Deeply Programmable Network (DPN) and Advanced Network Virtualization
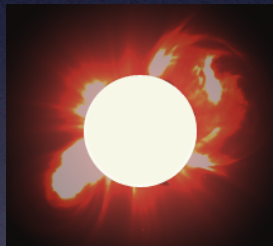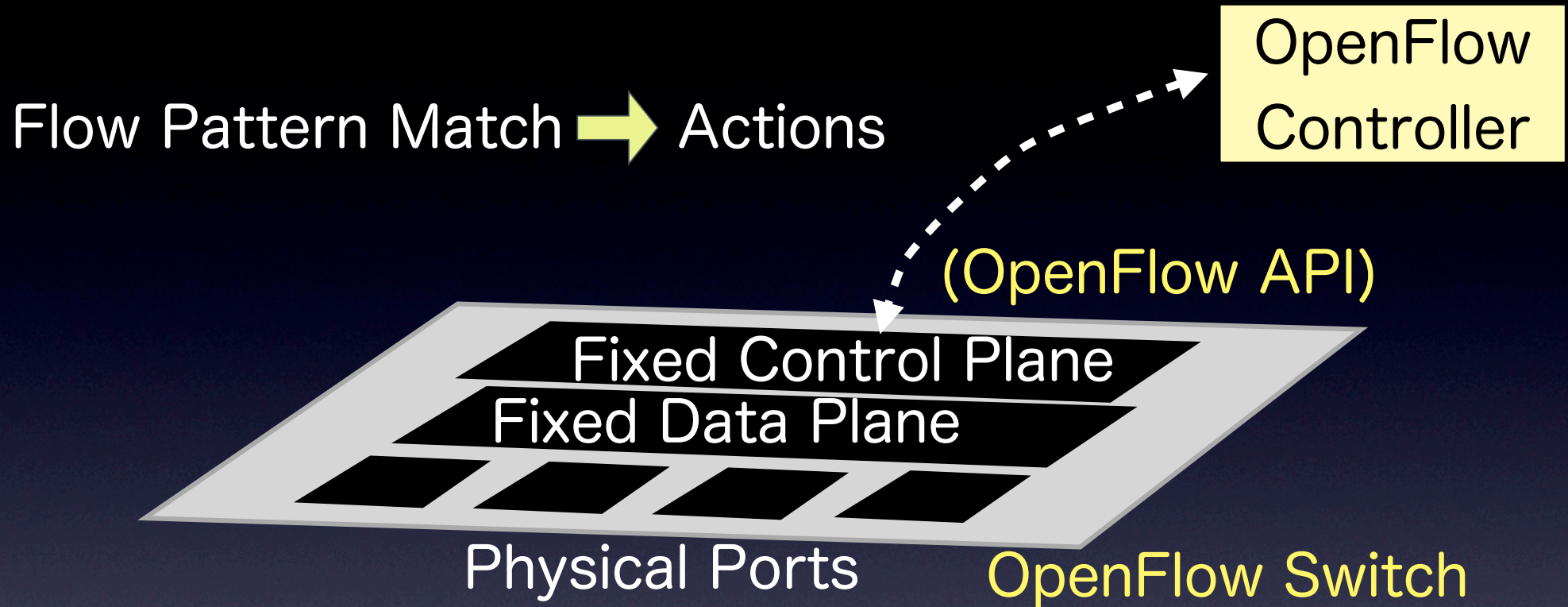

Aki Nakao

The University of Tokyo

2012/11/27

OpenFlow ∈ SDN

Flow Pattern Match ➡ Actions

OpenFlow Controller

(OpenFlow API)

Fixed Control Plane
Fixed Data Plane

Physical Ports   OpenFlow Switch

Although OpenFlow enables flexible control of flows, we want more:

- Complex packet processing / Non-IP protocols handling
- New classification rules based on more than 12 tuples
- Proprietary API definition / dynamic reprogramming of APIs
- Rapid prototyping of future network node...

2

For some of us, OpenFlow/API is at the right level of abstraction...

However...for some of us,

- We should be able to extend API for complex actions
- OpenFlow may be forcedly used in an inefficient manner, e.g.,
  - copying L7 bits to MAC for control based on those bits...
  - parsing tuples in a non-standard way to implement VXLAN etc.

Extending SDN Further?

- **Control-Plane Programmability**
  - Route Control
  - Access Control
  - Network Management
- **Data-Plane Programmability**
  - Packet Data Processing
    - Cache
    - Transcode
    - DPI
  - Handling New Protocols
    - IPvN (N>6)
    - New Layer2
    - Content Centric Network (CCN)
- **Meta-Control-Plane Programmability**
  - Defining new proprietary APIs

Target Scope of OpenFlow SDN

OpenFlow with external processors

Irrelevant with OpenFlow (+processors)

Out of scope of OpenFlow
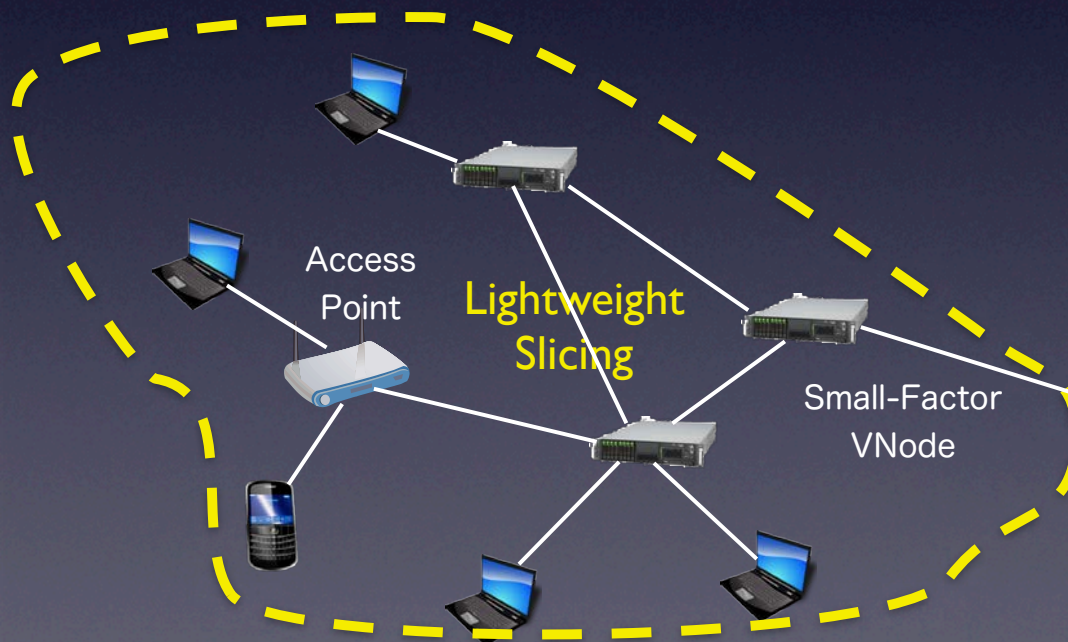
Scope of **Deeply Programmable Network (DPN)**
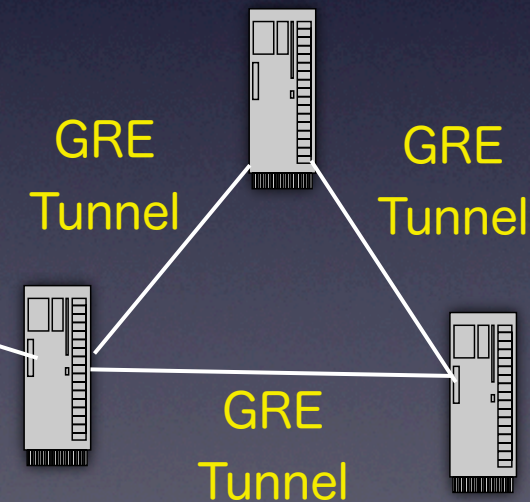
Deeper Programmability

