Deeply Programmable Network (DPN) through Advanced Network Virtualization

Aki Nakao
University of Tokyo
nakao@iii.u-tokyo.ac.jp

2012/8/24

Network Vitualization Symposium 2012

Deeply Programmable Network

- Network that flexibly adapts to demands by means of not only programmability in control plane for network operations and management, but also deeper programmability for data-plane for processing data and dealing with new protocols (non-IP), etc.
- Extension to SDN (Software Defined Network)

Deep Programmability

- Control-Plane Programmability
  - Route Control
  - Access Control
  - Netowrk Management
- Data-Plane Programmability
  - Packet Data Processing
    - Cache
    - Transcode
    - DPI
  - Handling New Protocols
    - IPvN (N>6)
    - New Layer2
    - Named Data Network (NDN)
    - Content Centric Network (CCN)

Possible with OpenFlow

Hard with OpenFlow

Possible with OpenFlow +Processors

Hard with OpenFlow Even with +Processors

Deeply Programmable Network

Deeper Programmability

How deep programmability do we want?

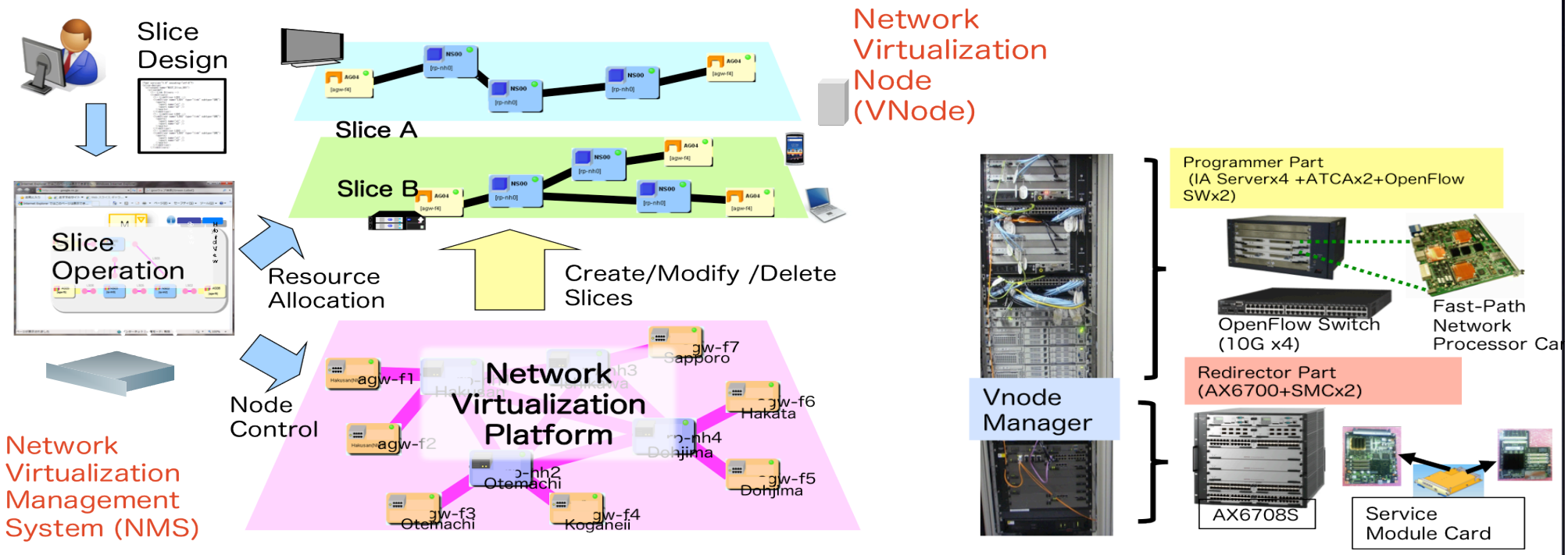
Several questions to ask:

- Control plane programmability only?
- Data plane too (cache, transcode, DPI)?
- OpenFlow in A Slice
- Can we program “OpenFlow”, “OpenFlow++”, or “OpenXXX” data-plane APIs?
- Can we change L2 protocol?

Yes! we can do all!

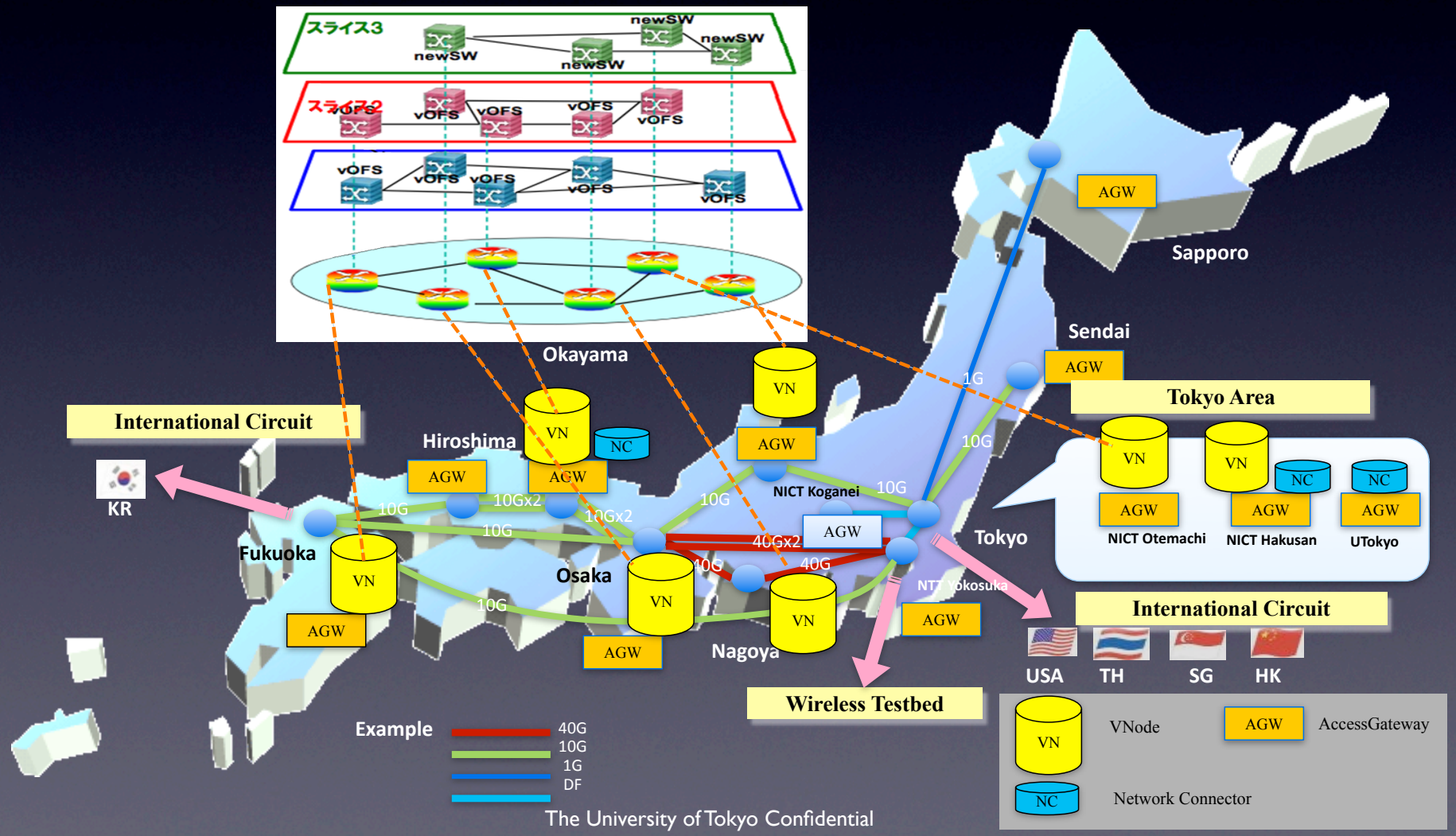
VNode Project

- VNode Infrastructure Enables Deeply Programmable Network
  (Project Leader: Aki Nakao)
- 2008-2010 Collaborative Research (NICT/Utokyo/NTT/NEC/Hitachi/Fujitsu)
- 2011-2014 Collaborative Research (Utokyo/NTT/NEC/Hitachi/Fujitsu/KDDI)



VNode Infrastructure (extended to US!)

