Le génie pour l'industrie

4th ITU workshop on Network 2030 jointly with ITU Forum on Future Applications and Services. Perspective 2030

Introducing FlexNGIA: A Flexible Internet Architecture for the Next-Generation Tactile Internet

Mohamed Faten Zhani (mfzhani@etsmtl.ca)
École de technologie supérieure (ÉTS Montreal), Canada

Joint work with Hesham ElBakoury (Futurewei Technologies, USA)

Saint Petersburg, Russian Federation, 21-23 May 2019

Outline

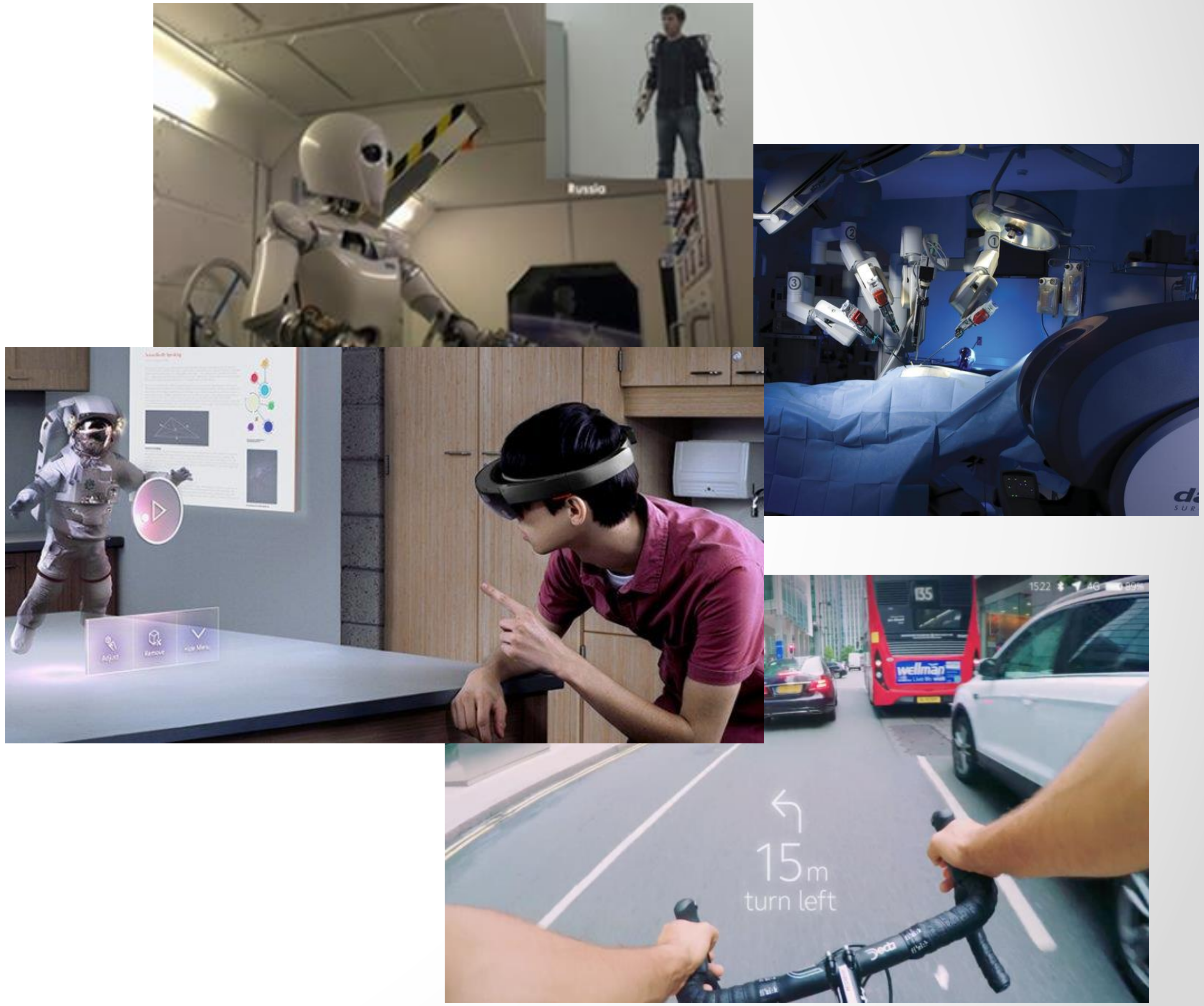
- A Glance into the Future
- Limitations of Today's Internet
- FlexNGIA: Fully-Flexible Next-Generation Internet Architecture
- Use cases
- Conclusion

M. F. Zhani, H. ElBakoury, "*FlexNGIA: A Flexible Internet Architecture for the Next-Generation Tactile Internet*," ArXiV 1905.07137, May 17, 2019
https://arxiv.org/abs/1905.07137

A Glance into the Future

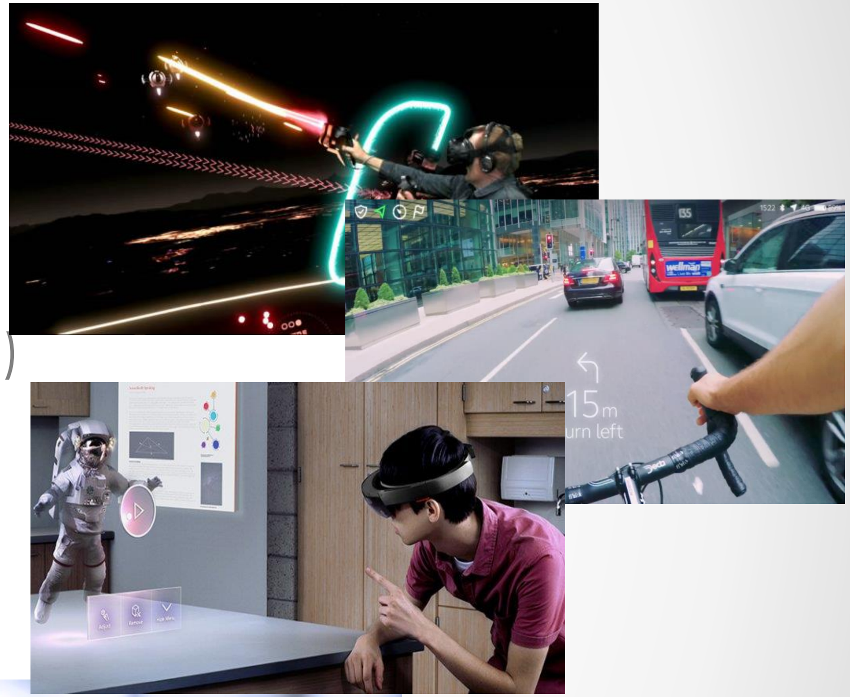
Future Applications

- Telepresence
- Virtual Reality
- Augmented Reality
- Holoportation



Requirements & Characteristics

- High processing power: real-time processing
- High bandwidth:
  - Holoportation: 30Gbps to 4.62 Tbps
  - Virtual Reality: (16K, 240 fps) ➔ 31.85 Gbps (true immersion)
- Ultra-low Latency
  - 1ms to 20ms (depending on the interactivity)
  - Multi-flow synchronization
- High availability
  - No downtime
  - Five 9's is not enough (99.999% ➔ 4ms/hour) [Google]
- High reliability



Requirements & Characteristics

- Octopus-like applications
  - Large number of flows
  - Traffic from multiple sensors/objects in different locations towards multiple destinations
- Changing requirements
  - Each flow has different performance requirements (e.g., throughput, latency, packet loss)
  - Requirements for each flow can change over time

M. F. Zhani, H. ElBakoury - FlexNGIA 2019 (https://arxiv.org/abs/1905.07137)

The "Tactile Internet"

- "**Extremely low latency** in combination with **high availability**, **reliability** and **security** will define the character of the *Tactile Internet*", ITU [3]

TACTILE INTERNET

Somatosensory
Computing
Reliability
Haptics
Real-time
Ultra-low latency
Perception
Security
Kinesthetic
Availability
Telepresence
Virtual Reality
Telesurgery
Industry 4.0
Augmented Reality
Humanoid robots
Autonomous Driving

Outline

- A Glance into the Future
- Limitations of Today's Internet
  - Internet Infrastructure and Services
  - Network Stack Layers and Headers
  - Sources of Latency
- FlexNGIA: Fully-Flexible Next-Generation Internet Architecture
- Use cases
- Conclusion