Deep Programmability for Network Edge

- "Tangible" small-form-factor (1U) VNode
- Deeply programmable, even at L2, yet high performance
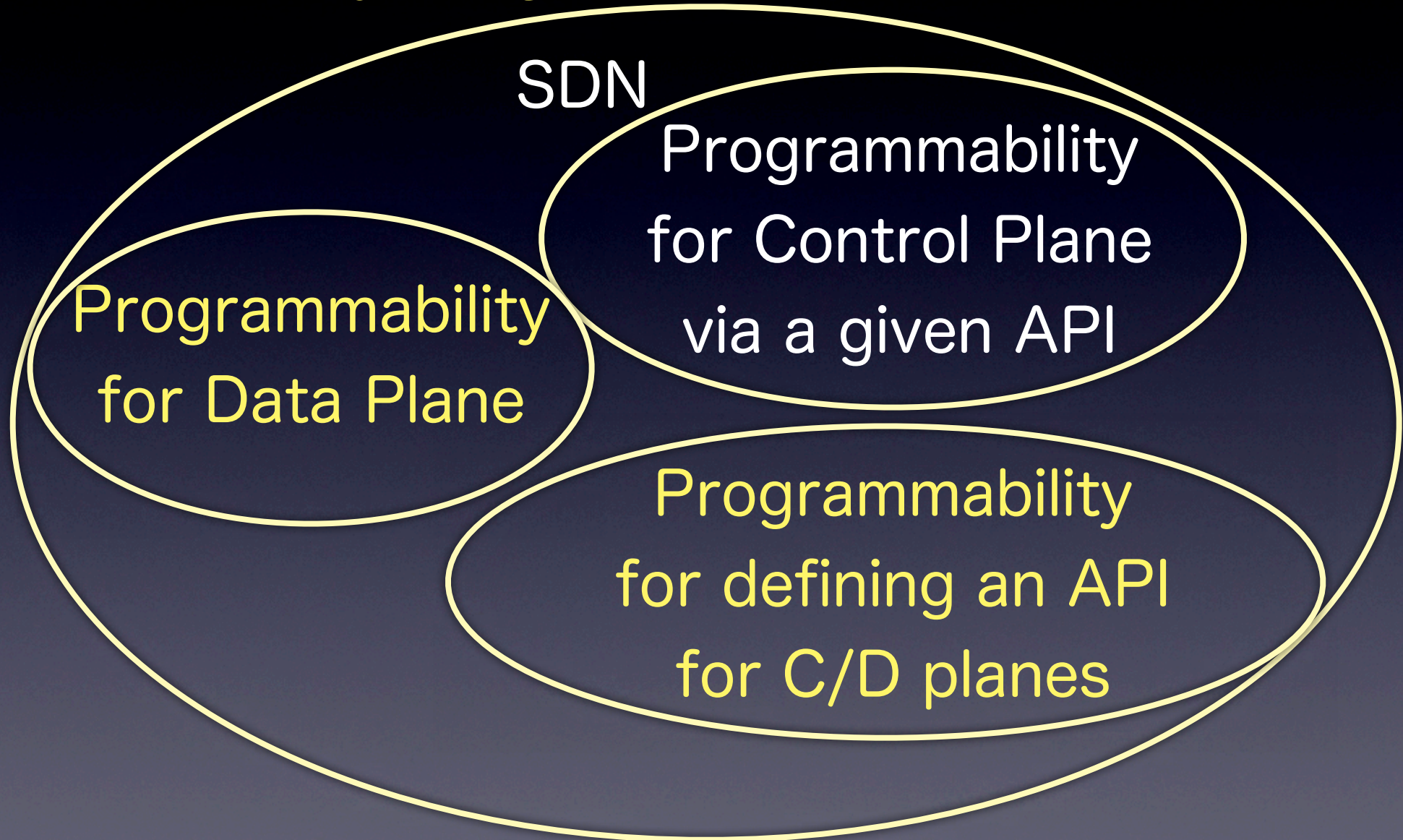- Fixed-mobile converged slicing
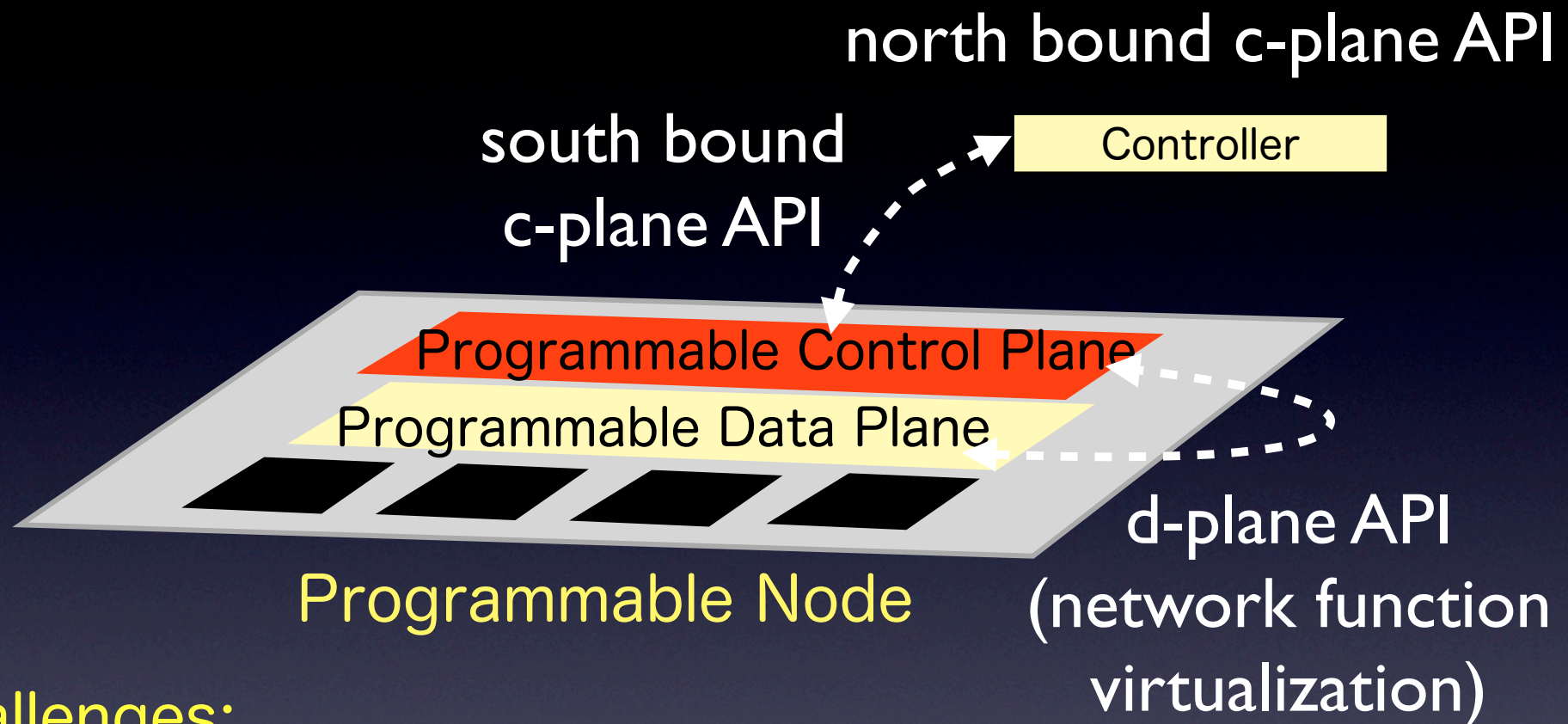