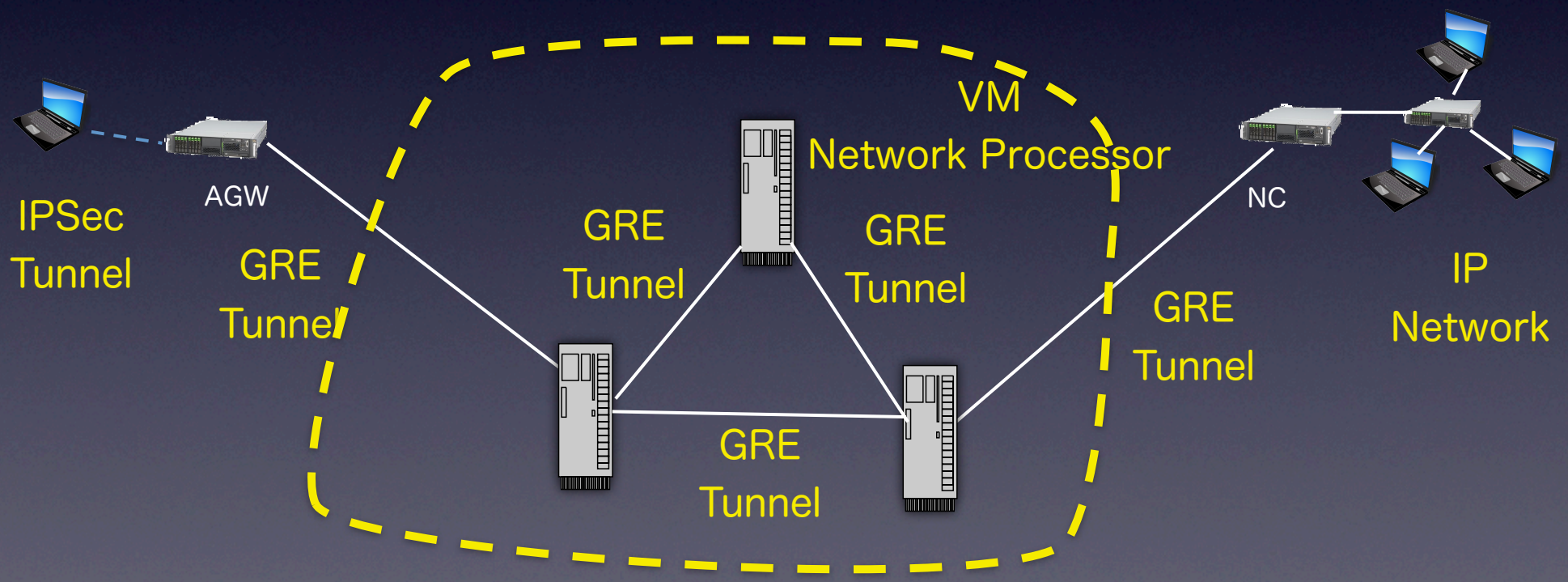
- 7 VNodes, 2 Network Connectors, 11 Access Gateways
- Deep Programmability for Experimenting with Arbitrary Protocols (Non-IP)
- Slice-Around-The-World Project (A VNode in U of Utah on ION/StarLight!)



Deep Programmability for Backbone Network

- Create slices for running arbitrary protocols
- Programmability for both control plane and data plane
- Mainly focused on wired / fixed network virtualization

