TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

LUCA MOGLIA

CONTROL PLANE PERFORMANCE COMPARISON IN VIRTUALIZED BASE STATION CONTROLLER

Master of Science Thesis

Examiner: Professor Pekka Loula
Examiner and topic approved by the Council of The Faculty of Business and Built Environment 4.3.2015

ABSTRACT

MOGLIA, LUCA: Control plane performance comparison in virtualized Base Station Controller
Tampere University of Technology
Master of Science Thesis, 65 pages
March 2015
Master's Degree Programme in Information Technology
Major: Telecommunications Technology
Examiner: Professor Pekka Loula

Keywords: BSC, virtualization, telco cloud, SDN, NFV, SRAN

Mobile networks are in a middle of big changes. Requirements for scaling the networks while data and subscriber amounts are rapidly increasing make current network architectures too ossified. Current way of deploying mobile network with proprietary hardware and software of telecom equipment vendors is no longer sustainable. Telecom operators are having or will have profit problems with the current way. Fundamentals of building networks have to be changed.

New paradigms from IT and internet world are coming also to mobile networks. Virtualized IT server hardware and cloud based service model together with latest concepts like SDN (Software Defined Networking) and NFV (Network Function Virtualization) are eagerly integrated as a parts of operator's mobile network. Although most studies are concentrating on latest generations of mobile networks, mainly 4G, also older generations need to be part of the whole network portfolio. Future networks need to be transparent for users. In this scope, RAN (Radio Access Network) is in the biggest role being in the middle of the mobile user and the core network. This thesis studies the Single RAN concept from 2G BSC (Base Station Controller) point of view and what cloud, SDN and NFV would mean in that context.

There were two main objectives for this study. First, based on literature, it was studied how telecom networks are technologically emerging from the current situation towards cloud service era. That was reflected to how all virtualization and cloud related technologies will possibly influence to the mobile network evolution and what high level steps are expected in the journey. As the study was focused in BSC, it was examined how the current architecture could be evolved to meet telecom industry requirements of the upcoming changes. Single (or common) radio access network is one part of the evolution and from the BSC point of view the scope was also to evaluate, what possibilities there are to merge the radio network controller functionality in 2G and 3G on top of common virtualized hardware. Secondly two different initiatives of possibly needed steps in BSC part of the evolution were studied in practice. Many small steps will be needed and these experiments were just very small pieces in the entity. Those, however, indicated that a lot of studies, experiments and BSC architecture redesign will be needed.

TIIVISTELMÄ

MOGLIA, LUCA: Hallintakerroksen suorituskyvyn vertailu virtualisoidussa tukiasemaohjaimessa
Tampereen teknillinen yliopisto
Diplomityö, 65 sivua
Maaliskuu 2015
Pääaine: Tietoliikennetekniikka
Tarkastaja: professori Pekka Loula

Avainsanat: Tukiasemaohjain, virtualisointi, teleoperaattorin pilvi, SDN, NFV, SRAN

Mobiiliverkot ovat suurten muutosten keskellä. Nykyinen verkkoarkkitehtuuri on liian kankea vastatakseen verkon skaalautumisvaatimuksiin tiedon ja tilaajamäärien nopeasti kasvaessa. Nykyinen tapa kehittää mobiiliverkkoja verkkolaitetoimittajien suljetuilla laitteisto- ja ohjelmistoratkaisuilla ei enää ole kestävää. Nykyisellä tavalla mobiiliverkko-operaattoreilla on tai tulee olemaan kannattavuusongelmia. Mobiiliverkkojen rakentamisperiaatteiden täytyy muuttua.

IT- ja internetmaailman uudet ajatusmallit tekevät tuloaan myös mobiiliverkkoihin. Virtualisoitu IT palvelinlaitteisto, pilvipalvelumallit yhdessä viimeisimpien käsitteiden, kuten SDN (Software Defined Networking) ja NFV (Network Function Virtualization) integroidaan innolla mobiiliverkkoihin. Vaikka useimmat tutkimukset keskittyvätkin viimeisiin mobiiliverkkoteknologioihin, kuten 4G, myös vanhempien teknologioiden täytyy olla osana laajempaa mobiiliverkkoa. Tässä tarkastelussa radioverkko on suurimmassa roolissa mobiilikäyttäjän ja runkoverkon välissä. Tämä lopputyö tutki yhden radioverkon mallia 2G-tukiasemaohjaimen näkökulmasta, ja mitä nuo SDN ja NFV voisivat siinä yhteydessä tarkoittaa.

Tässä lopputyössä oli kaksi pääkohdetta. Ensiksi, tutkittiin kirjallisuudesta, miten mobiiliverkot ovat teknisesti kehittymässä nykyisestä tilanteesta kohti pilvipalvelumallia. Tämä heijastuu siihen, miten kaikki virtualisointi ja pilvipalveluun liittyvät teknologiat tulevat mahdollisesti vaikuttamaan mobiiliverkkojen kehitykseen, ja mitä vaiheita kehitykseen liittyy. Koska lopputyö kohdistui 2G-tukiasemaohjaimeen, sen nykyistä arkkitehtuuria arvioitiin, miten se voisi kehittyä vastatakseen mobiiliverkkojen tuleviin vaatimuksiin. Yhteinen radioverkko on myös osa tätä kehitystä, siksi tuliaseman näkökulmasta pohdittiin myös, mitä vaihtoehtoja olisi yhdistää 2G- ja 3G-tukiasemaohjaimet toimivaksi yhteisessä virtualisoidussa laitteistossa. Toiseksi, tutkittiin käytännössä kahta mahdollista vaihetta, mitä saatetaan tarvita, kun tukiasemaohjainta virtualisoidaan. Nuo olivat pieniä vaiheita kokoisuuden kannalta mutta osoittivat, että tarvitaan paljon tutkimuksia, käytännön kokeiluja sekä tukiasemaohjaimen arkkitehtuurin uudelleen suunnittelua lopullisten tavoitteiden saavuttamiseksi.

PREFACE

This Master of Science Thesis has been written as a part of Master of Science Degree in Information technology from the Tampere University of Technology, Tampere, Finland. The thesis work has been carried out in the Department of Telecommunications technology of Pori during the year 2014 and 2015.

I would thank my mentors Juha Hartikainen (technical), Jaakko Pohjonen and Petri Tiirikka for general support in the thesis work process. I would like also thank Kimmo Koskinen, Mika Karento and Professor Pekka Loula for their contribution.

Eura, 21.3.2015

Luca Moglia

TABLE OF CONTENTS

LIST OF FIGURES

TERMS AND DEFINITIONS

2G	2^{nd} Generation in mobile network technology. Refers also to GSM, GPRS and EDGE
3G	3^{rd} Generation in mobile network technology. Refers also to UMTS, WCDMA and HSPA
3GPP	3rd Generation Partnership Project.
4G	4th Generation in mobile network technology. Refers also to LTE.
API	Application Programming Interface
ATCA	Advanced Telecommunications Computing Architecture
BBU	Base Band Unit
BCN	Box Controller Node
BCXU	Multicontroller BSC Signalling Unit
BSC	Base Station Controller
BSS	Business Support System Base Station Subsystem (in GERAN)
BTS	Base Transceiver Station
CAPEX	Capital Expenditure
CBC	Cell Broadcast Centre
CMS	Conversational Monitor System
COTS	Commercial Of The Shelf
CSP	Communication Service Provider
CN	Core Network
CPU	Central Processing Unit
CS	Circuit Switched
EDGE	Enhanced Data Rates for Global (GSM) Evolution
EPC	Evolved Packet Core
ETM	Ethernet Transmission processing for Multicontroller
ETSI	European Telecommunications Standards Institute
E-UTRAN	Evolved UMTS Terrestrial Radio Access Network
FU	Functional Unit
GEMU	GSM Emulator
GERAN	GSM EDGE Radio Access Network
GPRS	General Packet Radio Service
GSM	Global System for Mobile Communication
HA	High Availability
HPC	High Performance Computing
HSPA	High Speed uplink/downlink Packet Access
IAAS	Infrastructure As A Service

IOT	Internet Of Things
ISG	Industry Specification Group
KPI	Key Performance Indicator
KVM	Kernel-based Virtual Machine monitor
LTE	Long Term Evolution
M2M	Machine to Machine
MCMU	Marker and Cellular Management Unit
MCN	Mobile Cloud Networking
MGW	Media Gateway
MVMO	Mobile Virtual Network Operator
NMC	Network Management Centre
MME	Mobility Management Entity
MMI	Man to Machine Interface
MSC	Mobile Switching Center
NGMN	Next Generation Mobile Networks
NE	Network Element
NFV	Network Function Virtualization
NIC	Network Interface Card
NIST	National Institute of Standards and Technology
OMU	Operation and Maintenance Unit
OPEX	Operational Expenditure
ONF	Open Network Foundation
OS	Operating System
OSI	Open Systems Interconnection
OSS	Operating Support System
OTT	Over The Top
PAAS	Platform As A Service
PCUM	Packet Control Unit for Multicontroller
PS	Packed Switched
QEMU	Quick Emulator
RAM	Random Access Memory
RAN	Radio Access Network
RAT	Radio Access Technology (or Type)
RNC	Radio Network Controller
RRM	Radio Resource Management
RRU	Remote Radio Unit
SAAS	Software As A Service
SDL	Specification and Description Language
SDN	Software Defined Networking
SGSN	Serving GPRS Support Node
S-GW	Serving Gateway

SLA	Service Level Agreement
SMB	Small and Medium Business
SMLC	Serving Mobile Location Centre
SRAN	Single Radio Access Network
SRC	Single Radio Controller
TCO	Total Cost of Ownership
TNSDL	Tele Nokia Specification and Description Language
TTM	Time To Market
UC	Use Case
UMTS	Universal Mobile Telecommunication System
UTRAN	UMTS Terrestrial Radio Access Network
VLAN	Virtual LAN
VM	Virtual Machine
VMM	Virtual Machine Monitor
Wimax	Worldwide Interoperability for Microwave Access
WLAN	Wireless Local Area Network
WWRF	Wireless World Research Forum

1. INTRODUCTION

1.1 Background and motivation

It can be said there is a technological revolution ongoing in telecommunication industry, especially driven by evolution in mobile data technology and related services. Already for a few years telecom operators have been in high profitability pressure where current way of making business is coming to an end. Many aspects are influencing to the profit but one of the key indicators they follow is ARPU (Average Revenue Per User) [1]. It expresses the income generated by a typical subscriber per unit time in a telecommunication network hence showing the direction operators network profitability is heading.

The current way of building mobile network and its services is ossified. Each network node is typically dedicated and fixed vendor specific equipment with proprietary hardware and software. Provisioning new network services is ineffective due to purpose-built platforms with tightly coupled software and hardware. It makes development and customization cycle long and high cost. Also different generations of mobile network technology (2G, 3G, 4G and others) are having their own equipment and software but they need to work seamlessly together. Estimating needed mobile network capacity is most difficult in radio access network side as geographically moving subscribers and time of day tend to produce uneven load to the network. As a result the nodes need to be scaled for covering nearly peak loads, practically the capacity is low and ineffectively used most of the time. [2], [3]

Covering rapidly increasing mobile network capacity requirements with the current way gets constantly more complex and costly. Amount of mobile data is increasing exponentially along expansion of 4G network. Different OTT (Over The Top) smartphone applications (like Facebook and Whatsapp) and video in general are generating high amount off mobile network traffic and signalling. In addition, the increasing amount of IoT (Internet of Things) appliances brings lots of new subscribers to the networks, 40-50 billion by year 2020 in some estimations [4], [II].

All together these bring operator’s TCO (Total Cost of Ownership) compared to the profit to an unbearable level. Some change is needed. There is a common consensus that the telecom industry can also leverage the technology evolution used in IT world for some time already.

Virtualization and cloudification are seen as main technologies addressing the issues that the operators are having. They are going to dramatically reorganize the whole technology and business of telecom industry. It will be a very interesting transformation and it is motivating to study the topic. The author of this thesis has a long experience in R&D of 2G radio network controller, BSC (Base Station Controller), hence it also made it more interesting to do a study specifically from a BSC perspective.

1.2 Objectives

The thesis can be divided in two perspectives. The first part is a literature study about how telecom networks are technologically transforming from the current situation towards cloud computing era. Which technologies will possibly have an influence to the evolution and which high level steps are expected during the journey? Available academic and industrial studies are evaluated to form an understanding where we currently are, what has been standardized and which will be the next practical offerings in the field. Especially studies from the BSC point of view are most interesting. As the study is focused in BSC, it is examined how the current architecture could evolve to meet telecom industry requirements of upcoming changes. Single (or common) radio access network is one part of the evolution and from the BSC point of view the scope was also to evaluate, what possibilities there are to merge radio network controller functionality in 2G and 3G on top of common virtualized hardware. That kind of combined controller functionality could be further evolved as a cloud based service.

In the second part of the study two different experiments of possibly necessary evolution steps in BSC part were examined in practice. Experiment 1 was an attempt to build a configuration where the selected BSC control plane application would be run on top of the same virtualized hardware and software platform that is being evaluated for the 3G radio network controller, RNC (Radio Network Controller). A further target might be a common software platform and to possibly merge BSC and RNC into a single cloud radio controller. Instead of using the current proprietary hardware, a virtualized IT server hardware would be used. After having set up a working environment, the purpose was to run basic module testing to evaluate some basic functionalities of the application. In Experiment 2 the CPU performance of the current proprietary BSC hardware was compared with virtualized IT server hardware. Quite a heavy traffic profile, from the signalling point of view, was used in this experiment.

1.3 Thesis structure

Chapter 2 introduces the current radio access network used in 2G, 3G and 4G. The main network elements and their interfaces are depicted. The chapter also gathers some aspects of the current solutions of providing single radio access network and presents how it could evolve from the radio network controller point of view. Chapter 3 describes a current BSC network element variant from one telecom equipment vendor. Hardware and software architecture and their relation are described. Later in the thesis different virtualization and cloud architectures are sketched based on the current architecture. This network element is also used in Experiment 2 to represent the current BSC hardware. Chapter 4 introduces some key technologies and concepts, which are commonly seen as enablers in a telecom cloud. Chapter 5 presents some possible alternatives how the current architecture (presented in Chapter 3) could further evolve to integrate the enablers introduced in Chapter 4. Main drivers, for the enablers mentioned in chapter 4, would be to provide better radio network capacity and dynamic computing resource scaling in a cloud environment. Chapter 6 describes the two experiments, which had been selected as good practical items to be evaluated in this study. Finally Chapter 7 presents conclusions and further discussion of the covered topic.

2. RADIO ACCESS NETWORK

Radio Access Network (RAN) is a general term for the entity in telecommunication network, which is located between the mobile device and the core network. It can support different Radio Access Technologies (RAT). GSM (2G, Global System for Mobile Communication) / GPRS (2.5G, General Packet Radio Service) / EDGE (2.75G, Enhanced Data Rates for Global (GSM) Evolution), UMTS (3G, Universal Mobile Telecommunication System) / HSPA (3.5G, High Speed uplink/downlink Packet Access) and/or LTE (4G, Long Term Evolution) are the most common ones. However, also Wimax (Worldwide Interoperability for Microwave Access) or WLAN (Wireless Local Area Network) can be supported but they are not in the scope in this thesis.

2.1 Traditional RAN architecture

In basic concept there are separate networks for each RAT. 2G and 3G share same RAN architectural principles having dedicated controllers for a certain amount of base stations. In 4G controller functionality is distributed into base stations and core network hence there is no separate controller [5]. Figure 1 illustrates the key network elements, their interfaces and how the Control and user plane flow go between them.

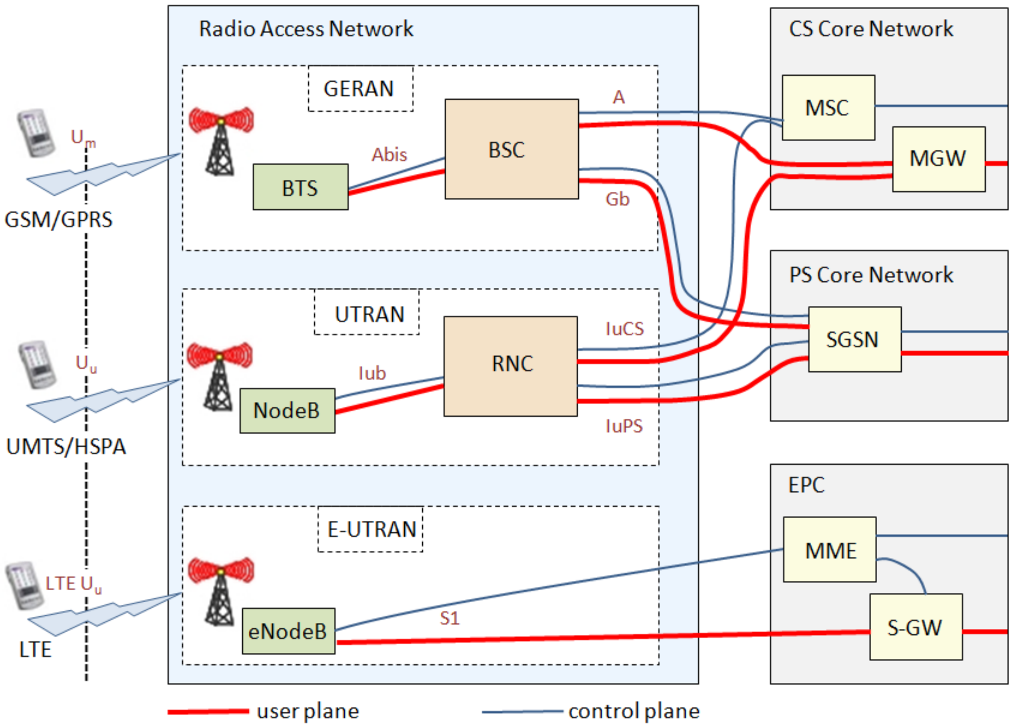


Figure 1. *Key RAN elements and their control plane and user plane.*

In 2G (GSM, GPRS and EDGE) the RAN is called GERAN (GSM EDGE Radio Access Network). It consists of BTSs (Base Transceiver Station), which are connected to BSCs with proprietary Abis interface. From BSC onwards the CS (Circuit Switched) service, voice calls or circuit switched data calls, go to CS core network via (open) A-interface. In this figure 3GPP Release-4 architecture is referred. That means MSC (Mobile Switching Center) is for signalling and MGW (Media Gateway) is for user data. Analogously, PS (Packed Switched) data go to PS core network via (open) Gb-interface. For both signalling and data there is common peer network element, SGSN (Serving GPRS Support Node). [6], [7]

In 3G (UMTS and HSPA) the RAN is called UTRAN (UMTS Terrestrial Radio Access Network). Its architecture is similar to GSM/GPRS/EDGE, only terms are different. Base station (called NodeB) is connected to RNC (Radio Network Controller) with Iub interface. Similarly, towards CS core network, signalling go to MSC and data to MGW. PS traffic go to PS core similarly as well, both signalling and data go to SGSN. Interfaces are called Iu CS/PS respectively.

In 4G (LTE) the RAN is called E-UTRAN (Evolved UMTS Terrestrial Radio Access Network) which consists only of base stations called eNodeB. The Base stations are connected directly to the core network, namely to EPC (Evolved Packet Core) with S1 interfaces. In EPC there are MME (Mobility Management Entity) for signalling and S-GW (Serving Gateway) for data. In LTE all user data go over same IP network including a normal voice call for instance. [8]

The rest of the core networks and interfaces are not depicted as they are not in the scope of this thesis. Those are for example towards the Internet or Location Service center.

Although not shown in the Figure 1, current base stations can be also multi RAT capable, which means they support several radio technologies. There can be a single base station which can serve a 2G/3G/4G capable mobile stations. This kind of base station is connected to BSC, RNC or EPC respectively. This Single RAN (SRAN) concept is described more in chapter 2.2.

Networking planes

Three high level entities (called planes) can be identified in network functions. Control plane controls resource allocations to users, controls service switching and takes care of signalling with other network elements. User (or Data) Plane is for the data generated by network user (payload). Management plane is for functions managing the two other planes.

The network follows OSI (Open Systems Interconnection) Reference Model with all the seven layers [9], excluding user plane in the upper OSI layers. In BSC (and the scope of this thesis) point of view three first levels (physical, data link and network) are the most essential. Also external BSC interfaces mentioned earlier (Abis, A and Gb) consist of dedicated protocols for these levels. More detailed discussion of the protocols is not in the scope of this study.

2.2 Single RAN

From the early days of mobile broadband evolution (mainly 4G but also 3G) there has been a lot of discussion of managing the evolving telecommunication networks. As GSM, HSPA and LTE are distinct technologies, developed independently and standardized separately, it is not easy to manage them all in the same network, which is deployed in a traditional way (as discussed in chapter 2.1). Figure 2 shows an example of radio access technologies used in an operator's network. In one physical site there can be three base stations, connected to dedicated controllers for 2G and 3G, and possibly separate transport network. Operator might also have other technologies deployed, such as Wimax or WLAN [10]. Smartphone penetration with their countless applications is also one of the lead drivers push mobile broadband evolution forward [11].