Network Edge Slicing

VNode Infrastructure

Access Point

Lightweight Slicing

Small-Factor VNode

GRE Tunnel

GRE Tunnel

GRE Tunnel

DPN as a super set of SDN

DPN (Deeply Programmable Network)

SDN

Programmability for Control Plane via a given API

Programmability for Data Plane

Programmability for defining an API for C/D planes

Making fully programmable network nodes?

north bound c-plane API

south bound c-plane API

Controller

Programmable Control Plane

Programmable Data Plane

d-plane API (network function virtualization)

Programmable Node

Challenges:

- Achieve both programmability and performance at the same time
- Instantly upgrade/downgrade switching logics
- Enable network virtualization (multiple logical slices)
- Make a slice fully programmable (data-plane, control-plane)

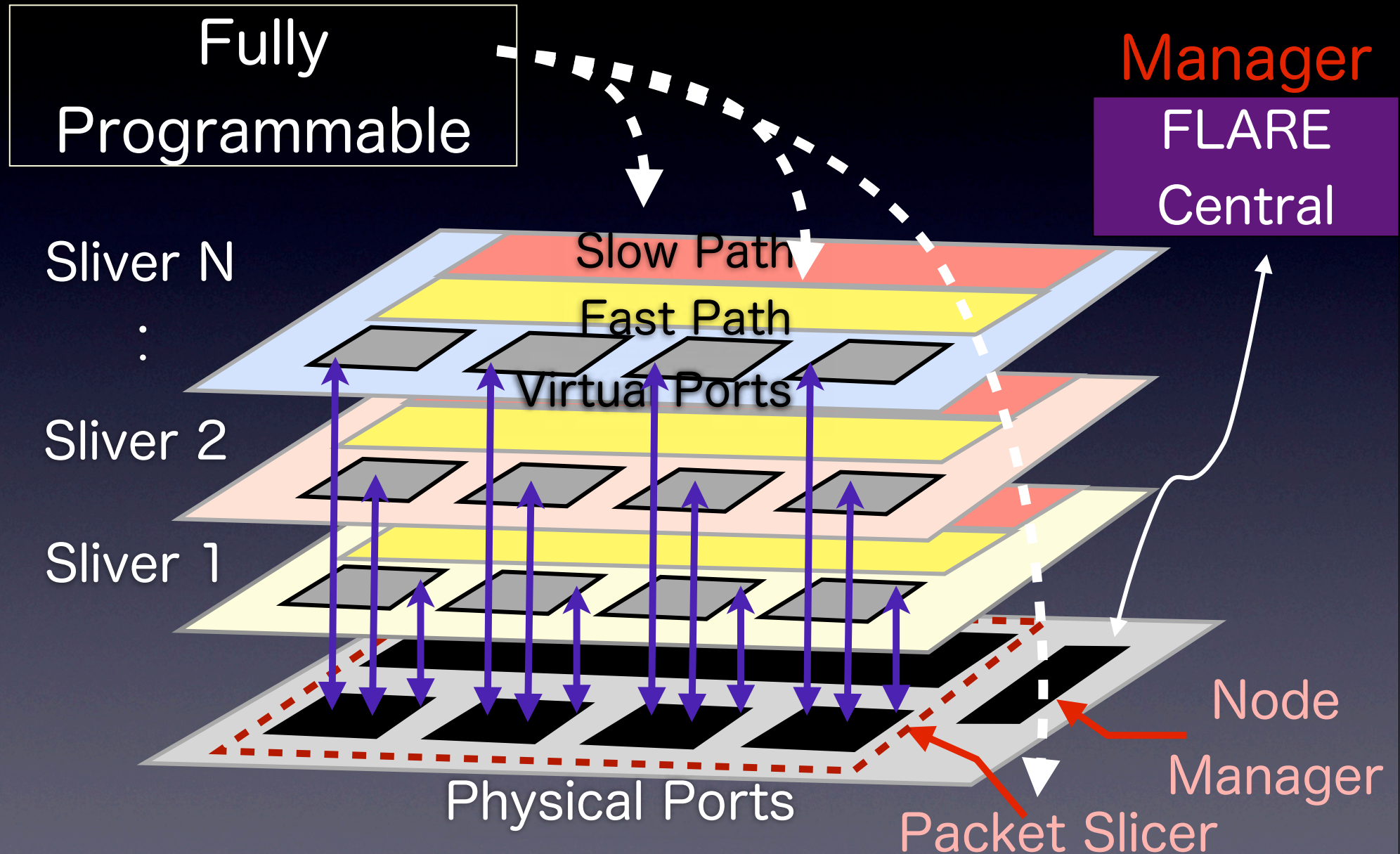# FLARE

SDN
Control Plane Programmability

DPN
SDN + Data Plane Programmability

Deeply Programmable Node

Network Virtualization
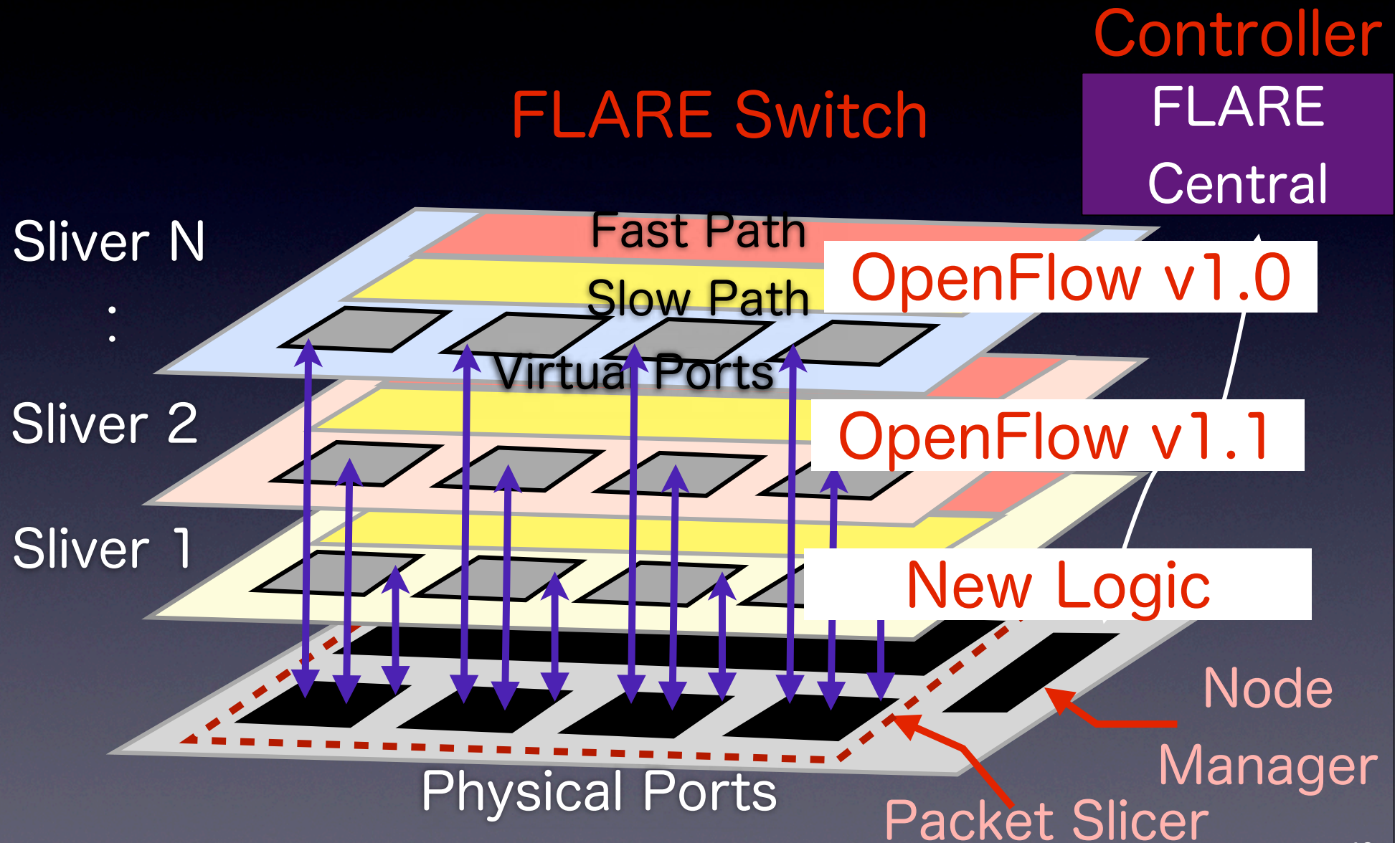
New Protocol Capability

FLARE Architecture

Fully Programmable

Manager

FLARE Central

Sliver N

Slow Path

Fast Path

Virtual Ports

Sliver 2

Sliver 1

Node Manager

Physical Ports

Packet Slicer

FLARE Switch Implementation

- Mini 1U / 1U / 2U Form Factor (only 200W)

- A combination of resource containers on many-core processor (fast path)＋x86 processor (slow path)

