Software-Defined Data Plane
Enhancing SDN and NFV

Aki Nakao
The University of Tokyo
GSC 18
2014/7/22
SDN Architecture

Applications (NFV)

Network Applications

Control Plane

Control-Plane Elements

Data Plane

Data-Plane Elements

North Bound Interface (NBI)

South Bound Interface (SBI)
Application Driven Thinking

Applications (NFV)

Network Applications

Control Plane

Control-Plane Elements

Data Plane

Data-Plane Elements

Current SDN (bottom up)

Future SDN should be (top down)
Application-Driven Thinking

Premise:
Programmable networking has been enabled by SDN and NFV

App-Driven Thinking:

- Think of “(killer) applications” first and then design network functions and interfaces APIs for SDN and NFV
- Not just OPEX/CAPEX reduction but create new values via SDN and NFV
Lets start with an example application!

“Next-Gen MVNO”
MVNO Customers Need More Flexible Subscription

Bandwidth Control According to Apps!

High Speed Mode (pay as you go)

App-specific traffic control enables more fine-grained subscription plans that can get an MVNO out of the “low cost” competition.
Application Driven SDN

There is a gap between application developers and SDN’s programming interface

• The current SDN is targeted at network operators (researchers) and not at application developers

• Flow abstraction in Southbound Interface is for operators

  <Flow Pattern> <Action> <Stat>

• App/Device abstraction is useful and intuitive for application developers

  <App/Device> <Action><Stat>

Solution: Application-Driven Software Defined Networking
GENI Engineering Conference 20 Demo:
App-Driven SDN QoS

Smartphone

FLARE
(Data Plane Programmable)

Chrome: Pass Thru
Firefox: Block
YouTube: Rate Limit
Our Proposal

FLARE (Deeply Programmable Node)
Parse and remove “trailers” and map between flows and apps

SDN Controller
Traffic Engineering based on headers

Packet Marking

SDN Controller

Smartphones attach app/device information to packets
FLARE detects app/device information and creates mapping between flows and apps/devices
Demo
We won the best demo award!
GEC20@UC Davis
Benefits

- Application Specific Traffic Engineering for MVNO
  - Application Name Based
  - Application Process Based (Fore/Background)
  - Device Type Based
  - Device State Based (Context / Location Aware)
- Parental Control
  - Not by apps on devices, but by networking
- Additional Value-Add services for specific applications
  - Differentiation for competing apps (e.g., Chrome vs. Firefox)
Contributions

• Insert application information in trailers of packets (e.g., TCP SYN)
  • Extensible to other protocols than TCP
  • Similar to “Google Fast TCP Open” but for different purpose
  • More bits usable than TCP/IP options

• Determine applications with 100% Accuracy
  • Cooperation between end-systems and programmable nodes

• Extensible to supervised learning without app
  • Machine learning using sampled data with app
Software Defined Data Plane

Applications

Network Applications

Packet Process

North-Bound Interface (NBI)

Control Plane

Control-Plane Elements

Packet Process

Data Plane

Programmable Data-Plane Elements

Publish API

E.g. OpenFlow Switches
Accelerate SDN Standardization

Operation and Evaluation Feedback -> Accelerate Standardization

Network Applications

Southbound Interface

Data-Plane Elements

Application Driven Thinking
Flow rule to match tcp syn packet #70

Closed  nitin456 opened this issue on Nov 14, 2013 · 1 comment

nitin456 commented on Nov 14, 2013

Hi All,

How can we insert a flow rule in open flow switch 1.3 to match all TCP SYN packets?
Hope for quick response.

Thanks,

Nitin

ederl commented on Nov 22, 2013

Sorry for the delay.

The OpenFlow spec does not support this, but it is not hard to add this to the code.

ederl closed this on Nov 22, 2013
Sliceable Software Defined Data Planes

Applications

Network Applications

Packet Process

North-Bound Interface (NBI)

Control Plane

Control-Plane Elements

Packet Process

Publish API

Data Plane

Programmable Data-Plane Elements

E.g. OpenFlow Switches
SDN data plane and NFV could be unified

SDN for Network Control

Orchestrator

NFV for data processing

Applications

Network Applications

Control-Plane Elements

Packet Process

Control-Plane Elements

Packet Process

North-Bound Interface (NBI)

Programmable Data-Plane Elements

Programmable Data-Plane Elements

Programmable Data-Plane Elements

Programmable Data-Plane Elements

E.g. OpenFlow Switches
FLARE
(Unifying SDN Software Defined Data Plane and NFV)

Physical Resources: CPU + NPU

- CPU
  - Intel x86
- Data Bus
  - PCIe Gen3
- NPU(s)
  - Tilera
- NIF

White Box
Physical Resources: CPU + NPU (+ GPU)
FLARE

Virtualization (Resource Container) -> Slices of Resources

Slices (Virtual Resource Containers)

Sliceable!
FLARE

- NFV Sliceable Programmable D-plane
- SDN Sliceable Programmable Control Plane

Physical Resources

- GPU
- Data Bus
- CPU
- Data Bus
- NPU(s)
- NIF

Slices (Virtual Resource Containers)
Hierarchical Resource Management

- General Purpose Processor(s)
- Network Processor(s)
- ...and more types of processors
FLARE-EX Prototype (32x10Gbps)
Toy-Block Networking

Deploying composite functions over federated slices
ToyBlock Networking GUI
Standardization Possibility

• Deep (Data Plane) Programmability
  • Slicing (Virtualization)
  • Evolvable APIs
  • Viewed as extension to SDN
  • Also viewed as extension to NFV
  • Rethinking architecture of SDN/NFV for enabling “Deep programmability”

• Application Driven Thinking
  • Top-Down approach is necessity

• Toy Block Networking
  • Accommodate a wide range of programmers
  • Marketing of reusable network function “Toy Blocks”
Conclusion

“Software Eats Everything”

“Application Driven Thinking empowered with Software Defined Data Plane will extend SDN/NFV further and accelerate standardization!”