wiley & Sons Ltd., 2020

[11] H. Holma, A. Toskala, J. Reunanen, "LTE Small Cell Optimization: 3GPP Evolution to Release 13", John Wiley & Sons Ltd, 2016.

[12] Berec, "Draft BEREC Guidelines on Very High Capacity Networks", BoR (20) 47, 5 Mar. 2020.

[13] Shift2Rail, X2Rail-1 D3.3 "Communication system Specification and Technology Guideline", 2019.

[14] Farooq, J., & Soler, J. (2017). Radio communication for Communications-Based Train Control (CBTC): A tutorial and survey. IEEE Communications Surveys & Tutorials, 19(3), 1377-1402. https://doi.org/10.1109/COMST.2017.2661384

[15] K. R. Parsons, "MCS Table and How To Use it", available at https://wlanprofessionals.com/mcs-table-and-how-to-use-it/

[16] European Commission, S2R-OC-IP2-02-2020 call, "Study on alternative bearers and on communication protocols", 2020. Available on line at: https://ec.europa.eu/info/funding-tenders/opportunities/portal/screen/opportunities/topic-details/s2r-oc-ip2-02-2020

[17] J. Daigle and J. Langford, "Models for analysis of packet voice communications systems", in IEEE Journal on Selected Areas in Communications, vol. 4, no. 6, pp. 847-855, Sep. 1986, doi: 10.1109/JSAC.1986.1146403.

[18] P. T. Brady, “A model for generating on-off speech patterns in two-way conversations,” Bell Syst. Tech. J., vol. 48, pp. 2445-2472, Sept. 1969.

[19] Electric Signalling Systems for Railways – Part 200: Safe Transmission Protocol According to DIN EN 50159 (VDE 0831-159)

[20] M. Mathis, et al., “The macroscopic behavior of the TCP congestion avoidance algorithm.” ACM SIGCOMM Computer Communication Review, Vol. 27, Is.3, 1997, pp. 67-82.

[21] E. M. M. De Andrade, “SECURITY FOR 5G COMMUNICATIONS IN CRITICAL SYSTEMS”, Ph.D. Dissertation, June 2020, Universidade de Coimbra (Portugal), available online (last accessed 27 April 2022)

[22] M. Heinrich, “Security Engineering in Safety-critical Railway Signalling”, Ph.D. Dissertation, December 2020, Technische Universität Darmstadt, available online (last accessed 27 April 2022)

Appendix

The reader is referred to [19] for the details concerning the operations of the RaSTA protocol. The performance of the RaSTA protocol can be analyzed under different assumptions. In particular, if we assume i) ideal channel transmission conditions (i.e. no packet loss), ii) no additional delay of packets due to queueing and iii) one-way data traffic from the source to the recipient, when considering the transmission procedures in [19] under the assumptions i)-iii) we have observed, by analytic calculations, that if the time required for receiving the acknowledge message from the recipient is larger than the round trip time, the RaSTA throughput can be lower than that achievable with TCP protocol for the same of amount of transmitted data in the same operating conditions i)-iii) (see [20]). The throughput reduction is proportional to the ratio between the round-trip time and the acknowledge time interval. This result is not surprising since, as indicated in [21], [22] the RaSTA protocol implements several mechanisms to guarantee high level of integrity and safety for transmission of train-to-ground signaling messages rather than to achieve throughput. In the case of packet loss, considering the throughput calculation procedure in [20] the RaSTA throughput is further reduced and the amount of TH reduction is proportional to the packet loss probability (as for any other transport protocol).