INtelligent solutions 2ward the Development of Railway Energy and Asset Management Systems in Europe

D2.1 IN2DREAMS Services, Use Cases and Requirements

DUE DATE OF DELIVERABLE: 30/06/2018

ACTUAL SUBMISSION DATE: 09/07/2018

Leader/Responsible of this Deliverable: Valeria Bagliano – RINA Consulting

Reviewed: Y

Document status		
Revision	Date	Description
1	20/06/2018	First issue
2	05/07/2018	Second issue
3	09/07/2018	Final Version after TMT approval and Quality check

Project funded from the European Union's Horizon 2020 research and innovation programme		
Dissemination Level		
PU	Public	X
CO	Confidential, restricted under conditions set out in Model Grant Agreement	
CI	Classified, information as referred to in Commission Decision 2001/844/EC	

Start date of project: 01/09/2017 Duration: 24 Months

Executive Summary

The main objective of WP2 in In2Dreams is to provide a communication platform that will be able to interconnect a growing number of devices (metering devices, sensors and smartphones) located either on-board of trains or at the trackside with the Open Data Management (ODM) platform.

The objective of this deliverable, in the scope of Task 2.1 of WP2, is to describe the overall architecture of the integrated communication platform. The definition of the system architecture includes technical specifications and service requirements. The deliverable describes optical wireless communication systems and the deployment of the Open Data Management (ODM) platform in 5G. Control and management of the communication platform based on Software Defined Networking (SDN) is also described.

This deliverable also describes LiFi technology application for wireless broadband and IoT connectivity within the rolling stock and provides an overview of the use cases examined in In2Dreams and developed in agreement with the Shift2Rail project In2Stempo. Among the different railway use cases studied and the 3 use cases presented in this deliverable, the use case on commercial line operation (Use Case 1), has been used to demonstrate the In2Dreams integrated communication platform.

Finally, train measurement setups and data portfolio are provided to validate data analytics infrastructure and design for support of selected use cases.

Abbreviations and Acronyms

Abbreviation	Description
BBU	Base Band Units
CBTC	Communications Based Train Control
CDPI	Control to Data-Plane Interface
C-RAN	Cloud-RAN
DC	Data Centres
DMP	Data Management Platform
EMU	Electric Multiple Unit
EU	European Union
FH	Front Haul
FSO	Free Space Optics
GA	Grant Agreement
GPP	General-Purpose Processors
H2020	Horizon 2020 framework programme
ICT	Information and Communication Technology
ILP	Integer Linear Programming
IoT	Internet of Things
LPF	Low Pass Filter
LTE	Long-Term Evolution
MIMO	Multiple-Input Multiple-Output
MQTT	Message Queuing Telemetry Transport
MZM	Mach-Zehnder Modulator
M2M	Machine-to-Machine
NFV	Network Function Virtualization

Abbreviation	Description
NFVI	Network Function Virtualization Infrastructures
OCC	Operations and Control Centre
ODL	OpenDayLight
ODM	Open Data Management
PNF	Physical Network Functions
PTP	Precision Time Protocol
QoS	Quality of Service
REST	Representational State Transfer
RH	Radio Heads
SDD	Spectral Direct Decoding
SDN	Software Defined Networking
VNF	Virtual Network Functions
VOQ	Virtual Output Queues
JU	Shift2Rail Joint Undertaking
WDM	Wavelength Division Multiplex

Contents

List of Figures

List of Tables

1 Introduction

The objective of WP2 is to define, appropriately model and implement a dynamic network communication platform interconnecting a variety of on-board and trackside sensing/monitoring devices to the railway operations and support centre.

In the scope of WP2, and specifically Task 2.1 "Services, Use Cases, Overall Architecture", the present deliverable 2.1 describes the overall architecture of the integrated communication platform.

Inputs to this deliverable have been gather from the Shift2Rail project In2Stempo; specifically, the definition of the railway use cases presented in Chapter 2 have been defined in close cooperation with this project.

Deliverable D2.1 of In2Dreams is structured in the following chapters:

- Chapter 2 provides an overview of the 3 use cases selected to demonstrate In2Dreams project results;
- Chapter 3 presents the integrated communication platform, including optical wireless communications systems, the deploying of the ODM platform in 5Gand LiFi technology application;
- Chapter 4 describes the approach to validate the use cases through data analytic.

The outputs of this deliverable will be used in the other tasks of WP2 where on board and train to ground communication technologies will be deployed.

2 Use Cases

2.1 Overview and Definition of the Use Cases

2.1.1 Use Case 1 – Commercial Line Operation

This use case is based on smart metering for a commercially operated railway line under normal traffic conditions.

Main objective is the fine mapping of relevant energy flows, optimization of interaction between rolling stock and infrastructure, identification of energy efficiency potentials, implementation of energy management options.

Data collected thanks to smart metering make the infrastructure manager or the railway operator informed about energy behaviour, thanks to combining real time energy use and accurate billing. The consumer can act in two directions: energy saving, knowing the over consumption sources, and billing reduction, switching the consumption to less expensive period, when possible. Smart metering makes easy two steps, measure and analyse, which are very useful to introduce DMAIC (Define, Measure, Analyse, Improve, Control) method for energy management system, which supports and is in line with the ISO 50001 standard. The smart metering use case for a commercially operated railway line under normal traffic conditions (CO-OP) will be realized at Network Rail on a line between Redhill and Tonbridge, South of London.

This use case is the most similar case to the Reims tramway experimentation in terms of voltage range and of objectives of this smart metering application [57].

2.1.2 Use Case 2 – Stationing and maintenance facilities

This use case is dedicated to smart metering use case for stationing and maintenance facilities.

Main objective of this use case is to study the energy management options and identify energy efficiency potentials in this specific area, monitoring of relevant infrastructure, monitoring of rolling stock status, improvement of operational performance. Many electricity users can be connected to substations or catenaries for non-electric traction needs. The most known connections are:

- HV network for signalling;
- Auxiliary transformer for low voltage network (building, command-control, etc.);
- Train stations;
- Railway maintenance centre.

To achieve the benefits described above and to get billing procedure easier, the implementation of smart metering in those cases can be very useful and interesting. The smart metering use case for stationing and maintenance facilities (STM-OP) will be realized at the Eurotunnel infrastructure in France.

2.1.3 Use Case 3 – Infrastructure Monitoring

This use case will aim at smart metering to monitor infrastructure assets or traction network.

The main objective of this use case is to study the energy monitoring options for individual infrastructure assets as well as for the complete traction power network. Monitoring options will be identified, in order to provide continuous monitoring of electrical infrastructure, optimization of the infrastructure performance and the implementation of preventive maintenance. The smart metering use case for electrical infrastructure monitoring through the detection of electrical anomalies (IN-OP) will be realized at the Eurotunnel infrastructure, mainly inside the 50.45-kilometre rail tunnel linking United Kingdom with northern France.

Data needed are very much depending on the monitoring application and could be: current measures, voltage, temperature, etc. A higher frequency sampling could be required but by using a trigger point data acquisition efforts and data volume can be minimized.

2.2 Technical Constraints and Functional Requirements

Technical constraints and functional requirements of each use case are presented in Table 1.

Table 1: Technical Constraints and Functional Requirements

Use Case 1 – Commercial Line Operation	• The Network Rail experimentation site concerns a 20km-long line of two tracks between Redhill and Tonbridge; • power supply is 750 V DC, a very used voltage for British railway networks; • Rolling stock type is Class 377, a common British electric multiple-unit train (EMU) from Bombardier, 4 or 5 cars, with a maximum speed of 160 km/h, a car length of 20.4 m, and a power of 800 kW – 1200 kW with power supply by contact shoe.

Use Case 2 – Stationing and maintenance facilities	• The Eurotunnel site on the French site is composed of several railway buildings which composes the French terminal: ○ Depot: will concern the management of the energy consumption of the infrastructure and of the rolling stock; ○ Maintenance facilities: will concern the management of the energy consumption of the infrastructure and of the rolling stock, and the lighting; ○ Terminals for loading and unloading vehicles on the trains: will concern the management of the energy consumption of the infrastructure and of the rolling stock, and the lighting; ○ Railway Passenger building: will concern the lighting and the HVAC. • Power supply of the electrical infrastructure is 25 kV AC, and the buildings are supplied with high voltage, converted to common supply voltage for the lighting and the HVAC; • Rolling stock used on this field, which are passenger shuttles for cars and buses and truck-shuttles for the lorries are coupled with the same electrical locomotives, which a power of 5.6 MW for the oldest engine, and 7 MW for the updated ones.
Use Case 3 – Infrastructure Monitoring	• It concerns a 50.45-kilometre rail tunnel linking United Kingdom with northern France, composed of two tracks. Maximum train speed is 160 km/h; • Power supply of the infrastructure is 25 kV AC. In normal situation, one power supply substation coming from France is sufficient. • The Eurotunnel is operated with 4 different types of trains: ○ Truck shuttles (Eurotunnel rolling stock); ○ Passenger Shuttles (Eurotunnel rolling stock); ○ Eurostar trains: two kinds of trains are running: ▪ TGV TMST with a maximum power of 12 240 kW; ▪ Eurostar e320 with a maximum power of 16 000 kW; ○ Freight trains: many kinds of trains, circulating in the night – specific periods.

3. The Integrated Communication Platform

This chapter provides a high level view of the integrated communication platform that will be used to support the experimentation activities of the IN2DREAMS project. This infrastructure relies on the integration of an optical network solution with a heterogeneous wireless access network based on LTE, WiFi and LiFi technologies. An architectural framework inspired by the ETSI NFV/SDN reference model [3] enables control and management of this highly heterogeneous environment and facilitates end-to-end service orchestration with guaranteed Quality of Service. Preliminary theoretical studies and simulation results indicate that through tight integration of all networks, traditional technology barriers that prevent the deployment of cloud computing services in railway environments can be alleviated leading to significant improvements in terms of throughput, data density and energy efficiency.

3.1 ODM platforms in support of future railway systems

Information and Communication Technology (ICT) platforms for future railway systems are expected to support a wide range of applications with highly variable performance attributes covering both operational and end-user service requirements. These platforms are expected to offer services ranging from delay sensitive video to infotainment services, and from best effort applications to ultra-reliable ones such as M2M (Machine-to-Machine) communications. An important consideration in the design of these platform is the very high mobility of train transportation systems beyond 2020 that in many cases may exceed 500 km/h. In addition to high mobility scenarios, connectivity for zero to low mobility cases (interconnecting devices at stations and substations) must be also supported. Other applications, such as remote maintenance of rolling stock and remote processing will have central role in future railway platforms [1]].

In response to these challenges, IN2DREAMS relies on an advanced communication platform enabling connectivity between a variety of monitoring devices and computational resources through a heterogeneous network infrastructure. The connectivity, coordination and collaboration required is provided, on an on-demand basis in accordance to the cloud computing paradigm. To enable this opportunity there is a need for interconnecting ground infrastructures and on-board systems with the Operations and Control Centre (OCC), where the Data Centres (DCs) are hosted, through a heterogeneous network integrating wired and wireless network technologies. In this environment, optical network solutions can be deployed to interconnect distributed DCs, as they provide abundant capacity, long reach transmission capabilities, carrier-grade attributes and energy efficiency. At the same time, spectrum efficient wireless network technologies such as Long-Term Evolution (LTE) and WiFi can be effectively used to provide connectivity services to a large pool of mobile users and end-devices.

To address capacity limitations and high-speed mobility requirements of future railway systems, the communication platform will be managed through a flexible control plane offering the ability to create infrastructure slices over the heterogeneous network. Through this approach, railway system operators will be able to instantiate and operate several virtual infrastructures enabling multi-tenancy, supporting jointly energy and telecom services. This will allow operational and end-user services (e.g., Communications Based Train Control CBTC, Voice and data between central Command & Control and driver/cabin, streaming of surveillance video inside train and along railway infrastructure, monitoring of infrastructure devices, fleet management etc.,) currently provided through multiple technology-specific communication networks to be multiplexed over common infrastructures providing significant benefits in terms of cost and energy efficiency. The relevant benefits will be discussed and quantified using a purposely developed optimization tool.

3.2 Scenario Description and Key Technology Components

We consider a network infrastructure that relies on a set of optical and wireless network technologies to interconnect a variety of end-devices and compute resources. Through this approach, data obtained from various sources (monitoring devices, users and social media) can be dynamically and in real-time directed to the Operations and Control Centre (OCC) for processing.

The wireless domain of this infrastructure comprises cellular WiFi, LiFi and LTE technologies for the on-board and on-board to trackside communications (Figure 1). These exhibit a high degree of heterogeneity [2]] as they differ both in terms of operational and performance parameters, including spectrum use; antenna characteristics, physical layer encoding, sharing of the available spectrum by multiple users as well as maximum bit rate and reach. LTE is among the prime wireless access cellular technologies in 4G networks as it offers a theoretical net bit-rate capacity of up to 100 Mbps per macro-cell in the downlink and 50 Mbps per macro-cell in the uplink if a 20 MHz channel is used. These data rates can be further increased through Multiple-Input Multiple-Output (MIMO) technology. At the same time, LTE can provide improved Quality of Service (QoS) characteristics such as low packet transmission delays, fast and seamless handovers supporting high speed vehicular communications scenarios and operation with different bandwidth allocations.

In LTE systems, baseband signal processing functionalities are performed by the Base Band Units (BBUs) that are either co-located with the antenna Radio Heads (RHs) or located remotely exploiting the concept of Cloud-RAN (C-RAN) [7]. RHs are connected to the BBUs through high bandwidth links known as Front Haul (FH). C-RAN is expected to bring significant benefits in high mobility scenarios as it enables fast coordination and grouping of several cells forming super-cells with much larger size. To quantify the benefits of centralization in high mobility scenarios, let us consider the case where eNBs are placed 1.2 km apart. For a fast moving objects (i.e. trains) with a speed of 300km/h, handovers will be performed every 7s, leading to overutilization of network resources [4][5]. However, by clustering several eNBs together handover frequency can be radically reduced.

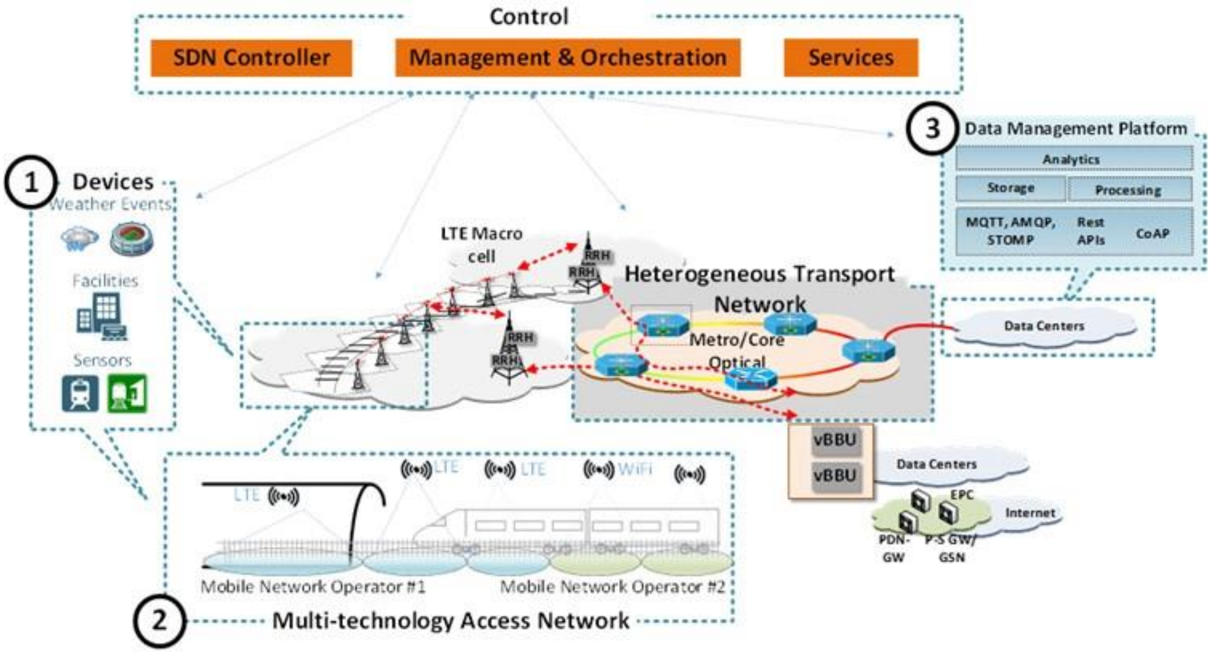

Figure 1: Converged Heterogeneous Network and Compute Infrastructures supporting railway services: Use case where data are collected from various devices (1) are transmitted over a 5G network (2) to the cloud based data management platform (3)

To support the transport network requirements associated with C-RAN in railway environment, we propose the adoption of an optical transport solution offering high capacity and advanced features including dynamic bandwidth allocation both in the time and frequency domain [8]. Given the technology heterogeneity of the proposed infrastructures, a critical function is interfacing between technology domains including isolation of flows, flexible scheduling schemes QoS differentiation mechanisms and mapping of different QoS classes across different domains. This can be achieved adopting flexible hardware functions that allow hardware repurposing through concepts such as hardware programmability. Hardware programmability can potentially enable dynamic and on demand sharing of resources guaranteeing also the required levels of isolation and security. In this context, programmable Network Interface Controllers that are commonly used to bridge different technology domains at the data plane can play a key role. These controllers have a unique ability to provide hardware level performance exploiting software flexibility and can offer not only network processing functions but also hardware support for a wide variety of communication protocols and mechanisms [9]. To enhance spectral efficiency, macro-cells can be complemented with small cells as they allow higher rates of frequency reuse over carefully designed geographical areas with easy access to the network backbone. In addition to small cells, given that WiFi networks are readily available in almost every public or private area and are easy to install and manage, significant benefits are expected by the joint consideration of WiFi and LTE systems. Additionally, the small cell concept can easily be extended to Visible Light Communications to overcome the high interference generated by the close reuse of radio frequency spectrum in heterogeneous networks. A network with multiple optical Access Points is referred to as an attocell network [4][10]. Since this operates in the visible light spectrum, the optical attocells do not interfere with any RF network. Therefore, the optical attocell layer adds data transmission capacity and enhances coverage while existing RF networks are not detrimentally affected.

3.3 Control and Management of the Integrated ICT Solution

As already discussed the proposed ICT platform (Figure 1) exhibits a large degree of heterogeneity in terms of technologies. To address the challenge of managing and operating this type of complex heterogeneous infrastructure, we propose the integration of the Software Defined Networking (SDN) and Network Function Virtualization (NFV) approaches. In SDN, the control plane is decoupled from the data plane and is managed by a logically centralized controller that has a holistic view of the network [ETSI 2014]. In early SDN deployments the data plane implementations only supported packet forwarding related functionalities. However, the advent of new high performing technologies such as LiFi and dynamic optical railway network solutions necessitate the execution of much more complex networking functions such as scheduling, network monitoring and management, resource virtualization, isolation etc. In response to this, SDN controlled programmable hardware infrastructures can now effectively support implementation of these functionalities using high level programming languages. At the same time, NFV enables the execution of network functions on compute resources by leveraging software virtualization techniques [ETSI 2015]. Through joint SDN and NFV consideration, significant benefits can be achieved, associated with flexible, dynamic and efficient use of the infrastructure resources, simplification of the infrastructure and its management, increased scalability and sustainability as well as provisioning of orchestrated end-to-end services.

