Telco Distributed DC with Transport Protocol Enhancement for 5G Mobile Networks

A Survey

Jun Cheng & Karl-Johan Grinnemo

Faculty of Health, Science and Technology

Computer Science

Telco Distributed DC with Transport Protocol Enhancement for 5G Mobile Networks

A Survey

Jun Cheng & Karl-Johan Grinnemo

WORKING PAPER | December 2017

Telco Distributed DC with Transport Protocol Enhancement for 5G Mobile Networks - A Survey

Jun Cheng, Karl-Johan Grinnemo

WORKING PAPER | December 2017

ISBN 978-91-7063-830-5 (print)

ISBN 978-91-7063-925-8 (PDF)

Distribution:
Karlstad University
Faculty of Health, Science and Technology
Department of Mathematics and Computer Science
SE-651 88 Karlstad, Sweden
+46 54 700 10 00

print: Universitetstryckeriet, Karlstad 2017

WWW.KAU.SE

Abstract

Distributed data center hosts telco virtual network functions, mixing workloads that require data transport through transport protocols with either low end-to-end latency or large bandwidth for high throughput, e.g., from tough requirements in 5G use cases. A trend is the use relatively inexpensive, off-the-shelf switches in data center networks, where the dominated transport traffic is TCP traffic. Today's TCP protocol will not be able to meet such requirements. The transport protocol evolution is driven by transport performance (latency and throughput) and robust enhancements in data centers, which include new transport protocols and protocol extensions such as DCTCP, MPTCP and QUIC protocols and lead to intensive standardization works and contributions to 3GPP and IETF.

By implementing ECN based congestion control instead of the packet-loss based TCP AIMD congestion control algorithm, DCTCP not only solves the latency issue in TCP congestion control caused by the switch buffer bloating but also achieves an improved performance on the packet loss and throughput. The DCTCP can also co-exist with normal TCP by applying a modern coupled queue management algorithm in the switches of DC networks, which fulfills IETF L4S architecture. MPTCP is an extension to TCP, which can be implemented in DC's Fat tree architecture to improve transport throughput and shorten the latency by mitigating the bandwidth issue caused by TCP connection collision within the data center. The QUIC is a reliable and multiplexed transport protocol over UDP transport, which includes many of the latest transport improvements and innovation, which can be used to improve the transport performance on streaming media delivery.

The Clos topology is a commonly used network topology in a distributed data center. In the Clos architecture, an over-provisioned fabric cannot handle full wire-speed traffic, thus there is a need to have a mechanism to handle overload situations, e.g., by scaling out the fabric. However, this will introduce more end-to-end latency in those cases the switch buffer is bloated, and will cause transport flow congestion.

In this survey paper, DCTCP, MPTCP and QUIC are discussed as solutions for transport performance enhancement for 5G mobile networks to avoid the transport flow congestion caused by the switch buffer bloating from overloaded switch queue in data centers.

Contents

1 Introduction

The Data center (DC) infrastructure and DC networking designs with associated performance evaluation and improvement have been receiving significant interest from both academia and industry due to the growing importance of data centers in supporting and sustaining the rapidly growing Internet-based applications including search (e.g., Google and Bing), video content hosting and distribution (e.g., YouTube and Netflix), social networking (e.g., Facebook and Twitter), and large-scale computations (e.g., data mining, bioinformatics, and indexing). In the modern DC, Cloud computing by the virtualized system is applied for integration of the computing and data infrastructures to provide a scalable, agile and cost-effective approach to support the ever-growing critical Telco and IT needs, in terms of computation, storage and networks, of both enterprises and the general public [2] [3] [4].

The Ericsson Hyperscale Datacenter System (HDS) is based on a Clos architecture [1] [13] [14]. In the Clos network topology, the over-provisioned fabric cannot handle full wire-speed traffic on all ports for reason of cost, and thus needs to have a mechanism to handle the switch buffer bloating queue and the overload. (At overload, the links will be congested.) The transport flow congestion is the reason for the throughput and latency performance reduction in DC and this should be carefully avoided in the transport design and in the system architecture in DC.

1.1 Abbreviations

5G	5th generation mobile networks
5G RAN	5G Radio Network
5G CN	5G Core Network
5G NR	5G New Radio
AIMD	Additive increase/multiplicative decrease
ASs	BGP Autonomous Systems
AQM	Active Queue Management
BGP	Border Gateway Protocol
BBR	Bottleneck Bandwidth and RTT congestion control
CEE	Cloud Execution Environment
CIC	Cloud Infrastructure Controller
CNaaS	Core Network as a Service
CO	Telco Central Office
COTS	Commercial off the Shelf
CSC	Cloud SDN Controller
CSS	Cloud SDN Switch

Cubic	Cubic Congestion Control Algorithm
DC	Data Center
DC-GW	Data Center Gateway
DCTCP-bis	Second/updated version of DCTCP
DCTCP	Data Center Transmission control protocol
DCN	Data Center Network
DHCP	Dynamic Host Configuration Protocol
DPDK	Data Plane Development Kit
E2E	End to End
EBGP	Exterior Border Gateway Protocol
ECM	Ericsson Cloud Manager
ECN	Explicit Congestion Notification
ECT	ECN-Capable Transport
ECT(0)	ECN CE codepoint
ECT(1)	ECN CE codepoint
eMBB	Enhanced Mobile Broadband
eNB	E-UTRAN Node B, Evolved Node B
eNodeB	E-UTRAN Node B, Evolved Node B
ETL	Extract, Transform and Load
FEC	Forward Error Correction
Fuel	Mirantis OpenStack software life cycle management tool used to deploy CEE
GRE	Generic Routing Encapsulation
HDS	Ericsson Hyperscale Datacenter System
HoLB	Head of line Blocking
IaaS	Infrastructure as a Service
IMS	IP Multimedia Subsystem
IoT	Internet of Things
IP	Internet Protocol
IT	Information Technology
L4S	Low Latency, Low Loss, Scalable Throughput
LTE	Long Term Evolution (4G)
MC-LAG	Multi-Chassis Link Aggregation Group
Micro DC	Micro Data Center (Distributed DC)
mMTC	Massive Machine Type Communications
MPLS	Multiprotocol Label Switching
MPTCP	Multipath TCP
NFV	Network Function Virtualization
NAT	Network Address Translation
NFV	Network Functions Virtualization
NFVi	Network Functions Virtualization Infrastructure
NG-CO	Next Generation Central Office
NIC	Network Interface Controller
OTT	Over the Top, a media service for streaming content provider

OVS	Open Virtual Switch
POD	Performance Optimized Datacenter
QoS	Quality of Service
QUIC	Quick UDP Internet Connections
REST	Representational State Transfer
RTO	Retransmission Timeout
RTT	Round Trip Time
SDI	Software Defined Infrastructure
SDN	Software Defined Networking
SLA	Service Level Agreement
SNMP	Simple Network Management Protocol
SPDY	A transporting protocol for web content
SR-IOV	Single Root-IOV, Input/Output Virtualization
SSH	Secure Shell
TCP	Transmission Control Protocol
ToR	Top of Rack switch
UE	User Equipment
URLLC	Ultra-Reliable and Low Latency Communications
VLAN	Virtual Local Area Network
VxLAN	Virtual eXtensible Local Area Network
VID	VLAN Identifier
vAPP	Virtual Application
VIP	Virtual IP
vCIC	Virtual Cloud Infrastructure Controller
VIM	Virtual Infrastructure Manager
VM	Virtual Machine
VNF	Virtual Network Function
vNIC	Virtual Network Interface Card
VPN	Virtual Private Network
vPOD	Virtual Performance Optimized Datacenter
vPort	Virtual Port
VTEP	VxLAN Tunnel Endpoint
VR	Virtual Router
VRF	Virtual Routing and Forwarding
VRRP	Virtual Router Redundancy Protocol
vSwitch	Virtual Switch
WAN	Wide Area Network

2 Overview of Data Center Architectures

The majority of data centers use Ethernet switches to interconnect the servers, there are still many different ways to implement the interconnections, leading to different data center network topologies. Each of these different topologies is characterized by different resource requirements, aiming to bring enhancements to the performance of data centers. In the following sections, some representative topologies used in data centers will be discussed. If servers and switches are regarded as boxes, wires as lines, the topology of every data center network can be represented as a graph.

2.1 Data Center Network Topologies

All data center topologies can be classified into two categories: fixed and flexible. That is, if the network topology cannot be modified after the network is deployed, we classify it as a fixed architecture; otherwise as a flexible architecture.

For the fixed topology, standard tree based architectures and the recursive variants are widely used in designing data center networks [2] [3]. The standard tree based architectures includes Basic tree, Fat tree and Clos network architectures as shown in Figure 1.

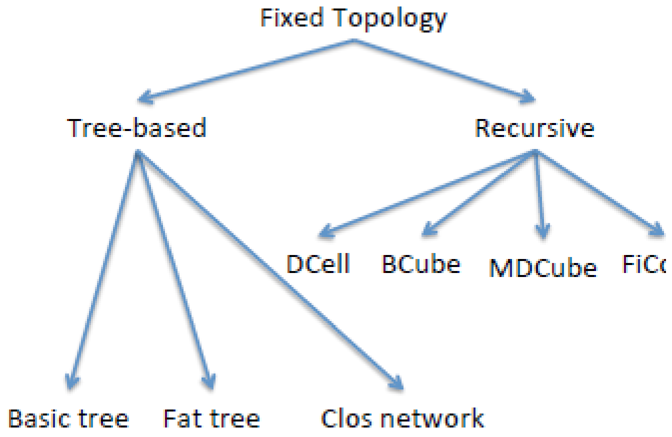


Figure 1. The standard tree-based architectures in fixed DC topology.

2.2 Basic Tree

Basic tree topologies consist of either two or three levels of switches/routers, with the servers as leaves. In a basic tree topology, the higher-tier switches need to support data communication among a large number of servers. Thus, the switches with higher performance and reliability are required in these tiers. The number of servers in a tree architecture is limited by the numbers of ports on the switches. Figure 2 demonstrates a 3-level Basic tree topology [4].

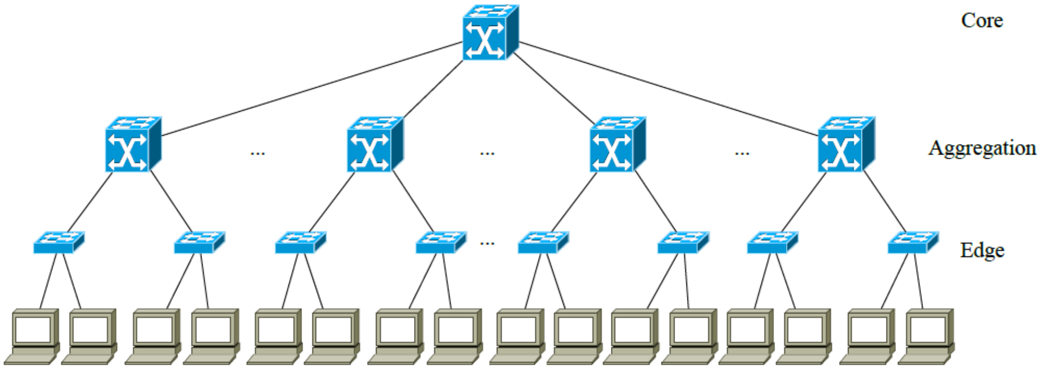


Figure 2. A 3-level Basic tree topology.

2.3 Clos Network Topology

More than 50 years ago, Charles Clos proposed the non-blocking Clos network topology for telephone circuits [9]. The Clos topology deliver a high bandwidth, and many of the commercial data center networks adopt a special instance of the Clos topology called Fat tree [2]. The Clos network topology is shown in Figure 3.

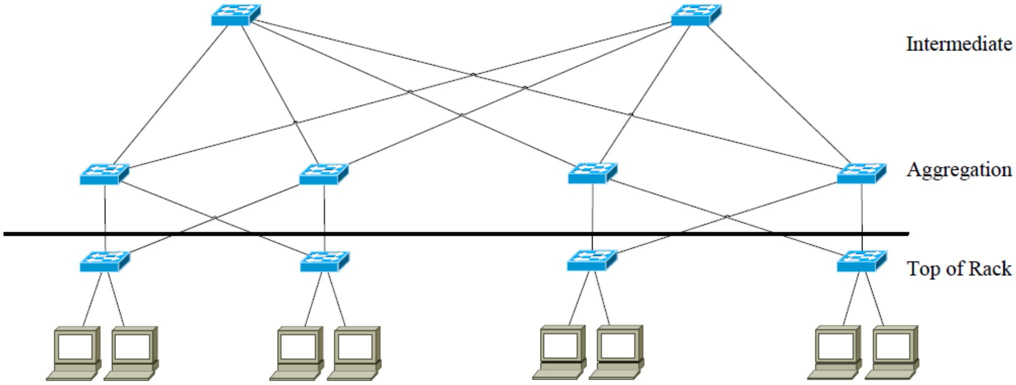


Figure 3. Clos Network Topology [2].

The Clos Network topology has a multi-rooted tree architecture. When deployed in data center networks, a Clos Network usually consists of three levels of switches: Top-of-Rack (ToR) switches, which are directly connected to servers; the aggregation switches connected to the ToR switches; and the intermediate switches connected to the aggregation switches. These three levels of switches are termed “input”, “middle”, and “output” switches in the original Clos terminology [9].

2.4 Google Fat Tree

Many of the commercial data center networks have adopted a special instance of Clos topologies called Fat tree. Fat tree is an extended version of a tree topology [2]. Google 3-level Fat tree topology shows in Figure 4 [11].

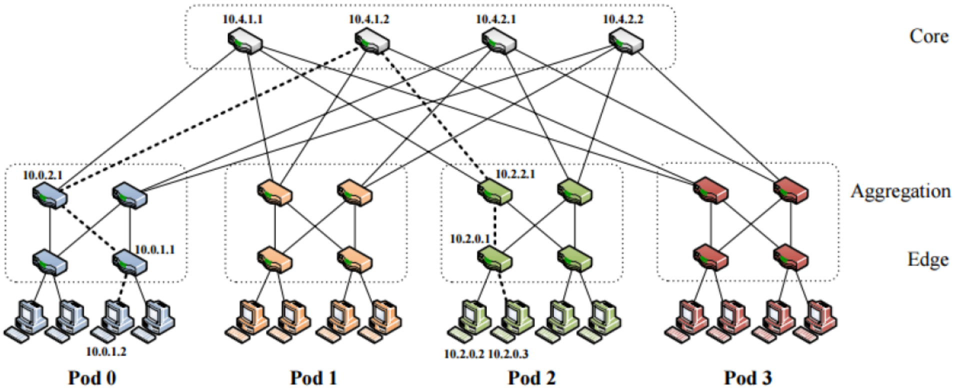


Figure 4. Google 3-level Fat Tree Topology.

In Google 3-level Fat tree topology, unlike in the Basic tree topology, all the 3-levels use the same kind of off-the-shelf switches, and the high-performance switches are not necessary in the aggregate and core levels [2].