- 4x10Gbps (20Gbps Non-blocking）, 2x10G+8x1G Planned

- Up to 15 slow-path slivers can be instantiated

- Linux programmability at slow/fast-path slivers and packet slicer

- Parallel programming for high performance at fast-path

- OpenFlow switch logic and API can be programmed

Control Plane Versioning
Change according to flows, time, etc

Controller

FLARE Switch

FLARE Central

Sliver N

Fast Path

OpenFlow v1.0

Slow Path

Virtual Ports

Sliver 2

OpenFlow v1.1

Sliver 1

New Logic

Node Manager

Physical Ports
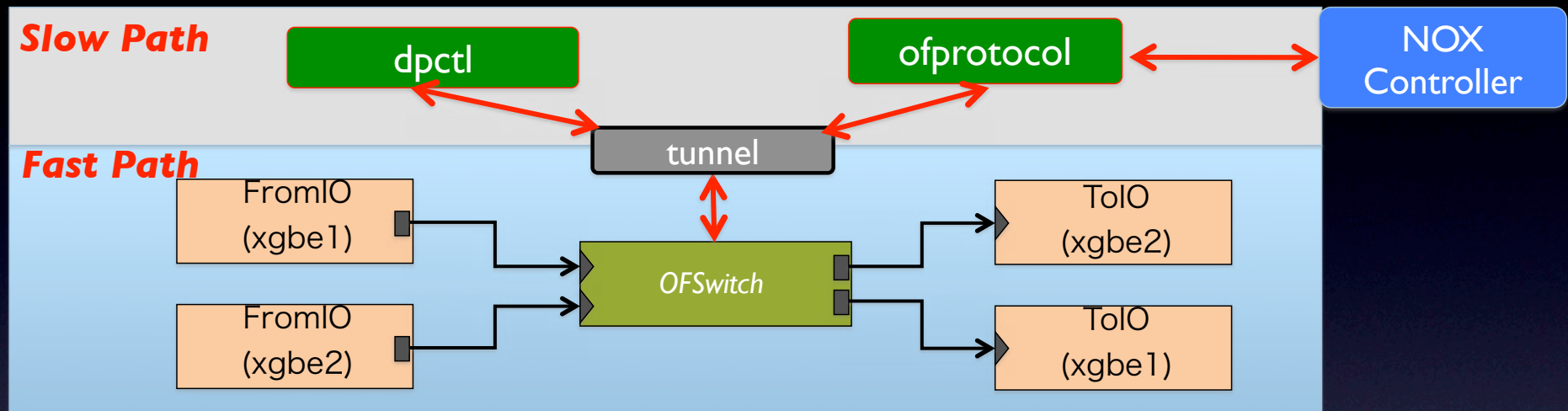
Packet Slicer

Network Virtualization

FLARE AP   FLARE Switch

FLARE Manager
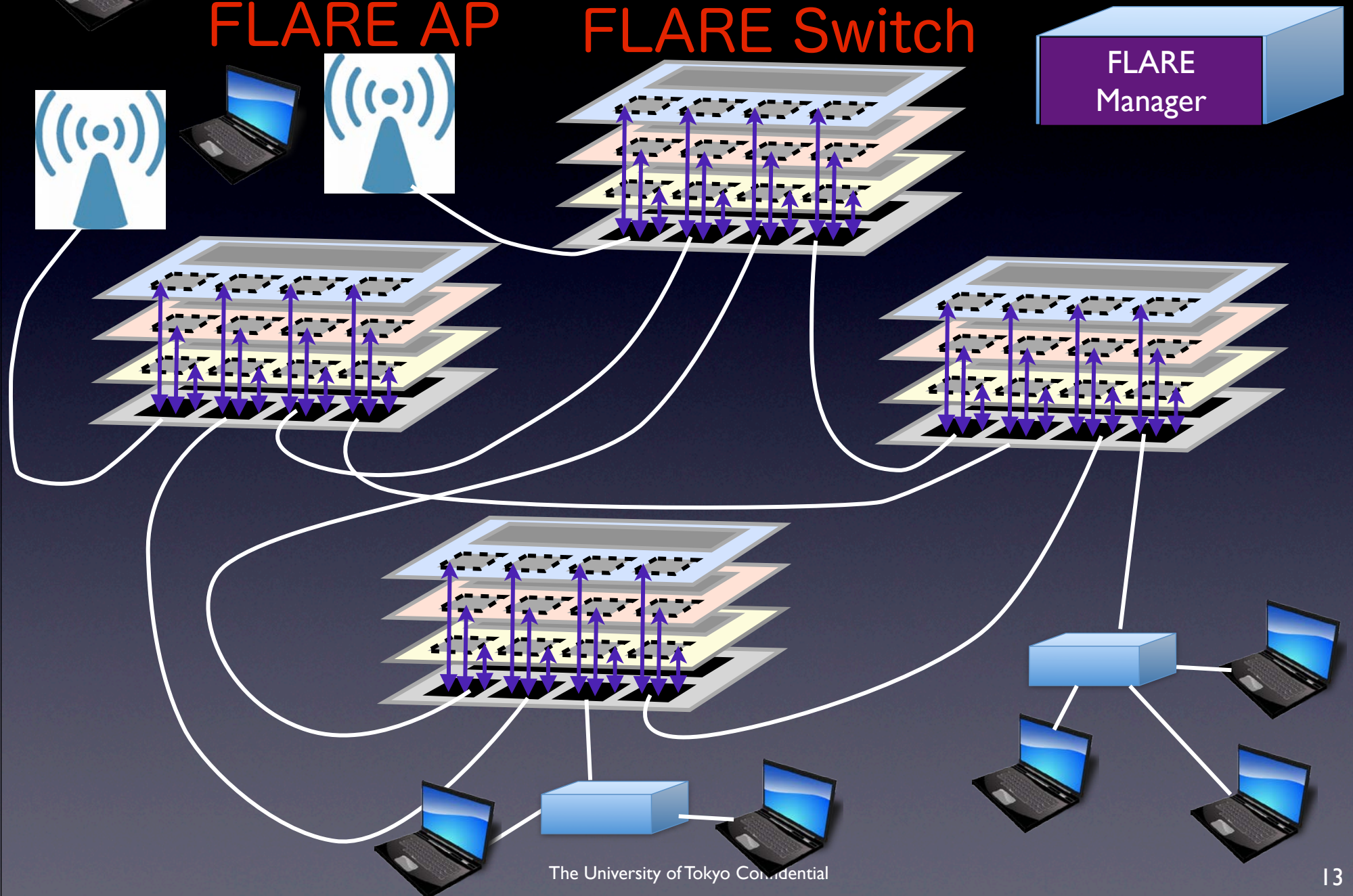
Programming Model
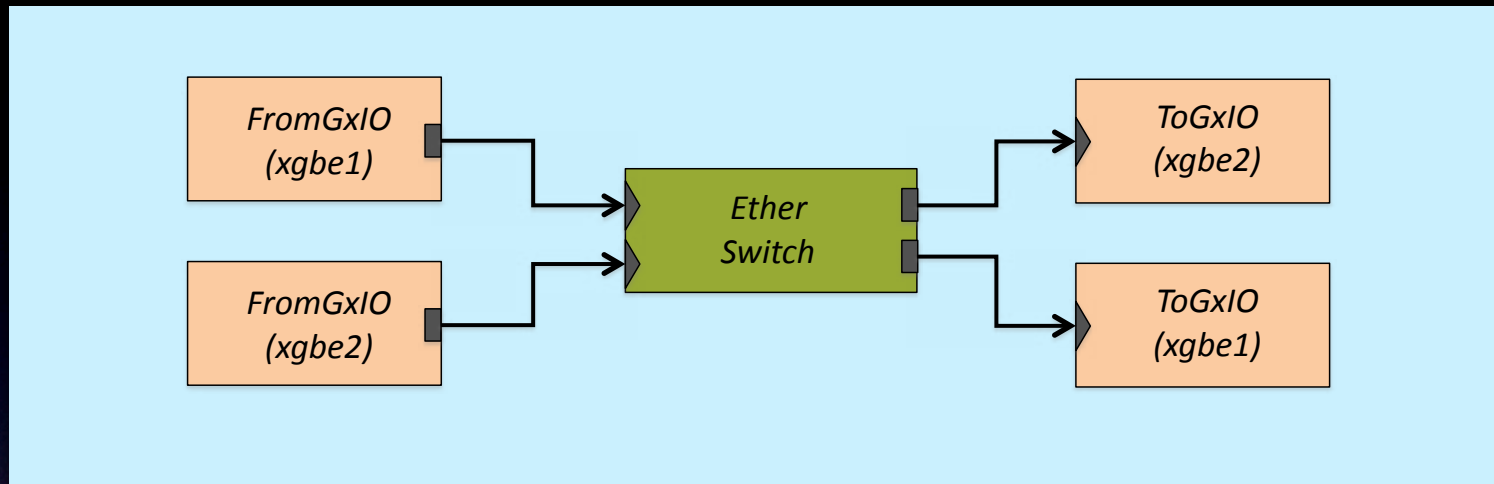