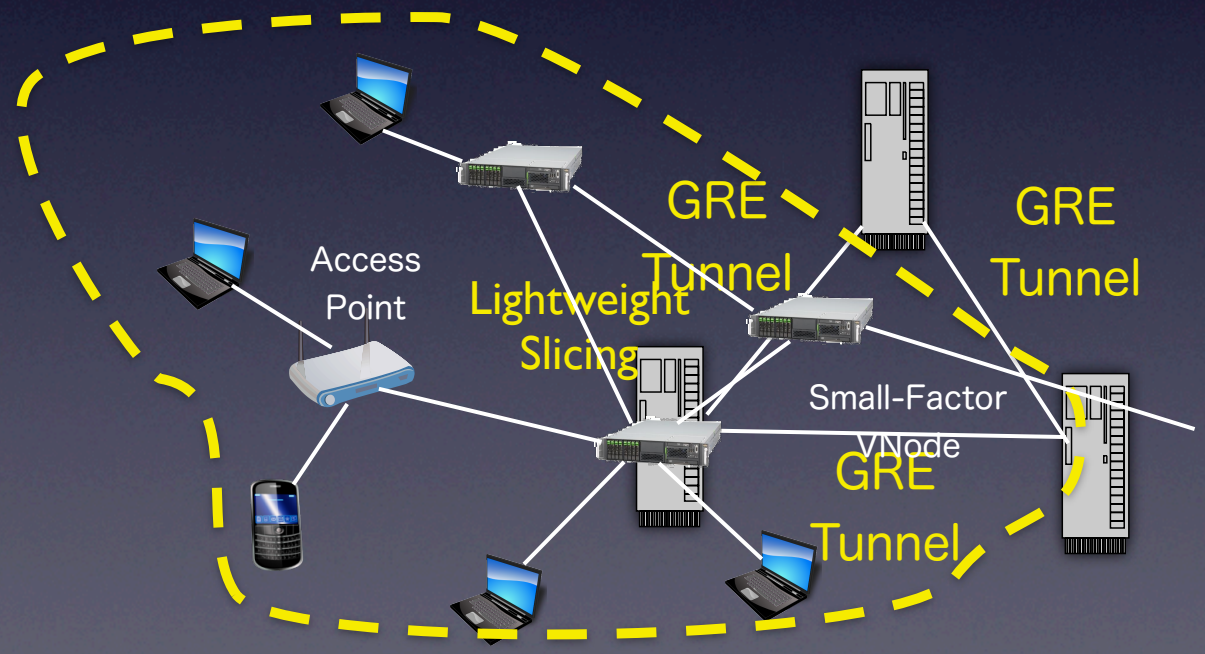
VNode Infrastructure



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

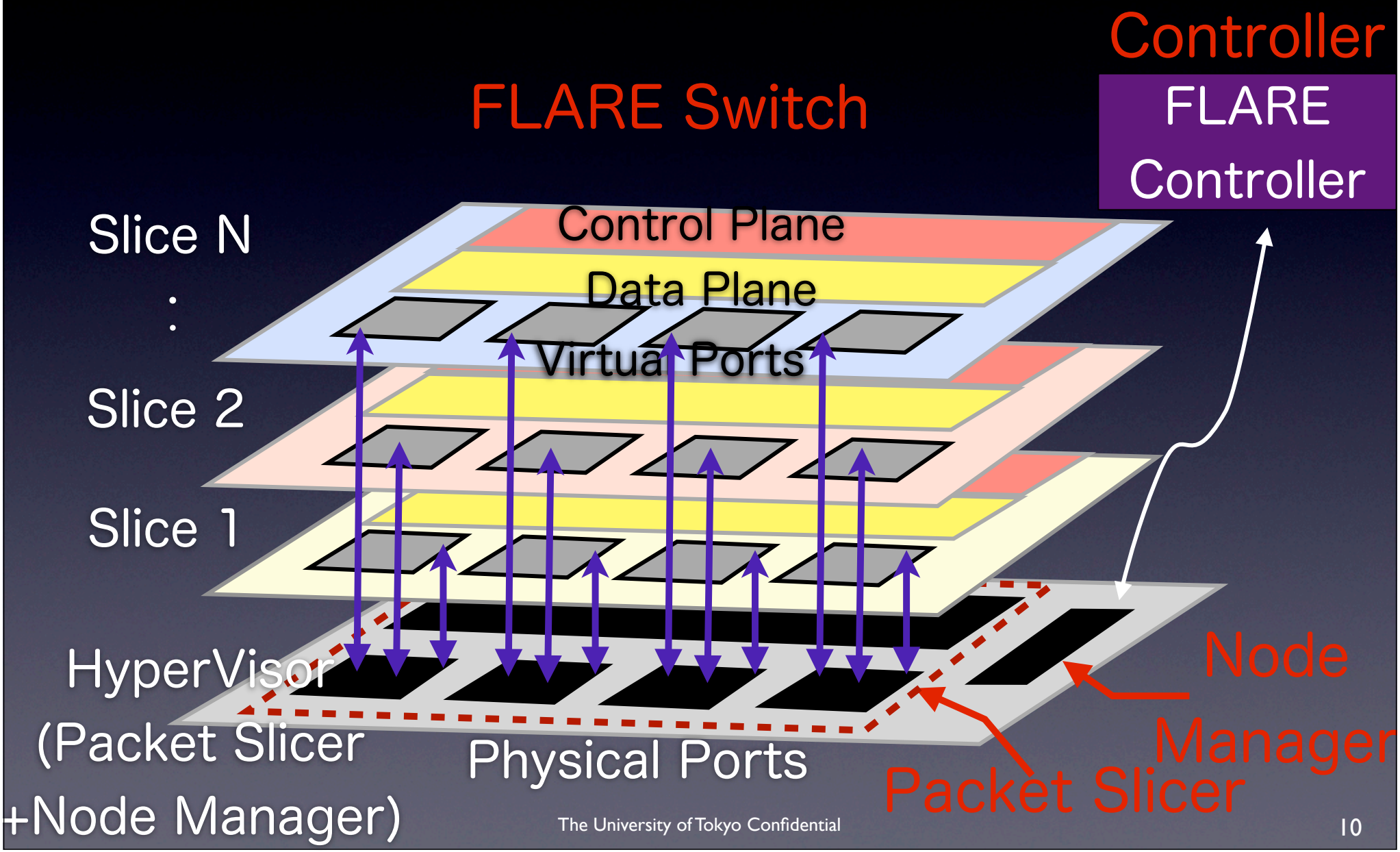


FLARE

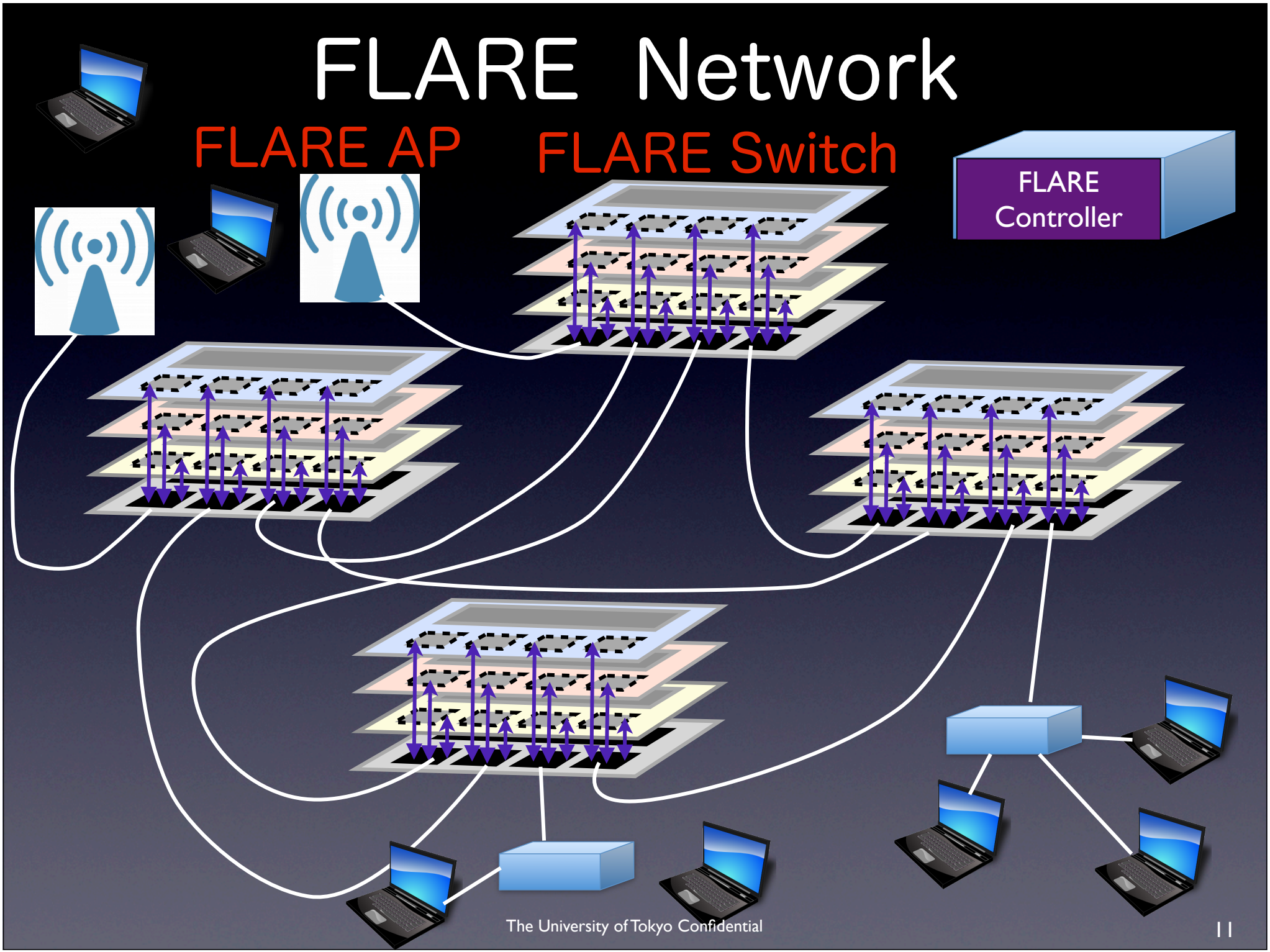
Open Deeply Programmable Switch

Challenges:

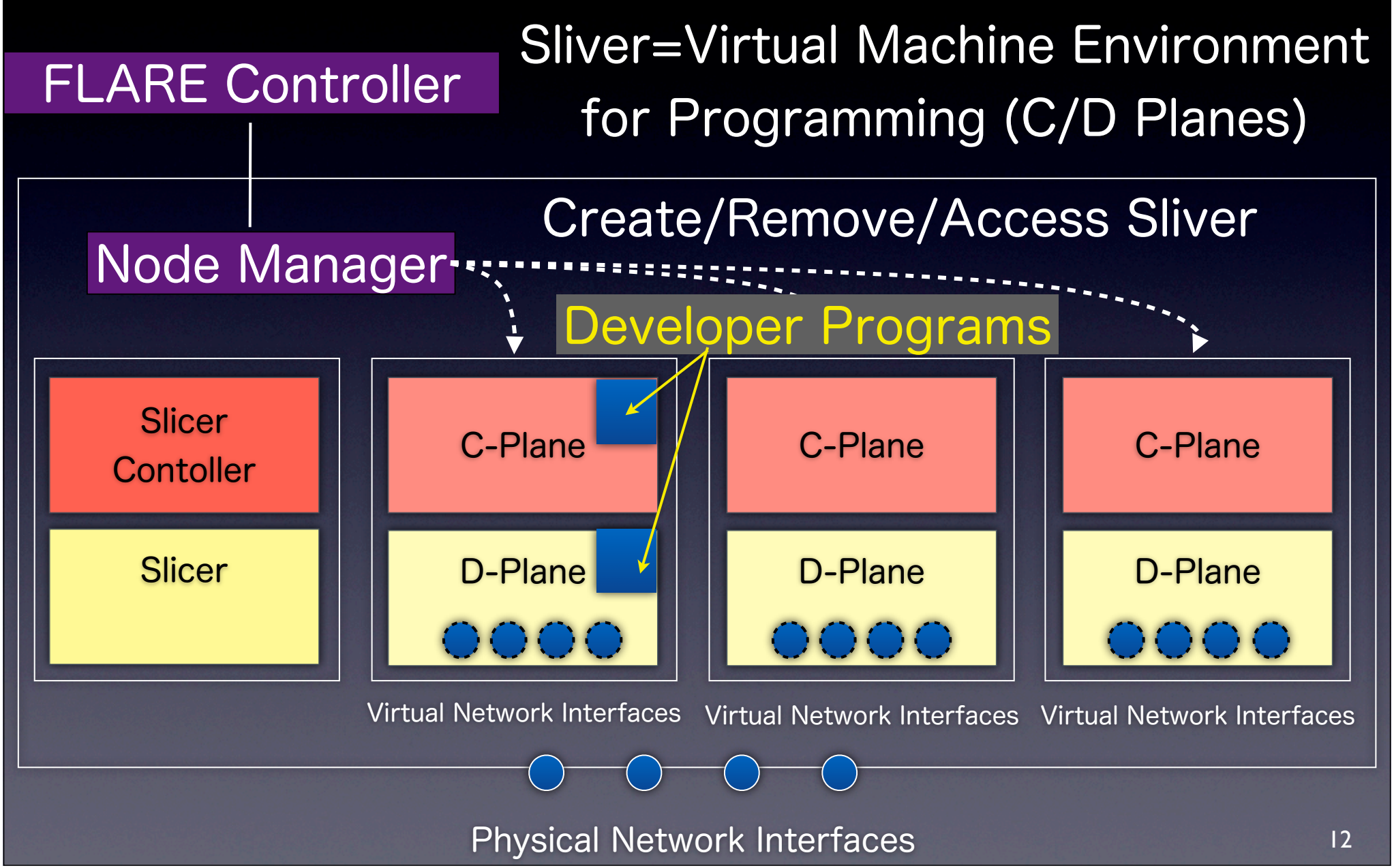
FLARE Architecture



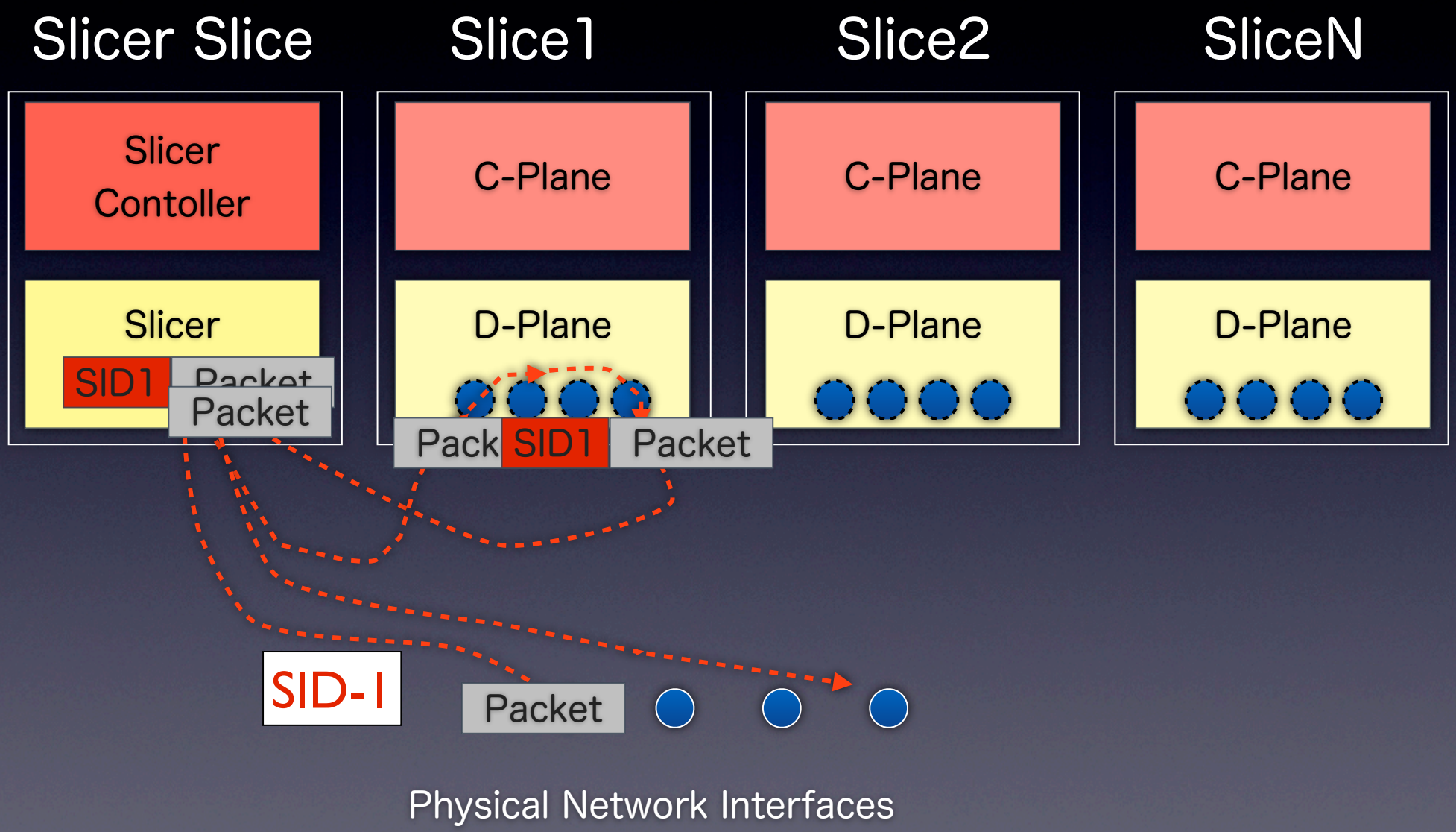
FLARE Network



Solar Wind Version



Per-Slice Switching



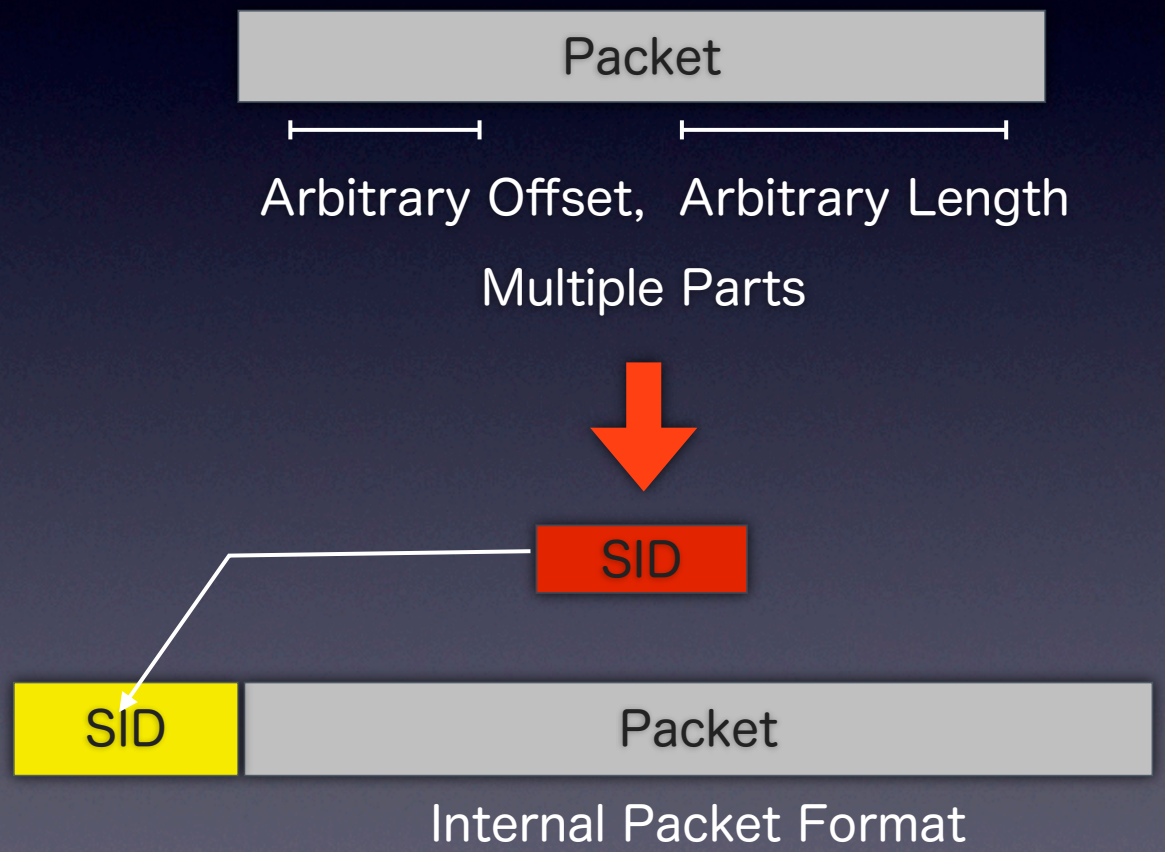
Three Primitives in FLARE

- Extract SliceID (Slicer)
- Demux/Mux to/from Slices (Redirector)
- Execute Programs on Slices (Programmer)

Extracting Slice ID

General Slicing

(Supported in DarkFlow Version)

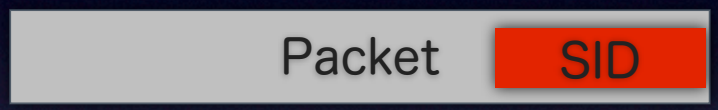


Light Weight Network Virtualization
(Application of FLARE to NV)