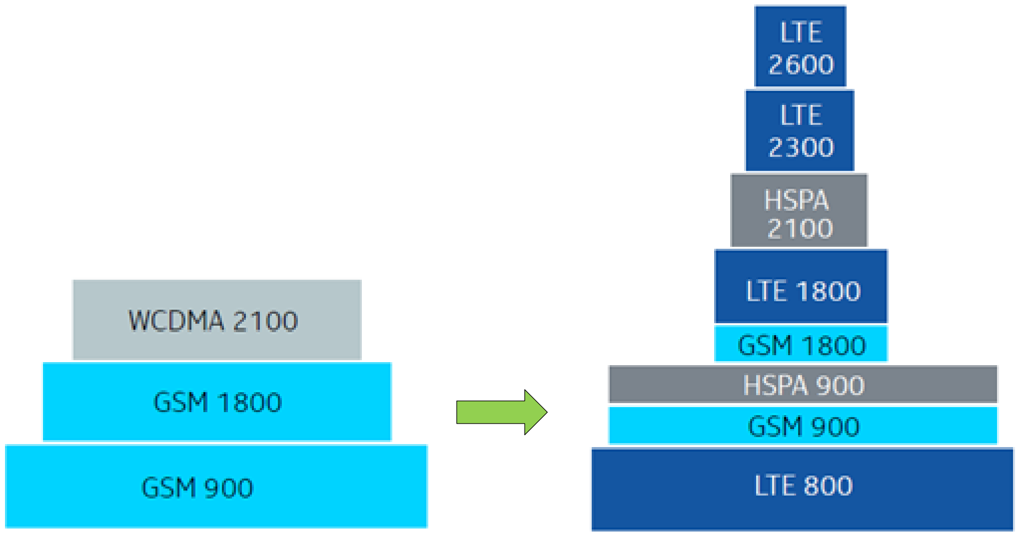


Figure 2. Evolution of radio access technologies from the RAN point of view *[12]*.

Single (or Common) RAN is not standardized technology and the telecom equipment vendors have different solutions to gather all their related solutions under an umbrella term to handle operator's expectations. The benefits that the operators desire includes [12], [11]:

- Efficient use of radio spectrum and site re-farming
- Efficient shared use of hardware
- Smooth evolution of GSM, HSPA and LTE
- Simplified network architecture
- Reduced energy consumption, green thinking

- Reduced operational expenses; one team can deploy changes to one network at the time (no need to be done separately for each RAT network element)
- Converged planning, operations and management
- Simplified, fully IP-based transport
- Automated 3GPP compliant security

It is seen that the operators have two network strategies to consider when deploying LTE:

1. **LTE overlay**, where a separate LTE radio access network (RAN) is deployed alongside any the legacy 2G and 3G RAN and core networks.
2. **Single RAN (SRAN)**, where a single base station unit provides the functionality of a new LTE base station and also replaces legacy 2G and 3G base stations at every radio site.

IT has been estimated that TCO is 21% lower with SRAN (option 2), which is a clear indication of the path that operators are choosing [13]. As mentioned earlier a RAN consists of base stations and controllers in case of 2G and 3G. To meet SRAN concept, the most of the changes are needed to base station side however. The key is to have the multi RAT base stations to effectively communicate with multi RAT mobile devices by sharing the operator's expensive radio spectrum assets in the most appropriate way depending on the end-user needs. Another aspect is to manage all kind of cell types in a single heterogeneous network, like Macro-, Micro-, Pico- and Femto cells and also WiFi cells [14]. This is the area where the most of discussion of different solutions are ongoing.

From RAN architectural point of view, it is then about transferring the user data towards core network with minimum amount of investments to existing telecom network topology. As LTE base stations are connected to the core network (EPC) via all-IP network, it is an advantage if also the existing 2G and 3G base stations can also utilize the same IP based transport network.

As also controllers are part of RAN, there are also solutions available from telecom equipment vendors where 2G and 3G radio networks are controlled with a single controller, a combined BSC and RNC. This as well belongs to single RAN evolution although it is not a key element in overall strategy. One solution is introduced in more detail in chapter 3. Basically virtualizing RAN controllers (BSC/RNC) with shared physical resources brings dynamic and scalable resource utilization as well.

Since SRAN, especially from the controller point of view, is not yet a mature technology, there are continuous discussions ongoing in different forums between interest groups (telecom operator and telecom vendors mainly). There are different proposals to reach synergies and benefits with different technologies. One example is SRC (Single Radio Controller) concept introduced by WWRF (Wireless World Research Forum)

[10]. Idea is to gather BSC and RNC functionalities as controllers and enhance the interworking with LTE as well. Also core network (CN) side would be common in the proposal. As an example of benefits, handover management between cells of different technologies would be much more efficient. This is illustrated in Figure 3, where much less inter network element signalling is needed and decisions can be made already in the single radio controller level.

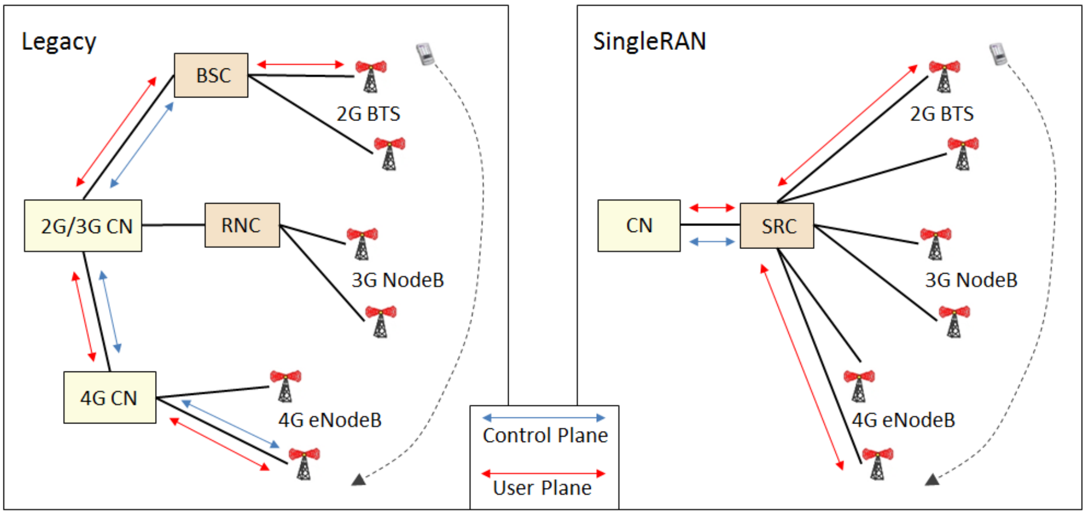


Figure 3. Example of benefit with SRC in multi RAT handover *[10]*.

On top of this proposal, there could be also multi RAT evolution in base station end, as discussed earlier with traditional architecture. In this thesis the scope is in base station controller end.

This would be then a good platform for future heterogeneous networks which most probably will be a basis for next generation (5G) technology. In RAT point of view it would mean adopting existing radio technologies to as common radio network. [15]

As an evolutional step of all-IP telecom network, also cloud is taking increasingly bigger role in this area. As well as in RAN architecture the idea would be to offer radio access network as a cloud service. This is discussed more in chapter 4. However, virtualization of RAN meets different challenges compared with virtualization of mobile CN. It is seen that same IT virtualization solutions of wired network and server domain are usable also in mobile CN. RAN characteristics related to wireless access links such as user mobility, dynamic amount of user in certain area or varying radio channel conditions makes network virtualization of the wireless resources across multiple entities more challenging. [16]

2.2.1 Common in vendor specific solutions

There have been SRAN solutions available for several years from telecom equipment vendors. The selection is increasing all the time while also telecom operators are more and more interested in solutions for managing the evolving multi RAT telecom network. Used terms vary between vendors but there are certain similarities as well: [17], [18], [19], [20], [12], [21], [22]

- In the base station end, practically all vendors have kind of modular elements for radio unit and baseband units with common hardware and which functionality is changeable with software update. They more and less clearly had visions to merge and share the functionality of these modules with different RATs.
- In the controller end, nearly all vendors had solution for common BSC and RNC functionality, which also is based on modular structure. Modules have common hardware as well and the functionality of the module is changeable with software upgrade as well (based on operator needs).
- In front/backhaul end, nearly all vendors had also vision (although not yet ready solutions) of shared transport network from radio interface to base station and from base station to controller end.

These are kind in line with visions of multi RAT base stations and controllers with other parties (than telecom equipment vendors) as well, as also described earlier. Also NGMN (Next Generation Mobile Networks) Alliance[1] has suggested an approach where centralized RAN can be formed with similar kind of modular components, which the vendors have already or are planning to be developed [23], [24].

[1] A mobile telecommunications association of mobile operators, vendors, manufacturers and research institutes evaluating candidate technologies for the next evolutions of wireless networks

3. MULTICONTROLLER BSC

In this chapter is described a Multicontroller platform based BSC solution from one of the leading telecom equipment vendor. Target is to provide SRAN technology with high capacity and future-proof controllers for IP based radio networks. The same controller would serve GERAN and UTRAN [25].

As BSC is in the scope of this thesis, only that side is described in more detail in following chapters. Same BSC is also referred later in Chapters 5 and 6 as legacy hardware.

3.1 General

BSC can be modularly scaled to meet any traffic mix and network topology. It is predicted that 2G technology need is constantly decreasing towards minimum level while 3G and further 4G are increasing [26], [12]. For this transition solutions like Multicontroller platform helps telecom operators to meet all the changing requirements for network capacity and deliver needed end-user services [25]. Multicontroller BSC supports only IP based connections on external interfaces to peer network elements.

Origin of BSC SW/HW architecture is based on HW architecture, resiliency and SW model supported by DX200 platform, and usage of only TDM interfaces. PS data support (GPRS, EGPRS) and IP interfaces are later add-ons to the old architecture.

3.2 HW architecture

Multicontroller platform architecture consists of modules which can be configured for BSC or RNC with software upgrade [12]. Minimum of two BCN (Box Controller Node) module configuration forms the basic setup [27]. Modules have own power supply and they are connected to each other with Ethernet. In Figure 4 there is a single module.

Figure 4. BCN module used to build Multicontroller BSC configurations *[28]*.

In the first deployment there can be all modules configured for GSM but as subscriber usage patterns change over time, the modules can be reconfigured from 2G to 3G. Module physical dimension is 4U (rack units) [28] in standard 19-inch server rack [29]. From practical part of the thesis point of view, the CPU details are the most essential. Those are described in more detail in chapter 6.2.1 where is also done comparison with the IT rack server CPU details.

In Figure 5 is presented the high level Module concept of Multicontroller BSC. Each Module can have control plane and user plane processing units as per need. Some Modules can have also management plane processing units. The units are briefly introduced in next subchapters.

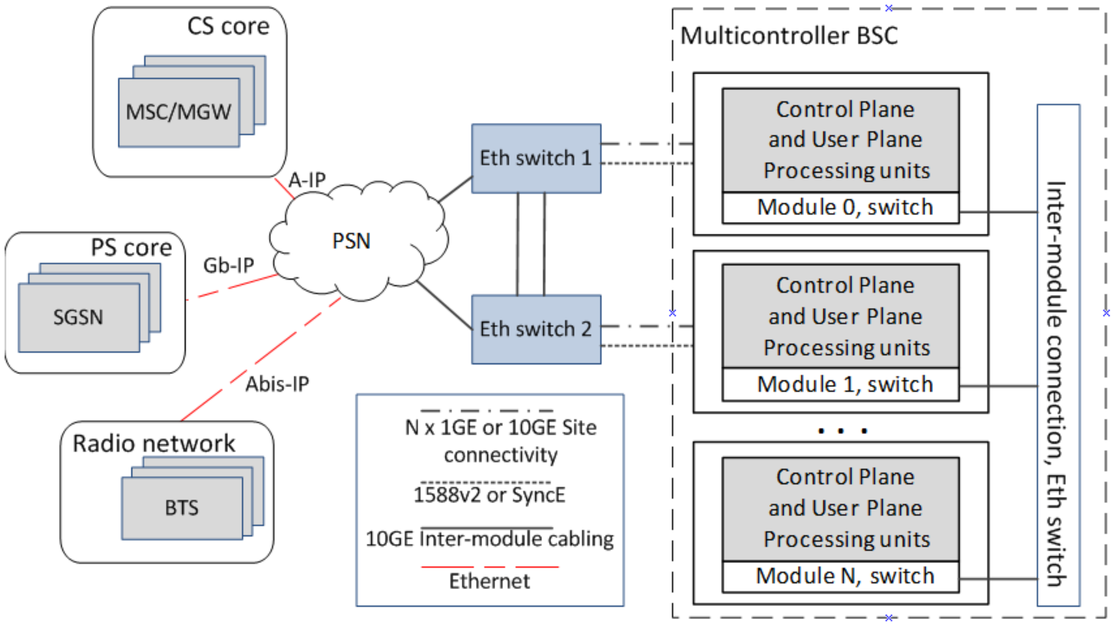


Figure 5. Multicontroller BSC high level HW architecture and main external interfaces *[28]*.

There are also depicted external interfaces to peer network elements to give better overall view how it places to the architecture. For simplicity, all the interfaces are not depicted; these are for example to NMC (Network Management Centre), SMLC (Serving Mobile Location Centre) and CBC (Cell Broadcast Centre).

3.2.1 Marker and cellular management unit (MCMU)

The MCMU has centralized management functions for controlling cells, radio channels and external interfaces for the BSC. The MCMU reserves and keeps track of the radio resources requested by the MSC and the handover procedures of the BSC. The MCMU also manages the configuration of the radio network. MCMU is 2N redundant, which means there is one unit in working at time and another is a spare unit.[28]

3.2.2 Multicontroller BSC signalling unit (BCXU)

The Multicontroller BSC signalling unit (BCXU) is a TRX-capacity unit of the BSC. It performs those BSC functions that are highly dependent on the volume of traffic. The BCXU consists of two parts, which correspond to the A-interface (towards MSC) and Packet Abis interface (towards radio network). The A-interface part of the BCXU is responsible for the following tasks:

- Performing the functions of SIGTRAN based SS7 signalling
- Performing all message handling and processing functions of the signalling channels connected to it

The Packet Abis interface part of the BCXU is responsible for the following tasks:

- Controlling the air interface channels associated with transceivers and Abis signalling channels.
- The handover and power control

BCXU is N+1 redundant, which means there is N amount of BCXU units in working state and one common spare unit to replace any of them if needed. [28]

3.2.3 Packet control unit for Multicontroller (PCUM)

The PCUM unit is an independent processing unit but logically connected to the BCXU. The PCUM unit performs all the data processing tasks related to GPRS/EDGE traffic. It implements packet switched traffic oriented Gb and Packet Abis interfaces in the BSC.

The PCUM controls GPRS/EDGE radio resources and acts as the key unit in the following procedures:

- GPRS/EDGE radio resource allocation and management
- GPRS/EDGE radio connection establishment and management
- Data transfer
- Coding scheme selection
- PCUM statistics

PCUM is N+M redundant, which means there is N amount of PCUM units in working state and M amount of spare units to replace any of them if needed. [28]

3.2.4 Ethernet transmission processing for Multicontroller (ETM)

In the Multicontroller BSC, A- and Abis- interfaces are connected to an IP network. The ETM functionality terminates user plane, which is related to A (over IP) and Packet Abis (over IP) -interfaces in the BSC. The general ETM functionality contains ETME and ETMA functionalities, where ETME handles Abis-interface and ETMA handles A-interface.

ETME is N+M redundant, which means there is N amount of ETME units in working state and M amount of spare units to replace any of them if needed. ETMA is SN+ redundant meaning group of units forms a working unit resource pool and traffic is redistributed between other units in case of failure (a pool have certain amount of extra capacity for that). [28]

3.2.5 Operation and maintenance unit (OMU)

The Operation and Maintenance Unit (OMU) is an interface between the BSC and a higher-level network management system and/or the user. The OMU can also be used for local operations and maintenance. The OMU receives fault indications from the BSC. It can produce local alarm display to the user or send the fault indications to Network Management Centre. In the event of a fault, the OMU, together with MCMU, automatically activates appropriate recovery and diagnostics procedures within the BSC. Recovery can also be activated by the MCMU if the OMU is lost. The tasks of the OMU can be divided into four groups:

- Traffic control functions
- Maintenance functions
- System configuration administration functions
- System management functions

OMU is not redundant, which means there is only one unit [28]. In unit fault situation on above functionalities are impacted, not control- or user plane functions.

3.3 SW architecture

BSC SW architecture is divided in several program blocks which takes part of different functionalities. In high level and telecom point of view they serve control-, user- and management plane. As mentioned earlier, different planes are mainly handled by dedicated units but program blocks in other units has also supporting role in them.

On the other hand, software is divided in platform and application parts, as generally done. In Figure 6 is depicted the high level software architecture in Multicontroller BSC.

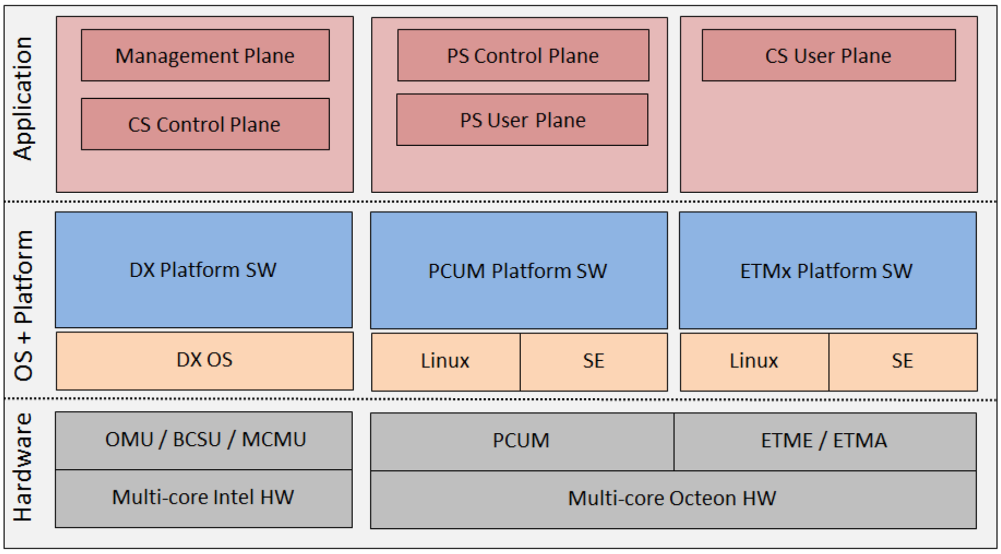


Figure 6. Multicontroller BSC SW architecture.

The traditional DX200 telecom functionality mainly for CS control plane and management plane runs on Intel hardware, dedicated DX OS and DX platform. On the other side PS user plane, PS control plane and CS user plane runs on Octeon hardware, Linux and Simple Executive OSs and PCUM/ETM platform software. Following subchapters describes entities in more detail.

3.3.1 Platform part

Basically platform offers different services for other platform functionalities but also overlaying application layer. Because of BSC evolution history, there are separate platforms used in OMU, MCMU and BCXU units than ones used by PCUM and ETM units.

OMU, MCMU and BCXU

Since early phase of GSM evolution these units were present in BSC. A dedicated proprietary platform software (DX200 [30]) has been in use at the beginning. In Table 1 are mentioned some of the main functionalities which are offered by platform software.

Table 1. *Main functionalities of platform software in OMU, MCMU and BCXU units [31].*

Main category	**Sub category**	**Example of functionalities**
Switching Platform	Basic switching services	Trunk network maintenance, Common statistics, Routing analysis
Computing Platform	System Maintenance	- HW testing services - System supervision - Alarm system - Diagnostic

	Basic computer services	- Recovery - SW and HW administration - TCP/IP and OSI management - LAN management - I/O system - File system - Licence management - User interface management (MMI) - SIGTRAN&SS7
Operating System		HW oriented functionalities such as messaging, program block management, warm up services and overload control

PCUM

Also due BSC evolution for GPRS and IP connectivity, PCUM and ETM functionality were adapted part of BSC later on. Because of this and also different nature of needed functionality a separate proprietary platform software is in use. As both units mainly serves User plane data, also platform focuses on services for processing uplink and downlink packets.

For PCUM there are separate operating systems running on different Octeon cores depending on the needed functionalities. Linux is serving functionalities for Gb-interface, Radio Resource management (packet switched traffic channels) and packet processing quality management. SE (Simple Executive) is serving such functionalities like for UL/DL Radio Link Layer, packet scheduler or packet framing/de-framing.

ETM

For ETM there is similarly dedicated platform layer running on Linux or SE. In User plane platform supports functionalities for processing Ethernet packet to/from radio network to ETME or to/from MGW (Media Gateway) to ETMA. There are also some supporting functionalities needed for Management and Control plane.

3.3.2 Application part

BSC application part handles the functionality specified for GSM/GPRS/EGPRS in 3GPP standards [32] with support by the platform part. Like in case of platform, also application part can divided for program blocks in OMU/MCMU/BCXU and ones in PCUM and ETM.