2.5 Facebook Fat Tree

In order to achieve high bisection bandwidth, rapid network deployment and performance scalability to keep up with the agile nature of applications running in the data centers [12], Facebook deployed a version of the Fat tree topology. The Facebook Fat tree topology is shown in Figure 5.

In the Facebook Fat tree, the work load is performed at server Performance Optimized Datacenters (PODs) and a standard unit of the network as shown in Figure 5. The uplink bandwidth of each ToR is four times (4*40G = 16*10G) the downlink bandwidth for each server connected to it. To implement building-wide connectivity, it created four independent ”planes” of spine switches [Tier 3 switch], each scalable up to 48 independent devices within a plane [13].

Border Gateway Protocol (BGP4) is used as a control protocol for routing, whereas a centralized controller is deployed to be able to override any routing paths whenever required, taking a “distributed control, centralized override” approach. In order to use all the available paths between two hosts, ECMP (Equal Cost Multiple Path) routing with flow based hashing is implemented [3].

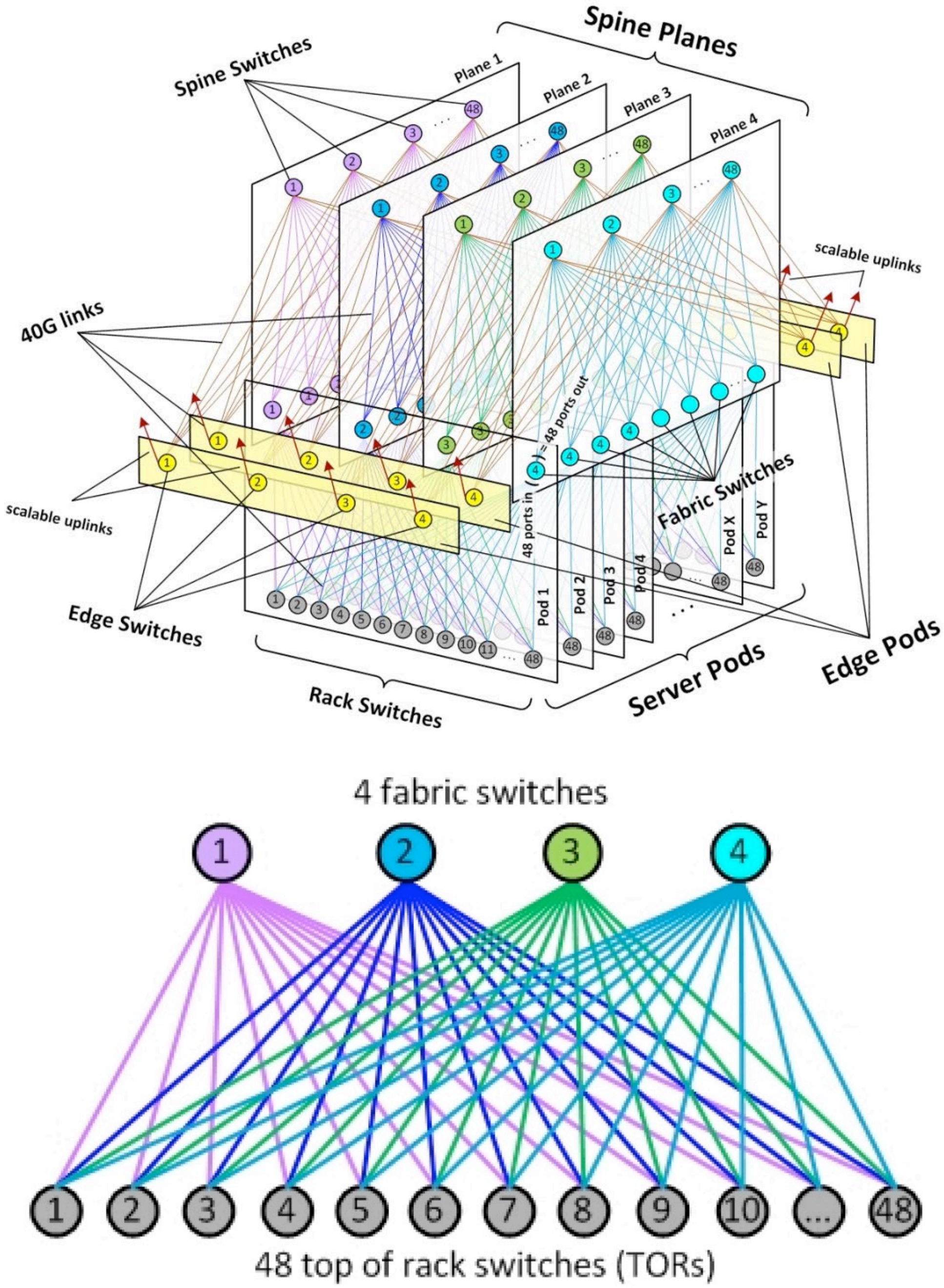


Figure 5. Facebook Fat Tree Topology.

2.6 DC Architecture for 5G Mobile Networks

The DC for 5G mobile networks will continue using the common DC architecture with the tree-based fixed topology e.g., Fat tree and/or Clos networks. From a system-architecture point of view, there will be no special DC architecture for distributed/Micro DC on Telco Cloud compared with the Central DC. That is, Telco VNFs/applications on distributed/Micro DC will continue using the general DC architecture on the topology, however, the relevant construction and implementation details from different telecom data center suppliers will be adapted to their own specific hardware and software environments for the most effective use of their resources (HW, SW and networking) and for achieving a better performance in their specific distributed/Micro DC environment.

2.6.1 Ericsson's Distributed/Micro DC Architecture

The Ericsson HDS 8000 is based on the Intel Rack Scale Design [35]. It is a disaggregated hardware architecture. Its software-defined infrastructure and optical backplane are designed to manage the data center resources of telecom operators, service providers, and enterprises. They provide the compute power, storage capacity, and networking capabilities necessary for hosting a hyperscale cloud.

The Ericsson distributed/Micro DC at the cloud edge is shown in Figure 16. Some quality of service requirements, e.g., low end-to-end latency in 5G use cases, can be difficult to fulfill in the central data center deployment. These critical cases can only be achieved by NFV(s) located at distributed/Micro DC by using optimized transport and communication options in an active network slicing.

Ericsson's distributed/Micro DC is based on HDS 8000 [14]. Ericsson and Intel cooperated to establish and bring to market the Ericsson HDS 8000 as a best-in-class Hyperscale solution based on Intel RSA principles, and to drive adoption of RSA as a standard for all datacenter solutions.

In the HDS 8000, There are built-in fiber optic interfaces on the system down to individual servers for the optimal communications. With these fiber optic configurations, it is possible to disassemble the system down to the components and let them communicate with each other by means of light. With such a solution, special hardware for a particular purpose becomes superfluous [30]. Therefore, nothing is needed to work together in one and the same box. The CPUs, memories and hard drives become individual components, which are connected with a bus within a single optical network.

The transport networks in distributed/Micro DC could also be implemented in optics, especially when the DC is completely optically designed. In the long term, the full VNFs/virtual applications, or even entire systems can be moved to and run in the cloud by virtualizing everything in a completely optical networked environment in a distributed/micro DC. The optical network, system virtualization and the data transport/flow is the building blocks in Ericsson's DC Architecture for 5G Mobile Networks.

Ericsson's data center architecture follows Ericsson's cloud strategy, and is based on the HDS 8000 product with fiber optic interfaces, i.e., the product which could best use the Ericsson's cloud, data center and networking resources to achieve better performance in the distributed/micro and centralized DC environment [30].

3 Data Center Networking

The Data Center (DC) is a pool of computational, storage, and network resources interconnected via a communication network. Datacenter networking is often divided into overlay and underlay networking. The underlay networking is the physical network constituting the fabric, the Data Center Gateway (DC-GW) and server interconnect, while the overlay is the application networking as shown in Figure 6.

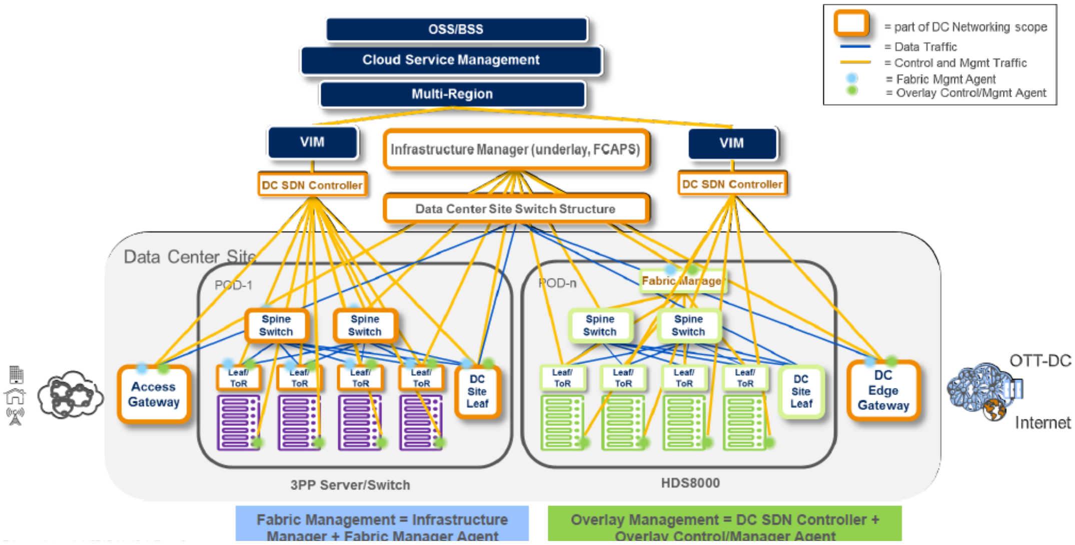


Figure 6. A demonstration of Data Center Networking [13].

The Performance Optimized Data Center (POD) defines a set of compute, network, and storage hardware resources with internal networking between the servers and internal connectivity from the servers to the storage hardware.

3.1 Data Center Network

Since the Data Center Network (DCN) interconnects the compute resources, it is a key resource in a data center. A DCN needs to be scalable, flexible and efficient to be able to connect hundreds or thousands of servers in order to meet the growing demands of cloud computing.

3.1.1 Clos Network

A Clos network is a kind of multistage circuit switching network and many high-capacity datacenter networks are built using switches in a multi-layer Clos architecture. The Clos network uses simple switch-routers and combine them in layers to build a flexible and extensible fabric.

The Clos network design used in the Ericsson HDS 8000 datacenter is shown in Figure 7 (upper graph) and the extended L2 underlay on Clos fabric (down graph): leaf switches connect to racks of servers and storage systems, and spine switches are used to interconnect racks. This leads to a design which is scalable to tens of thousands of servers.

The high-performance switches are not necessary to be used in the leaf and spine layers, where off-the-shelf switches can be used.

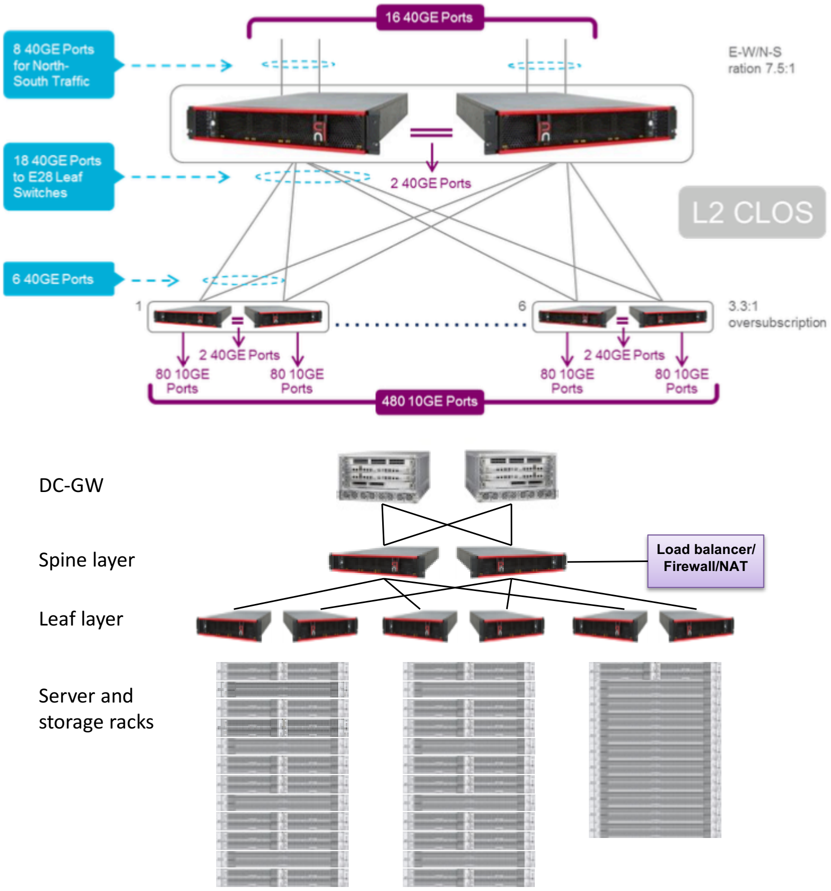


Figure 7. The Clos network design in Ericsson HDS datacenter (upper graph) and the extended L2 underlay on a Clos fabric (down graph).

The HDS 8000 concept of a Virtual POD (vPOD) enables a logical partitioning of physical hardware components providing an optimal combination of compute, storage hardware, and network resources to fulfill a specific customer's needs.

3.1.2 Spine

The idea with a Clos spine layer is to use simple switch routers and combine them in layers to build flexible and extensible fabrics; especially, if one uses layer 3, ECMP (Equal Cost Multi-Path) and BGP [14], the spine layer can in principle be scaled up to any size with full active-active redundancy.

The spine layer handles both east-west and north-south traffic, and may handle both data and storage traffic. Therefore, dimensioning is important. Typically, a spine layer has some degree of over-provisioning for reasons of cost. This means that it cannot handle full wire-speed of all leaf ports, e.g., full server network capacity. Fortunately, a well-designed spine-layer can be extended with capacity so that it can become non-blocking just by adding more switches.

Since an over-provisioned fabric cannot handle full wire-speed traffic, it needs to have a mechanism to handle overload. At overload, links are congested and buffers overflow. When buffers overflow, packets are dropped or a flow control mechanism needs to be implemented that pushes back origin traffic.

Quality of service mechanisms are also essential for handling overflow situations (which flows to drop) as well as ensuring end-to-end quality assurances. For example, storage traffic is typically highly sensitivity to latency.

3.1.3 Leaf/ToR

A leaf provides access to equipment which needs the fabric for interconnect. The leaf layer therefore needs to be scalable in terms of number of ports, such as 10GE, or nowadays moving up to 25GE or even 100GE.

A datacenter also typically provides additional services, both virtual and physical, that can be used by tenants for flexibility, performance, and optimization. This includes routing, load-balancing, NAT, firewalling, etc. Appliances can be used for these purposes, but it is common to use virtual services that run on regular compute servers [14].

Hyper-converged datacenters (vPOD concept) mix computing and storage resources so that they can be flexibly configured into very large multi-purpose data-centers [14].