- Tunneling (Good for Fixed Networks)
- VLAN (Limited to Ethernet/ 12bits)
- MPLS (Originally Traffic Engineering)

Essentially just need globally unique Slice ID

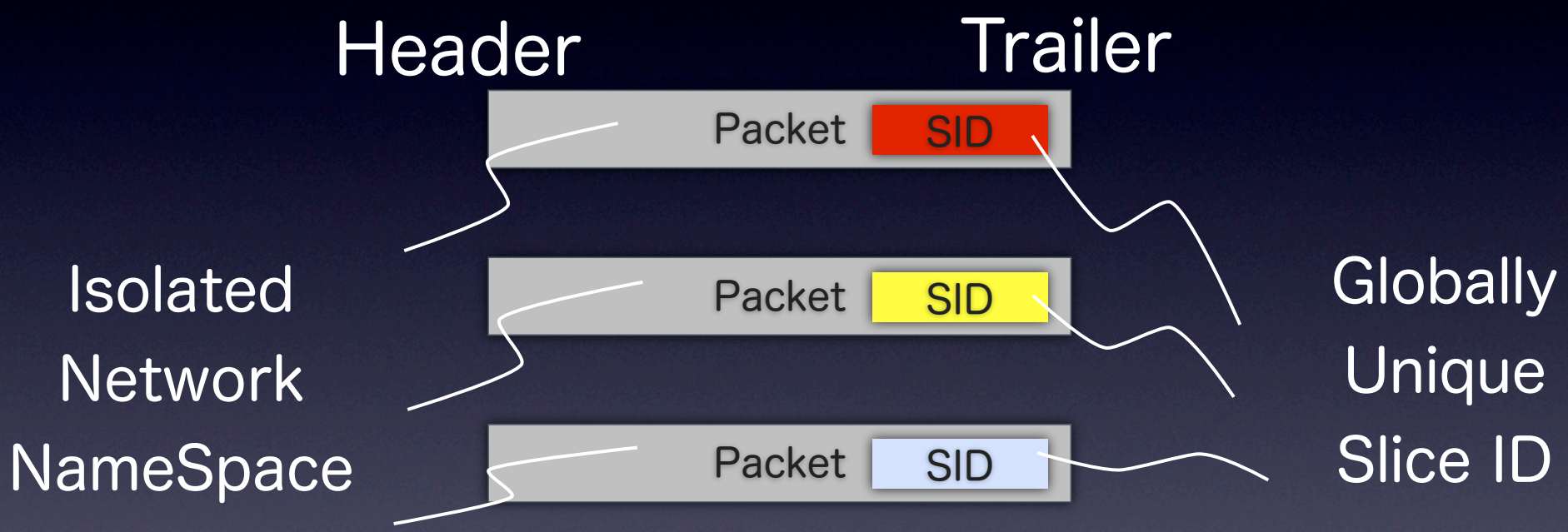
Trailer Slicing



Globally Unique SID Embedded in Trailer

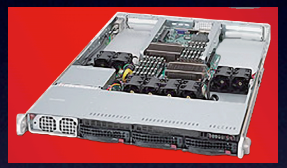
Slicer-slice identifies Slice ID(SID)

Network Virtualization
Through Slice ID Embedded in Packet Trailers

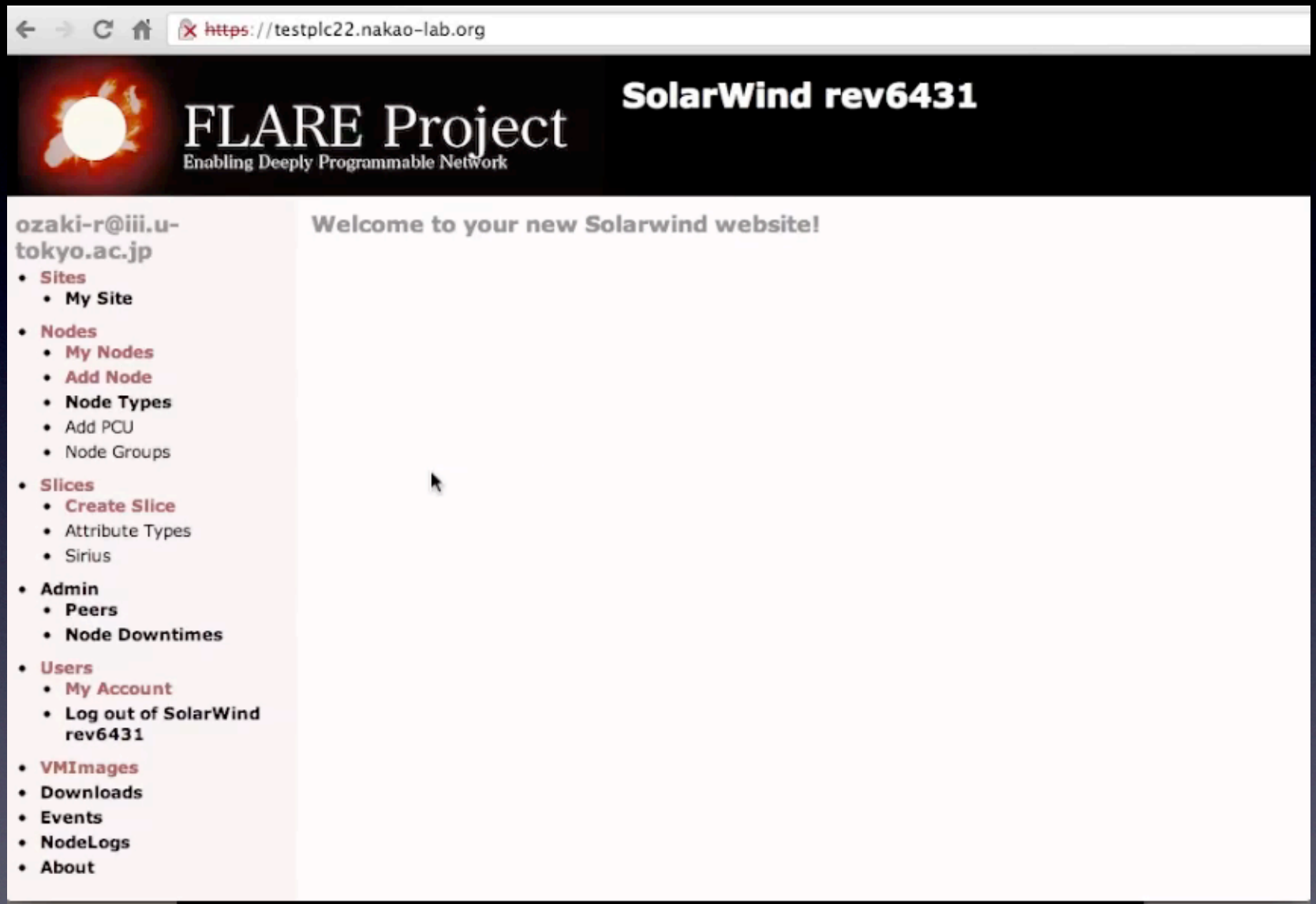


- SID visible at all the layers
  (if layered architecture is adopted at all)
- VLAN Stitching may not be necessary

