The Challenges of SDN/OpenFlow in an Operational and Large-scale Network

Jun Bi

Tsinghua University/CERNET

OPEN NETWORKING SUMMIT
April 17 2012

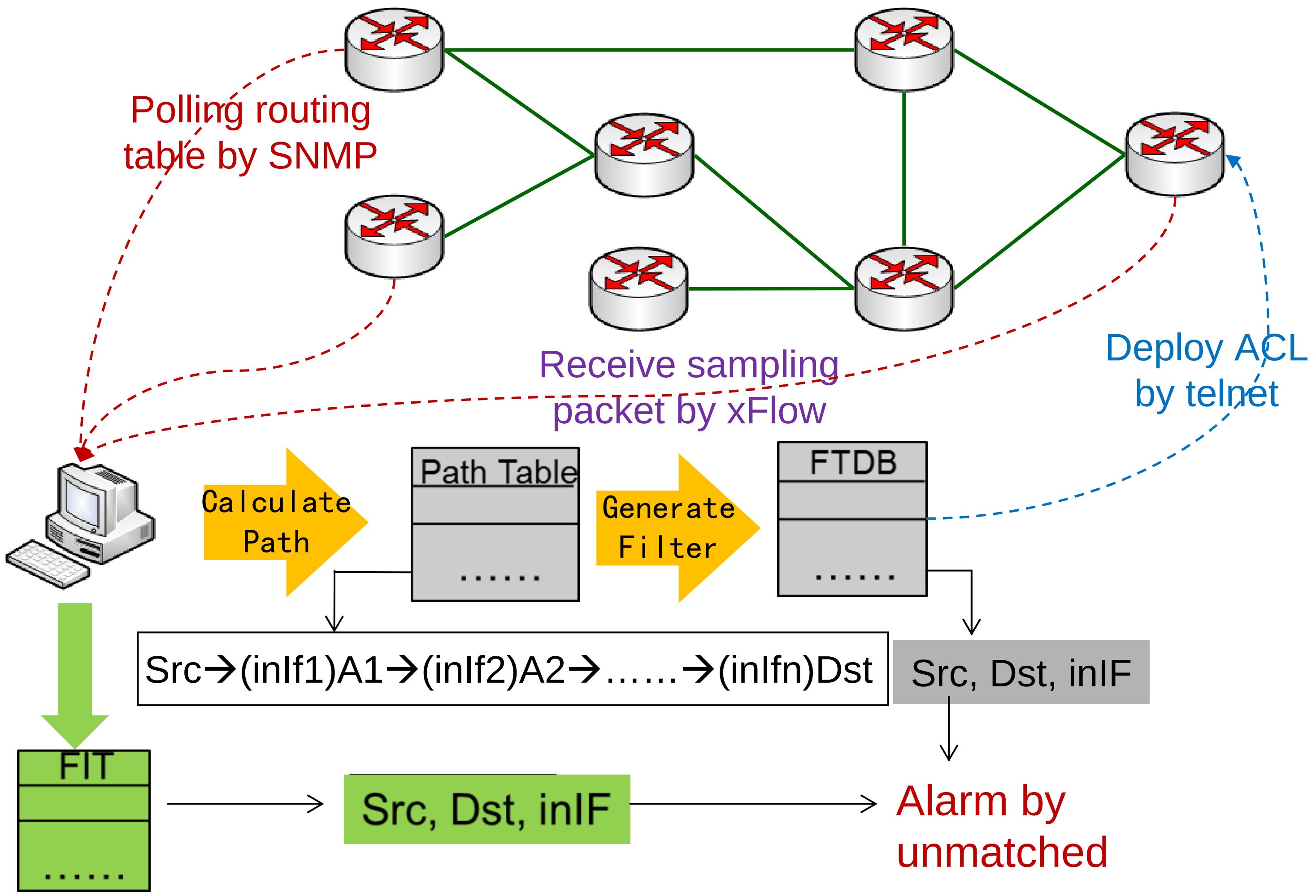
Outline

- Intra-AS (campus level) IPv6 source address validation using OpenFlow (with extension)
  - Good for introducing new IP services to network
- Planning next step if we run SDN as a common infrastructure for new services and architectures
  - Some personal viewpoints and thoughts on design challenges
  - Forwarding abstraction for Post-IP architectures
  - Control abstraction for scalable NOS and programmable /manageable virtualization platform
  - Inter-AS policies negotiation abstraction