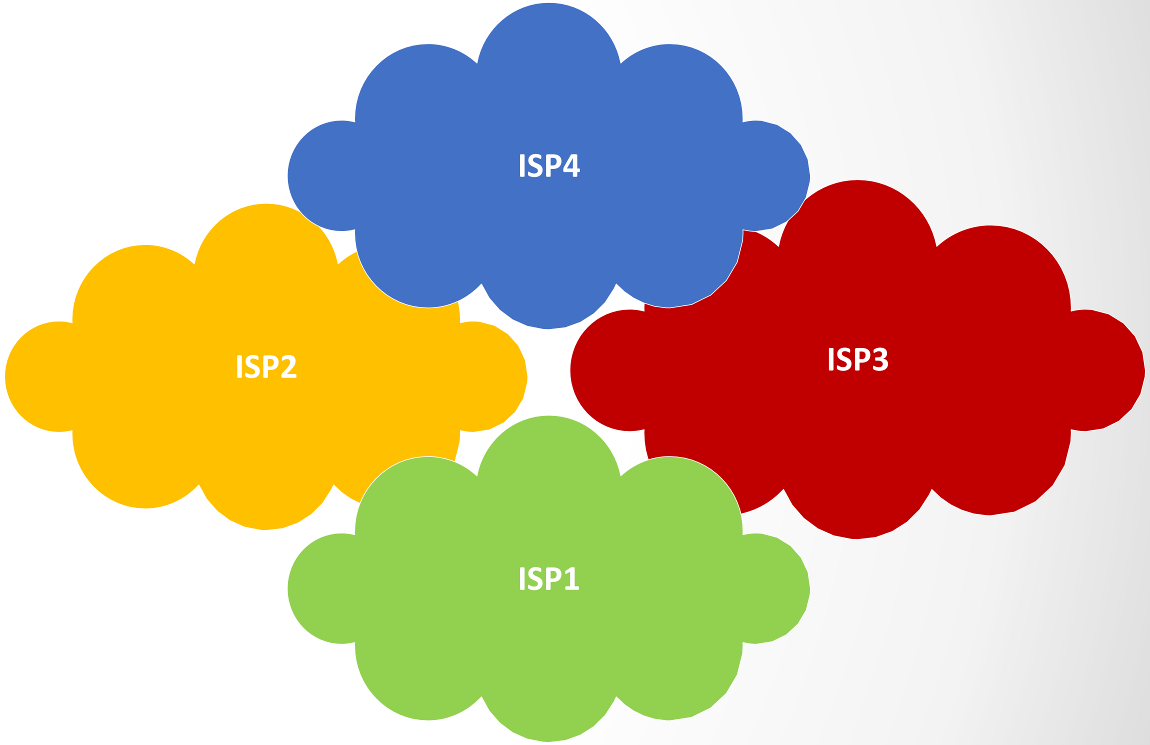
Internet Infrastructure and Services

- A network of networks
- Offered service: “Best effort” data delivery.. no more

  ➔No control over the infrastructure

  ➔No control over the end-to-end path and quality of service

  ➔No performance guarantees



Le génie pour l'industrie

Transport Layer Protocols

- Current Transport protocols:
  - Traditionally and mainly UDP and TCP
  - Modern protocols: SCTP and QUIC

But let's focus first on TCP..

- One-size-fits-all service offering: TCP offers reliability, data retransmission, congestion and flow control
  - No all services are needed in all applications
  - These services are mandatory for all data and throughout the whole communication
  - They may be needed only for some time or some data only during the communication

Transport Layer Protocols (cont)

- Blind congestion control
  - TCP is not really aware of the real state of the network
  - TCP cannot detect packet loss accurately (e.g., timeout, 3 duplicate ACK)
- High retransmission delays
  - 1 retransmission: timeout + e2e delay (~ 3x e2e delay)
  - Even with selective repeat retransmission (~ 3x e2e delay)

Transport Layer Protocols (cont)

- The two end points limitation
  - A single application is transferring data from different objects/sensors (may be located in different sources/destinations)
  - TCP flow is not aware of the "other" application' flows:

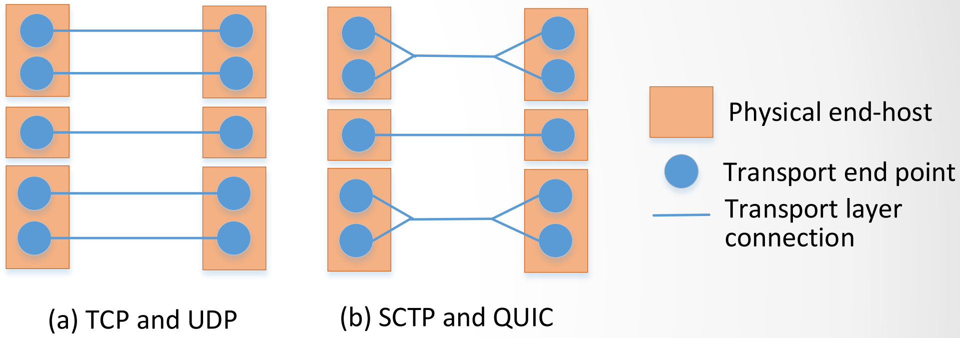
  ➔ Each flow is operated independently from the others

  - The transport layer and the network are not aware which flows belong to the same application (priority?)

Transport Layer Protocols (cont)

What are the limitations of SCTP and QUIC?

- E2E communication: multiple flows (streams) of the same application cannot connect more than two end-points



- A blind congestion control
- No support from the network: the network knows better about its state

  → Can better locate and manage congestion
  → Predict and detect more efficiently congestions/failures/problems…
  → Can retransmit faster
  → Can provide better garantees in terms of delay and packet loss

Le génie pour l'industrie

Network Layer Protocols

- Offer only data delivery (through routing)
- Do not provide performance guarantees
- Are not aware of the applications
  - The application composition (in terms of flows)
  - Performance requirements of each of these flows and how these requirement change over time
- Drop packets « blindly »
- No collaboration with the transport layer
  - Do not provide explicit feedback or support to transport layer (maybe ECN is interesting but it is not enough)
  - Do not help with other transport services (e.g., reliability)

Network Stack header

Problems with current headers:

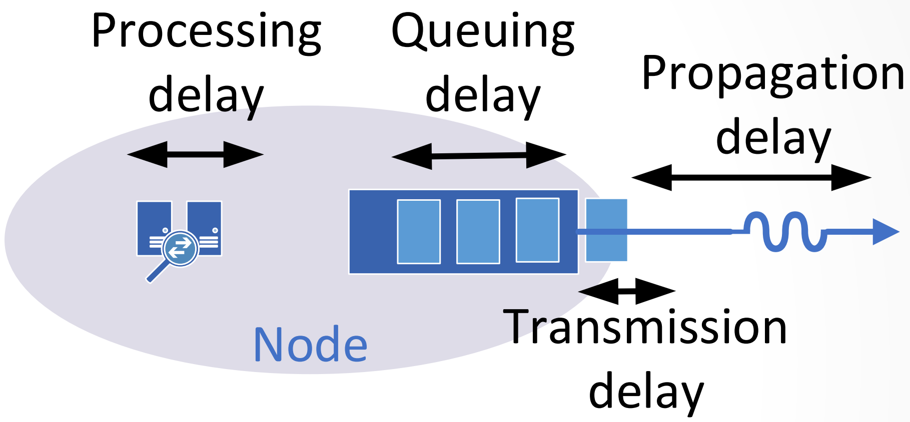
- Do not provide additional informations about objects/sensors, flows belonging to the same application, applications' requirements, etc.
- Not flexible enough: It is not easy to incorporate meta-data and commands

