UNIVERSITY OF MINNESOTA

Network Support for Emerging Applications:
Flexible Network Services as Frameworks?

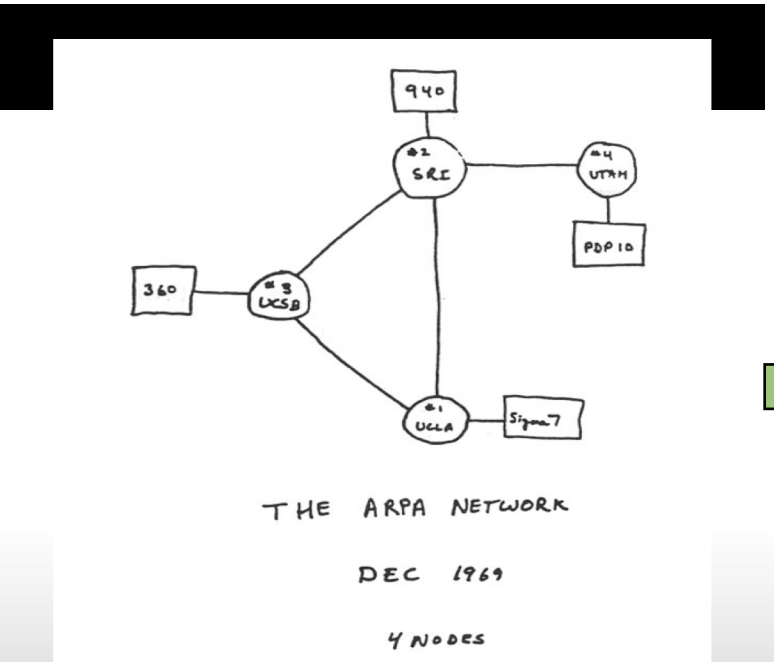
Zhi-Li Zhang

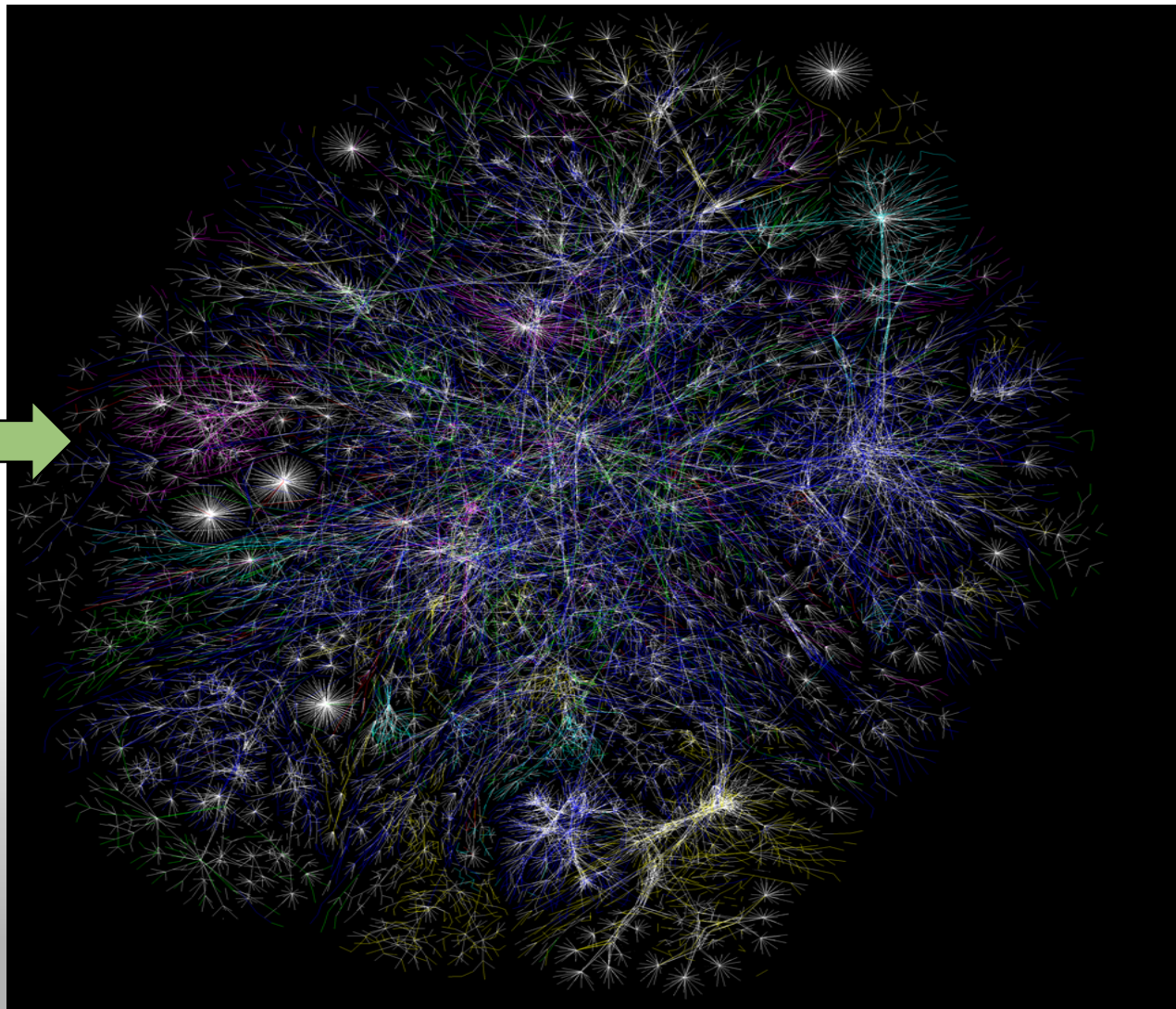
Qwest Chair Professor &
Distinguished McKnight University Professor

Dept. of Computer Science & Eng.,
University of Minnesota

Internet: A Huge Success Story

- From the original four-node ARPNET research experiment to today's global information infrastructure





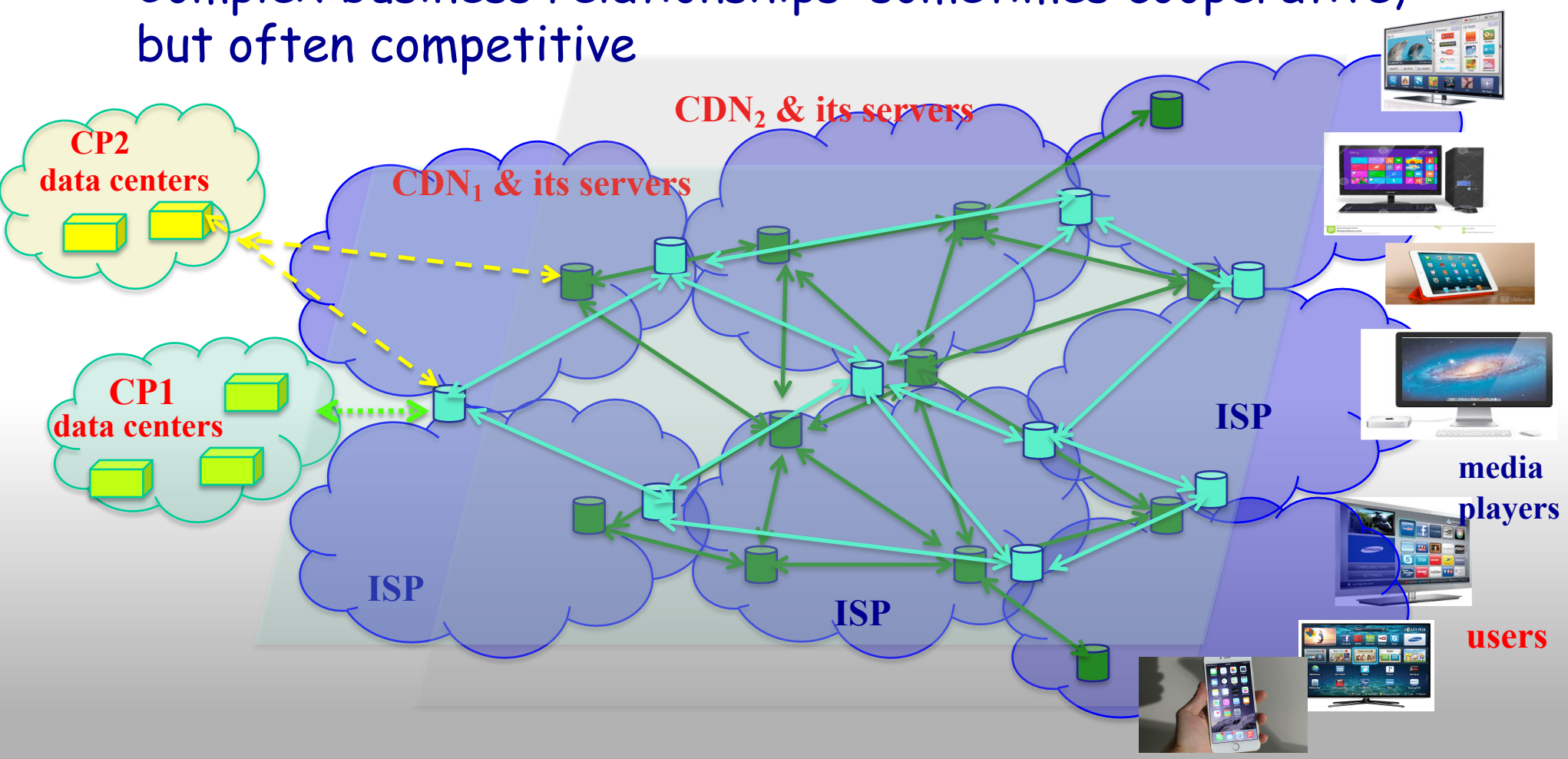
Success of Today's Internet

Today's Internet can be primarily characterized by its success as a (*human-centric, content-oriented*) information delivery platform

- Web access, search engine, e-commerce, social networking, multimedia (music/video) streaming, cloud storage, …
  - users search for and interact with websites (or "content"), or with other users;
  - users consume or generate information
  - *static* vs. *dynamic* content
- Rise of web (and HTTP) – coupled with emergence of mobile technologies – led to cloud computing and CDNs
  - Huge data centers with massive compute and storage capacities to store information, process user requests and generate content they desire
  - CDNs with geographically distributed edge servers to "scale out" and facilitate "speedier" information delivery

Content Distribution Ecosystem

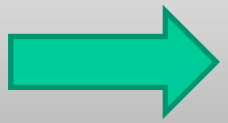
- Multiple major entities involved!
  - content providers (CPs), content distribution networks (CDNs), ISPs and of course, end systems & users
  - some entities may assume multiple roles
- Complex business relationships: sometimes cooperative, but often competitive



(Video) Content is the King

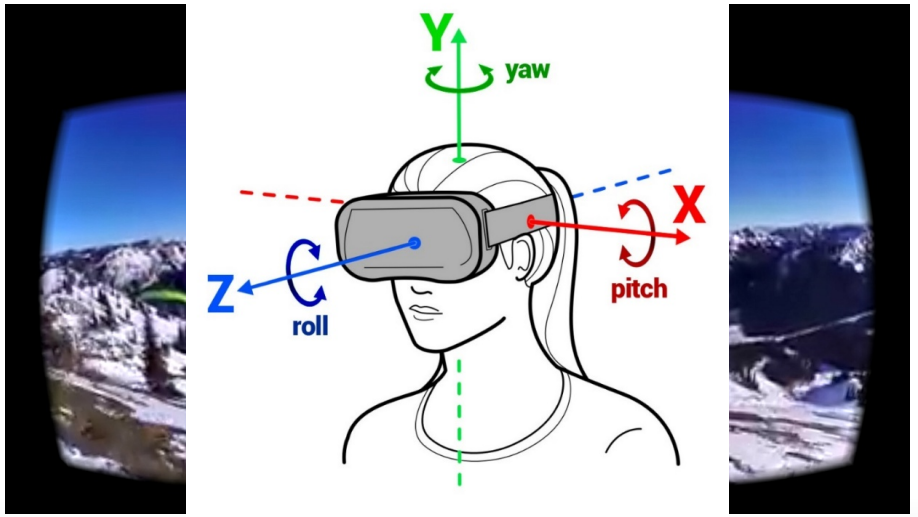
- Video dominates Internet traffic today
  - based on some projections, video up to 80% of Internet traffic

- Rise of (user generated) ultra-short (mobile) videos popularized by TikTok



But Video Content is Also Evolving

- From SD/HD to 4K/8K to 360 to volumetric & AR/VR
  -- "holographic media"? Hydrogen phone by RED?
  - ➢ not only huge bandwidth requirement
  - ➢ but also support for interactivity (thus low latency, jitter, …)



Traditional Videos
2D
0-DoF (degree-of-freedom)

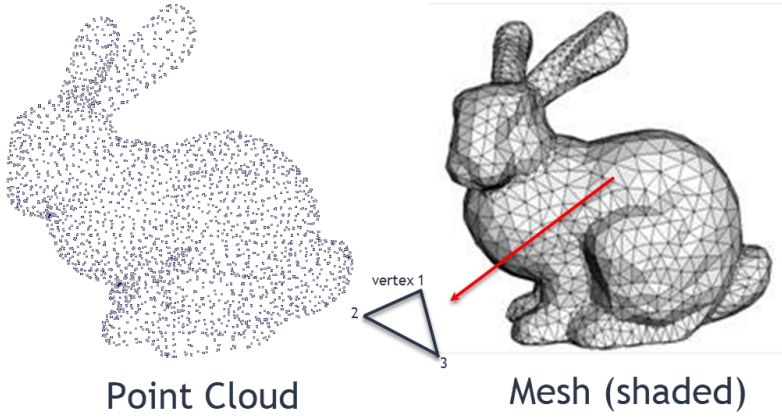
360° Videos
2D (Spherical)
3-DoF

Video source: https://www.youtube.com/watch?v=aO3TAke7_MI

Slides courtesy of Feng Qian

Volumetric Video

- From SD/HD to 4K/8K to 360 to volumetric video & AR/VR
  - ➢ not only huge bandwidth requirement
  - ➢ but also support for interactivity (thus low latency)



3D point cloud or mesh

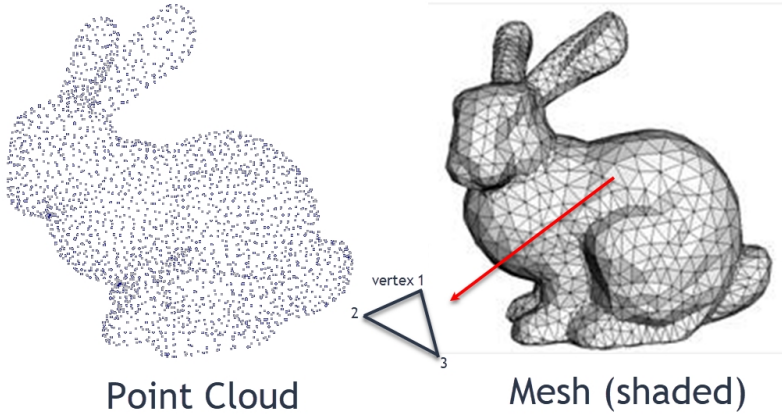
6-DoF

- Captured by RGB-**D** cameras with **D**epth sensors
- Immersive telepresence experience

Slides courtesy of Feng Qian

Volumetric Video & Streaming

- From SD/HD to 4K/8K to volumetric video & AR/VR
  - not only huge bandwidth requirement
  - but also support for interactivity (thus low latency)



- Rendering uncompressed video is fast
  - using Samsung Galaxy S8
  - on-device GPU

Avg # Points	Render FPS
12.6K	1111
25.2K	776
50.4K	352
75.5K	233
100.7K	173

3D point cloud or mesh

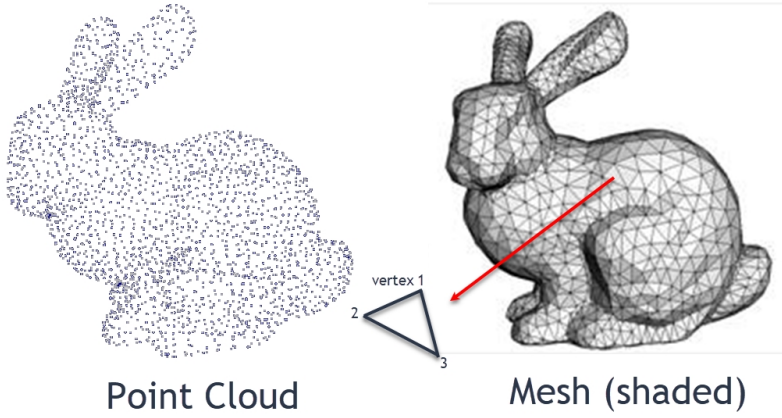
6-DoF

- Captured by RGB-**D** cameras with **D**epth sensors
- Immersive telepresence experience

Slides courtesy of Feng Qian

Volumetric Video & Streaming

- From SD/HD to 4K/8K to volumetric video & AR/VR
  - not only huge bandwidth requirement
  - but also support for interactivity (thus low latency)
- Rendering uncompressed video is fast
  - using Samsung Galaxy S8
  - on device GPU



➢ But requires a lot of bandwidth!

Points / Frame	Frame Size	Bitrate at 24 FPS
12.6K	0.11MB	21 Mbps
25.2K	0.23MB	43 Mbps
50.4K	0.45MB	86 Mbps
75.5K	0.68MB	172 Mbps
100.7K	0.91MB	344 Mbps

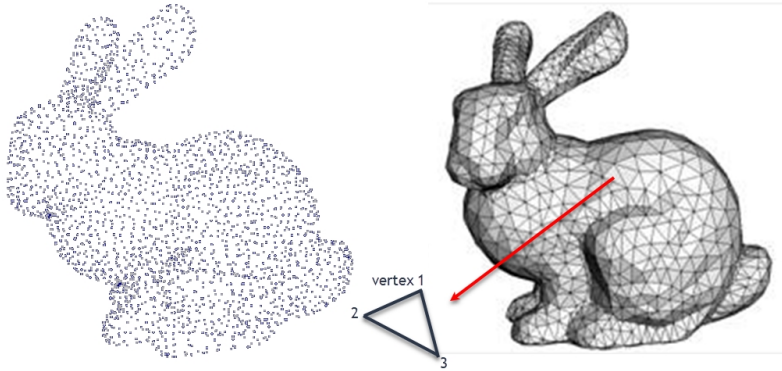
9 bytes/point * 50K points/frame * 24 FPS * 8

60 FPS: up to 1Gbps

Slides courtesy of Feng Qian

Volumetric Video & Streaming

- From SD/HD to 4K/8K to volumetric video & AR/VR
  - ➢ not only huge bandwidth requirement
  - ➢ but also support for interactivity (thus low latency)



Point Cloud Mesh (shaded)

- Decoding on today's COTS smartphones is challenging!
  - performance of decoding using a single CPU core is poor
  - multi-cores provide limited performance gains

Avg # Points	Decoding FPS
12.6K	14
25.2K	7
50.4K	4
75.5K	2
100.7K	1

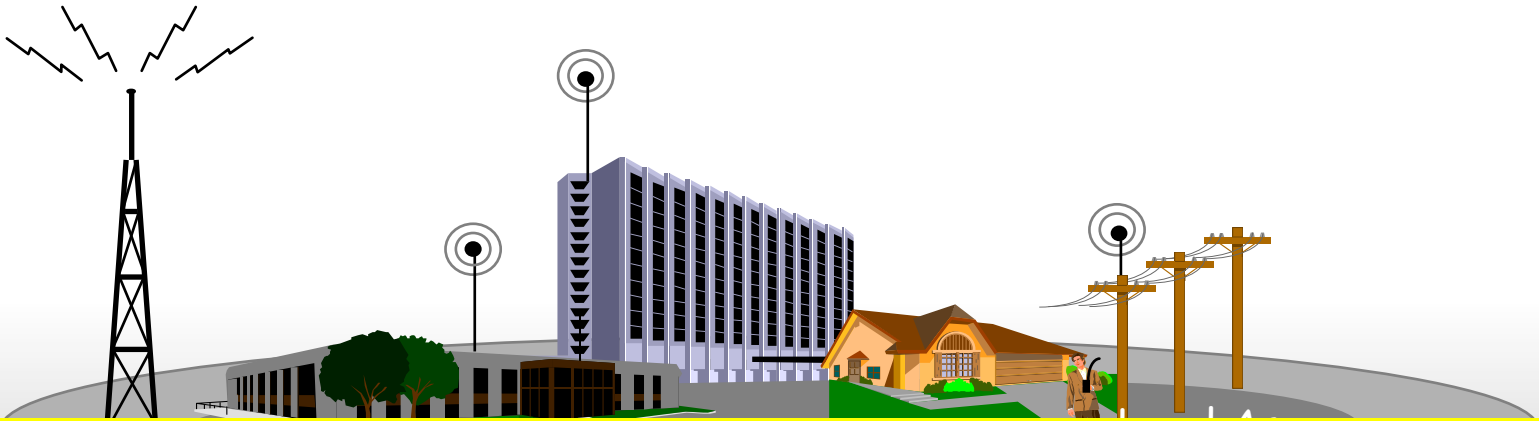
encoding/decoding alg: octree (state-of-art)

Role of Edge Computing: "in-network" processing using commodity servers at the network edge?