3.1.4 Datacenter Gateway

A Datacenter gateway (DC-GW) provides the external connectivity for a datacenter. Depending on the size of the DC and the fabrics, a DC-GW can serve a single fabric or many fabrics. In very small cases the DC-GW can even be embedded in software or be provided by a layer-3 capability in the spine layer itself.

The cost of the DC-GW is often significantly higher than the fabric switches and the dimensioning of north-south traffic and port densities are therefore important to design correctly.

In case of multiple fabrics, an aggregation network is necessary to interconnect the fabrics with the DC-GW. This also handles inter-fabric traffic and further off-loads the DC-GW. An aggregation network can also inter-connect common services and resources [14].

3.2 Data Center Networking

Datacenter networking is divided into overlay and underlay networking. The data center network may be sliced into separate logical data network partitions. A logical network partition in conjunction with connected computer systems and storage hardware elements is referred to as a vPOD.

An overlay network that is built on top of another network (underlay), using some form of encapsulation, or indirection, to decouple the overlay network service from the underlying infrastructure. An overlay provides service termination and encapsulation, and is orchestrated by the Virtual Infrastructure Manager. Underlay is about networking hardware and topology, and connects overlay and DC-GW.

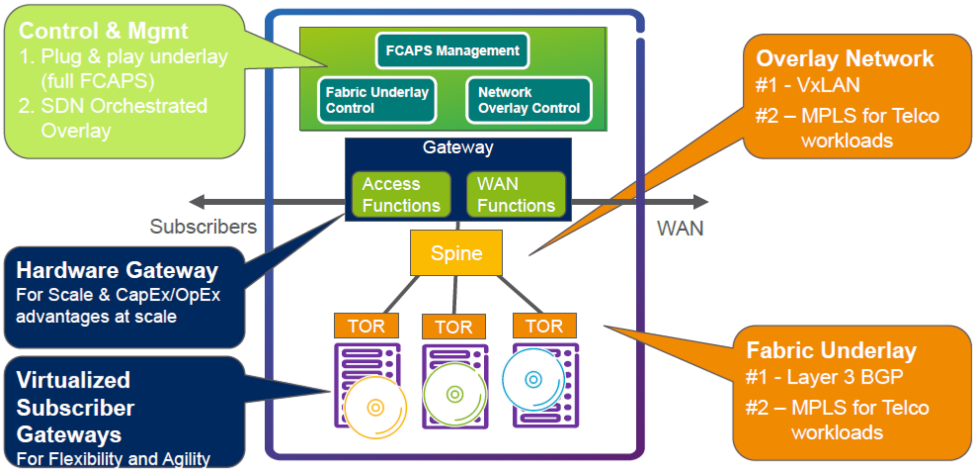

Control & Mgmt
1. Plug & play underlay (full FCAPS)
2. SDN Orchestrated Overlay
FCAPS Management
Fabric Underlay Control
Network Overlay Control
Gateway
Access Functions
WAN Functions
Subscribers
WAN
Spine
TOR
TOR
TOR
Hardware Gateway
For Scale & CapEx/OpEx advantages at scale
Virtualized Subscriber Gateways
For Flexibility and Agility
Overlay Network
#1 - VxLAN
#2 – MPLS for Telco workloads
Fabric Underlay
#1 - Layer 3 BGP
#2 – MPLS for Telco workloads

Figure 8. Layer 3 Underlay with layer 2 VxLAN Overlay.

This overlay and underlay network division may not always exist or be well-defined, DC networking often deals with tunneling techniques to carry the overlay virtual network infrastructure over the underlying physical underlay.

Figure 8 shows an example of a layer 3 underlay (the fabric routes) with a layer 2 VxLAN overlay where VxLAN tunnel endpoints (VTEPs) forms an overlay between the Open virtual switches (OVS) on the server blades (and the DC-GW) so that direct layer 2 communication is possible between the VMs or containers. Traffic isolation is achieved with virtual network identifiers, see Figure 9 for details.

3.2.1 *Underlay*

The classical underlay is the network that controls the fabric. Its access is the leaves and its interior is the spines. However, underlay networking may also be extended into the hosts (e.g., VxLAN VTEP in OVS), and into the DC-GW (e.g., GRE tunnel endpoint in a VRF).

3.2.1.1 *Layer 2*

A strict layer 2 underlay is a bridged network with a single broadcast domain where each leaf and spine switches layer 2 frames and uses VLANs for traffic isolation. A layer 2 underlay is simple to deploy and extend.

In redundant fabrics, where leafs and spines are duplicated for load-balancing and redundancy, fabric switches are clustered and used a Multi-Chassis Link Aggregation Group (MC-LAG) with active-active load-balancing to utilize the bandwidth as much as possible. This means that traffic is evenly distributed among the links of the fabric.

A drawback with a layer 2 fabric is its scaling limitations. Broadcast traffic scales linearly with the number of origins due to ARP flooding, unknown unicast, and broadcast which means that broadcast traffic takes a larger part of the overall traffic for large fabrics. Further, the VLAN space is limited to 4096 which makes traffic isolation difficult with a larger number of deployments. Additionally, it is difficult to scale out MC-LAG to multiple spine layers which means that the flexibility in the spine layer is reduced.

3.2.1.2 Layer 3

In a layer 3 fabric, the leaf and spine switches perform layer 3 forwarding and routing. Path-finding is made by a dynamic routing protocol (or static routing for small fabrics). Many large fabrics run BGP using the well-known and proven scalable traffic policy mechanisms available in EBGP. In such a fabric, private ASs are used to within the fabric and multi-path BGP for load-balancing. Internal routing protocols like OSPF can also be used.

3.2.2 Overlay

While this division may not always exist, or be well-defined, the DC networking therefore often deals with tunneling techniques to carry the overlay virtual network infrastructure over the underlying physical underlay.

3.2.2.1 Layer 2

Many cloud environments are based on layer 2 techniques, most commonly, by single, one-hop IP networks. This includes most VMware, Openstack and container installations where the overlay networking is being made in the host.

Tenant isolation is then often implemented using VLANs and bridging, typically Open Virtual Switch (OVS). This means that different tenants may use the same bridge, but uses different switching domains within the bridge to maintain isolation.

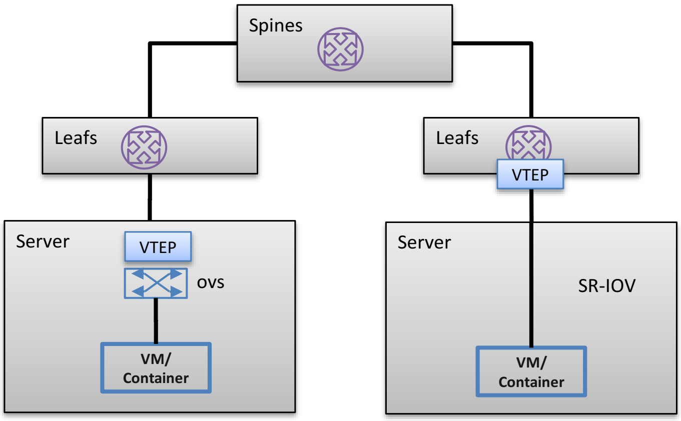


Figure 9. SR-IOV and hardware VTEP

A large effort has been made in increasing the performance of layer 2 overlay in the hosts. PCI bypass and Single Root-IOV, Input/Output Virtualization (SR-IOV) are techniques that bypass the regular device and protocol processing in the kernel so that link-level packets may be processed directly in user space [14]. Combined with Data Plane Development Kit (DPDK) and a user-space IP stack this means that high performance applications can run in VMs bypassing all bottlenecks in the Linux kernel or hypervisor layers.

Figure 9 shows an example of an SR-IOV bypass where a virtual function of the interface is mapped directly to the VM. In this example, a hardware VxLAN Tunnel Endpoint (VTEP) is placed in the leaf switch thus moving the overlay/underlay to within the fabric. The hardware VTEP is necessary since the SR-IOV bypasses the operating system functions that normally would host the OVS and VxLAN encapsulation.

The downside with SR-IOV and such techniques is the loss of generality and cardinality: The functionality normally provided by system functions need now be implemented by the application itself, and there are limitations based on the physical implementation.

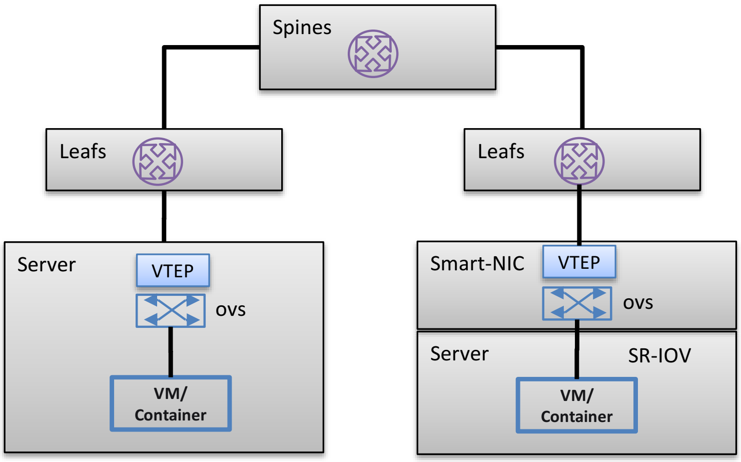


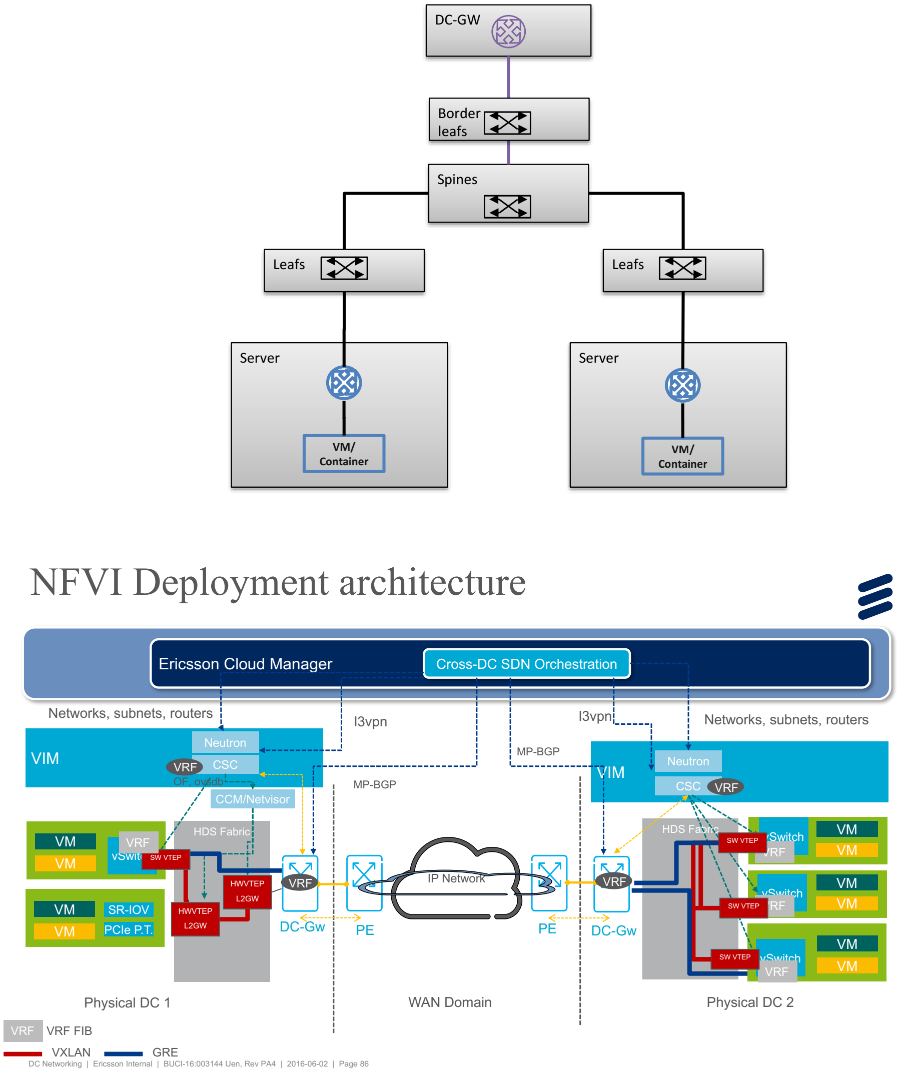
Figure 10. Smart-NIC in combination with SR-IOV, OVS and VXLAN.

In order to mitigate the SR-IOV downside, a NIC functionality need to be added in a smart-NIC. Figure 10 shows an example when an OVS and VxLAN VTEP have been added in the fast-path of an off-loaded NIC functionality. In this way, the full speed of SR-IOV is combined with the flexibility of an OVS.

3.2.2.2 Layer 3

While most virtualization techniques are based on layer 2, some emerging virtualizations are using a routed layer 3 approach. One example of this is project Calico [40] that uses routing in the kernel instead of OVS, as well as Ericsson’s VPN approach using distributed L3 overlay forwarding in vSwitch shown in Figure 11.

Tenant isolation is made with policies on a flat layer 3 network. Calico can be combined with many other virtualization techniques and seems to catch on especially well in container networking scenarios.



NFVI Deployment architecture

DC Networking | Ericsson Internal | BUCI-16:003144 Uen, Rev PA4 | 2016-06-02 | Page 86

Figure 11. Layer 3 overlay over layer 2 underlay (upper) and Ericsson's VPN approach using distributed L3 overlay forwarding in vSwitch (down).

Figure 11 shows an example of how Calico uses routing in the operating system kernel over a layer 2 fabric. The routers use BGP to exchange routes local to the server (upper), and Ericsson's own VPN approach using distributed L3 overlay forwarding in vSwitch (down).

3.2.3 Tunnels

Tunneling of overlay over underlay can be made in many ways. One common technique is VxLAN, but it is also possible to use GRE, MPLS, QinQ or any other tunneling technique depending on technology.

4 Traffic Engineering and Transport Protocols in Data Center

DC traffic includes traffic between a data center and the outside of Internet, and the traffic inside of a data center.

The use of relatively inexpensive, off-the-shelf switches is a trend in data center networks and about 99.91% of transport traffic is TCP traffic in IT DC [4] and about 15% of transport traffic is UDP traffic in VNF based distributed telco DC [38]. In such a hardware and traffic environment, there have been several problems that need to be solved in order to gear up the transport performance on the throughput and latency, which are specially required on VNF(s) for distributed Cloud edge deployment for 5G applications and IoT usages. Using the DCTCP in DC is a solution for performance improvement and the robustness in DC [2] to meet specially the latency requirement in these critical use cases.

4.1 Data Center Traffic Engineering

When exploring IT DC traffic, TCP bit flows that cover 99.91% of the traffic [4] are used to describe the data traffic. The TCP flow is a sequence of packets that share common values in a subset of fields of their head, e.g., the 5-tuple (source; source port; destination; destination port; protocol). The existence of a flow usually implies that these packets are sequentially and logically related. By allocating packets into flows, routing protocols can achieve load-balancing by distributing flows into different parallel paths between two nodes, while avoiding packet reordering problem by limiting a flow into only one path during a given time.

