Software Defined Data Plane
For Deep Data Plane Programmability

Aki Nakao
Professor, The University of Tokyo
Chairman, 5GMF Network Architecture Committee
2017/11/20
Challenges in Data Plane Programmability

- Ease of programming
- Processing & forwarding performance
  - Extending SDN Southbound / Actions
  - SDN/NFV cross-layer optimization
- Cost
Network virtualization platform

- **Common SDN**
  - Software based managing network with separating C-plane/D-plane
  - Cutting OPEX/CAPEX by automation by software and Constructing NW by common HW

- **Network virtualization platform**
  - Realizing “Deep programmability” by totally virtualization of networking and computing adding separating C/D-plane
  - Realizing service chaining without limitation of physical network
Our Research Activities on Network Slicing

Japan

2008 VNode Project Phase 1 (NICT/Utokyo/NTT/NEC/Hitachi/Fujitsu)
2011 Vnode/FLARE Project Phase 2 (Utokyo/NTT/NEC/Hitachi/Fujitsu/KDDI)
2014 Vnode Project Completion
2014 5G/IoT Slicing

US

2008 GENI Kick Off ($12M 29 institutions)
2009 GEC4 (Mar) GEC5(Jul) GEC6(Nov)
2010 GEC7 (Mar) GEC8(Jul) GEC9(Nov) plenary
2011 GEC10 (Mar) GEC11(Jul) GEC12(Nov) plenary
2012 GEC13 (Mar) GEC14(Jul) GEC15(Nov) plenary
2013 GEC16 (Mar) GEC17(Jul) GEC18(Nov)
2014 GEC19 (Mar) GEC20(Jul) GEC21(Nov) Best Demo
2014 GEC22 (Mar) GEC23(Jul) GEC24(Nov)

Worldwide

2002- PlanetLab

2008- PlanetLab

2008 GENI Kick Off ($12M 29 institutions)
2009 GEC4 (Mar) GEC5(Jul) GEC6(Nov)
2010 GEC7 (Mar) GEC8(Jul) GEC9(Nov) plenary
2011 GEC10 (Mar) GEC11(Jul) GEC12(Nov) plenary
2012 GEC13 (Mar) GEC14(Jul) GEC15(Nov) plenary
2013 GEC16 (Mar) GEC17(Jul) GEC18(Nov)
2014 GEC19 (Mar) GEC20(Jul) GEC21(Nov) Best Demo
2014 GEC22 (Mar) GEC23(Jul) GEC24(Nov)

All Rights Reserved by Akihiro Nakao, 2017
FLARE Board V1.3 (New)

36 Core Sliceable Data Plane Board
FLARE Node Architecture

South Bound APIs
Distributed Controller Design

C-plane (Intel x86 processor)

D-plane (Many core processor)

Node Manager

Slicer Controller

Slicer

C-plane

D-plane

C-plane

D-plane

C-plane

D-plane

Physical Network Interfaces

VIF

VIF

VIF

Virtual Forwarding Context (VFC)
Slice Architecture on NPU

LXC: Linux Container on Zero Overhead Linux (ZOL)
FLARE D-plane H/W Platforms

- Many Core NPU
- Mellanox TileGX36
- Mellanox TileGX72
- Mellanox BlueField (Planned)
- Many Core CPU
- Intel x86 Only (DPDK)
- AMD Epyc/ Threadripper (On-Going)
- Reconfigurable ASIC
  - Intel x86 + Cavium Thunder X (Planned)
  - Intel x86 + Barefoot/P4 (Planned)
TILE-Gx36™:
Scaling to a broad range of applications

- 36 Processor Cores
- 866M, 1.2GHz, 1.5GHz clk
- 12 MBytes total cache
- 40 Gbps total packet I/O
  - 4 ports 10GbE (XAUI)
  - 16 ports 1GbE (SGMII)
- 48 Gbps PCIe I/O
  - 2 16Gbps Stream IO ports
- Wire-speed packet engine
  - 60Mpps
- MiCA engine:
  - 20 Gbps crypto
  - Compress & decompress
Take Inspiration from ASICs

ASICs have high performance and low power
- Custom-routed, short wires
- Lots of ALUs, registers, memories – huge on-chip parallelism

But how to build a programmable chip?
Replace Long Wires with Routed Interconnect

[IEEE Computer ’97]
... To Distributed ALUs, Routed Bypass Network

Scalar Operand Network (SON) [TPDS 2005]
...to a Distributed Shared Cache
Distributed Everything + Routed Interconnect → Tiled Multicore

Each tile is a processor, so programmable
TILE-Gx36™:
Scaling to a broad range of applications

- 36 Processor Cores
- 866M, 1.2GHz, 1.5GHz clk
- 12 MBytes total cache

- 40 Gbps total packet I/O
  - 4 ports 10GbE (XAUI)
  - 16 ports 1GbE (SGMII)
- 48 Gbps PCIe I/O
  - 2 16Gbps Stream IO ports
- Wire-speed packet engine
  - 60Mpps

- MiCA engine:
  - 20 Gbps crypto
  - Compress & decompress
Seeking viable applications of Software Defined Data Plane...
Application-Specific MEC Processing

Classify packets by App to NFV_VLANs

Classify reverse traffic to NFV_VLANs

MEC Server

NFV_VLAN 1 (HTTP Cache)
NFV_VLAN 2 (Transcoding)
NFV_VLAN 3 (BW Control)
NFV_VLAN k (Pass-through)

Internet

FLARE1

FLARE2

Chrome
YouTube
Tethering
Default

All Rights Reserved by Akihiro Nakao, 2017
Application Identification

Remote console of programmable network node (FLARE)

Smartphone connected to our MVNO
Smartphones/wearables attach the information of applications and devices at the trailers of TCP SYNs. FLARE detects the information and creates mapping between flows and the information on applications and/or devices.
Application Specific Traffic Breakdown

Chrome, Total, Facebook, Tethering, YouTube

Data categories: total, tethering, mediaserver, com.android.chrome

Time periods: Tue 4PM, Tue 8PM, Wed 12AM, Wed 4AM, Wed 8AM, Wed 12P

All Rights Reserved by Akihiro Nakao, 2017
Per-Applications QoS

```bash
./bandwidth.sh up_down_firefox_100k.json
```

```json
[{
  "ip":"flare://any",
  "app":"firefox.exe",
  "direction":"up",
  "bandwidth":"100"
},
{
  "ip":"flare://any",
  "app":"firefox.exe",
  "direction":"down",
  "bandwidth":"100"
}
```

Applications

Upstream

Any UE

Specific Application

Downstream 100kbps

Applications

100kbps
Anomaly Detection algorithm → Hierarchical Heavy Hitters Revisited

Prefix-length-sum as aggregation order

Make /32 higher in the order
Challenges in Data Plane Programmability

- Ease of programming
- Processing & forwarding performance
  - Extending SDN Southbound / Actions
  - SDN/NFV cross-layer optimization
- Cost

And “viable” applications...