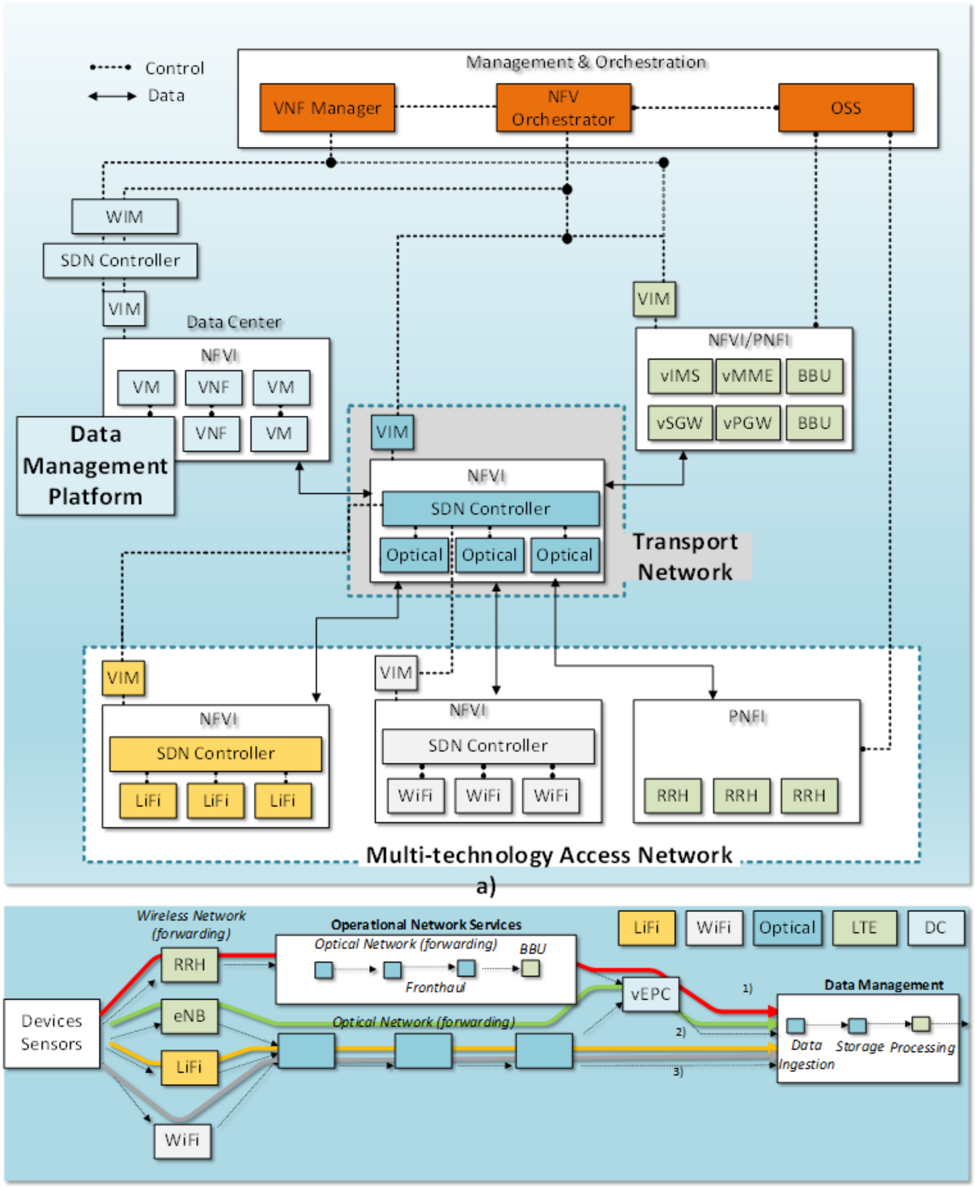

Figure 2: a) Example of an SDN/NFV-based control and management framework for Heterogeneous Network and Compute Infrastructures, b) Service chaining over heterogeneous network infrastructures supporting IoT services 1) IoT data stream over C-RAN, 2) IoT data stream over LTE, 3) IoT data stream over LiFi/WiFi

Examples of features that enable these benefits include the option to virtualize the separate control plane, using NFV and deploy Virtual Network Functions (VNFs). These are controlled by the SDN controller, to allow on demand resource allocation, able to support dynamically changing workloads [ETSI 2015]. SDN network elements can be treated as VNFs, since they can be implemented as software running on general-purpose platforms in virtualized environments. Both SDN and non-SDN models can be supported by SDN network elements. On the other hand, network applications can include SDN controller functions, or interact with SDN controllers and can themselves provide VNFs. Network elements controlled by SDN controllers can also provide Physical Network Functions (PNFs). Service Chaining (SC), combining and orchestrating physical and virtual network functions to support end-to-end service provisioning over heterogeneous environments, is considered to be one possible network application.

A typical example of an SDN /NFV architectural framework is illustrated in **Error! Reference source not found.** (a). It is observed that Network Function Virtualization Infrastructures (NFVI) comprising LiFi and WiFi components together with traditional non-virtualized physical infrastructures (e.g. LTE deploying RRHs) are inter-connected with the pool of computing resources, through SDN based optical network domains. Each WiFi/LiFi administration domain may host multiple SDN data plane elements and expose its own virtualised resources through an SDN controller to the upper layer SDN controllers. In our case the upper layer as illustrated in **Error! Reference source not found.** (a) refers to the optical layer. The hierarchical SDN controller approach adopted can assist in improving network performance and scalability as well as limit reliability issues [11] . In the proposed architecture, the top network controller will manage network resource abstractions exposed by the lower level controllers that are responsible to manage the associated network elements. Orchestration of both computation resources necessary to support the IoT use case and network resources is performed by the NFV Orchestrator and can be used for the support of multi-tenant chains, facilitating virtual infrastructure provider operational models. It is also responsible to interact with third party or legacy resources and support systems (OSS).

3.4 Modeling and Optimization

To address the great diversity of requirements introduced by the upcoming services in a cost-effective and energy efficient manner, optimal resource assignment considering the unique application and device characteristics is needed. In achieving this goal, the development of intelligent optimization algorithms considering different Key Performance Indicators (i.e. capacity, latency, energy consumption) for all physical and virtual network providers can play a key role. In the SDN/NFV architecture shown in **Error! Reference source not found.**, this process is located at the management and orchestration layer, offering to network service providers suitable tools that can assist in performing a broad range of tasks, including [11]:

- activities related to service chain management, able to Create/Delete/Update network SCs and a set of other relevant network functions;
- management of SCs considering virtual and/or physical resources and definition of traffic rules to dictate the selection of the optimal chain out of a set of possible chains;
- scale-in/out functionalities such as the, ability to bring up/down multiple network functions on an on-demand basis;
- traffic offloading from one forwarding entity to another;
- unified orchestration of compute and network elements;
- service orchestration with legacy or third party Operation Support System (OSS).

The combination of these tools facilitates the support of any mix of services, use cases and applications and can assist in addressing both technical and business challenges anticipated to arise in future network infrastructures. A specific use case that can be used to highlight the role of these tools is the provisioning IoT Services in railway environments, deploying a heterogeneous network infrastructure. Though this approach, scalability issues raised in the current railway OCC systems can be addressed by offloading intensive tasks (or accessing hosted content) to data management platforms hosted in the cloud.

To provide cloud-based IoT services, the orchestrator instantiates different type of VNFs that are deployed and chained together, each having specific processing and bandwidth requirements. Based on the type of wireless access technology (i.e., RRH, eNB, WiFi, LiFi) used to forward data from sensing devices (End-Point A in **Error! Reference source not found.** (b)) to the optical transport and the DCs where the Open Data Management (ODM) platform is hosted, multiple candidate service chains can be created. To realize each SC, sufficient network bandwidth and processing capacity must be allocated, corresponding to specific physical resources, for the interconnection and deployment of VNFs. VNFs are then processed in the order defined by the corresponding SC. For example, SC1 in **Error! Reference source not found.** (b) illustrates the case of traffic forwarding to remote DCs over the RRHs. To realize this, wireless signals received by the RRHs are forwarded over an optical transport network to the BBU pool and then to the DC location. Flow conservation as well as mapping and aggregation/de-aggregation of traffic between different domains should be also satisfied.

Apart from network and capacity constraints, end-to-end delay is an important Key Performance Indicator that needs to be also considered in the analysis. In highly loaded heterogeneous networks, end-to-end delay can be greatly influenced by queuing delays associated with the interfaces. Therefore, applying specific queuing policies and scheduling strategies at these locations is very important. Significant delay benefits can be achieved by instantiating the necessary network functions and reserving the required virtual/physical resources. End-to-end delay can be mathematically modelled through queuing models and the adoption of closed form approximations derived by modeling the different network domains as open, closed and/or mixed queuing networks. An example is illustrated in Figure 3 (c) where a three-dimensional Markov chain is adopted to model the three wireless access technology domains i.e. LTE, LiFi and WiFi. Each dimension of the Markov chains corresponds to a different virtualized wireless access domain with its state space defined as $S=\{ ├(i,j,k)┤ | i≤I, j≤J, k≤K\}$, where i,j and k correspond to the virtualized resources used across the LTE, LiFi and WiFi dimension respectively and (i,j,k) is a feasible state in S. Note that I, J and K correspond to the maximum set of resources that can be allocated to a specific provider. A key characteristic of the proposed scheme is that it allows modeling of traffic offloading decisions from one entity to another (i.e. LiFi to WiFi or LTE) as well as modeling of the arrival of a new service request by modifying the corresponding state i.e. $(i,j,k) \rightarrow (i+1,j,k)$ when a new forwarding decision is applied through the LTE network. The steady state probabilities of the Markov process can be determined in a unique way using the well-known matrix-geometric solution techniques and the corresponding service delay can be determined.

Markov chain models can be effectively used to evaluate the performance of domains where statistical independence between arrivals and services exists. Therefore, they can be applied to describe scenarios where virtual resources are realized through isolated physical resources such as different channels, spectrum, wavelengths etc. However, these models cannot be extended to technology domains where common buffers are shared among multiple virtual flows. A typical example for this exception applies to the edge nodes of the optical domain where common FIFO queues can be traversed by several virtual flows. A solution to this problem is to adopt the concept of Virtual Output Queues (VOQ) that can achieve traffic isolation among flows, providing at the same time flow-level bandwidth provisioning with strict delay guarantees (see Figure 2 c)) [12] . It should be noted that VOQ do not refer to physical entities, but correspond to pointers pointing to specific packets of the physical queues. In practice, they can be implemented in programmable hardware through the development of appropriate flow scheduling algorithms whereas centralized control can be implemented in Openflow.

3.5 Mobility considerations

An additional consideration to be taken into account during the operation of this type of infrastructures is train mobility. To handle mobility, redundant physical resources should be reserved to support uninterrupted service chaining. The amount of redundant resources increases with the speed of end-user mobility, the size of the wireless cells (mobile users associated with small cells will exhibit very frequent handovers) and the traffic model adopted. Based on their technical characteristics, the wireless access technologies adopted in this work, can address end-user mobility with different levels of effectiveness. E.g. LiFi providing high capacity levels is most suitable for indoor environments with limited end-user mobility, whereas WiFi and LTE can support lower capacities but higher mobility levels, with LTE being the most suitable technology for high speed vehicular communications. To maximize the benefits provided by the available technologies users with low mobility are offloaded to the LiFi domain releasing WiFi and LTE resources for mobile users. It is clear that, seamless handovers for a mobile user can be 100% guaranteed only if the required amount of resources is reserved for all its neighbouring cells. However, to limit overprovisioning of resources, a more practical approach is to relate the reserved resources in the neighbouring cells with the handover probabilities across LiFi and LTE cells and reserve a specific set of resources for handover purposes.

Similarly to the static cases described above, a five-dimensional Markov chain can be adopted to evaluate the performance of the virtualized wireless access network under mobility where two additional dimensions have been introduced to model handovers across WiFi and LTE. Given that users are much more sensitive to call dropping than call blocking a percentage of the virtualized resources is reserved for handovers. Thus, new service requests can use resources up to a specific threshold above which new requests are dropped. On the other hand, mobile users are dropped when all resources are already in use.

In this case also, a closed form approximation of the systems' state probabilities can be extracted using dimensional reduction techniques. The redundant resource requirement imposed for mobility purposes propagates also from the wireless access domain to the optical network and compute domains as depicted in Figure 3 c).

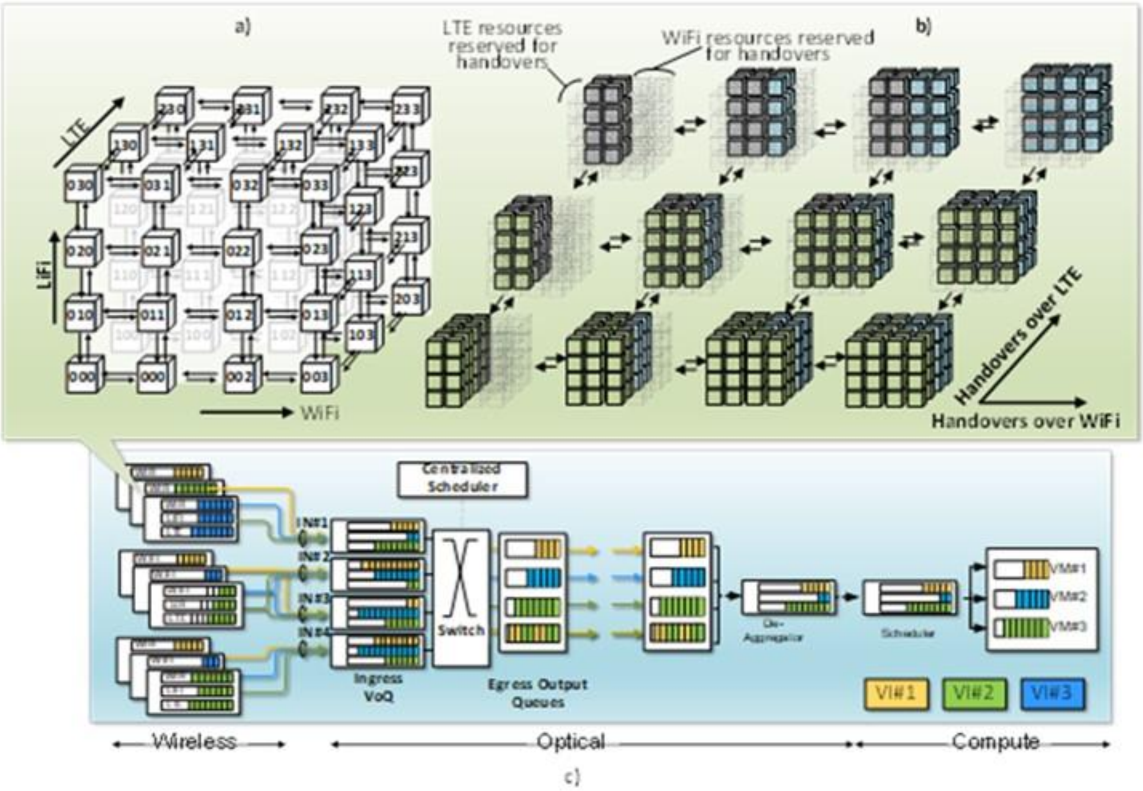

Figure 3: Modeling queuing delays in converged network environments a) three-dimensional Markov chain for estimating delays in virtualized wireless access network with traffic offloading capabilities from one forwarding entity to another b) five-dimensional Markov chain modeling mobility, c) End-to-end model as a network of queues

Taking into account the above considerations, a multi-objective optimization problem can be formulated that optimizes the performance of the converged network and computation infrastructure considering also the battery lifetime of the sensing devices under delay and mobility constraints. The description of a similar multi-objective optimization framework can be found in [8] [Tzanakaki et al 2015].

The output of this optimization problem can drive the selection of the optimal SC out of set of multiple chains. In addition, it can identify possible locations where VNFs or PNFs can be placed as well as the optimal wireless access technology that should be used.

3.6 Performance Evaluation - Simulation Environment and Parameters

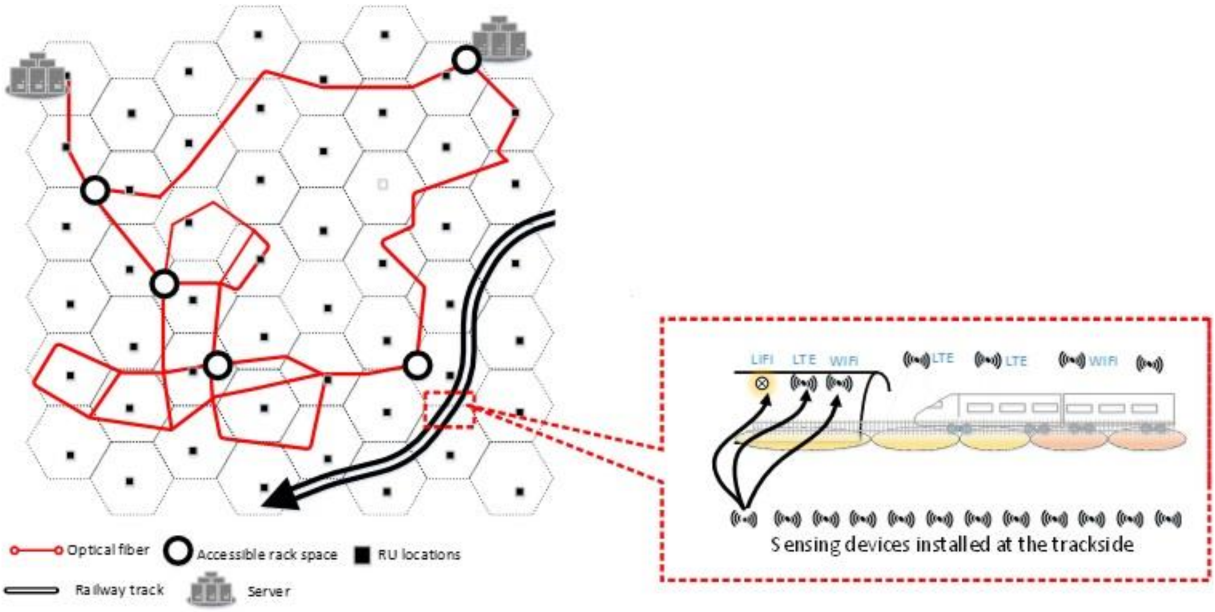

Figure 4: Bristol is Open Network Topology

The proposed framework is evaluated using the infrastructure topology, illustrated in Figure 4. This infrastructure covers a 10x10 km2 area over which 50 RRH units are uniformly distributed and comprises a set of optical edge nodes in the optical segment, and optical point-to-point links for front hauling the RRHs. The optical network technology adopted deploys a single fiber per link, 4 wavelengths per fibre, wavelength channels of 10 Gbps each, minimum bandwidth granularity of 100 Mbps and maximum link capacity of 40 Gbps. The power consumption model for the optical nodes is provided in [8]. Furthermore, a 2×2 MIMO transmission with adaptive rank 10 MHz bandwidth adjustment has been considered, while background network traffic over the serviced area according to real datasets reported in [14].

The railway-related network traffic is generated by a set of sensing devices installed both on-board and at the trackside. This traffic needs to be transferred at the servers where the ODM platform is located and processed by a specific set of computing resources. The power consumption of each device when information is transmitted over the LTE is 0.3W and 1.3W under idle and transmission/reception mode, respectively. The whole area is also covered by a set of WiFi access points offering 135 Mbps capacity having power consumption 1.28 W during data transmission, 0.94 W during data reception, 0.82 W under idle mode and 64 mW under sleep mode. In addition to this, a set of LiFi access points providing indoor coverage offering 300 Mbps data rate with 0.66 W power consumption under 200 lx light intensity is considered. Finally, each DC has a processing capacity of 80 Giga IPS and its power consumption follows the step-size power consumption model [7]2015]. The objective of the optimization framework is to identify the optimal SCs in order to jointly optimize the performance of converged network and computation infrastructure as well as the battery lifetime of the sensing devices. The former can be achieved by identifying the optimal routing paths and the location of the DCs where demands need to be processed, whereas the latter by ensuring that all sensing devices will forward their traffic through the optimal wireless access network technology. Through the appropriate selection of the optimal wireless access technology (WiFi, LiFi, LTE), sensing devices will try to prolong their battery lifetime without violating QoS specifications.

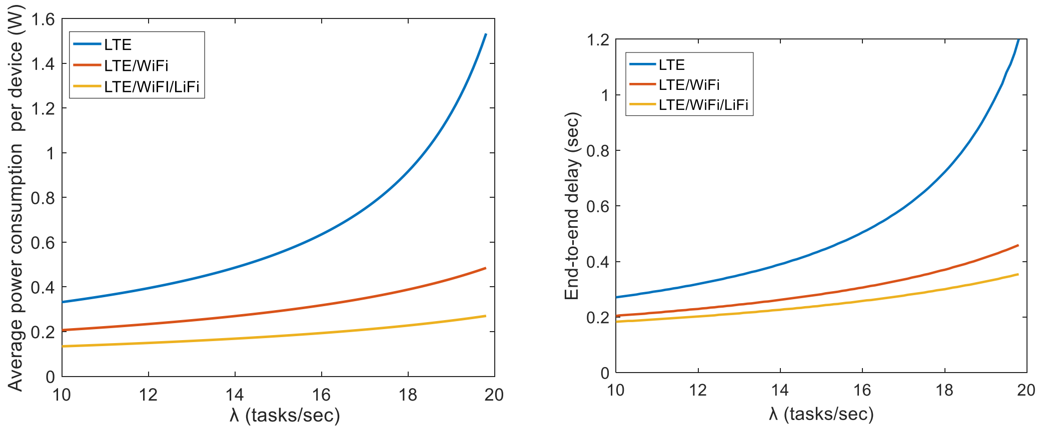

Figure 5: Impact of network integration on end-to-end service delay and sensing device power consumption (compute-to-network ratio= 0.03)

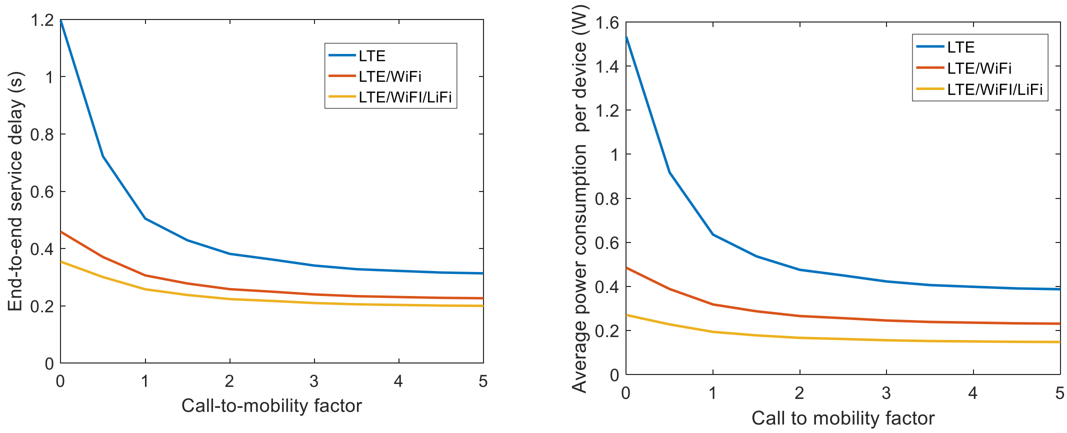

Figure 6: Impact of mobility on end-to-end service delay and device power consumption for Wireless Access Technologies (λ=10)

To achieve this, the Network-to-compute resources and the Call-to-mobility [Fang 2002] ratios are introduced to capture the communication cost and the average speed of the sensing devices, respectively. The former is used to capture the relation between computational and network bandwidth requirements, while the latter is defined as the fraction of the service holding time over the cell residence time. From Figure 5 it is observed that with the increase of the network load both the end-to-end service delay and the power consumption of the mobile devices is increased. However, for higher degree of network convergence we are able to drastically reduce the service delay and the power consumed by the mobile devices.