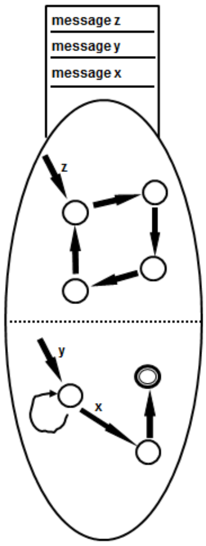
OMU, MCMU and BCXU

As described in Figure 6, mainly control plane (Telecom part) and management plane (O&M) functions are handled by applications running in these units. Functionalities of application part are roughly described in Table 2.

Table 2. *Main functionalities of application software in OMU, MCMU and BCXU units.*

Main category	Sub category	Example of functionalities
Telecom	L3 signalling	- Termination point of these external interfaces o Abis (also O&M) o A o Lb o BSC-BSC o BSC-CBC - System information sending
	Centralized radio resource management	- Radio channel (de)reservation for CS and PS traffic
	Handover and Power Control	- MS handover and power control management
	Overload protection	- Protects CPU loading of the units by buffering requests from external interfaces when needed. Also applications in PCUM and ETME participate on this.
O&M	RNW (Radio Network) Configuration Management	- User interface (MMI) for configuration handling - Database service for RNW configuration
	Radio Network Recovery	- Keeping RNW entities (BTS/TRX/TSL) operational
	BSS Alarm Handling	- RNW alarm/fault management
	Statistics	- Centralized management of measurements and counters (performance management)
	Gb-interface management	- Centralized management of the configuration in Database and PCUM
	BTS Testing	- Centralized testing functionalities
	BTS SW Management	- Centralized BTS SW management
	Network management interface	- Centralized functionality to support BSC and RNW configuration from network management subsystem. Includes also fault and performance/statistics indications.
	Location services	- Location based services for different purposes like emergency, commercial or operator own use

Application level architecture is based on basic, telecom and O&M services provided by the DX200 platform and dedicated DX OS. Scope of the thesis is in application on DX200 platform hence it is described a bit further. Software is mostly implemented using proprietary programming language TNSDL (Tele Nokia Specification and Description Language, derived from standardized SDL-88 language) [33], which is well supported by DX OS. Software modules form modular software architecture. Each module is a kind of process running in one or more state machine (also called sub automatons), all have their own message queue. Figure 7 illustrates message queue based triggering method.



Each small circle represents a state, meaning also a message waiting point. Messages (x, y and z) are handled one by one from the queue and dedicated actions are performed, such an operation is called transition. Each higher level scenario may consist of several messages, states and transitions between them. These state machine method suites well for telecommunication software with real-time requirements. [34]

Data is shared with messages between different processes and each process is using or proving services to each other. In message based communication in SDL (Specification and Description Language) sender of message not exactly knows when receiver gets the message and eventually responds to the sender hence service is called asynchronous. [33]

Figure 7. *Message based state transitions in TNSDL [34].*

PCUM

Application part, with assistance of the platform, controls PS resources of control plane and user plane towards proprietary Abis interface. On the other side, it communicates with SGSN via 3GPP standardised Gb-interface protocols [32]. Framing/de-framing of user plane and control plane packets for different protocol levels are the most essential part of PCUM functionality.

ETM

Application in ETME, as well with assistance of the platform, mainly handless user plane packets to/from BTS via Abis interface. For uplink direction CS data is forwarded to ETMA and PS data is forwarded to PCUM via internal IP network. For downlink direction it goes vice versa.

4. ENABLING TECHNOLOGIES FOR A CLOUD CONTROLLER

In the past years also telecommunication companies have been increasingly interested gaining benefits of virtualization and cloudification which have been seen in the IT world. The same principles apply also in Telco network, where costly network equipment is running inefficiently.

This chapter introduces virtualization concept. It also discusses which other aspects need to be considered before a RAN controller can provide the services in a cloud setting.

4.1 Virtualization in general

Modern virtualization technologies have been commonly used in IT world for a long time. During the past few years it has been rapidly expanding also to the telecommunication networks where similar benefits are expected.

Below subchapters briefly revise the background of the virtualization. After that different virtualization technologies and techniques are introduced. Some of those are also used in practical part of the thesis (Chapter 6.2).

4.1.1 Background and history

As a concept in the computing world, virtualization means hiding real hardware from the other parts of the system, including the operating system. It is a way to logically simulate or emulate physical computing resources. [35], [36], [37]

Already in the late 1950s the problem of time sharing in large foot print computing systems, called main frames, was discussed. At that time hardware was expensive and needed a lot of work to setup up. Also using the computers was not so effective, engineers had to use key punches and submit batch jobs one at the time, meaning there was no multitasking possibilities [38]. However, in 1960s there started to be real attempts by different companies, IBM as most memorable with their S/360 followed by CP-67 [39]. In the latter the hypervisor concept was introduced for the first time with the name CMS (Conversational Monitor System), later called also VMM (Virtual Machine Monitor) [38]. Virtualization evolved further as an important technology in main frame computing in late 1960s and 1970s.

During the 1980s and 1990s computing hardware costs dropped dramatically while also companies like Intel and Microsoft developed their x86 client-server architecture and multitasking operating system. It was a time of IT hardware expansion. It also meant the need for virtualization was not anymore dependent on the original time sharing problem.[40]

However, different motivations appeared to further develop the virtualization in x86-environment. The trend where x86 servers were not able to simultaneously run multiple operating systems together with their low price, drove enterprises to deploy dedicated hardware for each application [38]. That caused different problems in the field [35]:

- Low utilization of hardware. Each server was running inefficiently due the one application per server –logic.
- Increasing physical infrastructure costs. The ever increasing amount of the dedicated servers needed more and more space and then increased the facility costs. In 2000s, also energy consumption for the hardware but also cooling capacity and expenses have brought own influence here.
- Increasing IT management costs. The amount of cheap IT components needed a big amount of special talent work and management.
- High maintenance of end-user desktops. In distributed client-server computing, managing hardware, software and their different setup combinations of enterprises desktops became more and more complex and expensive.

Practically every component in a computer, basically in a server, can be virtualized. In addition many network devices and concepts can be also virtualized. Common with all of them is to make a single physical target appear as many, in a way multiplying it, and by that optimize the usage of physical resources [37]. Figure 8 lists some typical targets for virtualization [41], [39], [42].

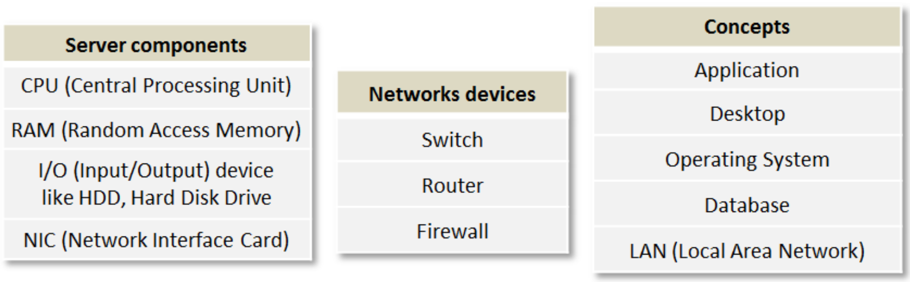


Figure 8. *Typical targets of virtualization in IT world.*

Because there are several technical alternatives, it is not easy to compare and evaluate the pros and cons for a desired solution for the target environment. Several vendors have proprietary solution but there are also open source solutions in the market. Available information is exhausting. In the next sub chapter are presented most common virtualization technologies and software. The scope is in virtualizing physical resources of a computer, also a BSC processing unit, which basically also is a computer operating in telecommunication network.

4.1.2 Technologies and techniques

As discussed, the key software layer in virtualization is hypervisor (or VMM). By emulating physical hardware of computer (CPU, RAM, I/O or NIC) it allows multiple OSs to run on one physical computer. In this context each OS is called guest OS and with hypervisor they all appear to have the physical resources of host computer. Together with applications and guest OS they comprise a Virtual Machine (VM), a kind of virtual computer. Benefit in that is all the VMs seem have all needed resources but they still cannot disrupt each other or the host computer itself. In virtualization hardware and software are decoupled which is essential in cloud architecture as well.[35], [40]

With live migration a VM can be moved from one physical host computer to another on run-time even without noticeable effect to the service provided by the applications running on VM [43]. This has an important meaning when it comes to redundancy and maintenance but also in load balancing of physical resources of virtualized hardware.

Allocating VMs to physical host computers, creating and deleting VMs can be done manually but desired way is to handle them automatically. A separate management entity (also a term orchestration is used) can create, move or destroy VMs based on the need of particular service. For the migration, VM is saved to file(s) in source host computer, transferred to new host computer and started in again there.

Furthermore there can be made division of two different x86 hypervisor types, namely bare metal (Type-1) and hosted (Type-2). In bare metal the virtualization layer is installed directly on to a server hardware, which means there is no OS between (like Windows or Linux). In hosted virtualization an OS is installed first on top of physical hardware, like done normally without virtualisation. On top of that is installed the virtualization layer. Figure 9 illustrates the architecture before virtualization and Figure 10 with mentioned hypervisor types.

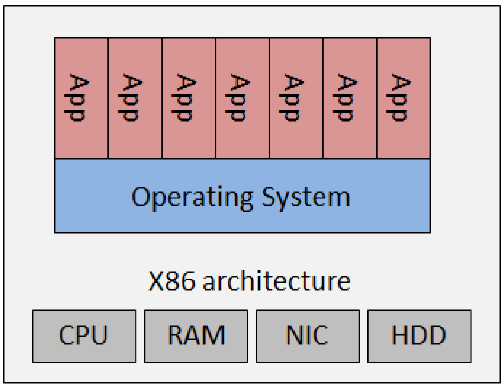


Figure 9. *HW and SW architecture without virtualization.*

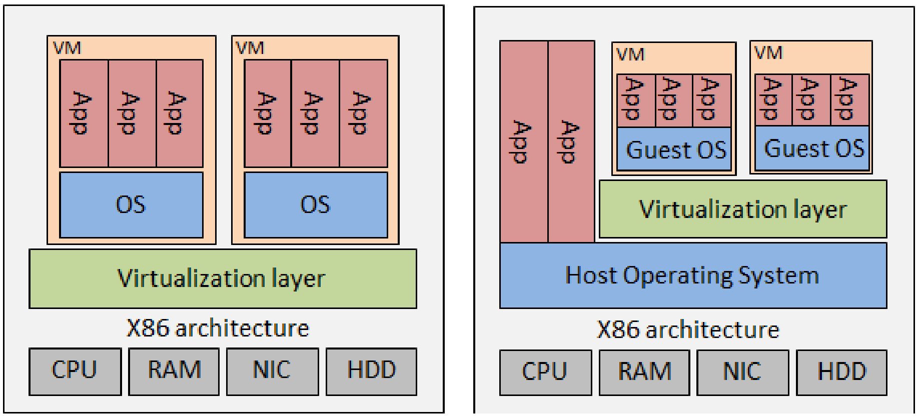


Figure 10. *HW and SW architecture with bare-metal and hosted virtualization*

As a further comparison there can be made some separation of these virtualization types: [36], [41], [40], [44], [45]

Some characteristics of **Bare-metal virtualization** (Type-1)

- Since it is installed right on the physical hardware, it offers better performance and scalability than hosted virtualization. It has been observed that physical resource utilization performance is 85% - 98% compared to native (without virtualization), which means there is not much overhead in most optimal cases. In ideal situation there is practically no overhead due virtualization layer, which is also one of further development targets for that. Also term native hypervisor is used for this.
- Smaller range of hardware is supported.
- There are more advanced features for physical resource management, high availability and security.
- It supports more VMs on physical CPU.
- Disadvantages are that most considerable solutions are not free. Also they are not usually running on user workstations.
- Example of hypervisor softwares: Citrix XenServer, Open Source KVM, VMWare ESX Server, VMWare vSphere, Microsoft Hyper-V Server.

Some characteristics of **Hosted virtualization** (Type-2)

- A host OS (like Windows/Linux/Mac) is required on top of the physical hardware.
- VMs can use all the physical resources that also host OS can use.
- Better hardware compatibility since the host OS offers all the hardware device drivers. This helps in deployment of systems where software is running on top of different hardware.
- Physical resource utilization performance is suffering more overhead due host operating system, it has been observed that it is 70% - 90% compared to native

- Example of softwares: VMWare GSX Server, Microsoft Virtual Server, Sun VirtualBox.

Mentioned hypervisor softwares do not always clearly belong into either category as they might have characteristics from both hypervisor types [46]. One of them is KVM (Kernel-based Virtual Machine monitor), which is very popular open source solution [47], also in telecom network virtualization deployments. Also virtualized server used in Experiment-2 (refer chapter 6.2) is configured with KVM. It is basically Type-2 hypervisor as it runs as a process on host (Linux) OS but it also converts the kernel into a bare-metal hypervisor and by that way have direct access to physical hardware resources (Type-1) [45]. It can also benefit the advantages of hardware assisted virtualization. Figure 11 represent main components of KVM hypervisor in Linux. Each VM is handled as a normal Linux process, which are emulated with a device emulator called QEMU (Quick Emulator). KVM itself consists of a loadable kernel module/object (*kvm.ko*) that provides the core virtualization infrastructure, and a processor specific module, *kvm-intel.ko* or *kvm-amd.ko* directly interacting with processor via virtualization extension. *Libvirt* process provides management functions of KVM components. [48]

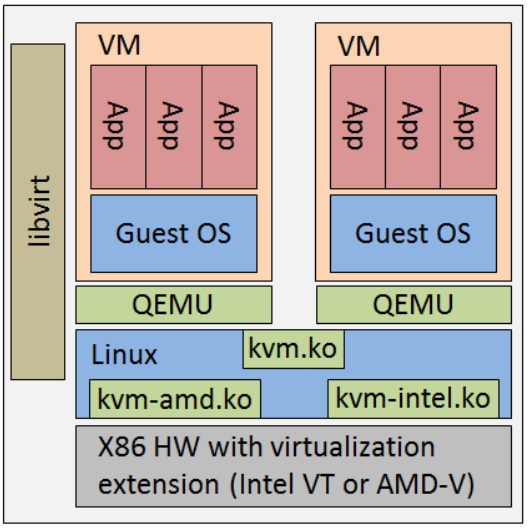


Figure 11. *Virtualized environment with KVM hypervisor and main components of KVM.*

CPU virtualization

CPU virtualization is probably the most common and concrete target for virtualization since the computing performance is the key metric when comparing virtualized environments with different solutions, and also to non-virtualized environment. There has been developed some techniques to further optimize virtualized x86 operating system. Main driver is in characteristics of x86-architecture, with four levels of privileges, known as Ring 0, 1, 2 and 3, which specify privileges for OS and applications to access computers hardware. The most trusted and privileged processes run in lowest protection ring. User applications typically run in Ring 3 and OS runs in Ring 0 [49]. CPU virtualization techniques depicted in Figure 12 can be separated briefly as follow [49], [35], [40]:

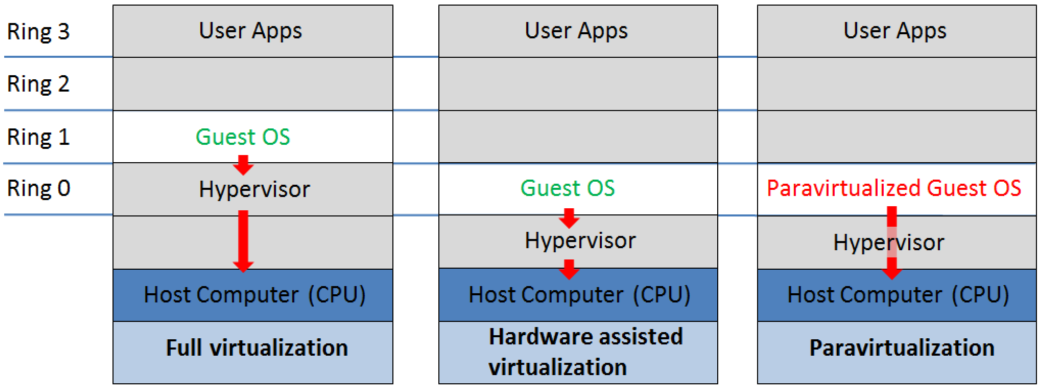


Figure 12. *CPU virtualization techniques; Full Virtualization, Hardware Assisted Virtualization and Paravirtualization [35].*

Following characteristics can be identified from the CPU virtualization techniques:

- **Full virtualization** is providing total abstraction of underlying physical hardware creates complete virtualized environment where the guest OS is running. Guest OS or application on top of that does not need any modification, meaning they are not aware of the below virtualized environment.
- **OS assisted virtualization** or **paravirtualization** presents each Virtual Machine with an abstraction of the hardware which is similar but not exactly identical with the underlying physical hardware. In other words, VM interface is defined so that existing non-virtualizable instructions set are replaced with easily virtualized and more efficient equivalents. Guest OS needs modification so they are aware that there is virtualized environment underlying. This is drawback since it needs porting to be compatible with different OSs.
- **Hardware assisted virtualization** is technique developed by hardware vendors (like Intel VT-x and AMD-V) where there are target privileged instructions with new CPU execution mode feature, which allows hypervisor to run in a new root mode in Ring-0. When selecting hardware for virtualization it is important that these kinds of techniques are supported as x86-architecture was not originally designed to handle virtualization.

Containers (LXC, Linux Containers)

Another technology compared with hypervisor is OS level, container based virtualization. In this method OS kernel is divided in multiple segments called containers or virtualization engines, each running a VM. In this way each VM not only shares a single type of OS but also a single instance of an OS (or kernel of it). Figure 13 depicts this concept further.

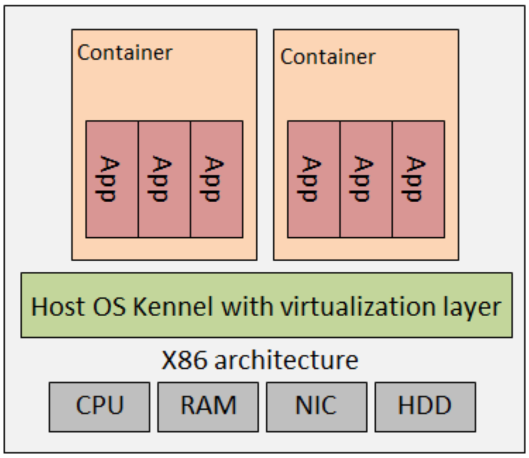


Figure 13. *HW and SW architecture with Container based virtualization.*

Containers have the advantage of providing lower overhead and are often more efficient particularly in high I/O environments. However, they restrict on-top system or process to run by the same host OS (it is not possible, for instance, to run Windows inside a container on a Linux OS), and the isolation between virtual machines is in general not as good as with hypervisor. Further, if a guest manages to crash its operating system (for instance due to a bug in the Linux kernel), this can affect the entire host, because the operating system is shared between all on-top systems or processes (inside containers). [46], [50]

4.2 Network virtualization

Network virtualization is seen as a key requirement for cloud networking [3]. Sharing network infrastructure with multiple tenants gains similar benefits as hardware virtualization in general. In a virtualization layer a virtual switch (vSwitch) can provide multiplayer networking services for VMs.

In wider perspective there can be identified two most important network virtualization concepts, which are introduced in following subchapters. Those are NFV (Network Function Virtualization) and SDN (Software Defined Networking).

4.2.1 Network Function Virtualization

It is stated that NFV (Network Function Virtualization) is starting new era in telecommunication networking. It provides radically new technology for deploying modern networks. [51]

Probably the first major initiative for NFV came from a group of telecom operators at *SDN and OpenFlow World Congress* on late 2012 [52], mostly due issues and concerns discussed earlier in the thesis. They believed that NFV could address those issues by leveraging the same IT virtualization technology to consolidate many current network elements onto industry standard high-volume servers, switches or storage, which could be located in data centres, network nodes and in the end user premises. At concept level every function taking place in their IT or telecommunication network could be virtualized. The concept is high level is illustrated in Figure 14. Conceptually every User plane and Control plane function in fixed and mobile network can be handled with NFV, which earlier were handled by dedicated network equipment.