FLARE (SolarWind Version) Implementation

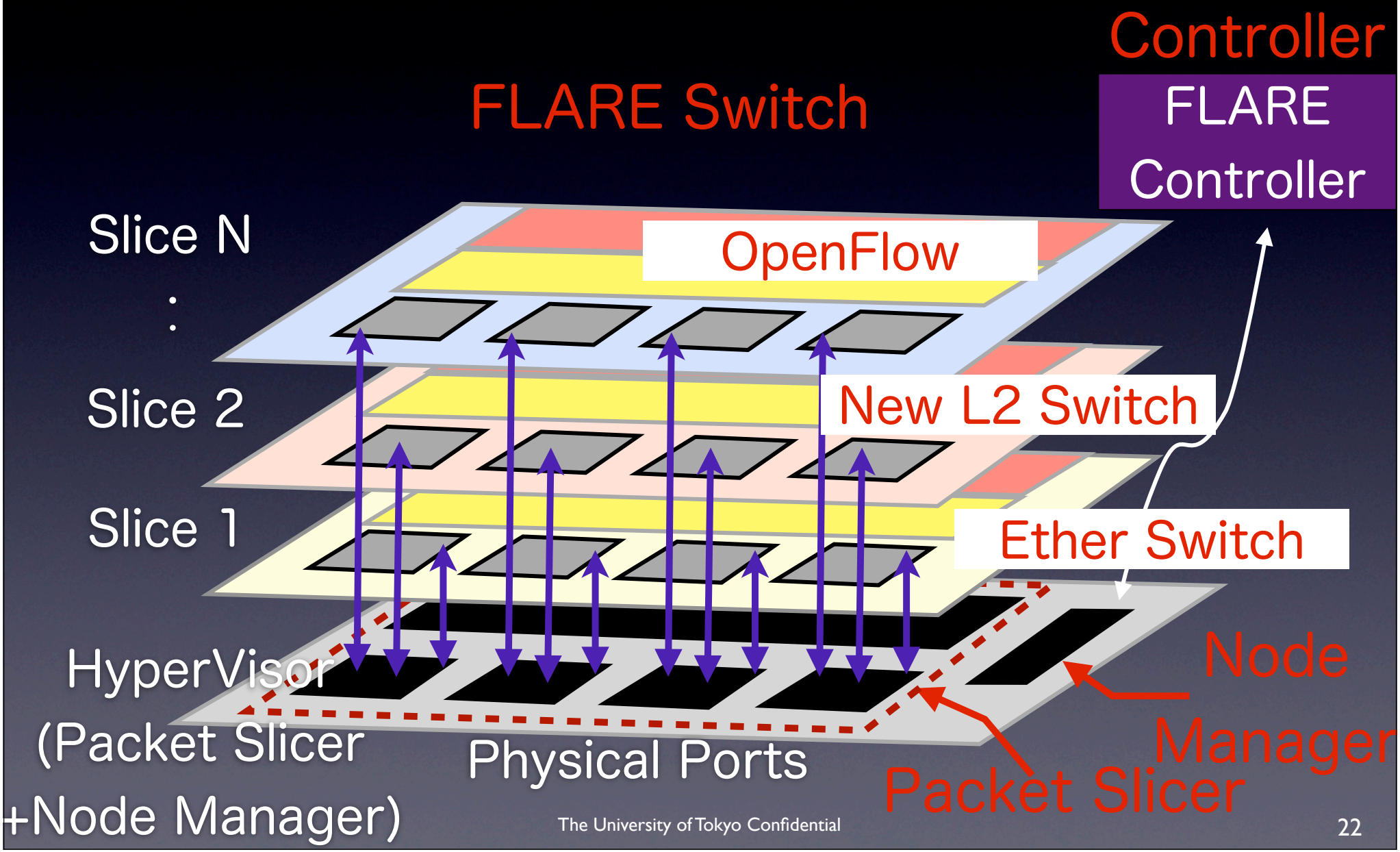
- Hybrid of many-core processors + x86 processors
- LXC Resource Container Running on Data/Control Planes
- 1U Form Factor 
- Trailer Slicing (Embedding Slice ID in Packet Trailers)
- 4 x 10Gbps (20 Gbps Non-Blocking Switching)
- Up to 16 Data Plane Slices
- Control Plane & Data Plane Linux Programmability (Multi-Threaded Click)
- OpenFlow In A Slice (Implemented)

FLARE Controller



Flare Controller

Multi-Protocol/Control Coexistence



OpenFlow vs. FLARE Differences

	OpenFlow	FLARE
Arbitrary Frame	X (Internet Protocol)	O (Non-Internet Protocol)
Control Plane Programmability	O	Multiple APIs OpenFlow++ Multiple OpenFlow SWs
Data Plane Programmability (Packet Process)	X	O
Data Plane Programmability (New Protocol)	X	O

OpenFlow vs. FLARE Differences

	OpenFlow	FLARE
Programming Model	Flow Pattern Match Action	Linux Container (LXC) Multi-Threaded Click
Processing Environment	Not Integrated	Integrated
Slicing Programmability	X (Fixed to Flow Matching)	O (fully supported in DarkFlow version)

How deep programmability do we want?

Several questions to ask:

- Control plane programmability only?
- Data plane too (cache, transcode, DPI)?
- Can we program “OpenFlow”, “OpenFlow++”, or “OpenXXX” dataplane APIs?
- Can we change L2 protocol?