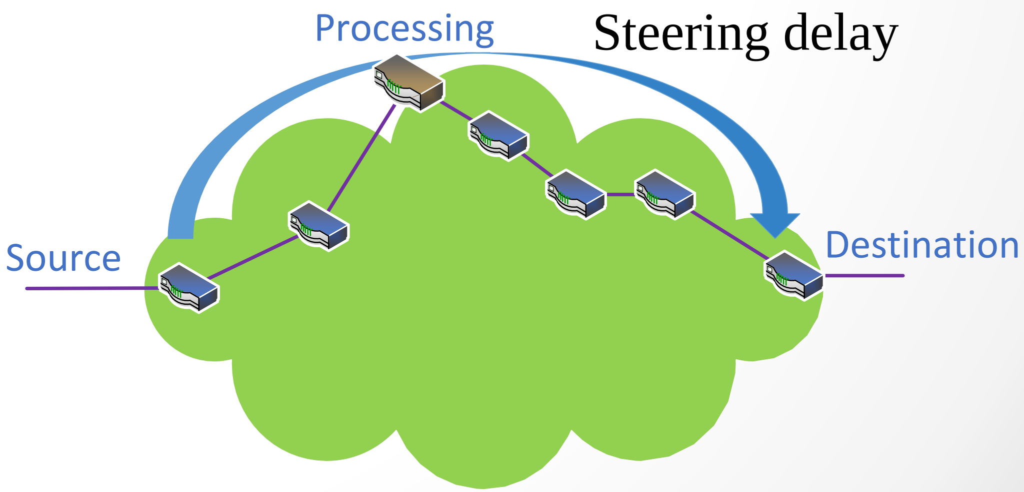
Le génie pour l'industrie

Sources of Latency

Types of Delay

- Processing delay
- Queuing delay
- Transmission delay
- Propagation delay
- Steering delay
- End-to-end delay





Outline

- A Glance into the Future
- Limitations of Today's Internet
- FlexNGIA: Fully-Flexible Next-Generation Internet Architecture
  - Future Internet Infrastructure and services
  - Business Model
  - Management Framework
  - Network Protocol Stack/Functions
  - Stack Headers
- Use cases
- Conclusion

Future Internet Infrastructure and Services

How does Future Internet look like?

- Still a network of networks..
- What is new?
  - More services: Service Function chains
    ➔More advanced functions
    ➔More than just delivery
  - Stringent performance guarantees

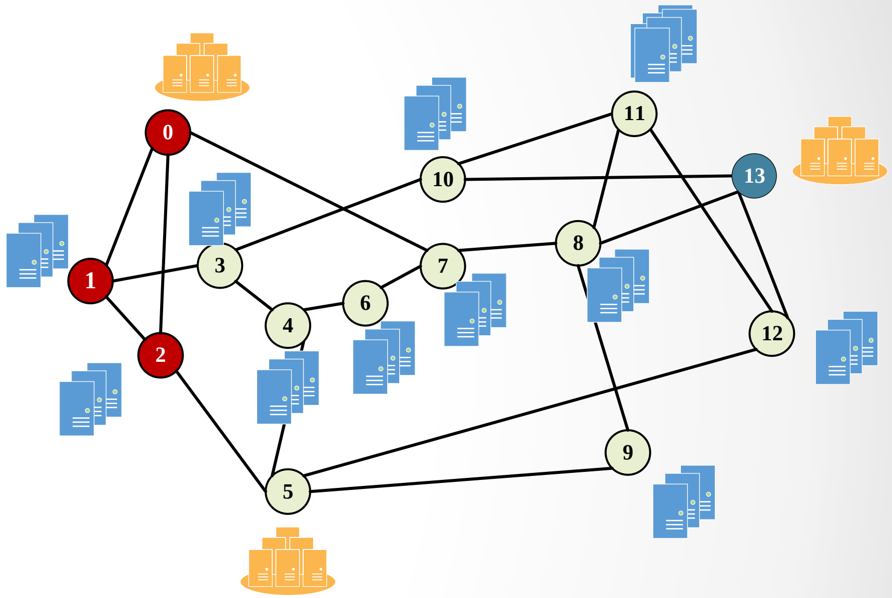
Future Internet Infrastructure and Services

- Computing resources are everywhere
    - Available at the edge and at the core of the network
    - Commodity servers but also dedicated hardware, FPGA, GPU, NPU, etc.

  ➔ In-Network computing

  ➔ Reduce steering delay

  ➔ Full Programmability: Any function could be provisioned anywhere (virtual machines/containers)



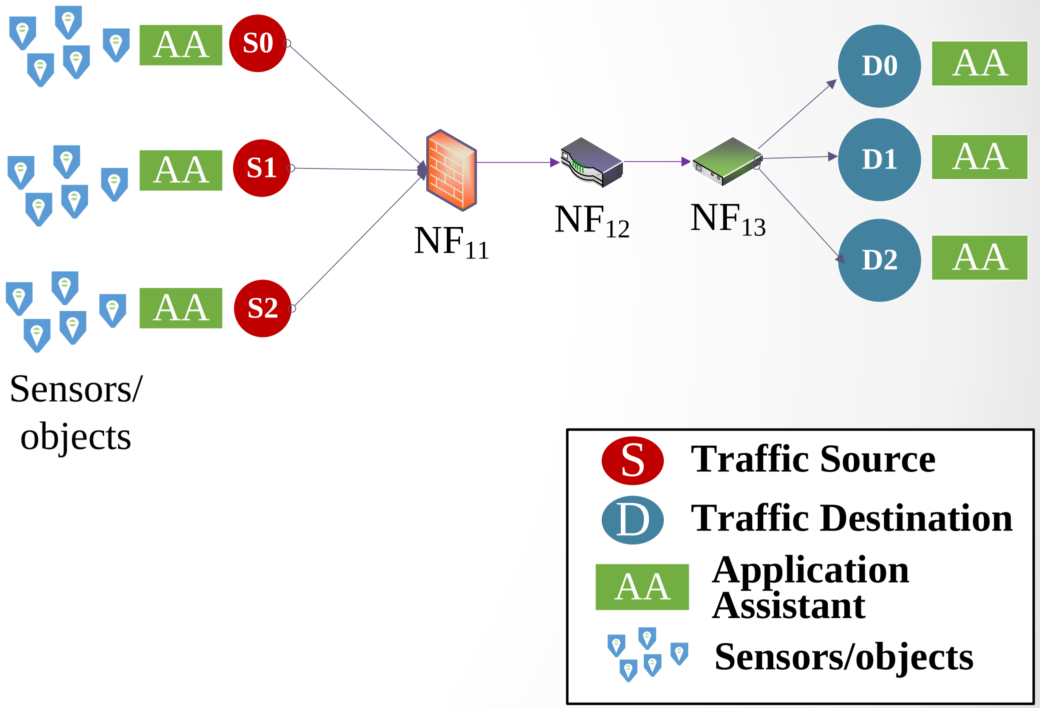
Cloud Data Center
Micro cloud

Future Internet Infrastructure and Services

Le génie pour l'industrie

Service Function Chain (SFC)

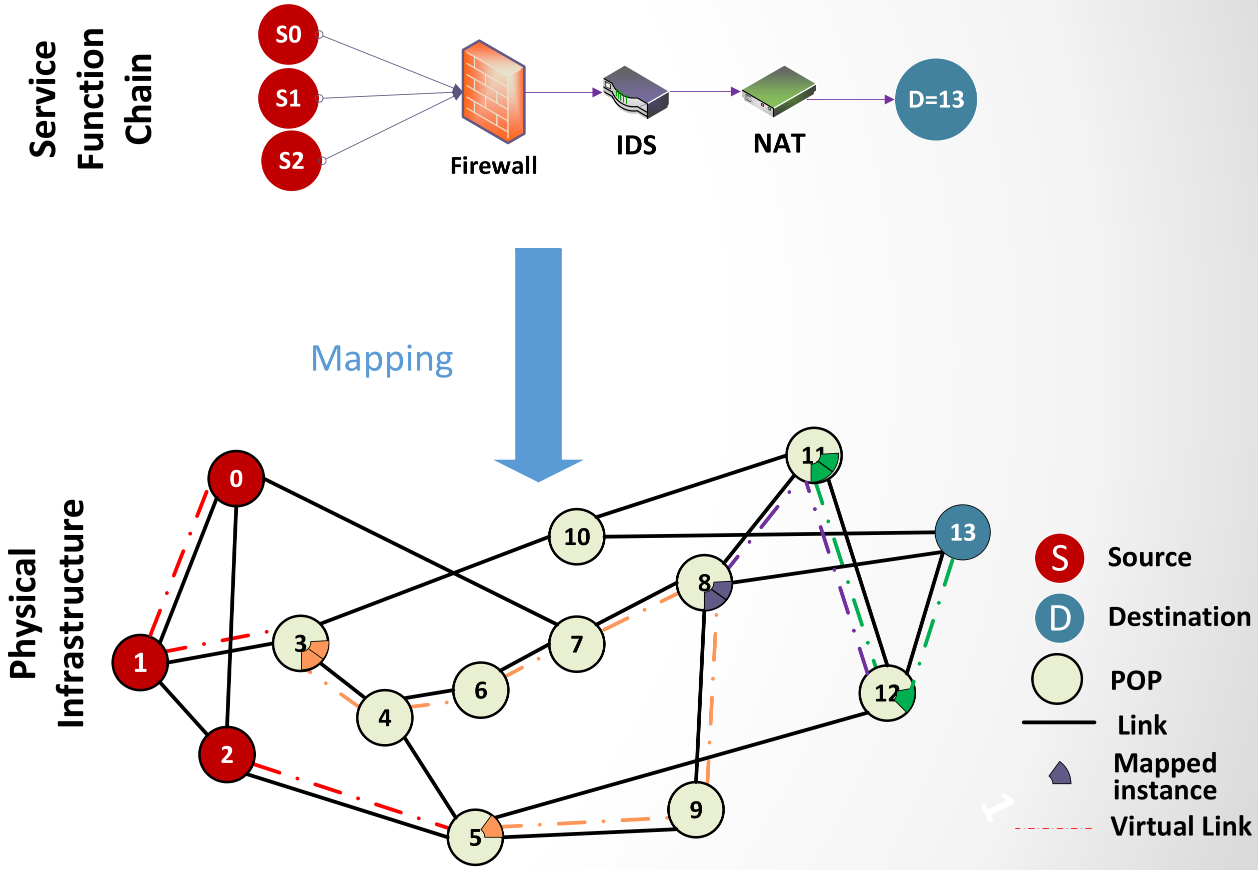
- Multiple connected network functions
- Multiple sources and destinations
- Made out from Network functions
- Defines, for each network function, the type, software, input/output packet format, expected processing delay, buffer size
- Defines performance requirements (e.g., throughput, packet loss, end-to-end delay, jitter)