Slides courtesy of Feng Qian

More gadgets are plugged in …

- servers, desktops, laptops, …
- smart mobile phones, iPads, e-readers, …
- smart TV, cameras, AR/VR goggles, …
- *now lightbulbs, thermostats, car, etc., soon toasters, fridges, …* ☺



thanks in large part to innovations in wireless technologies
WiFi, bluetooth, NFC, Zigbee, 3/4G cellular networks, now 5G, …

On The Horizon:
Internet of Things (IoT)

from inter-connections of human users (w/ content) to interconnections of things" (i.e., "IoT"), namely

- from "human-centric" (information generated for & consumed by humans) to "things-centric"
- connecting the cyber space with the physical world

- ➢ Industrial Digitalization
- ➢ Cyber-Physical Systems (CPS)

- smart home, smart building, …
- smart cities & communities, …

On The Horizon:
Internet of Things (IoT)

from inter-connections of human users (w/ content) to interconnections of things" (i.e., "IoT"), namely

- from "human-centric" (information generated for & consumed by humans) to "things-centric"
- connecting the cyber space with the physical world

Many exciting new applications & services are emerging!

- They are more *complex*, with *diverse* requirements
- *Need better support from networks*
  - ➔ relying solely on innovations in end devices/systems & cloud are no long sufficient!

Networks Are Rapidly Evolving Too (esp. in last decade)!

Besides faster NICs, fatter pipes & innovations in wireless

Two Emerging Intertwined Trends Reshaping the Networking Field

- Software-Defined Networking
- Network Function Virtualization

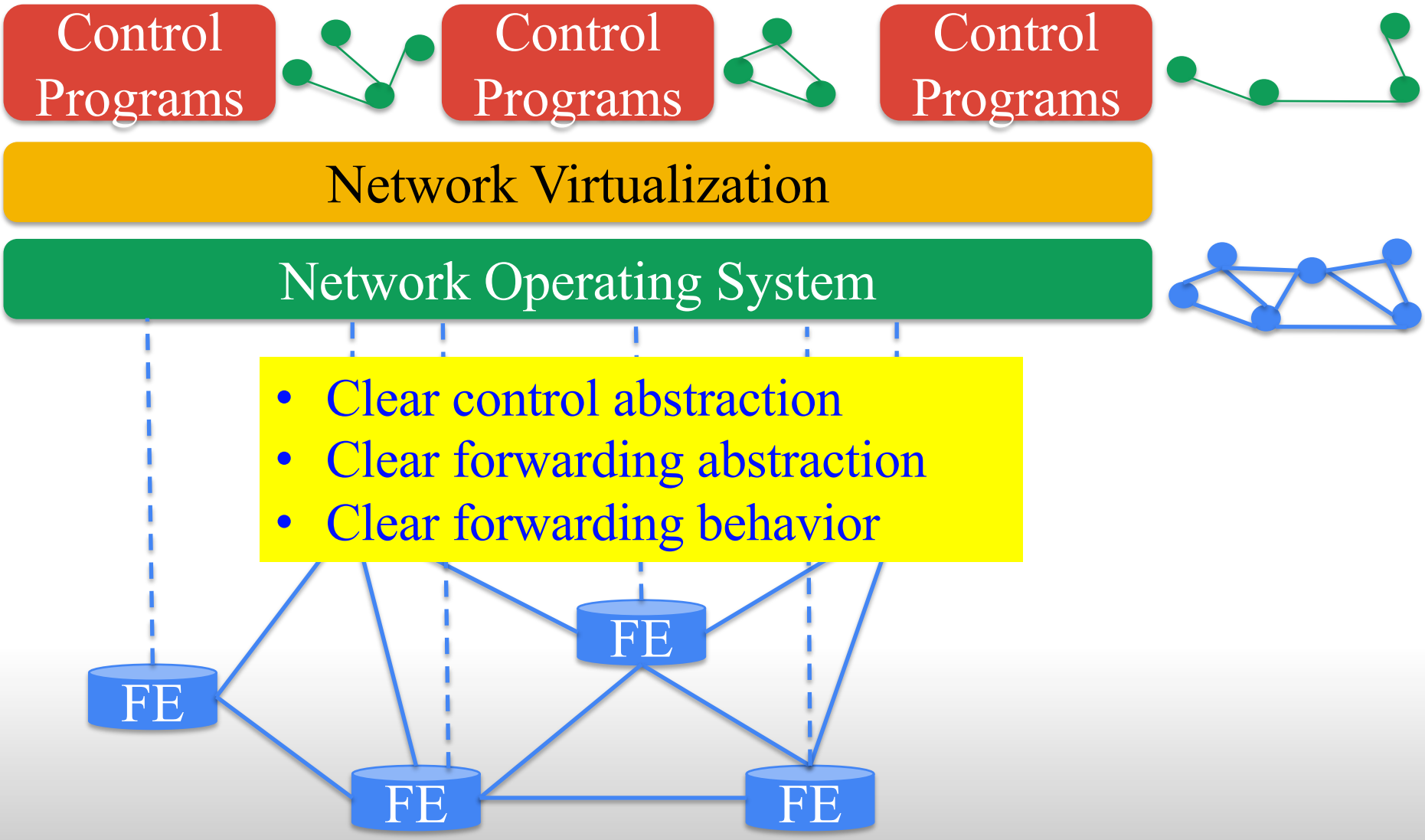
Software Defined Networking

A network in which the control plane is physically separate from the forwarding plane

and

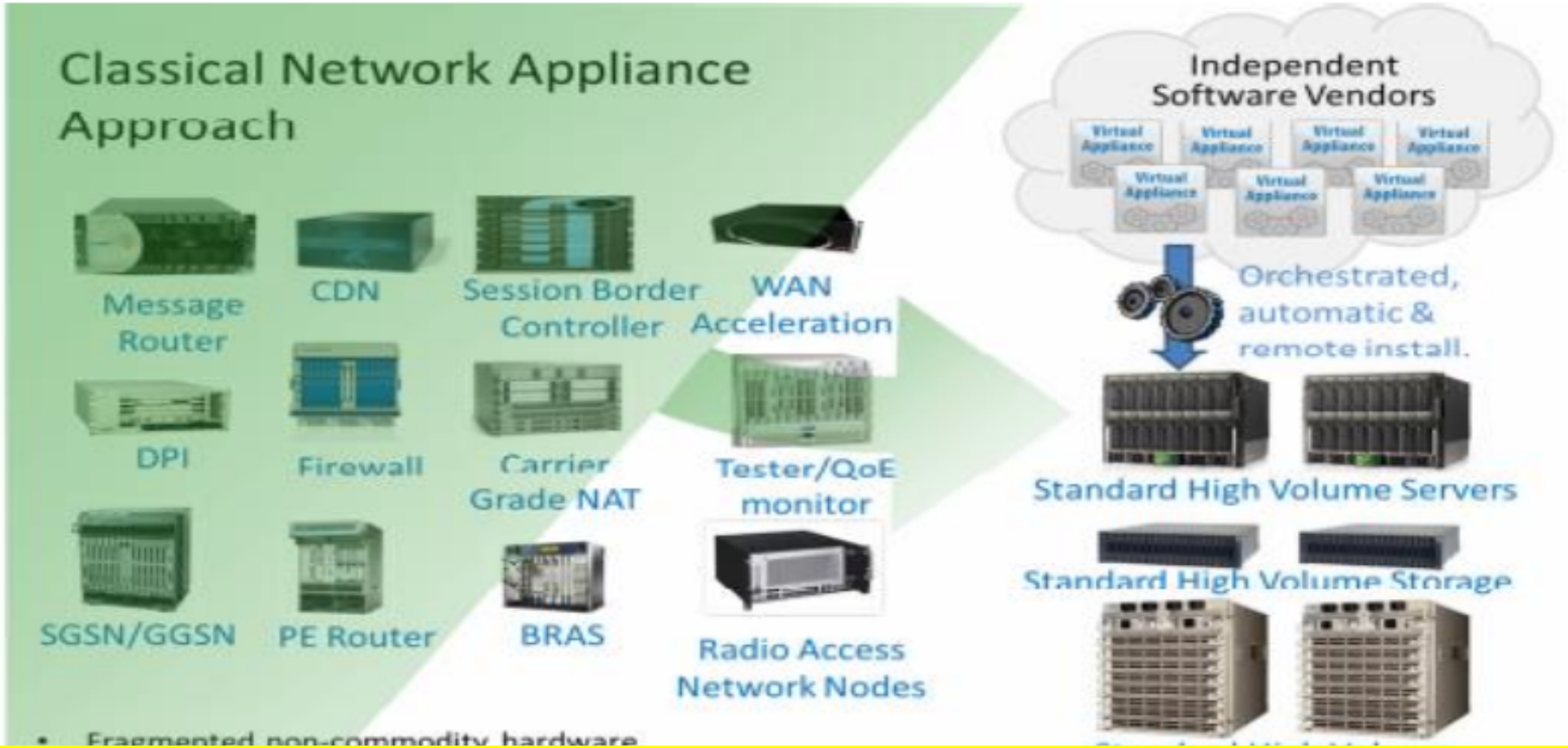
A single (logically centralized) control plane controls several forwarding devices

Software Defined Networking



➢ Enable a more programmable data plane

Network Function Virtualization



Unlike SDN which came out of academia,
NFV initiated by industry, inspired by cloud computing (& DevOps)

- utilizing commodity servers for scalability, availability & velocity
- and partly (and initially) with the goal to reduce CAPEX

Networks Are Changing Too!

Two Emerging Intertwined Trends Reshaping the Networking Field

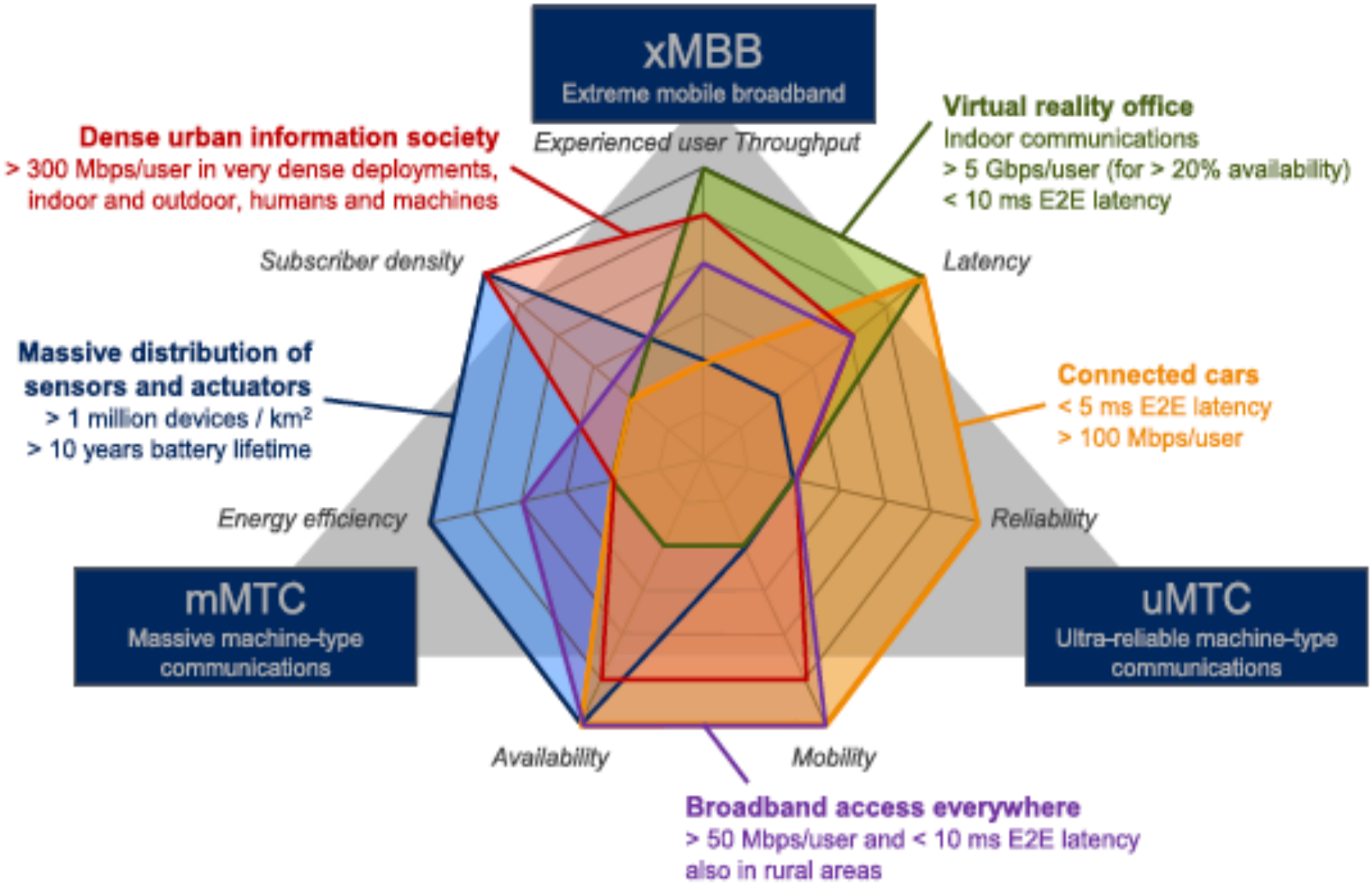
- Software-Defined Networking
- Network Function Virtualization

& Emerging 5G

+ (Mobile) Edge Computing

IoT: Killer App for 5G?

5G: Diverse Services with Very Divergent Requirements



Networks Are Changing Too!

Two Emerging Intertwined Trends Reshaping the Networking Field

- Software-Defined Networking
- Network Function Virtualization

& 5G + (Mobile) Edge Computing

- *They are primarily motivated by the desires & needs of network vendors, operators or ISPs*
- *Not entirely driven by "real challenges" in providing better support for (emerging) applications & services*

Network Support for Applications?

Emerging applications are increasingly diverse and complex

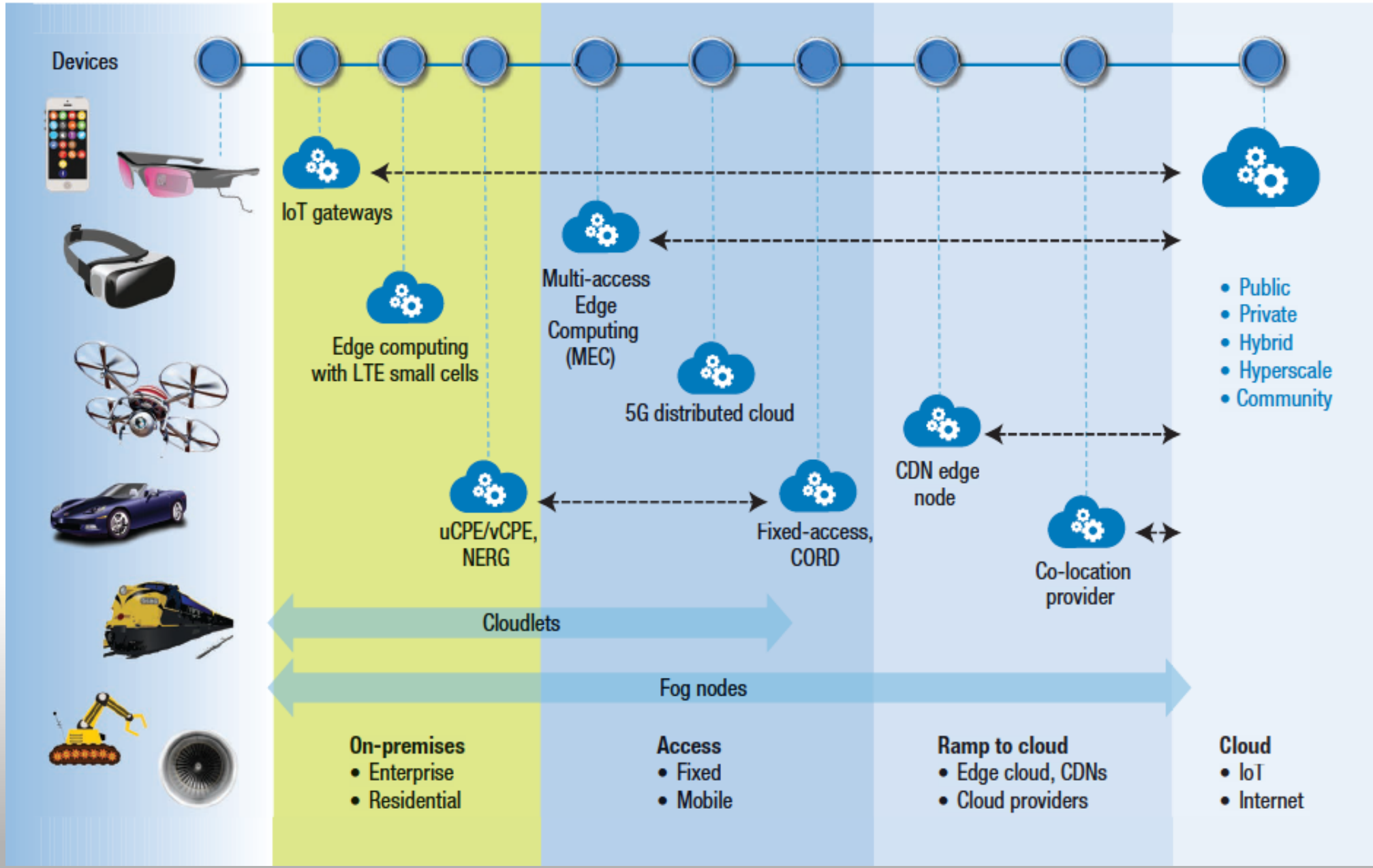
- Vastly different requirements: bandwidth, latency, jitter, …
- Perhaps more importantly, vastly different "semantics"
  - ➢ not all bits are the same & can have different meanings: not all video frames/objects or data streams are equally important or valuable

How can we leverage new networking innovations to provide better support for emerging diverse & complex applications?

