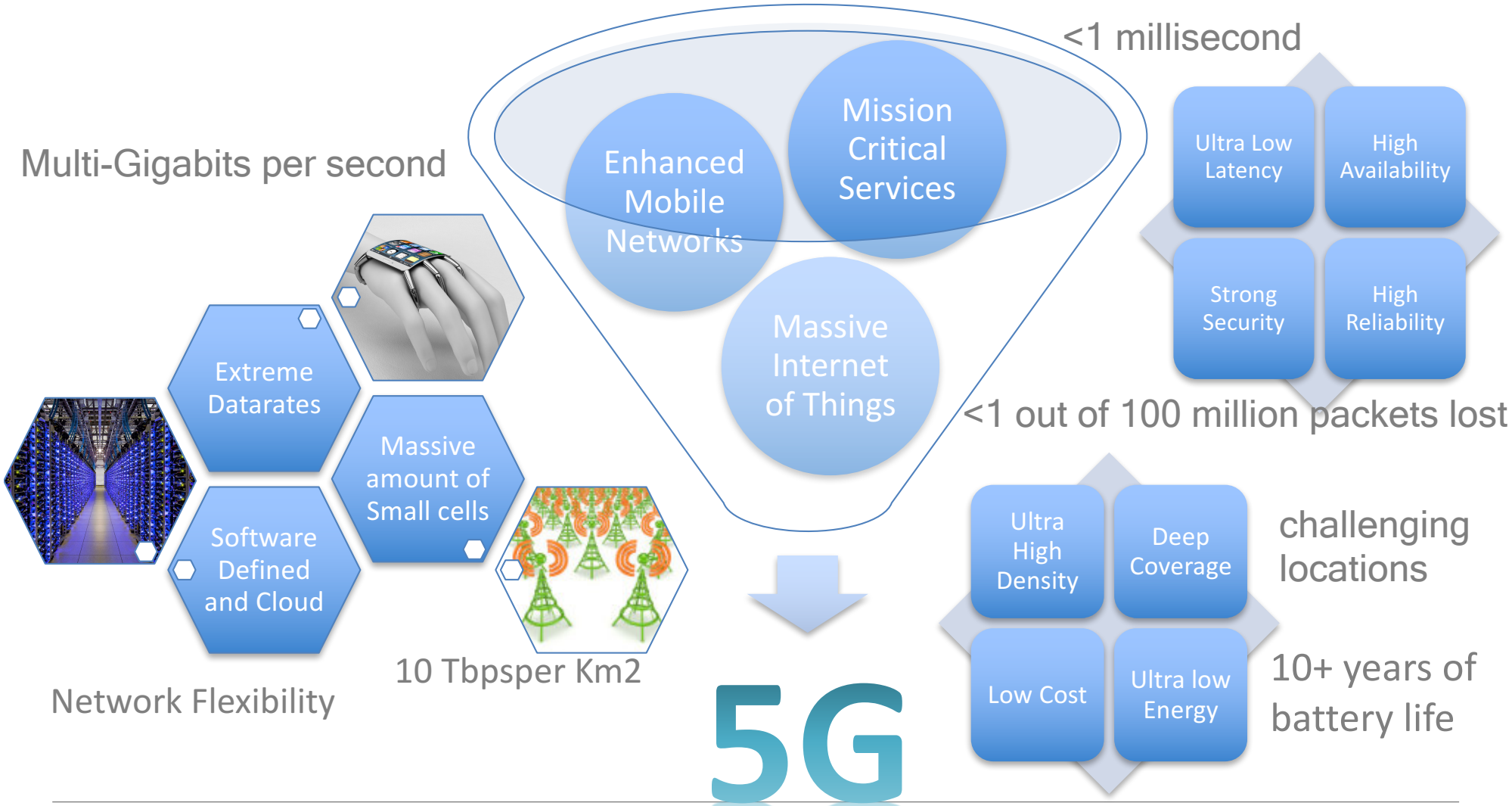
ON POWER EFFICIENT VIRTUAL NETWORK FUNCTIONS PLACEMENT FOR 5G NETWORKS

ANDREAS KASSLER, KARLSTADS UNIVERSITET

COMPUTER SCIENCE
DATAVETENSKAP

THE 5G VISION – UNIFIED CONNECTIVITY



COMPUTER SCIENCE
DATAVETENSKAP

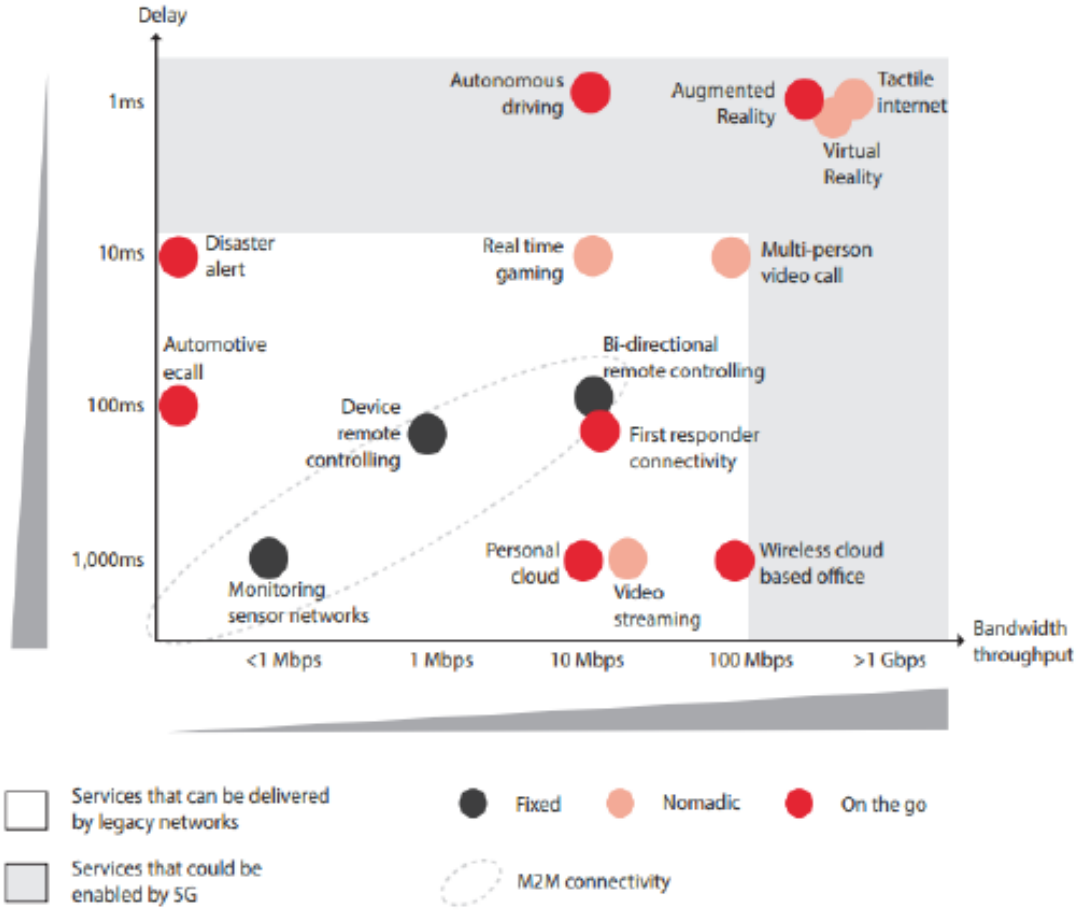
EXTREME MOBILE BROADBAND

PICTURES COURTESY QUALCOMM

- Extreme Capacity to each user and extremely dense deployments
  - Multi Gbps air interface
  - Challenging interference management and backhaul design
  - Flexible architecture: NFV, SDN and Cloud