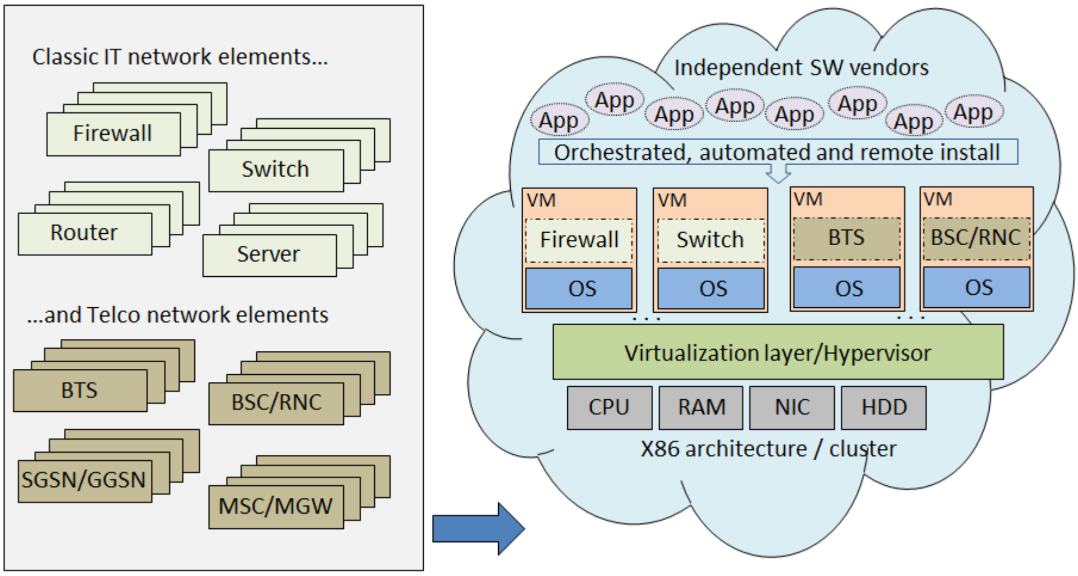


Figure 14. *Transformation from classical network appliance approach (of dedicated, standalone boxes) to virtualized network function approach.*

At practical level single network equipment consist of several sub functionalities covered by corresponding applications. Those virtualized applications are orchestrated in centralized manner, automated and remotely installed. Configurable amount of VMs are needed to cover all of the functionalities. It also further depends on an application size and complexity whether it needs one or many VMs. From BSC point of view virtualized network functionalities could be as discussed in chapter 3.3.2.

The telecom operators also foresee in the whitepaper [52] that NFV could bring many benefits such as:

- Reduced cost of equipment (CAPEX) but also reduced power consumption (OPEX - running, cooling and other functions) by consolidating the hardware thus exploiting the economies of scale of the IT industry.
- Increased speed of TTM (Time To Market) as also deployment (no need to install and configure dedicated hardware) cycle gets simplified. Due virtualization, proprietary hardware related features are no longer needed, so only software is developed.

- Common platform allows use of multi-version and multi-tenancy of network application. This allows telecom operators to share their physical resources over services and over different customer base.
- Possible to introduce services by geography or customer’s general needs, service can be rapidly scaled up or down when required.
- Makes evolving different eco-systems possible and encourages openness. It opens software market to pure software entrants (like small enterprises and academia). By that way it encourages for more innovation to bring new services and revenue possibilities, quickly and at lower risk.

At high level these benefits seem reasonable. However, the following challenges still needed to be addressed:

- Having high performance virtualized network applications, which are portable between different virtualization environments running on different hardware. Moreover, mixing hardware, hypervisors, orchestration softwares and network applications from different vendors, without having significant integration costs and by avoiding vendor lock-in.
- Having parallel use of existing dedicated hardware based network platforms but still have efficient migration path to fully virtualized network platforms.
- Managing and orchestrating numerous amounts of virtualized network applications (especially alongside legacy system) while handling security aspects and misconfiguration.
- Scaling of services is possible only when all the virtualized functions can be automated.
- Resilience to hardware and software faults needs to be handled appropriately. Basically from service point of view in similar manner as with legacy system.

Shortly later a group of world’s leading telecom operators selected ETSI (European Telecommunications Standards Institute) to be home of ISG (Industry Specification Group) for NFV [53]. Since that the group has extended with many other parties, including telecommunication equipment vendors and telecom operators. Standardisation work has been going on in several smaller specification groups, and the first high level Use Cases (UC) have been identified [54], but the work still continues. Experts from the ISGs are developing the required standards for NFV as well as sharing their experiences of NFV development and early implementation.

There is also an open source project called Open Platform for NFV, which is focusing on accelerating the evolution of NFV. OPNFV project works closely with ETSI’s NFV ISG for having standards for an open NFV reference platform. [55]

4.2.2 Software Defined Networking

Computer networks have evolved in many directions during their existence of several decades. Purposes and requirements have changed a lot since early days being increasingly very complex and difficult to manage, which causes more risks and expenses. Network consist of many kinds equipment such routers, switches and various middle boxes like network address translators, firewalls, traffic shaping devices, server load balancers and so on. They all tend to have their own control software, especially routers and switches typically have increasing amount of complex proprietary software. Vast variety of devices, vendors, network protocols and related aspects have already a while being speeding the new evolution of the networks. [56]

Already since mid-1990 there have been various approaches to bring more intelligent to networks. Active and programmable networks and solutions such as of decoupling User plane and Control plane have been existed in different incompatible solutions many years already. Still, any revolutionary breakthrough has not occurred, until elaborated concept of SDN (Software Defined Networking) [57]. SDN is considered the first commonly agreed paradigm for coupling many challenges in the present ecosystem. It is said that SDN is transforming the whole current network architecture [58]. Also vast common interest towards SDN by academia, carriers, equipment vendors and different standardization groups [56] is telling that something revolutionary is happening.

In Figure 15 is depicted the high level difference between current and SDN based network architectures.

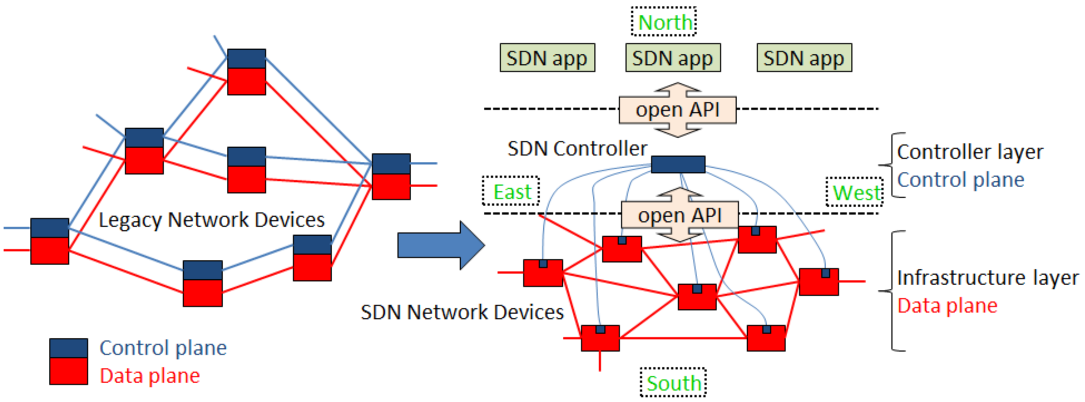


Figure 15. *High level difference between current and SDN based network architectures.*

In left side each box presents a network device such as switch or router, all having their own distributed control logic. In the right side the control logic is separated and centralized to a specific SDN controller. The controller can be also physically separated and virtualized (scalability, resiliency and performance aspects). [59]

In wireless network, like from GERAN point of view, SDN controller may perform control plane tasks belonging to BTS or BSC. Those functionalities can be also virtualized with [60] NFV, as discussed in earlier chapter. [60]

SDN has many characteristics but the following are usually mentioned [61], [58]:

Separation of control plane and data plane. In SDN data plane consist of only data forwarding devices, which only have forwarding tables (for the packets) prepared by control plane. Not much other intelligence, so complexity of different network devices has clearly simplified, reflecting also to expenses. Data plane component can be a physical device but also a virtualized one (like vSwitch or vFirewall).

Centralizing the control plane. Control plane has been decoupled but more importantly, centralized to single entity, SDN controller. One clear benefit is that needed changes to network architecture or different policies can be made much faster compared to the current distributed architecture [59]. Network can be further divided into slices which all are managed by own SDN controller but still seamlessly working together [56].

Programmability of the control plane. As the control plane is centralized, programming policies, rules and other control plane functionalities is much easier and flexible. It allows a variety of controller software to dynamically change data plane behaviour according to different needs.

Standardized APIs (Application Programming Interface). In SDN terminology interfaces are named based on point of direction. All APIs are open and under continuous development and standardization efforts.

- Southbound API defines interaction rules between data plane (forwarding) devices and controller. As interface is open there are many solutions available but clearly most popular and interesting is OpenFlow[2] . There is a separate open source driven standardization forum, ONF (Open Network Foundation) [62] consisting majority of networking related enterprises.
- Northbound API present interfaces for all possible applications developed on top SDN based network. At the moment there are no specific standard initiatives as the field is still evolving and many instances have their own interests and needs. Maybe the most discussed solutions at the moment are open source driven Open Daylight [63] and Floodlight [64]. Also cloud functionalities can be integrated via Northbound API.

[2] Basic principle in OpenFlow is to tag packets belonging to same contexts (for example a video stream) thus avoid routing of each related packets separately. Flow information of each packet is searched from internal flow table, if not present then checked from the SDN controller.

- Eastbound and Westbound APIs have not either any standards but in conceptual level they can provide interface between SDN controllers to share information of controller state if needed in some applications.

SDN can be a building block for wireless mobile networks as well, although most of the interest in wired domains currently. There are some high level initiatives by academia for example [59] but also a separate Wireless & Mobile Working Group under ONF has been established [65]. Purpose of the WG is to define use cases but also architectural and protocol requirements to extend ONF related technologies to wireless and mobile domain. However, ONF highlights mostly different scenarios in LTE architecture and also some WiFi scenario. 2G/3G concept in SDN is not covered, yet.

SDN can be adapted with the virtualization approach discussed earlier, NFV. They have similarities (benefit from automation and virtualization and that way reducing OPEX/CAPEX) but they still are independent and complementary technologies. In SDN the underlying transport network is optimized, whereas in NFV the higher level network functions are virtualized.[66]

As a subset of SDN, term SDWN (Software Defined Wireless Networking) is used. It is not a standardised technology but there is a set of different approaches (such as OpenRadio, SoftRAN or MobileFlow) how SDN can be applied specifically in wireless network [60]. It is probable that these are adopted in a standardization effort, perhaps under ONF.

Another high level proposal is to leverage SDN in software-defined RAN architecture in OpenRAN concept [67]. In their proposal the current physical RRU (Remote Radio Unit) and BBU (Base Band Unit) in BTS end, as well as BSC and RNC in controller end, are replaced as virtualized units. All the units are controlled by SDN control plane, which is said to make RAN more controllable and evolvable. However, also this concept leaves much study and research efforts into practical architectural level.

4.3 Cloud computing

In certain way cloud computing can be compared to electricity grid. Both can be called as services, which are used without awareness how they are produced. Term cloud refers to an abstract way where service user does not know and/or care where the information comes from, like it has been with Internet over times.

Although cloud computing and services have existed already few decades, until recent years there has not been clear agreed definition for it. Either there has not been any standard, which technically clearly defines all the needed aspects for evolving business between cloud related service providers and users. The cloud is introduced in more detail in these chapters, especially from telecom networks point of view.

4.3.1 Definition

One of most cited definition for cloud computing is NIST (National Institute of Standards and Technology) view of it [68]. The same cloud model is also cited in this context. In Figure 16 there is a common view of these aspects.

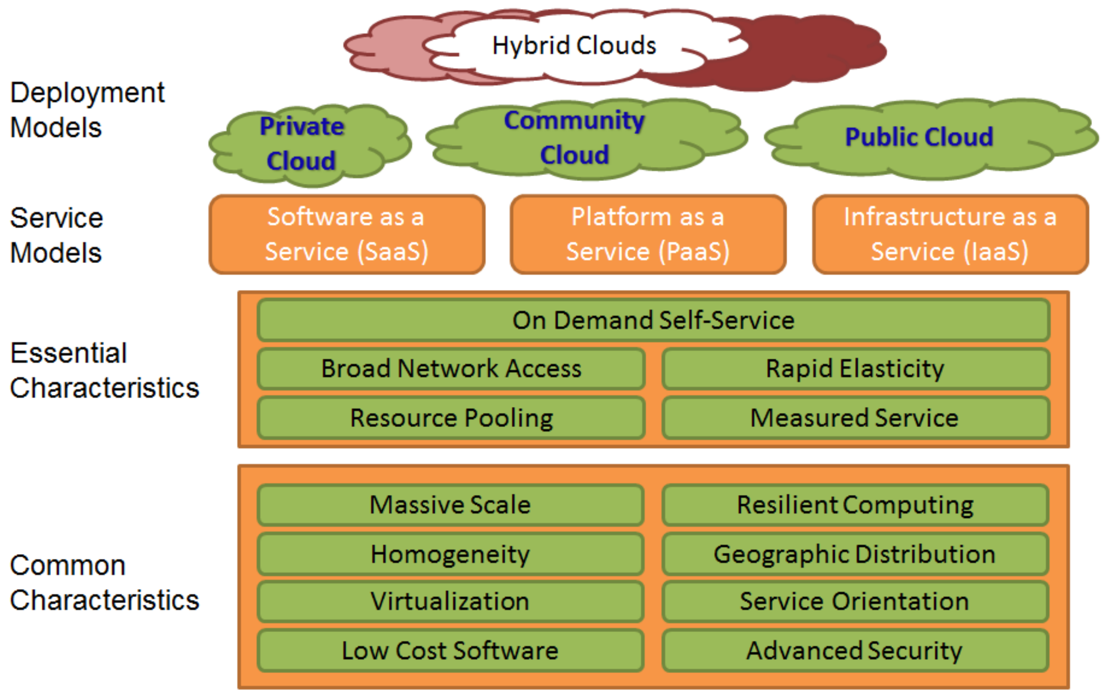


Figure 16. *NIST definition for cloud computing diagram [69].*

Related to the above figure, some characteristic of the cloud computing services are defined:

<u>Common Characteristics</u>

NIST has defined general characteristics of cloud depicted in the above figure. Weighting those or adding/removing some of those, NIST has left to reader's consideration.

<u>Essential Characteristics</u>

On-demand self-service. Cloud service customer can self-consider how to utilize computing capabilities like server time or network storage. These fore-agreed services are automatically available without manual interaction by service provider.

Broad network access. Services can be used over standard Internet access mechanisms and by that way promote location-independent use by diverse platforms such as smartphones.

Resource pooling. The service provider hosts physical resources (computing, network and storage) as virtualized concept. Those resources are dynamically (re)assigned for service users (might by many in multi-tenant environment).

Rapid elasticity. Resources are rapidly (practically imperceptibly) scaled for customers need, moreover based on the SLA (Service Level Agreement). For customer point of view there are unlimited resources that are paid for based on the usage.

Measured service. For charging of the usage, there is precise measurement capability for sharing transparent reports with the customer. Paying for the service is the most essential reason for it.

<u>Service models:</u>

***Software** as a Service (SaaS)*. Customers are using software of cloud providers over the Internet with various client devices and lightweight interface (web browser or separate application). The entire needed infrastructure, including network, servers, operating systems, storage, or even the individual applications are managed by service provider.

***Platform** as a Service (PaaS)*. Customers are using software development platforms and related tools over the Internet. They need only to develop, upload and install the software to cloud environment. Service provider handles the rest of needed infrastructure.

***Infrastructure** as a Service (IaaS)*. Customers are using only virtualized hardware, including computing, network and storage. They can install and control desired OSs and applications on top of that but the rest of the cloud infrastructure is managed by service provider.

<u>Deployment models:</u>

Public cloud. The cloud infrastructure is targeted for open use by general public, like everyone. It could be owned, managed and operated by enterprise or academic and infrastructure can be physically located in cloud provider's premises.

Private cloud. The cloud infrastructure is targeted for exclusive use, such as enterprise or government. Same instance (or a third party) owns, manages and operates it and infrastructure could be physically located in owner's premises.

Community cloud. The cloud infrastructure is targeted for exclusive use by a specific community of users from organization sharing same interest. It could be owned, managed and operated by the same users or a third party and infrastructure can be physically located in owner's premises.

Hybrid cloud. The cloud infrastructure is composition of unique entities of the three first models mentioned. Clouds are bound together by standardized or proprietary technology, which enables data and application portability between them. In NIST definition hybrid cloud appears to some new combination of cloud infrastructure, although some other might think there is just two separate could models parallel [70].

4.3.2 Cloud computing platform

Computing cloud needs a variety of tools and methods for orchestrating the virtualized resources. This software stack has several terms, depending on which functionality is emphasized. It can be called cloud computing platform but also cloud operating system or cloud middleware. Generally they all provide cloud infrastructure as service (IaaS). As there are no specific standards, different approaches have evolved during recent years. There are several commercial products, like Microsoft Azure [71] but also several open source projects like CloudStack [72] or OpenStack [73]. Amongst all, it seems that OpenStack has become the de-facto standard due its high popularity based on the extensive modularity and high community support. Generally, open source might be important also in this area for enterprises which need to avoid vendor lock-in[3] [74].

OpenStack consist of several compatible components, which together form a cloud operating system. All the components are independently developed in dedicated projects. Table 3 gathers the OpenStack components, used code names and brief description of each. Basically they can be divided in three main elements – Compute, Networking and Storage. Around those there are several other supporting components, either defined by OpenStack or 3rd party players. Benefit is all modules and their APIs are open and well documented and widely supported.

Table 3. *Modular components of OpenStack cloud platform concept [73], [75].*

Service	Code name	Description
Compute	Nova	Main computing engine controlling cloud computing platform as IaaS. It manages virtualization layer (VMs) by taking requests trough dashboard or API. It has wide support of available virtualization technologies.
Networking	Neutron (ex. Quantum)	Controls virtualized network connectivity between different OpenStack components or VMs.
Object Storage	Swift	Provides scalable and redundant storage system by spreading objects and files throughout multiple disks

[3] A situation where a telecom operator is forced to use products by a specific vendor due hardware and/or software restrictions.

		and servers in the data centre.
Block Storage	Cinder	Provides persistent block-level storage devices for Nova component, moreover VMs.
Identity Service	Keystone	Provides authentication service by centralized controlling which services users can have access.
Image Service	Glance	Provides services for storing and restoring disk images when needed by VMs.
Telemetry	Ceilometer	Monitors and measures performance of different cloud resources like use of VMs, CPU or memory.
Orchestration	Heat	Provides cloud orchestration service by enabling launching of cloud composite applications based in templates in the form of text files.
Database	Trove	Offers Database-As-A-Service for OpenStack components by providing scalable and reliable functionality for both relational and non-relational database engines.
Dashboard	Horizon	Provides a user interface for configuring and deploying different services (included above mentioned).

OpenStack is interesting also from Telco cloud point of view. Many telecom equipment vendors have decided to adapt at least some components to their products [76], [III], [IV], [77], [42]. This further strengthens OpenStack support as a general cloud OS. Figure 17 illustrates how basic components could be used in Telco cloud environment.

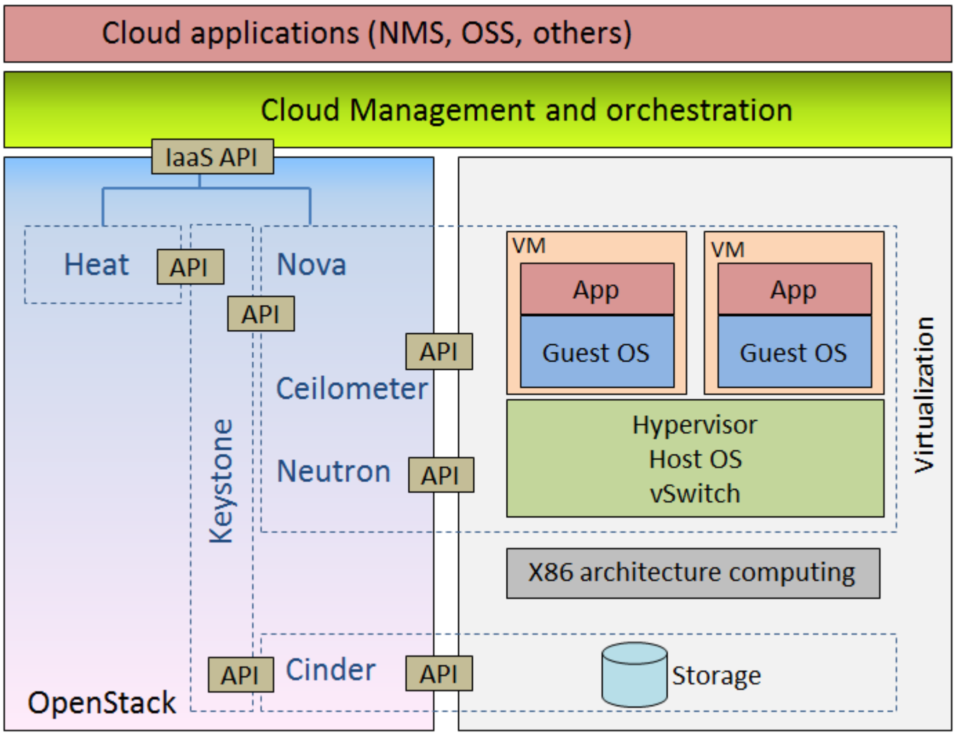


Figure 17. *OpenStack components in Telco cloud environment.*

From IaaS point of view the main components are virtualized functions related to computing (Nova), networking (Neutron) and storing (Cinder). They can be managed by separate applications via separate cloud stack but OpenStack Heat can be used in cloud orchestration. Keystone can be used controlling access to different OpenStack components from any cloud application. Ceilometer can be used to gather different metrics from virtualization layer for any cloud application needing it. On the highest level in Telco cloud are different application related support functions such as Network Management System (NMS) or Operation Support System (OSS).

