Application-driven Virtual Network Slicing for Future Broadcast Core Network

Zuqing Zhu

University of Science and Technology of China

zqzhu@ieee.org

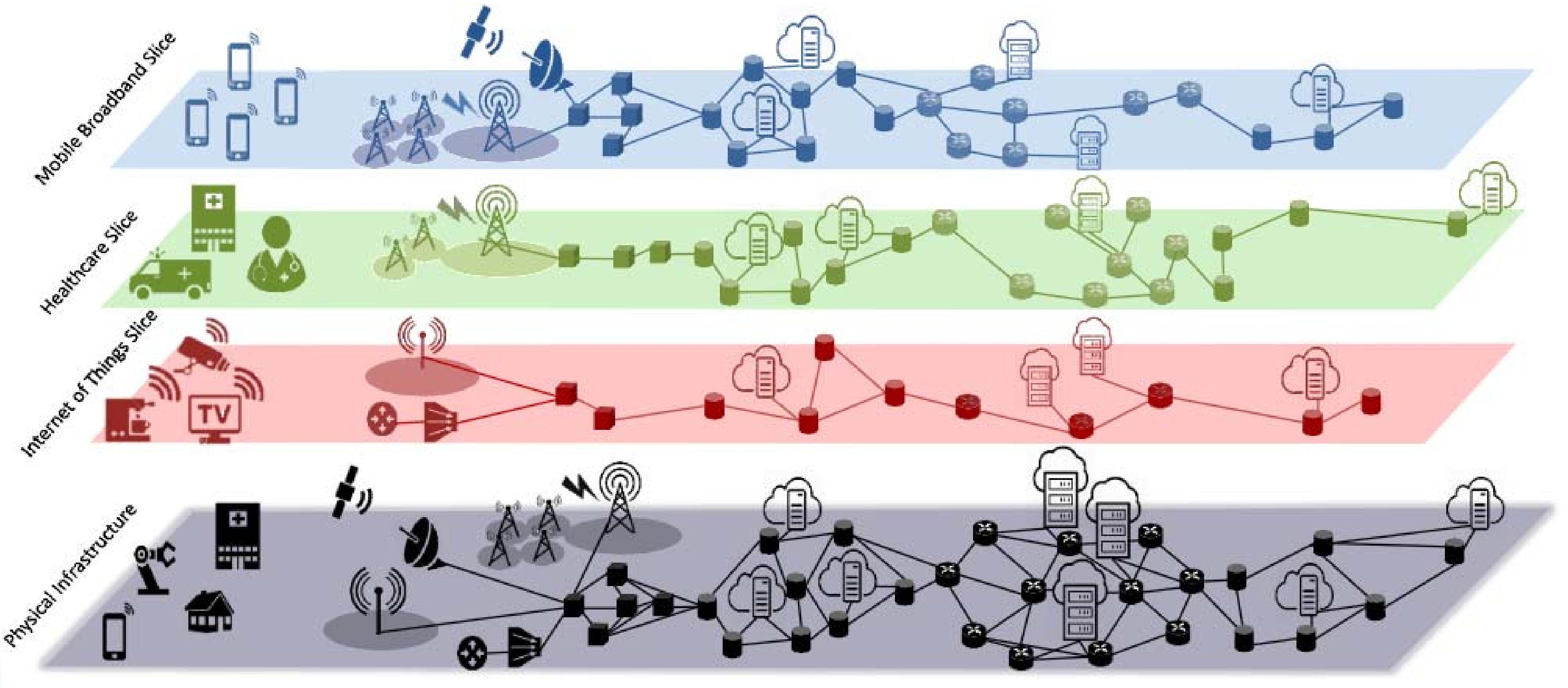
Future Broadcast Core Networks

- Fast Development
  - Global Internet users: 5.3 billion by 2023 (66% of total population).
  - Networked devices: 29.3 billion by 2023 (11% of them will be connected TVs).
- Architectural Scalability
  - Global traffic: annual growth rate of 25%, more than 32 ZB by 2023.
  - Average access speed: broadband at 110 Mbps, Wi-Fi at 92 Mbps, by 2023.
  - Now, ICT consumes more than 5% of total power used in the United States.
- Resource Utilization
  - Average utilization: Servers ~20%, Storage ~25%, Network ~30% (dynamic and unbalanced)
- Management Agility
  - Long lead time to deploy new services

[1] Cisco Annual Internet Report, 2018-2021, White Paper.
[2] Z. Kerravala, “A Data Center Fabric is Critical to a Next-Generation Unified Data Center,” Yankee Group White Paper.

Virtual Network Slicing Contributes a Potential Solution

- Virtualize network and IT resources in core networks.
- Create network slices dynamically to adapt to service demands.
- Make network slices share the same substrate resources but with proper isolation.



Enabling Techniques for Virtual Network Slicing

- Software-Defined Networking (SDN)
  - Separation of control and data planes to make the creation and management of virtual network slices easier, faster and more efficient.
- Data Plane
  - Network elements should enable resource virtualization and isolation.
  - Switches should be protocol independent for future-proof application support.
- Virtualization Layer
  - Sit in between the control and data planes.
  - Virtualize, allocate and isolate substrate resources in the data plane to build virtual network slices according to applications' QoS demands.
- Control Plane
  - Have the adaptivity to provision services in virtual network slices cost-effectively in dynamic network environments.
  - Can leverage artificial intelligence (AI) to achieve smart and timely decisions.