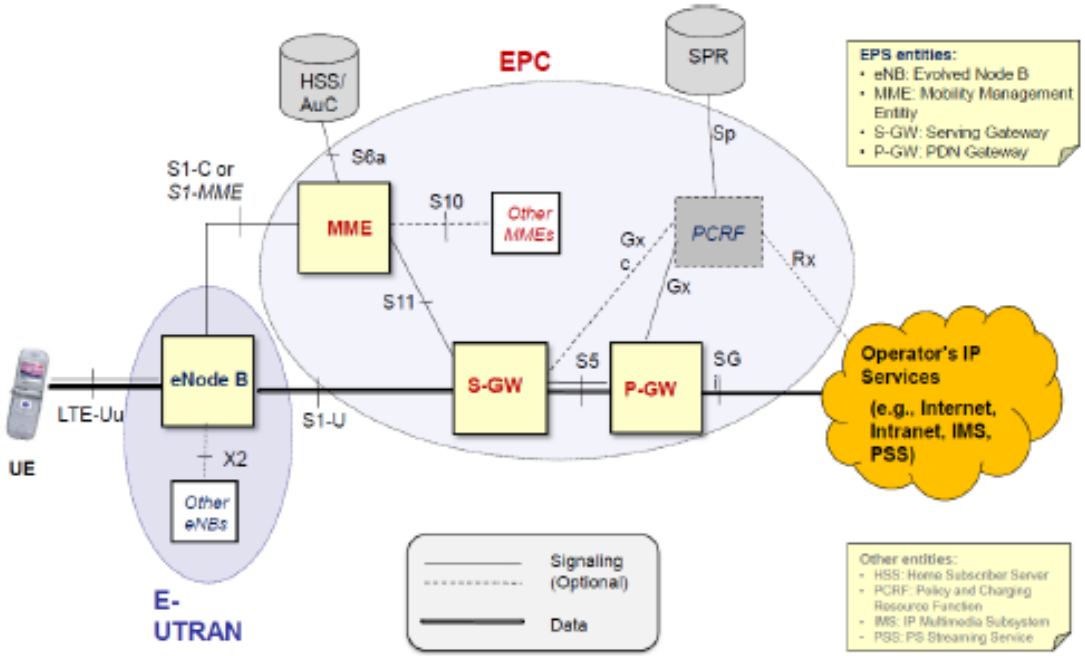
DO WE REALLY NEED 5G?

- Benefits
  - Can provide existing services at lower OPEX and CAPEX due to NFV and SDN
  - Rapid service innovation
  - Support new services that cannot be provided with 4G
  - New service industries, e.g. verticals for industry 2.0, automotive, etc.

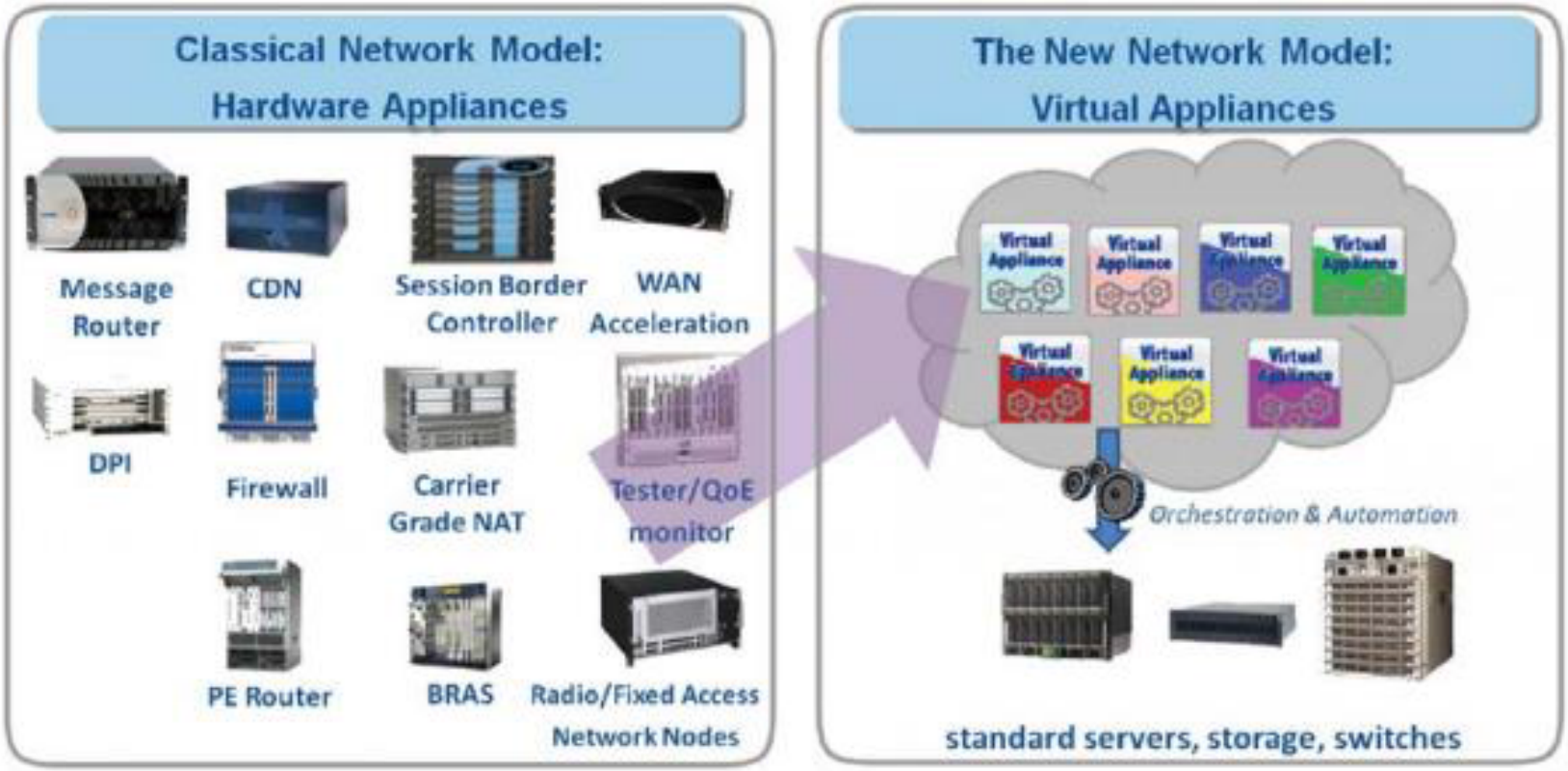


CURRENT (MOBILE) NETWORKS: A MESS

- Too complex
- Diverse protocol zoo
- Too slow to market
- Only big guys can innovate
- Mainframe mentality
- High OPEX
- Low utilization (peak provisioning)