As also discussed in [Tzanakaki 2013], when mobility is high (lower call-to-mobility factor), additional resources are required to support the seamless handovers in the wireless access domain. This additional resource requirement also propagates in the optical railway network and the DC domain in order to ensure availability of resources in all domains involved (wireless access and backhauling, optical railway network, and DCs) to support the requested services and enable effectively seamless and transparent end-to-end connectivity between sensors and the computing resources. This leads to underutilization of network resources and therefore increased delays. The present approach, through the higher degree of consolidation and the better utilization of the network resources it offers, can handle high degrees of mobility and also support services with significant communication requirements in a very efficient manner (Figure 6).

3.7 Optical Wireless Communications Systems

During the last few years, Free Space Optics (FSO) technology is an ideal solution for modern networks desiring maximum signal/data security and speed and can fully cover modern communication network requirements between train busses and train stations. An FSO system is a point to point technology that uses a laser beam through the air to send data and offers full duplex 2.5Gbps Gigabit Ethernet throughput. The main advantages of such technology is the high data rates they can achieve and the low installation and operational cost. Furthermore, FSO systems are license free so they are ideal for locations where radio interference and congestion make installation of radio frequency alternatives impossible. Their successful presence in demanding applications and networks such as High Frequency Stock Trading and real-time Military Theater of Operations, can guarantee high performance, reliability and security for train network applications. Although such systems where used for fixed transmitters and receivers, due to modern tracking techniques they can also be used between mobile transceivers, so links between train busses and train stations or even between trains can be deployed. So the FSO systems will provide communication network with high performance, availability and security that will meet the modern and future requirements of communications.

3.7.1 Technical Specifications

FSO systems use low power laser beams up to 200mW for distances up to 5km depending on the atmospheric conditions of the area. More specifically a typical FSO system can deploy links of ~5km with attenuation 3dB/km, ~1.5km at 10dB/km and ~1km at 17dB/km.

The wavelengths that are usually used by FSO links are 1550nm and 850nm. Such wavelengths have the advantage of relatively low absorption due to atmosphere and the same wavelengths are also used in optical fiber communications so it is easy to match an FSO link with an optical fiber link without converting the wavelength of the signal. The beamwidth at the receiver is not more that 1m, so multiple laser beam can be used in a small area. This narrow beamwidth offers high security as it is difficult to intervene to the beam without distorting it and at the same time being perceived. Such system also offers very low latency, less than 20ns. The power consumption of a typical FSO system is less than 50W and such a low consumption is vital for mobile communications such as train communications. The operating temperature is between -40ºC and 70 ºC, so they can be applied in almost every European city.

Table 2: Technical parameter values of an FSO system

Technical parameters	Typical Values
Wavelength	1550nm / 850nm
Beam Power	~200mW
Latency	<20ns
Power Consumption	<50Watts

Technical parameters	Typical Values
Operating Temperature	-40ºC and 70 ºC
Bit Error Rate (BER)	$<10^{-10}$

FSO systems have multi protocol support and are compatible with IEEE 1588 Precision Time Protocol (PtP). Another important feature of FSO systems is the clock and data recovery (CDR) for daisy chaining and also offer cascadable back-to-back operation for extended links. A typical FSO system uses the protocols presented in the Table 3.

Table 3: Protocols used by FSO systems

Protocol	Data Rate
Fast Ethernet	125Mbps, full duplex
OC-3/STM-1	155 Mbps, full duplex
Gigabit Ethernet	1.25 Gbps, full duplex
OC-12/STM-4	622 Mbps, full duplex

The element management and control, contains a network or USB management interface and modern systems also contain an embedded SNMP card. The software that is installed with the FSO system can monitor and in some cases control critical parameters such as received signal strength, power supply voltages, laser currents, power and temperatures, clock recovery/sync status and network interface signal status.

So, the railway network from Redhill to Leigh (26Km) can be fully covered using roughly 10 FSO transceivers. These FSO systems can be placed in the train stations between Redhill and Leigh (Nutfield, Godstone, Edenbridge, Penshurst). As long as the distance between the stations is more than 5km, more FSO transceivers or relay nodes can be placed between these stations, applying a multi-hop technique. At the same time, an FSO system installed on the train may communicate with the fixed FSO transceivers using tracking techniques. The communication between the train and the stations can be deployed using modern pointing, acquisition and tracking (PAT) techniques. Such systems have already been developed for mobile FSO communications and consists of optical/electrical subsystems that sense target's coordinates on the sensor using drive actuators, closed loop control subsystems that maintain the pointing accuracy by maintaining the best receiving power and geometric mapping systems that provide an accurate mapping between sensor and actuator frame. More precisely, a coarse pointing mechanism using agile two axis gimbals and a high performance fine pointing mechanism are used. The synchronization between the coarse pointing mechanism and the fine pointing mechanism improves the tracking accuracy by using predictive control based on the kinematics and dynamics for the system components. Such systems have low cost and complexity and have already succeed good performance with BER lower that 10-10.

This FSO network will provide fast, reliable and secure data flow between all the train stations and the trains.

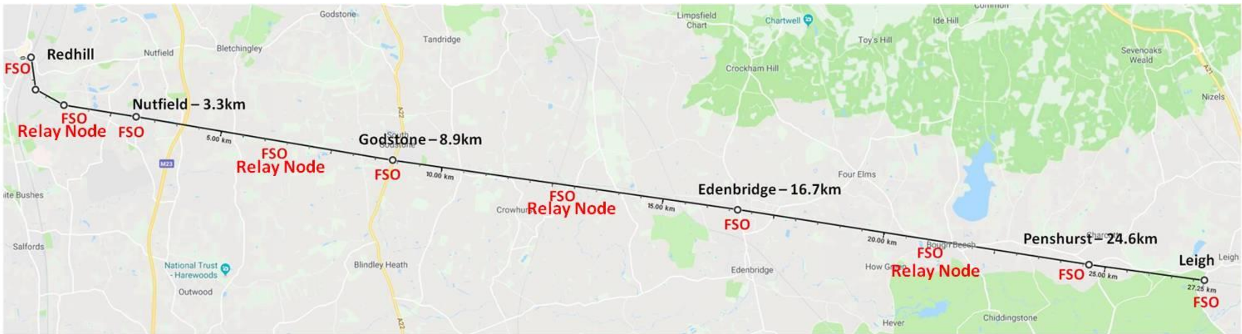

Figure 7: Communication coverage of railway network between Redhill and Leigh using FSO systems

While the train passes through a tunnel, the RF signal becomes very weak or in some cases can't even penetrate. So the communication may be interrupted without using multiple antennas in the tunnel. Optical communications may solve this problem by using the already existing tunnel lighting as visible light communication (VLC) transceivers communicating with the respective lights of the train. So in a long tunnel there is no need for placing FSO transceivers inside, only in the entrance and in the exit, creating a cascaded FSO-VLC system. The signal conversion between the FSO and the VLC system has low complexity and delay. Furthermore, the optical signal of the FSO transceiver can be easily converted to drive internal WiFi or LiFi systems of the train. So when the train passes through tunnels there will be no reduction of the quality of service if the FSO transceiver is placed in the tunnel.

3.7.2 Advantages

High Performance, reliability and availability

FSO systems are already commercially used for sending up to 1.25Gbps of data, voice and video simultaneously through the air and in the near future will be capable of speeds of 10Gbps using Wavelength Division Multiplex (WDM), a widely used multiplex technique in optical communications. OFDM can also be used, a technique with high impact in communications. FSO systems can cover distances up to 4Km maintaining the high performance. This distance can be extended in case of use of either amplify and forward (AF) or decode and forward (DF), relays. Due to the high bit rate they can achieve it is easily feasible to reduce it for increased availability in case external factor degrade the quality of the channel. Concerning the availability of an FSO system, they are able to reach the five 9's (99.999%).

Low Cost

Another important factor of FSO systems is the very low cost that offers fast and high return of investment (ROI). The low cost is also resulting from the low power that needs, up to a 100mW for a distance of 3Km providing high performance and reliability. Furthermore FSO technology is spectrum license free and can be deployed behind windows, eliminating the need for costly rooftop rights. The cost for an FSO system (transmitter-receiver) is between 10.000€ and 25.000€.

High security

The installation of an FSO system in a railway network will offer high level of communication security. Due to the narrow beamwidth of the laser beam, the intruder will have to expose himself or his equipment in order to intercept a portion of the transmitted power without bringing down the link. Furthermore, if an amount of the signal power is intercepted, the power at the receiver will significantly change of will be distorted. The most vulnerable area of potential interception in an FSO system is behind the transceiver. Because the incident beam of light has a larger cross-section that the lens of the receiver, the laser beam continuous to propagate behind is for some distance. An intruder could conceivably mount an unauthorized receiver in this area. This way can be solved by placing a blocking shield behind the transceiver. Except of high level security of the physical layer, FSO systems use special encryption schemes as the Fastlane KG-189, Taclane KG-75/175 (NSA Type 1) and all the AES and Triple DES encryption systems.

3.7.3 Challenges and enhancement techniques

The primary challenge to FSO based communications is dense fog due to the size of water droplets. Rain and snow have little effect of FSO technology. A solution to counter fog when deploying FSO systems is to install relay nodes between the transmitter and the receiver, known as multihop technique. These relay nodes can be AF nodes or DF. AF offers a simple and low delay solution while DF nodes are more effective and reliable. FSO installations in extremely foggy cities such as San Francisco have successfully achieved carrier-class reliability.

Another factor that degrades the performance of an FSO system is the misalignment due to building sway. The movement of buildings can upset receiver and transmitter alignment. A solution is the use of divergent beam to maintain connectivity. When combined with tracking, multiple beam FSO systems provide even greater performance and enhanced installation simplicity.

Finally, turbulence is another factor that affects the reliability and availability of the system. Heated air rising from the earth or man-made devices such as heating ducts, create temperature and pressure variations among different air pockets. This causes fluctuations in signal irradiance at the receiver, the so-called scintillation effect that creates a fading channel. Such fluctuations are very fast and intense and can't be predicted. So theoretically this phenomenon is investigated statistically. In order to reduce the impact of turbulence, the signal can be sent multiple times, a technique known as diversity The most of the times, diversity is realizing in space, in time or in wavelength. In the spatial diversity scheme, the FSO system uses multiple transmitters and/or receivers at different places that send and receive copies of the same signal, resulting to a decreased probability of error. In time diversity schemes, there is only one transmitter – receiver pair, but each piece of the information signal is retransmitted at different time slots. Finally, the wavelength diversity system, uses a composite transmitter and the signal is transmitted at the same time at different wavelengths towards a number of wavelength-selected receivers.

A technique that can always be used in order to enhance the performance of an FSO system is OFDM-FSO Transmission Link Incorporating OSSB and OTSB Schemes. By introducing the OFDM scheme, an effort has been made to probe the impact of the environment conditions and to design a high speed and long reach FSO system free from fading. It is concluded that hybrid OFDM-FSO system performs better in diverse channel conditions and upon comparing both OSSB and OTSB schemes OSSB performs better than OTSB at high data rate as it has more immunity against fading due to weather conditions.

Another enhancing technique is the SAC OCDMA Based FSO System. Spectral Amplitude Coding Optical Code Division Multiple Access technique is used in FSO system by the researchers. This multiplexing scheme has several advantages like flexibility of channel allocation, asynchronously operative ability, privacy enhancement, and network capacity increment. KS (Khazani-Syed) codes are used with SDD (Spectral Direct Decoding) technique. An optical external modulator (OEM) is used to modulate the code sequence with data. The data is an independent unipolar digital signal. Mach-Zehnder Modulator (MZM) is used and combination of modulated code sequences is transmitted through the FSO link and these sequences are separated by an optical splitter at the receiver end. The overlapping chips are discarded to avoid the interference at receiver end and decoder will only filter the non-overlapping chips. Optical band pass filters serve the purpose of encoders and decoders. A Low Pass Filter (LPF) is used to recover the original data. The performance of this system with SDD technique is analysed along with FSO system using intensity modulation with direct detection technique. SDD technique performs better and the link distance is increased.

Table 4: Performance of a commercial FSO system

Weather	Data Rate	Link Distance
Clear	2Gbps	10km
	5Gbps	6km
Low Haze	2Gbps	5.4km
	5Gbps	3.4km
Low Fog	2Gbps	1.35km
	5Gbps	1km

Another factor the decreases the performance of an FSO system is the optical noise of the sunlight at the receiver's input that deteriorates the signal to noise ratio (SNR) of the optical link. A very significant effect that increases the noise of an FSO system is the background optical noise with the main source being the sunlight radiation that is difficult to be avoided. A robust technique for reducing the effect of background noise is the application of optical filters at the receiver's photodiode in order to decrease the wavelength range received that is not used for data transmission. The SNR improvement, due to the VO2 optical bandpass filter, has already been estimated theoretically and verified experimentally. It works by reflecting unwanted wavelengths. In comparison to interference filters, VO2 is a simple layer and this makes fabrication of VO2 filter simpler and economically viable.

3.7.4 Safety

Safety in FSO systems can be a concern because the technology uses lasers for transmission. The proper use and safety of lasers have been discussed for more than 3 decades. The major concern involves the eye exposure to light beams so strict international standards have been set for safety. Nowadays all lasers of FSO systems are Class 1M that are safe for the naked eye. Furthermore all FSO systems comply with the following standards:

- International Laser Standard IEC 60825-1/A2:2001;
- European Standard EN 60825-1/A2:2000,DIN VDE 0837-1A/2;
- IEC/EN 60825-7:1998;
- DIN V VDE V 0837-7:1999;
- US user Standard ANSI Z 136.1, CDRH 21DRF.

3.7.5 Experimental Data

A commercial FSO system has been installed for research purposes in the city of Piraeus. The link distance is 3 Km and the technical specifications are presented in Table 5.

Table 5: Technical Specifications of FSO system

Parameter	Values
Data Protocol	Fast Ethernet, ATM, STM1, OC3, SMPTE
Distance	3km
Wavelength	830-860nm
Maximum Bit Rate	100Mbps
Output Power	<150mW - 3 Laser beams
Receiver field of view	2mrad
Sensitivity	-46dBm

The results of the data that were collected for almost 2 years, have proven that FSO system's performance remains high even in bad weather conditions (rain, haze). At the same time, the experimental data have verified many theoretical models used for channel modeling on various turbulence conditions, so the use of such theoretical models are accurate and robust in order to design an operational network.

3.8 Deploying the ODM platform in 5G

3.8.1 Introduction

Machine to Machine (M2M) communications and Internet of Things (IoT) have resulted in tremendous growth of globally generated data which according to the International Data Corporation is expected to exceed 163 zettabytes/year by 2025 [1]. Once analysed, this massive amount of data, generated by billions of connected devices ranging from smart devices and autonomous vehicles to remote sensors, can assist infrastructure providers, a variety of vertical industries, policy makers and the public to derive new insights and improve society's quality of life. 5G systems with their ability to offer high-speed/low-latency internet connectivity as well as easy access to storage and processing resources, can enable massive IoT to a broad spectrum of vertical industries [37].

To achieve this, 5G networks rely on a set of hardware, software and architectural innovations. These include: i) the concept of Cloud Radio Access Network (C-RAN) that allows a heterogeneous set of densified Remote Units (RUs) to offload their signal processing related functions to a Central Unit (CU). C-RAN provides significant benefits in improving wireless access network scalability, controllability, agility and flexibility, ii) a high capacity transport network to offer connectivity between the RUs and the CU for the support of the front haul (FH) services required by C-RAN, integrating advanced wireless and optical network elements , iii) a set of compute resources responsible for the processing of operational data and the provisioning of real-time data analytics, and iv) an intelligent control plane enabling precise network resource allocation and data forwarding to the appropriate processing platforms. In this context optical networking plays a key role in supporting the demanding transport network requirements in terms of capacity, delay and flexibility. A typical example of a 5G network exploiting a flexible optical transport solution to provide IoT services for vertical industries is the use case of railway systems. More specifically

Figure 8 illustrates a specific use case developed in the context of the In2Dreams project [38]. In this rail use case, data collected from various devices installed on-board and at the track side of the rolling stock (1) are transmitted over a multi-technology network (2) operating in accordance to the C-RAN paradigm (3) to the cloud-based Data Management Platform (DMP) (4) that is responsible to perform data collection, storage and processing.

In this environment it is very important to identify the optimal assignment of each task to the appropriate processing platform as this decision is expected to give significant efficiency gains. In the case of IOT over 5G, this decision is linked to the placement of BBU's and DMP's construction elements (i.e., message broker, control server and database manager) to suitable processing units. In current deployments, this is performed without taking into consideration the details and specificities of the individual processing functions of BBUs and IoT services [39], [40] which can provide significant efficiency gains [41]. To take advantage of the appropriate mapping of processing functions to suitable available compute resources that can be hosted at Data Centres (DCs) we rely on the concept of compute resource disaggregation. This approach allows individual allocation of processing functions, associated with a specific FH and IoT-Backhaul (BH) services, to different servers depending on the nature and volume of their processing requirements. To quantify the benefits of the proposed approach we have performed experiments analysing the processing requirements of, i) a typical BBU for an LTE system using an open source suite for benchmarking wireless systems (WiBench) [42] and, ii) an IOT platform using an actual DMP system support the experimental smart metering campaign of the In2Dreams project [38].