Data Plane: Limitations of OpenFlow

```
struct ofp_match {
   uint32_t wildcards;
   uint16_t  in_port;
   uint8_t   dl_src[]
   uint8_t   dl_dst[]
   uint16_t dl_vlan;
   uint8_t   dl_vlan_pcp
   uint16_t dl_type;
   uint8_t    nw_tos;
   uint8_t    nw_proto;
   uint32_t nw_src;
   uint32_t nw_dst;
   uint16_t tp_src;
   uint16_t tp_dst;
};
```

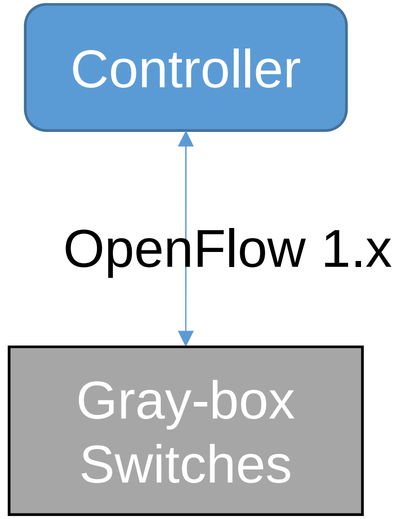
OpenFlow 1.0

```
enum oxm_ofb_match_fields {
   OFPXMT_OFB_IN_PORT
   OFPXMT_OFB_IN_PHY_PORT
   OFPXMT_OFB_METADATA
   OFPXMT_OFB_ETH_DST
   OFPXMT_OFB_ETH_SRC
   OFPXMT_OFB_ETH_TYPE
   OFPXMT_OFB_IPV4_SRC
   OFPXMT_OFB_IPV4_DST
         ……
   OFPXMT_OFB_IPV6_SRC
   OFPXMT_OFB_IPV6_DST
   OFPXMT_OFB_IPV6_FLABEL
   OFPXMT_OFB_IPV6_ND_TARGET
   OFPXMT_OFB_IPV6_ND_SLL
   OFPXMT_OFB_IPV6_ND_TLL
   OFPXMT_OFB_MPLS_LABEL
   OFPXMT_OFB_MPLS_TC
};
```

OpenFlow 1.2

```
enum oxm_ofb_match_fields {
 OFPXMT_OFB_IN_PORT
   OFPXMT_OFB_IN_PHY_PORT
   OFPXMT_OFB_METADATA
   OFPXMT_OFB_ETH_DST
   OFPXMT_OFB_ETH_SRC
   OFPXMT_OFB_ETH_TYPE
   OFPXMT_OFB_IPV4_SRC
   OFPXMT_OFB_IPV4_DST
         ……
   OFPXMT_OFB_IPV6_SRC
   OFPXMT_OFB_IPV6_DST
   OFPXMT_OFB_IPV6_FLABEL
   OFPXMT_OFB_IPV6_ND_TARGET
   OFPXMT_OFB_IPV6_ND_SLL
   OFPXMT_OFB_IPV6_ND_TLL
   OFPXMT_OFB_MPLS_LABEL
   OFPXMT_OFB_MPLS_TC
   OFPXMT_OFP_MPLS_BOS
   OFPXMT_OFB_PBB_ISID
   OFPXMT_OFB_TUNNEL_ID
   OFPXMT_OFB_IPV6_EXTHDR
};
```

OpenFlow 1.3



Data plane: Protocol-dependent and limited programmability.

Number of match fields in OpenFlow 1.x is increasing fast to adapt to more protocols!

OpenFlow-based data plane is NOT future-proof.

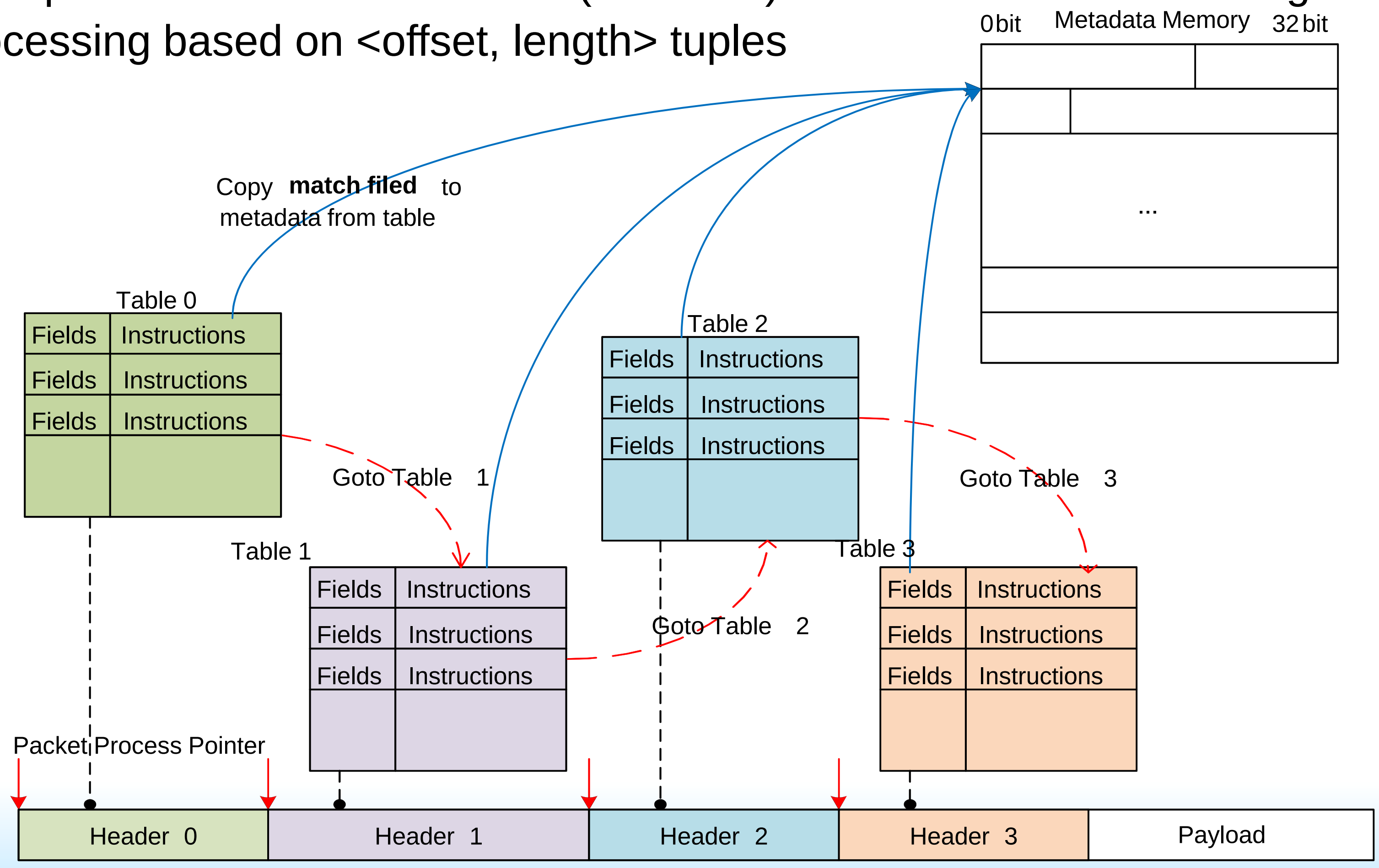
Programmable Data Plane (PDP)