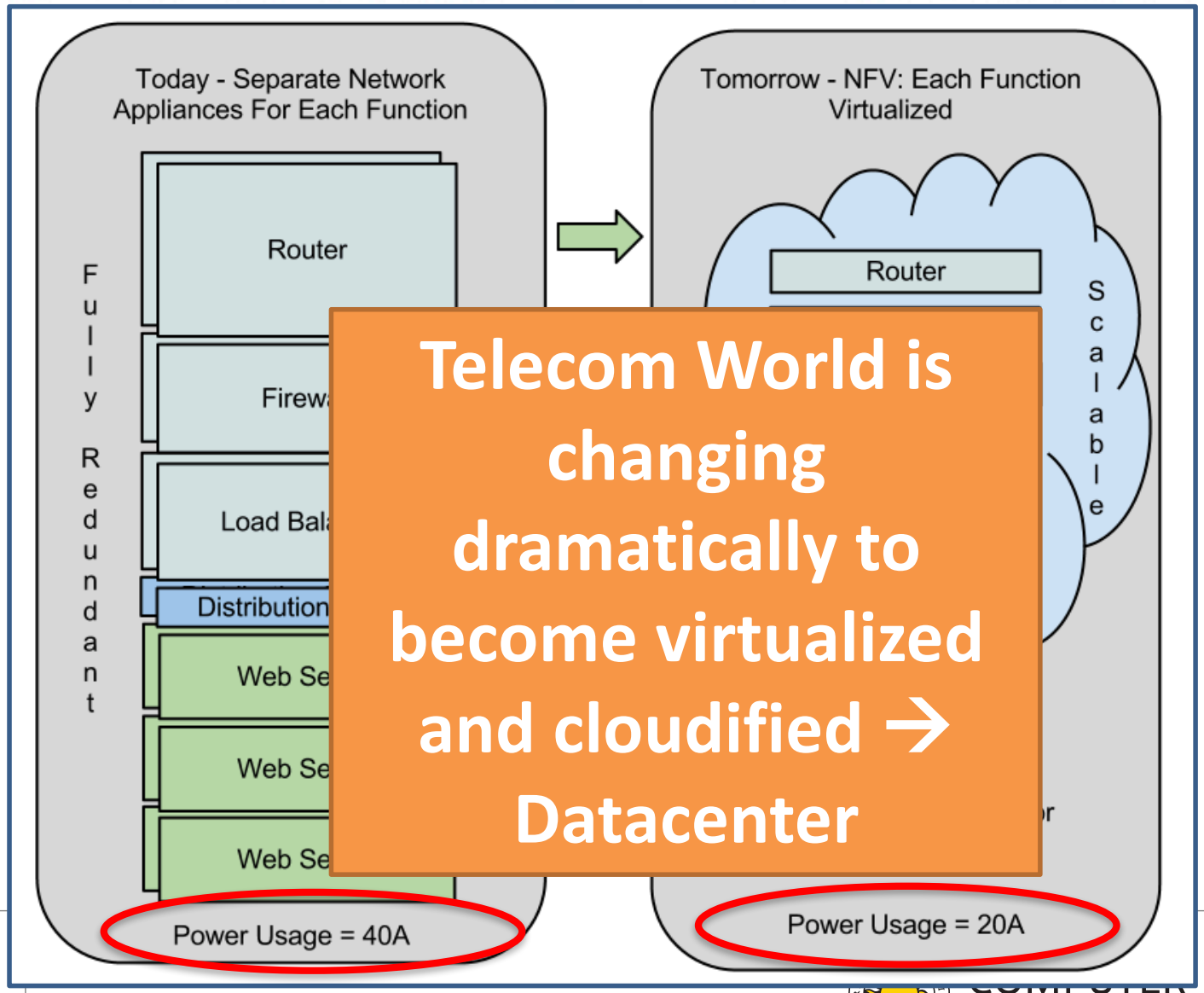


EVERYTHING GETTING VIRTUALIZED

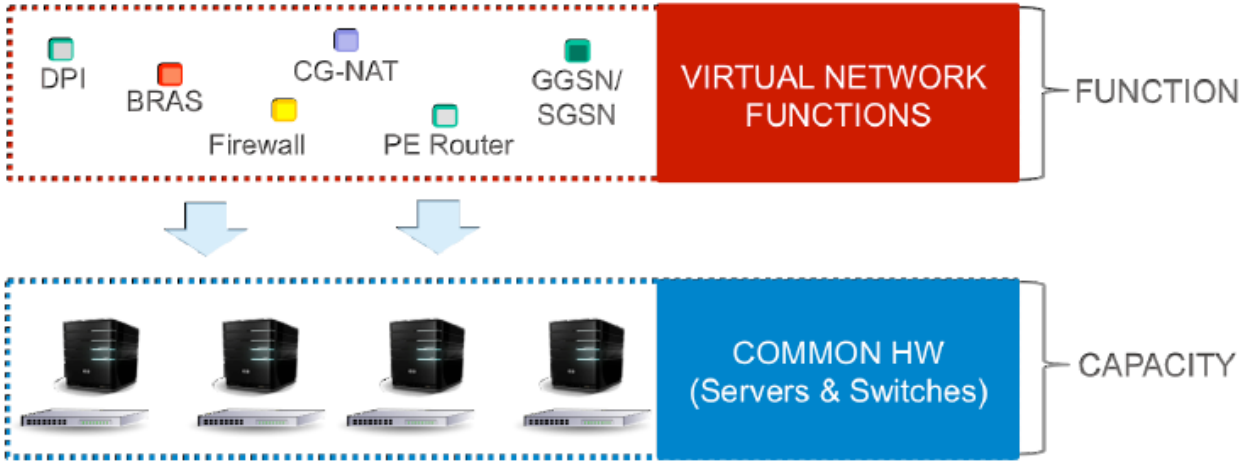


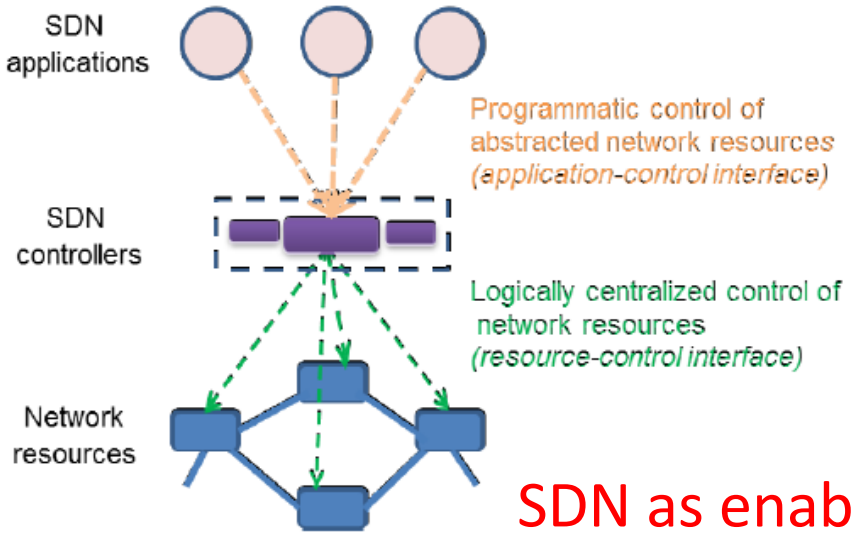
- Transformation of legacy equipment into software packages (many OpenSource) that run on standard high volume general purpose servers using Virtualization has already begun