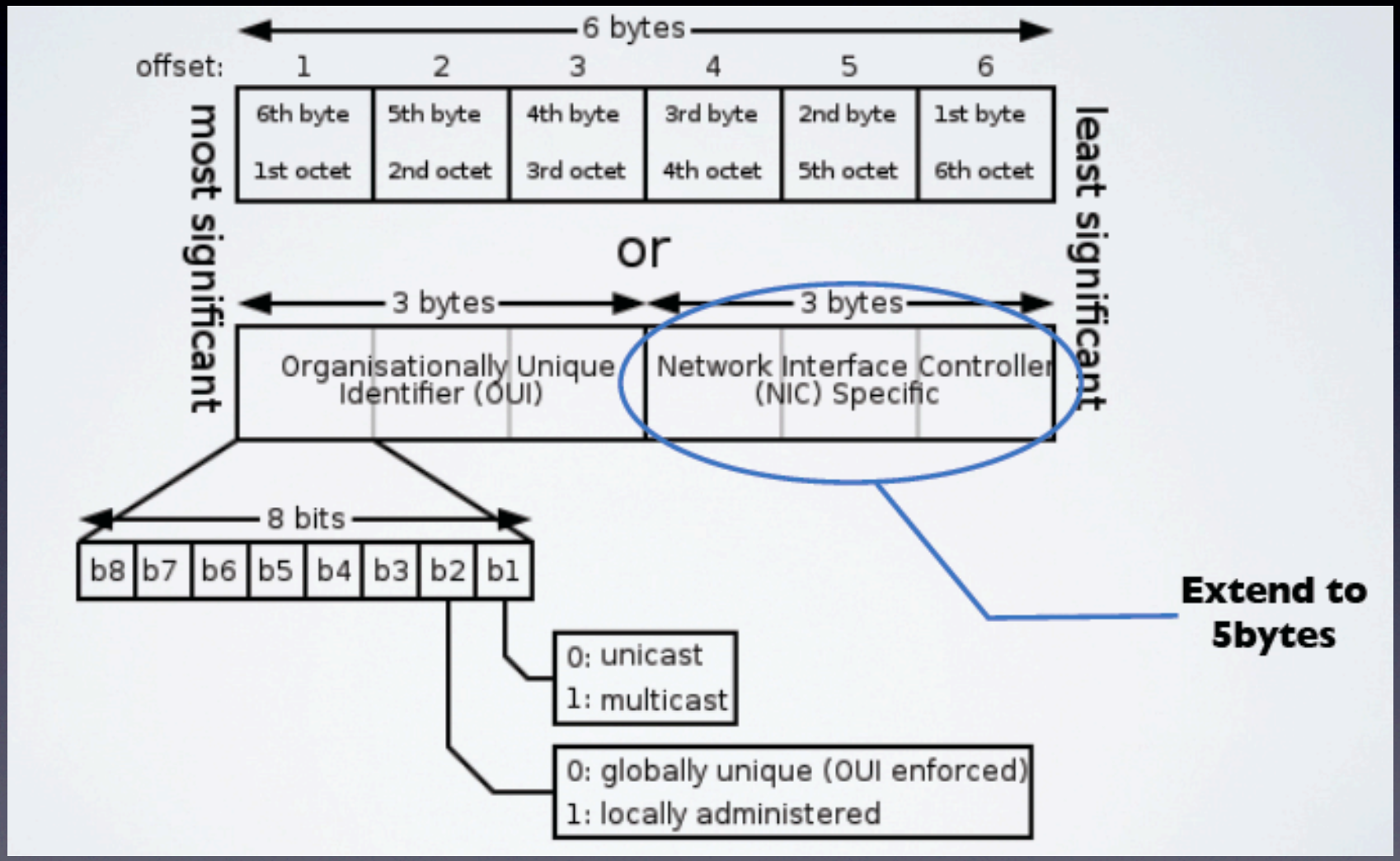
A Case in Data Center Network

- 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

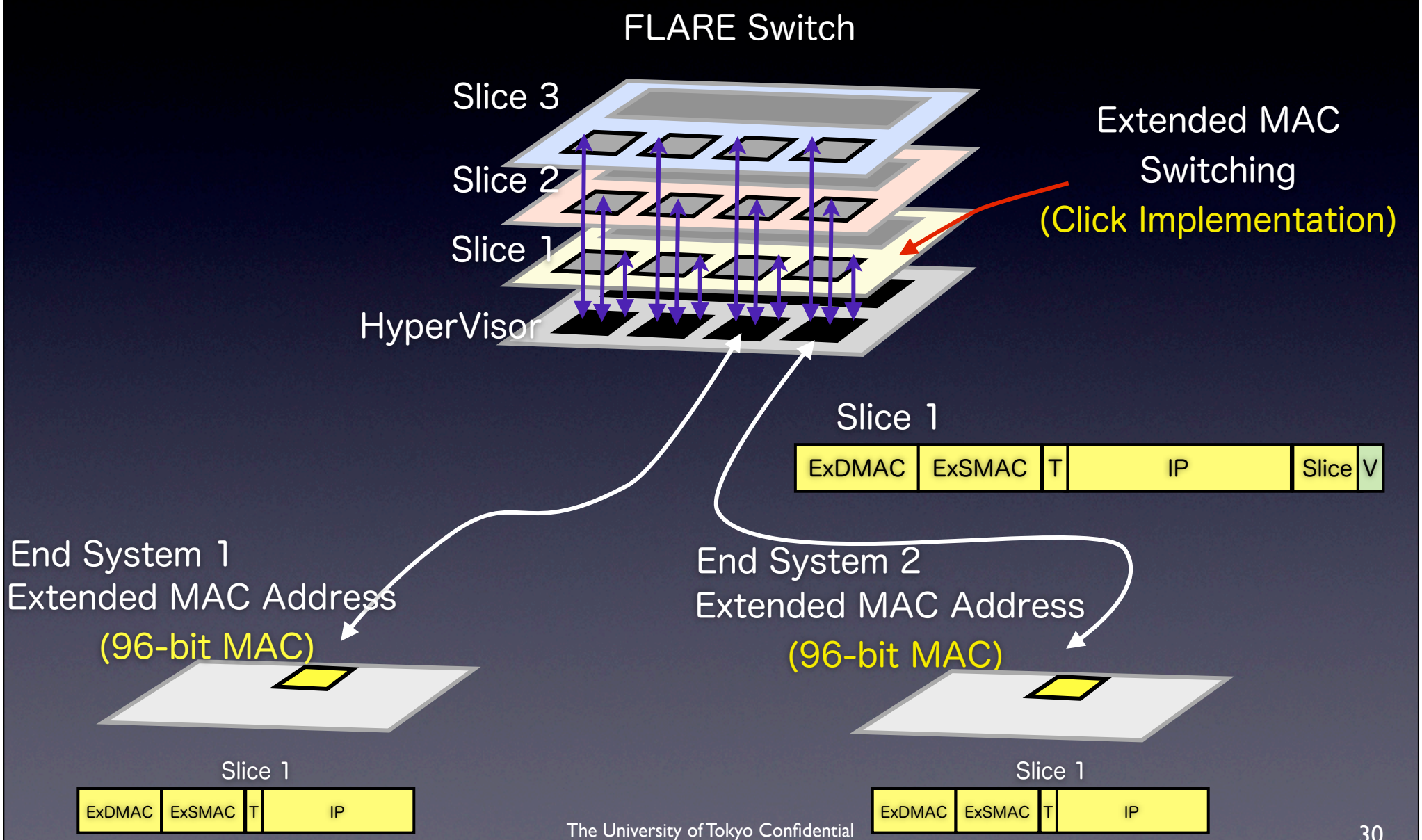
EUI-64 (64bit Extended Unique Identifier)



http://en.wikipedia.org/wiki/MAC_address

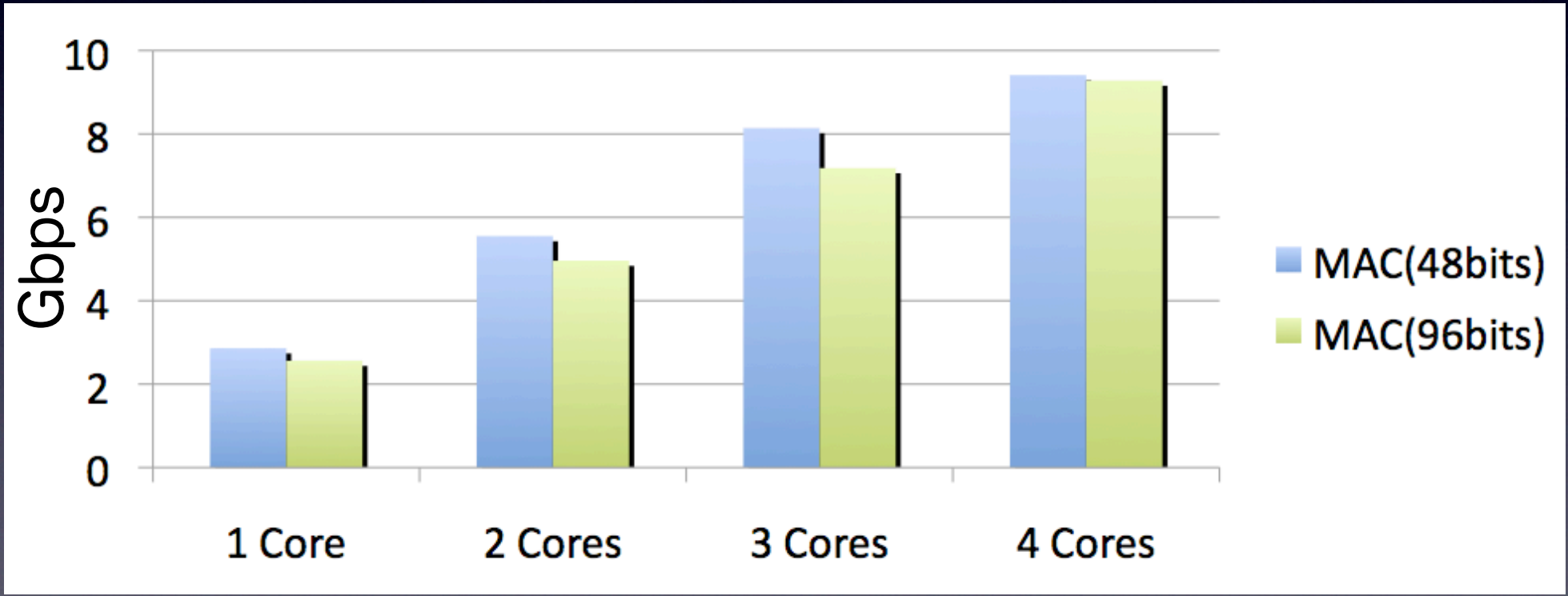
Why not just extend MAC ID space through deep programmability enabled by FLARE Switch?