- Protocol Independent Forwarding (PIF)
  - Open Networking Foundation (ONF) project with two approaches: P4 and POF.
- P4: Programming Protocol-Independent Packet Processors
  - Provide guidance on how to write and compile packet processing programs.
  - Program a switch in two stages: configuration and runtime, *i.e.*, a P4-based PDP switch needs to go offline to update packet processing pipelines.
  - Hardware solution: Barefoot switches with Tofino ASIC; Software solution: BMv2 software switch.
- POF: Protocol-Oblivious Forwarding
  - Provide guidance on what the underlying primitive instruction set should be.
  - Program a switch in runtime by installing flow tables (similar to OpenFlow), *i.e.*, a POF-based PDP switch can update packet processing pipelines dynamically.
  - Only has software solution, but hardware solution is not available.

Protocol-Oblivious Forwarding (POF)

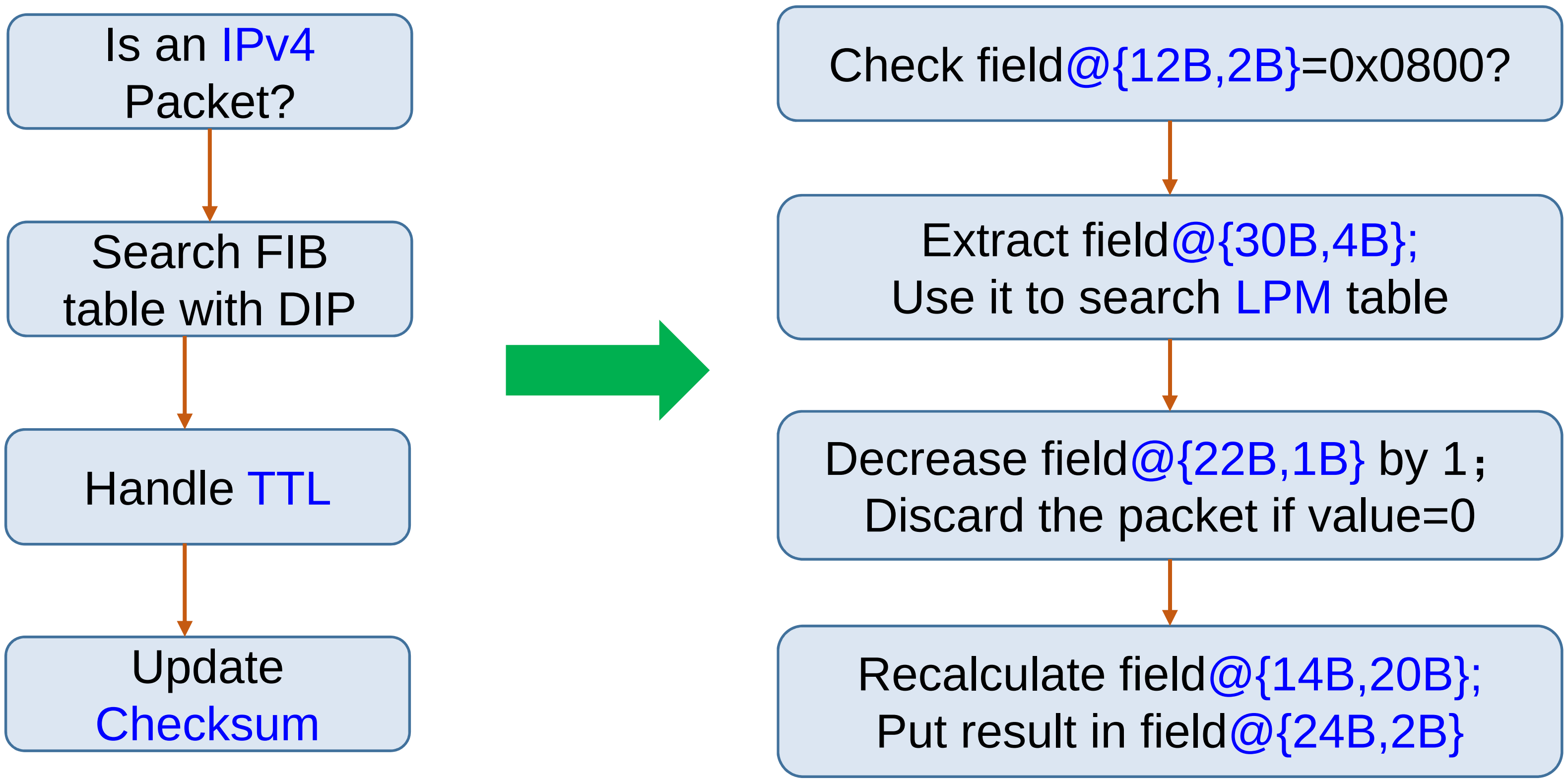
The data plane with POF is completely protocol-independent

- POF locates data fields in packets through <offset, length> tuples without a protocol parser
- POF provides an instruction set (POF-FIS) to facilitate table matching and packet processing based on <offset, length> tuples