NETWORK FUNCTION VIRTUALIZATION



NFV AND SDN

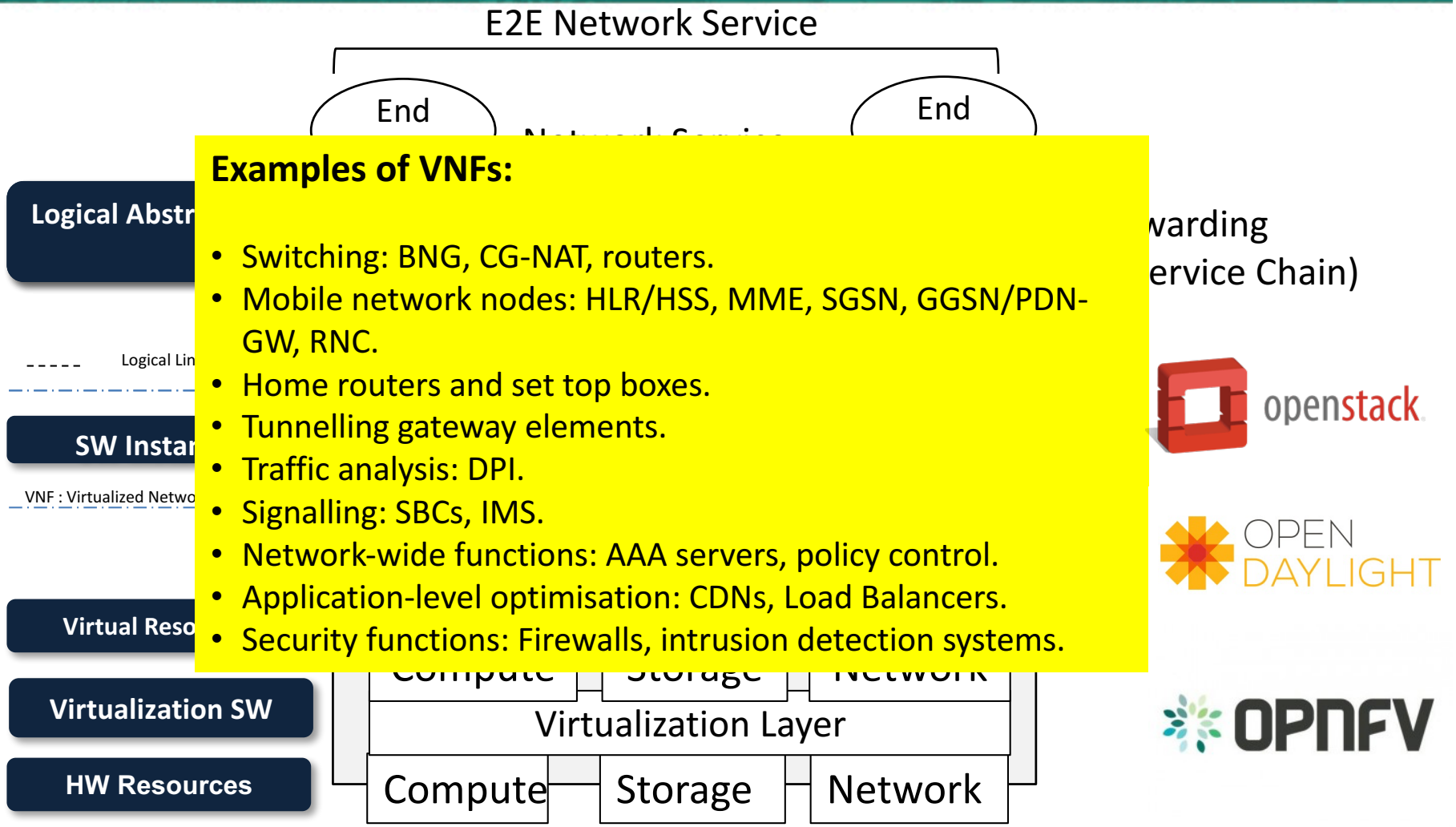




SDN as enabler for NFV

- NFV: Decouple functionality from capacity
  - Elasticity
  - Heterogeneity
- Software Defined Networking
  - Decouple control from data plane
  - Programmability
  - Abstraction and virtual network

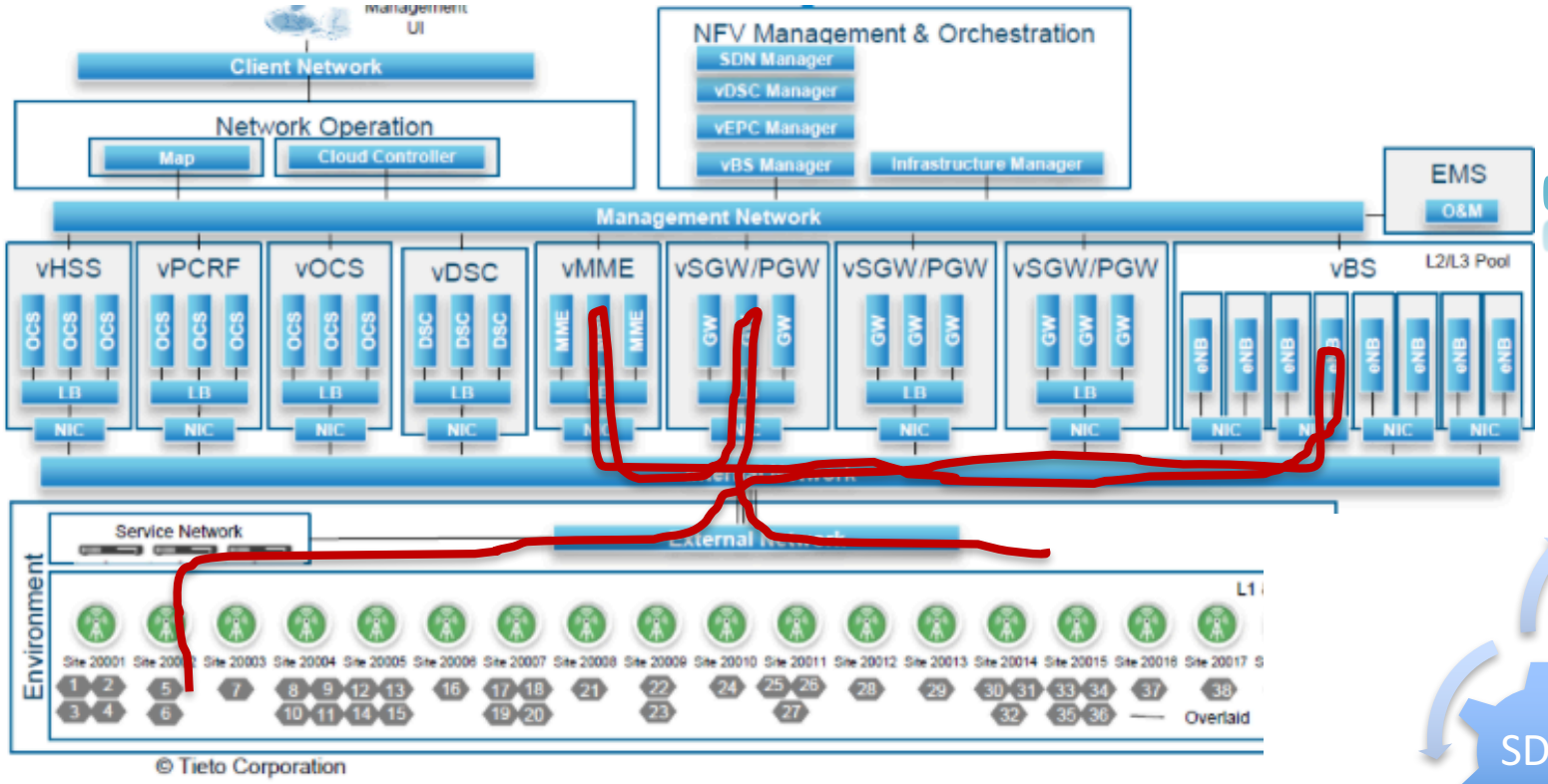
VIRTUAL NETWORK FUNTIONS

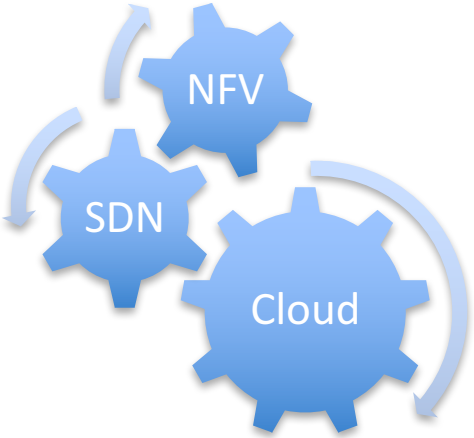


Examples of VNFs:

- Switching: BNG, CG-NAT, routers.
- Mobile network nodes: HLR/HSS, MME, SGSN, GGSN/PDN-GW, RNC.
- Home routers and set top boxes.
- Tunnelling gateway elements.
- Traffic analysis: DPI.
- Signalling: SBCs, IMS.
- Network-wide functions: AAA servers, policy control.
- Application-level optimisation: CDNs, Load Balancers.
- Security functions: Firewalls, intrusion detection systems.