Query processing is the most important for data center applications, which need to analyze massive data sets, e.g., web search. The data center consists of a set of commodity servers supporting map reduce jobs and a distributed replicated block store layer for persistent storage for ETL (Extract, Transform and Load) via query processing at different stages, which pull that data out of the source systems and placing it into a data warehouse.

4.1.1 Traffic Pattern

Two patterns, the so-called "work-seeds-bandwidth" and "scatter-gather" patterns comprise most of the traffic in the data center [2]. The "work-seeds-bandwidth" pattern shows that a large chunk of data traffic is among servers within a rack. In fact, using more detailed data, the researchers showed that the probability of exchanging no data is 89% for server pairs inside the same rack, and 99.5% for server pairs added from different racks. The "scatter-gather" pattern shows the traffic resulted from the map-reduce applications. A server either talks to almost every server or to less than 25% servers within the same rack, i.e., scatter (Extract). Further, a server either does not talk to any server outside its rack, or talks to about 1-10% of servers outside its rack i.e. gather (Transform and Load).

4.1.2 Traffic Characteristics

The traffic characteristics described above decide how to design network protocols for data centers. For example, the fact that most data exchange happens inside racks requires more bandwidth between servers in the same rack. Congestion is responsible for a large chunk of performance reduction, and the phases that cause most congestions should be carefully designed. Furthermore, since most data are transferred in short flows, it is not sufficient to schedule long flows only.

4.1.3 Congestion

Data centers often experiences congestion [2]. In fact, 86% of the links in a data center experience congestion lasting for more than 10 seconds, and 15% of data center links experience congestion lasting for more than 100 seconds. Moreover, most congestion events are short-lived, i.e., when the links are congested, more than 90% congestion events last for less than 10 seconds, while the longest congestion events lasted 382 seconds [2]. There is high correlation between congestion and read failures. In the map-reduce scenario, the reduce phase is responsible for a fair amount of the network traffic. The extract phase, which parses data blocks also contributed a fair amount of the flows on high utilization links.

4.1.4 Flow Characteristics

In a data center, there are 80% of flows that last for less than 10 seconds, and about less than 0.1% last for longer than 200 seconds. More than half of the total bytes belong to flows that last less than 25 seconds [2].

4.1.5 UDP transport traffic in Telco/Micro DC

Although TCP dominates data center transport (99.91% [4]), there is still a portion of UDP traffic in the data center as well; especially in the cases where the SDN based NFVi on Telco/Micro DC is applied. There are two reasons for such UDP traffic in Telco/Micro DC. Firstly, the telco VNF(s) originates from the legacy system where the transport traffic mix, including UDP, already exists. Secondly for some of the media and the real-time services, e.g., OTT and IMS, the transport efficiency of the real-time traffic is more important than the reliability: To some degree, media packet loss can be tolerated. To this end, in Ericsson's data center, the UDP based transport is still often used, and even non-IP protocols appear that inherited also from the telco NFV services.

The transport traffic mix on SDN based NFVi on Telco DC is described in [38], where the traffic mix depends on the packet size, and the UDP traffic is about 15% in an averaged traffic mix.

4.1.6 Storage traffic

In data center networks, the storage network domain also utilizes the data fabric. When the storage racks are used as an extended L2 underlay in a Clos fabric as shown in Figure 7, the storage networks are implemented as native VLANs in the data fabric, and the packet forwarding can be used on storage data as same as data transport. Therefore, the performance improvements on the throughput, latency and robustness by implementing of ECN and AQM in DC's data plane will also be beneficial for the storage traffic in the data center networks.

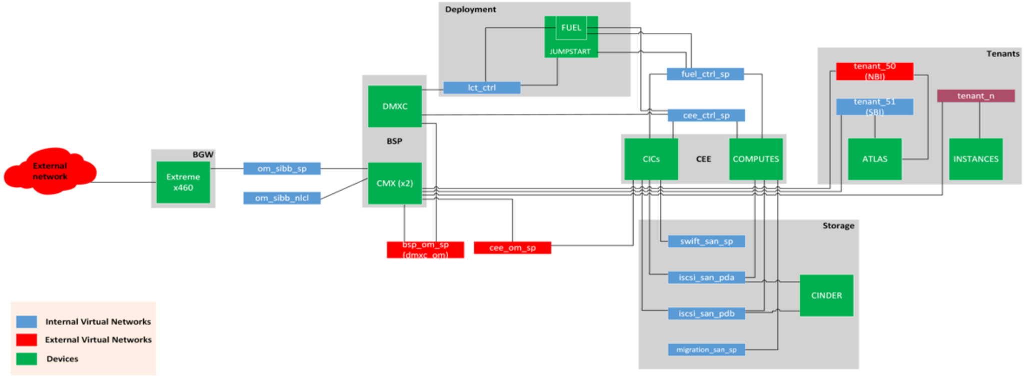
The storage traffic is typically highly sensitive to latency in the data center. Therefore, QoS mechanisms are essential for handling overflow situations in order to decide which flows can be dropped and to ensure the end-to-end quality assurances.

In Ericsson's Cloud Execution Environment (CEE) storage networks, OpenStack Block Storage (Cinder) and OpenStack Object Storage (Swift) are used as persistent storage (see Figure 12). The Block Storage provides persistent block storage to guest Virtual Machines, which can be configured to utilize the block storage as a distributed or centralized storage solution to realize persistent block storage volumes that can be made available to VMs running on compute nodes. The Block Storage is appropriate for performance sensitive scenarios such as database storage, expandable file systems, or providing a server with access to raw block level storage.

The Object Storage (Swift) allows the storage and retrieval of objects. CEE uses object Storage internally. The OpenStack Swift is a scalable redundant storage system, where objects and files are written to multiple disk drives, spread throughout servers in the data center.

In Ericsson's CEE storage networks, Ephemeral Storage controlled by the OpenStack Nova compute controller can be used as local ephemeral storage for non-persistent storage.

In Ericsson's OpenStack Block Storage (Cinder), a ScaleIO storage system can be used on a dedicated vPOD or shared with all other vPODs belonging to the POD. The transport is based on iSCSI multi-homing and path diversity. In the storage traffic, IP MTU supports up to 9216 bytes, however, the NFVi supports an IP MTU up to 2000 bytes only in data traffic domain.



Storage

VLAN Name	VLAN ID	Subnet	IP Address	Allocation	Logical Connectivity Description
iscsi_san_pda(iscsi-left)	4011	192.168.11.0/25	192.168.11.1	Storage node A	Cinder (iSCSI)
			192.168.11.2	Storage node B	
			192.168.11.20 - 192.168.11.126	Dynamic	
iscsi_san_pdb(iscsi-right)	4012	192.168.12.0/25	192.168.12.1	Storage node A	Cinder (iSCSI)
			192.168.12.2	Storage node B	
			192.168.12.20 - 192.168.12.126	Dynamic	
swift_san_sp	4013	192.168.13.0/29	192.168.13.1 - 192.168.13.6	Dynamic	Swift

Figure 12. Ericsson's CEE storage networks, OpenStack Block Storage (Cinder) and OpenStack Object Storage (Swift) are used as persistent storage, and the storage networks are implemented as native VLANs.

Ericsson HDS 8000 is a platform for the next-generation datacenter, which was designed from the start for storage solutions based on software-defined storage (SDS). The HDS 8000 enables a flexible combination of compute and storage sleds via the optical backplane and storage pooling. The optical backplane provides the flexibility to connect compute sleds to storage pools via Serial Attached SCSI (SAS) with a bandwidth of up to 8 x 12G per compute sled. This enables optimized configurations for virtually any storage-intense application and SDS solution.

4.2 Data Center Transport Protocols

One trend in data center networks is the use of relatively inexpensive, off-the-shelf switches [2]. Furthermore, 99.91% of traffic in IT DC is TCP traffic [4] and about 15% of transport traffic is UDP traffic in VNF based distributed telco DC [38]. In such a hardware and transport traffic environment, there have been several issues that need to be solved at the transport-stack level to meet the performance (throughput and latency) and robustness requirements from applications on DC.

Although many data center networks use the standard transport protocol, TCP, there are other transport protocol proposals specifically designed for special features on data center networks in order to improve and mitigate the network characteristics and/or for optimizing the performance and robustness on the traffic flows in DC networks. In a Micro DC, where the telco VNF(s) on the distributed edge Cloud for 5G and IoT applications are involved, other transport protocols may be used as well for specific applications, e.g., DCTCP. Using the DCTCP [2] [6] [8] and an equivalent ECN-enabled SCTP in DC are the solutions for performance and robustness improvement in DC.

The map-reduce design pattern (ETL) is probably the most popular distributed computing model [2]. This model provides several requirements for network performance:

- Low latency.
- Good tolerance for burst traffic.
- High throughput for long-time connections.

To search for business by browser, e.g., the first two requirements will contribute to user experience; the last bullet comes from the main worker nodes at the lowest level to be regularly updated to refresh the aged data. So that the "high throughput" requirements will contribute the quality of search results that cannot be ignored. We cannot determine in advance which of these three factors are the more important ones. But the reality is that it is not possible to optimize for lower delay and higher throughput at the same time. By using the concurrency transport protocols, the optimization of the throughput may lead to a good result. However, the optimization delay tends to leak of good solution.

4.2.1 Transport protocol and characteristics

The authors in [4] investigated 6000 servers in over 150 racks in the scaled IT DC, sampled more than 150 terabytes of compressed data, and summarized the traffic transport protocols used as well as the traffic characteristics as follows:

- TCP contributes with 99.91% of the traffic.

- In DC, long-lived flow (large traffic) and short-lived flow (small traffic) coexist. From the flow point of view, small flow accounted for the vast majority of the traffic. But the large flow contributed to the vast majority of the number of bytes of the traffic. The traffic consists of query traffic (2KB to 20KB in size), delay-sensitive short messages (100KB to 1MB), and throughput-sensitive long flows (1MB to 100MB).

- The higher the concurrency of flow, the smaller the flow size. For example, when all flows are considered in [4], the median number of concurrent flow of 36, which results from the smaller flows contributed by the breadth of the partition/aggregate traffic pattern. If only flows of more than 1MB are considered, the median number of concurrent large flows is only one flow. These large flows are big enough that they last several RTTs, and can consume significant buffer space and causing queue buildup.

Therefore, for the TCP traffic, the throughput-sensitive large flows, delay sensitive short flows and burst query traffic, co-exist in a data center network.

4.2.2 TCP Performance in a DC

Impacts on TCP performance in DC can be summarized as the following three categories [4]:

- Incast:

  Refers to such a phenomenon: one client to send a request to the N servers at the same time, the client must wait for the arrival of all the N server responses before continuing with the next action. When N servers send the responses to the client at the same time, the simultaneous multiple "responses" cause the switch buffer to build up and the consequence is the buffer overflow and packet loss. So that the server will wait for TCP timer based retransmission to allow the client to continue. This incast phenomenon will impede both the high throughput and the low latency. The current study on incast shows that reducing the TCP RTO will mitigate the problem, but this does not solve the root cause of the problems.

- Queue buildup:

  Due to the "aggressive" nature of TCP, a large amount of TCP packets in the network traffic may cause a large amount of oscillation in the length of the switch queue. When the switch queue length is increased, there will be two side effects: small flow packet loss by incast, small flow delay in the queue will take a longer time (between 1ms vs 12ms in 1Gb network).

- Buffer pressure:

  Because many of the switches on the cache is shared between the ports. Therefore, a short flow on one port can easily be affected by the large flow on other ports because of the lack of caching.

4.3 DC Traffic Engineering for 5G Mobile Networks

There is some focus on DC traffic engineering for 5G mobile networks. The latency requirement from 5G covering the radio and transport networks is more tough to fulfil, which has a transport bottleneck that relays on the latency performance in the flow control of the transport protocol. Further shorten the end-to-end latency needs to push Micro DC to the cloud edge in Next Generation Central Office (NG-CO) for NFV deployment that performs the virtualized telco network functions. Therefore, to push Micro DC to the cloud edge is a reasonable and effective solution for the end-to-end latency reduction for the 5G applications and services.

The low latency for short flows is what is exactly required from 5G networks in the 5G use cases. For further end-to-end latency reduction, using ECN enabled DCTCP instead of normal TCP congestion control algorithm will provide shorter end-to-end latency for short flows as well as a high burst tolerance. Therefore, using ECN enabled DCTCP as the transport protocol in the 5G network will benefit the 5G RAN, 5G CN in 5G transport networks by its lower latency characteristics.

5 Data Center Transport Protocol Evolution and Performance Optimization

5.1 Data Center TCP (DCTCP)

Alizadeh et al. [4] proposed the DCTCP congestion control scheme, which is a TCP-like protocol designed for data centers. DCTCP uses Explicit Congestion Notification (ECN) as the input for the flow congestion control with lower latency performance advantage compared to the packet loss based congestion control in the normal TCP. The goal of DCTCP is to achieve high burst tolerance, low latency and high throughput performances with commodity shallow-buffered switches. DCTCP achieves these goals by reacting to congestion in proportion to the extent of congestion directly. It uses a marking scheme at switches that sets the Congestion Experienced (CE) code point of packets as soon as the buffer occupancy (instantaneous queue size) exceeds a fixed small threshold. Then, the source node reacts by reducing the window by a factor which depends on the fraction of marked packets: the larger the fraction, the bigger the decrease factor. The DCTCP algorithm consists of three main components.

5.1.1 Marking at the Switch

ECN uses the two least significant (rightmost) bits of the DiffServ field in the IPv4 or IPv6 header to encode four different code points:

00 – Non-ECT

10 – ECT(0), One ECT code point is needed only

01 – ECT(1), Sender receives an ECN-Echo ACK packet for adjust the congestion window)

11 – CE

When both endpoints support ECN they mark their packets with ECT(0) or ECT(1). It may change the code point to CE instead of dropping the packet, which depends on a parameter i.e. the marking threshold K. A packet is marked with the CE code point if the queue occupancy is larger than K on its arrival.

5.1.2 *Echo at the Receiver*

The DCTCP receiver will convey back the CE code point in the ACK packet header. When the received CE code point mark appeared, or disappeared the ACK will immediately be sent back to the sender. Otherwise, delayed ACKing is used, i.e., the CE code point in the cumulative ACK for every *m* consecutively received packets is set in order to reduce the load in the data center. The DCTCP sender side can also evaluate the degree of the congestion by the range of the TSN numbers on the congested packets.

5.1.3 *Controller at the Sender*

The sender maintains an estimate of the fraction of packets that are marked, which is updated once for every window of data. After estimating this fraction, the sender decides whether the network is experiencing a congestion event, and changes the window size accordingly. This is in contrast to standard TCP where congestion events detected with triple-duplicate ACKs lead to a halving of the congestion window.

5.2 MPTCP

5.2.1 *Multipath TCP (MPTCP)*

Regular TCP restricts communication to a single path per connection, where a path is a source/destination address pair. MPTCP is an extension to TCP that adds the capability of simultaneously using multiple paths per single connection, which can be used to connect the Internet via different interfaces and/or ports simultaneously [32].