An Example on POF

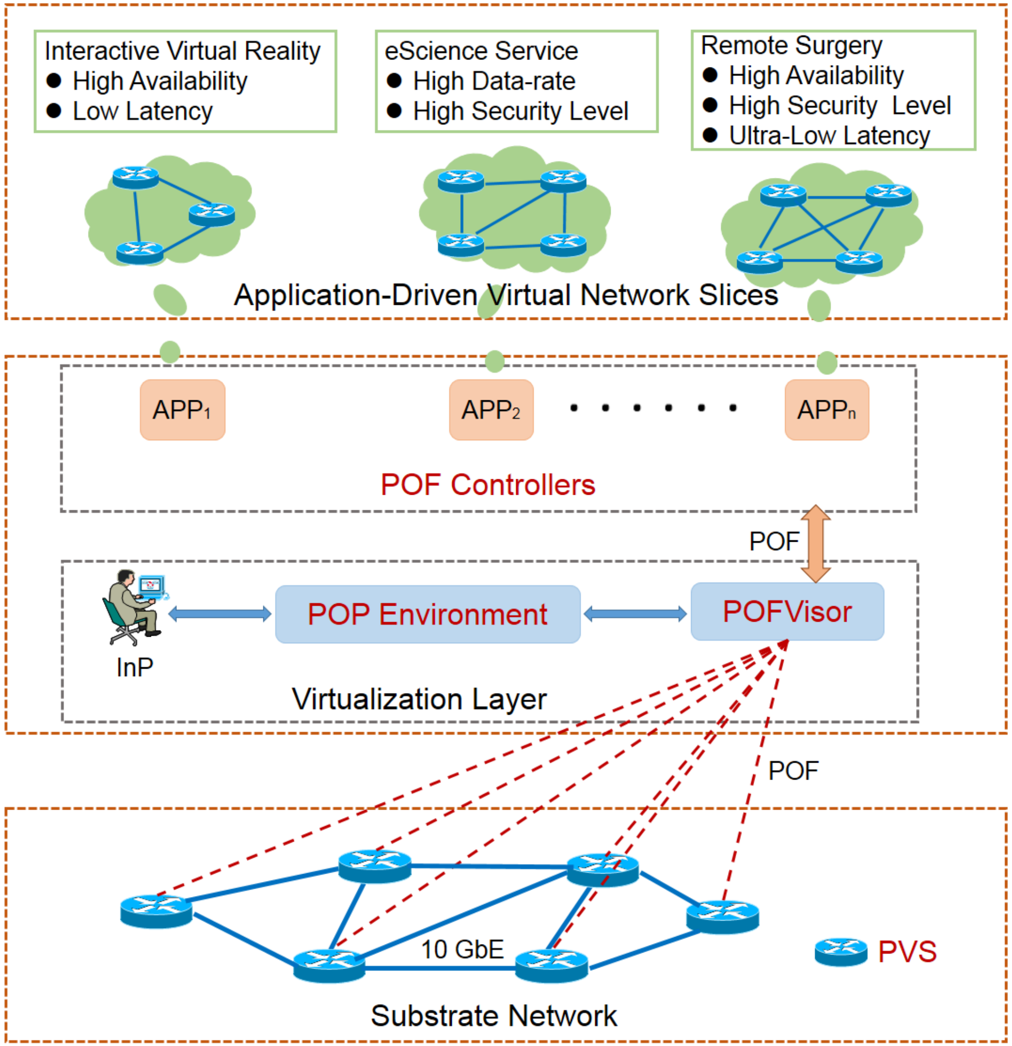
To Process and Forward an IPv4 Packet



As a protocol-independent protocol, POF does not need to know “what” but only need to know “how”.

POF-based Virtual Network Slicing System

- Control Plane:
  - POF Controller
- Virtualization Layer:
  - POFVisor
  - POF Programming (POP) Environment
- Data Plane:
  - High-Throughput POF Software Switch (PVS)



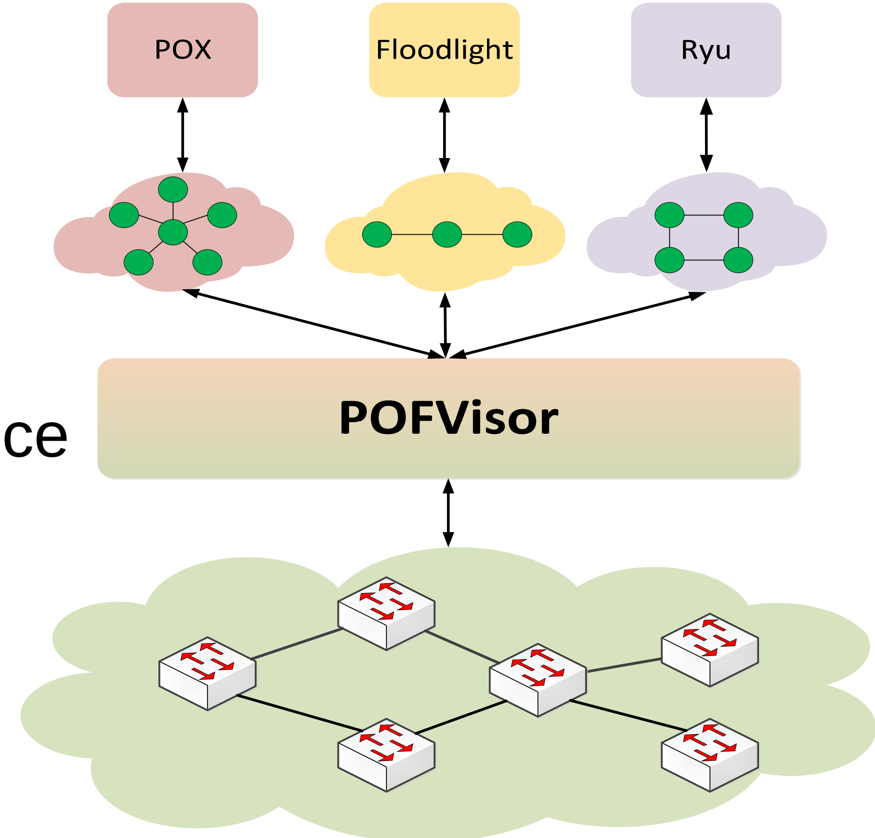
POFVisor: Network Virtualization Hypervisor

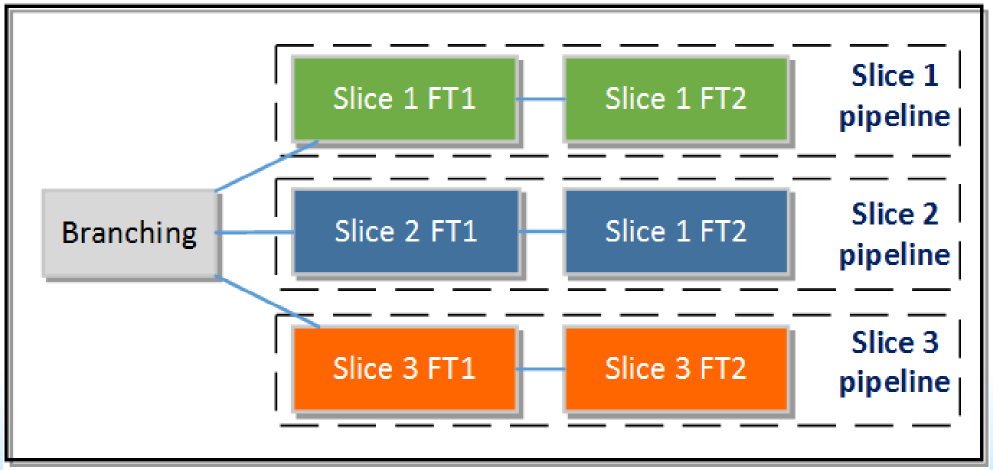
Isolation / QoS guarantee

- Isolation via pipeline branching in substrate switch
- Per slice QoS management by metering individual pipeline