Business Model

Network Operators

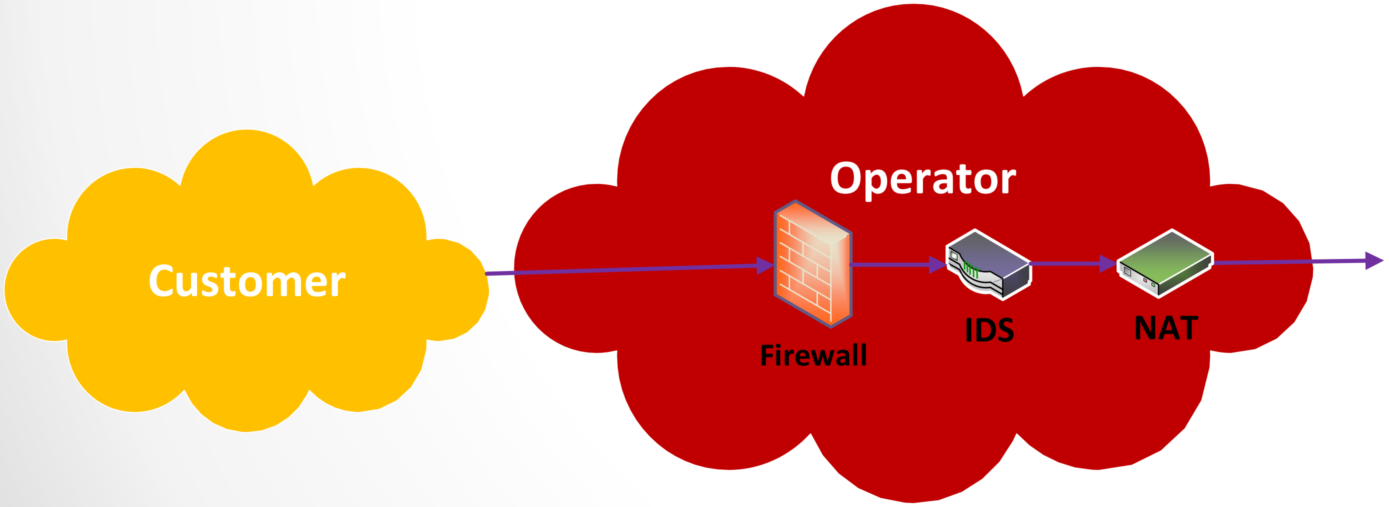
- Own and manage the physical infrastructure (i.e., one network)
- Deploy platforms and software required to run network functions
- The service could be simply data delivery or a SFC
- Provision and manage SFCs



Business Model (cont)

Customers

- Could be other network operators, companies or Institutions
- Define the required SFC and Identify the chain sources/destinations
- Rely on the operator to provision and manage the SFC and satisfy SLA

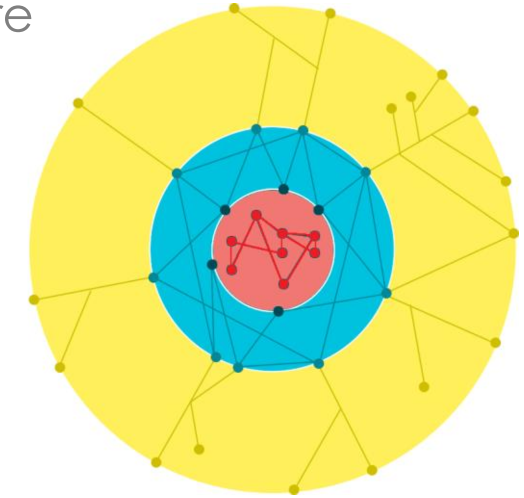


- SFC composition
- SLA requirements for the SFC
  - Bandwidth
  - End-to-end delay
  - Reliability, availability
- SLA requirements for each NFs
  - Processing power
  - Packet format(s)
  - Packet drop criteria…

ÉTS
Le génie pour l'industrie

Business Model (cont)

- Example of potential Network Operators:
  - ISPs (e.g., AT&T or Bell Canada) and web-scale companies (e.g., Google, Facebook, Amazon)
  - Example: Google Cloud Platform
    - World wide global Infrastructure
    - Software defined platform
    - Full control over the infrastructure



15 Data centers
100 Points of Presence (PoPs)
1000+ Edge nodes

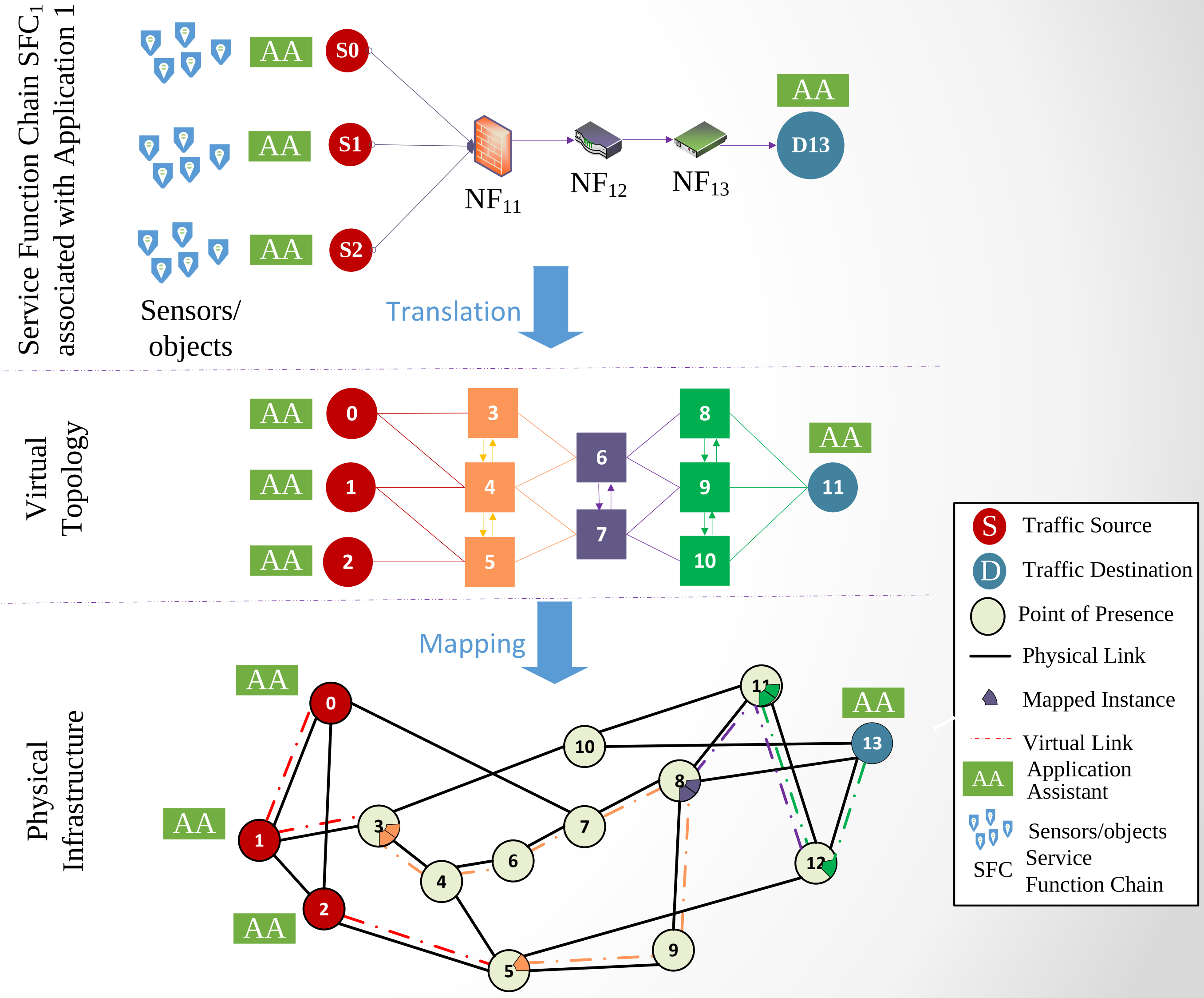
M. F. Zhani, H. ElBakoury - FlexNGIA 2019 (https://arxiv.org/abs/1905.07137)