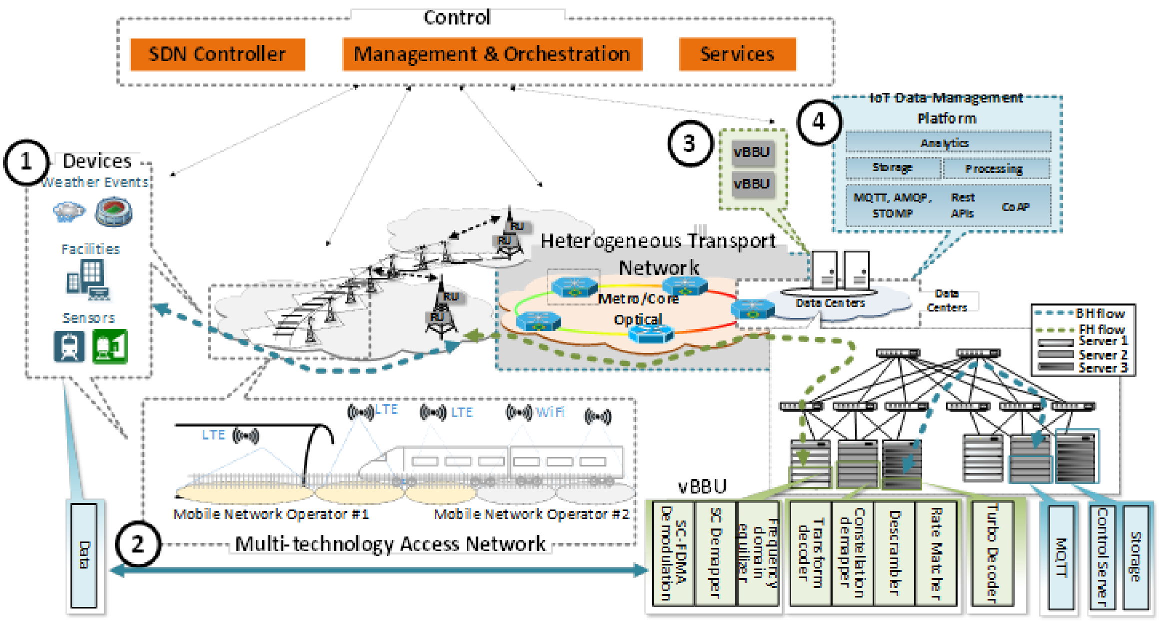

Figure 8: IOT services over 5G supporting verticals: data collected from sensing devices (1) are transmitted over a multi-technology network (2) operating in accordance to the C-RAN paradigm (3) to the cloud-based data management platform (4) that is responsible for data collection, storage and processing

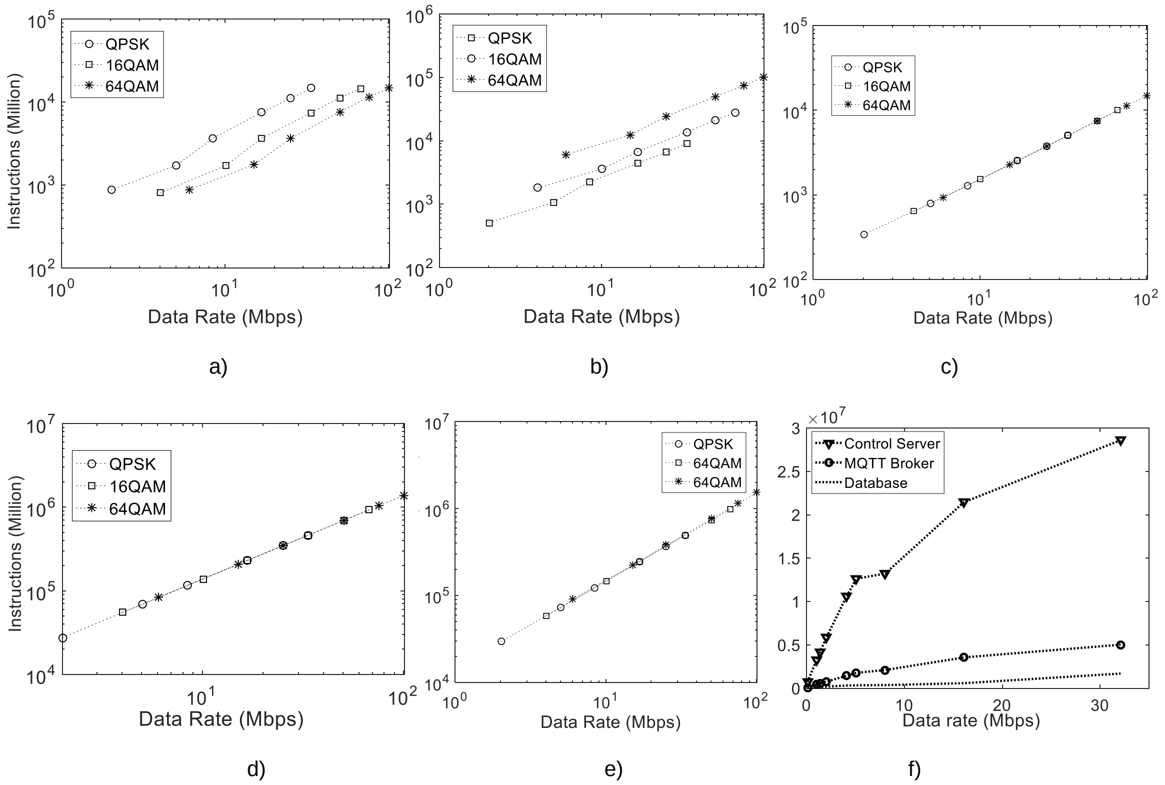

Figure 9: Instructions per second under various data rate for the following main contributing functions a)Equalizer, b) Demodulation, c) Rate Matcher, d) Turbo Decoder, e) Total BBU Instructions, f) DMP components

In addition, a Multistage Integer Linear Programming (ILP) modelling framework was developed to assign the construction elements of the nested FH/IOT-BH chain to suitable servers available at the DC. The output of our experiments was used as a realistic input to our ILP model to evaluate the energy consumption requirements of the compute resources for the proposed approach where softwarised BBUs [42] and IOT DMP are placed within the DCs (referred to as SW-IOT). As the proposed approach is based on the concept of disaggregation we refer to it as the disaggregated SW-IOT (DSW-IOT).

Table 6: Technical specifications of the servers used in the numerical evaluations

CPU Description	MHz	Chips	Cores	Total	RAM (GB)	Max IPS	Max Power (Watt)	Idle (Watt)	(IPS/watt)
Intel E3-1260L v5	2900	1	4	8	16	31802689	47.9	14.4	529836
Intel Xeon E5-2699 v4	2200	2	44	88	64	226542946	247	46.6	775056
Intel Xeon Platinum 8180	2500	2	56	112	192	378651974	426	39.5	833346
Intel Xeon Platinum 8176	2100	8	224	448	768	1304763025	1478	206	766748

3.8.2 Use case Definition

We consider the case where IoT services are provided over a generic 5G C-RAN infrastructure. For C-RAN, the RU processing requirements are supported by a set of compute resources located at the DCs. The compute resources comprise a set of General-Purpose Processors (GPPs) with each processor having specific processing capacity and performance per Watt [41]. Servers are organized following the simple tree structure of Figure 8 and are responsible to provide the necessary compute power for the provisioning of FH and IOT-BH services. The compute requirements for BBU processing for each RU is calculated by the sum of all contributing computing elements responsible to perform the required functions including, SC-FDMA demodulation, SC demapper, frequency domain equalizer, transform decoder, constellation demapper, descrambler, rate matcher and turdo decoder. As shown in Figure 8, these functions are executed in a specific order. In addition to the operation of the C-RAN, sensing devices transmit their readings to the DMP. The DMP comprises three main components including the Message Queuing Telemetry Transport (MQTT) broker, the central server and the time series database. Our objective is to identify the optimal GPP where each function can be allocated so the total power consumption at the DC is minimized satisfying, at the same time, QoS constraints imposed by the C-RAN and the IOT services.

To achieve this, we initially calculate the actual processing requirements of each BBU function and of each DMP component service through the WiBench and the DMP platform for smart metering railway services reported in [38] respectively, under various wireless access system configurations and different IoT loads. The processing requirements of each function are then used as input to the multistage ILP-based optimization framework that creates nested service chains that assign each function to a suitable GPP in an energy efficient manner.

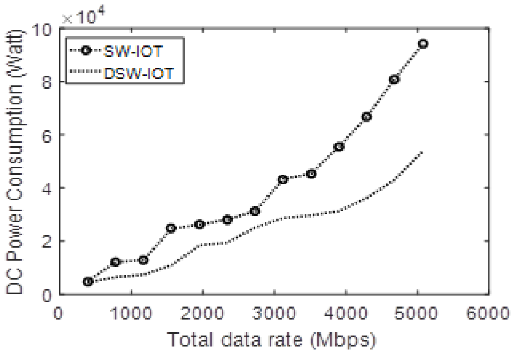

Figure 10: DC power consumption for the traditional softwarized IOT (SW-IOT) and the disaggregated SW-IOT (DSW-IOT) as a function of the total wireless access traffic

3.8.3 Benchmarking Framework and Results

Experiments were conducted using WiBench, an open source implementation of the LTE protocol stack. Intel's VTune Amplifier 2018 was used to provide the performance analysis. For the DMP, we analyzed the processing requirements of the message brokering servers (MQTT), ii) the Control Server that receives the data either from the MQTT broker or HTTP requests and forwards them for storage and iii) the time series database used for storage. The investigation of each BBU function, for various configurations of the LTE uplink system, was conducted through a series of experiments. Results show a linear dependence, for every function, between the number of instructions performed and the data rate (Figure 9 a)-e)). Instructions increase exponentially with the IOT data rate for the Constellation De-mapper, and linear for the Rate

Matcher and the Turbo Decoder, while for all other functions are independent. The total number of instructions, for the BBU processing, is mainly determined by the Turbo Decoder, which involves 1 to 4 orders of magnitude higher instruction number compared to the other functions, depending on the data rate. The processing requirements of the DMP were derived by calculating the instructions per second required by its components as a function of the total IoT load (Figure 9 f)).

To quantify the benefits of the proposed approach, the simple DC network topology of Fig.1 is considered. This topology comprises 6 racks each comprising 48 servers. Connectivity between racks is provided assuming the intra-DC optical network solution described in [43]. In the numerical calculations we consider four types of servers, randomly placed within the racks. The technical specifications of these servers are provided in Table 2, while their power consumption follows the linear stepwise function described in [44]. For the wireless access, we consider the topology described in [38] in which the served area is covered by a set of RUs providing connectivity for the IoT devices installed on-board and at the track side of the rolling stock. RUs forward their FH flows to the DCs for processing Figure 10, compares the performance of the proposed optimization DSW-IoT scheme in terms of power consumption with the conventional SW-IoT as a function of the served IoT traffic. As expected, the power consumption at the DCs increases with the IoT load. However, the DSW-IOT offers much better performance due to its increased ability to mix and match compute and network resources, leading to improved utilization of the servers and to higher energy efficiency.

3.9 Communication Platform: Control and Management based on Software Defined Networking

Mobile and wireless networks require continuous adjustments, as the wireless medium becomes the primary and sometimes the only method of an increasing number of people. Thus, mobile providers have to transfer higher traffic volumes and support more upgraded services. In addition, mobile networks must simultaneously support multiple generations of mobile services (3G, 4G\LTE) along with a range of user services (VoIP, string media, messaging), resulting in a variety of traffic properties.

Static traditional networks cannot cope with the challenges of additional bandwidth required to meet the explosion of data traffic. The movement towards a denser network (Densification) in order to increase the efficiency of the spectral range is imperative. This leads to a network of much smaller cells, reducing the distance between the terminals and the access point (RAP). Mobile providers go into small-cell technology to increase capacity through reuse of frequencies, especially in densely populated areas. Small cells will not be homogeneous but will form a flexible heterogeneous network where resources can be dynamically adapted, as the users' demand in space, time and spectral resources varies. However, placing cells closer is one of the factors that increase interference between cells, which is accentuated for higher bandwidths such as 4G LTE services. Furthermore, the existence of complex policies to ensure the proper access for the proper service and the proper handoff of control among access types is essential.

Currently, mobile networks are based on Operations support systems (boss) and management systems that require significant specialization and platform resources. Because these systems require manual intervention, networks are prone to misconfiguration errors and delays in provisioning and troubleshooting. However, an efficient management is crucial nowadays, especially due to the fact that conventional devices today need to support 3G, 4G, as well as Wi-Fi and Bluetooth connectivity at the same time. Such diverse wireless technologies require mobile providers to maintain and operate remote access and backhaul and core networks, increasing both cost and management complexity. In addition, providers need flexible implementation choices to move from older to more recent technologies without affecting the customer's experience.

The ever-changing business demands, require rapid implementation of new mobile services and rapid adoption of new technologies, imposing the need for a new network architecture that goes beyond the limitations of the current network architecture, in which it would require weeks or even months to introduce new services due to manual procedures for activating, delivering and securing.

Software-Defined Networks (SDN) are an emerging network architecture that aim to provide to network administrators the ability of programmable, flexible and dynamic resource management through centralized control. The fundamental concept of this architecture is the separation of the network control from the forwarding functions. The control plane is responsible for the routing of the packets and the data layer for transporting them. Unlike the traditional TCP / US architecture, where routing is in line with the static IP protocol, SDN provides the ability to use routing policies based on the requirements of each client and the needs of the network. Networks become programmable and manageable, making the infrastructure transparent for the applications and services they offer, while facilitating the resource allocation, the scalability in distributed data centers, and the virtualization of devices that is necessary for Cloud environments [1][2].

3.9.1 SDN principles

The SDN architecture is divided in three basic layers: the application layer (application plane), the control layer (control plane) and the infrastructure layer (data plane) Figure 11.

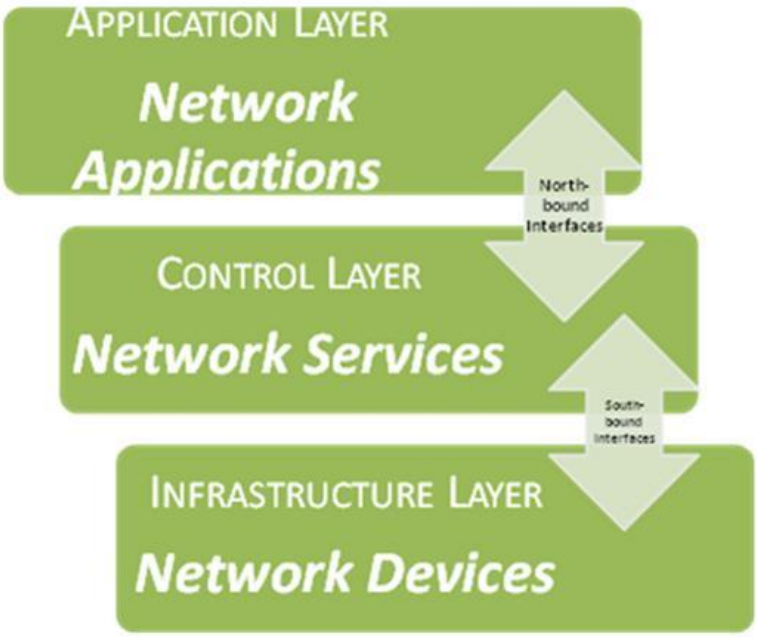

Figure 11: SDN Architecture

In the application plane reside the network applications through which the network administrator can immediately communicate with the controller in order to learn the network requirements and define the required behaviour to be applied to the network. The communication between the applications and the centralized controller is performed through the NorthBound SDN Interfaces (NBIs). The NBIs provide the application-level view of the network as well as the ability to directly interfere with network requirements.

The control plane contains the centralized network controller and is responsible for setting the forwarding rules to the network devices, through the Southbound SDN interface. On a physical level, the controller can be distributed to different physical machines in order to provide scalability and optimization of processes. However, the controller behaves as a unique logical unit, with its own operating system making uniform decisions. There are two main operations that a controller must perform. The first is to transfer the system requirements that are provided from the application plane, to the data plane. The second is to notify the application plane for any changes to the network operation that are perceived from the data plane. It consists essentially of NorthBound Agents that contribute to the application-control plane communication, the control centre that is responsible for the decision-making processes, and the CDPI (Control to Data-Plane Interface) Drivers that are responsible for the proper communication with the data plane, by providing programmatic control of all data forwarding features, report of traffic statistics on the network, and updates to the controller in the event of a network change or problem.

The infrastructure layer is a network logical entity that provides supervision and control of data forwarding and processing that take place on the network. It consists of a CDPI Agent in order to communicate with the controller and a set of devices responsible for the forwarding. The data layer - although behaving as a unified module - may be composed of distributed physical devices with shared resources, and can also work with non-SDN devices.

Lastly, outside the application, control and infrastructure layer resides the Management and Admin of the SDN. Although, it is not part of the SDN architecture, it is basically the reason for the development of SDN, that is, the control that can now be applied to the networks that support SDN. Through the application layer, the administrator can update the network by defining appropriate behaviours. The communication of the applications and the administrator is accomplished with the help of the controller.

3.9.1.1 OPENFLOW PROTOCOL

OpenFlow is the first standard protocol for the communication of the control plane and the data plane in SDN architectures. It allows the direct access to the network devices, i.e. virtual and physical switches and routers, and the management of their data forwarding platform. OpenFlow protocol is essential for the separation of the control from network devices and its relocation to a logically distributed control software. It is applied to both sides of the interface between the SDN control software and the network infrastructure [2].

OpenFlow uses the concept of flow to identify network traffic. The concept of flow is based on predefined mapping rules that can be statically or dynamically set by the controller. It also allows the administrator to specify how network traffic will be distributed according to network resources, application requirements, and routine network traffic patterns. In contrast with traditional IP networks, where in Internet routing two streams with the same start and end points will follow the same route because of the same IP addresses, in the SDN architecture two or more streams may have the same source and destination points and follow different routes with different priorities, depending on the routing policy that are assigned to.

Three kinds of messages are supported by OpenFlow: Controller-to-Switch, Asynchronous, and Symmetric. Controller-to-switch messages are originated from the Controller and enable the direct management and supervision of the state of a switch. Asynchronous messages start from the switch and aim to inform the Controller either about events that happen on the network or about changes in the state of the switch. Finally, symmetric messages originate either from the Controller or the switch and are sent without any prior request [3].

SDN that supports OpenFlow protocol can be deployed on existing networks. Networking devices can support both OpenFlow and conventional TCP / IP routing. There is the possibility of hybrid switches operating in any environment (SDN or not) or implementing centralized control over applications (network monitoring and management), so it is not necessary to replace each network device with SDN switches. [3] This is a great advantage for network providers that can be led into the SDN architecture gradually, relying on their traditional networks, turning them into SDNs.

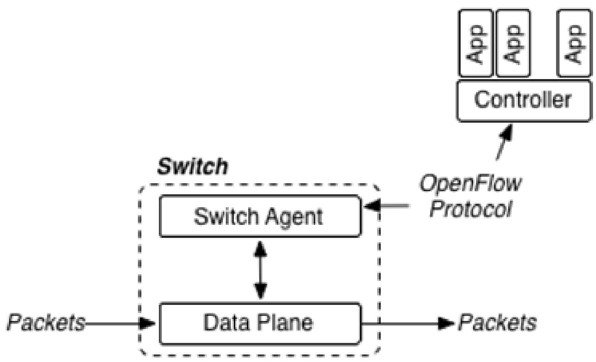

Figure 12: Architecture of OpenFlow switch [2]

3.9.1.2 OPENFLOW SWITCH

The OpenFlow switch consists of two parts: the switch-agent and the data platform (Figure 12). The switch-agent communicates through the OpenFlow protocol with the controller and with the data plane through an appropriate protocol. It is responsible for translating the controller's commands into the appropriate machine language and sending them to the data platform. It also receives from the data platform any notifications and converts them into OpenFlow messages understandable by the controller and promotes them.

The data platform performs the management and forwarding of the packets. Depending on the configuration, it may send packets to the switch-agent for processing. It includes flow tables and related actions for each flow, thus the packets are forwarded according to which flow they belong to and to which flow-table action they will be matched. [2]

An OpenFlow switch can be any packet-forwarding device (router and switch) and consists of one or more flow-tables that are used to match and forward packets, a group table, and a secure communication channel to a Controller. [3] The Controller manages the OpenFlow switch through the channel using the

OpenFlow protocol (Figure 13). With the implementation of OpenFlow, the controller can proactively or reactively add, update, and delete flow tables. Each flow table contains recordings for the flows with counters and actions for each flow. Each package entering the OpenFlow switch is contrasted with the flows of the first flow-table and continues with additional flow-tables. In the case of matching with a flow, the set of commands associated with that flow are applied, otherwise the outcome depends on the configuration of the table-miss flow. For example, the packet can be forwarded to the Controller via the OpenFlow channel, discarded, or continued in the next flow-table.

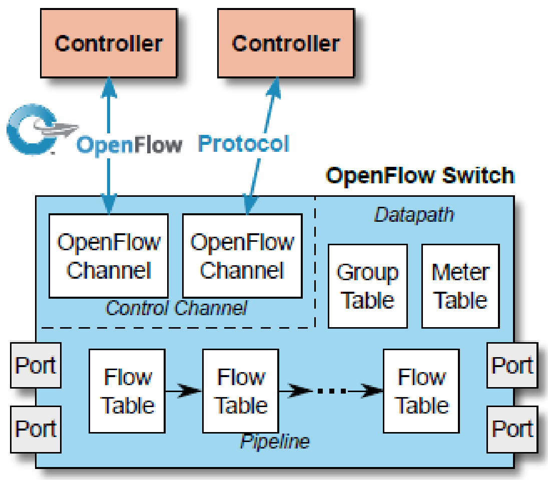

Figure 13: Main Components of an OpenFlow switch [3]

3.9.1.3 OPENDAYLIGHT CONTROLLER

The OpenDayLight (ODL) controller was developed by the Linux Foundation community and it is an open source project on Software Defined Networking. Its code is written in JAVA language and it can be extendable.[4] Major companies in the industry such as Cisco, Brocade, Juniper Networks, IBM, Vmware, Ericsson, Microsoft, Big Switch Networks as well as independent Open Source engineers are working to help develop the ODL project. ODL supports the OpenFlow protocol as well as other SDN architecture protocols and technologies since they can be properly configured by the administrator [5].

The architecture of ODL controller is show in Figure 14. The controller platform incorporates both Northbound and Southbound interface. The Northbound interface provides services to the controller as well as a set of REST API applications that can be used to manage and configure the network infrastructure. Access to the Northbound interface of the controller is possible after authentication and licensing of the user. REST (Representational State Transfer) is an architectural design used in web applications (WEB development). Systems designed in REST architecture have good performance, reliability and scalability in terms of the number of users they can gradually support. RESTful systems typically communicate through the HTTP protocol through the GET, PUT, POST, DELETE methods commonly used by the browser to retrieve and send data to remote servers. [6] Outside of the browser, the communication with RESTful systems can be accomplished via a terminal and in particular via the curl bash command, along with the appropriate arguments (username, code, URI). Curl can send any required text in json or xml format. The information returned by the system is also in json or xml format.