Deep programmability for each slice

- Full protocol header manipulation
- Self-defined header structures and sequences



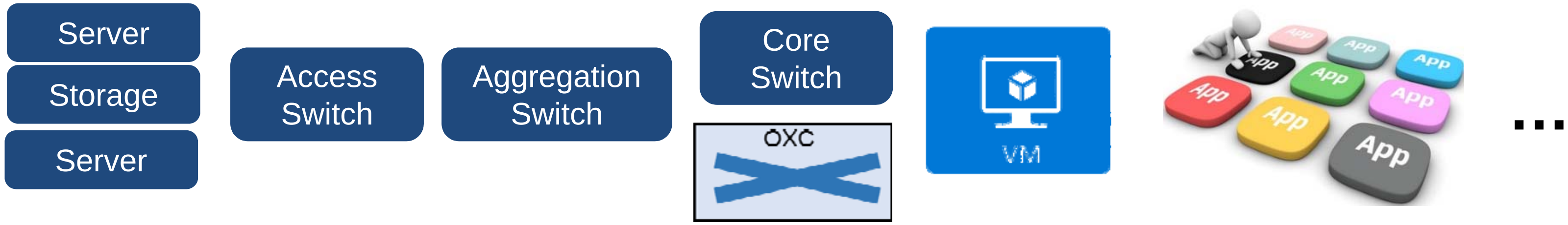


Agile virtual-to-substrate network mapping

- One-to-many / many-to-one
- Big switch

Intelligent Control Plane for Application-driven Virtual Network Slicing

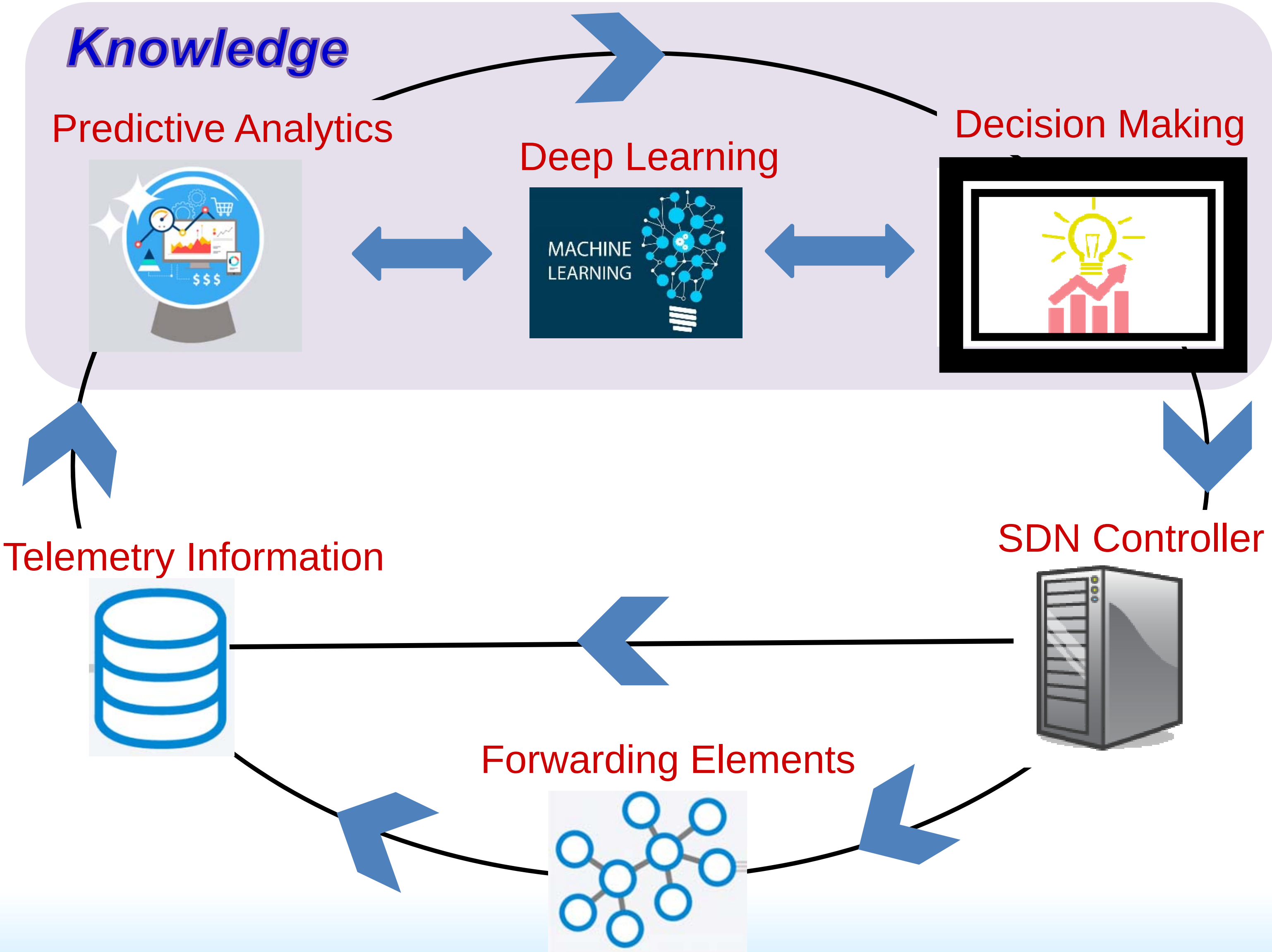
- Control plane needs to manage a large variety of network elements for virtual network slicing