- Increasingly programmable data plane (Openflow, P4, whitebox switches, etc.) and "smartNICs" (e.g., DPDK, RDMA, FPGA, NPU, …)
- Virtualized network functions running on commodity servers
- New network control & management paradigms, ……

Networking Becomes Critical

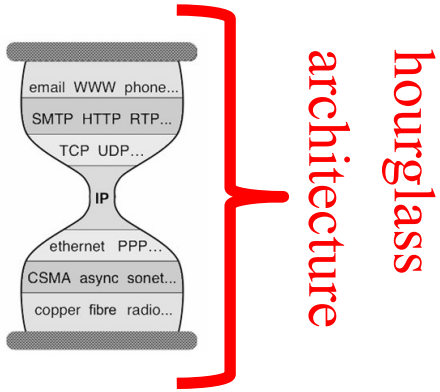
App Support: *from Device to Mobile Edge to Cloud*



Network Support for Applications?

Today's networks largely offer only a "one size fits all" solution

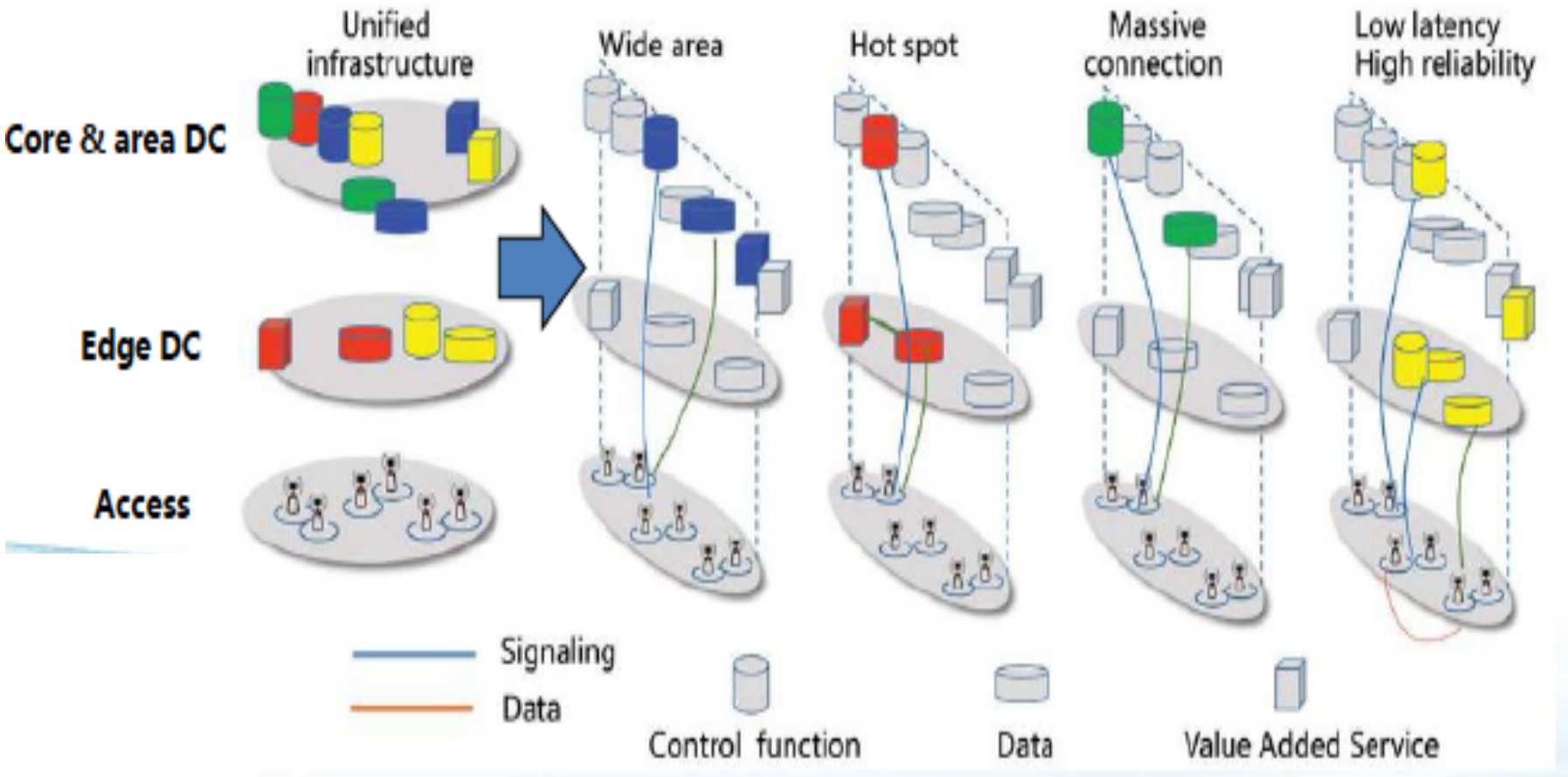
- "best-effort" IP net. service, w/ TCP/UDP transport on end systems
- Networks as a "bag of protocols"
- Apps often build own "communications middleware" with various "high-level" abstractions/semantics
  - mostly built on top of TCP, which suffers many issues
  - a bit more on these later



Emerging applications likely require "end-to-end" and "in-network" support: from end devices to *edge/network* to cloud --

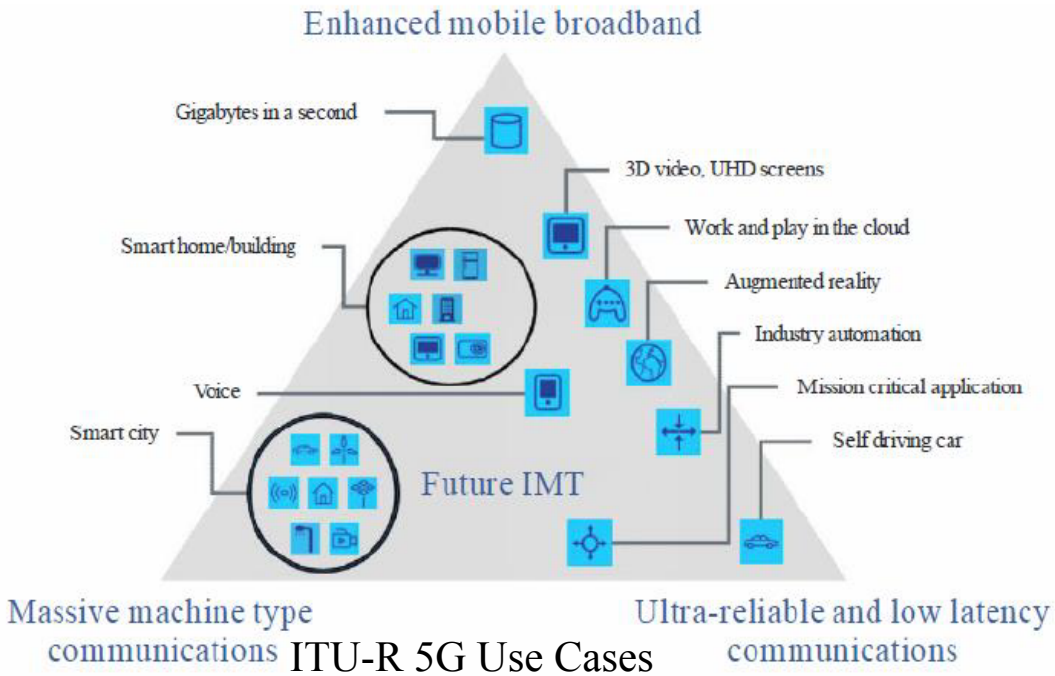
Clearly, need to rethink & redesign "network architectures" !?

Network Slicing ?

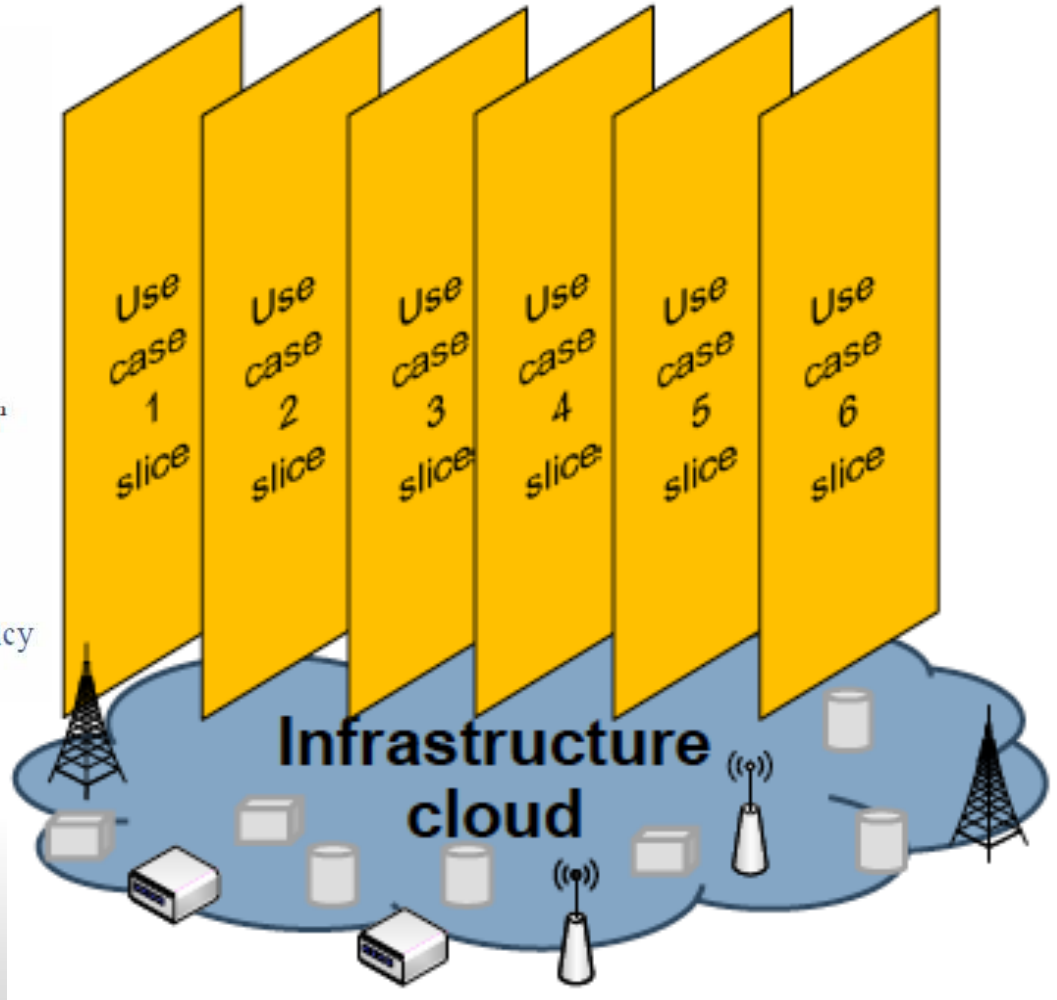


Network slicing is a big buzz word!

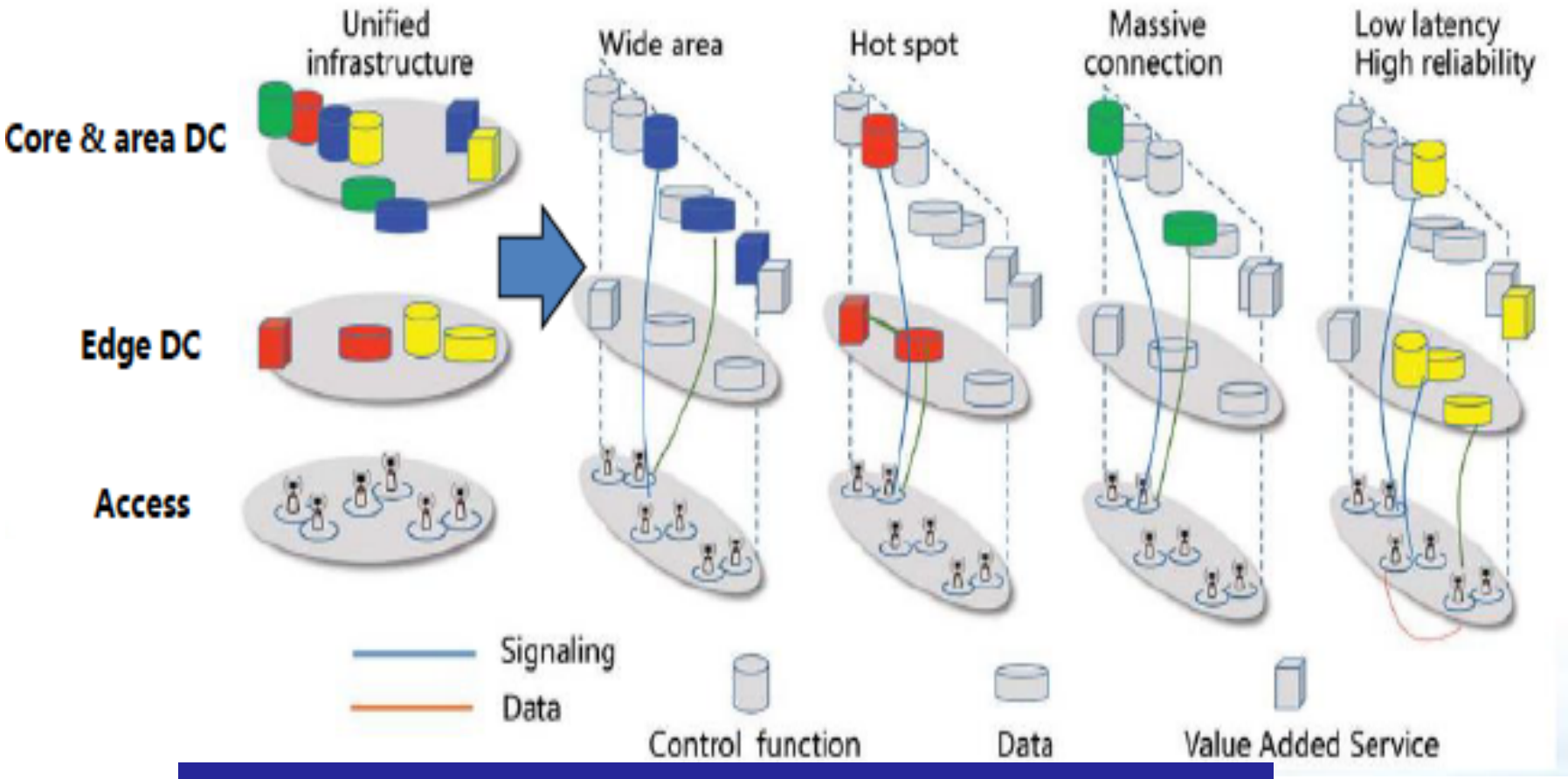
5G (Virtualized) Network Architecture for Supporting Diverse Applications/Services?



ITU-R 5G Use Cases



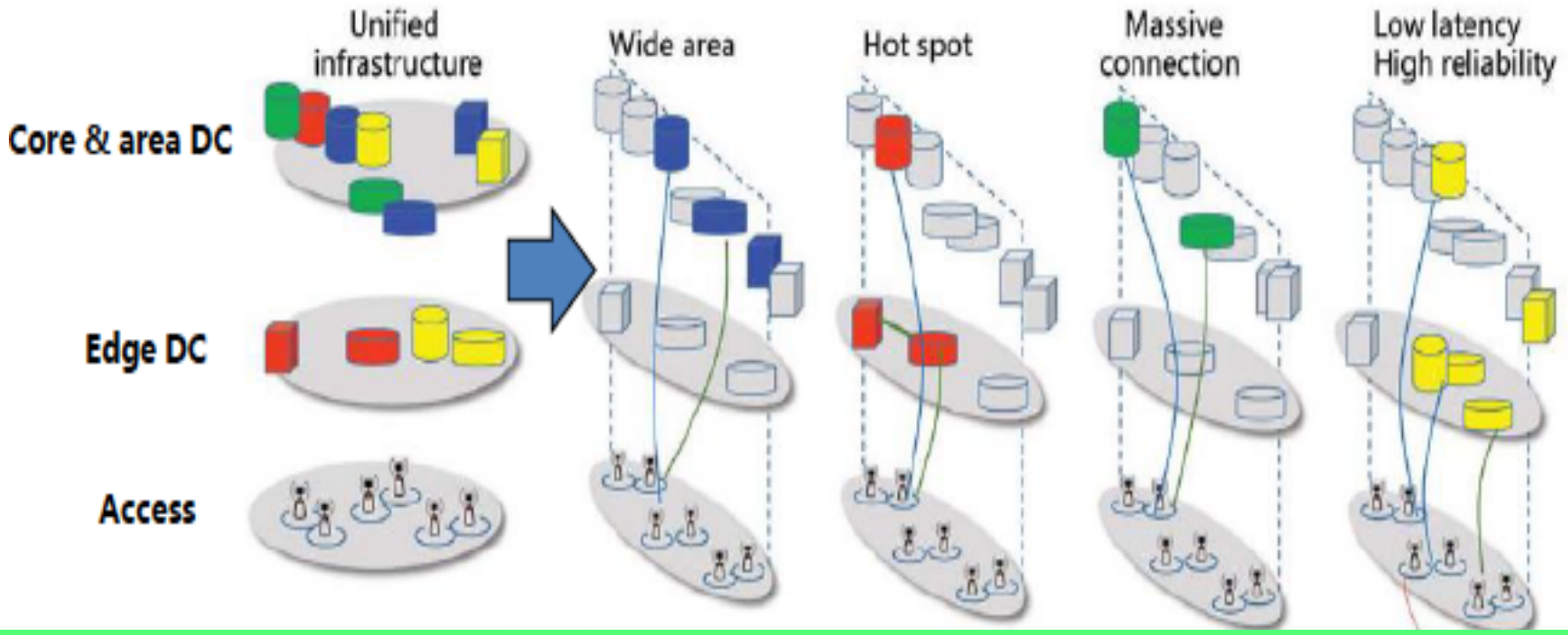
Network Slicing



But How?

- One network slice per service, or per service provider?
- Or per service instance (tenant, user, device, flow, ...)?
- How does it relate to server virtualization (VMs, container, …)?
- How to form an *end-to-end network slice* from various "shards"?

Network Slicing



How to effectively support a single challenging service/application like volumetric (AR/VR) video?

- One network slice per service, or per service provider?
- Or per service instance (tenant, user, device, flow, ...)?
- How does it relate to server virtualization (VMs, container, …)?
- How to form an *end-to-end network slice* from various "shards"?

Network Slicing

How to effectively support a single challenging service/application like volumetric (AR/VR) video?