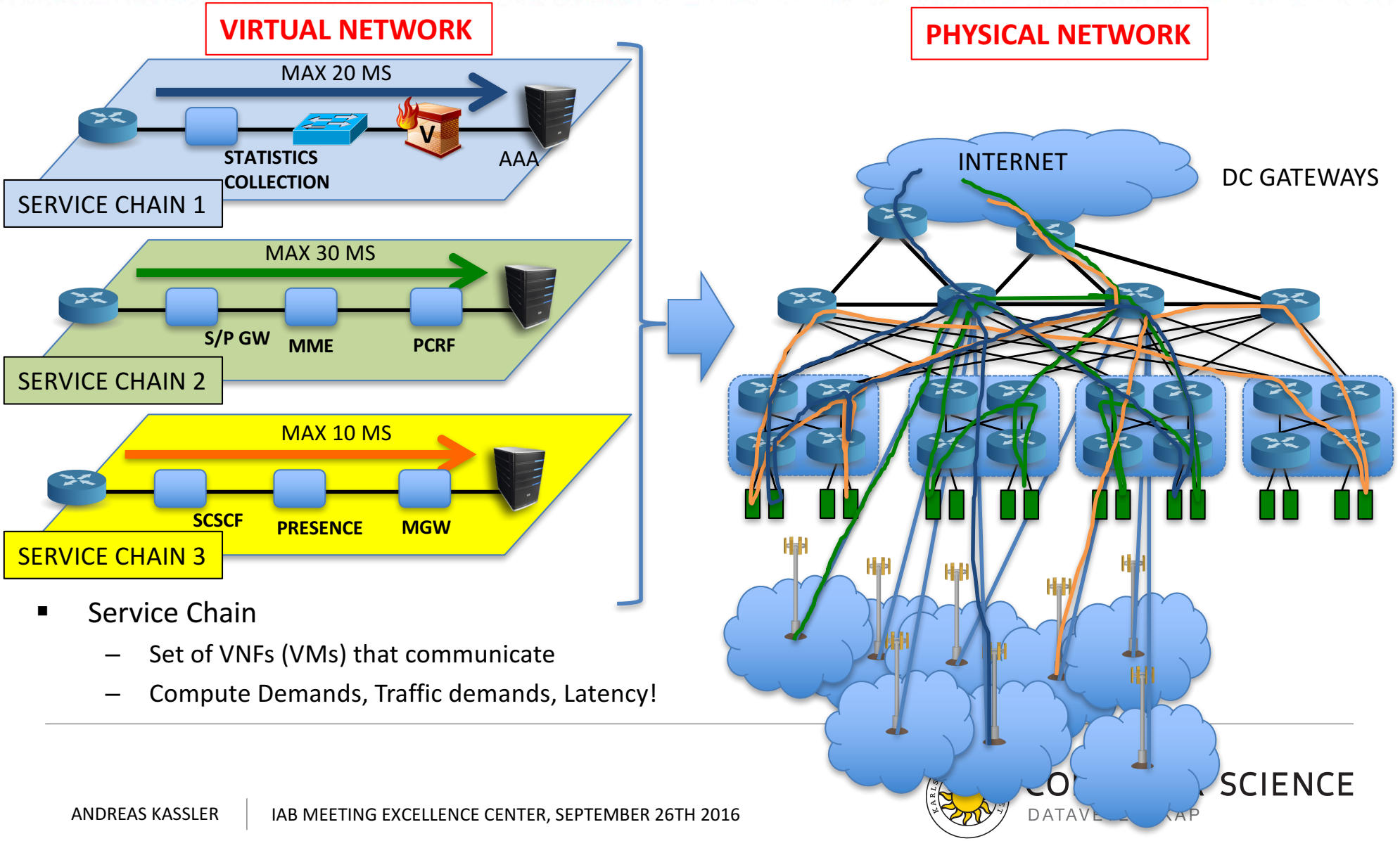
5G EXAMPLE DEPLOYMENT FOR C-RAN





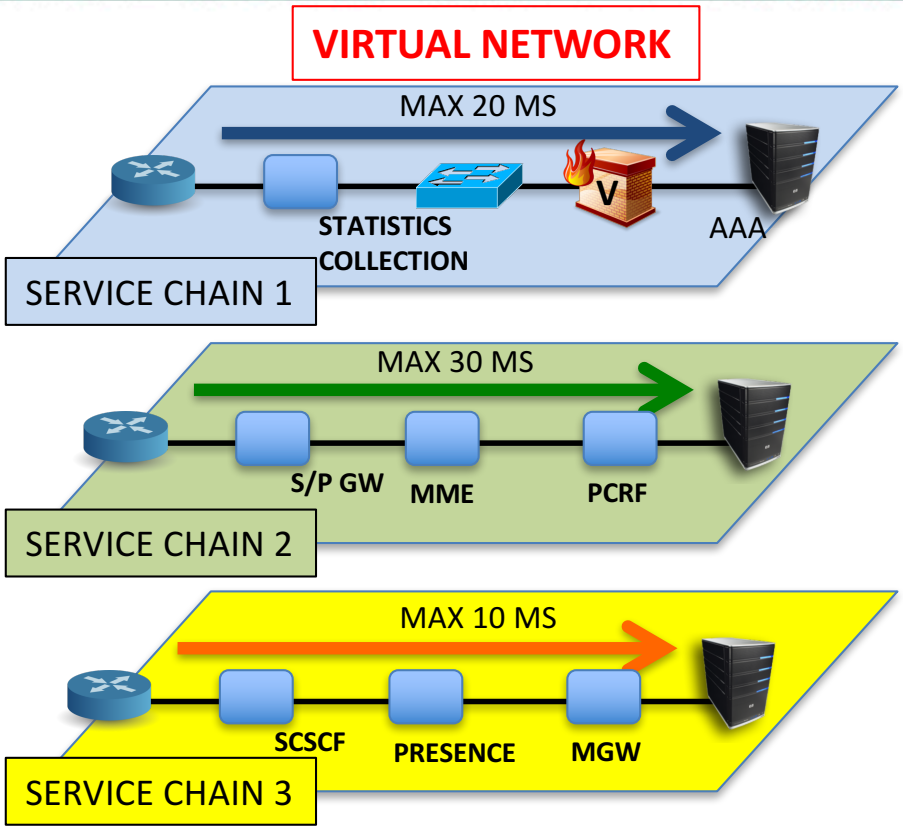
- Local Cloud, Remote Cloud, where to place?