4.3.3 Telco cloud

Telecom operators, which in many cases are also Internet service providers, have become also interested in benefits of cloud computing. For some years they have been struggling with unfavourable trend where network usage has been and will be strongly increasing. Data transfer per device is increasing due to evolving application needs, in an estimation to 1GB/day by year 2020 [22]. Also amount of wireless connected devices is increasing, in an estimation more than doubled compared to the current level, meaning over 40 billion devices [4], even 50 billion devices has been estimated by the same time [II]. Such amount of devices (subscribers from network point of view) is very challenging network resource management point of view for instance [14]. Majority of this amount of devices will be different smart appliances like different sensor nodes. This concept is also called M2M (machine-to-machine) communication or IoT (Internet-of-Things). Building mobile broadband networks with proprietary, dedicated and stand-alone network elements in the current ossified architecture is not economically sustainable after some time [1]. For example load balancing is difficult especially in peak times while most of the time there is a lot of free capacity [78]. Another example could mass events like Super Bowl [79] where (tens of) thousands attendees use network services temporarily. With current networks operators needs to temporarily increase local radio network capacity for that period of time. That might mean more Radio Units and Baseband units in BTS end but even additional radio network controller (BSC/RNC).

By using cloud computing technology, telecom operators are able to virtualize the telecom network functions with inexpensive COTS (Commercial of the Shelf) IT hardware or ATCA (Advanced Telecommunications Computing Architecture) blades. In above referred events the computing resources can be much faster increased in virtualized data centres. This includes Baseband Units in BTS end but also radio network controlling capacity (BSC/RNC). In operator's expectations, all the current functions can be virtualized and cloudified. In addition to gain advantage of virtualization as discussed earlier, such a Telco cloud could share the same IT infrastructure and management with operator's other IT systems [80]. Cloud services are seen also as an opportunity get new business and revenues for the operators, but for increasing internal efficiency as well [81]. Operators do not necessarily need to provide all the cloud based functions and services – instead, due to cloud models (like IaaS discussed earlier), a service provider can provide the functions outside operator's core competence [77].

From cloud definition point of view a private cloud (or some hybrid) is used. Depending on the operators business, cloud service model is selected. In case of hybrid cloud, different enterprise's functions are deployed with separate clouds but they all use more or less the shared virtualized hardware. Figure 18 illustrates an option where three cloud entities are used in as hybrid cloud:

- Left: Actual telecom network is virtualized with NFV and are forming a separate private cloud. These functions consists of RAN (BTS and BSC) and core network functionalities. These can have high security and real time requirements and hence are more demanding for the IaaS.
- Centre: Operator IT functions and part of telecom network functions like OSS (Operating Support System) or BSS (Business Support System) are forming a separate private cloud.
- Right: SMB (Small and Medium Business), different enterprises, industries and other 3rd party instances are forming a separate public cloud. This can be an own value adding ecosystem bringing also new revenues to the operator.

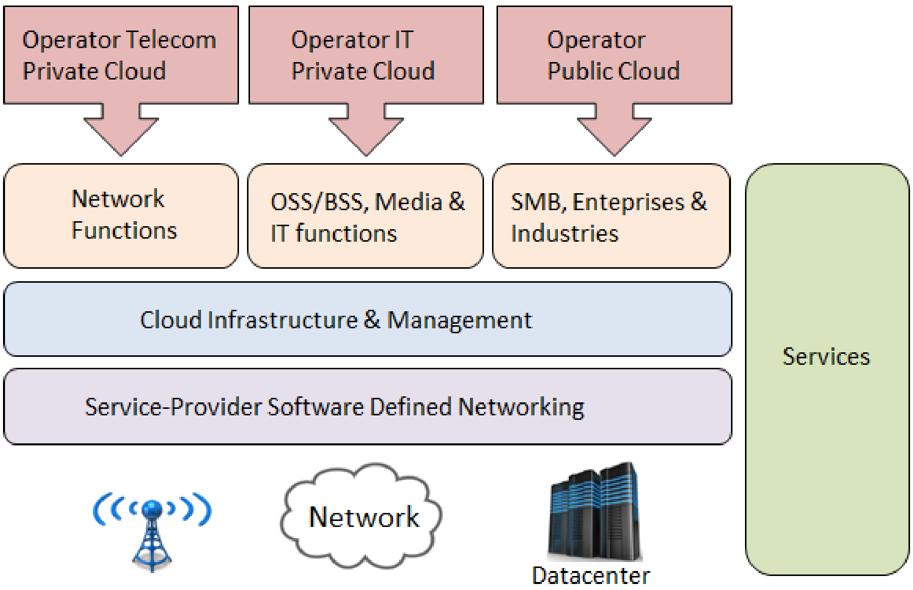


Figure 18. *Example of different cloud deployment models for Telco environment [II].*

All the clouds depicted in the above figure are running on common cloud infrastructure. Real time functions might still require additional performance. Depending on the service requirements they can physically locate in big centralized data centres or distributed over several data centres spread over geographical manner. Mobile network functionalities (like RAN internal signalling or CS/PS user plane) requires low latency, therefore there needs to be several small scale data centres close enough to mobile users [82], [77]. This is a difference compared to the traditional IT cloud architecture where services can be globally provided on handful of large data centres [51].

Right side of the above figure presents services for handling different cloud entities. Services for managing all cloud entities can also be provided by 3rd party enterprises due to open APIs.

Some fundamental challenges arise when transforming of classical telecommunications networks to cloud based services. These basically originate from differences in architectural characteristics of current and cloud based networks. In general, porting of current network elements to cloud without SW architecture changes, will not give all benefits of cloud computing, but merely just virtualization. Following main characteristics could be still identified:

- **Resiliency**. In current network nodes (like BSC) the service availability over failures in the dedicated hardware units is covered by redundant units, as discussed in chapter 3.2. In virtualized computing of Telco cloud redundancy has to be handled in virtualization infrastructure. This is more discussed in chapter 5.1. [83]
- **End-To-End security**. Mobile networks in general but especially RAN transport network is transitioning from current isolated connections to open IP network. This brings basically similar security threads as in current Internet world but also some mobile network specific issues. For example, heterogeneous security solutions during network evolution over 2G, 3G and 4G forms complex and incompatible security portion for mobile user. On the other hand, new security thinking is needed also between end-user and cloud service, which provides universal security and authentication transparently on the radio access of different technologies. [84], [85] [86], [87]
- **Performance**. Typically similar carrier-grade performance is expected also by Telco cloud as by legacy network [II]. Compared to IT cloud, Telco cloud is much more performance demanding hence a special technology is needed. SLA parameters, such as throughput and latency shall not deteriorate when virtualization and cloud layers are integrated. However, performance of virtualized hardware should be adequate compared to legacy proprietary hardware with the same radio network capacity. Leverage of hardware accelerators are limited in virtualized environment, which causes challenges with used general purpose hardware. [88], [89], [90].

Telco cloud can bring also totally new ways to operate the virtualized and cloudified telecommunication network with all the related services. There can be also virtual operators, such as MVNO (Mobile Virtual Network Operator) [91]. An operator does not need to own the network infrastructure but it can buy it as a service from other party. The service then covers all the SLAs for performance and security for example.

There are no specific standards in Telco cloud area although different activities are ongoing in several interest groups. One example is EU-funded MCN (Mobile Cloud Networking) project focusing on virtualization of cellular networks [92]. Their aim is to develop fully cloud-based platform for mobile communications and applications. New cloud service model RANaaS (Radio Access Network-as-a-Service) is introduced but it mainly concentrates to LTE as mobile technology, meaning it does not depict the scenarios from BSC (or RNC) point of view, which is in the scope of this thesis.

Another example of Telco cloud related consortium is UNIFY, co-funded by European Commission FB7 Integrated project. Their aim is to provide different ways for flexible service creation within the context of unified cloud and telecom operator networks, especially focusing on Telco functions. [2]

5. TARGETS FOR BASE STATION CONTROLLER EVOLUTION

Previous chapters discussed different perspectives and building blocks for a base station controller working on a cloud in several small steps. There are various technical challenges to be solved in the way. Real time requirements related to telecom network brings own challenges. At high level, steps are similar to in IT world. Figure 19 depicts in high level characteristics of the main steps. Steps are described in more detail in the following sub chapters.

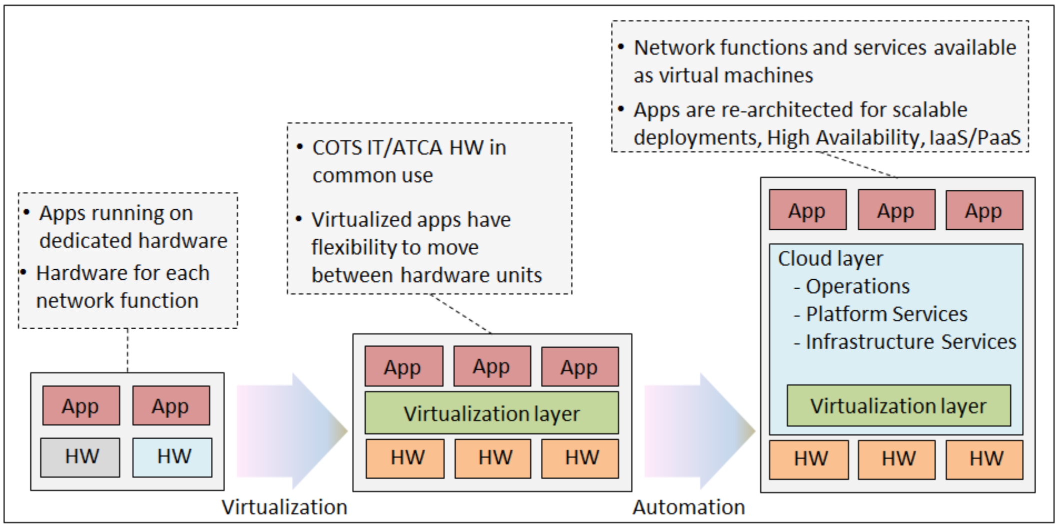


Figure 19. *A* high level path from the current architecture to a cloud controller architecture *[93]*.

As a part of this thesis an experiment was carried out related to the virtualization steps depicted in the above figure. It is small a step on the way of common 2G/3G cloud controller.

5.1 Virtualized Base Station Controller

Virtualizing underlying hardware of BSC gains similar benefits as discussed generally for IT world earlier. With the current BSC architecture, hardware independence and fast deployment are the biggest ones. Virtualization is a fundamental for further cloud evolution hence it definitely has interest among telecom operators. Practically in virtualization step the current software is just ported on virtualized environments. With BSC architectural improvements (discussed more in Chapter 5.4) also dynamic scaling of resources would be a benefit.

Main drivers in virtualization of mobile networks are LTE technology driven. Equipment vendors are just about starting their studies and demos for BSC or RNC while in core network or in 4G in general the virtualized network elements are already available [94]. For BSC (introduced in chapter 3) there can be many alternatives but one is depicted in Figure 20.

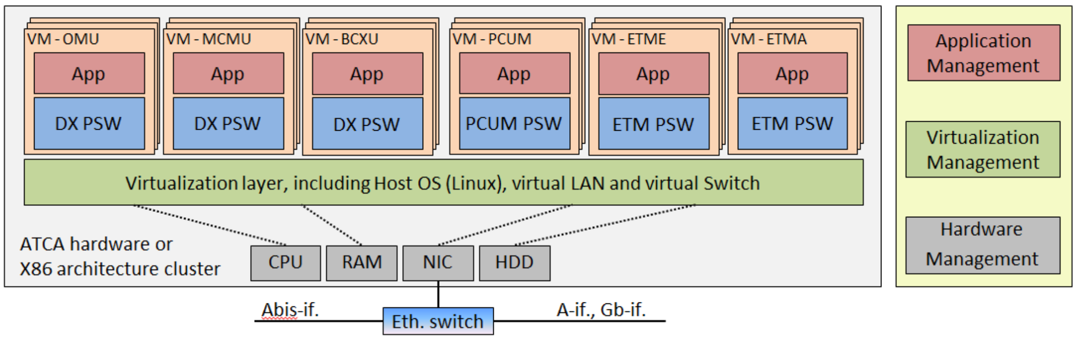


Figure 20. *Example of virtualized Multicontroller BSC environment.*

In the above example (Figure 20) hardware can be general purpose IT (x86) or ATCA hardware. Host OS can be Linux. Each virtual unit (corresponding to functional unit in the legacy architecture) can run in an own VM with current OS, platform software and applications. VMs can have virtual network interface between each other via virtual switch (Open vSwitch for instance). For outside networking to peer network elements there can be common physical network switch and traffic for each unit is separated with Virtual LAN (VLAN). Hardware and host OS, virtualization layer and virtualized applications are managed with separate management applications like OpenStack.

Although virtualization has benefits it also needs to fulfil carrier grade High Availability (HA) requirements set for current standalone equipment. Resiliency is one important factor for a highly fault tolerant telecom system. From virtualization point of view, strong VM resiliency is needed for IT server hardware used in data centres.

When the BSC software is ported to virtualized environment, current functional units (refer chapter 3.2) are mapped directly to VMs. Current redundancy mechanisms (2N, N+M, N+) are still supported by the platform and used as currently. This is problematic because the platform is not anymore aware of the underlying hardware. When configuring the network element, VMs must be located to server hardware in such a way that a failure of a server does not interrupt the functionality of the network element (for example VMs in the same 2N redundancy group are on different physical servers). If several critical VMs are on the same server, the current redundancy functionality might not be able to recover the network element. Running each VM on a dedicated server would be the most straightforward solution from the current redundancy functionality point of view but it does not provide optimal usage of hardware. Amount of VMs serving N+M or N+ redundant functional units, can be increased (scale out) or decreased (scale in), depending on capacity need.

Virtualization technique is selected based on experiments. Latency in virtualized system shall be at the same level as in legacy hardware. To gain needed performance it is possible that different techniques are mixed, like paravirtualization and hardware assisted virtualization. Especially CPU and NIC virtualization shall cause very minimal latency. Also hypervisor selection should be flexible and possible selectable by telecom operator. There might be multi-vendor virtualized environment, and even different hypervisor types could be supported between VM live migration.

5.2 Common virtualized controller

As a part of further controller evolution in 2G and 3G single RAN, one possible way would be combine functionalities of BSC and RNC. As discussed earlier in chapter 2.2.1 there are already some solutions from telecom equipment vendors where the proprietary modular hardware can be used for either BSC or RNC with software upgrade only.

The next step would be to run virtualized BSC and RNC on common hardware at the same time. Figure 21 depicts an example where (2G) BSC and (3G) RNC former FUs (Functional Unit) are running on in separate VMs.

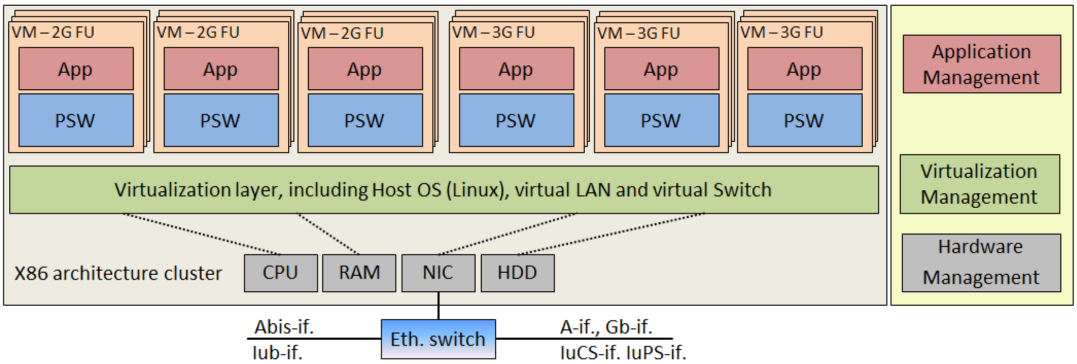


Figure 21. *Common 2G/3G controller in virtualized environment.*

In the above example (Figure 21) hardware can be general purpose IT (x86) or ATCA hardware, which is virtualized. Host OS can be Linux. Each virtual unit (corresponding to Functional Units in legacy 2G and 3G architectures) can run in own VM with current OS, platform software and applications. In the above example platform software is harmonized, meaning all the virtual units runs on common platform. Each VM could also have its current platform software and OS. VMs can have virtual network interface between each other via virtual switch (Open vSwitch for instance). For outside networking to peer network elements there can be common physical network switch and traffic for each units are separated with Virtual LAN (VLAN). Hardware and host OS, virtualization layer and virtualized applications are managed with separate management applications.

The concept can be extended to other radio technologies besides 2G and 3G, as suggested by WWRF [10]. There could be applications handling LTE, WiFi or Wimax related functionalities running on their own FUs/VMs.

5.3 Cloud Controller

As discussed earlier, mobile network cloudification has been focused on LTE and core network in general. However it is seen that cloud technology is just taking the first steps in mobile networks. Currently used dedicated hardware will be used a long time together with evolving cloud and network virtualization technologies. As well as 2G and 3G radio technologies will coexist with 4G (and further) for many years to come, especially the highly penetrated 2G as global umbrella coverage for 3G and 4G. Each of them has also specific characteristics, which are suitable for different services [60]. For example IoT appliances, which have become more common, might require only low data speed and amount but wide network coverage. This can be well handled with 2G technology. On the other hand High Definition video streams require high bandwidth and low latency where 3G/4G technology can best serve the needed service. That is one reason why 2G/3G and their key devices, controllers, must evolve as well. [42]

When NFV standards [54] are is evolved for enough mature level it is possible to virtualize different mobile network functionalities also in controller. Of course many prerequisite works can be already done. There could be also possibility to integrate core network and controller functions on same virtualized hardware running on common hardware platform [42]. In addition to NFV also SDN concept can be applied network forwarding level.

NFV could be applied also in BSC level itself by virtualizing all the current network functionalities. Those functionalities can be the same as described in chapter 3.3. Together with corresponding functionalities of RNC a single RAN controller can form a virtualized entity, which can offer RAN controller as a service in cloud manner. SDN concept can be utilized also by decoupling control plane and user plane. That could mean the cloud controller handles only CS/PS control plane and CS user plane goes through SDN forwarding devices (PS user plane need to go via PCUM). All the functionalities can be managed through open API of cloud platform, which could be OpenStack based. In Figure 22 is illustrated an alternative in high level how the earlier mentioned common controller can utilize NFV and SDN and form a cloud controller.

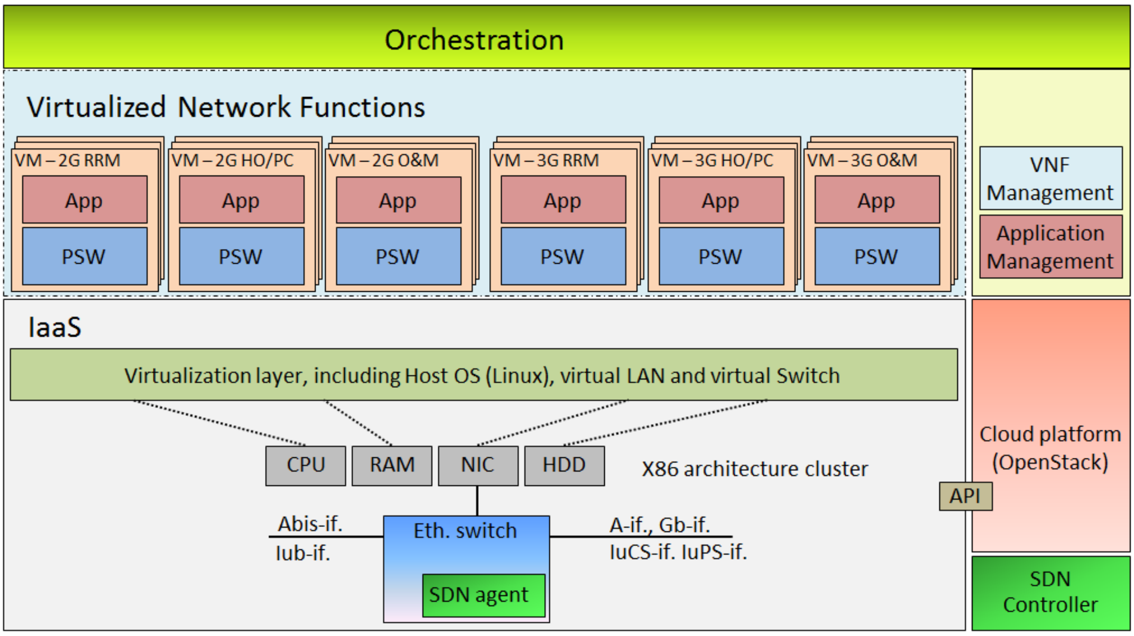


Figure 22. *Common BSC and RNC with NFV, SDN and cloud perspective, control plane and user plane decoupled.*

In the above example (Figure 22) hardware can be general purpose IT (x86) or ATCA hardware, which is virtualized. Host OS can be Linux. Each network function like RRM (Radio Resource Management), HO/PC (Handover and Power Control) or O&M (Operation and Maintenance) runs in a separate VM. In the above example platform software is harmonized, meaning all units have common platform. VMs can have virtual network interface between each other via virtual switch (Open vSwitch for instance). For outside networking to peer network elements there can be common physical network switch and traffic for each units are separated with virtual LAN (VLAN). With SDN approach traditional BSC control plane and SDN controller functions can be separated from user plane. It would mean that at least ETME/ETMA functionalities were separated to a SDN capable forwarding device (switch). For PCUM that might not be possible. Current Multicontroller BSC architecture however limits this approach (these improvements and limitations are discussed in the next chapter). Hardware and host OS, virtualization layer and VNF applications are managed with separate management applications via open APIs of cloud stack. Orchestration layer utilizes open interfaces for managing and automating functions in virtualized and cloudified environment.