Source: cloud.google.com

Resource Management Framework

Resource Allocation

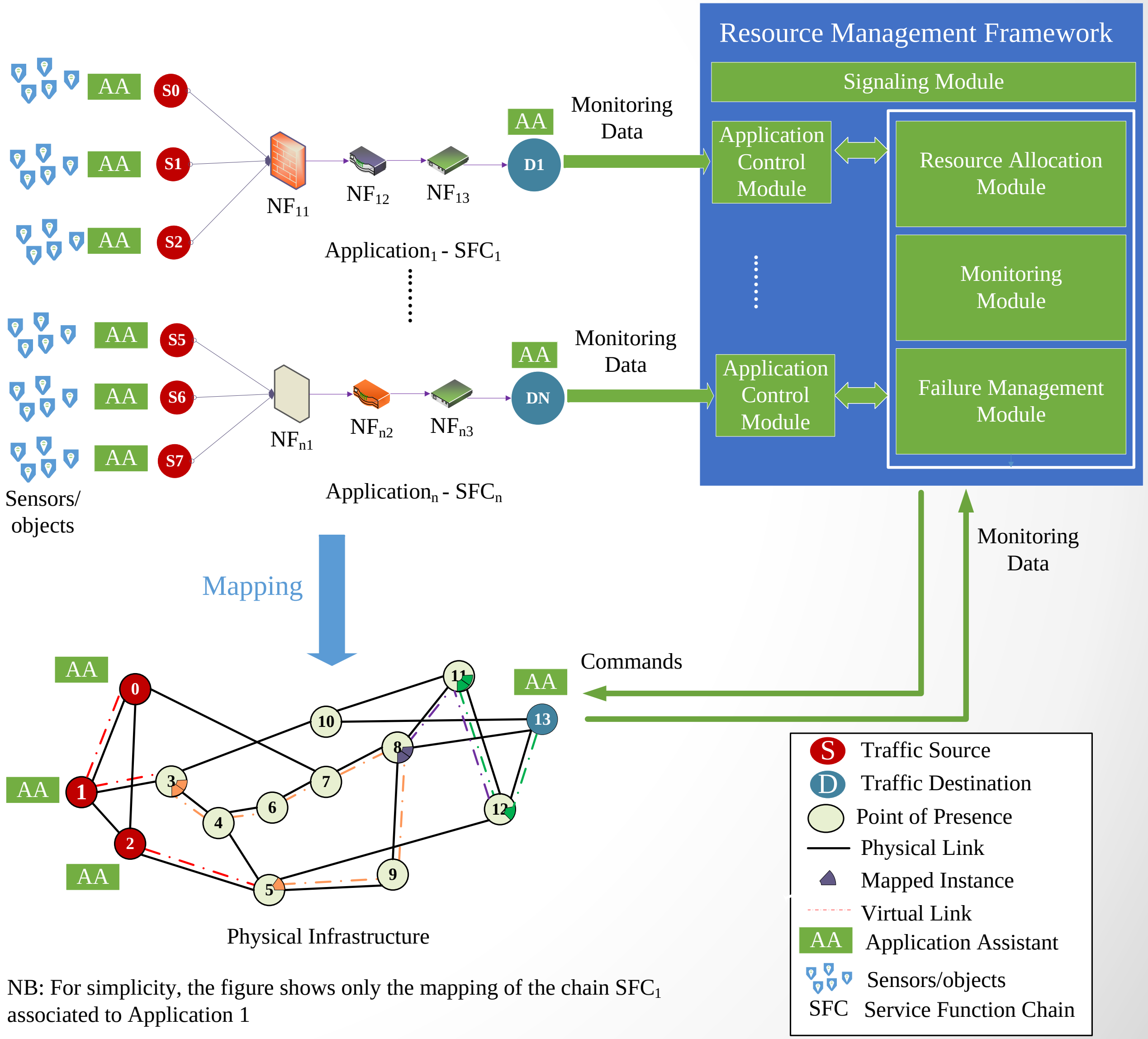
- The Service Function Chain (SFC) is defined by the application designer
- 2-step resource allocation:
  - Translation: the SFC is translated into a virtual topology
  - Mapping: virtual topology are mappa



Resource Management Framework

Application Control Module

- One module for each application
- Ensures application' requirements are satisfied at run-time
- Decides of the type of network functions, the number of instances and their resource requirements

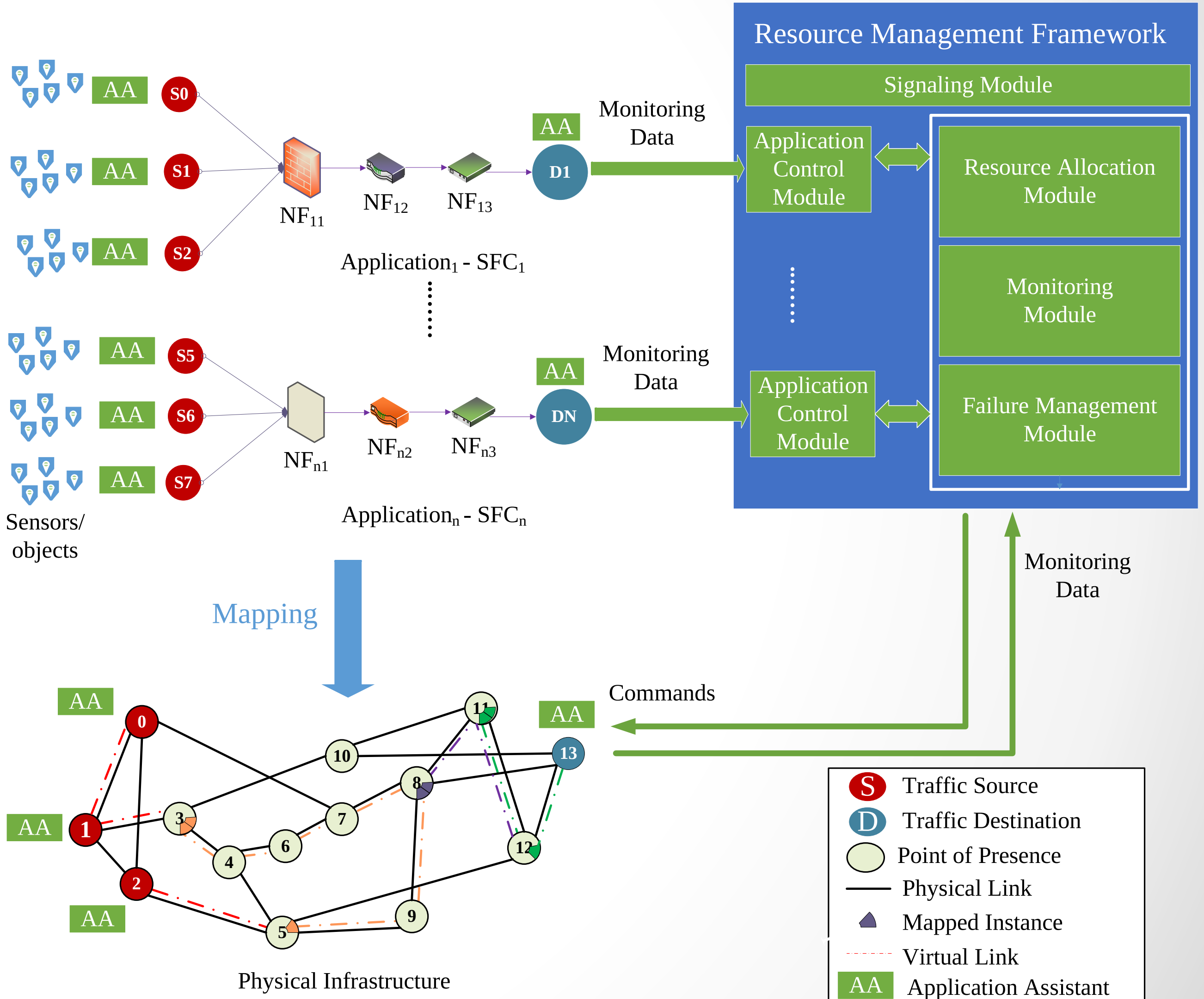


NB: For simplicity, the figure shows only the mapping of the chain SFC_1 associated to Application 1