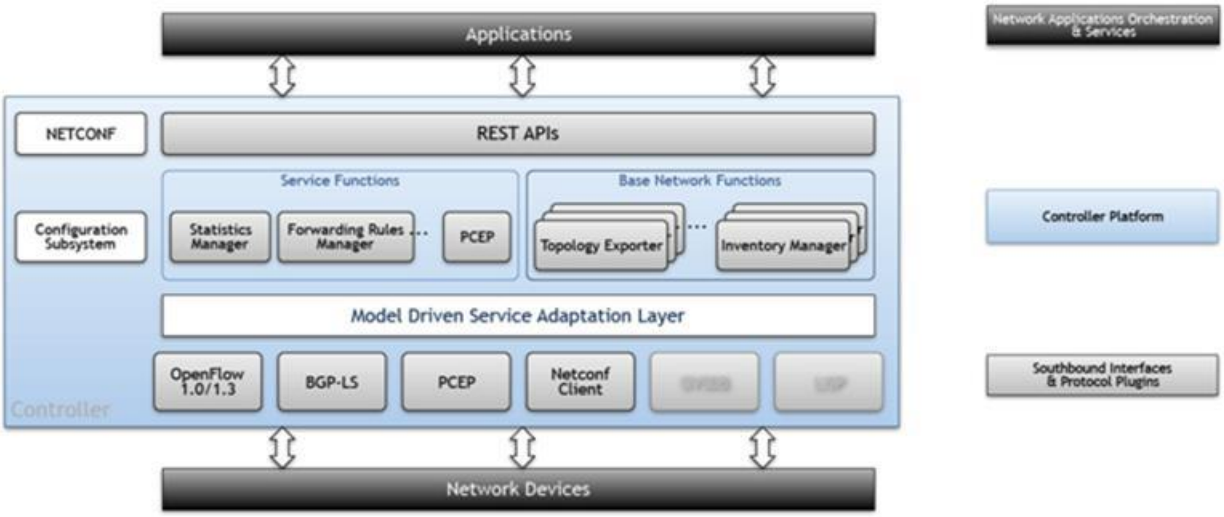

Figure 14: Architecture of OpenDayLight Controller [5]

The Southbound interface implements protocols for managing and controlling the underlying network infrastructure. Several extensions are being implemented either to support network protocols or to directly communicate with the hardware. The most prominent protocols implemented are OpenFlow, NetConf, SNMP, OVSDB.

The controller platform communicates with the network using the Southbound interface and provides basic network services through a set of functions described below [7]:

- Host Tracker: It saves information about system terminals (MAC addresses, IP, switch type, port number) and provides an API to retrieve information about them. It can work in a static or dynamic way. Statically, the Host Tracker database is manually handled through Northbound APIs. Dynamically Host Tracker uses the ARP protocol to detect information;
- L2 Switch: Upon the arrival of a packet, this function learns the MAC address of the source. If the destination is known, it transfers the packet to the destination, otherwise it sends a broadcast message (i.e. a message to all devices) to all external network ports;
- OpenFlow Forwarding Rules Manager: It manages basic forwarding rules, resolves any conflict between these rules and validates them. Furthermore, it communicates with Southbound extensions and loads the OpenFlow rules to the switches;
- OpenFlow Statistic Manager: It implements the statistics collection by sending statistics requests to all active nodes of the network and storing their responses to the operational store. It also returns information to the Northbound API for the following:
    - node-connector (the port of the connected switch);
    - flow;
    - meter;
    - table;
    - group statistics.

- OpenFlow Switch Manager: It provides information about the nodes (switches) of the network as well as the ports to which they are connected. As long as the controller discovers new network elements, their information is stored in the Switch Manager data tree. The Northbound API can be used to retrieve this information;
- Topology Processing: It saves and handles information about the network devices. When the controller starts, the Topology Manager creates the central node of the topology. Then it is informed regularly of the topology and of any alarms or changes on this tree.

In ODL, the integration of Southbound and Northbound APIs with the data structures used in various controller components is provided through the Model Driven Service Abstraction Layer (MD-SAL). In MD-SAL, any data associated with the operation state of the network is stored in a Document Object Model (DOM), known as a data tree. There are two kinds of data trees in the ODL controller:

- Configuration: It is used to store information that will not be deleted after the controller is restarted. In addition, configurations that are sent to the controller via the REST API are also stored in the configuration tree;
- Operational: Here the controller stores temporary information that arise while performing processes on the network, as well as information that are discovered from the Configuration data field.

3.9.1.4 OPENDAYLIGHT USER INTERFACE (dlux)

The OpenDayLight controller enables the graphical representation of the network state and the ability to change its configuration through its Graphical User Interface (DLUX). Below there is a list with the main applications of DLUX.

- Topology:

The network topology is depicted at: http://<odl-IP>: 8181/index.html#/topology (Figure 15).

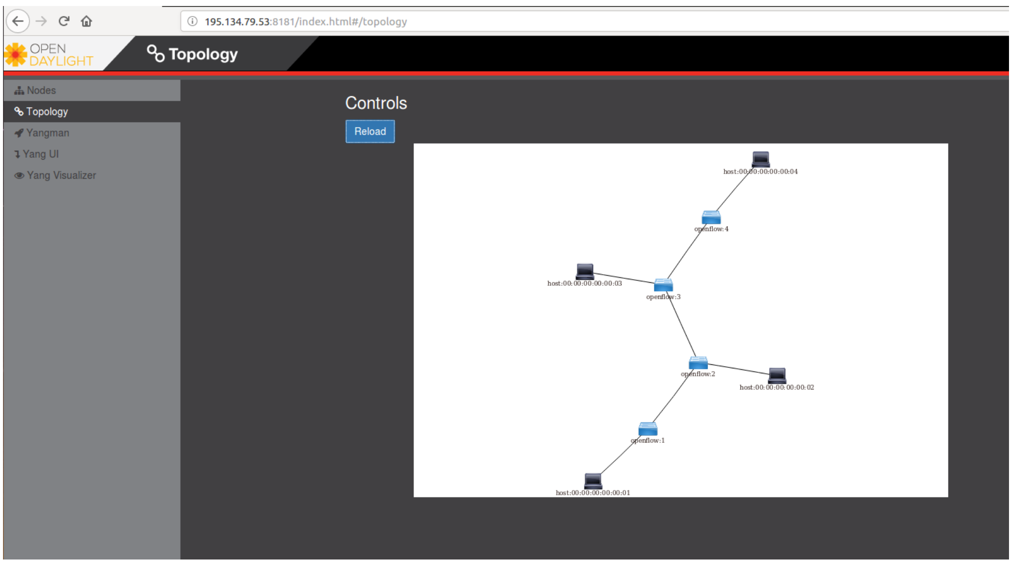

Figure 15: Graphical display of a linear network topology with 4 switches in ODL DLUX

- Nodes:

  The network node statistics can be found at : http://<odl-IP>: 8181/index.html#/node/index (Figure 16).

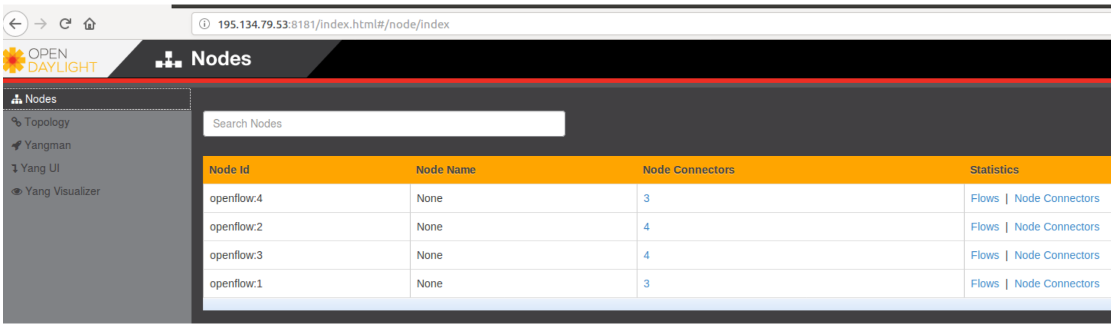

Figure 16: Representation of network node statistics in ODL DLUX

- Yangui:

The Yang UI module facilitates the interaction with the YANG-based MD-SAL datastore. The exposed APIs can be found at : http://<odl-IP>:8181/index.html# /yangui /index . Four possible function can be used:

  - GET to get data from ODL;
  - PUT and POST for sending data to ODL for saving;
  - DELETE for sending data to ODL for deleting.

The xpath for all these operations must be defined. The format of the exchanged data between the applications and the ODL controller is json or xml. The bottom part displays output after a request is successfully sent to the controller. An example for extracting information about the network topology through is show in Figure 17, where the data are displayed in a tree structure and in Figure 18, where data are displayed in json format.

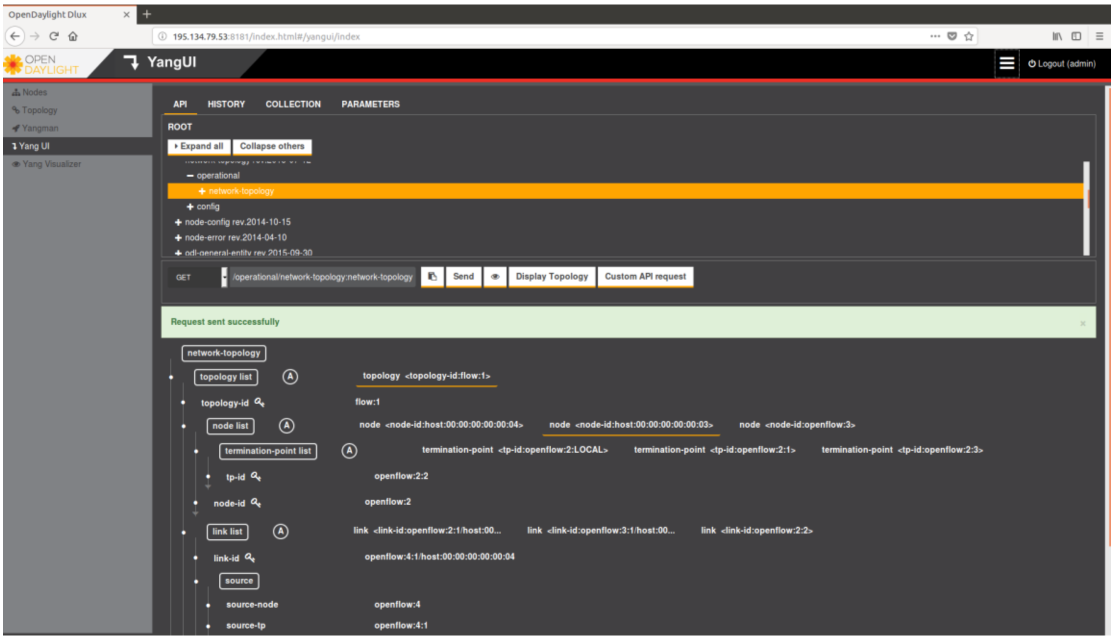

Figure 17: Output of network-topology API

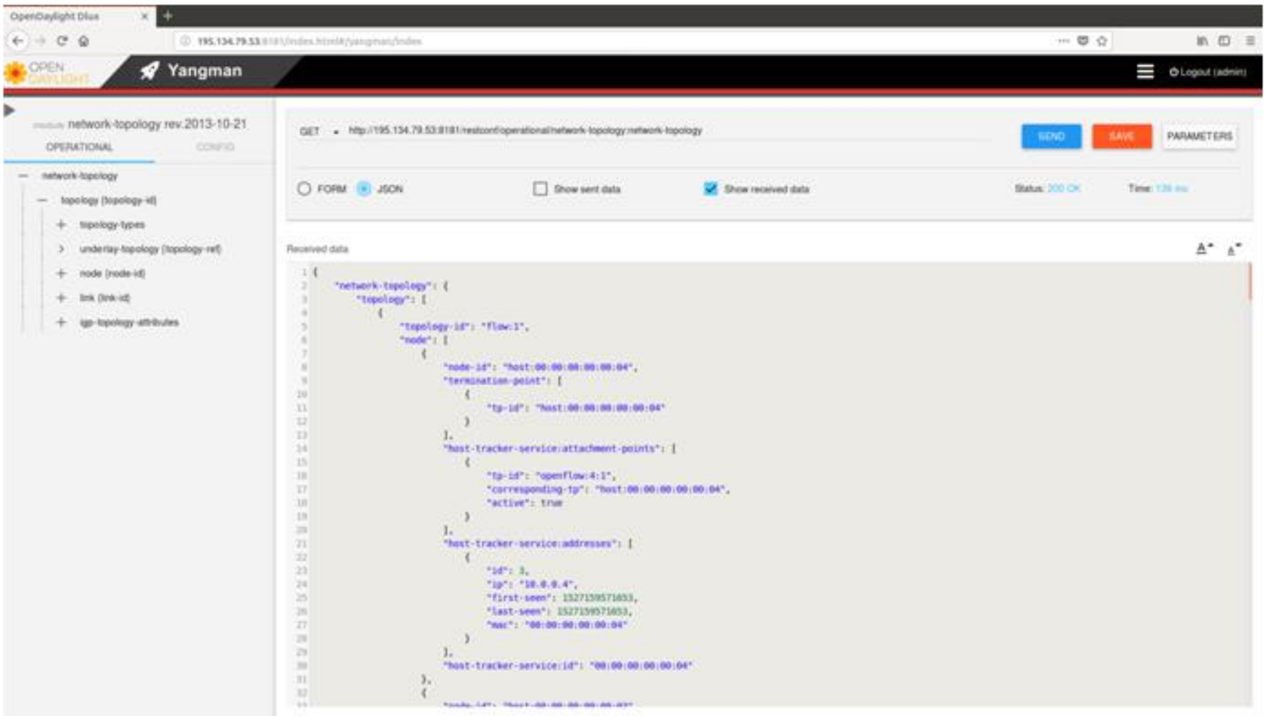

Figure 18: Extracted network topology from ODL DLUX in json format

3.9.2 MININET

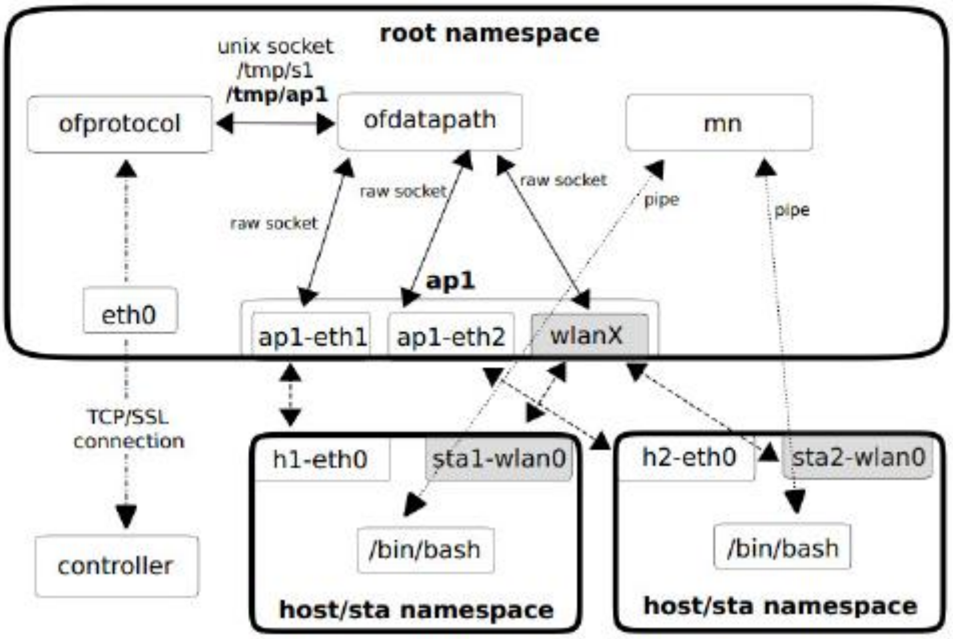

Figure 19: Architecture of a simple two host topology in extended Mininet [11]

Mininet is a network emulator that can emulate and perform the functions of virtual hosts, switches, controllers, and links on a single physical host or virtual machine (VM) [8]. This is accomplished by the implementation of Linux namespaces. Network namespaces are containers for network state. Each network namespace has its own interfaces, ports, and routing tables and provides exclusive processes. A host in Mininet is simply a shell process (e.g. bash) moved into its own network namespace with the unshare system call [9]. They behave like real devices that the user can connect via ssh (network protocol for secure data transfer) and run programs. The packets for the execution of the programs are transferred through virtual Ethernet interfaces that can be configured. Switches support the OpenFlow protocol for high flexibility in routing customization in Software Defined Networking architectures. [10]

Mininet has been extended (Mininet-WiFi) in order to support Wi-Fi stations and access points (AP), and be able to emulate the attributes of a mobile station (i.e. position and movement relative to the access points)[11][12]. Figure 19 depicts the basic components of a simple topology with two hosts and an access point. In this extended version, a wireless station is implemented by adding a WiFi interface is in the regular host implementation, in order to be able to connect to an Openflow switch with AP capabilities. The virtualized APs are created through hostapd3 (Host Access Point Daemon). The current implementation supports the configuration of many AP features, such as the ssid, channel, mode, password, cryptography, etc. Finally, The wireless channel is emulated by the usage Linux Traffic Control (TC) tools.

Mininet has default installed topologies. The command mn starts a minimal topology with two hosts and one switch and provides a CLI (Command Line Interface). This command comes with a set of options in order to be able to emulate any network topology, i.e. the option “ --wifi” enables the Mininet-WiFi.

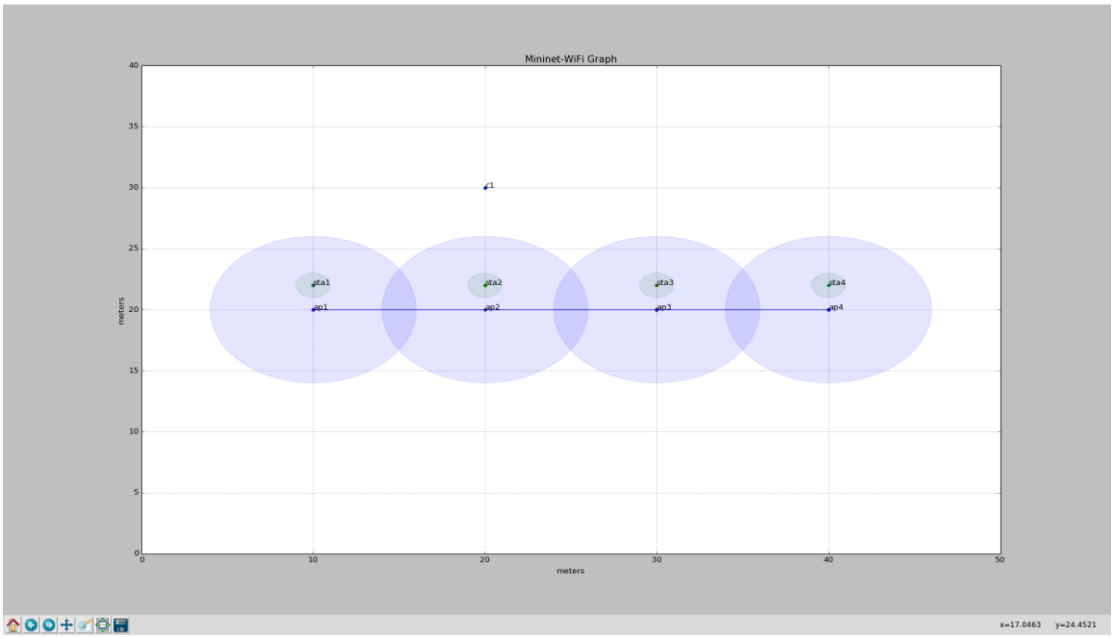

Figure 20: Mininet Wi-Fi graph of a linear network topology with 4 APs and 4 mobile stations

Except the default method that mininet provides for the creation of network topologies, users can easily create custom topologies, using simple Python scripts. Mininet-WiFi provides a graphical display showing the location of network nodes in a graph. Figure 20 shows an exemplary network topology with 4 APs emulated in Mininet-WiFi.

Users can interact with the network nodes, through the functions that are provided by the mininet CLI. Some of the basic functions are shown below with the usage of the linear network shown in Figure 20 ([10][12]).