Multi-Threaded Modular Programming

e.g., Click Software Modular Router

- Arbitrary switch logic(s) can be implemented in fast-path, slow-path and slicer sliver
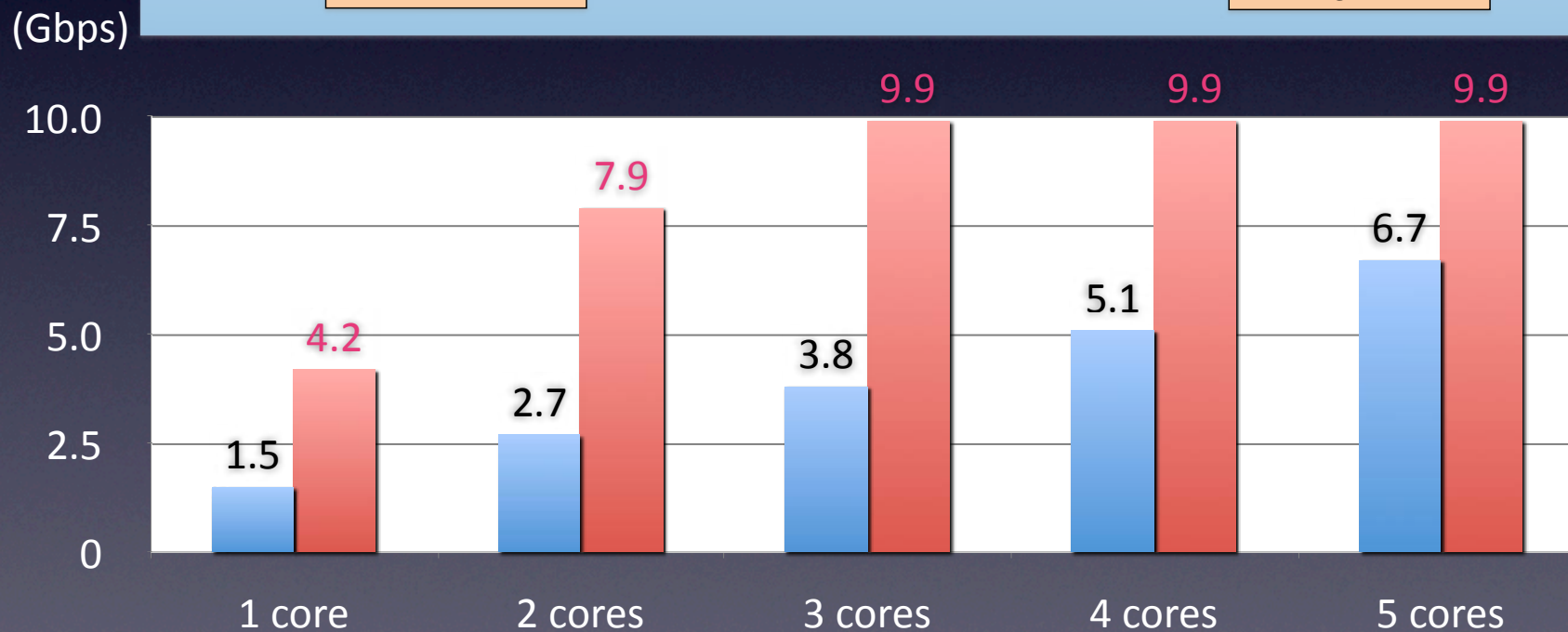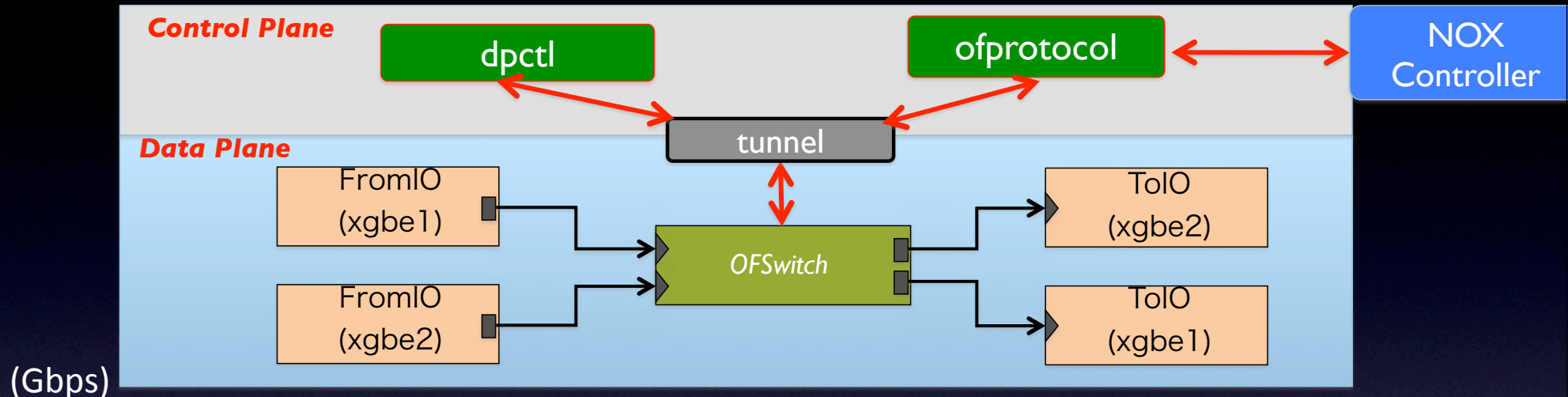- Ready-made software modules (Ethernet, CRC, ARP, IPv4, IPv6, IPSec, GRE, NAT, many more...)

Ethernet Switch
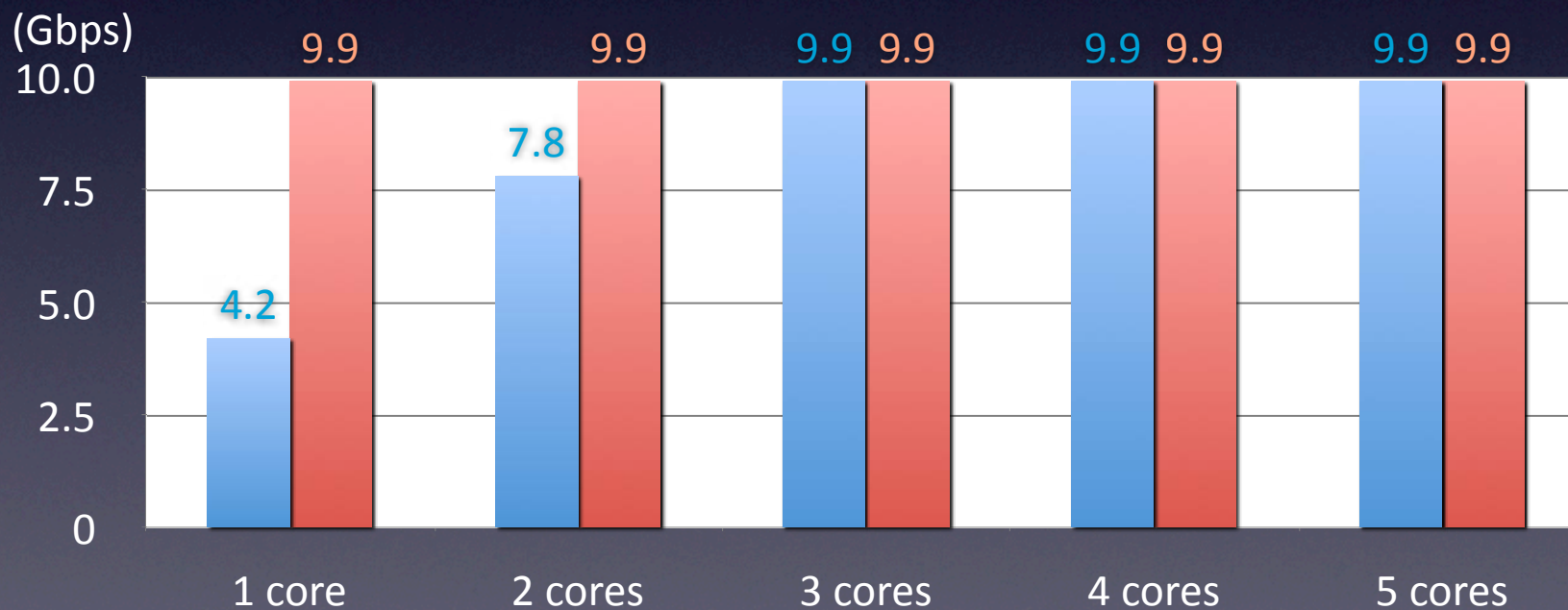
Switching Performance

Legend:
- pkt_size=512B (blue)
- pkt_size=1514B (red)

OpenFlow Switch

Switching Performance

How deep programmability do we want?

Several questions to ask:

- Control plane programmability only?