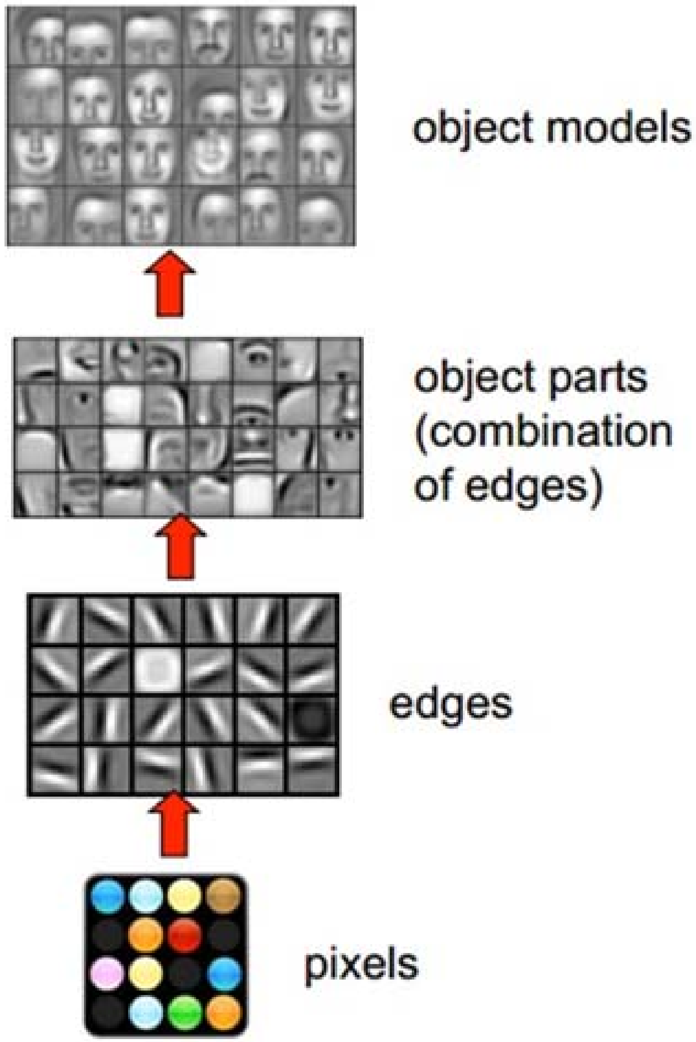
- Control plane needs to make automatic and timely decisions to ensure various QoS/QoE to applications
  - Latency is an important metric for many applications: Global average fixed latency of cloud services 31 msec, Asia Pacific 21 msec, Europe 27 msec [1]
  - A 1 msec advantage in trading applications can be worth $100 million a year to a major brokerage firm [5].

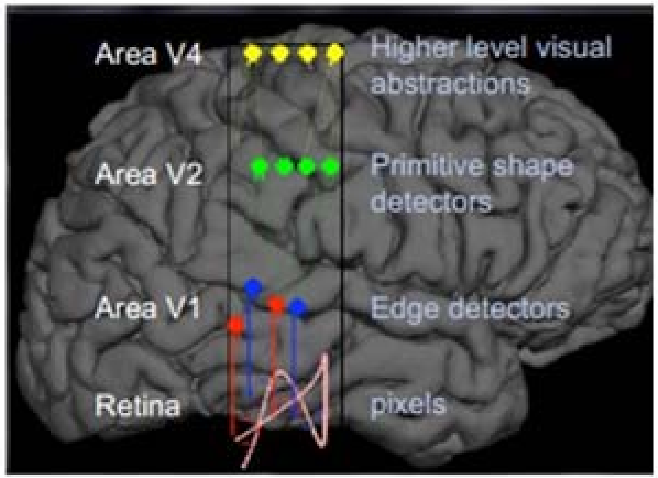
[1] Cisco Global Clould Index, Forecast and Methodology, 2016-2012 White Paper.
[5] Information Week Magazine.