- nodes: Display of network nodes.

```
mininet-wifi> nodes
available nodes are:
ap1 ap2 ap3 ap4 c1 sta1 sta2 sta3 sta4
```

Figure 21: Display of nodes in Mininet-WiFi

- net: Display of network links.

```
mininet-wifi> net
c1 c1-eth0:ap1-eth0 c1-eth1:ap2-eth0 c1-eth2:ap3-eth0 c1-eth3:ap4-eth0
sta1 sta1-wlan0:wifi
sta2 sta2-wlan0:wifi
sta3 sta3-wlan0:wifi
sta4 sta4-wlan0:wifi
ap1 ap1-eth0:c1-eth0 ap1-wlan1:wifi ap1-eth2:ap2-eth2
ap2 ap2-eth0:c1-eth1 ap2-wlan1:wifi ap2-eth2:ap1-eth2 ap2-eth3:ap3-eth2
ap3 ap3-eth0:c1-eth2 ap3-wlan1:wifi ap3-eth2:ap2-eth3 ap3-eth3:ap4-eth2
ap4 ap4-eth0:c1-eth3 ap4-wlan1:wifi ap4-eth2:ap3-eth3
```

Figure 22: Display of network links in Mininet-WiFi

- dump: display of information of all nodes.

```
mininet-wifi> dump
<RemoteController c1: 192.168.123.1:6653 pid=3648>
<Station sta1: sta1-wlan0:10.0.0.1 pid=3656>
<Station sta2: sta2-wlan0:10.0.0.2 pid=3658>
<Station sta3: sta3-wlan0:10.0.0.3 pid=3660>
<Station sta4: sta4-wlan0:10.0.0.4 pid=3662>
<UserAP ap1: ap1-eth0:192.168.123.2,ap1-wlan1:None,ap1-eth2:None pid=36
64>
<UserAP ap2: ap2-eth0:192.168.123.3,ap2-wlan1:None,ap2-eth2:None,ap2-et
h3:None pid=3668>
<UserAP ap3: ap3-eth0:192.168.123.4,ap3-wlan1:None,ap3-eth2:None,ap3-et
h3:None pid=3672>
<UserAP ap4: ap4-eth0:192.168.123.5,ap4-wlan1:None,ap4-eth2:None pid=36
76>
```

Figure 23: Display of network information in Mininet-WiFi

- distance: calculation of the distance between two network nodes.

```
mininet-wifi> distance ap1 ap4
The distance between ap1 and ap4 is 30.00 meters
mininet-wifi>
```

Figure 24: Display of a distance between two network nodes in Mininet-WiFi

- py <nodeID>.params: Display of information about the APs or the mobile station.

```
mininet-wifi> py ap1.params
{'txpower': [1], 'wlan': ['ap1-wlan1'], 'ssid': ['1-ssid'], 'antennaHei
ght': [1.0], 'driver': 'nl80211', 'inNamespace': True, 'stationsInRange
': {<Station sta1: sta1-wlan0:10.0.0.1 pid=3656> : -44.0}, 'antennaGain
': [5.0], 'mac': ['02:00:00:00:04:00'], 'range': [6.0], 'frequency': [2
.412], 'mode': ['g'], 'associatedStations': [<Station sta1: sta1-wlan0:
10.0.0.1 pid=3656> ], 'position': [10.0, 20.0, 0.0], 'channel': ['1']}
```

```
mininet-wifi> py sta1.params
{'txpower': [1], 'wlan': ['sta1-wlan0'], 'ip': ['10.0.0.1/8'], 'mac': [
'02:00:00:00:00:00'], 'antennaGain': [5.0], 'apsInRange': [<UserAP ap1:
 ap1-eth0:192.168.123.2,ap1-wlan1:None,ap1-eth2:None pid=3664> ], 'rang
e': [1.0], 'frequency': [2.412], 'mode': ['g'], 'associatedTo': [<UserA
P ap1: ap1-eth0:192.168.123.2,ap1-wlan1:None,ap1-eth2:None pid=3664> ],
 'antennaHeight': [1.0], 'position': [10.0, 22.0, 0.0], 'channel': ['1'
], 'rssi': [-44.0]}
mininet-wifi>
```

Figure 25: Display of information about the network nodes in Mininet-WiFi

- dpctl: management utility over the OpenFlow switch.

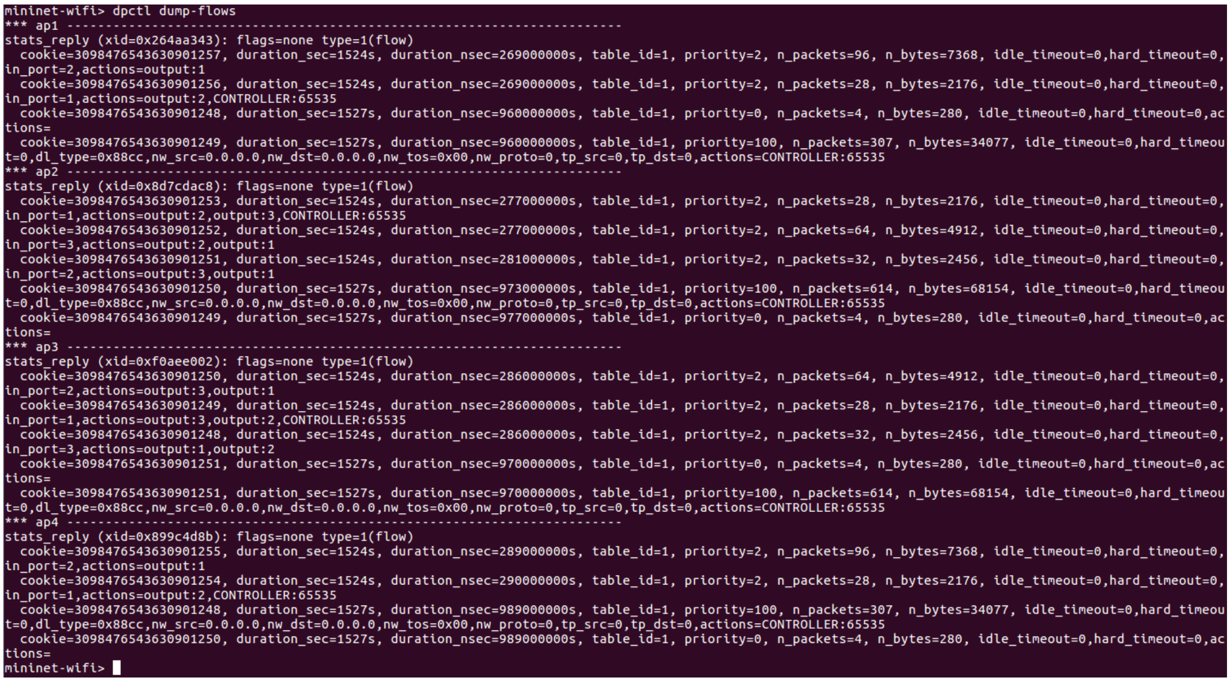

Figure 26: Display of the flow tables in Mininet-WiFi

Mininet has been widely used by network engineers to develop programs and study their results for the following reasons:

- It can create any topology with a simple Python script;
- It is fast and it allows the immediate start of the network;
- It is a project of Open Software, which means that everyone can contribute to it and customize according to his needs;
- It supports the OpenFlow protocol and it is a simple and cost-effective way for experimenting on SDN networks. It is therefore possible for Mininet to customize packet forwarding as long as switches are programmable with OpenFlow;
- It can run real programs since it can perform any program that runs in Linux;
- Provides CLI for debugging and commands for network experimentation.

The key advantage in emulating networks with Mininet is that the code developed for OpenFlow switches and other network devices can be run in a real system with minimal changes. Thus, the administrator can use Mininet to quickly control its application and, when it is ready, implement it on a real network.

3.9.3 Control plane latency in SDN

SDN main concept is the separation of the control from the data forwarding by providing a centralized control over the network through the controller. A key problem with the nature of the SDN architecture is the centralized approach for the placement of the controller. Network topologies with a large number of devices and long distances between them (creating corresponding delays in communication between the switch and the central controller) require the creation of a more distributed architecture that allow the continuing operation of time sensitive applications. The ultimate goal is to create SDNs that are easily expandable and cover large geographical areas.

In the following, an approach for measuring the control plane latency is being discussed. The main idea is the creation of a northbound application over the ODL platform - as a network administrator - that communicates externally with the controller, using the REST APIs. The application implements at first step a mechanism for collecting data about the network topology and at second step a mechanism for sending echo messages to all switches simultaneously, in order to measure the time response of ODL controller.

TOOLS

All the information needed by ODL were exported in the form of json files (alternatively xml could have been used). The formatting of large files, the editing and the mass export of specific data, the jq command was used, which is basically a filter for json files. The jq library contributed to the practical aspect of the application and to its optimization in terms of time and space.

ODL REQUIREMENTS

The version of ODL that was used for this problem was Nitrogen S2. To run the controller, some features must be installed to support the l2switch function, the mdsal model, the restconf protocol, the connectivity capability, and the support for the graphical display and network configuration of ODL (DLUX). Thus, the following features were installed:

- odl-restconf: Allows access to RESTCONF API;
- odl-l2switch-switch: Provides network functionality similar to an Ethernet switch;
- odl-mdsal-apidocs: Allows access to Yang API;
- odl-dlux-all: ODL graphical user interface.

NETWORK TOPOLOGY

A linear topology was chosen to be used for the implementation, in order to simulate the line connection of two train stations (Redhill-Tonbridge) shown Figure 27. The controller communicates with the switches that have AP capabilities through a dedicated control network (Outofband Controller) [3]. The transfer delay was not emulated in the control network, therefore the extracted results depict the processing time of the controller. The number of the APs range from 4 to 256, in order to see how the number of them affects the control plane latency. At http://<odl-IP>: 8181/index.html#/topology we can see the representation of the network topology with 4 APs in ODL DLUX (Figure 28).

Figure 27: The train line that was emulated with Mininet.

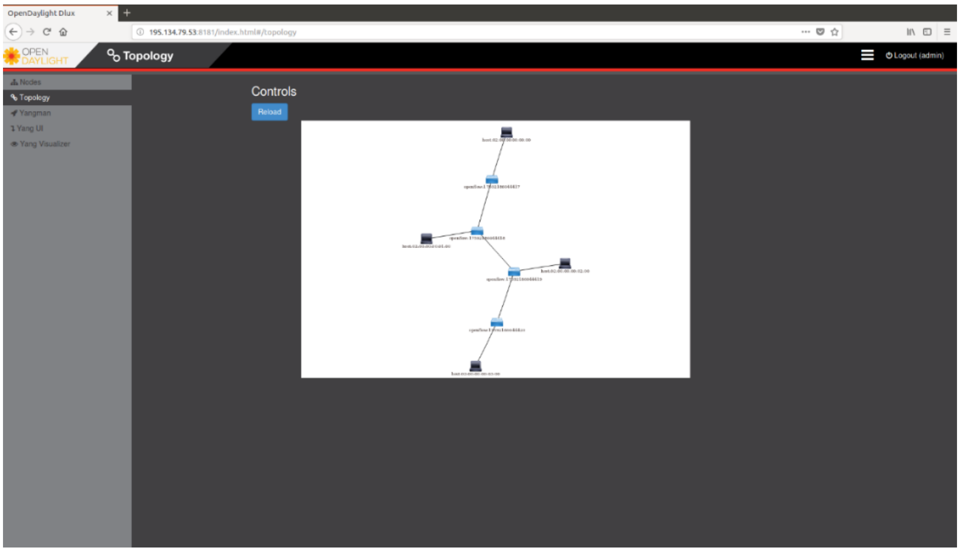

Figure 28: Graphical display of a network topology with 4 APs through ODL DLUX

APPLICATION'S ARCHITECTURE

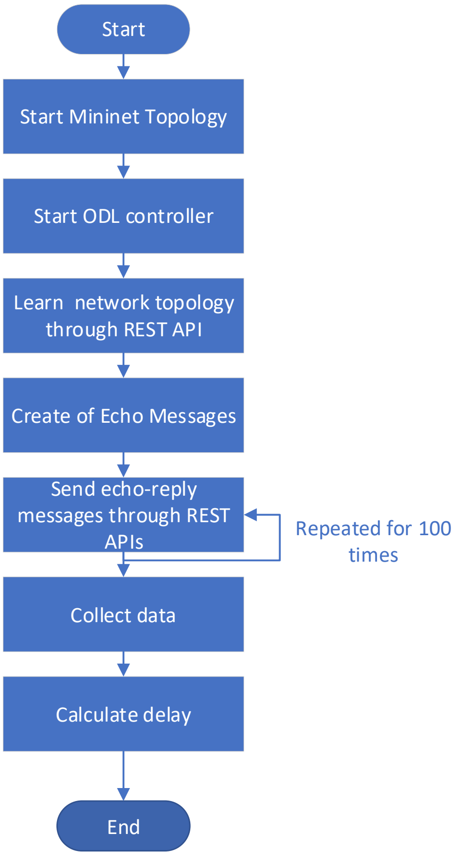

Figure 29: Flowchart of the SDN application

The application takes advantage of the ODL's REST architecture by using the REST API at http://<ODL-IP>:8181/restconf/. In general, modifications are only made in the config state, which automatically updates the operational state of the controller. From the operational state, the network manager receives the desired information.

Python was chosen as the programming language because of its ability of executing bash commands. Python was combined with bash shell scripting to take advantage of the ODL REST API with the usage of the curl command.

In the beginning, through REST API we extract the topology of the network. Specifically, using the curl (bash command), with the appropriate headings and the GET method in the following URL:

curl -u <USERNAME>:<PASSWORD> -X GET -s http://<ODL-IP>:8181/restconf/operational/network-topology:network-topology/

we obtain in json (or xml) format, information about the network. In particular, the obtained json file includes the id of every link in the network, as well as the source and destination node and port of each link. From the above, the number of switches is exported.

For measuring the control plane delay, the application sends simultaneously echo messages to all the switches of the network through the NB REST interface and records the time elapsed for receiving a reply. Thus, curl commands with POST method are employed in the following URL:

curl -u <USERNAME>:<PASSWORD> -H 'Content-Type: application/yang.operation+json' -X POST -s -d @data.json -w %{time_total} -o /dev/null http://195.134.79.53:8181/restconf/operations/sal-echo:send-echo

The REST API above requires an input in json format (data.json), that specifies the destination AP of the echo message. Thus, the application creates as many data.json files as the number of the APs, that will serve as the input of each POST request. The curl commands are executed in parallel and their number is equal to the number of the APs in the network, in order to see how the number of APs affects the time responsiveness of ODL controller. For higher accuracy the above procedure is repeated 100 times, and the delay is considered as the average delay of each repetition. The results for linear topologies with different number of APs are shown Figure 30.

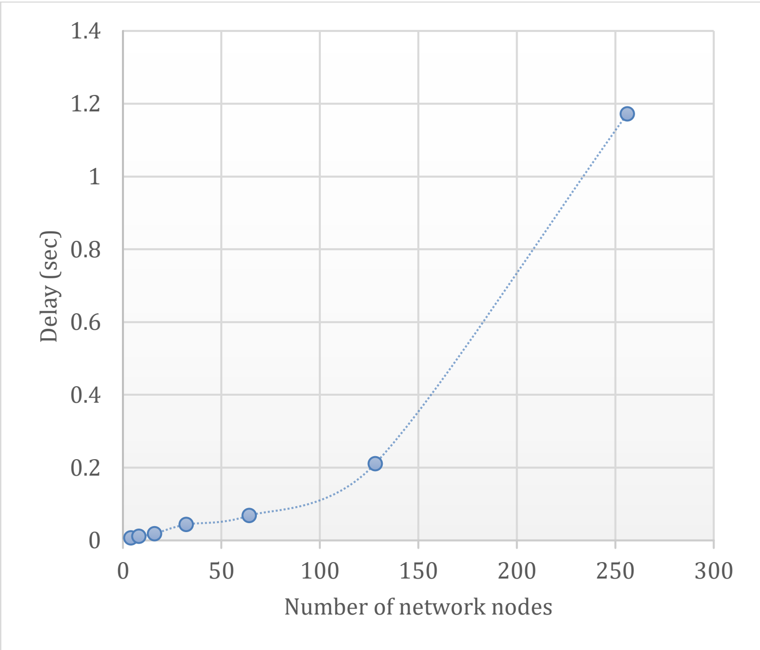

Figure 30: Dependence of processing time of SDN controller on the number of the network nodes

3.10 LiFi technology application

The objective of the chapter is to study and demonstrate LiFi technology application for wireless broadband and IoT connectivity within the rolling stock. LiFi technology has a number of advantages over radio such as its spectral efficiency per unit area as LiFi can provide high data rate within a small coverage area. Among the other advantages very low signal leakage outside LiFi cell and non-interference to radio signals can be mentioned.

The LiFi technology provided by pureLiFi is fully networked, full duplex and can deliver up to 43 Mbps per link. The LiFi access points (with Ethernet connection) and USB dongles (to be connected to the user equipment) are easy to install and use. Moreover, access points used in this project are equipped with extra memory so that contents caching can be deployed and tested. For the IoT connectivity, sensors can be attached to the USB dongle using Raspberry Pi boards.

Within the scope of the project, a proof-of-concept is considered to demonstrate the overall concept.

3.10.1 Scenario overview

The high-level idea behind the proposed scenario is to use LiFi technology as wireless communications access for:

- Broadband coverage inside the rolling stock;
- IoT connectivity inside rolling stock;
- Contents caching for edge cloud, edge computing, etc.;
- High throughput link between rolling stock and train station while train is stopped in the station.

The scenario is illustrated in Figure 31 where each access point is assumed to cover an area of around 4 passenger seats. Full area of the rolling stock can be covered by a network of LiFi access points. Broadband/internet coverage and sensors connectivity within the rolling stock is covered by this LiFi network. Each access point is equipped with up to 64 GB internal memory which can be used for contents caching, edge computing, etc., so that the broadband connection is maintained even when the Internet connection is loose or lost. The study or implementation of such algorithms however is beyond the scope of this project (unless a project partner is willing to perform it).

An extra access point per rolling stock can be considered for the train to train-station high speed link when the train is stopped at the station. The information gathered from sensors during the trip can be saved in the access point and be communicated for off-line processing to the network when the external link is connected.

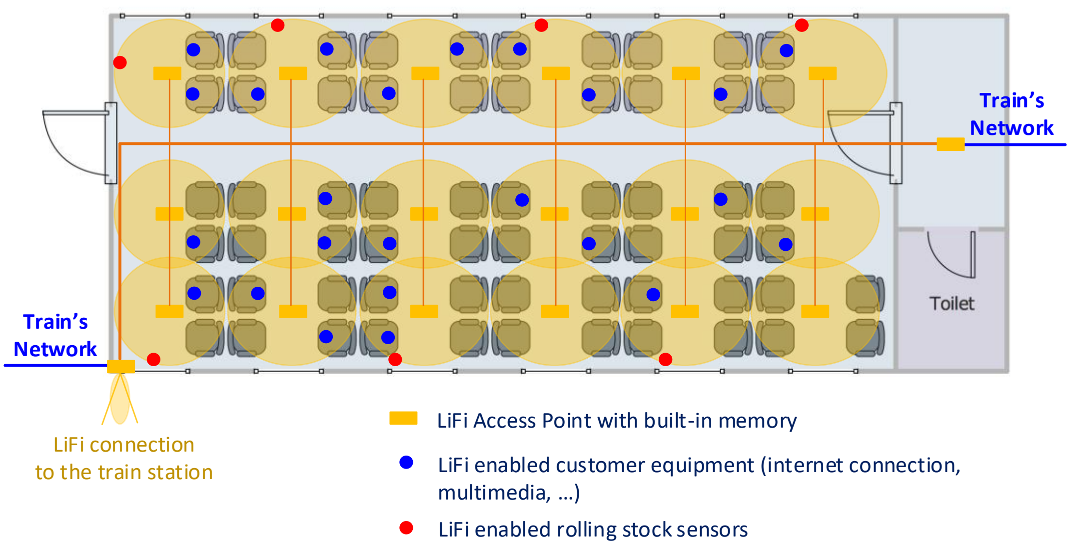

Figure 31 Scenario description: LiFi broadband/IoT coverage within rolling stock

3.10.1.1 *Proof-of-concept*

In case there is a rolling stock available by the project, a setup consisted of 8 access points, covering 8x4 =32 seats, will be implemented for the demonstration purpose. Sensors of interest can be identified and linked to the LiFi dongle using Raspberry Pi boards to make LiFi enabled sensors for the IoT connectivity. The external LiFi link between rolling stock and train station LiFi access point is not considered in this demonstration scenario. The project plan and required efforts are being estimate for 32 seats coverage proof-of-concept and related studies and developments. The setup includes:

- LiFi XC access points with up to 64 GB internal memory (with the luminaire/lamp) x8;
- LiFi XC USB dongle x10;
- Raspberry Pi board (including the battery pack) x3.

3.10.2 Technology specifications

For the proposed scenario pureLiFi's LiFi XC products are being considered. The specifications for LiFi XC can be found in Table 8.

Table 7: LIFI XC specifications

Parameter	Value	Unit	Comment
Downlink peak throughput	43	Mbps	
Upload peak throughput	43	Mbps	
Minimum operational distance	1.0	m	
Maximum operational distance	6.0	m	
No of supported users per AP	8	-	
Inter-AP handover	-	-	Supported
Cell diameter @ 2.5m range	2.8	m	
Cell diameter @ 3m range	3.5	m	
Lamp's required peak voltage	55	V DC	
Lamp's required peak current	700	mA	
Station physical dimensions	85 x 30 x 11	mm	
AP physical dimensions	88 x 88 x 20	mm	
Station weight	42	g	
AP weight	200	g	
Station data interface	-	-	USB2.0
AP data interface	-	-	Gigabit Ethernet

4. Validation of the Use Cases

The following chapter reviews the described train measurement setups and data portfolio, to validate data analytics infrastructure and design for support of selected use cases.

4.1 Review of data portfolio and data processing setup

The preliminary data portfolio for setting up the analytical data pipeline was adopted from In2Rail historic measurement data. The project included experimentation of the tram network of Reims (France), which started in 2016 within the In2Rail project framework. ALSTOM, who designed and commissioned the Reims tram, has instrumented train measurement equipment and one substation in order to collect real-time data on the energy consumption of the different systems on-board and trackside. The measurement data from IN2RAILS serve as a benchmark data for portfolio for In2Dreams data analytics setup and modelling.