- Data plane too (cache, transcode, DPI)?
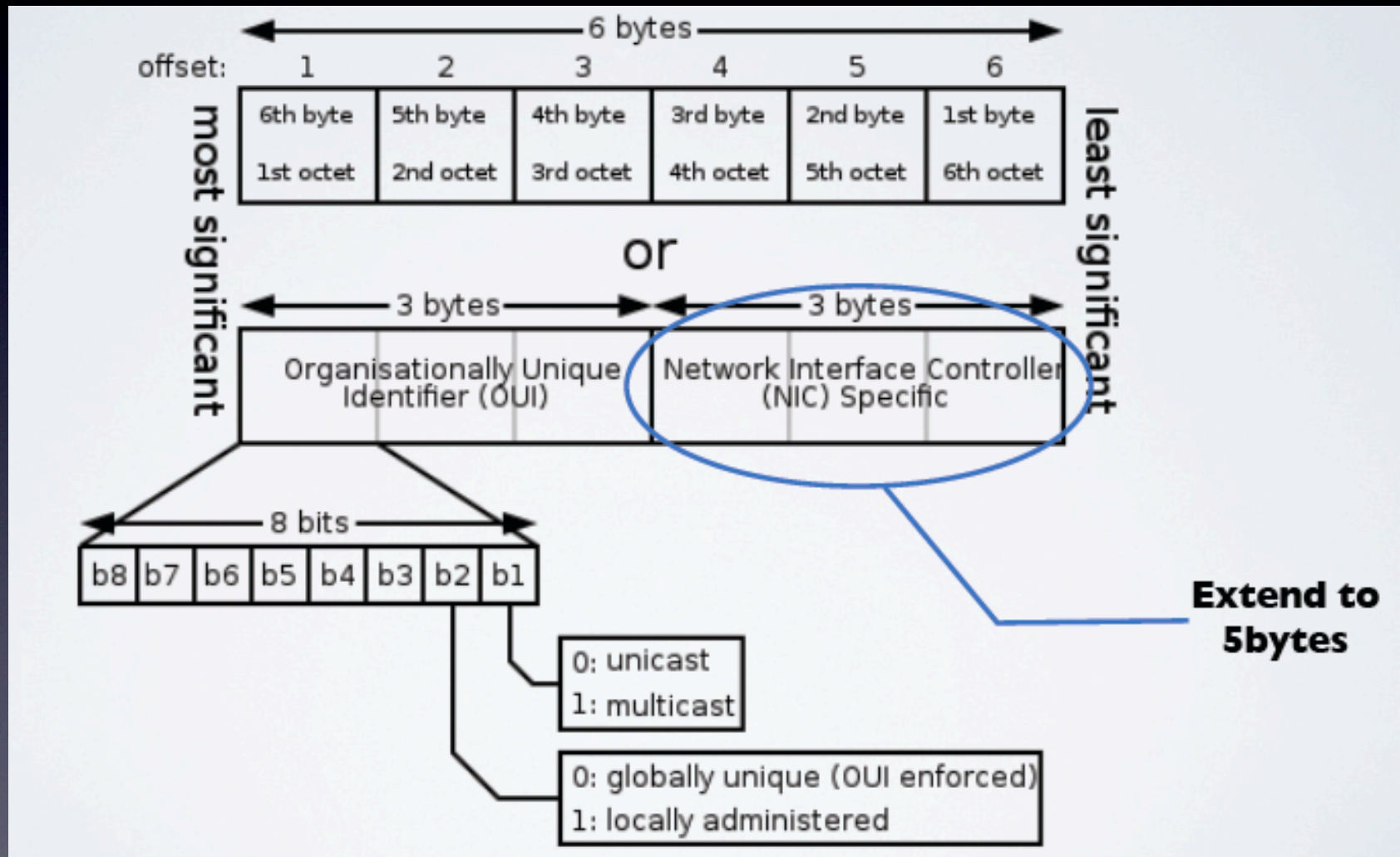
- Can we define a new L2 protocol?

A Case in Data Center Network

- Limitation in MAC address space
  - Conflict of MAC addresses in VM migration
- Limitation in VID (802.1Q) space
  - The number of tenants increases in IaaS

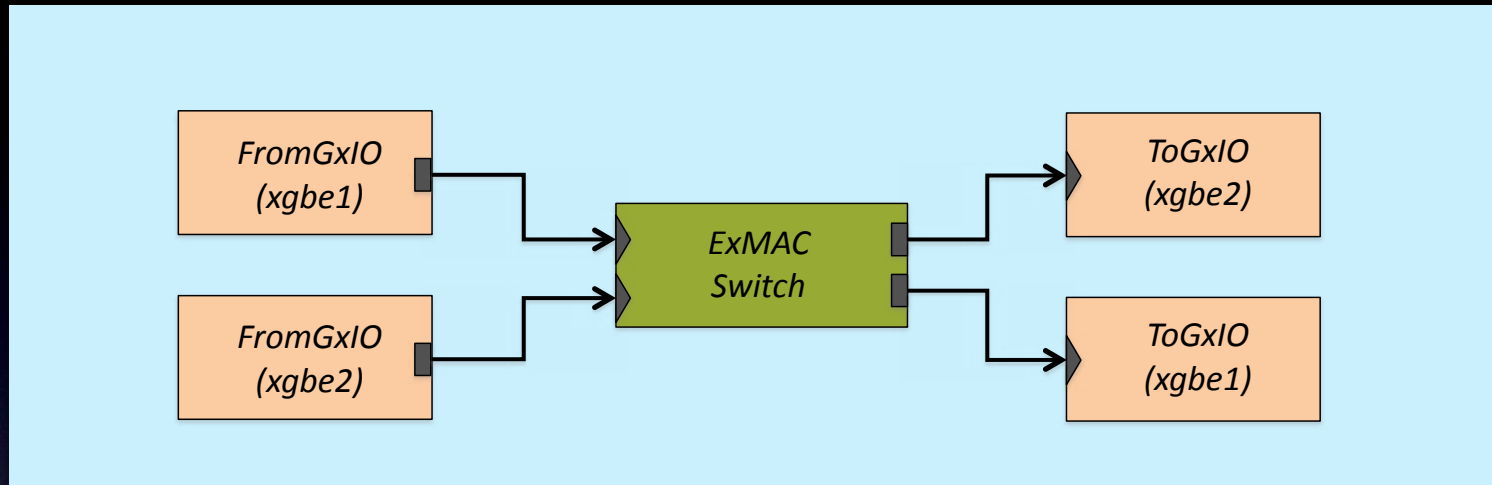Data Center Network depends heavily on L2 leading to solutions such as EUI-64 and VXLAN