Knowledge-defined Control Plane



Human Brain: Levels of Knowledge





Decision Making

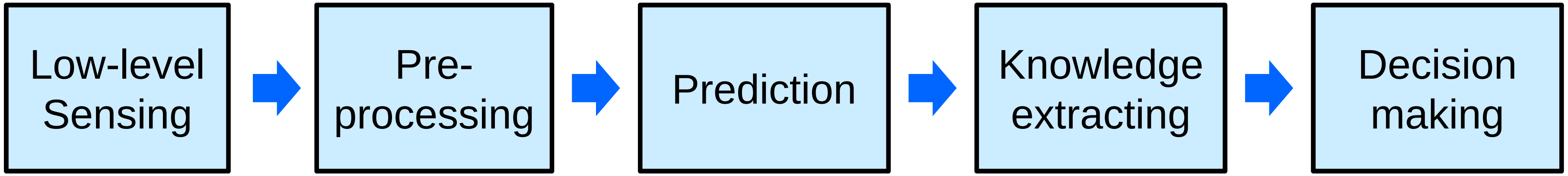
High-level Knowledge
Extraction

Low-level Knowledge
Extraction

Low-level Sensing

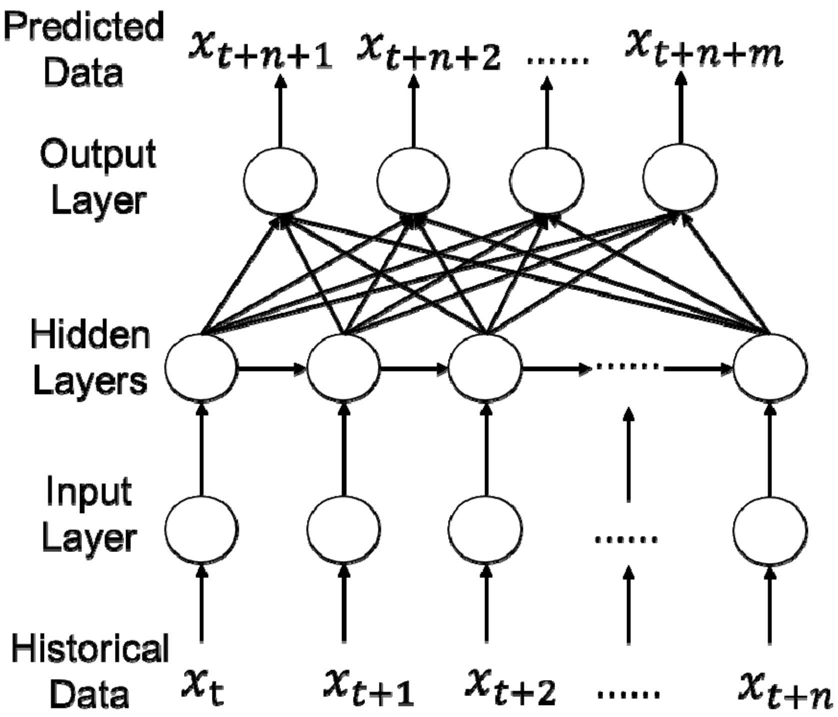
Predictive Analytics

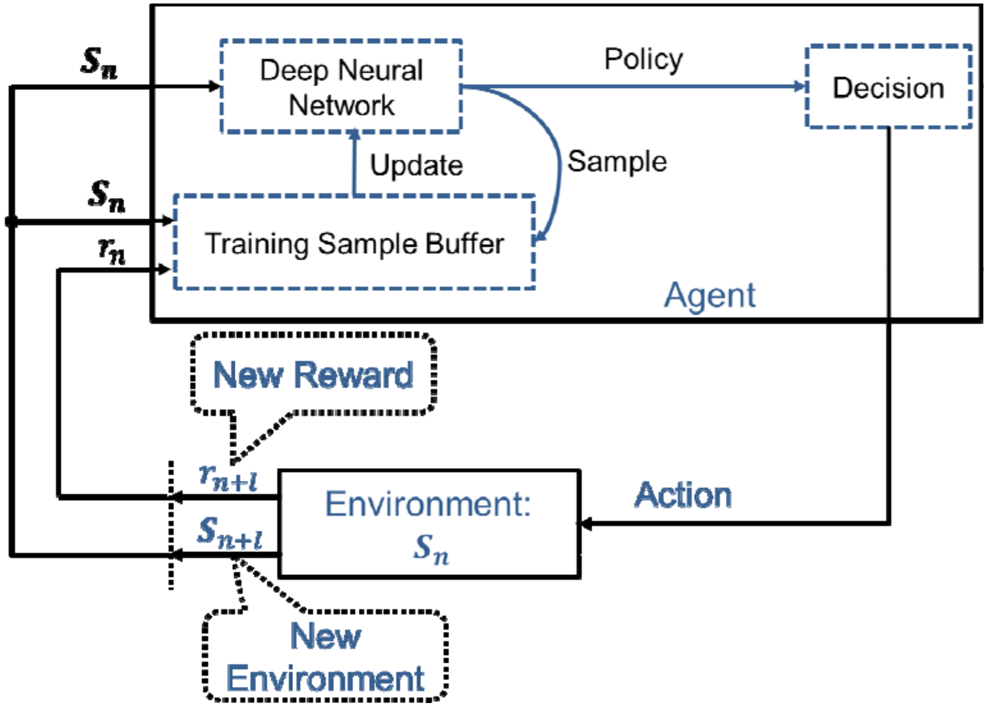
- Predictive analytics: Forecast based on memory + Decision making based on knowledge.



- We design the control plane to include multiple AI modules
  - AI modules coordinate to handle the complex tasks of knowledge-defined network orchestration (KD-NO).
  - AI modules extract different levels of knowledge from the telemetry data collected in the substrate network.
  - Modular design, better scalability, and easier to train and maintain the AI modules.

AI：Deep Learning and Deep Reinforcement Learning





Deep Learning
- Offline Training,
- Online Transfer Learning
- Prediction

Deep Reinforcement Learning
- Online Training
- Decision Making

TensorFlow

Preprocessing and Prediction: Low-level Knowledge Extraction

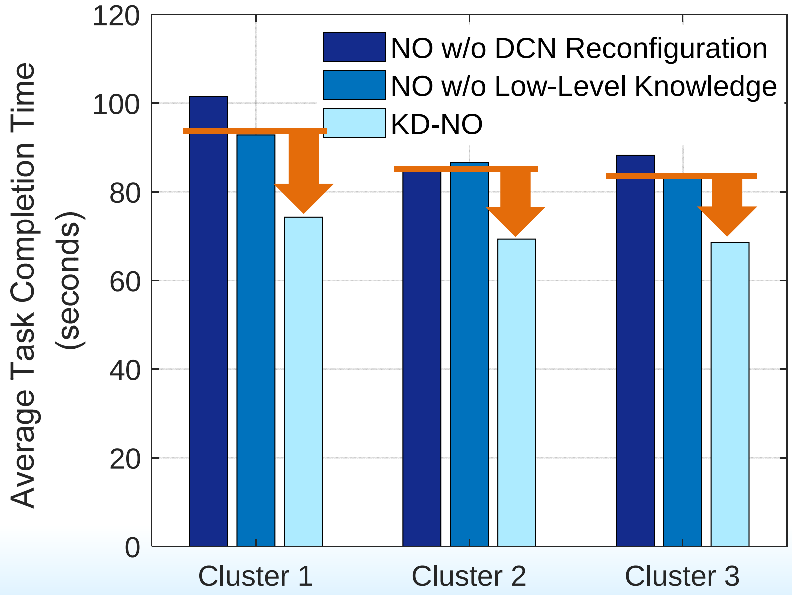
- Analyzing telemetry data of all the VMs and network elements and managing them accordingly would not be feasible.
  - Numerous VMs and network elements in virtual network slices
  - Huge volume of telemetry data to analyze, if we want full coverage
- Preprocessing:
  - Screening telemetry data to find the "major contributors"
  - Determining predictability by checking the auto-correlation of data
  - Minimizing allocated predictors with cross-correlation: traces with strong cross-correlation can be forecasted with the same predictor.
- Prediction:
  - Offline training: only train the necessary predictors with historical traces
  - Online transfer learning: light-weighted retraining with online data

High-level Knowledge Extraction

- High-level knowledge:
  - Matching degree between the embedding schemes of virtual network slices and the applications running in them.
- Hot-spots:
  - Highly-loaded servers and congested switch ports in the substrate network.