Resource Management Framework

Ressource allocation Module

- Allocates resources requested by the application control module
- Achieves the network operator's high-level objectives (e.g., network utilization, energy efficiency)

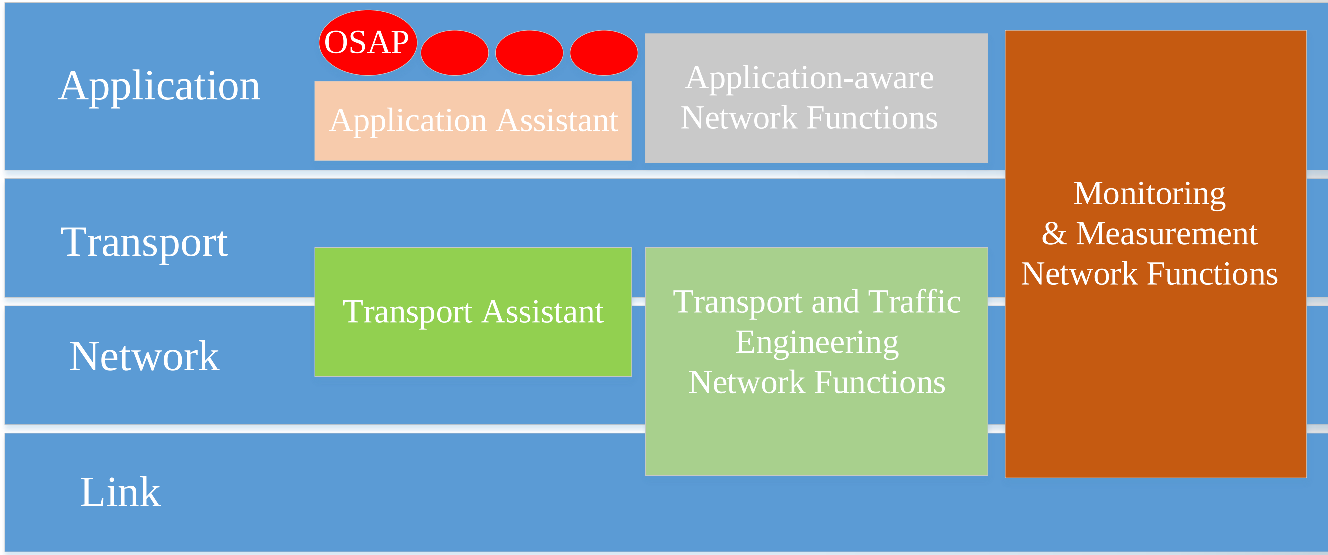


NB: For simplicity, the figure shows only the mapping of the chain SFC_1 associated to Application 1

M. F. Zhani, H. ElBakoury - FlexNGIA 2019 (https://arxiv.org/abs/1905.07137)

Network Protocol Stack/Functions

- Basic Network Functions (e.g., packet forwarding)
- Advanced Network Functions:
  - Could operate at any layer
  - Only limited by our imagination
  - Examples: packet grouping, caching and retransmission, data processing (e.g., image/video cropping, compression, rendering, ML), application-aware flow multiplexing (e.g., incorporating/merging data)



➔ Functions could break the end-to-end principle

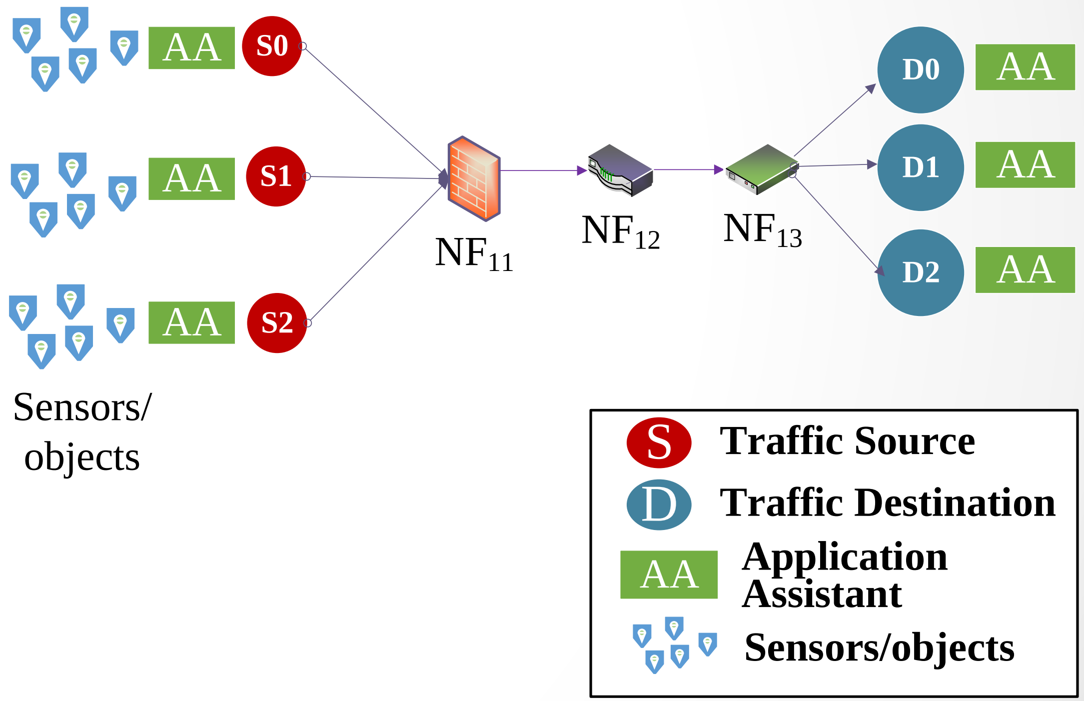
➔ SDN++: SDN should go beyond configuring forwarding rules and should provide the ability to dynamically configure these new functions

Network Protocol Stack/Functions Application Assistant

Application Assistant (AA)

- One AA at each end-point
- Interfaces with objects/sensors
- Measures the application performance and user QoE
- Identifies the applications’ requirements at run-time
- Adds additional metadata To be used by subsequent Network Functions

➔ Application-Aware Network Services



Network Protocol Stack/Functions Transport Assistant

ÉTS
Le génie pour l'industrie

Transport Assistant (TA)

- A cross-layer Network Function
- Combines services of the transport and network layers
- Manages all the flows of the same application
- Implements Transport/Network functions (e.g., congestion control, packet loss detection, packet cache and retransmission, routing)
- One or multiple TA could be provisioned in the same SFC

Application Layer

Transport Layer (TCP)
- E2E communication
- Blind congestion Control
- Inaccurate Packet Loss Detection
- Guaranteed Reliability
- E2E Packet Retransmission Process

Network Layer
- IP protocol (header and addressing)
- Routing Protocols/SDN
- ICMP for Control Information
- No Advanced Network Functions

Cross-Layer Transport
- Multi-point communication
- Network-assisted congestion control
- Network-assisted reliability and performance guarantees
- Accurate packet loss detection
- Variable performance and reliability Requirements over time
- Variable Header
- Meta-data and commands within packet headers
- Advanced Network Functions

Link Layer

Network Protocol Stack/Functions Transport Assistant (cont)