Mac Address Extension
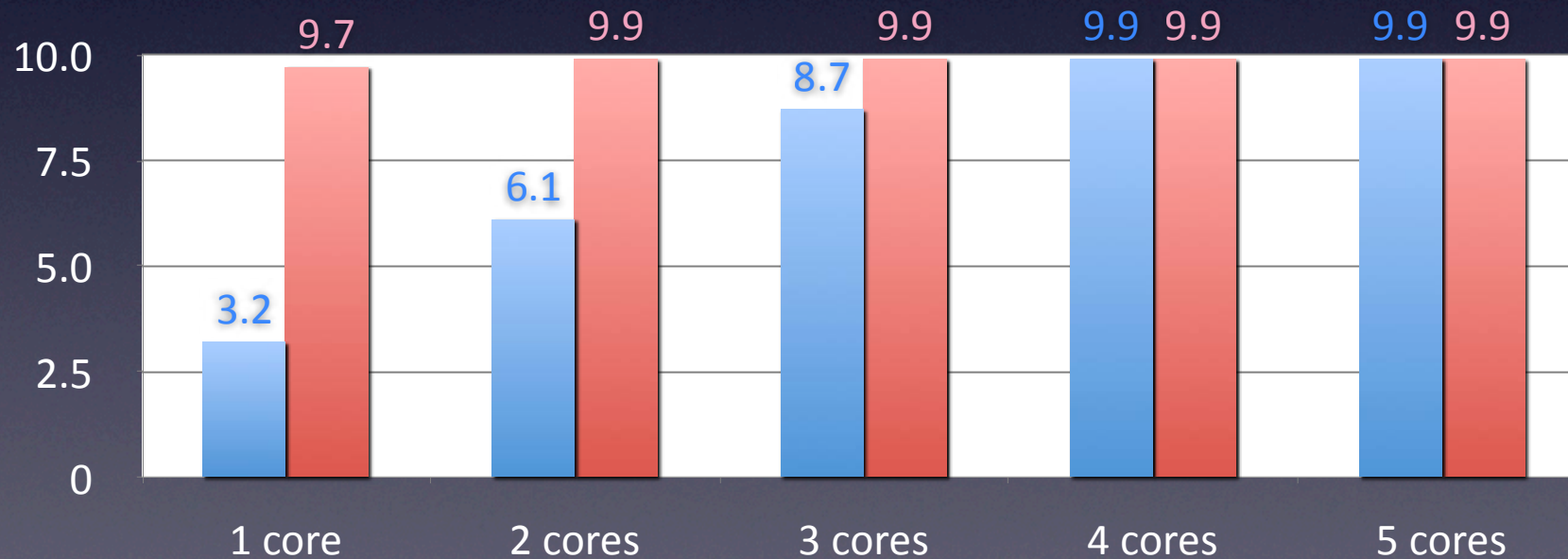
EUI-64 (64bit Extended Unique Identifier)

Extended MAC Switching

Switching Performance

Inter-Cloud VM Migration With Extended MAC

UTokyo(Tokyo)

Guest VM	Guest VM
Extended MAC1	Extended MAC2

Starbed(Hokuriku)

Guest VM	Guest VM
Extended MAC1	Extended MAC2

WAN Migration

FLARE ———————————— FLARE

Extended DMAC	Extended SMAC	Type	IP Datagram
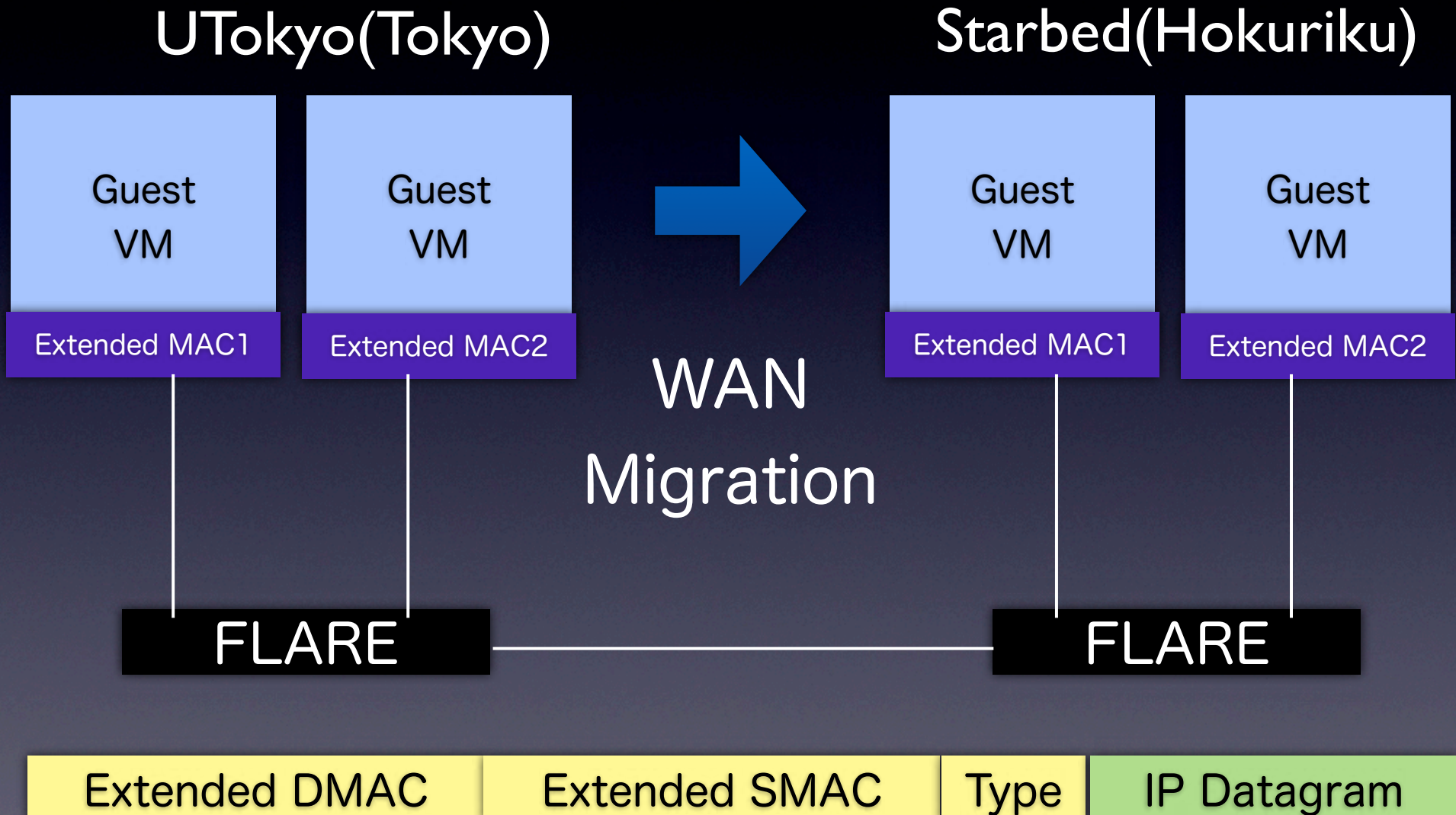
FLARE at ITPro EXPO 2012
Beyond OpenFlow/SDN

10Gb/s SFP

MPLS 2012
(with Cisco & Juniper)

Conclusion

- Programmability for Data-plane and (re)defining APIs for C/D planes is considered extension to Software Defined Network (SDN) and an important topic to explore

- Inter-cloud network may benefit from deep programmability for enabling in-network services and defining new protocols.