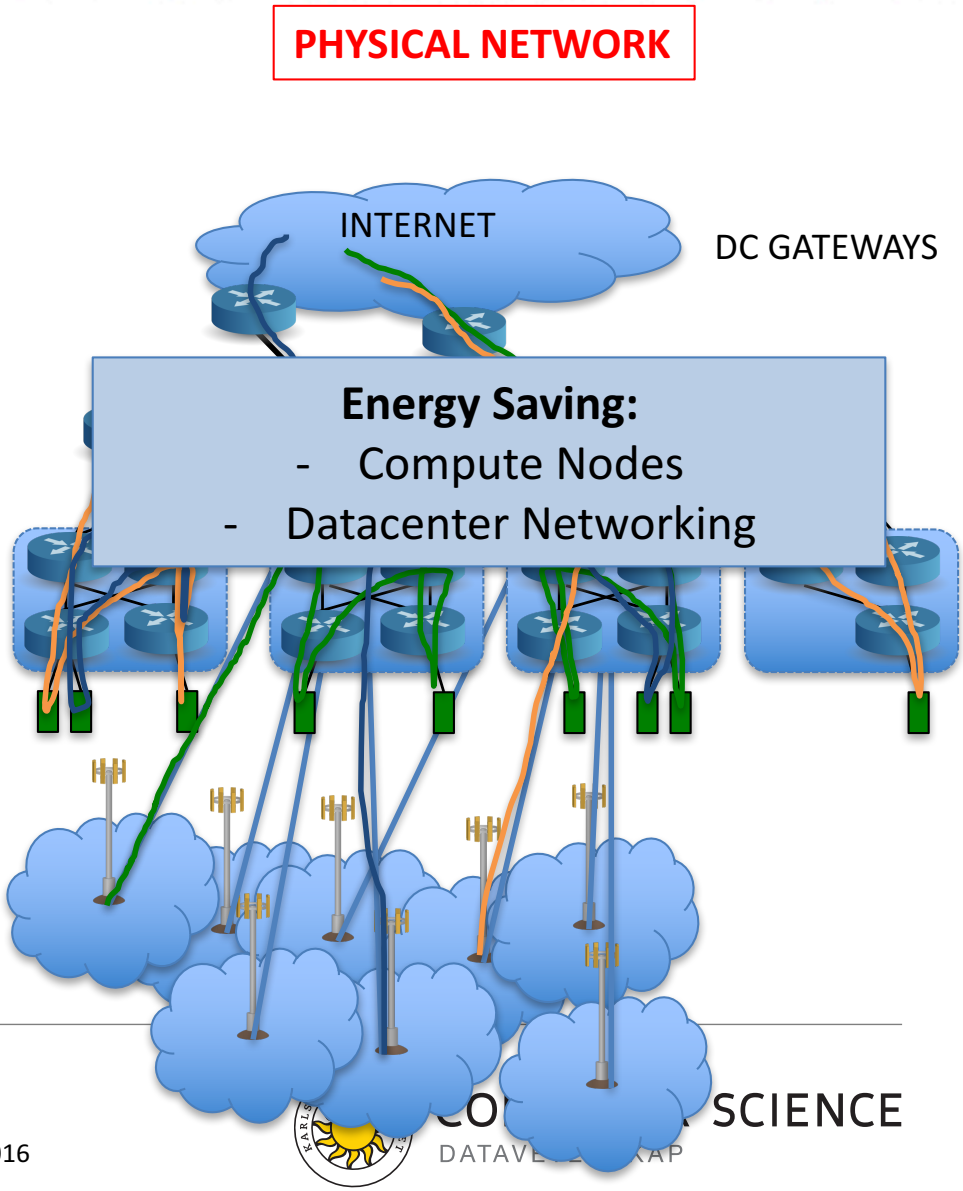
GREEN NFV PLACEMENT



- Service Chain
  - Set of VNFs (VMs) that communicate
  - Compute Demands, Traffic demands, Latency!

GREEN NFV PLACEMENT

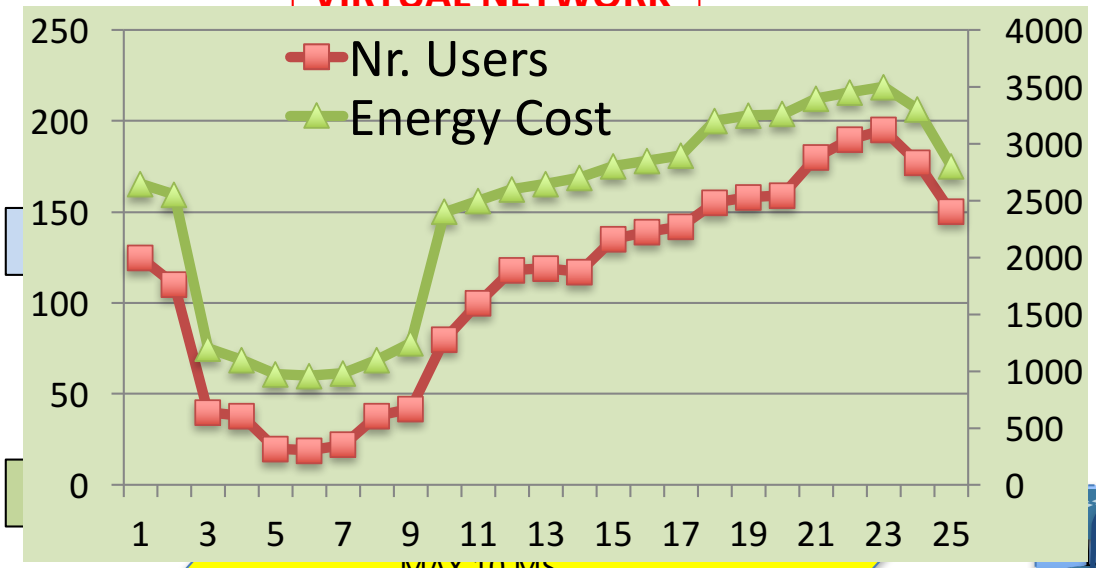




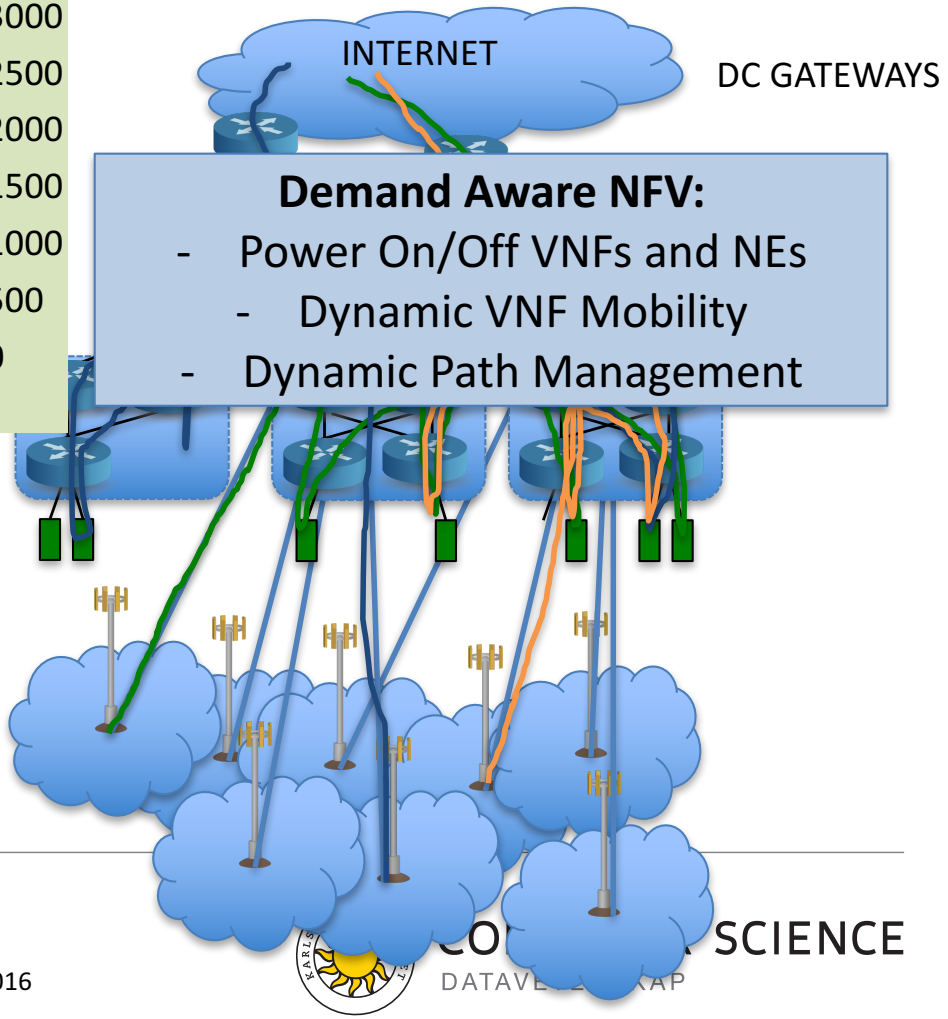
- Virtualized Environment + SDN
  - Seamless VM Mobility
  - Dynamic Path Computation