5.2.2 *Main benefit from MPTCP*

By allowing a single connection to use multiple interfaces simultaneously, MPTCP will distribute load over available interfaces, which will either increase bandwidth or throughput, or allow handover/failover from one interface to another.

MPTCP is backwards compatible, i.e., it falls back to TCP if the communicating peer does not support MPTCP. MPTCP is also transparent to the applications as the applications will benefit from MPTCP without any code changes while they use a standard socket API.

A comparison between network stacks with and without MPTCP support is shown in Figure 13.

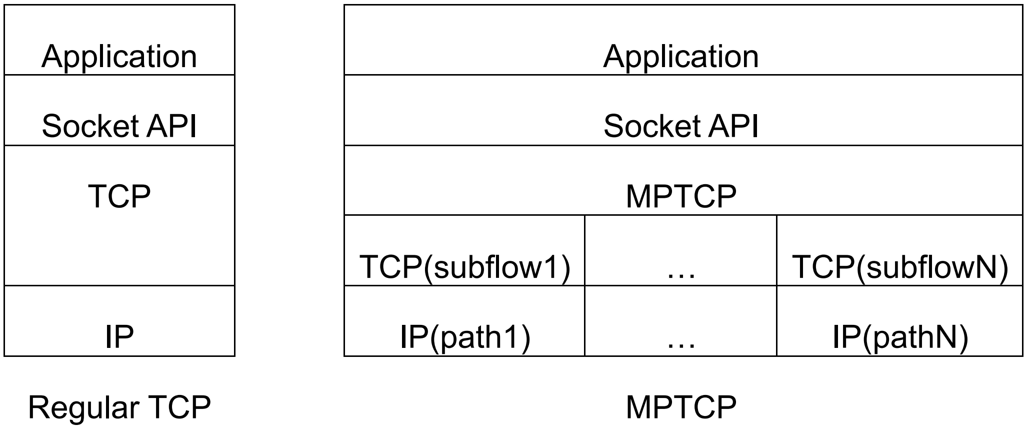


Figure 13. Regular TCP and MPTCP transport in network stacks.

When MPTCP is applied, each network interface has a separate IP address and each of MPTCP path is called as a subflow. Within an MPTCP path, a subflow can have a separated congestion control or a coupled congestion control algorithms over all MPTCP subflows. The MPTCP will aggregate a set of subflows by a sequence number on MPTCP level that achieves reliable in-order delivery.

5.2.3 *TCP connection collision issue in DC's Fat Tree Architecture and Scenarios on Transport Bandwidth Improvement by MPTCP*

Sreekumari and Jung [31] reported that a regular TCP has an issue to obtain a higher utilization of the available bandwidth in data center networks. In data center networks, most of the applications use single-path TCP. As a result, the path may get congested easily by collisions of different TCP connections and cannot utilize the available bandwidth fully as shown in the upper left graph of Figure 14, which results in the degradation of throughput and severe unfairness. For overcoming this problem, multi-path TCP (MPTCP) was proposed in [33] and demonstrated in the upper right graph of Figure 14.

The advantage of this MPTCP approach in DC is that the coupled congestion control algorithms in each MPTCP end system can act on very short time scales to move traffic away from the more congested paths and place it on the less congested ones. As a result, the packet loss rates can be minimized and thereby an improvement of the throughput and fairness of TCP in data center networks [31].

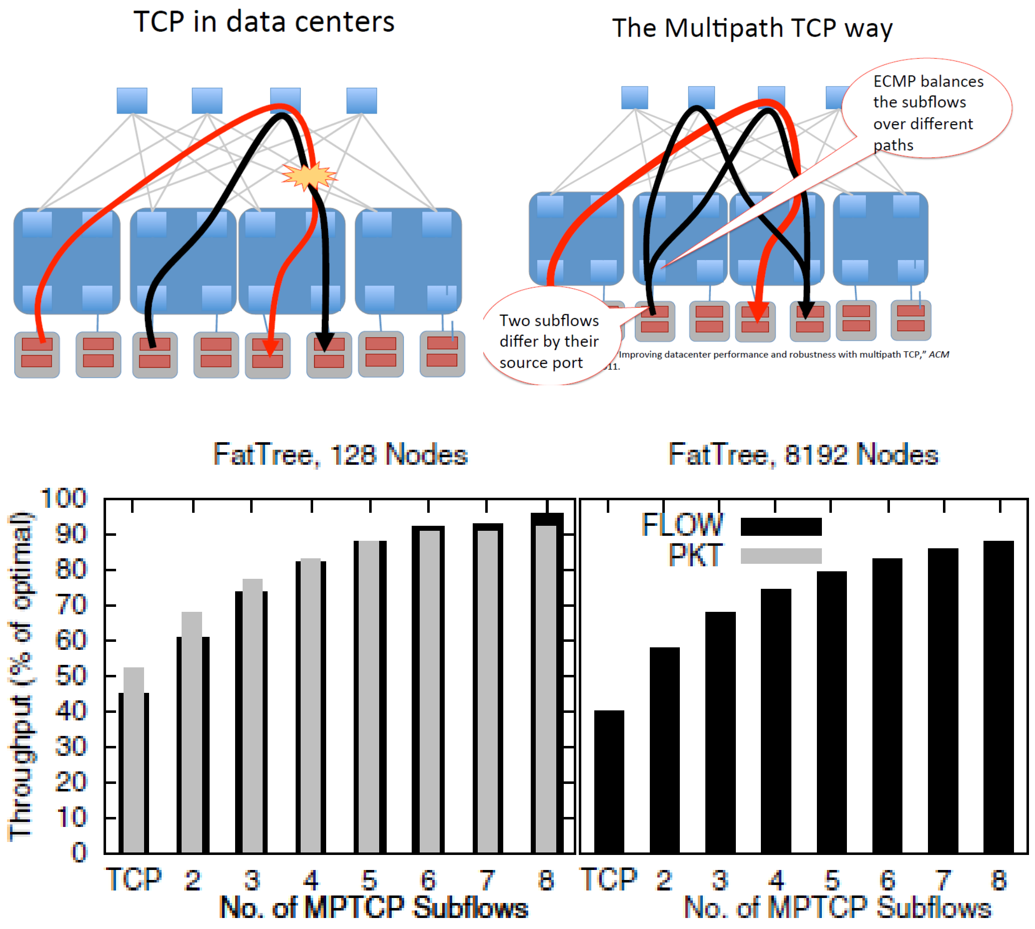


Figure 14. TCP connection collision issue and MPTCP multipath solution in DC's Fat Tree architecture (upper graph). More MPTCP subflows (paths) can get a better utilization in Fat Tree architecture, independent of scale (down graph).

The upper graph in Figure 14 illustrates a TCP connection collision in a DC Fat Tree Architecture and the bandwidth improvement by MPTCP from [32], and more MPTCP subflows for throughput optimization in a data center is shown in [34] and the down graph of Figure 14.

The MPTCP subflow optimal scheduling algorithm in the coupled congestion control seems to be an interesting research subject for optimization of data flow latency of the telco applications in data centers in order to further shorten the end-to-end latency to meet the latency requirements of 5G use cases.

5.3 QUIC

5.3.1 Quick UDP Internet Connections (QUIC)

QUIC is developed as an experimental transport protocol by Google to solve HTTP/2 performance limitation in SPDY with the built-in TLS/SSL improvements. IETF established a QUIC working group in 2016, and Internet Draft specifications on QUIC were submitted to the IETF for standardization [36] [37]. The QUIC working group foresees multipath support in a next step. The QUIC is also an open sourced browser project called Chromium, which can talk to Google's network services already today, e.g., YouTube and Gmail, in the Chrome browser.

Regular HTTP/2 over TLS/TCP and the HTTP/2 over QUIC over UDP transport protocol stacks are demonstrated in Figure 15.

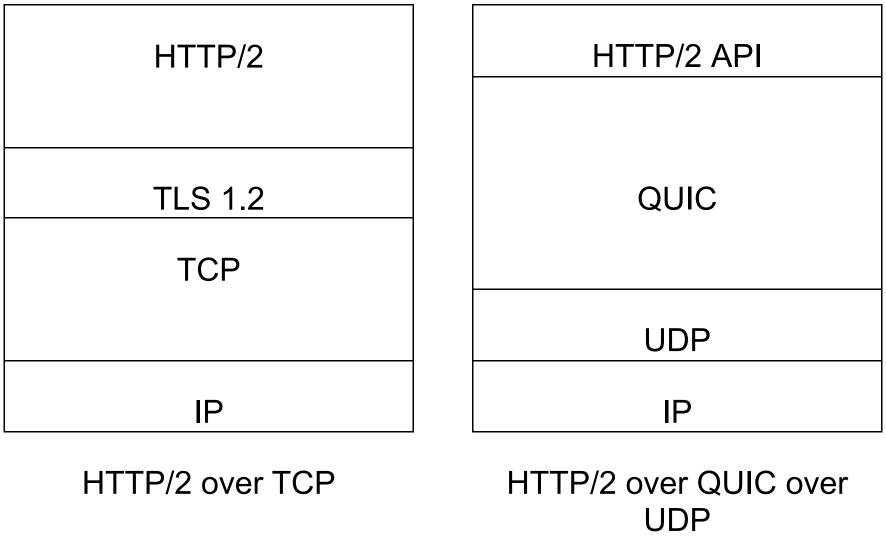


Figure 15. Regular HTTP/2 over TCP and HTTP/2 over QUIC over UDP transport in network stacks.

5.3.2 Main benefit from QUIC

The QUIC is a reliable and multiplexed transport protocol over UDP. Since QUIC runs over UDP, it avoids most problems with middleboxes. The QUIC is always encrypted with the reduced transport latency, which runs in the user-space, therefore, without a need to touch OS for the kernel update.

The main benefit from QUIC are:

- Dramatically reduces connection establishment time to none or only one RTT (first time).

- Improved congestion handling by the packet pacing that reduces the congestion and packet loss.
- Multiplexing without Head of Line blocking (HoLB); the support for multipath transport is ongoing.
- Error control by Forward Error Correction (FEC).

5.3.3 Service Scenario on Transport Improvement by QUIC

In Figure 6, Data Center Networking [13] shows how OTT media streaming services are implemented via DC, where OTT is a media service that allows streaming media contents distributed directly to the consumer over the internet, in parallel with the control logics in DC. In such an implementation, the QUIC protocol, running over UDP, can be used to improve the streaming media performance in terms of throughput and latency, as well as the service performance and robustness.

In data center networks, Bin Dai et.al. [39] suggested a network innovation which entailed using an SDN controller to optimize the multipath routings for QUIC connections to avoid bandwidth competition between streams in the same QUIC connection.

5.4 DCTCP performance benefit and co-exist with existing TCP traffic in DC

5.4.1 Main benefit from ECN

The main benefit with ECN is that IP packets are marked instead of discarded by congested nodes. This leads to a much lower packet loss, which significantly reduces the amount of retransmission.

ECN is transport protocol agnostic, while the ECN marking is done on the ECN bits in the IP header. The actual feedback of the ECN marking is performed by the transport protocols, thus no dedicated reverse in-band or out-of-band signaling channel on the IP layer or below is needed to manage the feedback for the ECN information to the transport protocol sender.

5.4.2 *DCTCP performance benefit on low latency for short flows in DC*

In the data center networks, an effective TCP transport demands on free buffer space available in switches involved in order to get a good enough quality of service performance. However, the bandwidth hungry "background" flows build up queues at the switches, which will impact the TCP transport performance of latency sensitive "foreground" traffic. When throughput-sensitive large flows, delay-sensitive short flows, and bursty query traffic co-exist in the same data center network, the short flow will be delayed by the long flow on the switch queue. DCTCP can improve this by providing the high burst tolerance and low latency for short flows [4].

DCTCP delivers the same or better throughput compared to TCP, while using 90% less buffer space. Operational performance measurements show that DCTCP enables applications to handle ten times the current background traffic, without impacting foreground traffic. When ten times increase in foreground traffic does not cause any timeouts, thus largely eliminating incast problems [4]. DCTCP maintains most of the other TCP features such as slow start, additive increase in congestion avoidance and recovery from packet loss.

The Microsoft Windows Server has DCTCP enabled by default since Windows Server 2012 [10]. In previous Windows versions, and non-server Windows versions, it is disabled by default. DCTCP with ECN support can be enabled using a shell command descripted in [10]:

```
netsh interface tcp set global ecncapability=enabled
```

5.4.3 *DCTCP co-exist with existing TCP and UDP traffic in DC*

DCTCP utilizes explicit congestion notification (ECN) to enhance the flow congestion control and replace the existing TCP packet loss based congestion control algorithm.

However, until recently, DCTCP does not co-exist with existing TCP traffic. So, DCTCP could only be deployed where a clean-slate environment could be arranged, such as in private data centres. A recent investigation by Bob Briscoe et.al. [15] proposes a dual queue coupled Active Queue Management (AQM) to allow DCTCP with scalable congestion controls to safely co-exist with regular TCP traffic [15]. The source code of the Dual Queue Coupled AQM for DCTCP can be found at [16]. IETF specification on the Dual Q Coupled AQM for Low Latency, Low Loss and Scalable Throughput can be found at [17].

In a NFV based infrastructure in a telco/micro DC, although the transport traffic is dominated by TCP transport, there is still about 15% UDP traffic needed for telco services [38]. Using IETF proposed dual queue coupled AQM allows DCTCP to safely co-exist with the UDP Internet traffic [15]. Since the UDP and regular TCP flows will co-exist in the same queue, and the DCTCP flows will run in the new queue in the switch managed by the IETF DualQ AQM for L4S [17]. Therefore, there will be no directly impact on the UDP transport from the latency and QoS perspective.

5.5 IETF L4S & DCTCP standardization

IETF has been enforcing the transport evolution on the latency improvement under Ultra Low Queueing Delay (L4S) Birds of Feather (BoF). The main problem that the L4S aims to solve is the congestion related extra switch queue delay caused by switch buffer bloating and overflow. The solution is to use ECN instead of the packet-loss based congestion control and classify the traffic into separated queues in switches [7] [28] [29].

There are three parts of ongoing standardization works:

- Identifier: A proposed new identifier for Low Latency, Low Loss, Scalable throughput (L4S) packets: Identifying Modified Explicit Congestion Notification (ECN) Semantics for Ultra-Low Queuing Delay [18]

- Network algorithm: Network operators can deploy a new simple active queue management algorithm, that complies with the few constraints specified here: DualQ Coupled AQM for Low Latency, Low Loss and Scalable Throughput [17]

- Host algorithm: The host algorithm developers can deploy new scalable TCP algorithms, e.g. Datacenter TCP (DCTCP): TCP Congestion Control for Datacenters [8]

L4S provides a generic ultra-low latency solution that allows TCP and DCTCP to co-exist in the same DC. There is much interest for L4S from different larger companies such as Ericsson, Vodafone, Microsoft etc.

Because the ECN bits are defined in the IP header and there is no copying of the ECN bits between the inner and outer IP headers, no deep packet inspection is needed. Consequently, ECN also work over IPsec tunnels. L4S will achieve high link utilization with low queue latency. Services that can exploit L4S include for instance 360-degrees Video and Virtual Reality. The usage of ECT(1) by L4S which could already be used in some deployment.