E.g., can we support 40K fans in a large sport stadium following their (resp.) favorite players using AR/VR?

A Detour

Quickly talk about two pieces of recent work

- Measurement study of real-world 5G deployment
  - *in collaboration w/ Prof. Feng Qian*
- Performance impact of multi-core server architecture on NFV at 100 Gbps line speed & beyond

Main Take-Aways

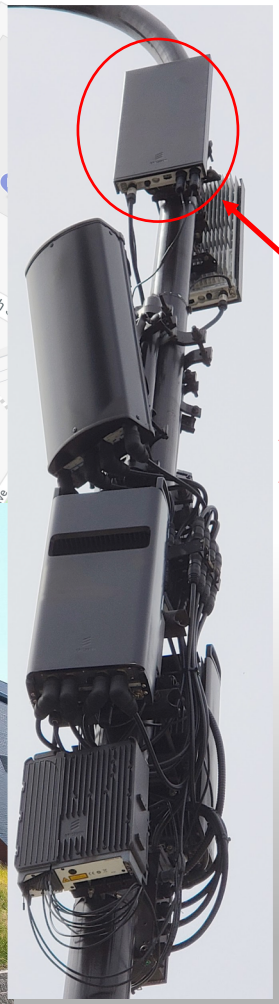
- Yes, 5G has the potential to support exciting new apps!
- Hugh implications on networking/edge computing: a lot of new challenges

➔ leading to & concluding w/ *main theme* of my talk

Commercial 5G Measurement Study

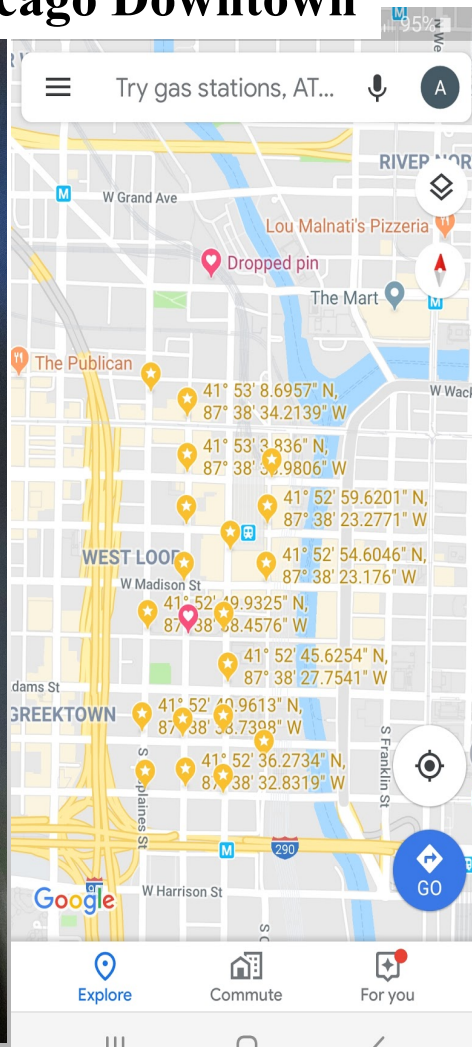
- Verizon deployed 1st commercial (mmWave) 5G in US in downtown Minneapolis & Chicago -- Non-standalone (only 5G-NR), core 4G LTE

Minneapolis Downtown East



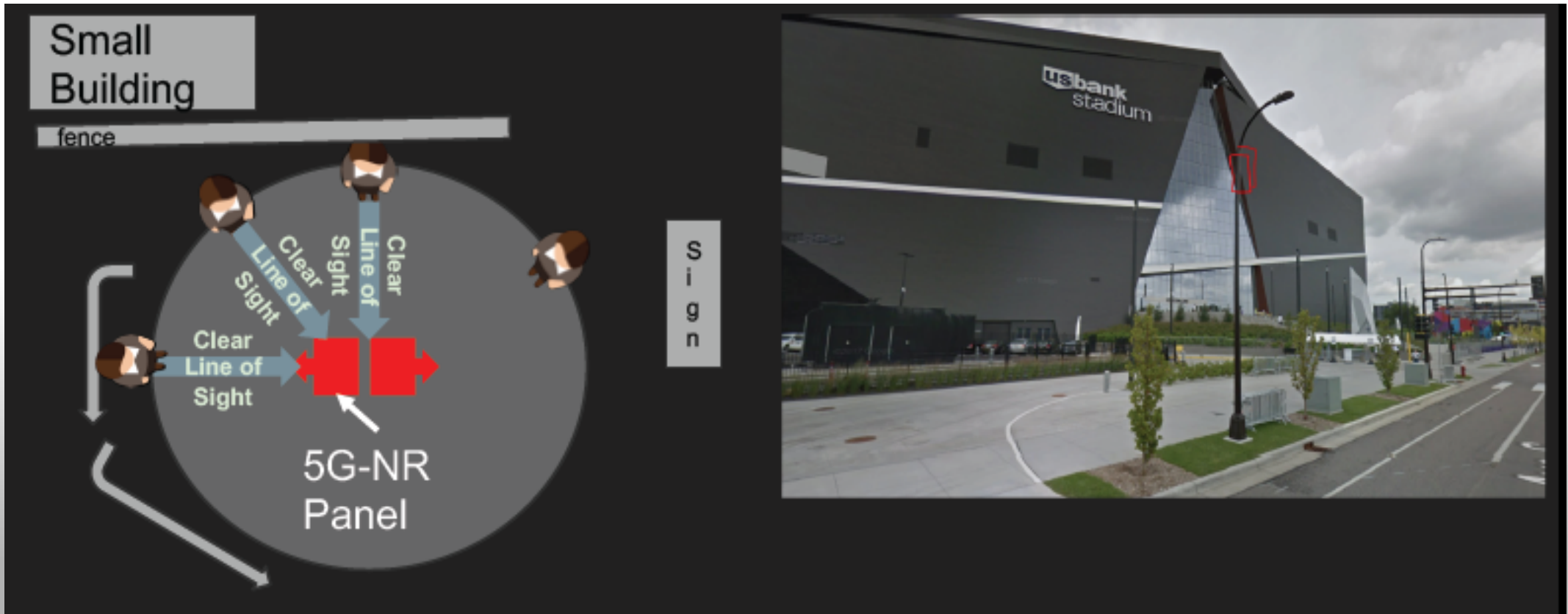
5G-NR panel

Chicago Downtown



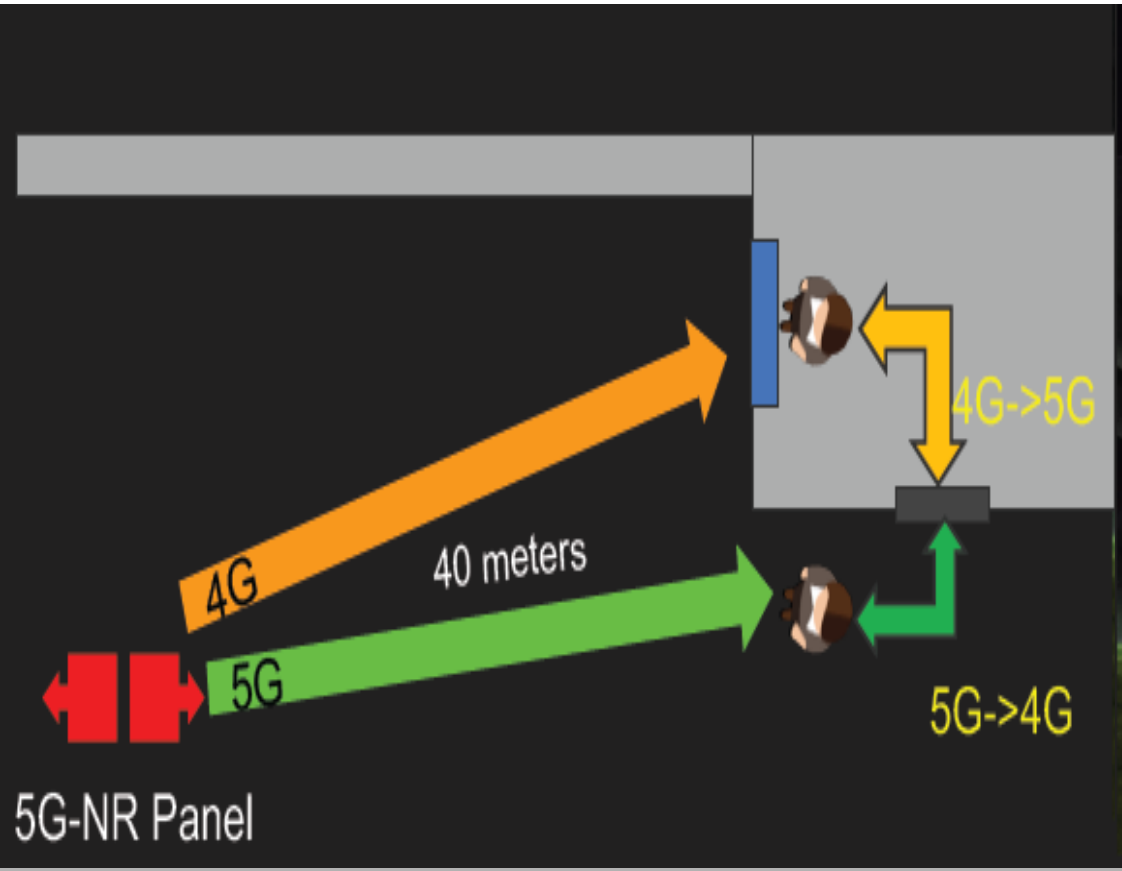
Commercial 5G Measurement Study

- 2 month-long measurement study using Samsung S10 5G handsets
  - orientation tests; varying distance; time of day; etc.
  - line of sight (LOS) vs. with various obstructions; different locations
  - mobility (walking vs. driving) and 5G-4G handoff



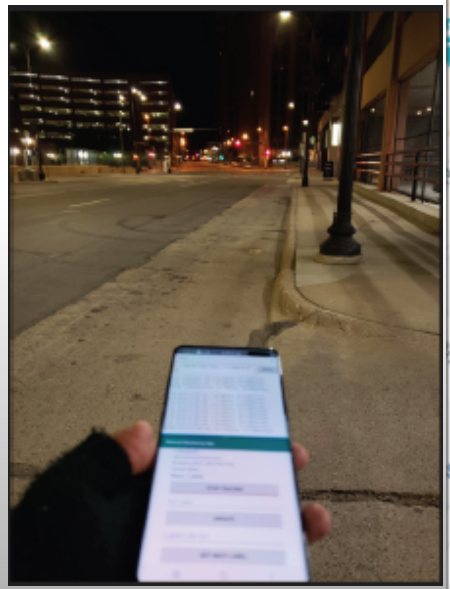
Commercial 5G Measurement Study

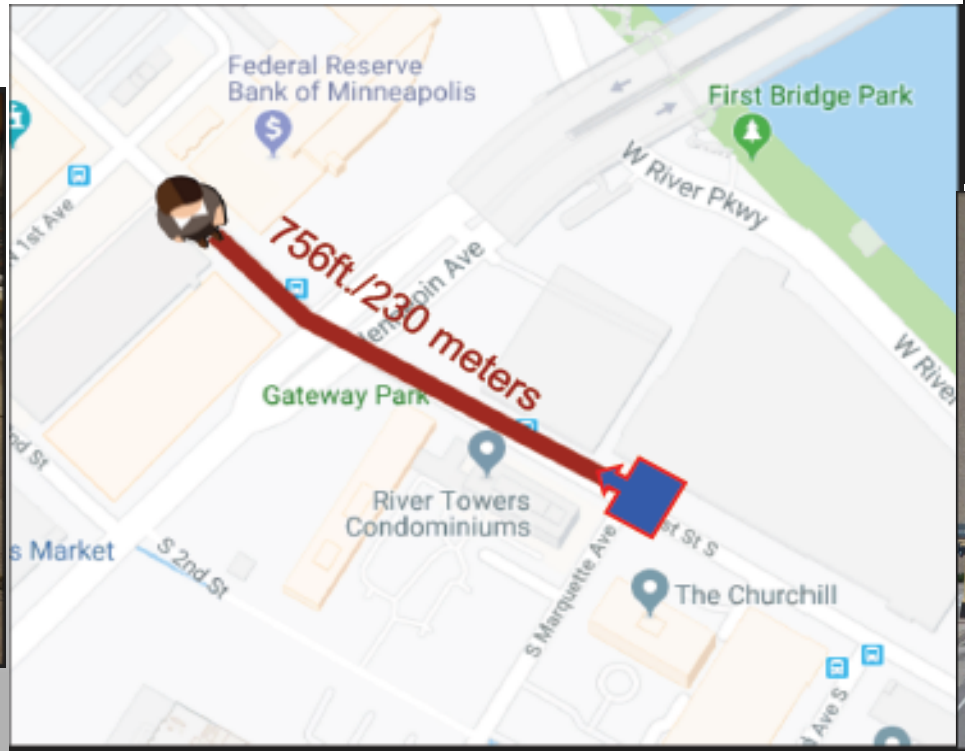
- 2 month-long measurement study using Samsung S10 5G handsets
  - orientation tests; varying distance; time of day; etc.
  - line of sight (LOS) vs. with various obstructions; different locations
  - mobility (walking vs. driving) and 5G-4G handoff



Commercial 5G Measurement Study

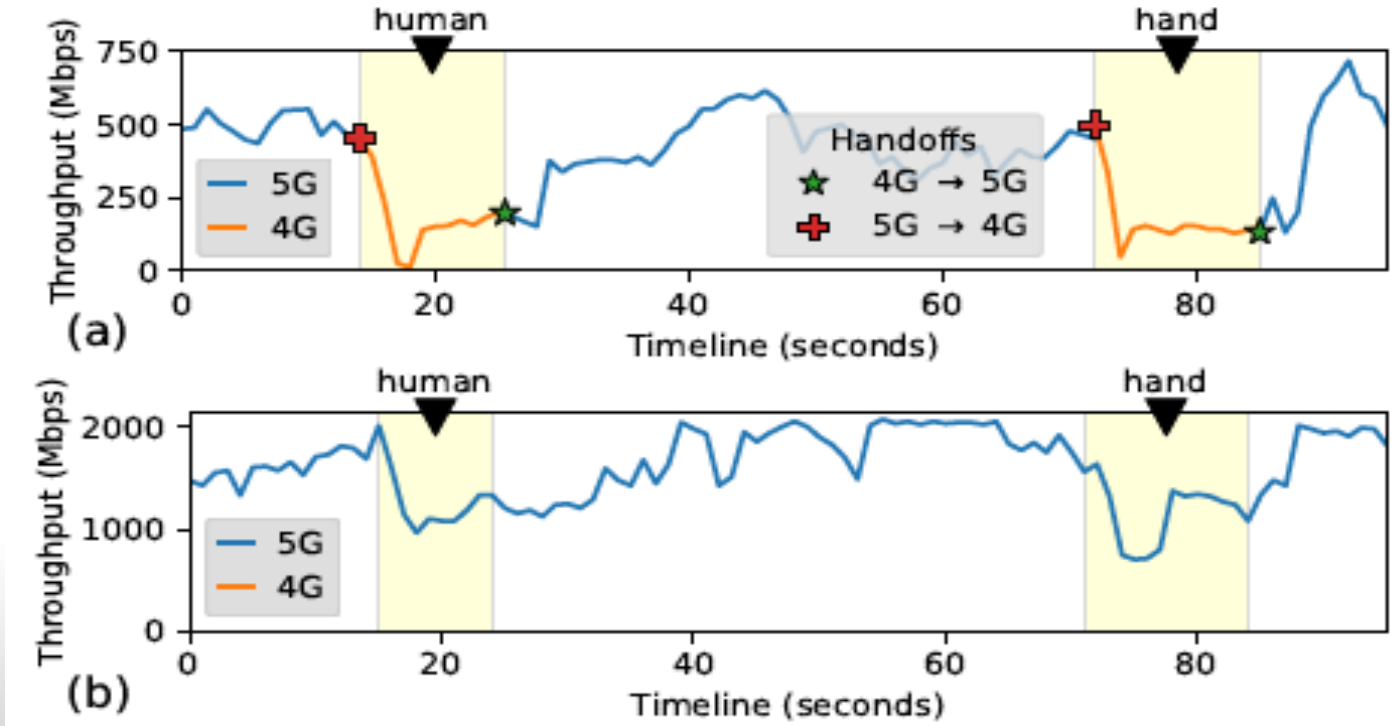
- 2 month-long measurement study using Samsung S10 5G handsets
  - orientation tests; varying distance; time of day; etc.
  - line of sight (LOS) vs. with various obstructions; different locations
  - mobility (walking vs. driving) and 5G-4G handoff





Commercial 5G Measurement Study

- 2 month-long measurement study using Samsung S10 5G handsets
  - orientation tests; distance; time of day; etc.
  - line of sight (LOS) vs. with various obstructions; different locations

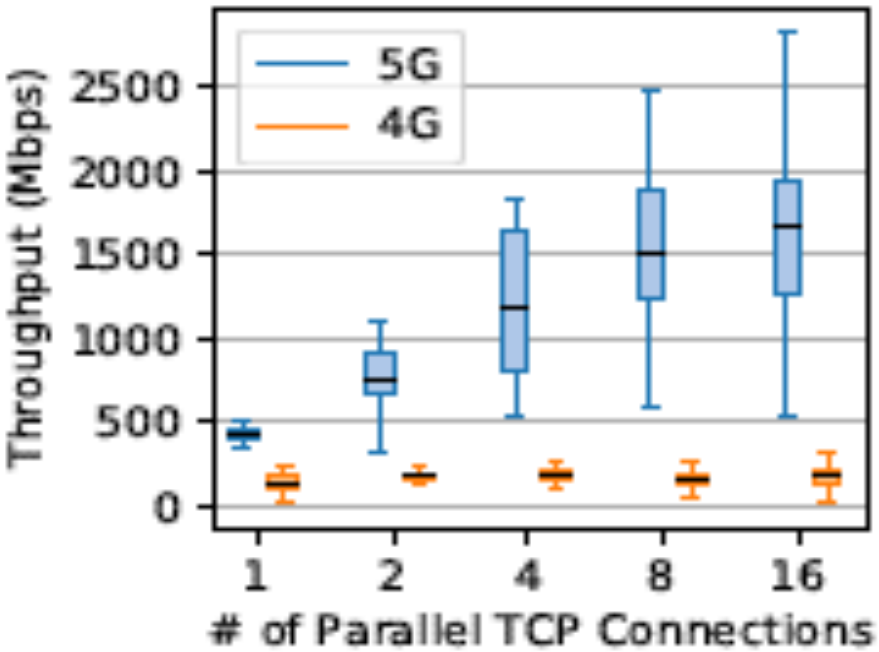


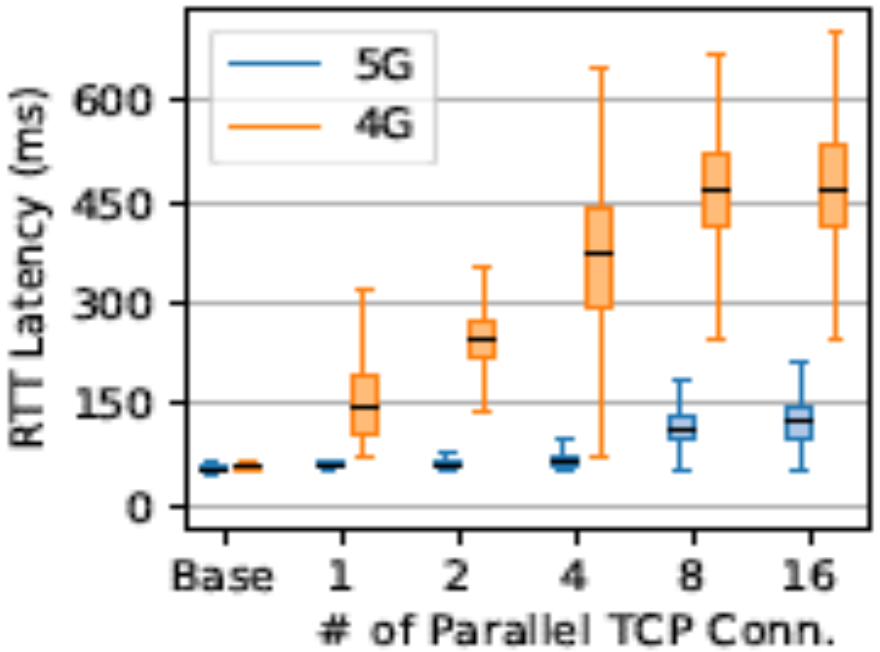
obstructions & locations *without (a)*
and *with (b) effective multi-paths*

