Data Plane Programmability in SDN

H. Farhady, H. Lee, A. Nakao

The University of Tokyo

CoolSDN’14
Contents

• Current SDN

• The Gap: Data plane programmability

• Some directions for the community
Intro to SDN

• Cloud computing
  – Improving Servers: OS virtualization, Centralized management
  – Improving the Network: Happening now .... (Network Virtualization, Centralized Control)

• SDN attacked the second problem
The History

• Mainframes --> PC
  – Customized --> General-purpose
  – (Market) Drivers
    • CAPEX: Cheaper devices
    • OPEX: Fewer administrators

• SS7 telephony --> Intelligent Networks (IN)
  – Hardwired --> Programmable
  – Drivers
    • Same as above
    • VAS, e.g. 0800-xxx

• SDN related history: Active Nets, ForCES, GMPLS
SDN

• Main proposal
  – Decouple control plane from data plane
  – Centralize the control plane
  – Provide programmability in the control plane

• Solution Framework components
  – Control plane, i.e. the controller
  – Data plane, SDN-enabled switches
  – Controller (aka SDN) Apps

• Dominant realization: OpenFlow
(some) Success Factors

• Engaging industry from the first day
• Providing reference implementation
  – Later transferred to OpenDaylight
• NPO/NGO/PPP think tank back-end
  – Crystallized as ONF
  – Standards, discussions, certification etc.
## Current SDN Research
(a snapshot)

### Controller implementations
- NOX, POX, Maestro, Ryu, MUL, Beacon, Floodlight, Jaxon, NodeFlow
- ovs-controller, IRIS, OESS, Flowvisor, RouteFlow, SNAC, Resonance

### Controller design aspects
- State consistency, scalability, flexibility, security, availability, placement

### Tools
- **Simulator**: Mininet, ns-3, OMNeT++, EstiNet 8.0
- **Framework**: Trema, Mirage, Wakame-vdc
- **Debugger**: OFlops, Cbench, NICE, STS, OFTest, OFRewind, Frenetic
- **Programming Languages**: FML, Procera, Frenetic, FRESCO (partially)

### Switches
- **Software switch**: Open vSwitch, OpenFaucet, CPqD ofsoftswitch13, Indigo, Pantou
- **Commercial switch**:
  - Arista (7050 series), Brocade (MLX, NetIron CES, NetIron XMR series, ICX 7750)
  - Dell Force10 (Z9000, S4810), Extreme networks (Summit X440/460/480/670)
  - HP (3500/3800/5400/6200/6600/8200), IBM (RackSwitch G8264), Juniper (MX series)
  - NEC (PF5240/5248/5820, PF1000 virtual switch), NoviFlow (NoviSwitch 1248/1132)
  - Pica8 (P-3290/3295/3780/3922)

### Testbeds
- Planetlab Europe, OFELIA (EU), National LambdaRail, Internet2, COTN (USA)
- VNode, RISE (JP), SURFnet (NL)

### Standardization
- ONF, IETF ForCES, IRTF SDNRG, ITU-T SG 11,13

### Application areas
- **Networks**: Data center, optical network, wireless network, home network
- **Softwares**: Network management, security, middle box, VM migration

Where is the focus?
The Gap

• The lack of research on data plane

• Why? Maybe ...
  – Since the long lasting belief about custom-built, hardware-centric, one-size-fits-all data plane
  – Advertized by vendors and some academia

• Software-centric, programmable data plane unlocks the innovation on data plane
Data Plane Programmability (motivations)

• Many innovative ideas need to touch data plane
  – Flexible Measurement (e.g., OpenSketch)
  – New architectures (e.g., NDN, XIA)
  – New protocols (e.g., New layer 2)
  – TCAM-less Switching

• Due to the advancements in the commodity HW industry we believe software data plane is realizable
The Opportunity

• We can reuse existing technologies to drive SDN data plane research engine faster
  – Again, since SDN is not a new concept
• Examples
  – Packet classification and Forwarding
  – Easy Programmability
  – Resource allocation
  – Security
  – State full packet processing
  – Wireless networking
  – Network Measurement
Deeply Programmable Networks

Applications

Northbound Interface

Controller

Southbound Interface

Flow table

CPU, storage, queue, etc.

Switch

Apps

Control plane

Data plane

SDN

DPN
Conclusion

• We believe SDN community should pay more attention to Data Plane and Data plane programmability research
• We can adapt already existing technologies to SDN