5.4 Multicontroller BSC architectural improvements

Existing architecture was presented in Chapter 3. Requirements for original BSC and current cloud BSC are quite different due to needed transport network support for example. There could be however made architectural changes to more efficiently take the gradual steps towards cloud based controller. Generally cloud architecture could mean following characteristics in telecom environment:

- Load sharing group for processing units of control plane and user plane. This helps scaling the processing resources according to the traffic load. Separate load sharing groups for user and control planes make it possible to scale user and control planes separately.
- Dedicated VMs for control plane and user plane applications. This helps in the scaling more accurately.
- Dedicated VM for centralized applications to maintain the NE internal information such as processing unit and radio network configuration. Database should be open source based.
- Common terminating endpoint for external interfaces of the NE. This keeps scaling of NE internal processing units invisible for peer NEs. The same functionality can act also a load balancer, which shares the traffic load between processing resources.
- OS should be open source based in NE processing units, in practice Linux.
- All management should be done via open interfaces.
- VM can be live migrated to another host computer in case of failure. The load balancer needs to be aware of such changes.

As currently only Ethernet/IP is supported on external interfaces, roles of the current functional units could be reconsidered. Following subchapters present what the changes could mean from the BSC point of view.

5.4.1 Functional unit and radio network association

Currently control plane and management plane from the radio network (Abis interface) are terminated at BCXU in BSC end. Control plane for each TRX is configured to a certain BCXU, and the management plane for each BTS is configured to a certain BCXU (management plane of BTS and control planes of TRXs below that can be also configured to different BCXUs). Sharing the load between BCXU units is not possible without reconfiguration of the radio network. Abis interface user plane is terminated at ETME, which is configured by O&M for each BTS.

From dynamic scalability point of view it would be better if serving BCXU and ETME are selected for each BTS transaction, depending on the load situation in the units. Instead of current N+1 (BCXU) and N+M (ETME) redundancy models, the SN+ would be more suitable for this purpose (as ETMA already is). It would then be better possible to scale in/out the needed amount of BCXU and ETME units depending of traffic load. Information of the radio network association for serving functional unit could be stored into centralized database or management plane VM.

5.4.2 PCUM redundancy model

As discussed earlier, N+M redundancy model is currently used for PCUM. Due to historical reasons PCUM still have logical association with BCXU although there is no physical association anymore.

SN+ redundancy model is preferred also for PCUM. Currently cells are configured for PCUM by O&M. Cells should be allocated to PCUM dynamically. Currently PS data in the Abis interface is going via ETME to PCUM. When redundancy models are changed and load balancer introduced (refer chapter 5.4.4), Abis interface can go directly to PCUM.

5.4.3 ETME and ETMA optimization

As only Ethernet/IP is supported on external interfaces, the ETME/ETMA architecture could be optimized. Originally separate user plane termination units for Abis and A interfaces were needed because the BSC had to support both TDM and IP options for both interfaces. Different connectivity combinations on Abis and A interfaces are no longer relevant in Multicontroller BSC and in virtualized environment. User plane processing in ETME and ETMA can be performed also in a single unit.

For performance point of view there could be only single ETM unit, which is SN+ redundant. It would also reduce the delay, which comes when packet are first processed by ETME unit and internally forwarded to ETMA unit (and vice versa). In virtualized environment where each functional unit runs in own VMs it would also reduce overhead caused by packet forwarding between VMs.

5.4.4 Common load balancer

On top of the previous proposals, there could be a single gateway terminating all interfaces. The same functionality could act as load balancer by selecting the best suitable unit in a load sharing group. User plane could go via the load balancer or directly to the selected unit. There would be the following load sharing groups: ETM for CS user plane, BCXU for CS control plane and PCUM for both PS control- and user plane. **Error! Reference source not found.** illustrates the combined Packet Gateway and Load Balancer, load sharing groups and how CS/PS and CP/UP flow goes through BSC. In this proposal each interface has dedicated IP address for clarity.

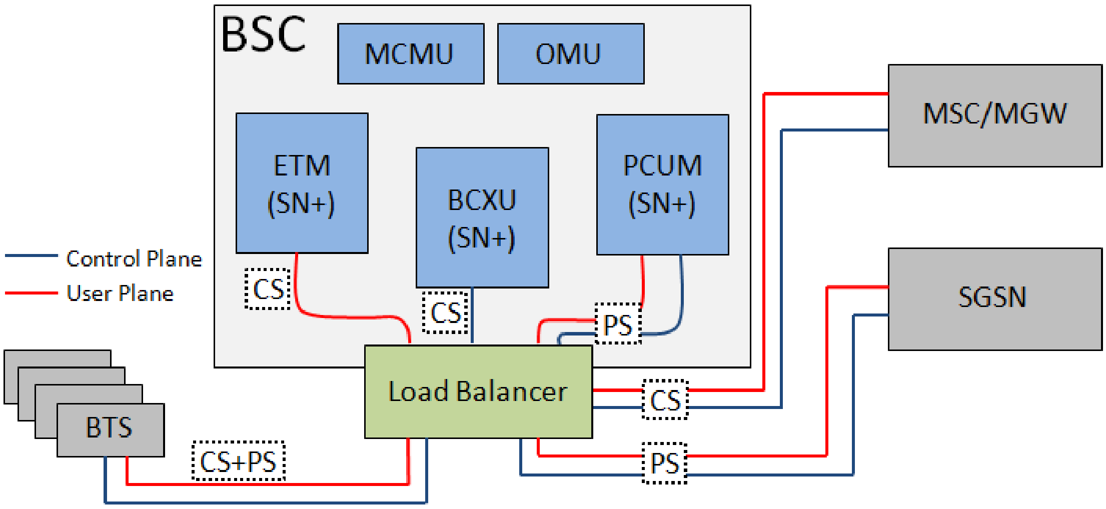


Figure 23. *Suggestion for load balancer functionality in evolved Multicontroller BSC architecture.*

Single interface towards external networks hides the internal architecture and configuration from other network elements and transport network. This makes dynamic scalability possible.

5.4.5 MCMU scalability

Centralized functions, such as radio resource management and radio network configuration management, are located in MCMU. Currently these functions are not scalable (which means those cannot be distributed) but during BSC evolution they have become extremely large and complex. Those tend to overload the CPU of the MCMU, like during functional unit or BSC restart. Capacity of the centralized functions becomes a bottle neck when the amount of control plane units is increased.

Distribution of centralized functions should be studied. Maybe some of those functions could be distributed to control plane units.

Instead of 2N redundant MCMU, centralized data could be stored in a database which has its own resiliency methods. For performance reasons, the database shall be distributed and memory based.

6. PRACTICAL EXPERIMENTS

Initially the Experiment 1 was planned to be the practical part of this thesis. However, it was found out already during the first phases that the experiment faced too many difficulties. As result it had to be interrupted and new practical part was selected. In chapter 6.1 it is still described what was the main idea and what kind of difficulties were faced.

Experiment 2 was then selected as the main practical part for the thesis. Experiences from that are also described below.

6.1 Experiment 1, RRM in virtualized BSC

Target of the experiment was to study feasibility of using common platform for virtualized BSC and RNC (as described in chapter 5.2). Compiling the whole BSC application SW in the RNC development environment against the RNC platform would have been too big task.

Also more SW changes would have been required. Therefore, it was decided to use only one application program block in the experiment. RRM (Radio Resource Management) was selected because most of the services it uses are supported in the RNC platform. Finally the purpose was to run some basic module test to verify that the functionality is similar as in legacy environment, in Multicontroller BSC.

6.1.1 HW and SW architecture

In the experiment virtualized COTS IT server hardware would have been used. Each functional unit would have run in a separate VM. After more detailed investigation it turned out that building the whole virtualized BSC environment needed much more effort what initially expected. Due to this, different approach was selected. In that approach the idea was to build a special software combination from already existing virtualized RNC platform and application software, and the BSC application integrated with the RNC applications software build. Software architecture is described in Figure 24.

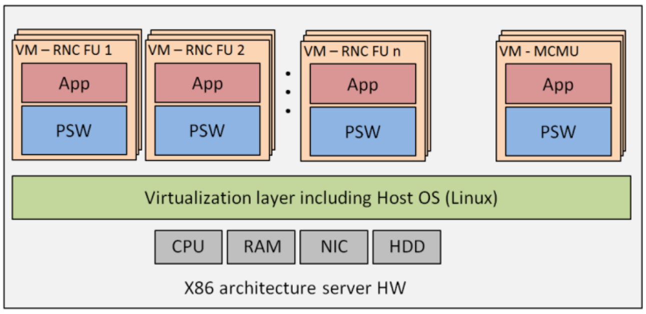


Figure 24. *HW and SW architecture in Experiment-1.*

The hardware and virtualization environment would have been common to RNC and BSC, as well as the PSW (Platform Software), which includes Guest OS. Each RNC FU runs in separate virtual machine and with special arrangements one of the FUs would have changed to be BSC FU. In RRM case the FU is MCMU. Virtual networking (VLANs and vSwitch) would not be *necessarily* needed as real communication is not needed between VMs.

6.1.2 Building application SW block for testing

Compiling application in BSC environment differs a lot from compiling in RNC environment. At the beginning this was not assumed to be a big issue but it proved to be different.

At First it was noticed that RRM uses several platform services, which were not available by RNC platform software. Dummy functions were implemented for faking RRM that the services would be available. Also some application specific file based services were bypassed with dummy implementation.

However the main problem was faced with the different compiling environment, which is Linux based in RNC side. Every phase of compilation process had difficulties, like translating TNSDL code to C, or linking existing C code or libraries with the TNSDL code. The compilation attempt was constant attempt-fail process, which would have needed more interacting co-operation at the same site. During the compilation attempts the BSC specific environment definitions were mostly used to exclude possible issues caused by the different definitions. Still it did not succeed.

As the experiment would have needed much more activities after the compilation, as briefly described earlier, and usable time was expiring, it was decided to interrupt this experience. Experiment 2 was selected instead.

6.2 Experiment 2, performance comparison

This study builds on another activity in ongoing BSC virtualization feasibility investigations at the same time as experiment 1 was attempted. In that activity CS scenarios (voice calls, SMS and location area updates) were run on virtualized environment, with DX based platform and BSC CS control plane application layer.

It was then decided to make performance comparison of the selected CS control plane based functionality between the legacy Multicontroller BSC and the experimental virtualized environment. CPU load was measured in both environments with similar usage.

6.2.1 SW and HW architecture in the experiment

Only OMU, MCMU and BCXU units were running on virtualized environment. PCUM, ETME and ETMA units were not yet ported to x86 based virtualized architecture. PCUM functionality (PS user plane and PS control plane) and ETME/ETMA (CS user plane) were run on the legacy proprietary hardware. Figure 25 depicts the configuration.

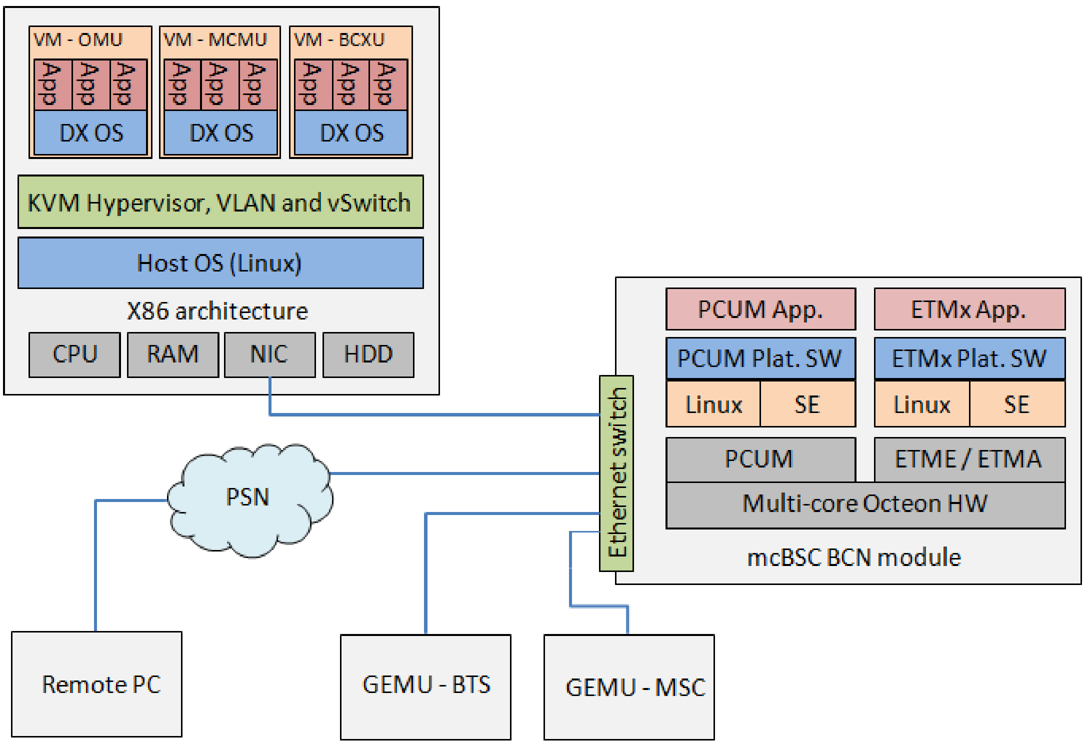


Figure 25. HW and SW architecture in Experiment-2.

The Multicontroller BCN module contains an Ethernet switch (as depicted also in Figure 5), which normally is configured for external interfaces of the BSC. In the test environment virtualized hardware and GEMU (simulating BTSs and MSC) are connected directly to it. For the remote connection of the test environment the Ethernet switch has interface to the enterprise's Packet Switched Network (PSN). Also real network elements BTS, MSC/MSS and SGSN can be connected to test setup via the same PSN but that was not in the scope of this performance comparison.

Some extra performance overhead caused by forwarding control plane packets between BCXU and ETME/ETMA via the Ethernet switch can be assumed. This is not taken into account in the performance comparison as it is considered to be minor. In further testing also PCUM/ETME/ETMA functionalities needs to be virtualized as illustrated in Figure 20, which will decrease that overhead. On the other hand, virtualization (and hypervisor layer) generates overhead itself.

Host OS was CentOS Linux release 6.4, 64-bit. The x86-architecture hardware mentioned in Figure 25 was a standard IT rack server, here HP Proliant (also in Figure 26). Its rack size is 1U, memory 16GB and HDD 1 TB. Multicontroller BSC was discussed earlier (Chapter 3.2).

Figure 26. *Figure 26. IT rack server (HP DL320e Gen8) used in Experiment 2. [95]*

From performance comparison point of view the CPU comparison is the most important. Table 4 below gathers the main characteristics of the CPU's used in test environments.

Table 4. *Comparison of CPU used in IT rack server and Multicontroller BSC (MCMU and BCXU computer units) [96].*

	IT rack server	Multicontroller BSC unit
CPU Architecture	x86, 64-bit	x86, 64-bit
Processor Base Frequency	2,90 GHz	1,73 GHz
# of Cores	2	2

KVM was already selected as hypervisor in the virtualized environment before the performance comparison was performed. As discussed, it is a popular selection, also in real-time demanding telecommunication industry. It could be studied more whether another hypervisor would more effective in the BSC environment. Also Linux container (LXC) usage could be studied as there is no hypervisor at all (might reduce overhead).

6.2.2 Performed tests and comparison

A relatively small radio network was used, Table 5 gathers the details. However, the used traffic profile was aggressive causing heavy CS Control Plane signalling traffic.

Roughly estimated, it is four times heavier traffic profile than in normal product performance testing. Selected traffic profile is affecting to applications especially in BCXU units hence their CPU load is monitored. CPU load in OMU and MCMU was also measured but not presented in graphs. By gradually activating more signalling traffic from the radio network it can be then estimated how CPU load trend would increase in theory with bigger radio networks.

Table 5. *Used radio network and traffic profile of each MS.*

Radio network	
Total BTS amount	48
Total TRX amount	96 (2 TRXs per each BTS)
MS amount per BTS	14
Total Radio timeslots	768
Total MS amount	672
Traffic profile	
CS call	30s (duration, repeated constantly)
MS originated SMS	every 9s
Location update procedure	every 6s

Traffic load is activated gradually by unlocking 16 BTSs at the time in the BTS-GEMU (BTS GSM Emulator). The BTS-GEMU then starts signalling traffic generation according to the defined traffic profile. Test continued until all 48 BTS were unlocked and all MSs were generating signalling traffic. Transaction amount presented in below figures 28 and 30 were taken from KPI (Key Performance Indicator) counters of the BSC.

There are three differences between virtualized BSC environment and Multicontroller BSC environment in test setups. The first two are related to CPU. Firstly, in virtualized environment the CPU clock frequency is higher (2,9 GHz vs. 1,73 GHz). Secondly the Virtual Machines of OMU, MCMU and BCXU are each running in a dedicated single CPU core, but in Multicontroller BSC, the DX platform can utilize dual cores by scheduling different application and platform processes in different cores. Performance increase factor in case of dual core CPU compared to single core has not been tested accurately. However, there is general experience that with BSC application the performance increase factor with dual core is 1,5 (not 2, which would be theoretical maximum). In this comparison the same factor is used. The benefits depend very much on how platform and application software can utilize multiple cores. In a study it was noted with certain type of applications, that the benefit can be negligible [97]. The third difference is generated traffic. It was not clear why transaction amount in Multicontroller setup was roughly 2/3 of the amount in virtualized environment. The same traffic profile was used, and there was nothing suspicious in BSC internal functionality, and BTS GEMU seemed to work normally. It was decided not to further investigate the reason for the difference. Instead, the difference was taken into account by scaling the traffic loads to be nearly equal in both BSCs.

Performance measurement in virtualized BSC

In virtualized BSC the CPU load was measured from Host OS point of view by taking Linux *top* printout in 1 second interval through the whole testing period. From the printout the CPU load caused by VM for BCXU unit was separated and plotted, refer Figure 27. There was a clear trend that the CPU load gradually increased along the increase in generated traffic. In case of MCMU and OMU, there was also slight trend but not that clear as in case of BCXU.

CPU load was measured also in Guest OS point of view. Similar kind of *top* printout provided by DMX OS was taken also in 1s interval. There was similar kind of trend in CPU loads, although load levels were a bit lower. However, in this case it was decided to compare the Host OS view with the Multicontroller units. By that way the OS has more reliable view on CPU load as there is no hypervisor layer, which tends to distort the Guest OS view of physical CPU load.

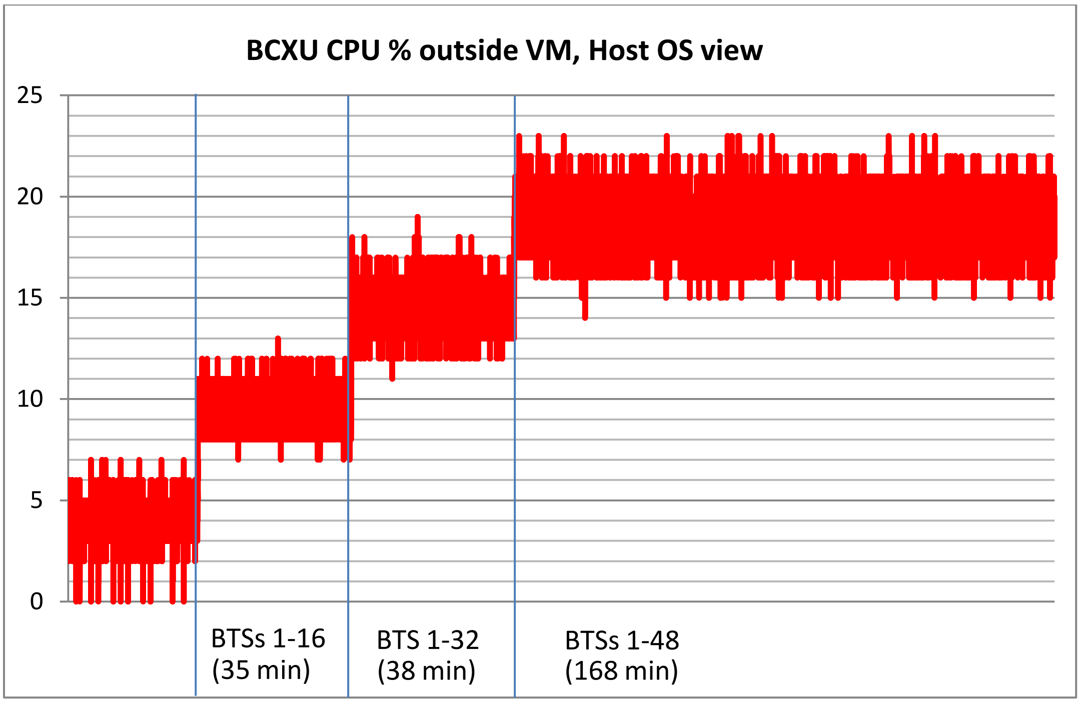


Figure 27. *CPU load % caused by the BCXU VM from Host OS (Linux) point of view.*

Figure 28 represents the transaction amount in certain time, which was generated from GEMUs to virtualized BSC. Time period is the same as in Figure 27. Vertical lines indicate places where traffic amount of each bundle of BTSs has reached its maximum.