Experimental Scenarios and Parameters

- Run Hadoop applications in 9 VMs that belong to three Hadoop clusters (*i.e.*, three virtual network slices).
  - Each cluster processes tasks generated according to Google traces.
- Scale original 24-hour task pattern down to 12 minutes
  - Tasks: CPU-bound ones (high CPU usage, low I/O and traffic loads), and I/O-bound ones (high I/O and traffic loads, low CPU usage)

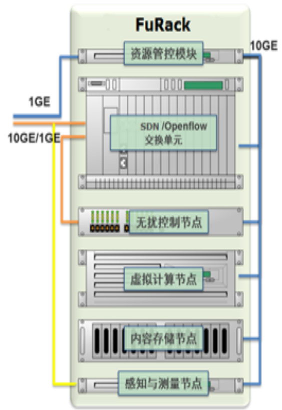


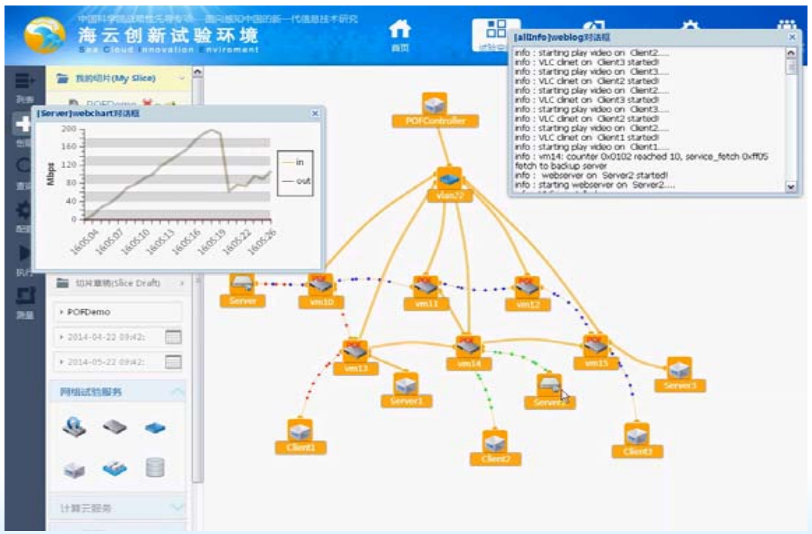
KD-NO reduces task completion time effectively.

FuNET: POF-enabled WAN Testbed

FuNET - Provide experimental services with sliced and virtualized WAN resources.

- **Software**
  - FuOS
  - FuES
- **Devices**
  - FuRack
- **Key technologies**
  - POF
  - SDN/NFV

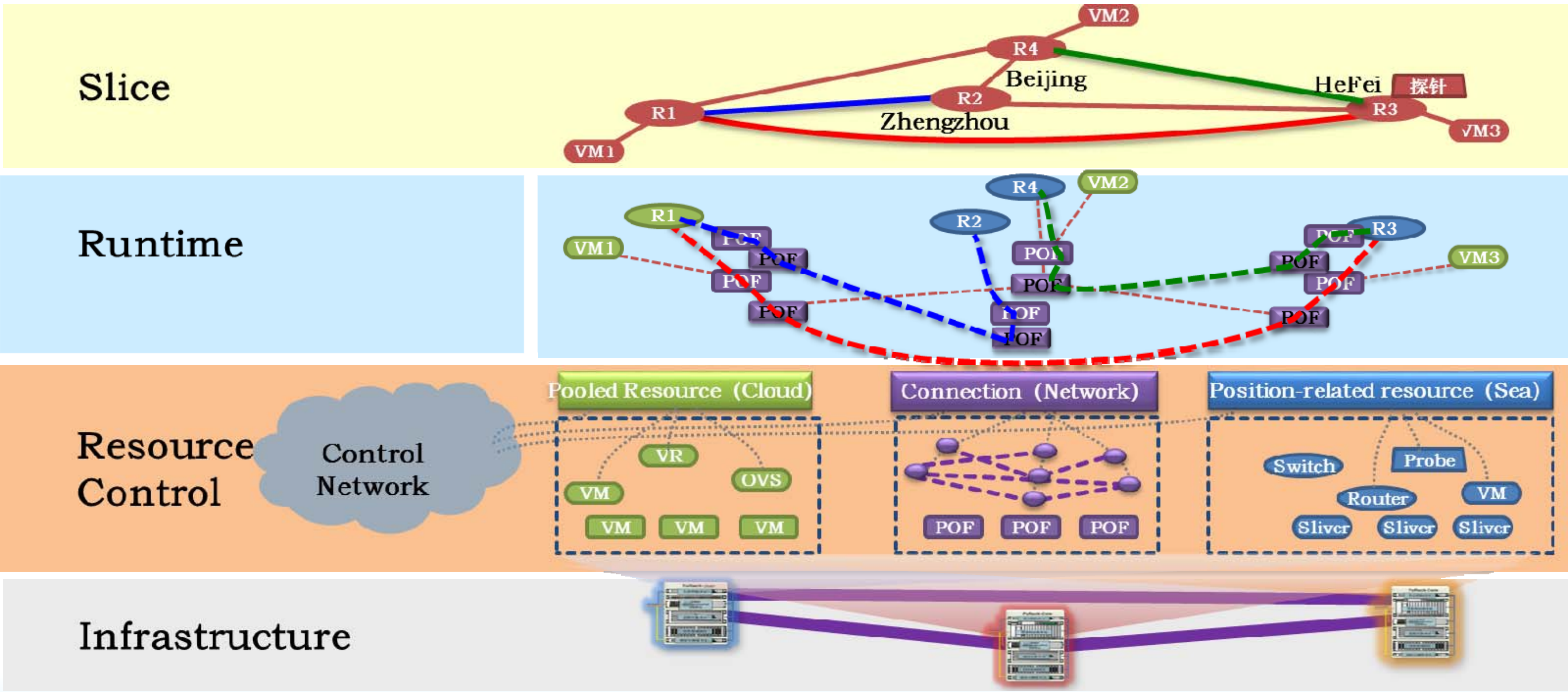




FuNET: POF-enabled WAN Testbed

❖ System Functionalities

- Mapping between virtual and substrate resources.
- Generate and schedule virtual network slices on demand.



Takebacks

- We discuss recent advances on the network slicing technologies for effectively supporting applications in future networks.
- To satisfy stringent QoS demands of applications, we cover innovations in data plane, virtualization layer, and control plane.
- For data plane, we introduce the PDP techniques that can make packet processing and forwarding protocol-independent.
- We show the open-source platform that can make virtualization layer work seamlessly with PDP.
- We implement AI-based techniques in control plane, to reach smart and timely orchestration decisions.

http://www.zuqingzhu.info
zqzhu@ieee.org