GREEN NFV PLACEMENT

VIRTUAL NETWORK



PHYSICAL NETWORK



- Virtualized Environment + SDN
  - Seamless VM Mobility
  - Dynamic Path Computation

MODEL TERMINOLOGY

- ***Virtual Network Infrastructure → Virtual Network Function →Virtual Network Function Components***

- Each VNFC can be run on a specific Virtual Machine (VM)
- VNF is modeled a family of Service Chains, which may be generically considered as groups of VNFCs exchanging traffic

  - E.g. One for the User Plane (user data)
  - E.g. One for the Control Plane (handover, signaling traffic...)

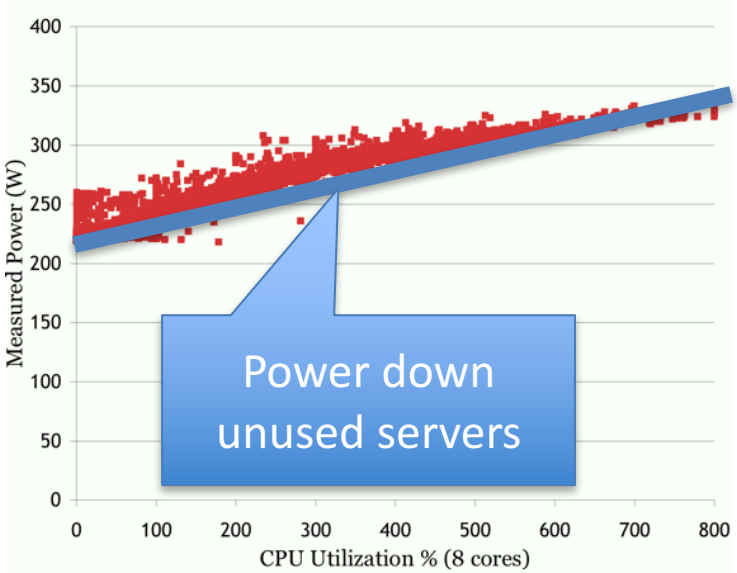
- Each group can be seen as a graph composed of VMs with a set of traffic demands, latencies constraints and resources demands

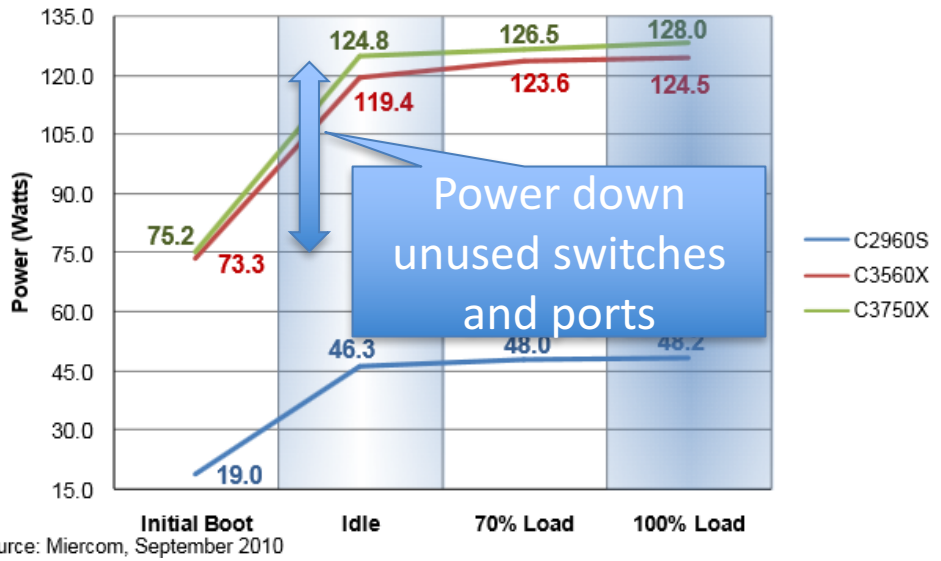
POWER EFFICIENT ROBUST VNFS PLACEMENT PROBLEM

- Power consumption depends on resource demands of VNFs
  - Given the amount of resources available at each server, the amount of CPU and RAM requirements for the VNF components, the power profile of the servers and switches, the problem consists in finding the placement for the VNFs and jointly the traffic flows routing that minimize the total power consumption due to the active servers, switches and links, given capacity of servers and links and latency bounds is not violated.

MODEL DEFINITION

- Each server and switch has a specified power profile





Source: Miercom, September 2010

MODEL INPUT AND OUTPUT

Input Parameters:

a_{rs}	is the amount of resource r available at server s
a_{vr}	is the amount of resource r requested by VNFC v
P_n	is the static power consumption of node n
P_s^{min}	is the idle power consumption of server s
P_s^{max}	is the maximum power consumption of server s
b^{v_1,v_2}	is the traffic demand between v_1 and v_2
$b_{i,j}$	is the capacity of the link (i,j)
$l_{i,j}$	is the latency of the link (i,j)
$P_{i,j}$	is the power consumption for the link (i,j)
$L_C^{v_1,v_2}$	is the maximum latency which can be tolerated by service chain C

Decision variables:

x_{vs}	is 1 if VNFC v is allocated to server s, 0 otherwise
y_s	is 1 if server s is active, 0 otherwise
z_n	is 1 if node n is active, 0 otherwise
$f_{i,j}^{v_1,v_2}$	is 1 if the traffic demand b^{v_1,v_2} is forwarded on link (i,j)
$w_{i,j}$	is 1 if the link (i,j) is used for transmitting any traffic

Table 1: Model Parameters and Decision Variables

OBJECTIVE FUNCTION

- Linear power model for servers, focus only on CPU
- Consider only active switches and links that serve traffic

$$\min \quad \sum_{s \in S} \left[P_s^{\min} \cdot y_s + (P_s^{\max} - P_s^{\min}) \cdot \frac{1}{a_{rs}} \cdot \sum_{v \in V} a_{vr} \cdot x_{vs} \right] \qquad (1)$$

$$+ \sum_{n \in N} P_n \cdot z_n + \sum_{(i,j) \in L} P_{ij} \cdot w_{ij} \qquad r = CPU \qquad (2)$$