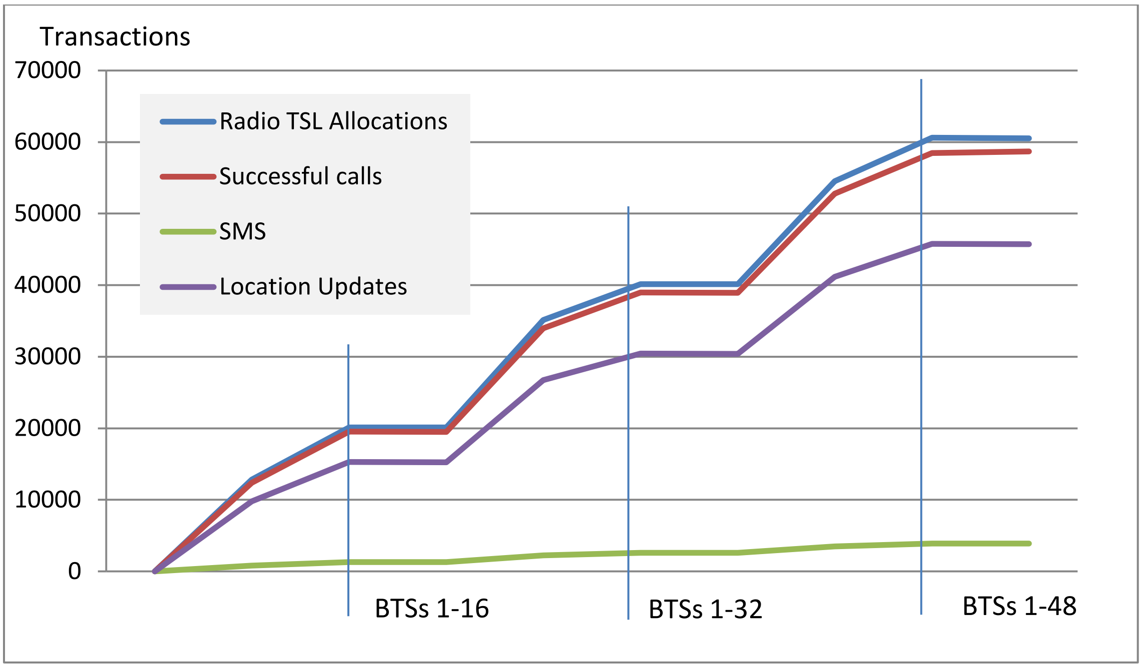


Figure 28. *Generated CS control plane traffic (number of transactions in certain time) in virtualized BSC as per traffic profile described in Table 1.*

Performance measurement in Multicontroller BSC

In Multicontroller BSC the CPU load was measured in a similar way as in the Host OS case in virtualized BSC. Figure 29 represents the results. BCXU CPU load trend due to transactions generated by the GEMU was also quite clear. The overall CPU load level was clearly lower.

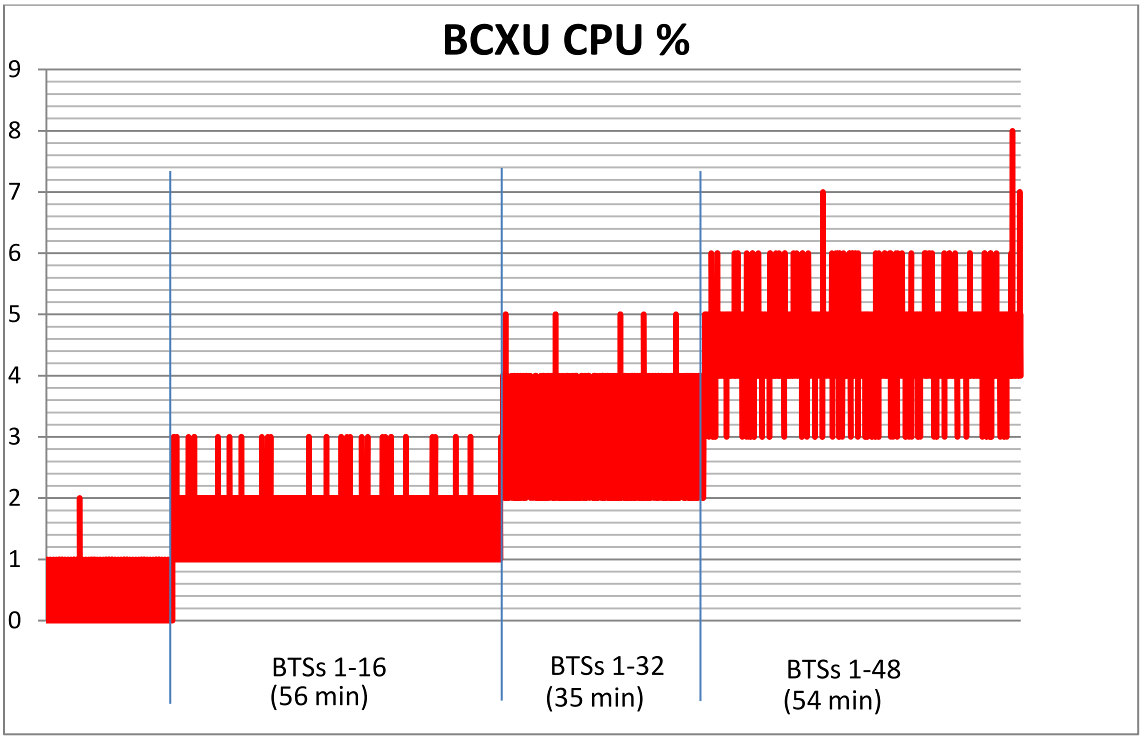


Figure 29. *Total CPU load % of the two CPU cores in Multicontroller BSC BCXU.*

Figure 30 represents the number of transaction, which were generated from GEMUs to Multicontroller BSC. Since there are two CPU cores the measured load is average load of those. Time period is the same as in Figure 29. Vertical lines indicate places where traffic amount of each bundle of BTSs has reached its maximum.

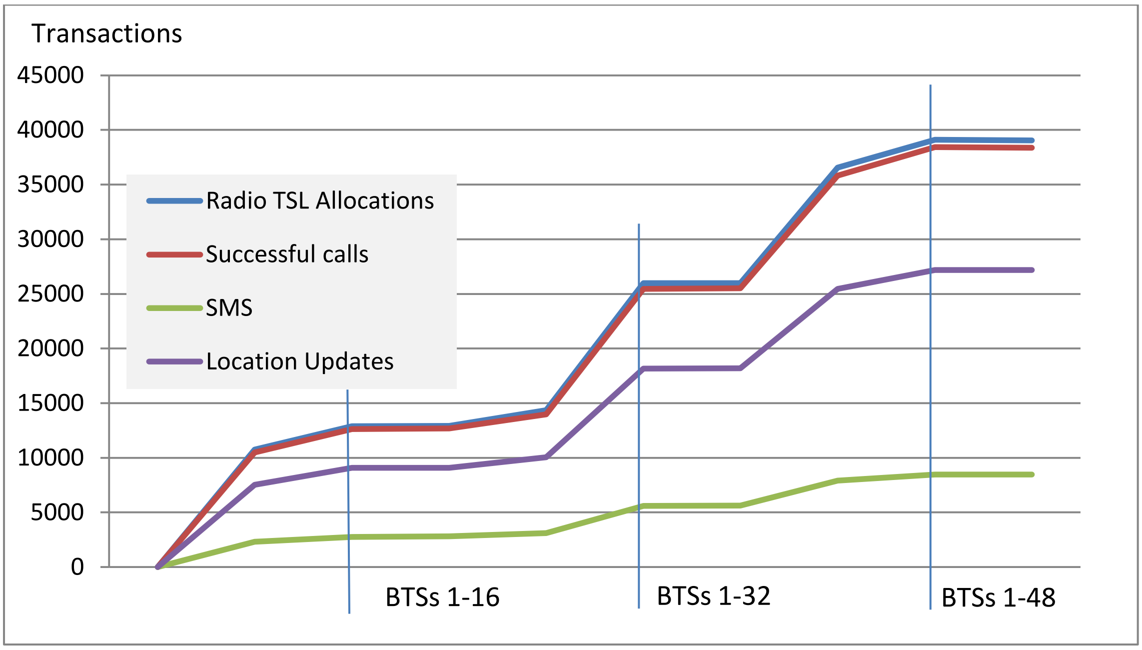


Figure 30. *Generated CS control plane traffic (number of transactions in certain time) in Multicontroller BSC as per traffic profile described in Table 1.*

CPU load percentage as a function of the number of TRX

The average CPU load per each bundle of activated traffic was estimated from the above graphs (Figure 27 and Figure 29). In CS control plane capacity point of view it seems that CPU of MCMU and OMU are not getting as linearly loaded as it does in CPU of BCXU. It is as expected as most of the functionality for CS control plane takes place in applications of BCXU. When nominal capacity of the BSC is measured, maximum 60% CPU load is allowed. Therefore, also in this experiment, 60% CPU load was taken as the limit when determining the BCXU capacity. CPU load generated by the first three bundles of BTS/TRXs were measured. The rest, extrapolated load values, increased also quite linearly when the traffic increased (9,6% -> 14,5% -> 18,3% and 1,4% -> 3,0% -> 4,4%). This makes the estimation for further traffic more reliable. In practice that would not be that idealistic as there are many other aspects affecting the linearity of the extrapolation. Might be that in higher CPU loads the TRX amount per certain CPU load percentage value is not always the same. That might bring more inaccuracy in extrapolation in highest CPU loads. Table 6 represents the BCXU CPU load percentage with measured and estimated TRX amounts. Measured CPU load is the average load during each period. In brackets there is standard deviation from the average load.

Table 6. BCXU *CPU load percentage as a function of the number of TRX in virtualized BSC and Multicontroller BSC with the selected traffic profile.*

BTS amount	TRX amount	Virtualized BSC CPU load	Multicontroller BSC CPU load
16	32	9,6 (0,9)	1,4
32	64	14,5 (1,1)	3,0
48	96	18,3 (1,7)	4,4
Extrapolation	128	22,8	5,9
	160	27,2	7,4
	192	31,5	8,9
	224	35,9	10,4
	256	40,2	11,9
	288	44,6	13,4
	320	48,9	14,9
	352	53,3	16,4
	384	57,6	17,9
	400	60,0	19,4
	416		20,9
	448		22,4
	480		23,9
	512		25,4
	544		26,9
	576		28,4
	608		29,9
	640		31,4
	672		32,9
	704		34,4
	736		35,9
	768		37,4
	800		38,9
	832		40,4
	864		41,9
	896		43,4
	928		44,9
	960		46,4
	992		47,9
	1024		49,4
	1056		50,9
	1088		52,4
	1120		53,9
	1152		55,4
	1184		56,9
	1216		58,4
	1248		59,9
	1250		60,0

Scaling the results

The three main differences need to be taken account when comparing the results.

1) Factor for comparing CPU clock speeds is **1,68** (2,9 GHz for virtualized BSC versus 1,73GHz for Multicontroller BSC BCXU)
2) Factor for comparing VM running in single core (in virtualized BSC) to dual core CPU (in Multicontroller BSC BCXU) is **1,5**.
3) Rough factor for difference in generated traffic is **1,60** It covers all transaction types (illustrated in Figure 28 and Figure 30) when the maximum amount of traffic is being generated.

Both IT rack server CPU and Multicontroller BSC BCXU CPU has 64-bit CPU architecture. However, host OS in IT rack server is 64-bit but DX OS in the Multicontroller BSC BCXU is 32-bit. The memory mount in BCXU unit does not exceed the 4 gigabyte hence it would not benefit of the 64-bit architecture [96], [98]. Due to this, this difference was not taken account in comparison.

Margin of error in these assumptions is quite a high. This estimation method means that following efficiency factor is determined (bolded factors above):

$$\frac{1250 \text{ TRX x } 1{,}68}{400 \text{ TRX x } 1{,}5 \text{ x } 1{,}60} = 2{,}19$$

Based on that it can be said that for this particular traffic profile it is needed 2,19 times more virtualized BCXU VMs (one VM per CPU core) compared to a Multicontroller BSC BCXU unit.

From the performance comparison point of view the best estimation for the efficiency factor was the most interesting. Optimistic and pessimistic efficiency factor were estimated by using factor-2 (single/dual core) as 1,2 and 1,8. Factors 1 and 3 can be assumed to be same. That gives the efficiency factor 2,73 (with 1,2) and 1,82 (with 1,8).

Other metrics

Also power consumption is an interesting metric. Power consumption for one IT rack server is 120W [98] and for one Multicontroller module 800W [28]. As one Multicontroller BSC module can host 8 computer units [28] it can be estimated that BCXU unit consumes then 100W. When the efficiency factor is taken into account the IT rack server running one BCXU VM consumes 2,63 times (2,19 * 120W / 100W) more power than Multicontroller BCXU in order to provide the same capacity. However, in this server there are 2 CPU cores available hence power consumption per one VM (BCXU unit) is roughly 1,3 more than Multicontroller BSC BCXU unit.

Also physical space consumption might be interesting metric. Used IT rack server takes 1U (from standard server rack), and in this case one server has 2 CPU cores. As the efficiency factor was roughly 2, it means that one (used kind of) IT rack server is needed per each Multicontroller BSC BCXU. Multicontroller module (Figure 4) takes 4U (from standard server rack) but there can be 8 computer units (like BCXU) per one module. So, it can be said that with this server one rack server takes twice amount of space. In other words, 8U rack space is needed for IT rack servers whereas a 4U Multicontroller BSC module (with 8 BCXUs) is needed for the same performance.

7. CONCLUSIONS

7.1 Cloud controller

Generally the virtualization domain with all the Telco related technologies was quite new for the writer before. It was very interesting to study concepts like NFV, SDN, OpenStack and Single RAN, and consider them for the current Multicontroller BSC. Understanding better the domain from telecom operator's point of view helped also understand why such a revolution is happening.

It became evident that the whole telecom industry is in the middle of big change. Virtualization and cloud technologies are strongly present in many publications. There are large amount of studies of virtualization and cloudification aspects but they mainly focus on the latest technologies, mainly 4G. There were not many studies from the BSC point of view. It seems that all the available studies were still very high level proposals and focusing more to the BTS end. Practically no studies of BSC perspective in virtualization were found. Also the proposed practical steps were strongly vendor specific as they all have their own proprietary hardware and software, which they need to customize. For that reason the studies by vendors tend to be non-public and mainly optimistic marketing material was available.

It was also found that the telecom equipment vendors are studying cloud (or centralized) RAN concept with high interest. Current RAN controller architectures, especially 2G BSC, have been originally designed for very different requirements to what cloud environment currently needs. This brings extra challenges to adapt cloud architecture principles in RAN platform and application software. Some ideas were mentioned in Chapter 5.4. It might be that BSC would not be cloudified alone but as a part of a bigger SRAN concept it is a topic that needs to be considered in the future.

7.2 Experiences from the practical experiments

The practical experiments of the study, eventually in two different areas, were helpful in providing a better practical understanding of virtualization. Linux was strongly present in both experiments, which also taught a lot to the author, whose Linux skills were at quite a basic level.

Experiment-1 (study feasibility of using common platform for virtualized BSC and RNC) proved out to be quite instructive even though it could not be finished, as making major changes to the software building environment of the current SW turned out to be too difficult. One reason was that the author had very few competences in that area. Also aligning the schedule with the support persons and working in different physical locations brought additional challenges. For future practical work in this area more studies in SW build area would be needed, and also for the other SW process steps.

Experiment-2 (performance comparison between the legacy Multicontroller BSC and the experimental virtualized environment) was completed quite successfully. With a relatively small but quite signalling oriented traffic profile, it was possible to actually measure the CPU load behaviour in virtualized BSC control plane. The results were compared with the measurements of the same traffic profile in the current proprietary BSC. As for the result, it was encouraging that CPU processing expense per TRX was at a reasonable level, although cost (caused by needed hardware investments) was not directly evaluated in this thesis. However, CS control plane performance was roughly two times weaker in the virtualized IT environment than in the dedicated HW environment (per one CPU core). This is not acceptable in a commercial product and hence BSC architecture changes should be considered also from the performance point of view.

7.3 Future work

Although the experiments were related only to a small step in the whole virtualization - and further cloudification - roadmap, it still provided encouraging promise of IT server capability in RAN. Definitely further studies, testing and demonstrations will be needed before firm conclusions can be made on BSC virtualization. Further study items could be related to virtualization of CS/PS user plane functionalities. Finally the whole virtualized BSC with interfaces to all real peer network elements could be tested. That would mean also heavy performance testing. By that way it can be demonstrated how CPU capacity in a virtualized BSC is utilized with full radio network configuration and traffic volume. Virtualized BSC needs to meet at least the same resiliency requirements as Multicontroller BSC and possibly even to exceed them. That could mean BSC architectural changes as discussed in Chapter 5.4, which would definitely also be one key area for further studies. Generally it needs to be evaluated, which architecture changes will be the most important in order to make the BSC optimal for the cloud.

As discussed in high level abstraction in Chapter 5, the next focus areas after virtualization are related to integrating cloud functionalities with the virtualized hardware. Also standardization needs to be followed. It is very important that cloud BSC has open application interfaces (API) to every needed direction.

REFERENCES

[1] S. Elby, "Software Defined Networks: A Carrier Perspective," Verizon, 19 11 2011. [Online]. Available: http://www.opennetsummit.org/archives/apr12/site/talks/elby-wed.pdf. [Accessed 27 8 2014].

[2] A. Császár, W. John, M. Kind, C. Meirosu, G. Pongrácz, D. Staessens, A. Takács, F.-J. Westphal, "Unifying Cloud and Carrier Network," *Proceeding UCC '13 Proceedings of the 2013 IEEE/ACM 6th International Conference on Utility and Cloud Computing,* pp. 452-457, 2013.

[3] S. N. D. Cai, "The Evolution of the Carrier Cloud Networking," Cisco Systems, 3 2013. [Online]. Available: http://www.ieeeexplore.net/stamp/stamp.jsp?tp=&arnumber=6525533. [Accessed 21 9 2014].

[4] ABI research, "The Internet of Things Will Drive Wireless Connected Devices to 40.9 Billion in 2020," ABI research, 20 8 2014. [Online]. Available: https://www.abiresearch.com/press/the-internet-of-things-will-drive-wireless-connect. [Accessed 27 8 2014].

[5] 3GPP, "LTE overview," 3GPP, [Online]. Available: http://www.3gpp.org/technologies/keywords-acronyms/98-lte. [Accessed 14 8 2014].

[6] S. K. Siddiqui, Roaming in Wireless Networks, McGraw-Hill Companies, Inc., 2006.

[7] J. Eberspächer et al., GSM – Architecture, Protocols and Services Third Edition, John Wiley & Sons, Ltd, 2009.

[8] H. Holma, A. Toskala, LTE for UMTS – OFDMA and SC-FDMA Based Radio Access, John Wiley & Sons, Ltd., 2009.

[9] Wikipedia, "OSI model," WIkipedia, [Online]. Available: http://en.wikipedia.org/wiki/OSI_model. [Accessed 12 8 2014].

[10] WWRF, "Multi-RAT Network Architecture," Wireless World Research Forum, 11 2013. [Online]. Available: http://www.wwrf.ch/files/wwrf/content/files/publications/outlook/Outlook9.pdf. [Accessed 15 8 2014].

[11] Huawei, "SingleRAN@Future," Huawei, 5 2011. [Online]. Available: http://fows.pl/wp-content/uploads/2011/05/FoWS-2011-SingleRAN@Future-Huawei.pdf. [Accessed 18 8 2014].

[12] Nokia, "Single RAN Advanced - general information of SRAN product

portfolio," Nokia, [Online]. Available: http://nsn.com/sites/default/files/document/nokia_single_ran_advanced_evolution_white_paper.pdf. [Accessed 16 8 2014].

[13] Analysys_Mason, "A SINGLE RAN STRATEGY CAN OFFER DOUBLE-DIGIT SAVINGS FOR LTE DEPLOYMENT: A REAL TCO COMPARISON," Analysys Mason, 28 11 2013. [Online]. Available: http://www.analysysmason.com/About-Us/News/Newsletter/Single-RAN-LTE-overlay-Oct2013-RDTW0/. [Accessed 15 8 2014].

[14] B. Haberland, F. Derakhshan, H. Grob-Lipski, R. Klotsche, W. Rehm, P. Schefczik and M. Soellner, "Radio Base Stations in the Cloud," *Bell Labs Technical Journal,* vol. 18, no. 1, pp. 129-152, 5 2013.