Commercial 5G Measurement Study

- 2 month-long measurement study using Samsung S10 5G handsets
- Gathered large amounts of data: 5G vs. 4G
  - RTT, route, pkt loss, jitter tests using ping, traceroute, iperf UDP vs. TCP

Bandwidth Probing Tests ➔ Under good conditions (e.g., LoS), consistently attaining 1Gbps or more bandwidth per device, with less variability & delay jitter

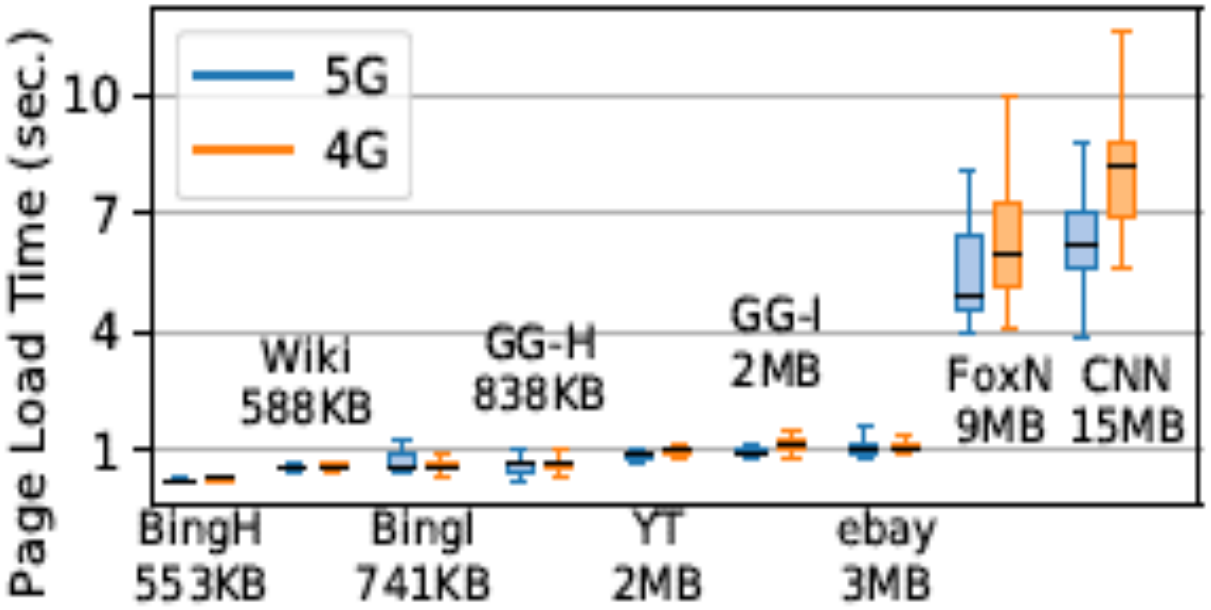




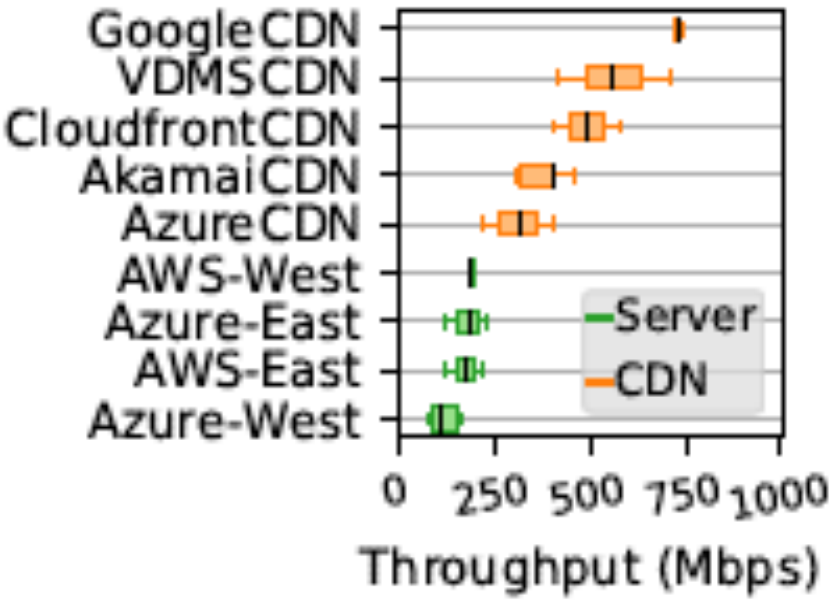
Commercial 5G Measurement Study

- 2 month-long measurement study using Samsung S10 5G handsets
- Gathered large amounts of data: 5G vs. 4G

Application (e.g., web browsing) Tests ➔ performance gap between 5G & 4G narrows
➢ bottlenecks may shift to end systems & core networks!



page load time over 9 web pages of different sizes

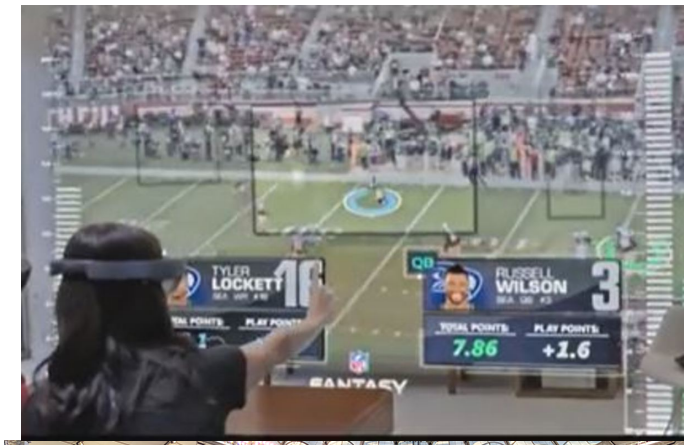


bulk download performance using 9 CDNs/cloud servers

5G, Edge Computing & Volumetric Video

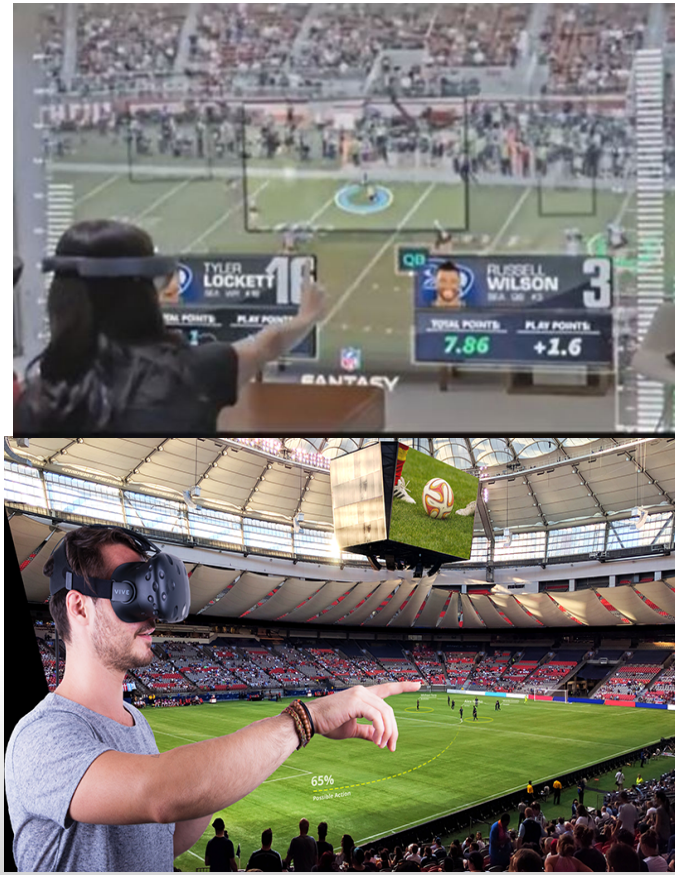
- 5G provides bw capacity to support volumetric video!
  - esp. video processing performed in an *edge cloud* nearby

How to effectively support *40K fans* in a large sport stadium following their (resp.) favorite players using AR/VR?



5G, Edge Computing & Volumetric Video

- 5G provides the bw capacity to support volumetric video!
  - esp. video processing performed in an edge cloud nearby
- With 40K fans in a stadium using AR/VR, each is following their own favorite player
- Suppose up to 1 Gbps bandwidth per user
  - ➔ 40K Gbps total
  - ➔ each user has multiple connections
- Given edge servers w/ 48 cores & 100 Gbps dual-port NICs, how many do we need?
  - ➢ or how many CPU cores do we need?



NFV & Net. Support for Edge Computing

AR/VR apps

edge video processing

access cor
e.g., for u
authentica

network mc
e.g., for acc

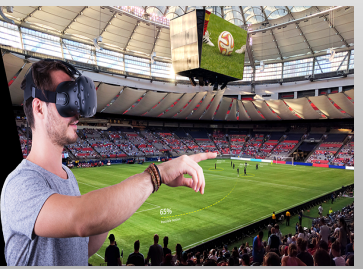
load bal
among edg
for video p

layer-3 forwarding

ACL → NM → LB → L3FW

5G-NR

NFV network packet processing

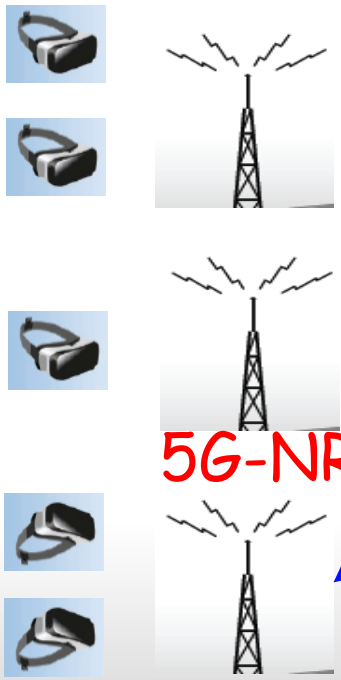


NFV & Net. Support for Edge Computing

AR/VR apps

edge video processing

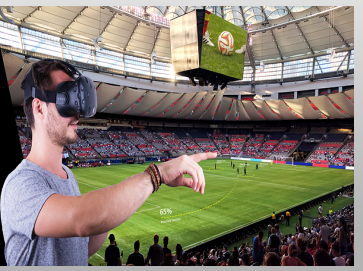
service function chain (SFC)



5G-NR

ACL → NM → LB → L3FW

NFV network packet processing

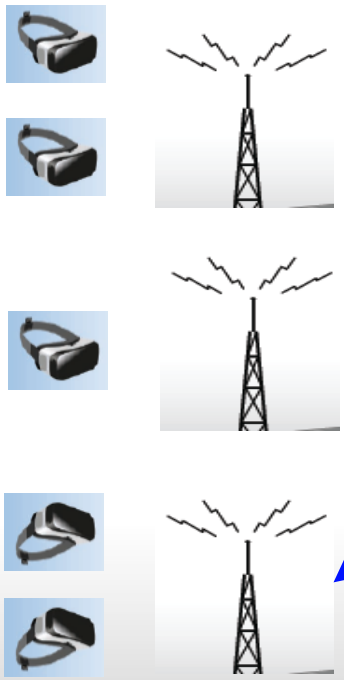


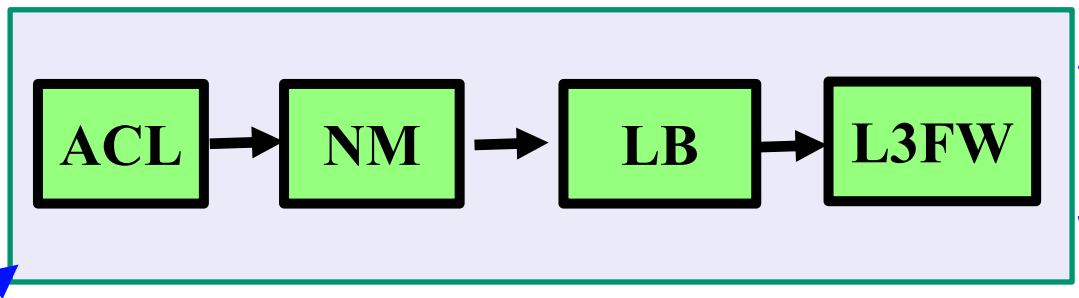
NFV & Net. Support for Edge Computing

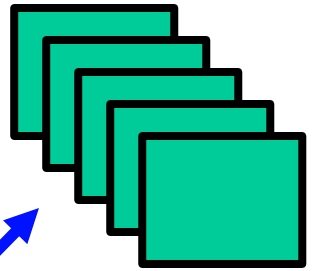
AR/VR apps

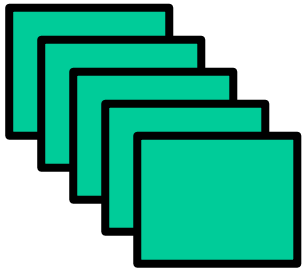
edge video processing

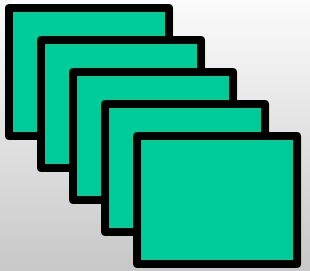
Can NFV process 40K Gbps traffic on commodity servers?



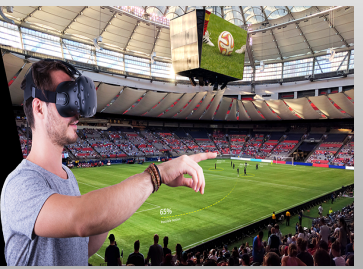








NFV network packet processing

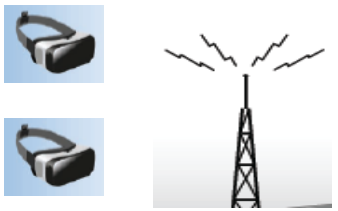


5G, NFV and Edge Computing

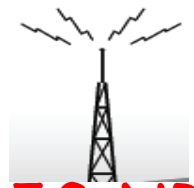
AR/VR apps

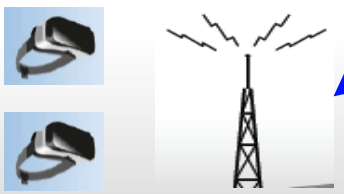
Can NFV process 40K Gbps traffic on commodity servers?