5.6 3GPP specified ECN standardization for Active Queue Management (AQM)

3GPP was recently updated to reference RFC 7567 [19] in 3GPP TS 36.300 [20], see section 11.6, for Evolved Universal Terrestrial Radio Access Network.

The eNB and the UE support of the Explicit Congestion Notification (ECN) are specified. ECN is beneficial especially for latency-sensitive interactive applications such as chat and gaming as well as for real-time voice and video, this because loss as a congestion signal is avoided, losses that would otherwise necessitate retransmission of packets with additional application delay as a result.

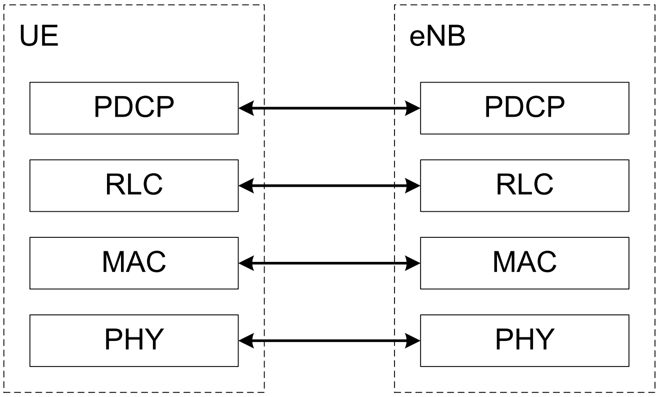


Figure 16. The user plane in radio protocol architecture.

Figure 16 show how the user plane layer 2 in the radio protocol architecture is split into the following sublayers: Medium Access Control (MAC), Radio Link Control (RLC) and Packet Data Convergence Protocol (PDCP). The eNB should set the Congestion Experienced (CE) codepoint ('11') in PDCP SDUs (Service Data Unit) in the downlink direction to indicate downlink (radio) congestion if those PDCP SDUs have one of the two ECN-Capable Transport (ECT) code points set. The eNB should set the Congestion Experienced (CE) code point ('11') in PDCP SDUs in the uplink direction to indicate uplink (radio) congestion if those PDCP SDUs have one of the two ECN-Capable Transport (ECT) code points set.

5.7 3GPP Enhanced ECN support in NR on 5G

3GPP RAN2 has posted a submission from Ericsson on an addition of enhanced ECN support in NR [21]. The intent is to add support for timely and efficient ECN marking in NR (5G). This will make it easier to implement support for L4S in 5G.

The proposal descripted that ECN support in the PDCP layer only becomes a limiting factor for services that require low latency and high throughput, and suggested to add support for enhanced ECN for Uplink and Downlink user plane traffic in NR.

The example simulation in [21] shows a comparison between three congestion control algorithms, CUBIC, BBR and DCTCP-bis, in a scenario where a large file is transmitted over a 20ms RTT bottleneck and the link rate changes between 20 and 100Mbps in 5 second steps. The three congestion control algorithms are run in one iteration each.

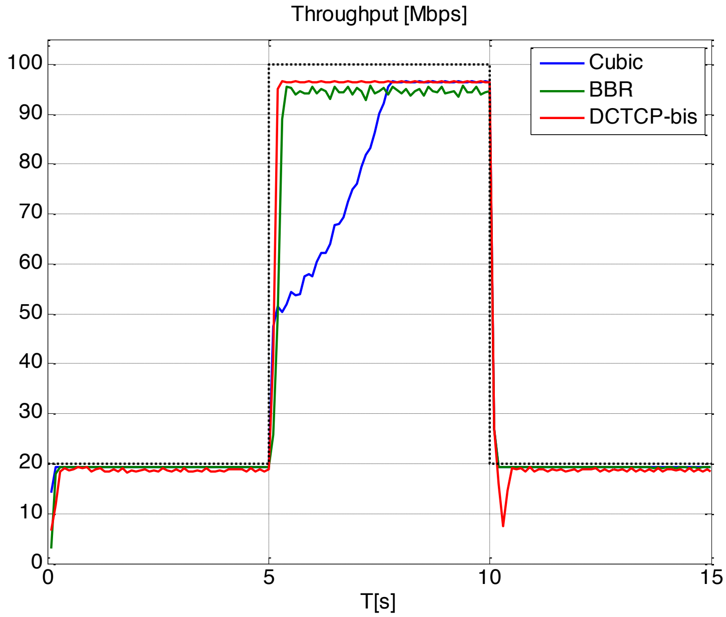


Figure 17. Throughput via different congestion control algorithms [21].

Figure 17 shows the throughput via different congestion control algorithms [21]. DCTCP-bis manages to fill the bottleneck most efficiently with BBR on second place. CUBIC has problems to ramp up the speed when the link bit rate increases due to the AIMD nature of CUBIC. The graph also shows that DCTCP-bis is slightly faster than BBR to reach full link utilization.

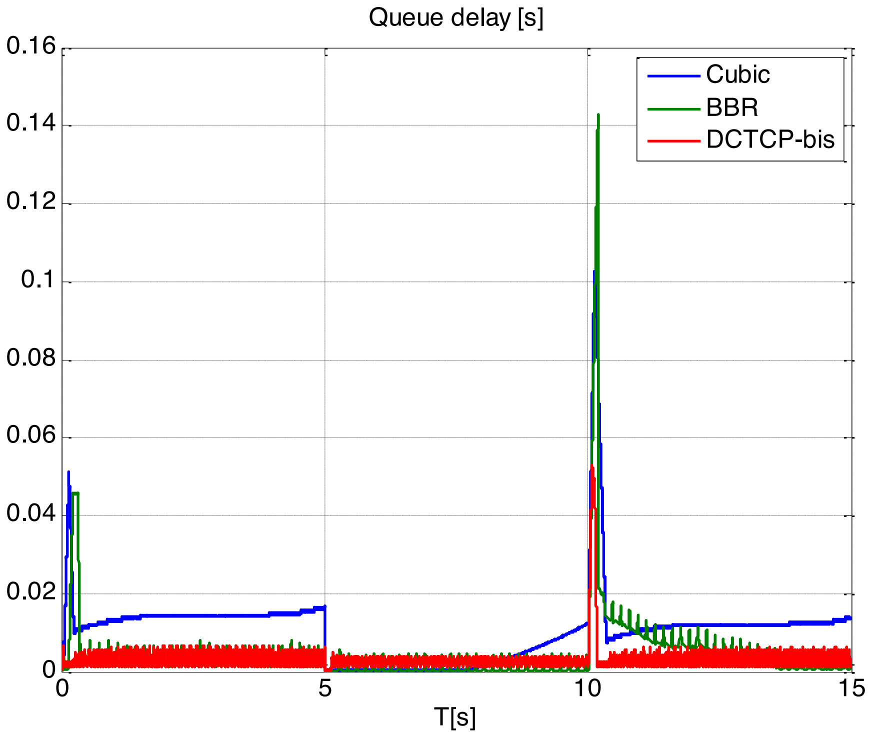


Figure 18. Queue delay via different congestion control algorithms [21].

Figure 18 shows the queue delay in the bottleneck via different congestion control algorithms [21]. CUBIC has the largest standing queue. This is because the AQM must be tuned with a higher mark/discard threshold to allow for full link utilization. The graph illustrates that both CUBIC and BBR gives a slow start delay spike, this can have a negative effect on other already ongoing flows as they will also experience this delay spike. DCTCP-bis does not give any noticeable delay spike in slow start, the latter indicates that the L4S concept, with enhanced ECN support in RAN and congestion control adapted for this, has potential to give a low queue delay even in complex scenarios with multiple parallel flows over the same bottleneck. BBR manages to reach roughly the same queue delay as DCTCP-bis. BBR, however, takes longer to restore a low queue delay when the bottleneck bandwidth drops from 100 to 20Mbps than what is the case for DCTCP-bis.

5.8 5G Latency Requirement on Transport Protocol

The new use cases being defined in 5G put new technical requirements on the networks that will require new technical innovations. Some of these technical innovations require a new air interface, something which is currently being discussed in terms of the 5G New Radio (NR). Other use cases require technology evolution from LTE (4G) as eMBB (enhanced Mobile Broadband). Some of these technical innovations in 5G can also benefit the current networks. Thus, the migration of the stepwise technology evolution will help operators to prepare for the introduction of 5G NR.

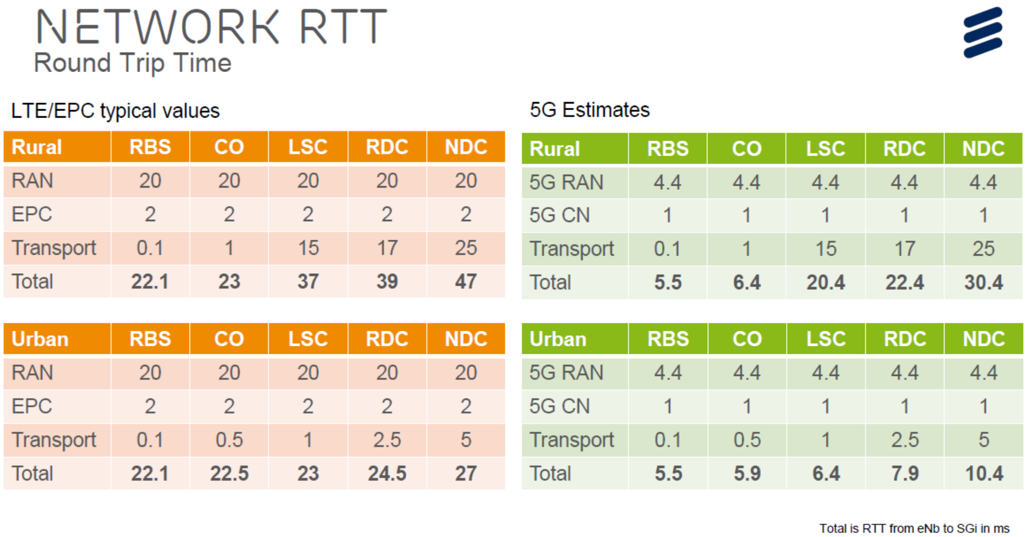
The 5G is depicted with three principal dimensions in 5G standardizations, which require different performances in different scenarios descripted in different use cases. 5G will not only be about eMBB, but also massive Machine Type Communications (mMTC) and Ultra-Reliable and Low Latency Communications (URLLC) use cases.

The 5G NR use cases will ultimately require a high bandwidth (10 Gbps data speed), low latency (1 msec latency), 10 – 100 times more connected devices than LTE (4G), as well as a 10 years' battery life for the low power IoT devices.

Ericsson and Telia performed a 5G outdoor trial which already demonstrated a 15 Gbps data speed, that is 40 times faster than Telia's current LTE deployment, and a 3 ms latency over the 5G radio link [22].

The latency requirement from 5G is more tough to fulfil on the radio and transport networks, which has a transport bottleneck that relies on the latency performance of the flow control in the transport protocol. A further reduction of the end-to-end latency needs to push Micro Data Center located to the distributed Cloud edge in NG-CO (Next Generation Central Office) for NFV deployment that performs the virtualized telco network functions. Therefore, it is reasonable and effective to push the micro DC to the cloud edge, and in so doing reducing the end-to-end latency for the use cases of 5G applications and services.

Figure 19 shows measurements of the latency performance for the LTE (4G) vs. 5G systems from a pre-commercial 5G trial on rural and urban areas. For 5G on urban areas, the measures show that the RTT (Round Trip Time) is about 6 ms for 5G radio, 5G core and the transport operations between UE and the NFV on micro data center at distributed cloud edge [23].



NETWORK RTT

Round Trip Time

LTE/EPC typical values

Rural	RBS	CO	LSC	RDC	NDC
RAN	20	20	20	20	20
EPC	2	2	2	2	2
Transport	0.1	1	15	17	25
Total	**22.1**	**23**	**37**	**39**	**47**

Urban	RBS	CO	LSC	RDC	NDC
RAN	20	20	20	20	20
EPC	2	2	2	2	2
Transport	0.1	0.5	1	2.5	5
Total	**22.1**	**22.5**	**23**	**24.5**	**27**

5G Estimates

Rural	RBS	CO	LSC	RDC	NDC
5G RAN	4.4	4.4	4.4	4.4	4.4
5G CN	1	1	1	1	1
Transport	0.1	1	15	17	25
Total	**5.5**	**6.4**	**20.4**	**22.4**	**30.4**

Urban	RBS	CO	LSC	RDC	NDC
5G RAN	4.4	4.4	4.4	4.4	4.4
5G CN	1	1	1	1	1
Transport	0.1	0.5	1	2.5	5
Total	**5.5**	**5.9**	**6.4**	**7.9**	**10.4**

Total is RTT from eNb to SGi in ms

Figure 19. RTT (Round Trip Time) estimates on 5G vs. LTE (4G) networks [23].

The low latency for short flows is exactly what is required in the 5G use cases, which is still challenging in today’s 5G trial deployment. For further end-to-end latency reduction, using ECN enabled DCTCP instead of normal TCP congestion control algorithm will provide shorter end-to-end latency for short flows as well as a high burst tolerance. Therefore, using ECN enabled DCTCP as the transport protocol in the 5G network will benefit the 5G RAN, 5G CN and 5G transport networks by its lower latency characteristics to meet such a challenging.

6 Telco Cloud vs IT Cloud Data Center Networking

6.1 Distributed DC

A data center connected to a parent central DC (Hub-and-Spoke architecture) needs the parent central DC to reach other DCs and/or external networks. A distributed DC could be located in aggregation or access networks instead of the IP core and the distributed DC is also known as a Micro DC.

In a longer perspective, it is expected that telco and IT cloud networking will converge into a converged solution. However, the telco legacy NFV applications and the VNF(s) may have requirements that differ from how generic IT-clouds are built, e.g., 5G latency requirements in the critical IoT applications and Ultra-Reliable and Low Latency Communications (URLLC) use cases. This will need to include the distributed data center or the Micro DC at the access edges, as demonstrated in Figure 20.