[15] R. C. H. S. X. Z. L. G. a. P. Z. I. Chih-Lin, "Toward Green and Soft: A 5G Perspective," *Communications Magazine, IEEE,* vol. 52, no. 2, pp. 66-73, 2 2014.

[16] T. Guo, R. Mahindra, S. Rangarajan, "Radio Access Network Virtualization for Future Mobile Carrier Networks," *Communications Magazine, IEEE,* vol. 51, no. 7, pp. 27-35, 7 2013.

[17] Ericsson, "Ericsson launches Evo RAN for sustainable radio networks," Ericsson, 13 2 2009. [Online]. Available: http://www.ericsson.com/thecompany/press/releases/2009/02/1290464. [Accessed 18 8 2014].

[18] Ericsson, "New generation controller to cater for mobile broadband growth," Ericsson, 10 2 2011. [Online]. Available: http://www.ericsson.com/news/120210_evo_244188811_c. [Accessed 18 8 2014].

[19] Huawei, "Huawei BSC6900 Multimode Base Station Controller Software Security Target," Huawei, 20 12 2011. [Online]. Available: http://www.commoncriteriaportal.org/files/epfiles/HUAWEI%20MBSC%20ST%201.07.pdf. [Accessed 19 8 2014].

[20] Alcatel-Lucent, "Alcatel-Lucent 9926 Distributed Base Station," Alcatel-Lucent, [Online]. Available: http://www3.alcatel-lucent.com/wps/portal/products/detail?LMSG_CABINET=Solution_Product_Catalog&LMSG_CONTENT_FILE=Products/Product_Detail_000568.xml&lu_lang_code=fr. [Accessed 18 8 2014].

[21] Samsung, "Samsung MBS," Samsung, [Online]. Available: http://www.samsung.com/global/business/telecommunication-systems/telecommunication-systems/multi-standard/smart-mbs. [Accessed 18 8 2014].

[22] Nokia, Nokia, [Online]. Available: http://nsn.com/portfolio/products/mobile-broadband/single-ran-advanced.

[Accessed 18 8 2014].

[23] J. H. W. J. Y. W. G. L. NGMN Alliance (C. Chen, "Suggestions on potential solutions to C-RAN by NGMN Alliance," NGMN Alliance, 3 1 2013. [Online]. [Accessed 9 9 2014].

[24] C. W. S. Next Generation Mobile Networks (NGMN), "RAN Evolution Project," China Mobile, EE, Deutsche Telekom, Telecom Italia, Telus, Alcatel-Lucent, Ceragon, Samsung, 5 2014. [Online]. Available: http://www.uppersideconferences.net/cloudran2014/pres-cran2014/day_1/Day_01_3_philippe_sehier.pdf. [Accessed 10 9 2014].

[25] Nokia, "Multicontroller Platform -- Multicontroller BSC and transcoder, Multicontroller RNC," Nokia, 2 2011. [Online]. Available: http://nsn.com/system/files/document/multicontroller_bsc_datasheet_0.pdf. [Accessed 14 8 2014].

[26] TelecomLead, "2G to 3G mogile broadband transition," Telecom Lead, 18 11 2013. [Online]. Available: http://www.telecomlead.com/news/2g-to-3g-mobile-broadband-transition-57936-45940. [Accessed 14 8 2014].

[27] Scribd, "GSM/EDGE BSS, RG20 Operating Documentation -- Integrated LAN Administration," Nokia, 2012. [Online]. Available: http://www.scribd.com/doc/198916699/Replacing-a-Lan-Switch. [Accessed 15 8 20014].

[28] Nokia, "Multicontroller BSC and Multicontroller Transcoder Product Description," Nokia, [Online]. Available: https://online.portal.nsn.com/. [Accessed 7 8 2014].

[29] Wikipedia, "19-inch rack," Wikipedia, 31 10 2014 (last modify). [Online]. Available: http://en.wikipedia.org/wiki/19-inch_rack. [Accessed 2 11 2014].

[30] Wikipedia, "Nokia DX200," Wikipedia, [Online]. Available: http://en.wikipedia.org/wiki/Nokia_DX_200. [Accessed 15 8 2014].

[31] Nokia, "Nokia internal documentation for platform functionaly," Nokia. [Online]. [Accessed 7 8 2014].

[32] 3GPP, "3GPP specifications," [Online]. Available: http://www.3gpp.org/specifications/specification-numbering. [Accessed 8 8 2014].

[33] Nokia, "The TNSDL book, 5th edition," 2013.

[34] Nokia, "DX 200 Software Architecture , internal documentation," 2001.

[35] D. ESI, "White Paper: IT Virtualization Technology and its Impact on Software Contract Terms," Department of Defense Chief Information Officer, 2012. [Online]. Available: http://www.esi.mil/download.aspx?id=3474. [Accessed 21 8 2014].

[36] N. R. D. Ruest, Virtualization: A Beginner’s Guide, Mc Graw Hill Companies, 2009.

[37] D. Kusnetzky, Virtualization: A Manager’s Guide, O’Reilly Media, Inc., 2011.

[38] J. Brodkin, "With long history of virtualization behind it, IBM looks to the future," NetworkWorld, 30 4 2009. [Online]. Available: http://www.networkworld.com/article/2254433/virtualization/with-long-history-of-virtualization-behind-it--ibm-looks-to-the-future.html. [Accessed 20 8 2014].

[39] S. Baca, "Virtualization for Newbies," Global Knowledge Training LLC, 2013. [Online]. Available: http://net-security.tradepub.com/free/w_gloc64/prgm.cgi. [Accessed 20 8 2014].

[40] M. Rosenblum, T. Garfinkel, "Virtual Machine Monitors: Current Technology and Future Trends," *Computer,* vol. 38, no. 5, pp. 39-47, 5 2005.

[41] J. Metzler, "Virtualization: Benefits, Challenges, and Solutions," Riverbed Technology, 2011. [Online]. Available: http://resources.idgenterprise.com/original/AST-0059219_Virtualization_-_Benefits_Challenges_and_Solutions.pdf. [Accessed 22 8 2014].

[42] ZTE, Intel, "Network Builders: Cloud Network Function," ZTE and Intel, 2 2014. [Online]. Available: http://networkbuilders.intel.com/docs/Network_Builders_Whitepaper_ZTE.pdf. [Accessed 5 9 2014].

[43] K. F. S. H. J. H. E. J. C. L. I. P. A. W. C. Clark, "Live Migration of Virtual Machines," *NSDI ’05: 2nd Symposium on Networked Systems Design & Implementation,* pp. 273-286, 2005.

[44] J. Rohrer, "Bare metal vs. hosted OS virtual solutions," Citrix experience by J. Rohrer, 19 2 2008. [Online]. Available: http://citrixxperience.com/2008/02/19/bare-metal-vs-hosted-os-virtual-solutions/. [Accessed 22 8 2014].

[45] A. Shah, "Kernel-based virtualization with KVM - Deep Virtue," *Linux Magazine,* no. 86, pp. 37-39, 1 2008.

[46] Flexiant, "Common_considerations_when_selecting_your_hypervisor," Flexiant, [Online]. Available: http://learn.flexiant.com/hypervisor-white-paper. [Accessed 23 8 2014].

[47] Open Virtualization Alliance, "KVM: choice in enterprise-class virtualization," Open Virtualization Alliance - Linux Foundation collaborative projects, [Online]. Available: https://openvirtualizationalliance.org. [Accessed 24 9 2014].

[48] J. F. Moore, "Linux is the fastest growing OS in the World," Lion's Wing Information Service, 4 2013. [Online]. Available: http://www.lions-wing.net/lessons/fastestgrowing/fastestgrowing.html. [Accessed 10 11 2014].

[49] VMWare, "Understanding Full Virtualization, Paravirtualization, and Hardware Assist," VMWare, 2007. [Online]. Available: http://www.vmware.com/files/pdf/VMware_paravirtualization.pdf. [Accessed 22 8 2014].

[50] StackExchange, "What is a Linux container and a Linux hypervisor?," 6 11 2014. [Online]. Available: http://unix.stackexchange.com/questions/166398/what-is-a-linux-container-and-a-linux-hypervisor. [Accessed 3 2 2015].

[51] Alcatel-Lucent, Telefonica, "Why distribution matters in NFV," Alcatel-Lucent, Telefonica, 2014. [Online]. Available: http://www.tmcnet.com/tmc/whitepapers/documents/whitepapers/2014/10462-why-distribution-matters-nfv.pdf. [Accessed 19 9 2014].

[52] ETSI, "Network Functions Virtualisation -- An Introduction, Benefits, Enablers, Challenges & Call for Action," ETSI, 10 2012. [Online]. Available: http://portal.etsi.org/NFV/NFV_White_Paper.pdf. [Accessed 26 8 2014].

[53] ETSI, "Network Functions Virtualisation," ETSI, 2014. [Online]. Available: http://www.etsi.org/technologies-clusters/technologies/nfv. [Accessed 26 8 2014].

[54] ETSI, "Network Functions Virtualisation (NFV) Use Cases," ETSI, 10 2013. [Online]. Available: http://www.etsi.org/deliver/etsi_gs/NFV/001_099/001/01.01.01_60/gs_NFV001v010101p.pdf. [Accessed 26 8 2014].

[55] OPNFV, "About OPNFV," Linux Foundation, Collaborative Projects, [Online]. Available: https://www.opnfv.org/about. [Accessed 3 2 2015].

[56] M. M. X.-N. N. K. O. T. T. B. A. A. Nunes, "A Survey of Software-Defined Networking: Past,Present, and Future of Programmable Networks," *Communications Surveys & Tutorials, IEEE ,* vol. 16, no. 3, pp. 1617-1634, 13 2 2014.

[57] R. J. Z. E. Feamster N., "The Road to SDN: An Intellectual History of Programmable Networks," 9 2013. [Online]. Available: https://www.cs.princeton.edu/courses/archive/fall13/cos597E/papers/sdnhistory.pdf. [Accessed 30 8 2014].

[58] Open Networking Foundation, "Software-Defined Networking: The New Norm for Networks," Open Networking Foundation (ONF), 13 4 2013. [Online]. Available:

https://www.opennetworking.org/images/stories/downloads/sdn-resources/white-papers/wp-sdn-newnorm.pdf. [Accessed 30 8 2014].

[59] A. D. L. O. P. S. A. B. L. M. C. H. J. J. C. Z. A. J. Bernardos, "An architecture for Software Defined wireless networking," *IEEE Wireless Communication,* vol. 21, no. 3, pp. 52-61, 6 2014.

[60] Y. L. D. J. L. Z. X. W. A. V. V. M. Yang, "Software-Defined and Virtualized Future Mobile and Wireless Networks: A Survey," *Mobile Networks and Applications,* pp. 1-12, 18 9 2014.

[61] S. P. R. Jain, "Network Virtualization and Software Defined Networking for Cloud Computing: A Survey," Washington University, 11 2013. [Online]. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6658648. [Accessed 25 8 2014].

[62] O. N. Foundation, "ONF Overview," Open Networking Foundation (ONF), 2013. [Online]. Available: https://www.opennetworking.org/images/stories/downloads/about/onf-what-why.pdf. [Accessed 29 8 2014].

[63] OpenDaylight, "OpenDaylight - About," OpenDaylight, [Online]. Available: http://www.opendaylight.org/project/about. [Accessed 30 8 2014].

[64] Floodlight, "Floodlight," Floodlight, [Online]. Available: http://www.projectfloodlight.org/floodlight/. [Accessed 30 8 2014].

[65] Open Network Foundation, "Wireless & Mobile Working group, overview," ONF, Wireless & Mobile Working group, [Online]. Available: https://www.opennetworking.org/images/stories/downloads/working-groups/charter-wireless-mobile.pdf.

[66] O. N. Foundation, "OpenFlow-enabled SDN and Network Functions Virtualization," ONF, 17 2 2014. [Online]. Available: https://www.opennetworking.org/images/stories/downloads/sdn-resources/solution-briefs/sb-sdn-nvf-solution.pdf. [Accessed 1 9 2014].

[67] M. Yang, Y. Li, D. Jin, L. Su, S. Ma, L. Zeng, "OpenRAN: A Software-defined RAN Architecture Via Virtualization," *ACM SIGCOMM Computer Communication Review ,* vol. 43, no. 4, pp. 549-550, 10 2013.

[68] P. Mell, T. Grance, "The NIST Definition of Cloud Computing," National Institute of Standards and Technology, 9 2011. [Online]. [Accessed 25 8 2014].

[69] P. Mell, T. Grance, "Effectively and Securely Using the Cloud Computing Paradigm," NIST, Information Technology Laboratory, 10 7 2009. [Online]. Available: http://csrc.nist.gov/groups/SNS/cloud-computing/cloud-computing-v26.ppt. [Accessed 25 8 2014].

[70] R.Bias, "“Hybrid” Clouds are Half-Baked," cloudscaling, 2 2 2010. [Online]. Available: http://www.cloudscaling.com/blog/cloud-computing/hybrid-clouds-are-half-baked/. [Accessed 25 8 2014].

[71] Microsoft, "Microsoft Azure," Microsoft, 2014. [Online]. Available: http://azure.microsoft.com/en-us/b14-34/. [Accessed 4 9 2014].

[72] Apache, "Apache CloudStack - Open Source Cloud Computing," 2014. [Online]. Available: http://cloudstack.apache.org/. [Accessed 4 9 2014].

[73] OpenStack, "Cloud Adminstrator Guide (current version)," OpenStack, 4 9 2014. [Online]. Available: http://docs.openstack.org/admin-guide-cloud/admin-guide-cloud.pdf. [Accessed 4 9 2014].

[74] Forrester Research, "State Of Cloud Platform Standards: Q1 2014," Forrester Research, 21 3 2014. [Online]. Available: http://www.forrester.com/State+Of+Cloud+Platform+Standards+Q1+2014/fulltext/-/E-res112621. [Accessed 4 9 2014].

[75] Openstack, "What is OpenStack," Openstack, [Online]. Available: http://www.openstack.org/. [Accessed 2 9 2014].

[76] B. Levine, "Nokia Offers Software-Defined 'Telco Cloud' with Juniper," B. Levine / NewsFactor Network, 5 5 2014. [Online]. Available: http://www.newsfactor.com/story.xhtml?story_id=0110006KDGZ7. [Accessed 2 9 2014].

[77] Alcatel-Lucent, "Carriers on the path to the cloud," Alcatel-Lucent and Heavy Reading, 2013. [Online]. Available: http://webform.alcatel-lucent.com/res/alu/survey/alu2CustomForm.jsp?cw=CWF273. [Accessed 2 9 2014].

[78] G. Karagiannis, A. Jamakovicy, A. Edmondsz, C. Paradax, T. Metsch, D. Pichonk, M.-I. Corici, S. Ruffinoyy, A. Gomesy, P. S. Crostazz, T. M. Bohnert, "Mobile Cloud Networking: Virtualisation of Cellular Networks," *Telecommunications (ICT), 2014 21st International Conference,* pp. 410-415, 16 3 2014.

[79] J. Erman, K.K. Ramakrishnan, "Understanding the Super-sized traffic of the Super Bowl," 2013.

[80] X. Zhiqun, C. Duan, H. Zhiyuan, S. Qunying, "Emerging of Telco Cloud," *Communications, China,* vol. 10, no. 6, pp. 79-85, 6 2013.

[81] B. a. A. S. Pocuca, "Possible scenarios for cloud-based offering by operators," NSN, Siemens, 5 2012. [Online]. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6240698. [Accessed 25 8 2014].

[82] S. Aleksic, I. Miladinovic, "Network Virtualization: Paving the Way to Carrier Clouds," in *Telecommunications Network Strategy and Planning*

Symposium (Networks), 2014 16th International , 2014.

[83] A. D. F. P. T. L. W. P. Bosch, "Telco Clouds and Virtual Telco: Consolidation, Convergence, and Beyond," Alcatel-Lucent (Bell Labs), 2011. [Online]. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5990511. [Accessed 19 9 2014].

[84] Nokia Solutions and Networks, "Network security - An imperative for the global economy," Nokia Solutions and Networks, 12 2013. [Online]. Available: http://nsn.com/sites/default/files/document/nsn_network_security_an_imperative_for_the_global_economy.pdf. [Accessed 15 9 2014].

[85] M. Vucinic, B. Tourancheau, F. Rousseau, A. Duda, L. Damon, R. Guizzetti, "OSCAR: Object Security Architecture for the Internet of Things," in *A World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2014 IEEE 15th International Symposium on* , 2014.

[86] 3GPP, "SA3 - Security," The 3rd Generation Partnership Project (3GPP), [Online]. Available: http://www.3gpp.org/specifications-groups/sa-plenary/sa3-security. [Accessed 15 9 2014].

[87] ITU-T, "Security architecture for systems providing end-to-end communications," ITU-T (INTERNATIONAL TELECOMMUNICATION UNION), 10 2013. [Online]. Available: https://www.itu.int/rec/T-REC-X.805-200310-I/en. [Accessed 15 9 2014].

[88] Nokia, "Technology Vision 2020, Reducing network latency to milliseconds," Nokia, 7 2014. [Online]. Available: https://nsn.com/sites/default/files/document/technology_vision_2020_reduce_latency_white_paper.pdf. [Accessed 19 9 2014].

[89] M. García-Valls, T. Cucinotta, C. Lu, "Challenges in Real-Time Virtualization and Predictable Cloud Computing," *Journal of Systems Architecture,* vol. 60, no. 9, pp. 726-740, 7 2014.

[90] Y. Cai, F. R. Yu, S. Bu, "Cloud Radio Access Networks (C-RAN) in Mobile Cloud Computing Systems," in *Computer Communications Workshops (INFOCOM WKSHPS), 2014 IEEE Conference*, 2014.

[91] Y. Raivio, R. Dave, "Cloud Computing in Mobile Networks – Case MVNO," Aalto University, 10 2011. [Online]. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6081085. [Accessed 21 9 2014].

[92] M. Mobile Cloud Networking, "Mobile Cloud Networking," Mobile Cloud Networking (MCN), [Online]. Available: http://www.mobile-cloud-networking.eu/site/. [Accessed 10 9 2014].

[93] D. N. Bau, "When is the time for SDN - Making SDN and NFV real,"

Juniper Networks, 2013. [Online]. Available: https://mit.ida.dk/IDAforum/u0631a/Documents/Internettets%20opbygning,%2005-11-13/Juniper%20Networks.pdf. [Accessed 17 9 2014].

[94] Nokia, "Nokia Networks shipping industry’s first commercial NFV solution," Nokia, 4 9 2014. [Online]. Available: http://nsn.com/news-events/press-room/press-releases/nokia-networks-shipping-industry-s-first-commercial-nfv-solution-networksperform. [Accessed 11 9 2014].

[95] H. Packard, "HP ProLiant DL Servers," HP (Hewlett Packard), [Online]. Available: http://www8.hp.com/us/en/products/proliant-servers/product-detail.html?oid=5249566. [Accessed 4 1 2015].

[96] Intel, "Compare Intel® Products," Intel, [Online]. Available: http://ark.intel.com/compare/77481,47645. [Accessed 18 1 2015].

[97] O. J. B. A. O. B. O. B. O. Fasiku A. I., "Comparison of Intel Single Core and Intel Dual Core Processor Performance," IJETAE (International Journal of Emerging Technology and Advanced Engineering), 2 2013. [Online]. Available: http://www.ijetae.com/files/Volume3Issue2/IJETAE_0213_113.pdf. [Accessed 10 1 2015].

[98] Wikipedia, "64-bit computing," Wikipedia, [Online]. Available: http://en.wikipedia.org/wiki/64-bit_computing. [Accessed 28 2 2015].

[99] PCPro, "HP ProLiant DL320e Gen8 v2 review," PCPro, 21 10 2013. [Online]. Available: http://www.pcpro.co.uk/reviews/servers/384550/hp-proliant-dl320e-gen8-v2. [Accessed 24 1 2015].

[100] A. I. Fasiku, J. B. Olawalee, O. B. Abiola, B. O. Ojedayo, "Comparison of Intel Single Core and Intel Dual Core Processor Performance," *International Journal of Scientific Research in Computer Science and Engineering,* vol. 01, no. 01, pp. 1-9, 2 2013.

OTHER REFERENCES

I. S. Elby by Verizon. *Software Defined Networks: A Carrier Perspective*, video presentation, 19.11.2010, cited 28.8.2014. Source: https://www.youtube.com/watch?v=xsoUexvljGk

II. J. Söderström and A. Kavanagh by Ericsson. *OpenStack: The engine of NFV*, video presentation, 13.5.2014, cited 28.8.2014. Source: https://www.youtube.com/watch?v=p89Ook4ok8A

III. S. Kindt, J. Rygaard and P. Knikkink, all by Nokia. Webinar: Tak*ing Telco cloud into action*. Presented 6.6.2014, cited 2.9.2014. Source: http://nsn.com/news-events/event/2014/taking-telco-cloud-into-action

IV. Interview of Edge Cloud director of Huawei*, Huawei Announced the Launch of Cloud Edge Solution*. Published 26.8.2014, Cited 2.9.2014. Source http://www.lightreading.com/huawei-video-resource-center/huawei-announced-the-launch-of-cloud-edge-solution/v/d-id/708038