Moreover, as shown on Figure 32 below, the ODM Platform from In2Rail was used as an initial infrastructure to build on the In2Dreams data platform.

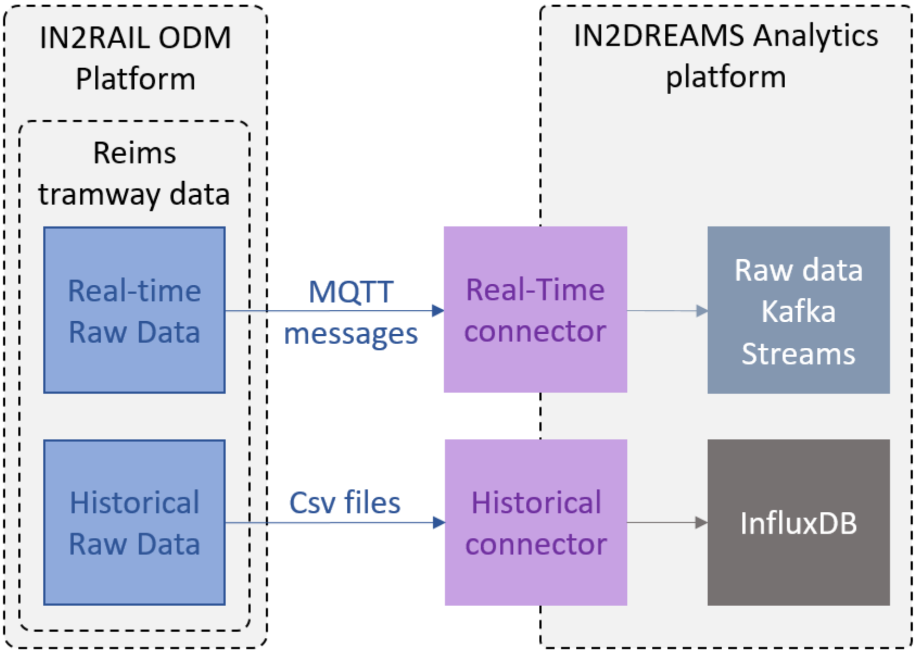

Figure 32: Input connection between Reims tramway data and In2Dreams platform

It is important to note that, ultimately, the In2Dreams platform will be hosted on the same network as the platform hosting the Reims tramway data, thus simplifying the connection between the data-source and the analytics platform.

Data portfolio for train and substation measurements

A review of data portfolio for platform analytics was benchmarked based on Reims substation measurement data, where measurements include Voltage and Current on the power supply line: Voltage: 850 kV, Current I, Current, Current III - with frequency 1s sampling.

The train measurement data is listed in Table 8 and present a large share of measurement data portfolio.

Table 8: Tram data frame with measurement frequency

Measurement	Unit	Frequency
External temperature	°C	0.5 s
Internal temperature	°C	0.5 s
Longitude position of the train	decimal points	0.5 s
Latitude position of the train	decimal points	0.5 s
Station ID or filter pressure	int or pa	0.5 s
Train speed	km/h	0.5 s
3-phase current from first HVAC sensor	A	0.5 s
Current from second traction unit	A	0.5 s
Current from first traction unit	A	0.5 s
CO_2 level inside the train	%	0.5 s
Rheostat current for resistor 22	A	0.5 s
Rheostat current for resistor 21	A	0.5 s
Rheostat current for resistor 12	A	0.5 s
Rheostat current for resistor 11	A	0.5 s
Overhead line or ground power supply	Boolean	
Total measured current	A	0.5 s
Total supplied voltage	V	0.5 s
Current from auxiliary converter	A	0.5 s
3-phase voltage of air conditioning (HVAC)	V	0.5 s
3-phase current from second HVAC sensors	A	0.5 s
Alert1		
Alert2		

Alert3		
Timestamp	int	int

All measurement data (substation and on-board) will be resampled on 1s data frequency as a baseline frequency for data modelling.

4.2 Two main data analytics use cases: on-board & platform data analytics

The In2Dreams project is designed to include multilevel data modelling, including building data infrastructure from the level of measurement, transportation integration and analytics, to business process modelling and data applications design. As In2Dreams project applies IoT data architecture, the on-board train units can eventually be used as connected nodes with edge data processing.

In a traditional cloud computing architecture, the actual processing of data occurs far away from the source. On the contrary, edge computing pushes the generation, collection, and analysis of data close to the point of origin, close to the data source. When considering next generation of smart infrastructures operated using IoT platforms, using edge analytics and distributed data processing poses as an essential part of data operated railways infrastructure.

The approach of pushing data analytics to the edge offers various benefits; low latencies on data transfer and communication, efficiency of resource usage (processing power, memory distribution), localization of data processing based operations and reliability with off-line data-based operation. Especially with mission critical infrastructure, such as railways, attaining resilience and operation reliability is at the crux of technology deployment.

As edge computing relies on proximity to the source of the data, it minimizes incurred data latency, which is advantageous to machine learning and data analytics for a number of reasons. In edge computing, the data doesn't have to make any round trip to the cloud, significantly reducing latency and offering close to real-time, data processing and automated decision-making. If we are to look at AI machine learning as a tool for data-based operation in railways infrastructure, in a cloud infrastructure, the excessive latency could well mean that mobile objects (trains) end up failing to react to any of the many events that unfold on the road on a daily basis. This could potentially lead to optimized operation or in extreme cases critical events.

From data analytics perspective, there are of course limitations to the operation you can do at the edge. Today’s machine learning algorithms are designed to run on powerful servers. Therefore, in the case of railways data analytics, most complex operation such as model training and data modelling still takes place in the cloud, with algorithms trained using large sets of historic data measurements. The data analytics use cases are more extensively analysed in WP5 and WP6, though the main focus of train on-board (edge) data analytics is to perform very short term data analytics, such as short term forecast (1 minute time horizon).

From the perspective of performing data analytics and data-based intelligent railways operation, the infrastructure can be therefore distilled into two main analytical use cases, namely (1) data analytics on on-board unit and (2) data analytics on cloud platform.

Platform data analytics use case

Figure 33 presents the conceptual analytical workflow with the main functionalities and tasks of each subsection within the workflow. The figure explains how the overall QMiner framework setup is conceptualized for data analytics on In2Dreams cloud platform.

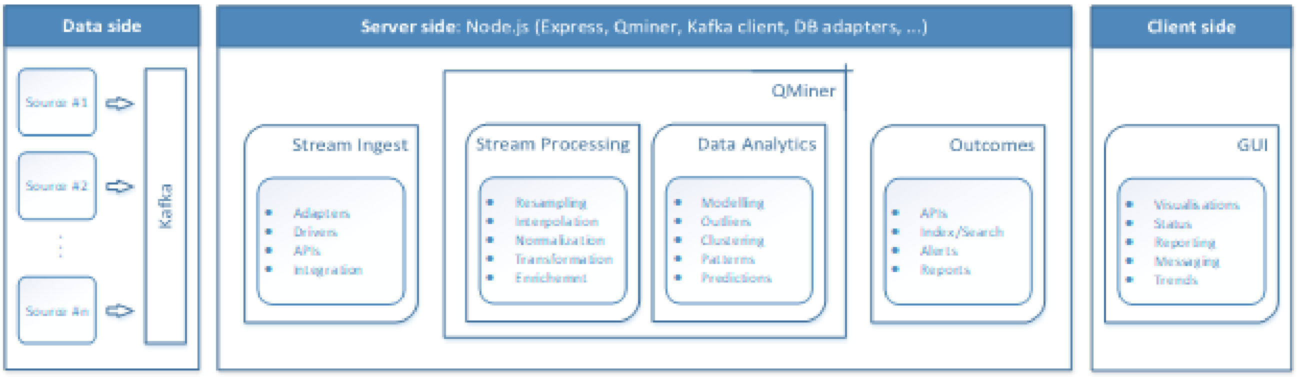

Figure 33: Full stack analytical workflow with QMiner for platform data analytics

Platform data analytics includes full data modelling and operation functionalities, including data modelling and model training based on historic. Moreover, the platform data analytics enables more complex models to be deployed, using external variables (such as weather, events, etc.) in feature engineering. Such approach enables building forecast analytics for more long term prediction horizons (1 day, 1 week ahead).

Edge data analytics use case

Edge infrastructure will support short term prediction models, where prediction horizon will be ranging from 5 seconds to 5 minutes. As processing power is limited, simple methods will be used on the edge analytics framework – a lightweight data analytics node. For the purpose, a streaming linear regression models were employed. Moreover, in modelling short term predictions horizons, such as designed for edge processing, weather or static data features do not introduce new information to the models, since all such information will already implicitly be encoded in past smart-meter readings. The setup of lightweight edge analytics modelling is presented on Figure 34 below.

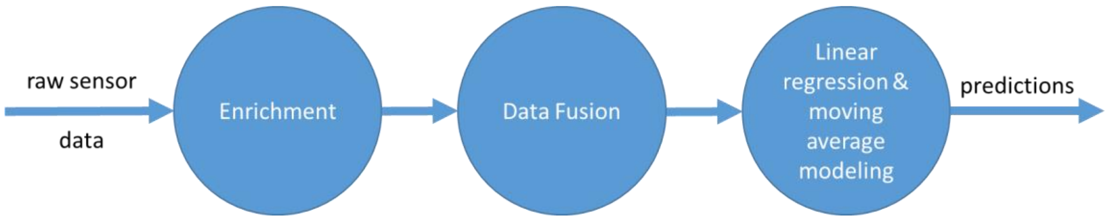

Figure 34: Analytical workflow for on-board data processing

The design of on-board "Light node" is optimized for real-time or close to real-time data operations, focusing on short term prediction horizons in data forecast analytics. Such approach enables low latencies in data processing and optimized data processing speed or HW demands. The data analytics use cases are more extensively analysed in WP5 and WP6, though the main focus of train on-board (edge) data analytics is to perform very short term data analytics, such as short term forecast (1 minute time horizon).

4.3 Testing edge vs. platform data analytics setup

Since the basic project setups are distilled into two main use case scenarios for data analytics, namely edge and cloud data analytics, while the project's data analytics architecture is based on open source framework Qminer (detailed described under WP6, D6.1.). The following chapter focuses on feasibility analysis and verification of using Qminer AI platform to support edge data processing for train on-board unit, as well as benchmark analysis to platform/cloud based data analytics.

The feasibility study included test setup on the edge units – ARM processor was included as a benchmark to basic platform analytics setup. The test procedure therefore included compiling Qminer analytics platform on ARM processors, unit testing of data fusion platform and performance testing on edge unit.

QMiner platform has been initially developed to be compliant with x86 processors and has compiled initially on Windows, later it was adapted for operation on Linux. As the core of Qminer library is written in C++, in order to be deployed on ARM processors it needs to be ported to ARM systems. To port the platform on the ARM processor, configurations and adaptations were performed, mainly related to variables conversions between floats and doubles.

As a simulation of train on-board unit, Raspberry Pi was used for deploy and testing. Finally, executing `npm install qminer` within node.js platform on Raspberry Pi enabled QMiner compilation. The code was optimized and can compile without any problem on ARM; displayed warnings with default compiler settings correspond to QMiner's compact coding style only.

4.3.1 Functional verification and testing for two main analytics use case setups

The initial prototype for data fusion in In2Dreams project has been deployed on a Raspberry Pi 2 unit. A plethora of different unit tests on the ARM processor was performed, to analyse that the critical component - data fusion component is working correctly. Same unit tests have been replicated on x86 processors (desktop machines in Windows and Linux).

The preliminary deployment assumes 3 types of data – smart meter energy data, weather forecast data and static features data (such as date-time features and holiday related features). Each node component was tested, as well as final data fusion component which connects all data nodes (objects). Results of unit tests are listed in Appendix 1:

All basic components, including the final data merger component yield successful unit tests. The results of functional (unit) testing shows platform deployed on simulated edge processor (Raspberry Pi) is operating functionally compliant to the scenario on the cloud. Further analysis has been performed to compare performance benchmark of two main analytical setups.

4.3.2 Data analytics performance benchmark and verification

The analysis was conducted with a full setup data analytics, including feature generation, data fusion and linear regression streaming model on a single node. Test setup for edge processing has implemented the following stream aggregates on the features:

- voltage
    - 200ms sliding window: exponential moving average;

- 1s sliding window: variance, moving average;
- 5s moving window: exponential moving average, variance, moving average, min, max.
- frequency
  - 200ms sliding window: exponential moving average;
  - 5s sliding window: exponential moving average, variance.

A streaming linear regression model was deployed with the following attribute configuration (psp_v = voltage, f1 = frequency; aggregates are defined with feature_name|aggregate_type| sliding_window_size_in_ms; feature vectors also include timeDiff – difference within time – feature extractors):

```
attributes: [
    { "time": 0,
      "attributes": [
        { type: "value", "name": "psp_v|ma|5000" },
        { type: "value", "name": "psp_v|ema|60000" },
        { type: "value", "name": "f1|variance|5000" },
        { type: "value", "name": "psp_v|variance|5000" },
        { type: "value", "name": "psp_v|min|5000" },
        { type: "value", "name": "psp_v|max|5000" },
        { type: "timeDiff", "name": "psp_v|ma|1000", "interval": 50 },
        { type: "timeDiff", "name": "psp_v|ma|1000", "interval": 250 }
      ]
    },
    { "time": -50,
      "attributes" : [
        { type: "timeDiff", "name": "psp_v|ma|1000", "interval": 50 }
      ]
    },
    { "time": -100,
      "attributes" : [
        { type: "timeDiff", "name": "psp_v|ma|1000", "interval": 50 }
      ]
    },
    { "time": -150,
      "attributes" : [
        { type: "timeDiff", "name": "psp_v|ma|1000", "interval": 50 }
      ]
    },
    { "time": -200,
      "attributes" : [
        { type: "timeDiff", "name": "psp_v|ma|1000", "interval": 50 }
```

```
        ]
      }
    ]
```

Figure 35: Test JSON configuration of linear regression model features in the data fusion/modelling infrastructure

As a benchmark, performance tests have been conducted on Raspberry Pi 2 and a server with Intel Xeon (R) CPU E5-2667 v2 @3.30GHz with 128GB of RAM, running Windows Server 2013 (64 bit). Results are depicted osn Figure 35.

Figure 36 presents latencies for processing data samples on server, using the two setups: light node and full node. We can see (note that scale is logarithmic, results are in milliseconds) that Light node is able to process one data record (message) in approximately 0,31 ms, while full node processes on data record in 110 ms.

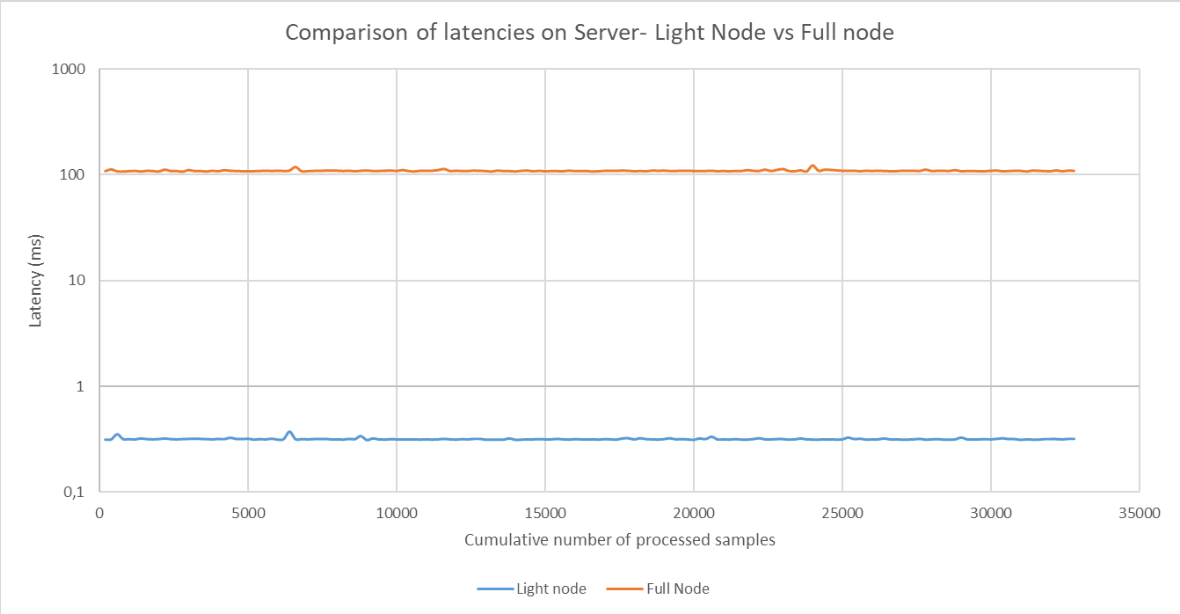

Figure 36: comparison of latencies on a server for Light node and Full node

Figure 37 presents test analysis for latencies for processing data samples on Raspberry Pi (note that results are in milliseconds seconds), using the two setups: light node and full node. The Light node is able to process ono data message in approximately 3,7 s, while full node processes yield processing speed of 950 ms per message (record).

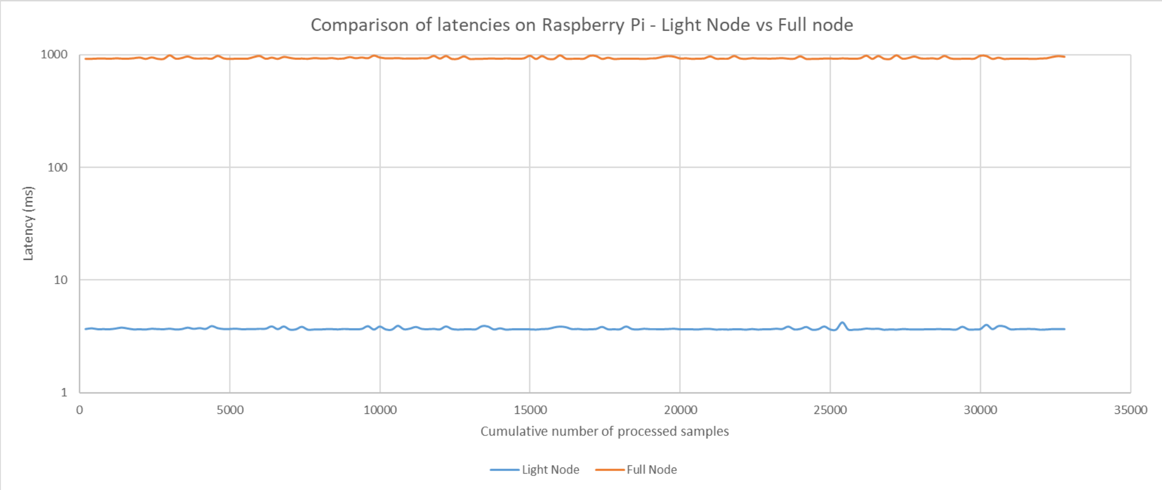

Figure 37: Comparison of latencies on a Raspberry Pi for Light node and Full node

Table 9: Latency measurements Lite node / Full node on server and Raspberry Pi on batch of 200 data samples

	Server [ms]	Raspbery Pi [ms]	Ratio
Light node	**0,31**	**3,68**	11,87
Full Node	**110,00**	**950,00**	8,64
Ratio	354,84	258,15	

Test results indicate that a single light node should be able to process 100 models or more on the on-board unit, if the data rate would be 2 messages/s (one sensor reading per 0.5s), considering that the processing speed is equal to Raspberry Pi. Though, considering the transport time, transformation, integration delays (API) and other, the final aggregated response time should be substantially longer for platform data processing. Additionally, increasing processing power on edge can increase response time to match specific use case demands. From the Figure 36 and Figure 37 we can also note that performance does not deteriorate through time, offering light node a stabile solution for deployment on on-board train units.

Edge (Raspberry Pi) throughput for the full node is approximately 1 messages/s, which is significantly less (more than 200 times slower), compare to throughput for the light node, which is 270 messages/s. Similarly, can be observed comparing performance on server; full node/light node ratio is 110ms/0,31ms, which is approximately 300 times slower for complex model. Comparing light node to full node, the processing time is therefore significantly faster for light node, both on Raspberry Pi, as well as on the server. Moreover, since the light node uses regression models, it is invariant to changes in train route adaptations, changes in traffic light regime, does not need model training, etc., which offers additional advantage in using light node for data processing on train on-board unit.

Figure 7 shows test of latencies with running multiple full node pipelines in parallel. The results indicate constant latency time invariant of number of full node pipelines until a certain threshold (when the hardware resources are 100% utilized), afterwards the performance decreases linearly with the increase of full node pipelines. From the figure we can observe that this threshold in our experiments was 70 full node pipelines. Based on the results from Table 9, we can estimate that we could run approximately 300 times more light node pipelines on the same hardware, since complex forecasting models are removed from the pipeline and are replaced with light node streaming model (e.g. incremental linear regression) from QMiner. After this point (threshold), in order to avoid the loss of (latency) performance, distributed setup (cluster) where we use several servers would be needed, or increasing processing resources on the single server. Note that this are the results for running full node pipelines in parallel (running multiple service instances at once). If the measurement frequency is lower than the processing latencies, one instance can still handle several pipelines sequentially.