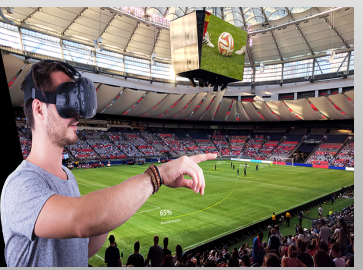
edge video processing

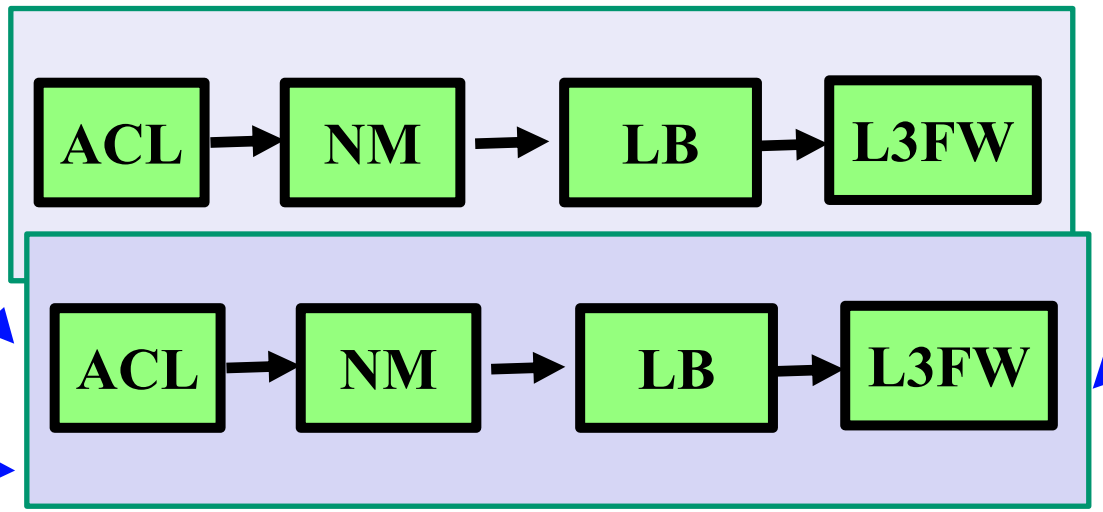


5G-NR

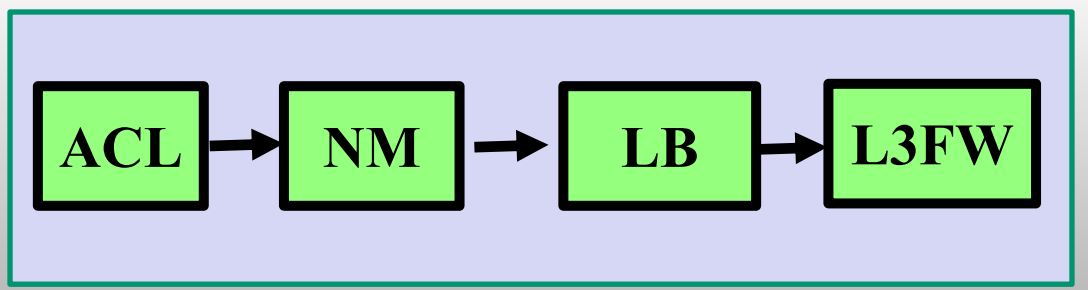


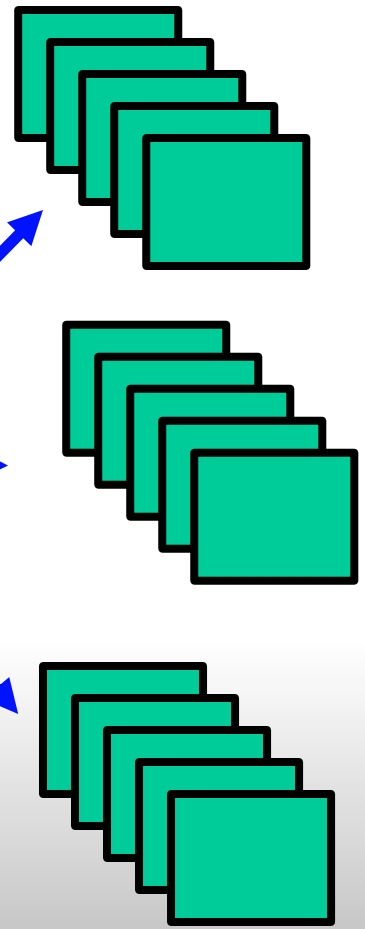






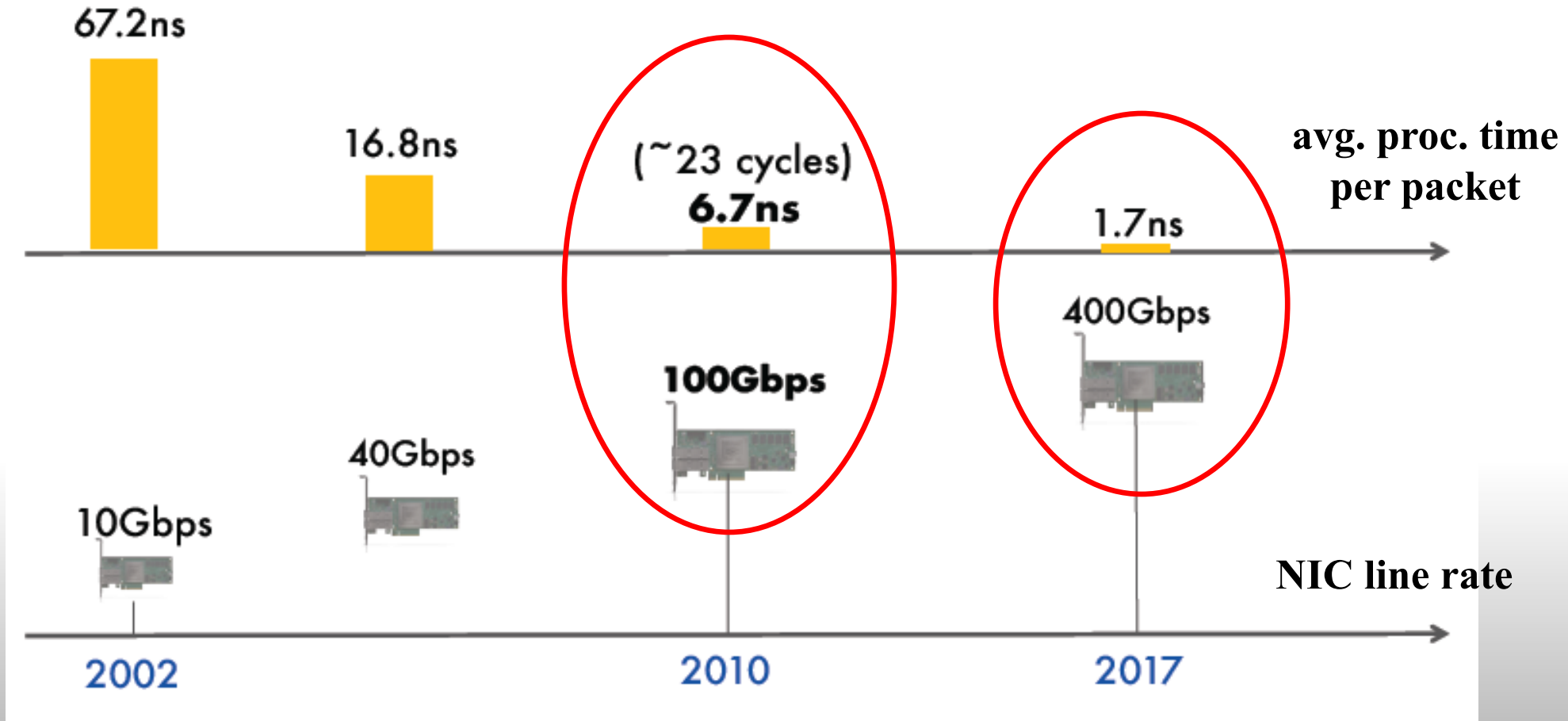
scale-out



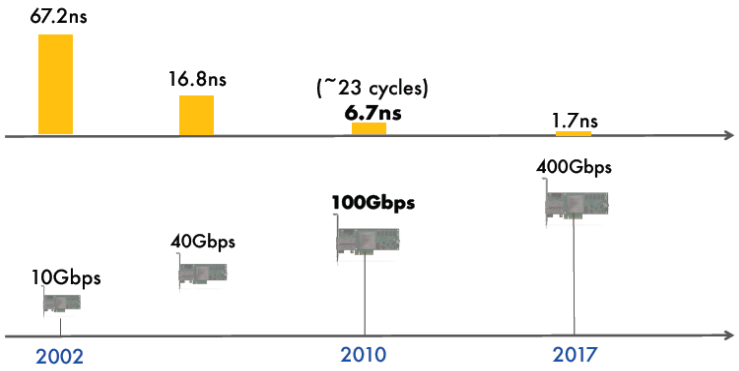


Software Packet Processing at 100+ Gbps

Keeping with the faster line speed via software packet processing is getting increasingly hard!

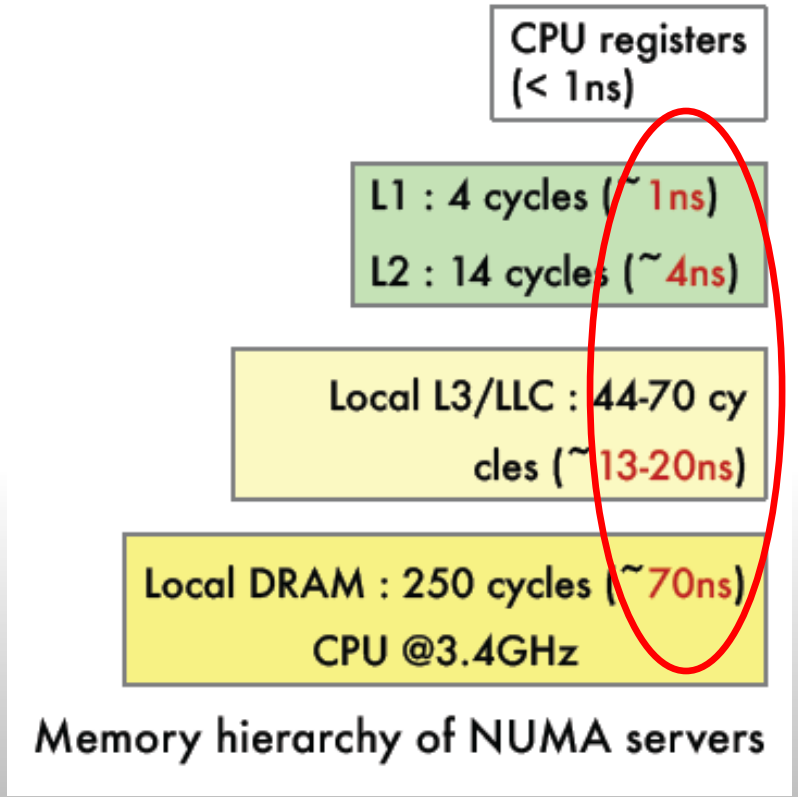


Sw Packet Processing & Multi-Core Servers

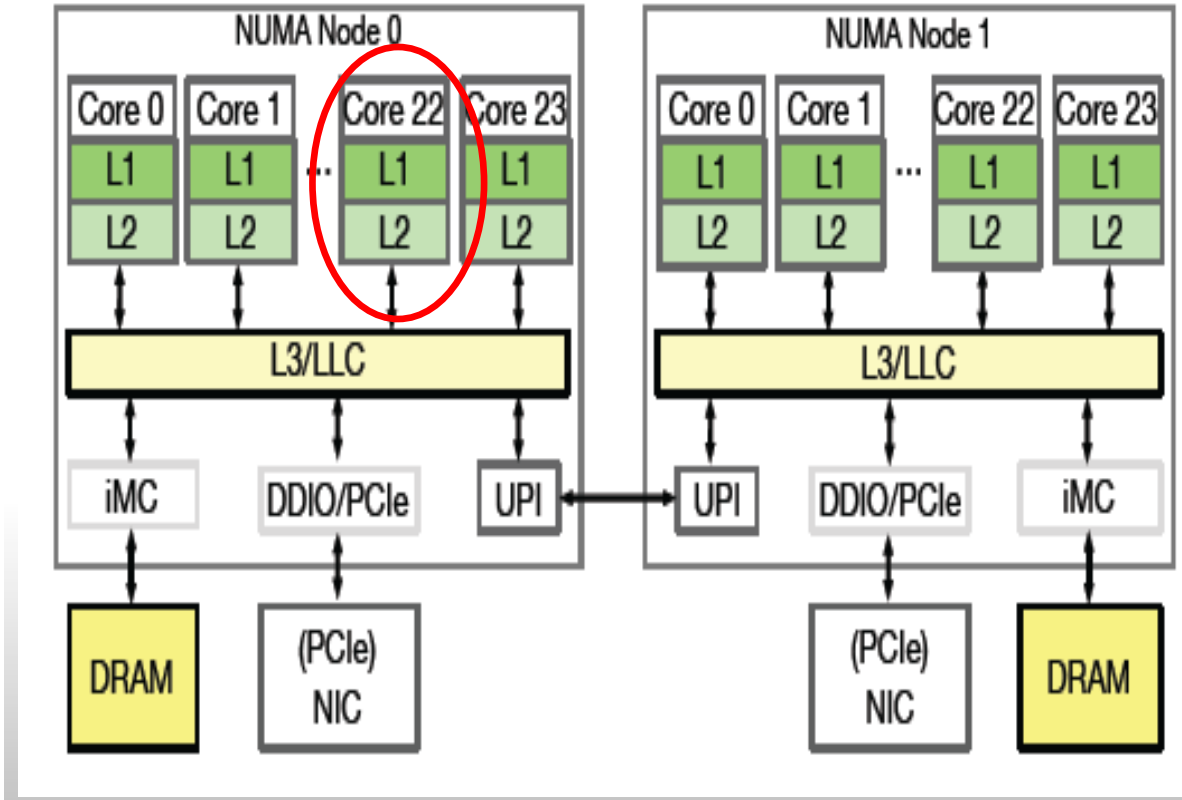


Ensuring most NF operations are L1/L2 bound

is important for 100Gbps line speed



Memory hierarchy of NUMA servers



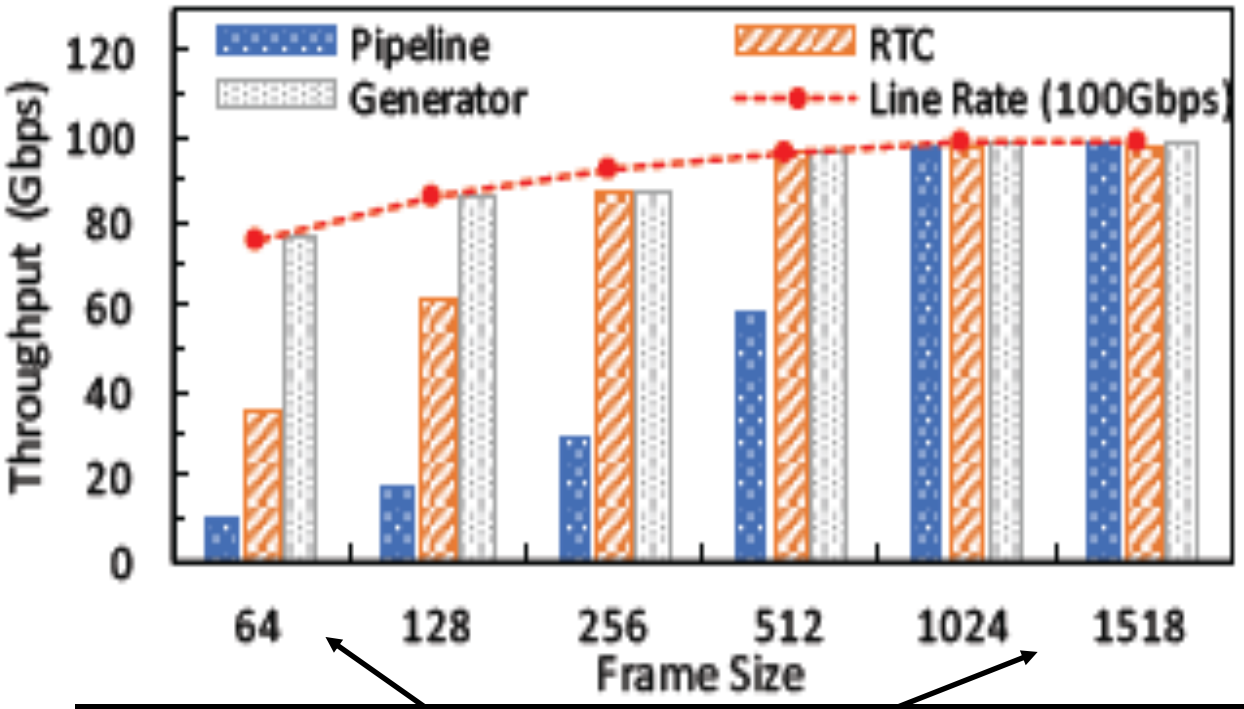
Intel(R) Xeon(R) Platinum 8168, dual CPU sockets w/ 24 cores each, CPU @2.7GHz clocked at 3.4GHz,

Good News …

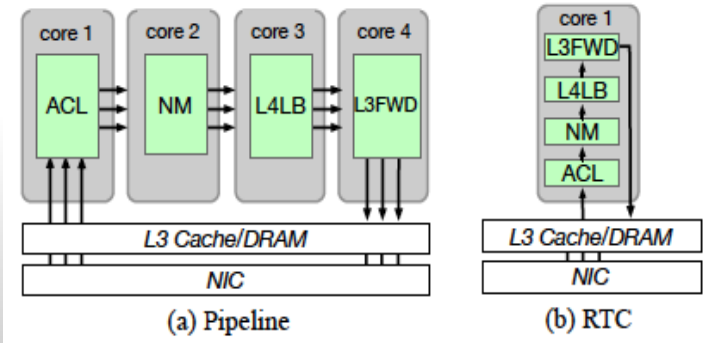
With larger packet sizes, 20 cores sufficient to meet 100 Gbps line rate

1 server (48 cores + 200Gbps NICs) ➔ 200 users (up to 1 Gbps bw per user)

but we need 200 servers just for edge network processing!



software packet processing using 20 cores, w/ diff. pkt sizes



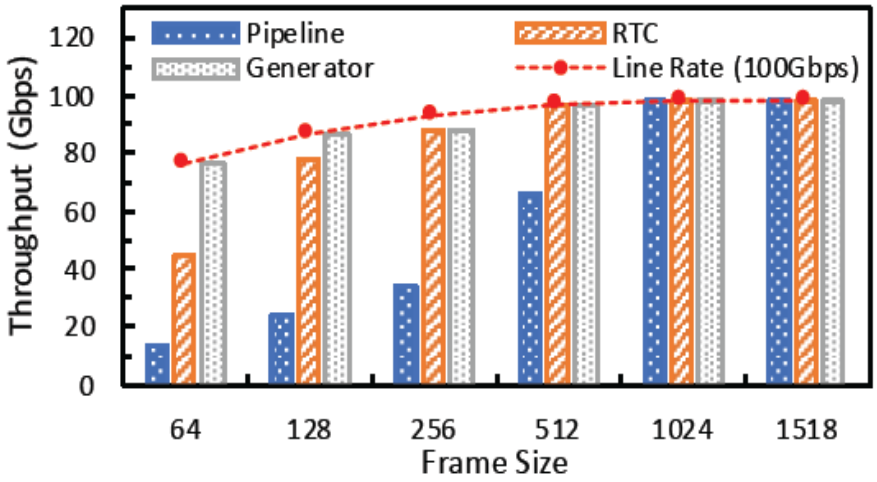
SFC execution models

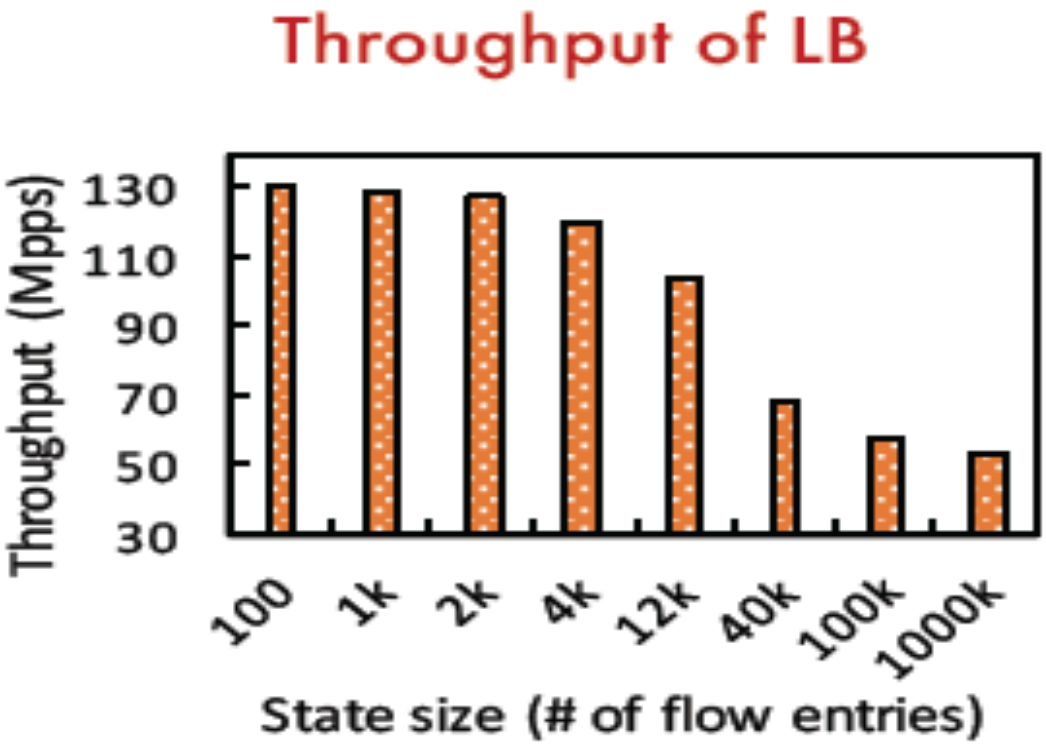
NFV needs to process more smaller packets than larger packets to keep up w/ line speed

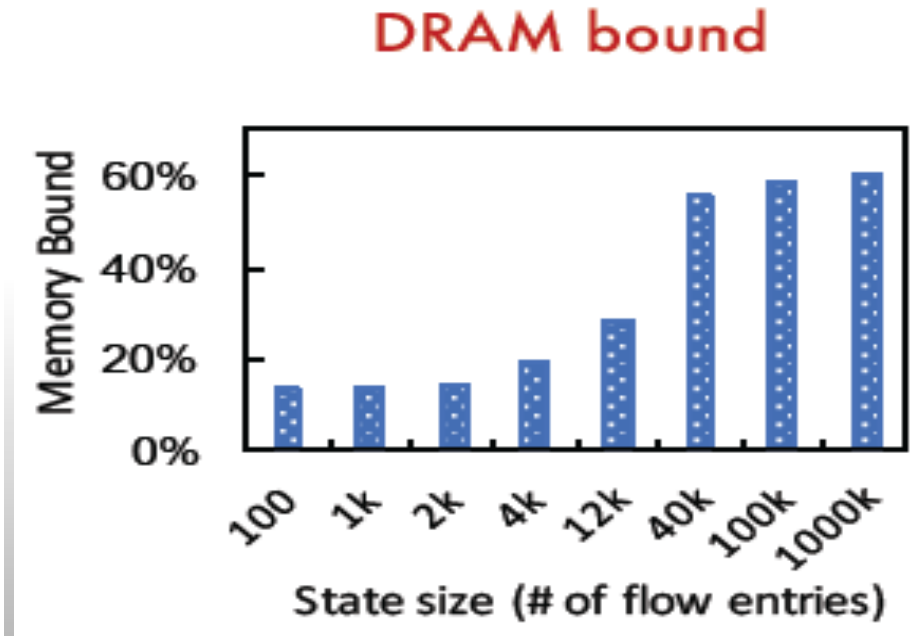
Bad News …

Earlier results hinge on optimist assumptions

Impact of state on *stateful* NF performance!