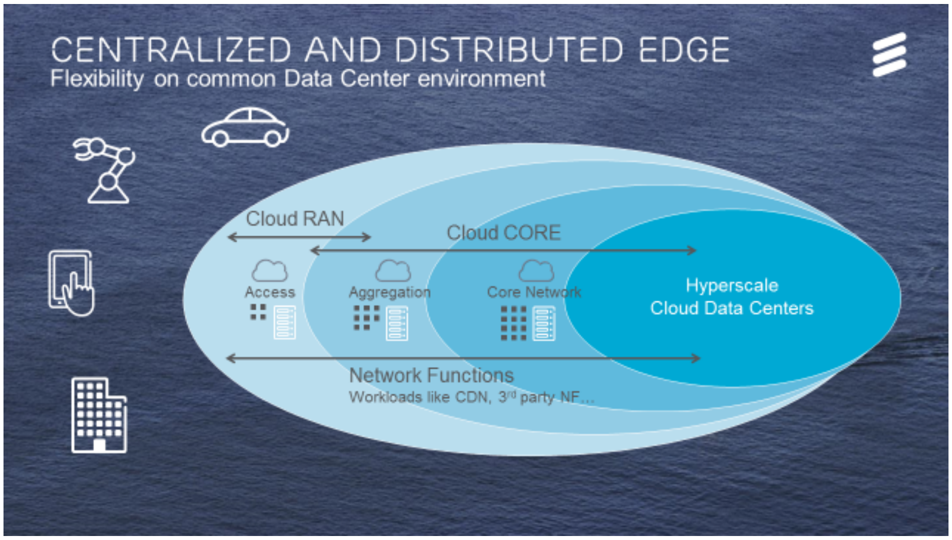
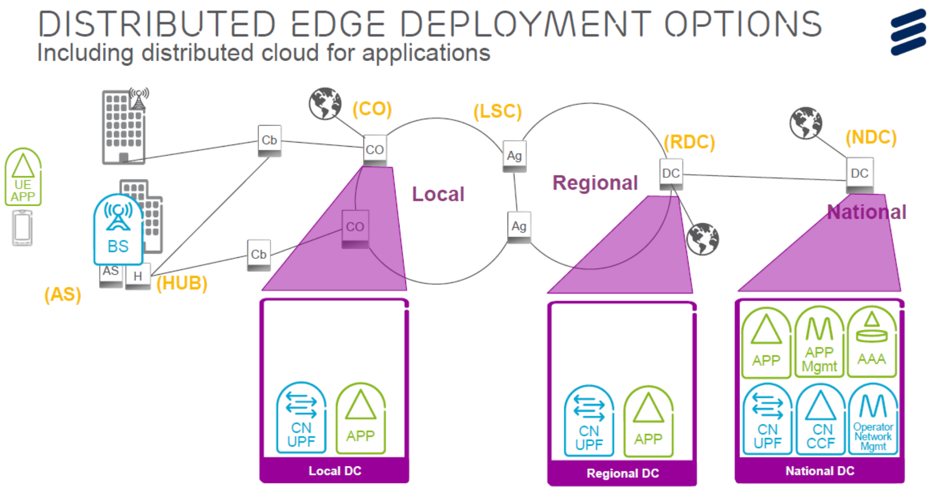

CENTRALIZED AND DISTRIBUTED EDGE
Flexibility on common Data Center environment
Cloud RAN
Cloud CORE
Access
Aggregation
Core Network
Hyperscale
Cloud Data Centers
Network Functions
Workloads like CDN, 3rd party NF…

Figure 20. Centralized DC and Distributed DC at the Cloud edge.

Some strict quality of service requirements in telecom applications and services include low latency end-to-end requirements that can be difficult to fulfill in a generic centralized DC environment, which can only be achieved by optimized NFV(s) in the local Micro DC in CO located at the distributed Cloud edge as shown in Figure 21 [23].



DISTRIBUTED EDGE DEPLOYMENT OPTIONS

Including distributed cloud for applications

Figure 21. The latency optimized NFV(s) in the local Micro DC located at CO.

There are some other arguments about the distributed DC that differ from the generic centralized DC and the NFV deployment at Micro DC at Cloud edge is really needed [14]:

- Direct access to layer 2 resources such as VLAN trunks. Modern IT-cloud platforms, in particular container environments are completely based on layer 3 techniques, that involve even layer 7, e.g., HTTP applications.

- IT-cloud applications often provide direct Internet access, and are therefore more dependent on NAT and firewalling than NFV, which often works in a more protected operator environment.

- NFV applications often have requirements on out-of-band access including control networks that are physically separated from the data network. This increases the cost and complexity of the DC architecture.

- From a redundancy perspective, NFV applications may be more prone to using active redundancy down to individual servers, while IT-cloud applications may rely more on redundancy in the network (e.g. via BGP reroute) to quickly switch over services from one datacenter to another.

- Legacy NFV applications tend to be large and implemented by virtual machines, while IT-cloud applications seem to be moving into a faster container environment where services can be booted up faster and therefore allow for a more flexible micro-services architecture.

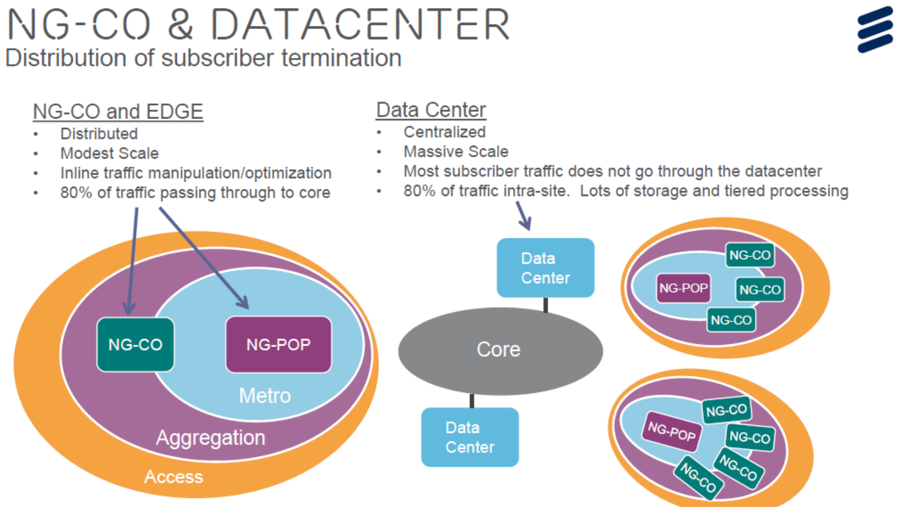
6.2 Central DC / Regional DC / National DC

All central DCs are connected to each other, which is also known as “Standard or Main DC”. It depends on the DC’s scale, the Regional DC and National DC are parts of the Central DC as show in Figure 21 [23].

6.3 Functionalities on Distributed DC in NG-CO and Centralized DC

A Distributed DC at NG-CO is usually located at the cloud edge and the telco applications are deployed as VNF(s) in the Distributed DC in order to meet the strict service requirements of lower end-to-end latency. The Distributed DC manages the online traffic optimization and manipulation at the transport protocol level to perform flow scheduling, flow control and congestion control in order to fulfil end-to-end latency requirements of 5G use cases for the critical IoT applications and the Ultra-Reliable and Low Latency Communications (URLLC) user cases. Around 80% of distributed traffic will pass through to cloud core [13].

The Central DC is located at the cloud core and is deployed with a massive scale as Centralized DC. The most subscriber traffic does not go through the Central DC. Within Central DC, 80% of traffic is intra-site, which involves lots of storage and tiered processing [13], as shown in Figure 22.



NG-CO & DATACENTER

Distribution of subscriber termination

NG-CO and EDGE

- Distributed
- Modest Scale
- Inline traffic manipulation/optimization
- 80% of traffic passing through to core

Data Center

- Centralized
- Massive Scale
- Most subscriber traffic does not go through the datacenter
- 80% of traffic intra-site. Lots of storage and tiered processing

Figure 22. Functionality difference of the Distributed DC (Micro DC) in NG-CO and Centralized DC (Central DC).

6.4 5G E2E example on Distributed and Centralized DC

Figure 23 shows a realization of a distributed DC at the CO and centralized DC, Regional DC and National DC on for 5G use case of eMBB applications [23]. The Regional DC and National DC are parts of the Central DC and all central DCs are connected to each other as shown in Figure 23 as descripted in [23].

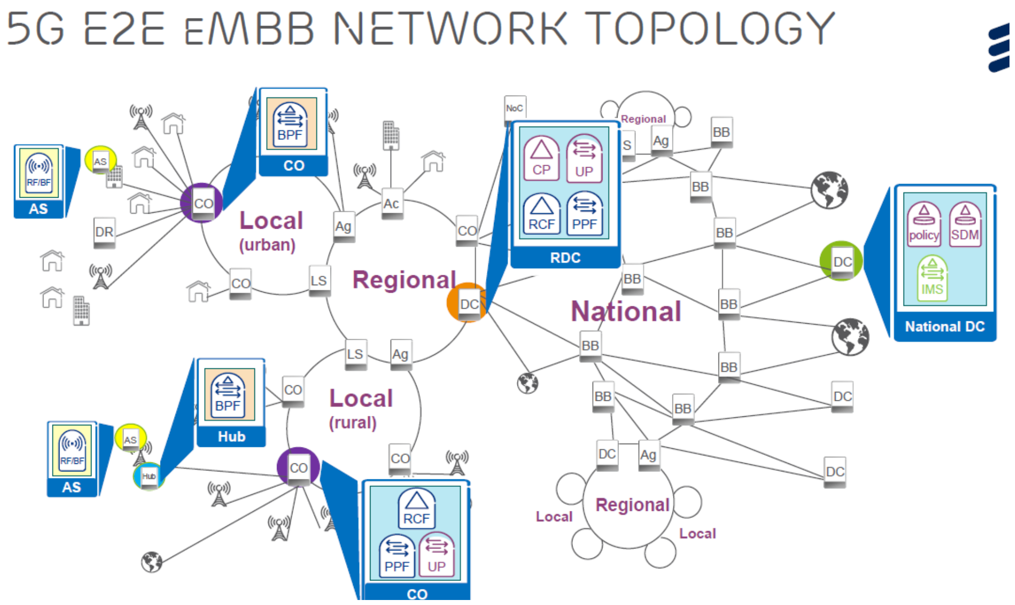


Figure 23. E2E example on Centralized and Distributed DC.

The 5G E2E realization of distributed and centralized DC demonstrates that there is a need on the transport protocol evolution to support 5G use cases on the lower end-to-end network latency, as well as the capacity and topology requirements. 5G Architecture provides a flexible deployment of network functions e.g. by NFV via distributed DC and big data ETL analytic via centralized DC.

To meet the strict 5G service requirements on end-to-end lower latency, Distributed DC needs to involve the online traffic optimization and manipulation at the transport protocol level to perform flow scheduling, flow control and congestion control in order to fulfil end-to-end latency requirements of 5G use cases for the critical IoT mMTC applications and the URLLC applications.

For end-to-end latency improvement, DCTCP is an approach to solve the TCP congestion control issue caused by the switch buffer bloating via implementing ECN based congestion control algorithm in transport layer [8]. DCTCP may co-exist with the normal TCP by applying a modern coupled queue management algorithm in the switches on DC networking [17].

7 Data Centre Geographic Redundancy

7.1 DC Geographic Redundancy

DC Geographic Redundancy for business continuity and disaster mitigation is an absolute must for business-critical systems including NFV based telco service and applications. The most stringent level is a tier 4 data center, which is designed to host the most mission critical computer systems, with fully redundant subsystems, the ability to continuously operate for an indefinite period of time during primary power outages.

The tier 4 data centers must meet many requirements to be certified as tier 4 [24]. One of those requirements is less than half an hour of down time per year. The minimum tier 4 requirements alone may not be enough to maintain 99.995% availability. Failure to do so may not only result in financial losses but in longer term financial damage due to loss of operational credibility. It is for this reason that alternate processing site for geographic redundancy is recommended to increase in the reliability for the data center on business continuity [25].

Geographic redundancy solves the vulnerabilities by geographically separating the backup equipment to decrease the likelihood that occurrences, e.g. power outages in the disaster [25].

7.2 Geographical Inter-DC / Data Center Interconnectivity

Communication between different geographical DC sites separated by a provider transport network. Two models for this are described in this document: “1:1 Mapping Model” and “Overlay Model”. Both models could be coexistence and used in parallel.

The transport network carrying Inter-DC traffic and connecting to external networks is assumed to be based on IP/MPLS or GRE Tunnel, mainly serving communication for telecom applications.

7.2.1 DC Geographic Redundancy: 1:1 Mapping Model

For some VRFs the 1:1 Mapping Model might be chosen, e.g., Infrastructure VRFs, as shown in Figure 24 [26], where IP/MPLS is used for inter-DC traffic and connecting to external networks.

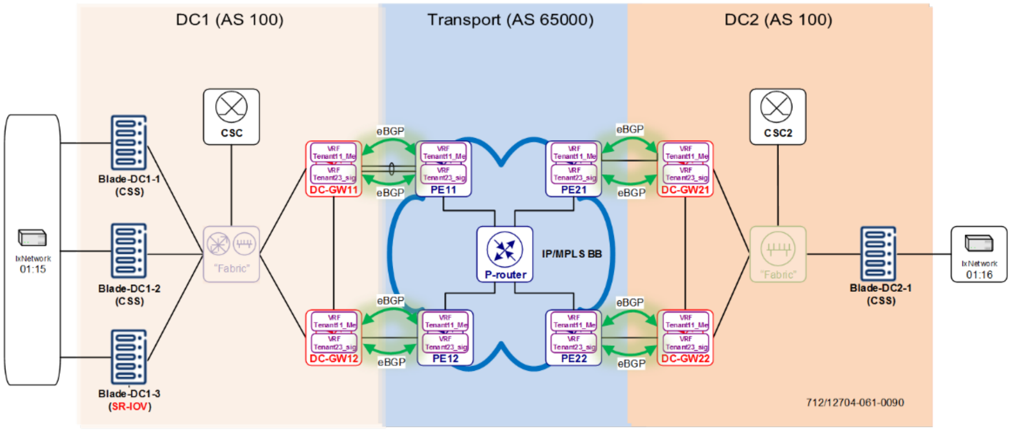


Figure 24. DC Geographic Redundancy: Mapping Model.

MPLS (Multiprotocol Label Switching) is a type of data-carrying technique for high-performance telecommunications networks. The MPLS directs data from one network node to the next node based on short path labels rather than long network addresses, avoiding complex lookups in a routing table. MPLS is located between layer 2 and 3 according to the traditional definition of the OSI model, and is sometimes called a 2.5-protocol. An MPLS node is called a router, but running mainly as a switch.

7.2.2 DC Geographic Redundancy: Overlay Model

For some VRFs the Overlay Model might be chosen, e.g., Application VRFs, as shown in Figure 25 [26], where a GRE Tunnel is used for inter-DC traffic.

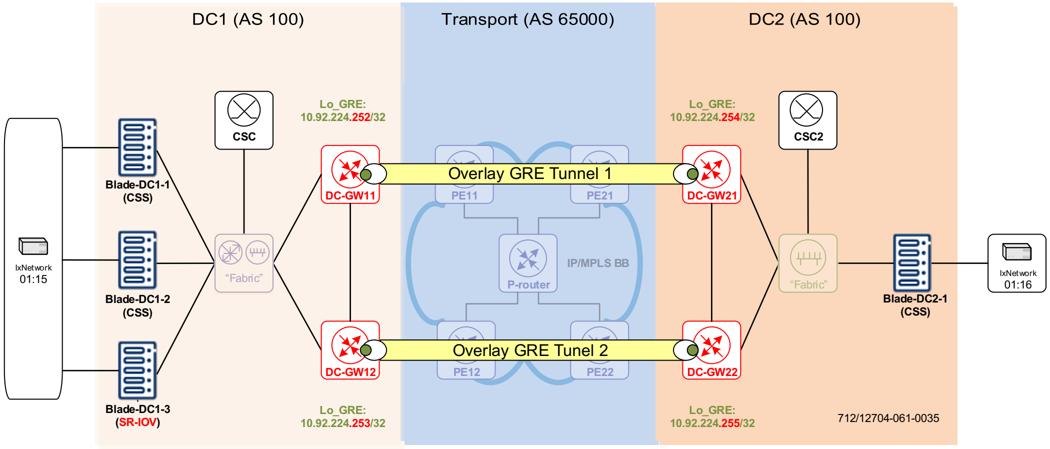


Figure 25. DC Geographic Redundancy: Overlay Model.