CONSTRAINTS

- Single path, unsplittable flow
- Binary decision $f_{ij}^{v_1,v_2} \in \{0,1\}, \forall\,(i,j) \in L, (v_1,v_2) \in \bigcup_{C \in \{\mathcal{C}\}} C$ variable of flow between two components on a link, traffic cannot be split.
- (3): Each VNFC needs to be put on exactly one server
- (4): if no VNFC is allocated to a server, then server is powered down
- (5): if some VNFC is allocated on a server, it must be powered on

$$\sum_{s \in S} x_{vs} = 1 \quad v \in V \tag{3}$$

$$y_s \leq \sum_{v \in V} x_{vs} \quad s \in S \tag{4}$$

$$x_{vs} \leq y_s \quad s \in S, v \in V \tag{5}$$

CONSTRAINTS

- (6): total allocated resources of all VNFCs per < server capacity
- (7-8): flow conservation constraints
  - LHS: flow balance of a node for data sent for a couple of VNFC of a service chain considering incoming flows and outgoing flows
  - RHS: summation over all servers connected to that Node n. If v1,v2 are not allocated to the servers, then summation =0 and node is transit node with 0 flow balance. Otherwise node is source or sink of flow with positive or negative balance

$$\sum_{v \in V} a_{vr} \cdot x_{vs} \leq a_{rs} \cdot y_s \quad s \in S, r \in R \tag{6}$$

$$\sum_{(n,i) \in L} b^{v_1,v_2} \cdot f_{ni}^{v_1,v_2} - \sum_{(i,n) \in L} b^{v_1,v_2} \cdot f_{in}^{v_1,v_2} = \tag{7}$$

$$\sum_{s \in S:\, n(s)=n} b^{v_1,v_2} \cdot (x_{v_1 s} - x_{v_2 s}) \quad n \in N,\ (v_1, v_2) \in \bigcup_{C \in \{\mathcal{C}\}} C \tag{8}$$

CONSTRAINTS

- (9): capacity constraint over all the flows for each link
- (10-13): together with (9): if traffic is sent over link, it need to be activated. If a link is activated, need to activate a network node
- (12-13) links the flow variables to the node activation

$$\sum_{(v_1,v_2)\in\bigcup_{C\in\{C\}} C} b^{v_1,v_2} f_{ij}^{v_1,v_2} \leq B_{ij}\, w_{ij} \quad (i,j) \in L \tag{9}$$

$$w_{ij} \leq z_i \quad (i,j) \in L \tag{10}$$

$$w_{ij} \leq z_j \quad (i,j) \in L \tag{11}$$

$$f_{ij}^{v_1,v_2} \leq z_i \quad (i,j) \in L \tag{12}$$

$$f_{ij}^{v_1,v_2} \leq z_j \quad (i,j) \in L \tag{13}$$

CONSTRAINTS

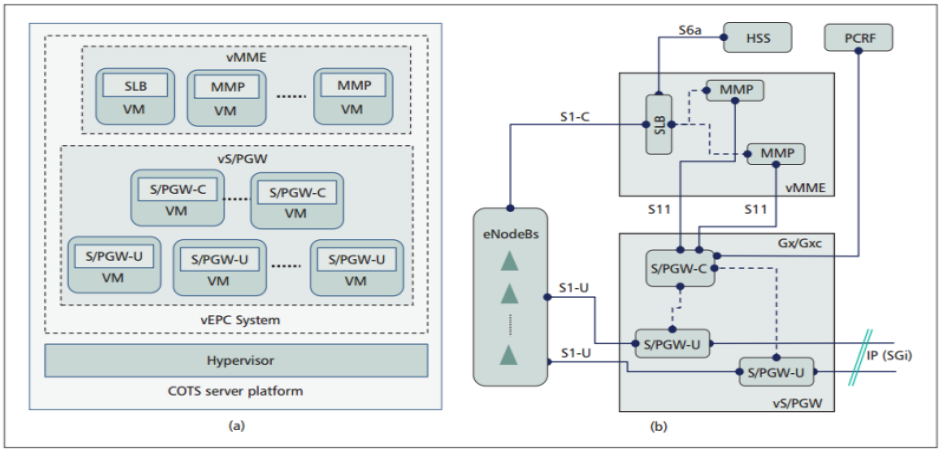
- (14): Latency requirement: For each service chain and component pair, summation of the latencies over the links used for sending data must respect the latency limit
- Even if two service chains share same VNFComponents, the traffic cannot be routed differently for each service chain. Traffic is exchanged between components and all the traffic between two components follows the same path

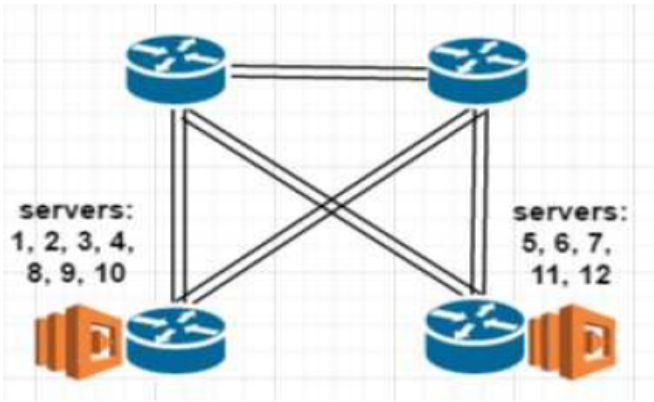
$$\sum_{(i,j)\in L} l_{ij} \cdot f_{ij}^{v_1,v_2} \leq L_C^{v_1,v_2} \quad C \in \mathcal{C}, (v_1, v_2) \in C \tag{14}$$

$$x_{vs} = \{0,1\}, y_s = \{0,1\}, z_n = \{0,1\}, f_{i,j}^{v_1,v_2} = \{0,1\}, w_{i,j} = \{0,1\}$$

EXAMPLE EXPERIMENTAL SCENARIO

- EPCaaS (Evolved Packet Core as a Service)
- Core part of the next generation mobile networks





Parameter	Values					
$\Delta r_{i,m}$	10%, 20%, 30% of the actual demand					
CP Load	500000 events per hour					
Γ	From 0 with step 2 till the total number of VMs					
Scenario	19VMs	4 Nodes	10Links	16 SC	12Servers	27 Demands

EXAMPLE INPUT FILE:

SOME TIPS

- Fast execution is important for real world deployment inside the Cloud
- Performance after 60, 300, 900, 3600 sec. interesting to compare
- For the more advanced: Can parallelize Many Metaheuristics and run it distributed on several solver cores/VMs using Map/Reduce on Hadoop (Amazon) or Apache Spark
  - https://arxiv.org/pdf/1312.0086.pdf
  - https://github.com/deib-polimi/hyperspark
- Can develop interesting Tabu search approach based on how the problem is structured
- Optaplanner can be interesting to look into
- We implemented Greedy 3 phase algorithm
- Genetic Algorithm for VM placement (no network) presented at IEEE CloudNet in 2016


thank you
danke
рахмат
謝謝
ngiyabonga
teşekkür ederim
gracias
merci
obrigado
спасибо
Баярлалаа
bedankt
dziękuję
go raibh maith agat
mochchakkeram
sukriya
kop khun krap
terima kasih
grazie
arigatô
dakujem
takk
tapadh leat
dank je
mersi
barka
welalin
tack
spas
kia ora
vinaka
blagodaram
misaotra
matondo
paldies
grazzi
mahalo
хвала
asante
manana
obrigada
tenki
murakoze
chokrane
mamnun
дякую
djiere dieuf
tau
faafetai lava
kiitos
dankie
dhanyavad
nandri
nanni
hvala
maururu
köszönöm
bayarlalaa
gracie
enkosi
chnorakaloutioun
gratias ago
gracies
sulpay
taiku
akun dankon aciu
sobodi
dėkuji
mèsi
sagolun
najis tuke
ありがとう
tanemirt
rahmet
trugarez
shukriya
мерси
merce
dhanyavadagalu
diolch
xiexie
ευχαριστώ
감사합니다
rahmat
kam sah hamnida
didi madloba
спасибі