Extended MAC (96bit) Switching

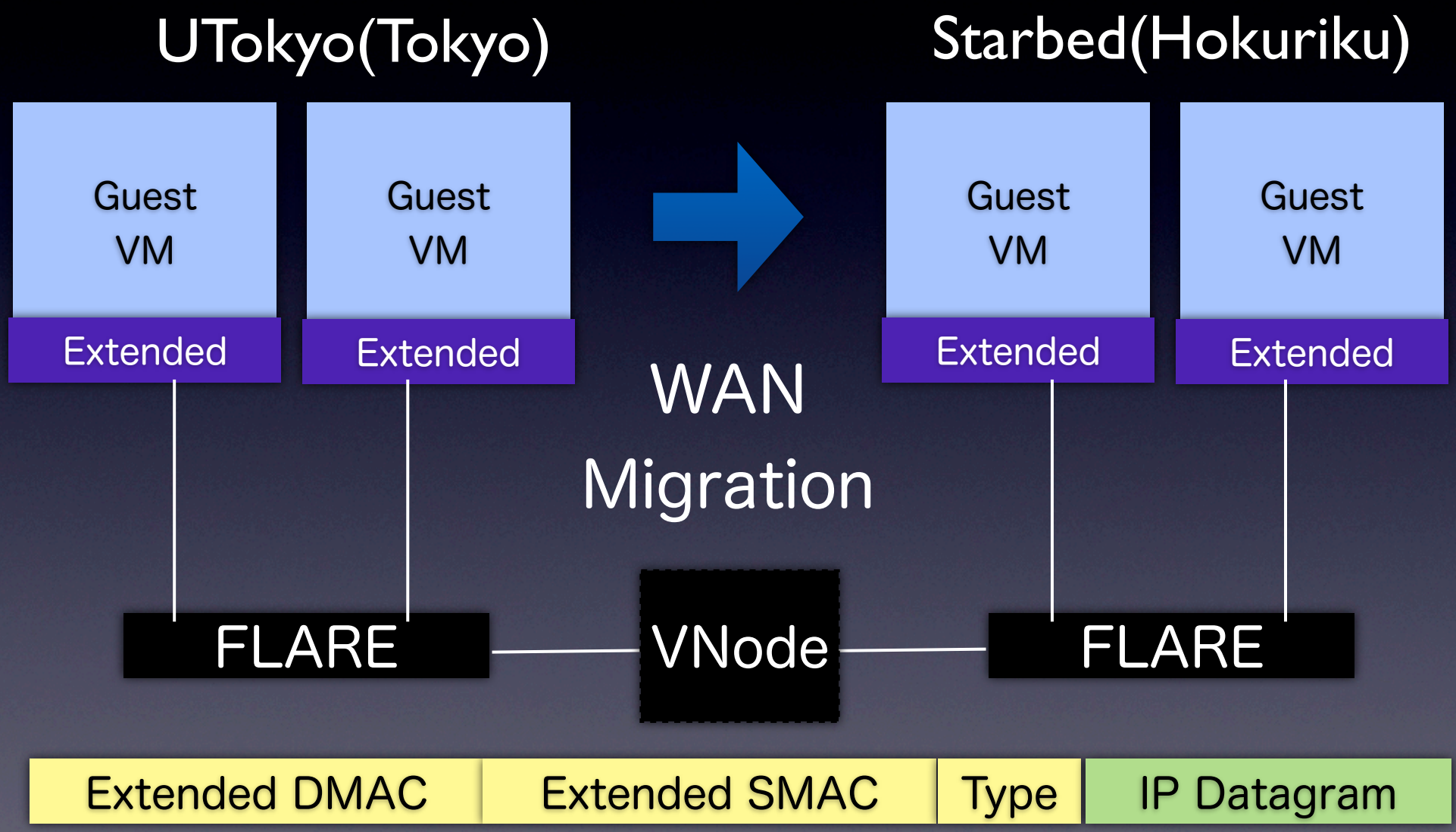


Performance Check

2 Port 10Gbps Switching Performance (Gbps)
MTU=1500 bytes



InterCloud DPN
(VM Migration With Extended MAC)



dell101.nakao-lab.org:1 (root)

Mon Jul 2, 3:17 PM

irtual Mac... | Virtual Machine Mana... | vm2-ub10 Virtual Mac...

vm1-ub10 Virtual Machine

File Virtual Machine View Send Key

Mon Jul 2, 3:17 PM duping

Help

duping@vm1-ub10: ~

```
11 00:2b:21 > ff:00:ff:ff h
n (0xff00), 0:
0:  ff00 ffff ffff 002b 212e 4c41 ff00 ffff
0:  ffff 002b 212e 4c42 0800 4500 0054 0000
0:  4000 4001 c141 1e1e 1e1e 64 0800
0:  40d3 6506 0066 de3c 0000 00 c060
0:  0300 0000 0000 1011 1213 1415 1617 1819
0:  1a1b 1c1d 1e1f 2021 2223
11 00:2b:21:2e:4c:42 > ff:00:ff:ff:ff:ff, eth
n (0xff00), length 110:
0:  ff00 ffff ffff 002b 212e 4c42 ff00 ffff
0:  ffff 002b 212e 4c41 0800 4500 0054 8f2c
```

ExDMAC

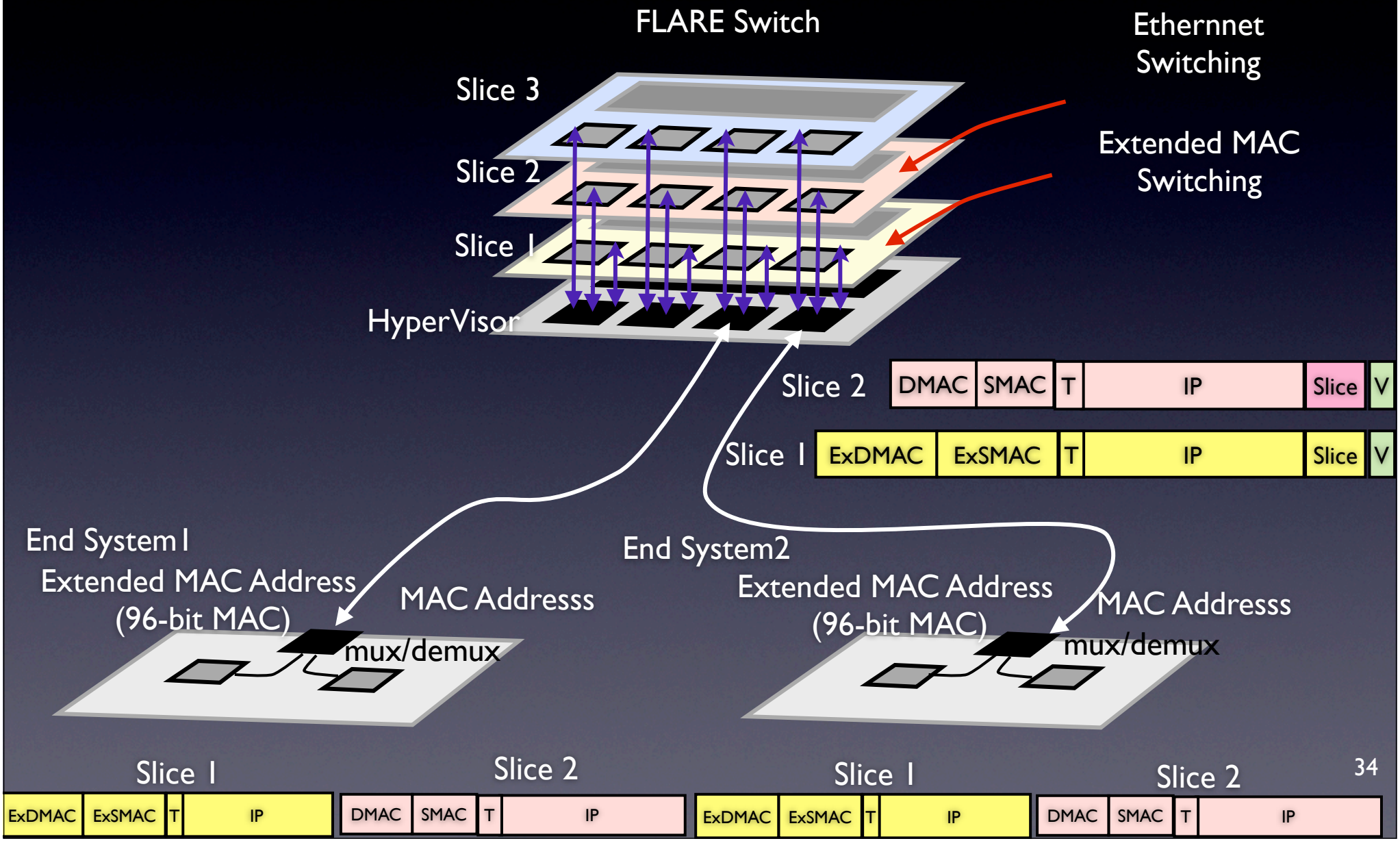
ExSMAC

TYPE

IP

```
time=0.531 ms
time=0.553 ms
time=0.543 ms
time=0.589 ms
time=1018 ms
time=18.7 ms
time=0.508 ms
time=0.550 ms
time=0.505 ms
time=0.556 ms
time=0.529 ms
time=0.554 ms
time=0.519 ms
time=0.544 ms
time=0.585 ms
time=0.527 ms
time=0.485 ms
```

Two Different Kinds of Switching via Tailer Slicing



Examples of Deep Programmability

- Cache Oriented Network Architecture (UTokyo)
- OpenFlow In A Slice (UTokyo)
- Content Oriented Network (Here and There)
- Crazy examples (Non-IP, Extended Mac)
- Artificial Delay for Traffic Control (UTokyo)

Conclusion

- SDN can be extended to Deeply Programmable Network
- We might recall what we initially aimed at, namely, by "Clean Slate" re-design of the Internet
- VNode/FLARE projects now put forth edge area network virtualization and deep programmability