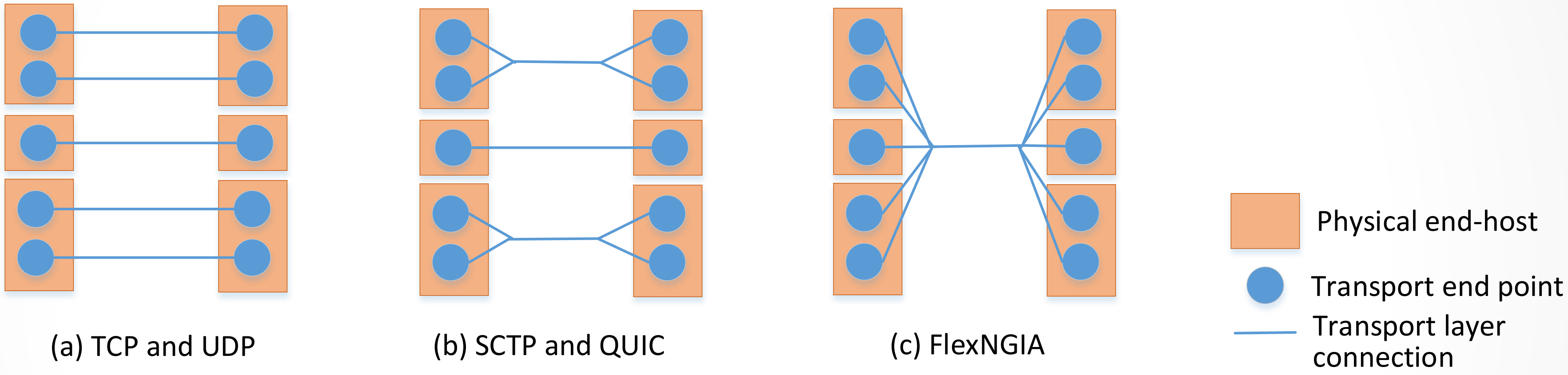


Illustration of how one single network application might be seen at the transport Layer

- Transport Assistants manage all these flows while taking into account that they all belong to the same application
- TAs monitor these flows, divide the total bandwidth allocated for the application among them.

M. F. Zhani, H. ElBakoury - FlexNGIA 2019 (https://arxiv.org/abs/1905.07137)

Network Stack Headers

- Signaling packets
  - Instantiate an application
  - Convey initial application requirements
- Data packets: carry data
  - Layer 2 header: contains mainly the application id used for packet forwarding (similar to VLANs)
  - Upper layers:
    - Fully flexible header format
    - Defined depending on the application
    - Network functions should be aware of the expected format

Network Stack Headers

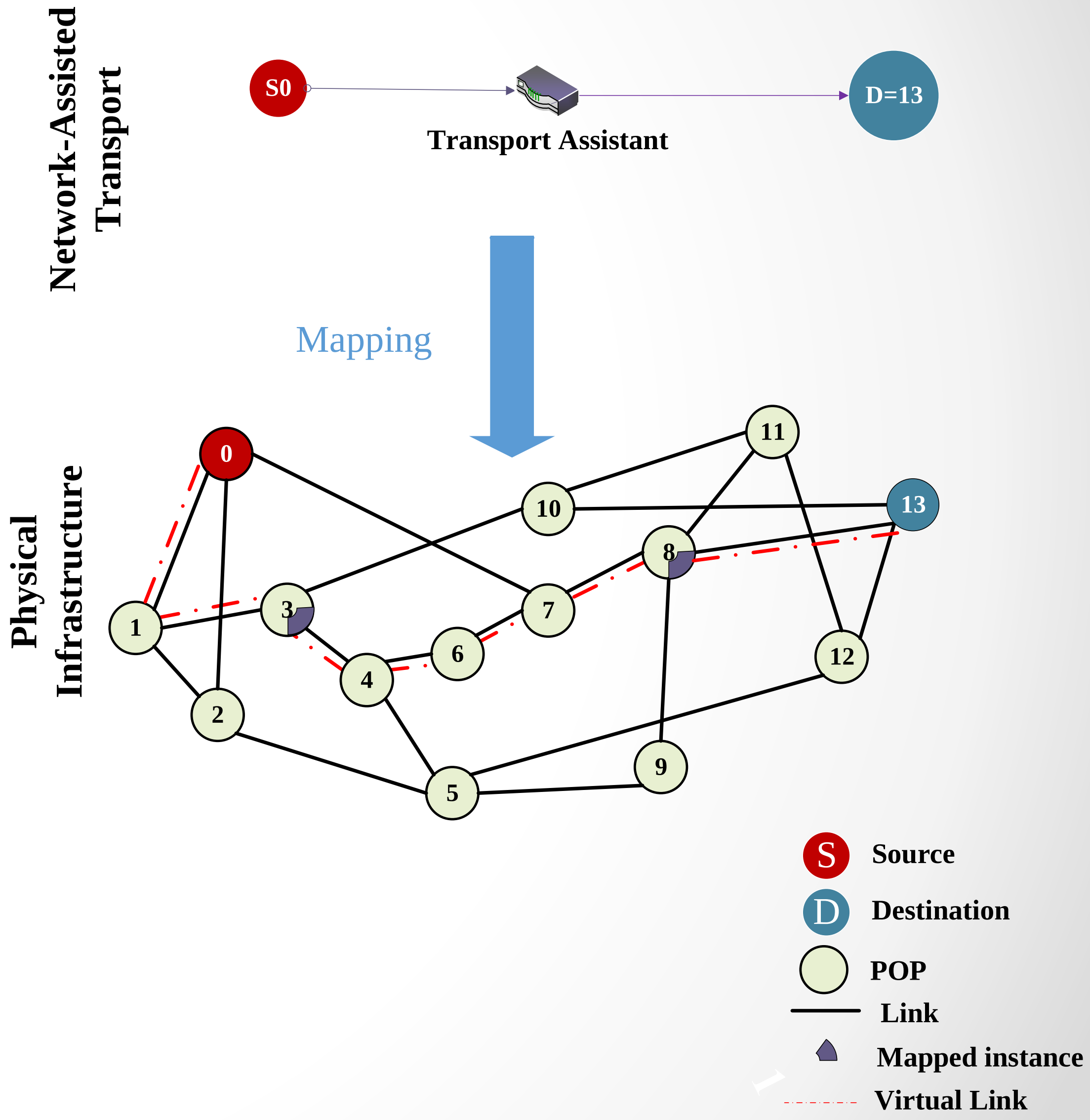
- Additional meta-data and commands should be included in the packet header
  - Examples of meta-data: Applications'id and its belonging flows, Object/sensor id, performance requirements (may be changing like the rate), data type, video encoding, layers, video/audio compression rate, quality, routing preferences
  - Example of commands : drop flow x first, cache packets of flow y, further compress flow z of type video if needed

Outline

- A Glance into the Future
- Limitations of Today's Internet
- FlexNGIA: Fully-Flexible Next-Generation Internet Architecture
- Use Cases
- Conclusion