GRE (Generic Routing Encapsulation) is a tunneling protocol developed for encapsulation of a wide variety of network layer protocols inside virtual point-to-point links over the general IP networks.

7.2.3 Panasonic Avionics CNaaS, Ericsson Geographic Redundancy DC Design

Figure 26 shows 5G-ready Core for Panasonic Avionics CNaaS deployment including Ericsson's Geographic Redundancy DC design [27].

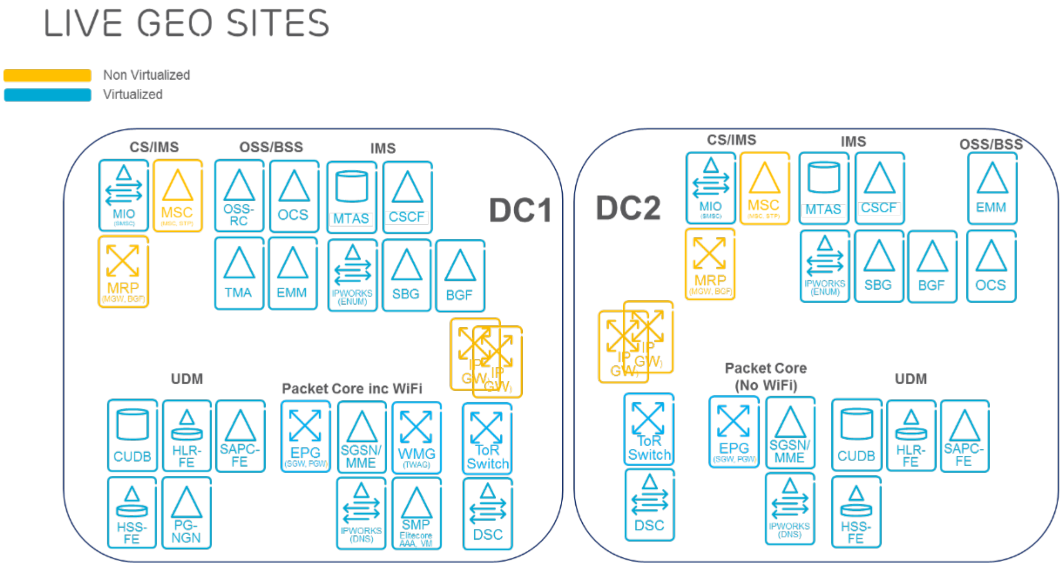
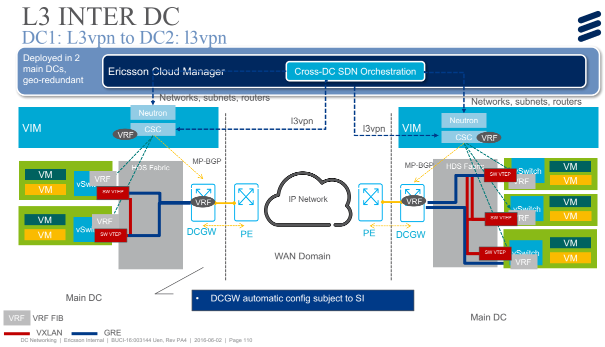


Figure 26. DC Geographic Redundancy design for Panasonic Avionics CNaaS.

7.2.4 Ericsson Geographic Redundancy Layer 3 Inter-DC

Figure 27 shows Ericsson's Geographic Redundancy Layer 3 Inter DC design [1].



L3 INTER DC
DC1: L3vpn to DC2: l3vpn
Deployed in 2 main DCs, geo-redundant
Ericsson Cloud Manager
Cross-DC SDN Orchestration
Networks, subnets, routers
Networks, subnets, routers
VIM
Neutron
CSC
VRF
l3vpn
l3vpn
VIM
Neutron
CSC
VRF
MP-BGP
HDS Fabric
VM
VM
vSwitch
VRF
SW VTEP
DCGW
PE
IP Network
WAN Domain
PE
DCGW
MP-BGP
HDS Fabric
vSwitch
VRF
SW VTEP
VM
Main DC
• DCGW automatic config subject to SI
Main DC
VRF FIB
VXLAN
GRE
DC Networking | Ericsson Internal | BUCI-16:003144 Uen, Rev PA4 | 2016-06-02 | Page 110

Figure 27. DC Geographic Redundancy Layer 3 Inter-DC.

8 Conclusion

In this survey paper, a transport protocol evolution involving the use of the DCTCP, MPTCP and the QUIC transport protocols is proposed to enhance the transport latency, throughput, and robustness, and to deal with the transport flow congestion issue caused by the switch buffer bloating in a data center. The implementation of these transport protocols for 5G mobile networks is also discussed, something that will lead to a lower latency, lower loss, and scalable throughput transport service that fulfils IETF L4S architecture in telco distributed data center.

The main contributions from these transport protocols are:

- By implementing ECN based congestion control, DCTCP will not only solve the latency issue in the TCP congestion control caused by the switch buffer bloating, but also achieve an improved performance on the packet loss and throughput. The DCTCP can co-exist with normal TCP.

- MPTCP in a DC Fat tree architecture can improve transport throughput and latency to mitigate the bandwidth performance issue caused by TCP connection collision within data center. MPTCP is backwards compatible, i.e., it falls back to TCP if the communicating peer does not support MPTCP.

- QUIC includes many of the latest transport improvements and innovations. A scenario shows that QUIC can be used by an SDN controller to optimize the multipath routings for QUIC connections to avoid bandwidth competition between streams in the same QUIC connection. These can be used to improve the transport performance on OTT streaming media delivery.

9 References

[1] DC Networking, BUCI System & Technology, Program Plan, Jun 2016.

[2] Yang Liu, Jogesh K. Muppala, Malathi Veeraraghavan, A Survey of Data Center Network Architectures, 2013. http://www.ece.virginia.edu/mv/pubs/recent-samples/Data-center-Survey.pdf

[3] Brian Lebiednik, Aman Mangal, Niharika Tiwari, A Survey and Evaluation of Data Center Network Topologies, 2016. https://arxiv.org/pdf/1605.01701v1

[4] M. Alizadeh et al., Data Center TCP (DCTCP) https://people.csail.mit.edu/alizadeh/papers/dctcp-sigcomm10.pdf

[5] Stanford DCTCP site: http://simula.stanford.edu/~alizade/Site/DCTCP.html or https://reproducingnetworkresearch.wordpress.com/2012/06/09/dctcp-2/

[6] Data Center TCP (DCTCP) https://www.microsoft.com/en-us/research/publication/data-center-tcp-dctcp/

[7] Ultra-Low Queuing Delay for All: https://riteproject.eu/dctth/

[8] IETF draft: Datacenter TCP (DCTCP): TCP Congestion Control for Datacenters https://tools.ietf.org/html/draft-ietf-tcpm-dctcp-06

[9] Charles Clos, A study of non-blocking switching networks, Mar 1953. https://math.dartmouth.edu/archive/m38s12/public_html/sources/Hall1935.pdf

[10] "Data Center Transmission Control Protocol (DCTCP) (Windows Server 2012)". http://technet.microsoft.com/en-us/library/hh997028.aspx

[11] M. Al-Fares, A. Loukissas, and A. Vahdat, “A scalable, commodity data center network architecture,” ACM SIGCOMM Computer Communication Review, vol. 38, no. 4, pp. 63–74, 2008.

[12] A. Andreyev. (2014, Nov.) Introducing data center fabric, the next-generation facebook data center network. [Online]. Available: https://code.facebook.com/posts/360346274145943/

[13] Christoph Meyer, Data Center Networking, BUCI Tech Day 2015. https://erilink.ericsson.se/eridoc/erl/objectId/09004cff8a3ecdb4?docno=&action=current&format=pdf

[14] Anders Jansson G, TEA Principles for Data Center Networking, 2016

[15] Koen De Schepper, Olga Bondarenko, Ing-Jyh Tsang, Bob Briscoe `Data Centre to the Home': Ultra-Low Latency for All" (2015), http://www.bobbriscoe.net/projects/latency/dctth_preprint.pdf

[16] Source code of Dual Queue Coupled AQM, https://github.com/olgabo/dualpi2

[17] IETF draft: DualQ Coupled AQM for Low Latency, Low Loss and Scalable Throughput, https://datatracker.ietf.org/doc/draft-ietf-tsvwg-aqm-dualq-coupled/

[18] IETF draft: Identifying Modified Explicit Congestion Notification (ECN) Semantics for Ultra-Low Queuing Delay, https://datatracker.ietf.org/doc/draft-ietf-tsvwg-ecn-l4s-id/

[19] IETF RFC 7567 IETF Recommendations Regarding Active Queue Management, https://tools.ietf.org/html/rfc7567

[20] 3GPP TS 36.300 V14.2.0 (2017-03), http://www.3gpp.org/ftp/Specs/archive/36_series/36.300/36300-e20.zip

[21] 3GPP TSG-RAN WG2 #98, Tdoc R2-1704368, Ericsson's proposal on the addition of enhanced ECN support in 5G NR, http://www.3gpp.org/ftp/TSG_RAN/WG2_RL2/TSGR2_98/Docs/R2-1704368.zip

[22] Telia and Ericsson demonstrate record-breaking speed and latency in live 5G field trial, https://www.ericsson.com/news/161013-telia-and-ericsson-demonstrate-record-breaking-speed-and-latency-in-live-5g-field-trial_244039853_c

[23] Göran Hall and Torbjörn Cagenius, BICP Tech Day 2016, 5G Ready Core Functional and Applied Network Architecture, https://erilink.ericsson.se/eridoc/erl/objectId/09004cff8b76be83?docno=&action=current&format=pdf

[24] Wikipedia, Data center, https://en.wikipedia.org/wiki/Data_center

[25] H. M. Brotherton and J. Eric Dietz, Data Center Site Redundancy, 2014, http://docs.lib.purdue.edu/cgi/viewcontent.cgi?article=1026 &context=i3r2

[26] EIN - Data Center Interconnect (DCI) Network Design, https://erilink.ericsson.se/eridoc/erl/objectId/09004cff8b58c1f7?action=current&format=ppt12

[27] Global success for 5G-ready Core with Panasonic Avionics, https://internal.ericsson.com/news/69815/global-success-5g-ready-core-panasonic-avionics?unit=31439876

[28] IETF draft: Low Latency, Low Loss, Scalable Throughput (L4S) Internet Service: Architecture, https://datatracker.ietf.org/doc/draft-ietf-tsvwg-l4s-arch/

[29] IETF draft: Explicit Congestion Notification (ECN) Experimentation, https://datatracker.ietf.org/doc/draft-ietf-tsvwg-ecn-experimentation/

[30] Ulf Ewaldsson, Ericsson's New Cloud Strategy, 2016. https://www.nyteknik.se/digitalisering/har-ar-ericssons-nya-molnstrategi-6801037

[31] Prasanthi Sreekumari and Jae-il Jung, Transport protocols for data center networks: a survey of issues, solutions and challenges, Photon Netw Commun (2016) 31:112–128

[32] Olivier Bonaventure, Multipath TCP, 2013. https://www.ietf.org/edu/tutorials/MultipathTCP-IETF87.pptx.pdf

[33] Ming, L., Lukyanenko, A., Tarkoma, S., Yla-Jaaski, A., MPTCP incast in data center networks. Commun. China **11**(4), 2014, 25–37

[34] Costin Raiciu, Sebastien Barre, Christopher Pluntke, Adam Greenhalgh, Damon Wischik, Mark Handley, Improving datacenter performance and robustness with multipath TCP, ACM SIGCOMM, 2011, 266-277 http://vincen.tl/cis700sp17/mptcp-sigcomm11.pdf

[35] Ericsson Hyperscale Datacenter System 8000, https://www.ericsson.com/hyperscale/cloud-infrastructure/hyperscale-datacenter-system

[36] IETF draft: QUIC: A UDP-Based Secure and Reliable Transport for HTTP/2, https://tools.ietf.org/html/draft-tsvwg-quic-protocol-02

[37] IETF draft: QUIC Loss Recovery and Congestion Control, https://tools.ietf.org/html/draft-tsvwg-quic-loss-recovery-01

[38] Irena Trajkovska, Michail-Alexandros Kourtis, et.al., SDN-based service function chaining mechanism and service prototype implementation in NFV scenario, Computer Standards & Interfaces, Volume 54, Part 4, November 2017, Pages 247-265

[39] Bin Daia, Guan Xua, et.al., Enabling network innovation in data center networks with software defined networking: A survey, Journal of Network and Computer Applications 94 (2017) 33–49

[40] Project Calico - A Pure Layer 3 Approach to Virtual Networking https://www.projectcalico.org

Telco Distributed DC with Transport Protocol Enhancement for 5G Mobile Networks

Distributed data center hosts telco virtual network functions, mixing workloads that require data transport through transport protocols with either low end-to-end latency or large bandwidth for high throughput, e.g., from tough requirements in 5G use cases. A trend is the use relatively inexpensive, off-the-shelf switches in data center networks, where the dominated transport traffic is TCP traffic. Today's TCP protocol will not be able to meet such requirements. The transport protocol evolution is driven by transport performance (latency and throughput) and robust enhancements in data centers, which include new transport protocols and protocol extensions such as DCTCP, MPTCP and QUIC protocols and lead to intensive standardization works and contributions to 3GPP and IETF.

By implementing ECN based congestion control instead of the packet-loss based TCP AIMD congestion control algorithm, DCTCP not only solves the latency issue in TCP congestion control caused by the switch buffer bloating but also achieves an improved performance on the packet loss and throughput. The DCTCP can also co-exist with normal TCP by applying a modern coupled queue management algorithm in the switches of DC networks, which fulfills IETF L4S architecture. MPTCP is an extension to TCP, which can be implemented in DC's Fat tree architecture to improve transport throughput and shorten the latency by mitigating the bandwidth issue caused by TCP connection collision within the data center. The QUIC is a reliable and multiplexed transport protocol over UDP transport, which includes many of the latest transport improvements and innovation, which can be used to improve the transport performance on streaming media delivery.

The Clos topology is a commonly used network topology in a distributed data center. In the Clos architecture, an over-provisioned fabric cannot handle full wire-speed traffic, thus there is a need to have a mechanism to handle overload situations, e.g., by scaling out the fabric. However, this will introduce more end-to-end latency in those cases the switch buffer is bloated, and will cause transport flow congestion.

In this survey paper, DCTCP, MPTCP and QUIC are discussed as solutions for transport performance enhancement for 5G mobile networks to avoid the transport flow congestion caused by the switch buffer bloating from overloaded switch queue in data centers.

Faculty of Health, Science and Technology

WORKING PAPER | December 2017