More Bad News …

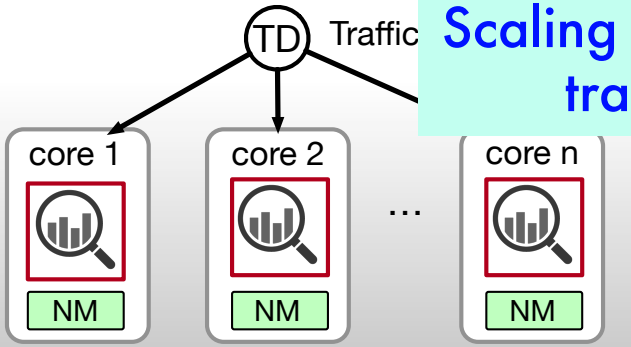
Performance gets even worse when NF instances share "state"

Scaling out NFV performance via multiple cores no longer linear!

In some worst cases, more cores can even hurt performance

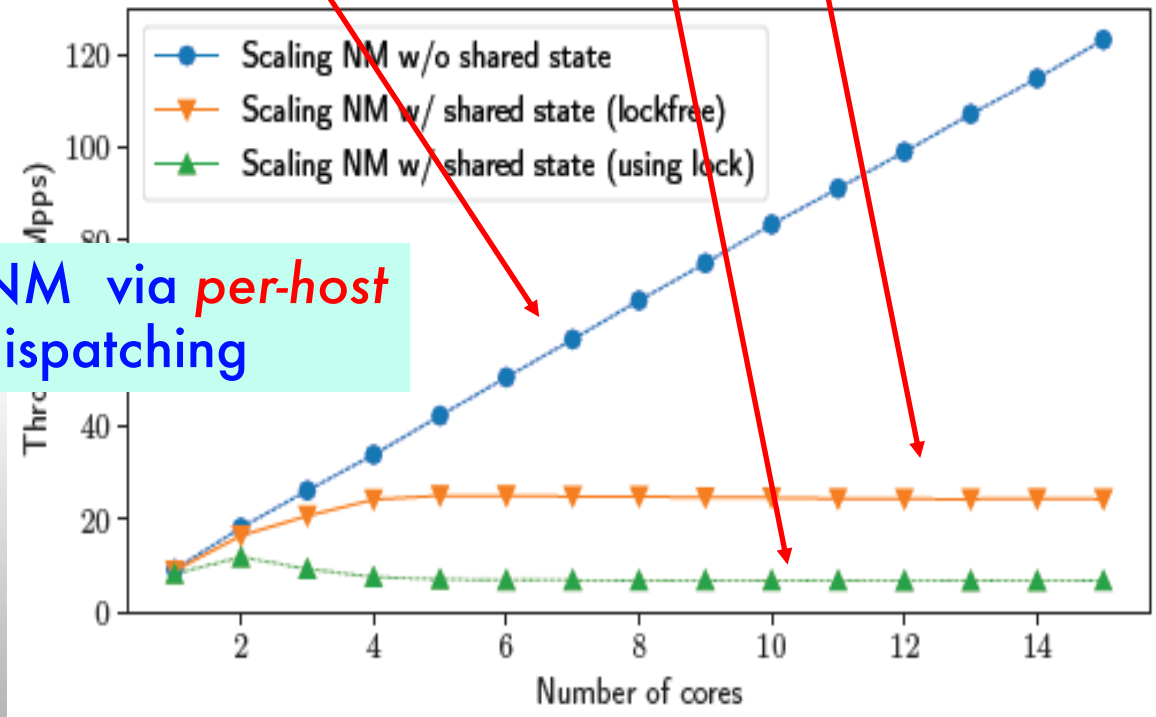
How to dispatch traffic & load balance among NF instances is crucial ➔ knowledge of state is important!

Scaling out NM via *per-flow* traffic dispatching



Shared L3/DRAM

Scaling out NM via *per-host* traffic dispatching



Network Support for Applications?

Emerging applications likely require "end-to-end" *and* "in-network" support: from end devices to edge to cloud --

How to provide effective "in-network" (edge) support for complex applications w/stringent requirements?

- Knowledge of network function as well as application function "semantics" is important
  - ➔ better "programming model" to expose such semantic info
- Can't afford to "manually" optimize each app per infrastructure
  - ➔ compiler/runtime system that can "automatically" account for & leverage hardware features & capabilities

Let's quickly look at how
computer & distributed systems support
diverse applications & their development
esp., how to deal complexity & scale

Application/Software Frameworks

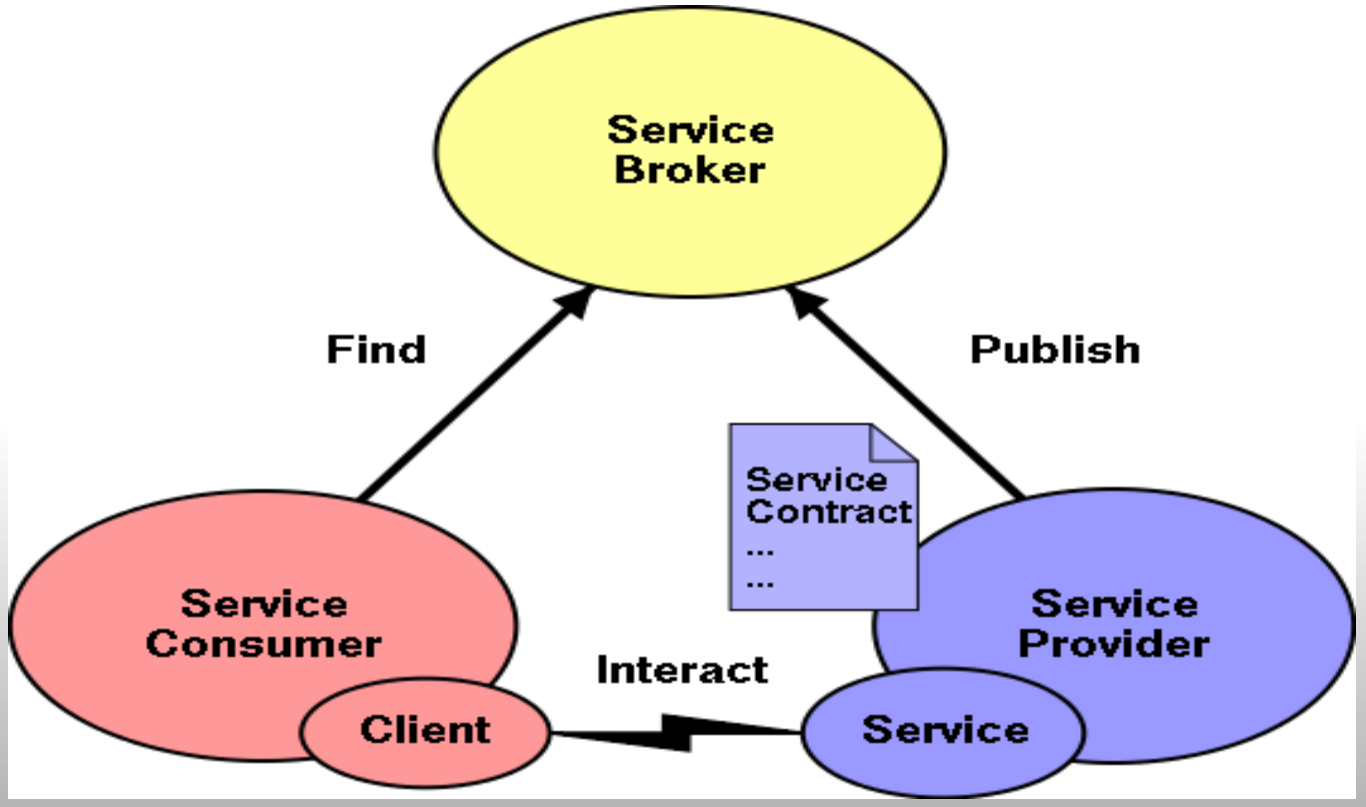
- Software system that implements "standard structure" (or generic functionality) to support target sets of applications
  - not merely a collection of libraries, but software *design patterns*
- Started w/ GUIs, then Service-oriented architecture (SOA)
  - e.g., MacApp, Microsoft Foundation Classes (MFC), .NET, CORBRA
- Popularized by Cloud Computing and Big Data Analytics, e.g.,
  - MapReduce, Spark, Ray, Tensorflow, PyTorch, …
  - Storm, Kafka, Akka, Finagle, Thrift, …
  - Kubernetes, Mesos, ONOS, ODL, …
- Most of today's Internet services and large-scale distributed applications are developed w/ application frameworks

Application/Software Frameworks

- Software systems that implement "standard structure" (or generic functionality) to support target sets of applications
  - not merely a collection of libraries, but software *design patterns*
- Emerged w/ GUIs, then Service-oriented architecture (SOA)
  - e.g., MacApp, Microsoft Foundation Classes (MFC), .NET, CORBRA
- Popularized by Cloud Computing and Big Data Analytics, e.g.,
  - MapReduce, Spark, Ray, Tensorflow, Caffe, PyTorch, …
  - Akka, Finagle, Storm, Kafka, Thrift, …
  - Kubernetes, Mesos, ONOS, ODL, …
- "Communications" (networking) is a key component of most app. (software) frameworks – many build their own "abstractions"

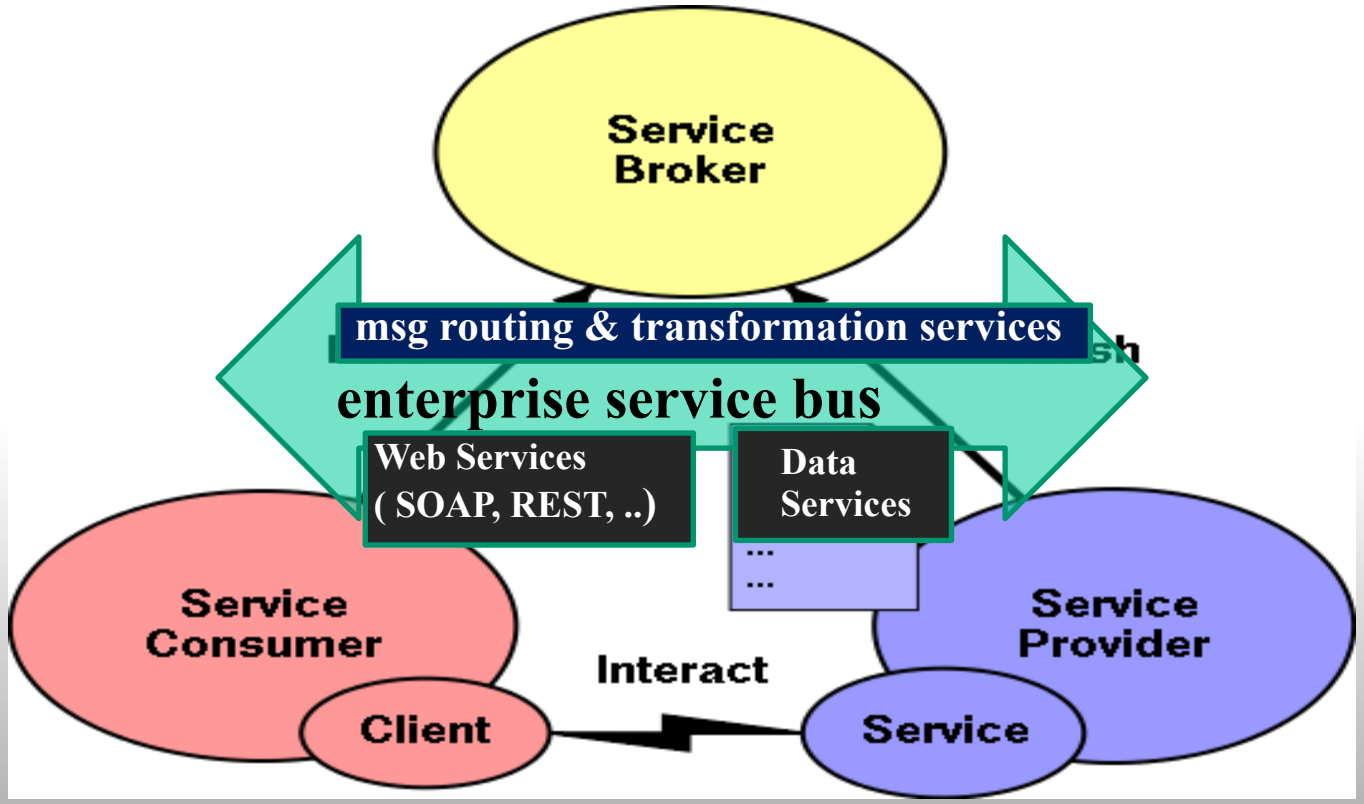
From SOA to Microservices

- Service-Oriented Architecture (SOA)
  - a collection of (loosely coupled) services, interacting with each other via communication (e.g., via SOAP, REST, RPC/RMI)
  - each service is independent of others, maintaining own state



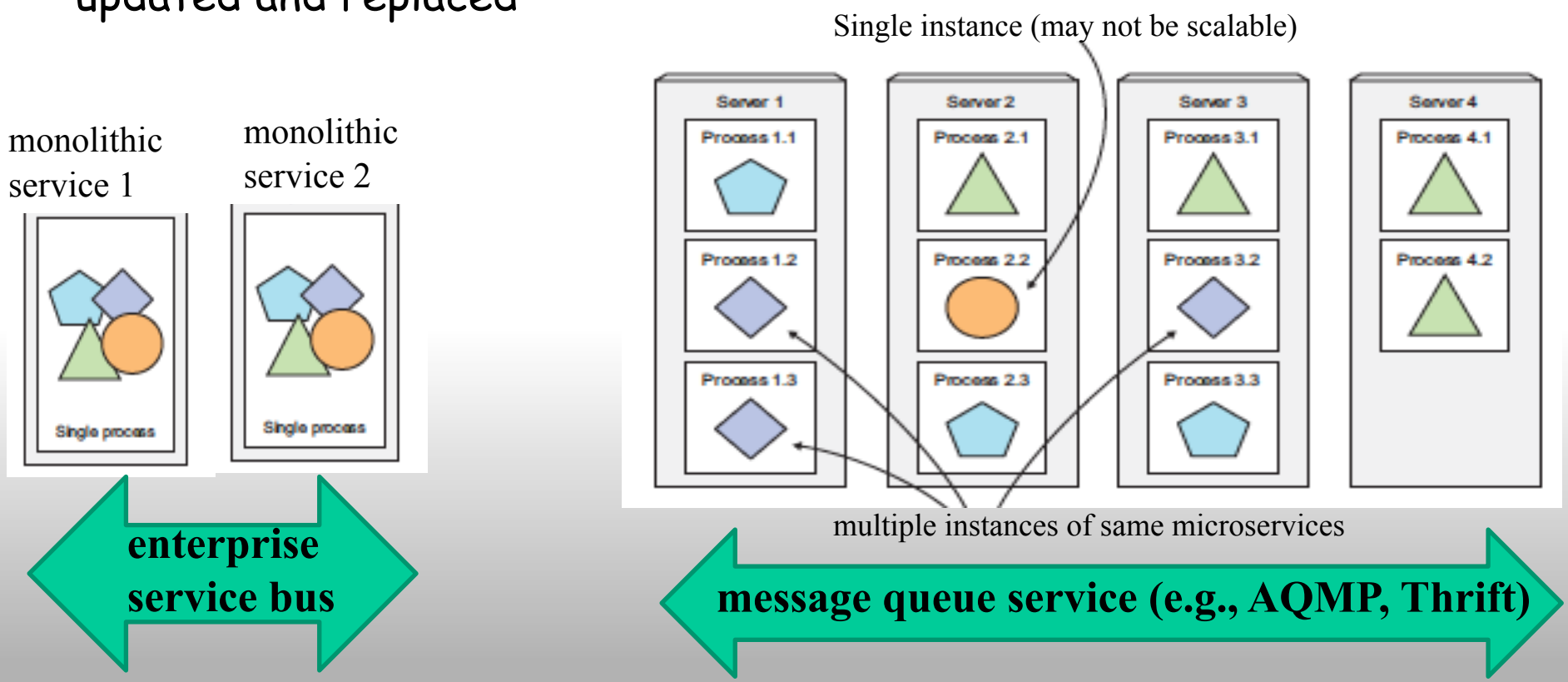
From SOA to Microservices

- Service-Oriented Architecture (SOA)
  - a collection of (loosely coupled) services, interacting with each other via communication (e.g., via SOAP, REST, RPC/RMI)
  - each service is independent of others, maintaining own state