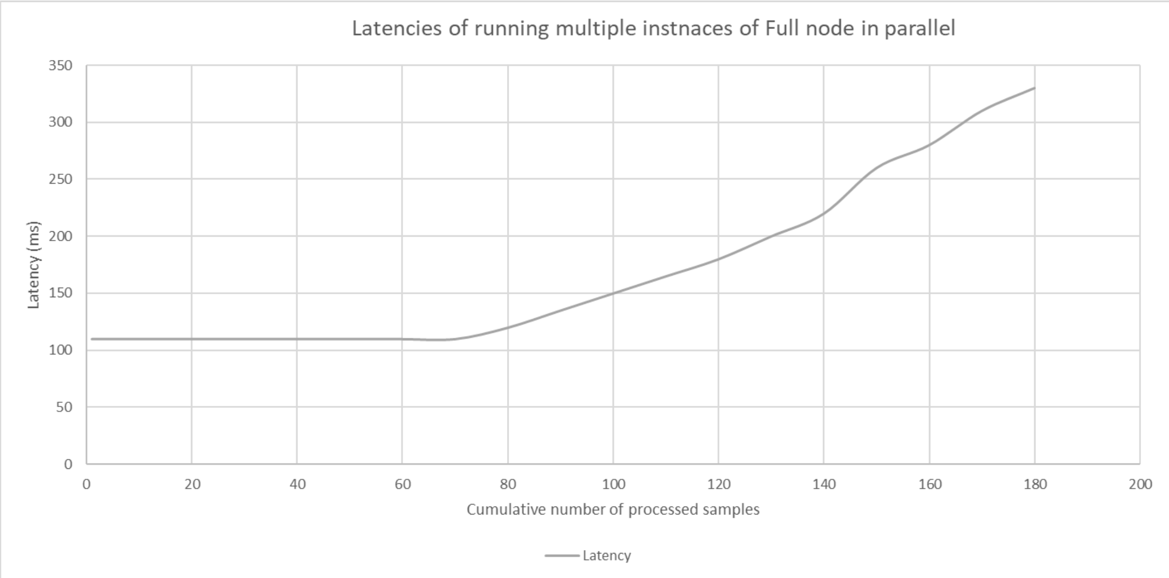

Figure 38: Latencies for running multiple instances of full node in parallel

Performance benchmark of light node and full node on edge and platform computing, shows significant advantage in processing speed for light node, which will enable to deploy short term forecasting modelling for on-board units. On the other hand, full node modelling enables building complex forecasting models with longer time horizon prediction, to support a wide plethora of analytical use cases on the platform.

4.4 Limitations and support for extensions

Both presented use cases (platform data analytics and edge data analytics use case) have some limitations that be considered when designing a solution for a specific end application use case. In this chapter, we list some of the main limitations and possible extensions of both scenarios to be considered in further application.

4.4.1 Platform data analytics use case

- **Latency**: one of the primary advantages of cloud computing is the ability to perform analytics on with the advanced state of the art (usually computationally complex) algorithms, which requires powerful servers. The main drawback here of course is the latency issue, since the request is sent from the client side to the server, where the computation is done and the result is then sent back to the client side. This might work for some use cases, where the analysis has to be done once in a while and speed is not crucial, but it is not a good design for high velocity real-time application (e.g. very short term load forecasting). One way to soften this issue is to speed up the computational time on the cloud server (with powerful machines, optimizations, parallelization …), but performing the computation directly on the end device when possible might be best (edge computing);

- **Connectivity**: another major possible drawback of the cloud computing architecture is the connectivity issue. Since all the processing is done on the remote server, the client is bound to the internet connectivity in order to work. For some use cases, for example infrastructure monitoring and analytics, where there is limited internet connections (e.g. rural parts or hard-to-reach areas due to terrain). In order to overcome this problem to some extent, client application can cache some of the data and use it when the connectivity enhanced. Even better solution, if it is possible, is complete edge computing where the entire computation is done on the same device where data is produced, which means that we are not bound to internet connectivity and outages;
- **Privacy and security**: lately privacy concerns have been a big issue globally. Sending and storing personal information to the remote storage can have privacy, legal and security ramifications. We have to take into account several regulations when dealing which such data. In addition, different countries also have different regulations that makes this issue even more complicated. Collecting data and performing entire computation only locally, on the edge, might be the solution.

4.4.2 Edge data analytics use case

- **Network traffic**: in order to reduce the latency, we have to reduce (or completely eliminate) the network traffic used by the end device and reduce the risk of a data bottleneck. By doing this, we lose the advantage of using other streaming data sources and services accessible over the internet. This means that we are more or less bound only to the data stream that we are collecting on the end device where the processing pipeline is deployed. Some of the external data that can improve the performance of the analytical component can be imported to the end device, but we are more or less bound to the static and not real-time data sources (e.g. holidays information for the region of end device);
- **Computing power**: by moving all the computation to the end device, we are bound to the computational resources of this unit, which is usually much less powerful than a server machine, let alone a cluster of machines. This means that we have to use the resources very efficiently, by optimizing the processing pipeline and by using low complex, yet efficient tools and algorithms (e.g. incremental linear regression). QMiner is designed in a very efficient C++ code and includes several efficient data processing algorithms;
- **Storage**: another limitation of the edge computing is the storage. While cloud servers are perfect for storing large amount of data, end devices usually have a very limited amount of storage. In addition, reading and writing can also cause a bottleneck, therefore efficient data management (data caching, buffering, streaming) and online incremental algorithms, which do not require access to historical data, are preferable. QMiner is a full stack platform that includes all the necessary tolls for designing light weigh streaming pipeline. It includes everything from efficient storage, data aggregation, feature extraction to online modelling. Consequently, big data mining and complex analytics is not possible on the edge, but it is perfect for supporting use cases where a high velocity, robust models are needed (e.g. very short term predictions on board units).

Overall, we can see that both setups offer various advantages with described limitations which needs to be considered. Yet, many of the successful applications are in fact hybrids, and uses the greatest benefits from the both use cases. Therefore, the designed architecture can support either use case or a hybrid combination of the two setups.

References

[1] 5G Vision. The 5G Infrastructure Public Private Partnership: the next generation of communication networks and services. 2015. [Online] https://5g-ppp.eu/wp-content/uploads/2015/02/5G-Vision-Brochure-v1.pdf

[2] GSMA Intelligence, Understanding 5G:Perspectives on future technological advancements in mobile, Dec.2015. [Online]: https://www.gsmaintelligence.com/research/?file=c88a32b3c59a11944a9c4e544fee7770&download

[3] ETSI GS NFV-SWA 001 V1.1.1 (2014-12), “Network Functions Virtualisation (NFV); Virtual Network Functions Architecture”, 2014.

[4] Wang, Y., Haas, H., "Dynamic Load Balancing With Handover in Hybrid Li-Fi and Wi-Fi Networks," in Journal of Lightwave Technology, vol. 33, no. 22, pp. 4671-4682, Nov.15, 15 2015

[5] CHINA mobile White paper, C-RAN: The Road Towards Green RAN, [Online]: http://labs.chinamobile.com/cran/

[6] Huawei, Building 4G on High-Speed Railways, [Online]: http://www1.huawei.com/en/static/HW-371912.pdf

[7] Tzanakaki, A., et al., "5G infrastructures supporting end-user and operational services: The 5G-XHaul architectural perspective," 2016 IEEE International Conference on Communications Workshops (ICC), Kuala Lumpur, 2016, pp. 57-62.

[8] Tzanakaki, A., et al., “Virtualization of heterogeneous wireless-optical network and IT support of cloud and mobile services,” IEEE Commun. Mag., 51(8), pp. 155-161, 2013.

[9] G. Brebner, "Programmable hardware for high performance SDN," Optical Fiber Communications Conference and Exhibition (OFC), 2015, Los Angeles, CA, 2015, pp. 1-3.

[10] C. Chen, D. A. Basnayaka and H. Haas, "Downlink Performance of Optical Attocell Networks," in Journal of Lightwave Technology, vol. 34, no. 1, pp. 137-156, Jan.1, 1 2016

[11] ETSI GS NFV-EVE 005 V1.1.1, Network Functions Virtualisation (NFV); Ecosystem; Report on SDN Usage in NFV Architectural Framework. 2015-12.

[12] H. Jin, D. Pan, J. Liu and N. Pissinou, "OpenFlow-Based Flow-Level Bandwidth Provisioning for CICQ Switches," in IEEE Transactions on Computers, vol. 62, no. 9, pp. 1799-1812, Sept. 2013.

[13] Y. Fang and I. Chlamtac, “Analytical Generalized Results for Handoff Probability in Wireless Networks,” IEEE Trans. Commun., vol. 50, no. 3, Mar. 2002, pp. 369–99.

[14] X. Chen et al., “Analyzing and modeling spatio-temporal dependence of cellular traffic at city scale,” in proc. of IEEE ICC, pp.3585-3591, 2015

[15] Aditi Malik and Preeti Singh, “Free Space Optics: Current Applications and Future Challenges,” International Journal of Optics, vol. 2015, Article ID 945483, 7 pages, 2015. doi:10.1155/2015/945483

[16] A. Hatziefremidis, K. E. Zarganis and K. E. Zoiros, "Flexible high power Free Space Optical system for aircraft to ground communications," 2014 16th International Conference on Transparent Optical Networks (ICTON), Graz, 2014, pp. 1-4.doi: 10.1109/ICTON.2014.6876612

[17] www.mrv.fso.com

[18] www.lightpointe.com

[19] Keun Son, Shiwen Mao, “A survey of free space optical networks”, digital communications and networks, Elsevier, 2017

[20] R. Paudel, Z. Ghassemlooy, H. Le-Minh and S. Rajbhandari, "Modelling of free space optical link for ground-to-train communications using a Gaussian source," in IET Optoelectronics, vol. 7, no. 1, pp. 1-8, February 2013.doi: 10.1049/iet-opt.2012.0047

[21] R. Paudel, Z. Ghassemlooy, H. Le-Minh, S. Rajbhandari and B. Livingstone, "Investigation of FSO ground-to-train communications in a laboratory environment," 2011 Second Asian Himalayas International Conference on Internet (AH-ICI), Kathmandu, 2011, pp.

[22] Achour M., 2002b, "Simulating atmospheric free space optical propagation; Part II, Haze, fog and low clouds attenuations", in SPIE proceedings, vol. 4873, 2002.

[23] Al-Habash M.A., Fisher K.W., Cornish C.S., Desmet K.N. and Nash J., 2002, "Comparison Between Experimental and Theoretical Probability of Fade for Free Space Optical Communications", Proc. SPIE, 4873, pp. 79-89.

[24] Bekkali A., Naila C.B., Kazaura K., Wakamori K. and Matsumoto M., 2010, "Transmission Analysis of OFDM-Based Wireless Services Over Turbulent Radio-on-FSO Links Modeled by Gamma-Gamma Distribution", IEEE Photonics Journal, vol. 2, no. 3, pp. 509-520.

[25] Dimitrov S., Sinanovic S. and Haas H., 2012, "Clipping Noise in OFDM-Based Optical Wireless Communication Systems", IEEE Transactions Commun., vol. 60, no. 4, pp. 1072-1081.

[26] Fornel F., Al Naboulsi M. and Sizun H.: "Fog attenuation prediction for optical and infrared waves", in Journal SPIE.

[27] Kampouraki M.N., Stassinakis A.N., Nistazakis H.E., Chronopoulos G.G., Tsigopoulos A.D., Fafalios M.E. and Tombras G.S., 2014, “ Experimental and theoretical bit rate estimation of turbulent FSO link over the maritime area at Piraeus port”, 6th International Conference from Scientific Computing to Computational Engineering IC-SCCE.

[28] Keddar D. and Arnon S., 2004, "Urban Optical Wireless Communication Networks: the Main Challenges and Possible Solutions", IEEE Opt. Com., pp. S1-S7.

[29] Majumdar A.K., 2005, "Free-Space Laser Communication Performance in the Atmospheric Channel", J. Opt. Fiber Commun. Rep., vol. 2, pp. 345-396.

[30] Ninos M.P., Nistazakis H.E., Stassinakis A.N., Tsigopoulos A.D. and Tombras G.S., 2015a, "BER estimation for a PSK OFDM RoFSO serially relayed link with pointing errors over gamma gamma turbulence channels", MIC electrical.

[31] Nistazakis H. E. and Tombras G. S., 2012, "On the use of wavelength and time diversity in optical wireless communication systems over gamma-gamma turbulence channels", Opt. Laser Technol.

[32] Nistazakis H.E, Stassinakis A.N., Tombras G.S., Muhammad S.Sheikh and Tsigopoulos A.D., 2014a, "K Modeled Turbulence and Nonlinear Clipping for QAM OFDM with FSO and Fiber Serially Linked", 20th International Conference of Microwave, Radar and Wireless Communications MIKON-2014.

[33] Rashed A., 2011, "High transmission bit rate of multi giga bit per second for short rage optical wireless access communication networks", International Journal of Advanced Science and Technology, vol. 32.

[34] Stassinakis A.N., Nistazakis H.E., Peppas K.P. and Tombras G.S., 2013a, "Improving the Availability of Terrestrial FSO Links Over Log Normal Atmospheric Turbulence Channels using Dispersive Chirped Gaussian Pulses", Elsevier Optics & Laser Technology, vol. 54, pp 329-334.

[35] Won Y.Y., Kwon H., Hong M. and Han S., 2009, "1.25Gb/s wire line and wireless data transmission in wavelength reusing WDM Passive optical Networks", Microwave and Optical Technology Letters, vol. 51, no. 3, pp 627-629.

[36] www.seagate.com/gb/en/our-story/data-age-2025/

[37] D. M West, “How 5G technology enables the health Internet of Things”, Brookings, 2016

[38] M. Anastasopoulos et al., "ICT platforms in support of future railway systems", in Proc. of TRA 2018

[39] F. Musumeci et al., "Optimal BBU Placement for 5G C-RAN Deployment Over WDM Aggregation Networks," J. Lightwave Technol. 34, 1963-1970 (2016_

[40] G. Paulraj et al., Resource-aware virtual machine migration in IOT cloud, *Future Gen. Comp. Sys*., 2018

[41] A. Tzanakaki, Optical networking: An important enabler for 5G (invited), ECOC 2017

[42] Q. Zheng et al, “WiBench: An Open Source Kernel Suite for Benchmarking Wireless Systems”, IEEE ISWC 2013

[43] J. Perelló *et al*., "All-optical packet/circuit switching-based data center network for enhanced scalability, latency, and throughput," in *IEEE Network*, 2013.

[44] A. Tzanakaki et al., "Wireless-Optical Network Convergence: Enabling the 5G Architecture to Support Operational and End-User Services," IEEE Comm. Mag., vol. 55, no. 10, pp. 184-192, Oct. 2017

[45] “Software-Defined Networking”, https://www.opennetworking.org/sdn-definition/

[46] “Openflow”, http://flowgrammable.org/sdn/openflow/

[47] “OpenFlow Switch Specification”, Version 1.5.1 (Protocol version 0x06), Open Networking Foundation,2015

[48] “OpenDayLight”, https://www.opendaylight.org/

[49] “OpenDayLight Project”, https://en.wikipedia.org/wiki/OpenDaylight_Project

[50] “REST (representational state transfer)”, http://searchsoa.techtarget.com/definition/REST

[51] “OpenDayLight Wiki”, https://wiki.opendaylight.org/view/Main_Page

[52] “Mininet Overview”, http://mininet.org/overview/

[53] Bob Lantz, Brandon Heller, and Nick McKeown. 2010. “A network in a laptop: rapid prototyping for software-defined networks”. In Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks (Hotnets-IX). ACM, New York, NY, USA, Article 19 , 6 pages

[54] “Introduction to Mininet”, https://github.com/mininet/mininet/wiki/Introduction-to-Mininet

[55] Ramon R.Fontes, Samira Afzal, Samuel H. B. Brito, Mateus A. S. Santos, Christian Esteve Rothenberg, “Mininet-WiFi: Emulating Software-Defined Wireless Networks”, Network and Service Management (CNSM), IEEE, Nov. 2015.

[56] “Mininet-WiFi”, https://github.com/intrig-unicamp/mininet-wifi.

[57] “Sensors & Metering Design”, In2Rail project, Deliverable D11.3.

Appendix 1: Results of Unit Tests

```
pi@raspberrypi:~/EnergyForecast/nrgStream-fusion $ npm test

> nrgstream-fusion@0.0.1 test /home/pi/EnergyForecast/nrgStream-fusion
> mocha tests

  streamFusion
    initialization
      ✓ base saved
      ✓ check number of fusion objects
      ✓ check number of stores
      ✓ check fusion model name
      ✓ check features topic name
      ✓ check types of nodes
      ✓ global config correct
      ✓ feature vector ok (43ms)

  streamFusion - extended
    large data test
      ✓ handling large data node time offsets (5282ms)

  streamingNode
    initialization
      ✓ base saved
      ✓ aggregates initialized
      ✓ config saved
      ✓ fusionNodeI correctly saved
      ✓ callback function should be set
      ✓ parent saved
      ✓ buffer empty
      ✓ buffer position is 0
      ✓ master flag set correctly
      ✓ isMaster function
      ✓ connectToKafka function exists
      ✓ broadcastAggregates function exists
      ✓ createAggregates function exists
```

```
      ✓ offsetExists function exists
      ✓ deleteObsoleteRows function exists
      ✓ checkDataAvailability function exists
      ✓ setSlaveOffset function exists
      ✓ getOffsetTimestamp function exists
      ✓ setMasterOffset function exists
      ✓ getAggregates function exists
      ✓ getPartialFeatureVector function exists
      ✓ master set correctly
      ✓ master offset set correctly
      ✓ slave offset set correctly: no data

streamingEnergyNode
  initialization
      ✓ base saved
      ✓ check if store exists
      ✓ check store structure
      ✓ aggregates initialized - number
      ✓ aggregates initialized - key names
      ✓ config saved
      ✓ fusionNodeI correctly saved
      ✓ callback function should be set
      ✓ parent saved
      ✓ buffer empty
      ✓ buffer position is 0
      ✓ master flag set correctly
      ✓ isMaster function
      ✓ connectToKafka function exists
      ✓ broadcastAggregates function exists
      ✓ createAggregates function exists
      ✓ offsetExists function exists
      ✓ deleteObsoleteRows function exists
      ✓ checkDataAvailability function exists
      ✓ setSlaveOffset function exists
      ✓ getOffsetTimestamp function exists
      ✓ setMasterOffset function exists
      ✓ getAggregates function exists
      ✓ getPartialFeatureVector function exists
      ✓ master set correctly
```

```
    ✓ master offset set correctly
    ✓ slave offset set correctly: no data
  data insertion
    ✓ data record record saved correctly
    ✓ stream aggregates calculated correctly for 1 insertion
    ✓ 11 more data insertions resampled by hour
    ✓ correct aggregate values
    ✓ correct resampling if interval is higher than 15 minutes
    ✓ coorect partial feature vector (1043ms)

streamingWeatherNode
  initialization
    ✓ base saved
    ✓ check if store exists
    ✓ check store structure
    ✓ aggregates initialized - number
    ✓ aggregates initialized - key names
    ✓ config saved
    ✓ fusionNodeI correctly saved
    ✓ callback function should be set
    ✓ parent saved
    ✓ buffer empty
    ✓ buffer position is 0
    ✓ master flag set correctly
    ✓ isMaster function
    ✓ connectToKafka function exists
    ✓ broadcastAggregates function exists
    ✓ createAggregates function exists
    ✓ offsetExists function exists
    ✓ deleteObsoleteRows function exists
    ✓ checkDataAvailability function exists
    ✓ setSlaveOffset function exists
    ✓ getOffsetTimestamp function exists
    ✓ setMasterOffset function exists
    ✓ getAggregates function exists
    ✓ getPartialFeatureVector function exists
    ✓ master set correctly
    ✓ master offset set correctly
    ✓ slave offset set correctly: no data
  data insertion
```

```
    ✓ data record record saved correctly
    ✓ data availability - offset does not exist
    ✓ correct partial feature vector

streamingStaticNode
  initialization
    ✓ base saved
    ✓ check if store exists
    ✓ check store structure
    ✓ aggregates initialized - number
    ✓ aggregates initialized - key names
    ✓ config saved
    ✓ fusionNodeI correctly saved
    ✓ callback function should be set
    ✓ parent saved
    ✓ buffer empty
    ✓ buffer position is 0
    ✓ master flag set correctly
    ✓ isMaster function
    ✓ connectToKafka function exists
    ✓ broadcastAggregates function exists
    ✓ createAggregates function exists
    ✓ offsetExists function exists
    ✓ deleteObsoleteRows function exists
    ✓ checkDataAvailability function exists
    ✓ setSlaveOffset function exists
    ✓ getOffsetTimestamp function exists
    ✓ setMasterOffset function exists
    ✓ getAggregates function exists
    ✓ getPartialFeatureVector function exists
```