Network-Assisted Data Transport

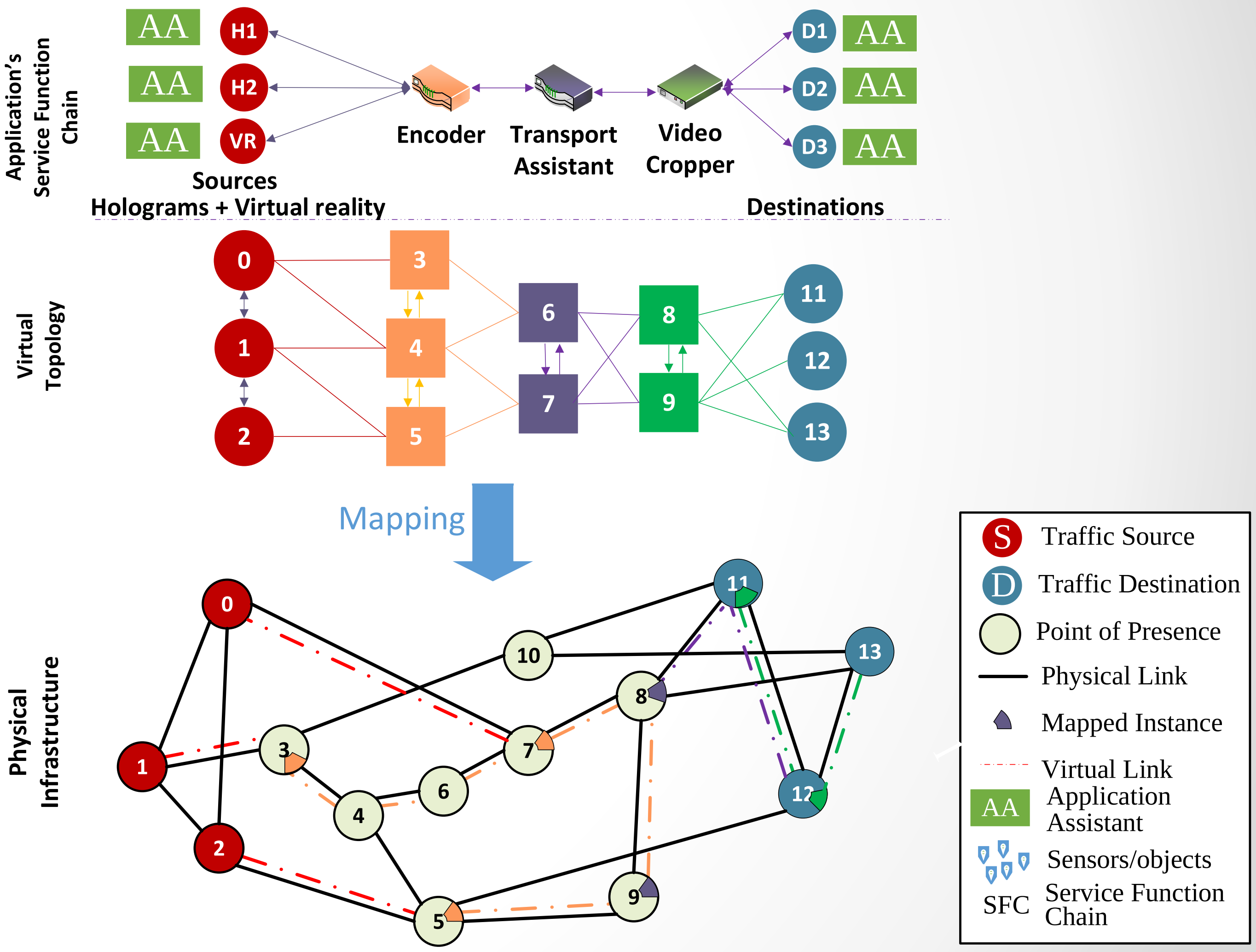
- Goal
  - Minimize retransmission delay
  - Improved congestion control
- Solution: service chain with a "transport Assistant" function
- Service of the Transport Assistant:
  - Caching and retransmissting packets
  - Detecting packet loss
  - Congestion control: adjusting rate, dropping packets, compression



M. F. Zhani, H. ElBakoury - FlexNGIA 2019 (https://arxiv.org/abs/1905.07137)

Mixed Virtual Reality and Holograms

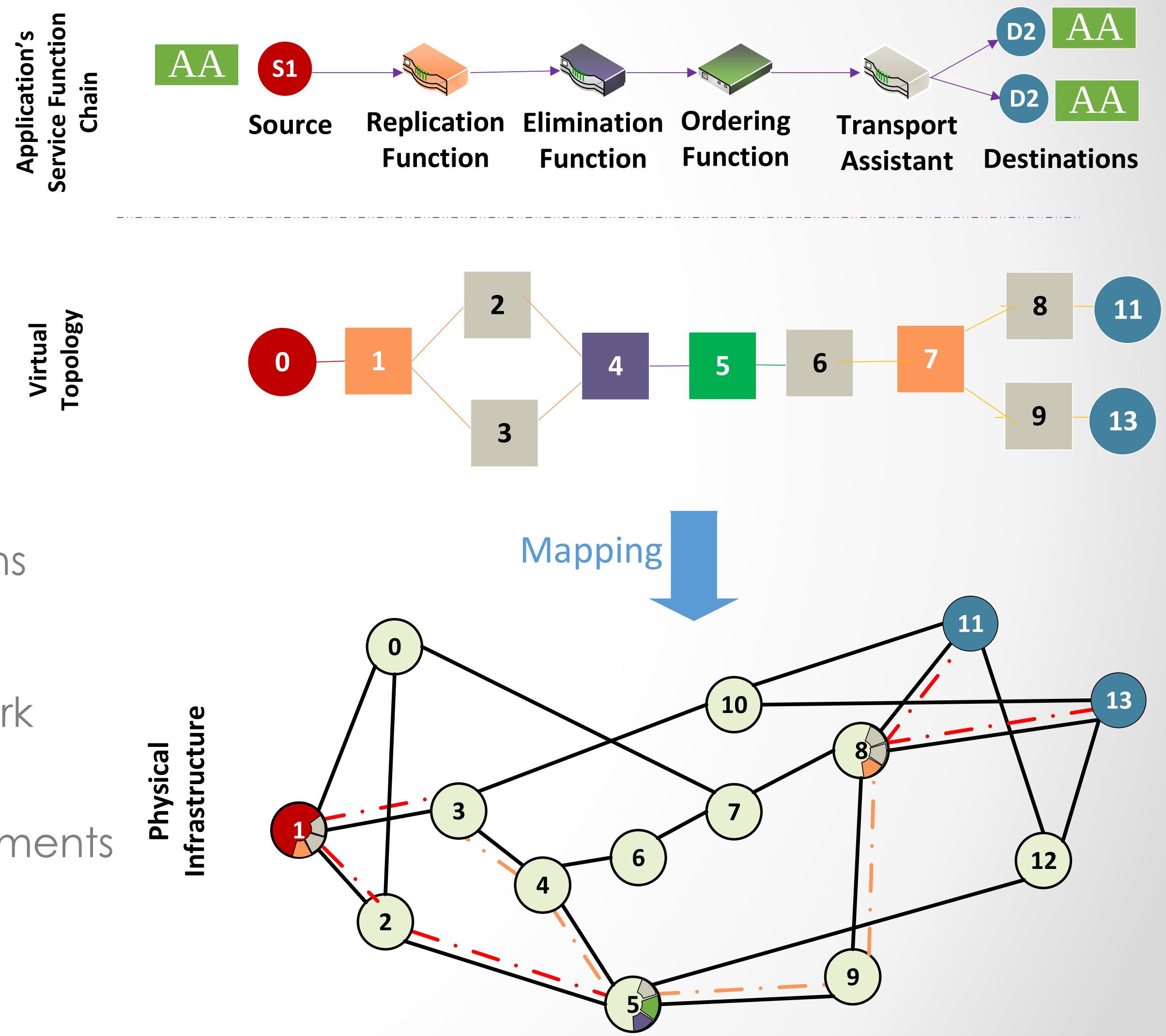
- Users are exploring a virtual reality environment with several human holograms and objects
- Challenges
  - How many intermediate functions?
  - What kind of functions?
  - How the traffic should steered from the flow sources?
  - How many instances for each function?
  - Where to place them?
- Example of deployement
  - Encoder: encode and compress video
  - Transport manager: congestion control
  - Video cropper: crop 3D objects



Determinsitic Networking

- Goal
  - Min and max end-to-end latency and jitter
  - Packet loss ratio
  - Upper bound on out-of-order packet delivery
- Existing solution: DetNet [IETF]
  - Resource allocation
  - Protection mechanisms: replicate data through different paths: 3 types of network functions
- DetNet over FlexNGIA:
  - DetNet Functions are deployed through the network
  - TA are also deployed to improve reliability
  - AA to evaluate at run-time the application requirements

➔ Delay guarantees + reliability + flexbility



Outline

- A Glance into the Future
- Limitations of Today's Internet
- FlexNGIA: Fully-Flexible Next-Generation Internet Architecture
- Use Cases
- Conclusion

Le génie pour l'industrie

Conclusion

FlexNGIA

Computing resources

- In-Network Computing: any function anywhere

Business model

- Multiple source destination Service Function Chains
- Stringent performance requirements

Cross-layer Design (Transport+Network)

- Breaking the end-to-end paradigm
- In-network advanced transport functions
- Better congestion control
- Stringent performance and reliability guarantees

Application-Aware Network Management

- Advanced functions tailored to applications
- App-aware traffic engineering

Flexible headers

- Tailored to the application

Looking for More Details?

- M. F. Zhani, H. ElBakoury, "*FlexNGIA: A Flexible Internet Architecture for the Next-Generation Tactile Internet,*"ArXiV 1905.07137, May 17, 2019 https://arxiv.org/abs/1905.07137

Thank You

Questions