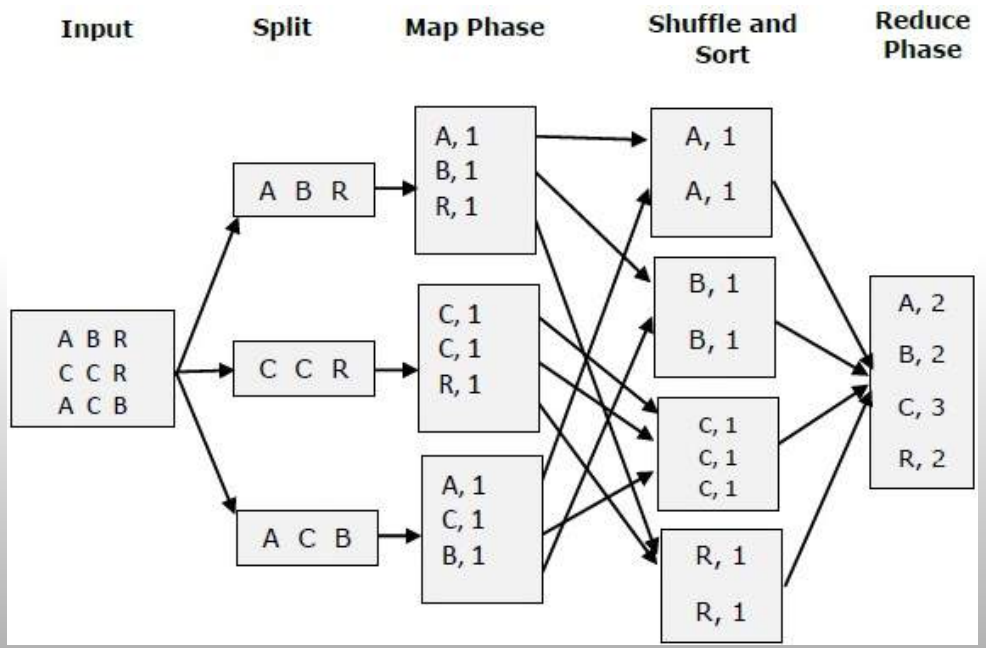
From SOA to Microservices

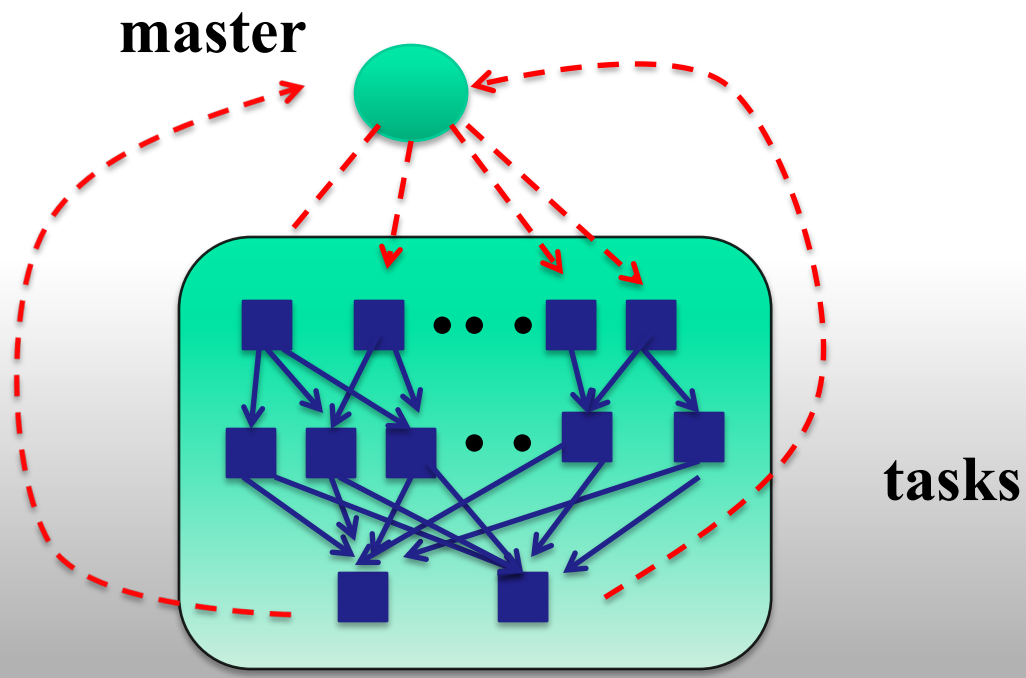
- Service-Oriented Architecture (SOA)
- Increasing demands for availability, scalability and velocity give rise to *microservice* architectures
  - monolithic service ➔ microservices that can be independently scaled, updated and replaced



Big Data Analytics Frameworks

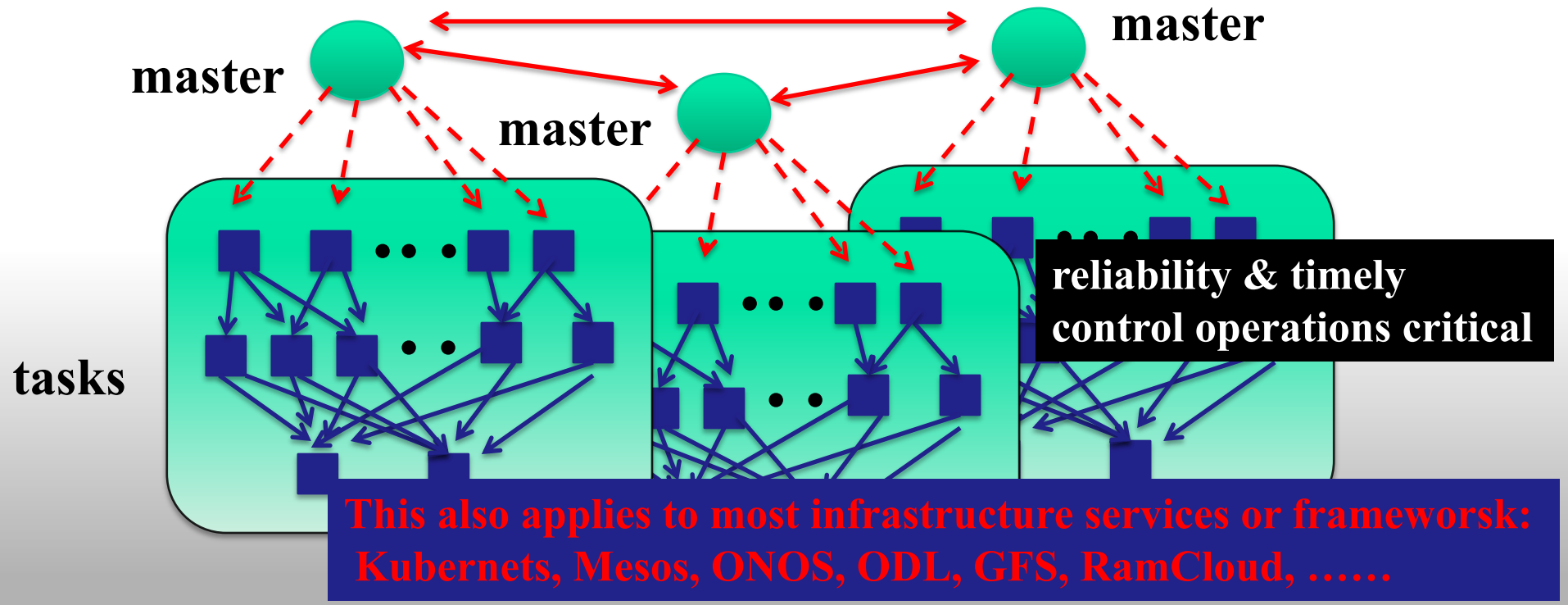
- Map-Reduce, Spark, Dryad, etc
  - master (control) and a collection of workers (compute tasks)
  - framework manages communications among master-to-workers and communications among workers
  - batch-mode, synchronous communications ➔ distinct patterns





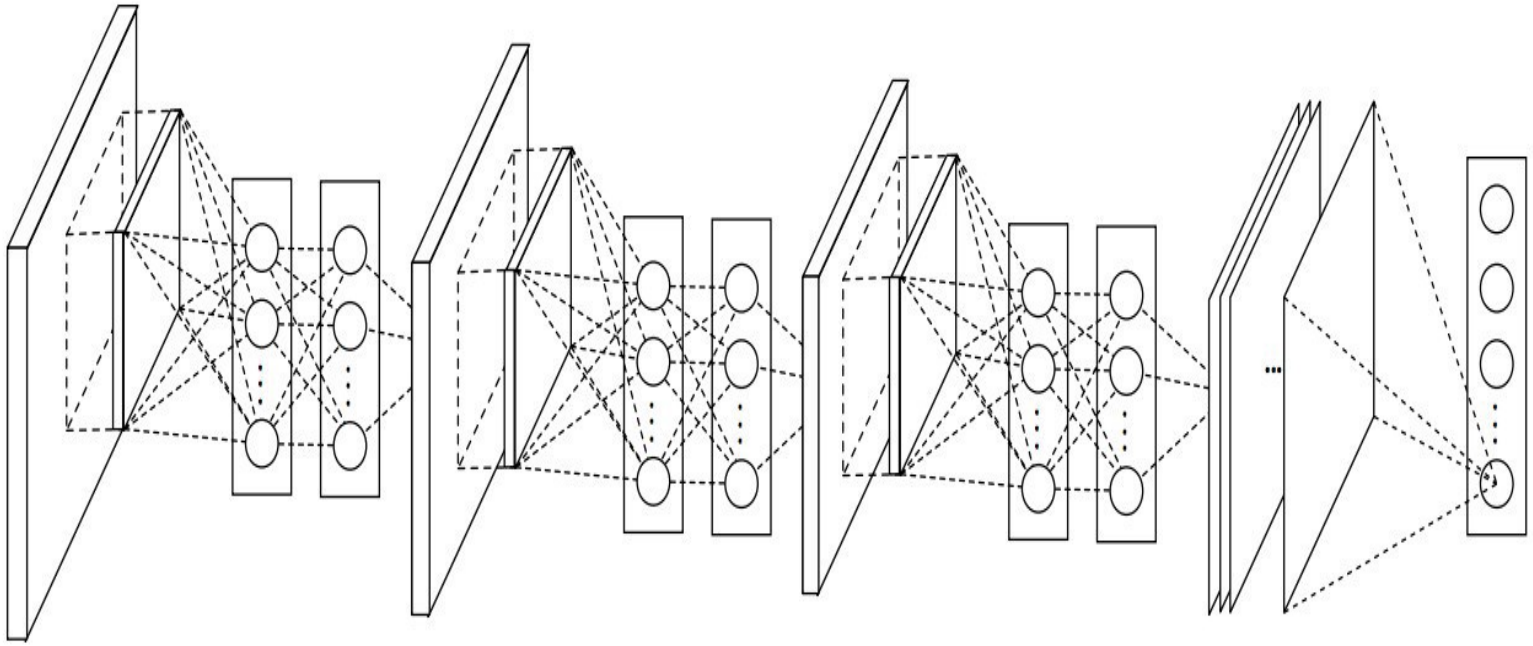
Big Data Analytics Frameworks

- Map-Reduce, Spark, Dryad, etc
  - master (control) and a collection of workers (computations)
  - framework manages communications among master-to-workers and communications among workers
  - batch-mode, synchronous communications ➔ distinct patterns



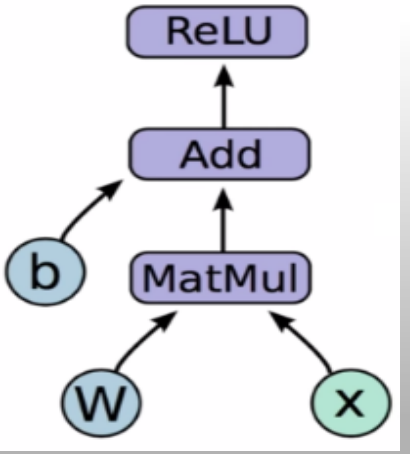
Deep Learning Frameworks

- Deep Learning Neural Networks
  - Multiple layers of neurons, with some relative simple computations (gradient computation, matrix multiplications, ReLU, SoftMax, …)
  - A lot of communications among neurons; iterative optimization steps



Deep Learning Frameworks

- Deep Learning Neural Networks
  - Multiple layers of neurons, with some relative simple computations (gradient computation, matrix multiplications, ReLU, SoftMax, …)
  - A lot of communications among neurons; iterative optimization steps
- Deep Learning Frameworks: Tensorflow, Caffe, PyTorch, …
  - Abstract out common "design patterns" to provide high-level prog. Models
  - Data (tensor) flow graphs, with "compute nodes" for gradient computation, vector/matrix multiplication, ReLU, SoftMax, common optimizers, …
- Ray for AI & reinforcement learning
  - Asynchronous communications more prevalent
- With GPU and TPU, synchronization ("state") & communication overheads become more critical



Network Support for Applications?

Today's networks offer only a "one size fits all" solution

- "best-effort" IP net. service, w/ TCP/UDP transport on end systems
- Networks as a "bag of protocols"
- App. frameworks build their own "communications middleware" with various "high-level" abstractions/semantics
  - ESB/message broker (pub-sub sys.), w/ msg transformation, routing, …
  - Message queue protocols: req/reply, req. only; at most/least once, …
  - Reliable msg. broadcasting (distributed "transactions") using consensus
- A lot of duplicate efforts; most built on top of TCP!
  - ➢ But TCP suffers many issues (w/ hard-coded reliability & congest control)
  - ➢ and kernel overheads -- see Keynote by Kyoungsoo Park at APNet'19

Network Support for Applications?

Today's networks offer only a "one size fits all" solution

- "best-effort" IP net. service, w/ TCP/UDP transport on end systems
- Networks as a "bag of protocols"
- App. frameworks build their own "communications middleware" with various "high-level" abstractions/semantics

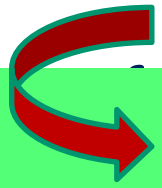
How can we leverage new networking innovations to provide better support for emerging diverse & complex applications?

- Increasingly programmable data plane (Openflow, P4, whitebox switches, etc.) and "smartNICs" (e.g., DPDK, RDMA, FPGA, NPU, …)
- Virtualized network functions running on commodity servers
- New network control & management paradigms, ……

Network Services as Frameworks?

Today's networks offer only a "one size fits all" solution

- "best-effort" IP net. service, w/ TCP/UDP transport on end systems
- from Networks as a "bag of protocols"



to network services as frameworks?

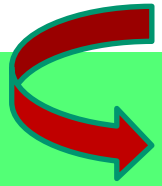
How can we leverage new networking innovations to provide better support for emerging diverse & complex applications?

- Increasingly programmable data plane (Openflow, P4, whitebox switches, etc.) and "smartNICs" (e.g., DPDK, RDMA, FPGA, NPU, …)
- Virtualized network functions running on commodity servers
- New network control & management paradigms, ……

Network Services as Frameworks?

Today's networks offer only a "one size fits all" solution

from Networks as a "bag of protocols"



to network services as frameworks?

- Elevating network services to higher-level frameworks by *co-designing* applications, distributed and networking systems

For each (type/category of) application or service,

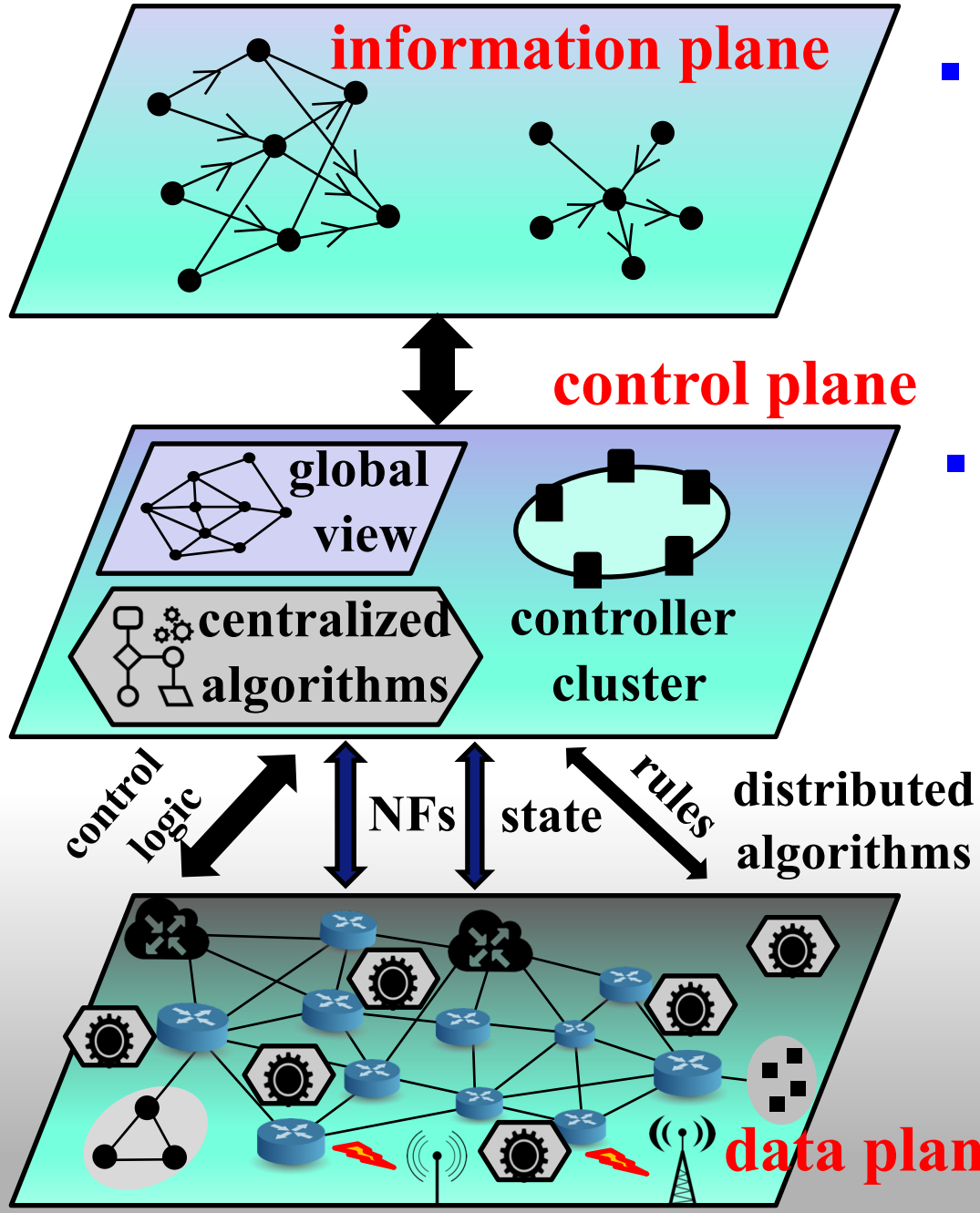
- abstract out generic comm. "design patterns" to provide higher level network service constructs & primitives, & support rich semantics!
- implementing certain "app/middleware" primitives as (virtual) NFs
- off-loading certain "app" or network functions to (smart) hardware

➔ also require software & hardware co-designs

Concluding Remarks:
Case for Network Services as Frameworks

- "Dumb" network arch. with a "one-size-fit-all" best-effort service no long meets the needs of emerging applications & services!
- Advances in server & networking technologies made it easier to support more diverse & flexible network services
  - new server/NIC support: DPDK, RDMA, NetFPGA, GPU, NPU. …
  - programmable switches (e.g., P4), SDN and NFV, 5G & beyond, …
- Elevating network services to higher-level frameworks by *co-designing* applications, distributed and networking systems
  - new high-level abstractions & net. primitives, programming models, …
  - compiler/runtime systems, software-hardware co-designs, …

A Three Plane View



- high-level abstractions & network primitives; (*declarative* & *granular*) programming models
- compiler & runtime systems, orchestration, centralized & distributed control algorithms, resource management, ..
- Software-defined network & system infrastructure
  - software-based vNFs
  - hardware accelerators
  - …..

Last Words of Caution:

- There are *unique* networking challenges!
- In other words, we cannot blindly "borrow" techniques developed for distributed systems and/or cloud computing!
  - providing increasing network bandwidth — diff. scaling requirements
  - supporting latency, jitter & other QoS — "deterministic" services
  - resilient network services, coping w/ network failures, …
  - …

➢ Existing "provably correct" distributed mechanisms (e.g., Raft) may break under different network assumptions, see, e.g., our APNet'18 paper:

"Raft Meets SDN: how to elect a leader in an 'unruly' network"

We're living in an exciting time

for networking research (again) !

Thank You!