Source Address Validation

- Source address spoofing still a problem
  - Arbor annual net. sec. report, MIT spoofer project, NANOG discussions
- False positive of uRPF due to generating filtering without global knowledge
  - e.g. asymmetric route, static route, fast reroute, ECMP
- We proposed CPF (Calculated Path Filtering)
  - An intra-AS source address validation (campus level)
  - Calculating Path Filtering based global knowledge
  - Implemented with SNMP, xFlow and Telnet in IP network
  - Deployed in 100 IPv6 campus networks of CERNET2
  - New version using Openflow

CPF Overview



FIT: Flow Info Table FTDB: Filter Table DB

Problems We Met

- Technical challenges (details in next slides)
  - No standard interfaces between control software and vendor devices, esp. for IPv6, which is new
  - No direct/full (internal) control to devices for operator's control software (all interfaces are in-direct control)

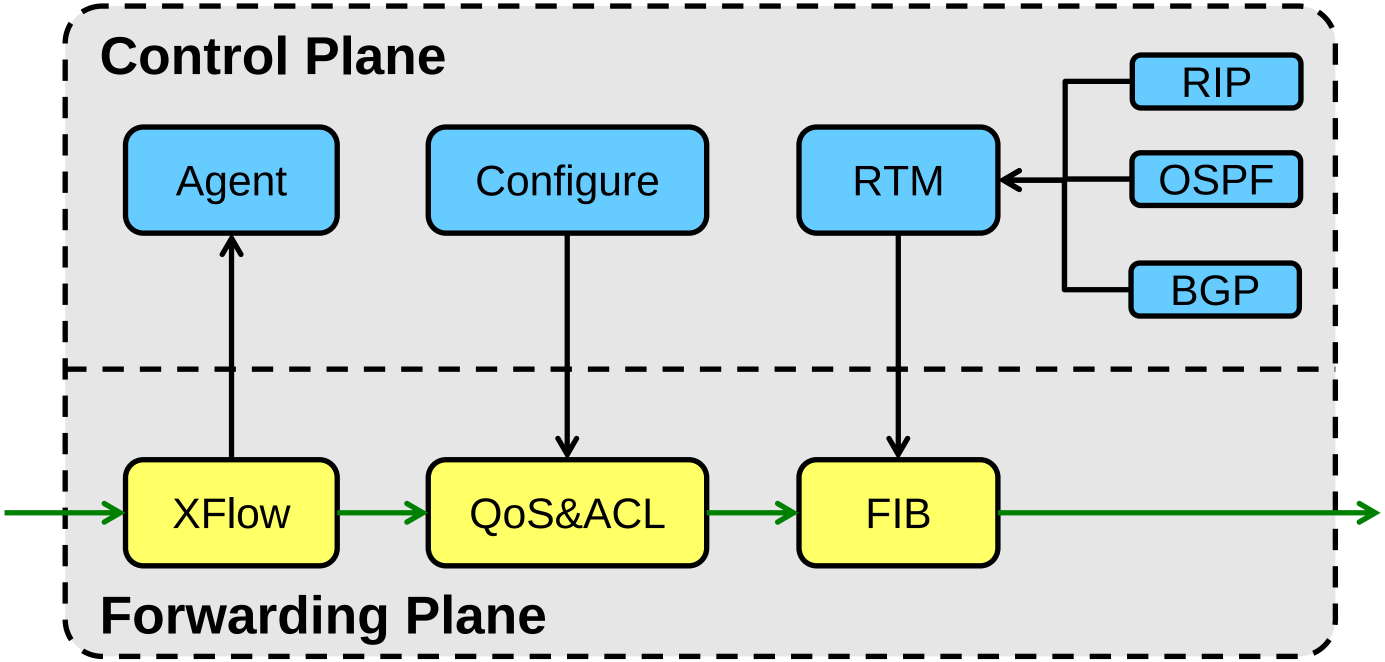
Technical Challenges We Met

- Getting routing table by SNMP ✓ OpenFlow
  - Poor compatibility and inconsistency of vendor implementation
    - ✓ IPV6-MIB (RFC2465)
    - ✓ IP Forwarding Table MIB (RFC4292/RFC4293)
    - ✓ Private MIB (Cisco)
- Configuring ACL by Telnet ✓ OpenFlow
  - Manual setting rather than automatic setting by scripts
    - ✓ scripts are not smart enough and weak for complex control
- Sampling packet by xFlow + OpenFlow Extension
  - Multiple sampling protocols
    - ✓ NetFlow/Net Streams: router vendors - Cisco/Huawei
    - ✓ sFlow: layer 3 switches vendors – ZTE, H3C, DCN, Ruijie
- Polling network status by snmp + OpenFlow Extension
  - Passive cognizance of network state changes
  - Longer convergence time may cause slight false positive when network change

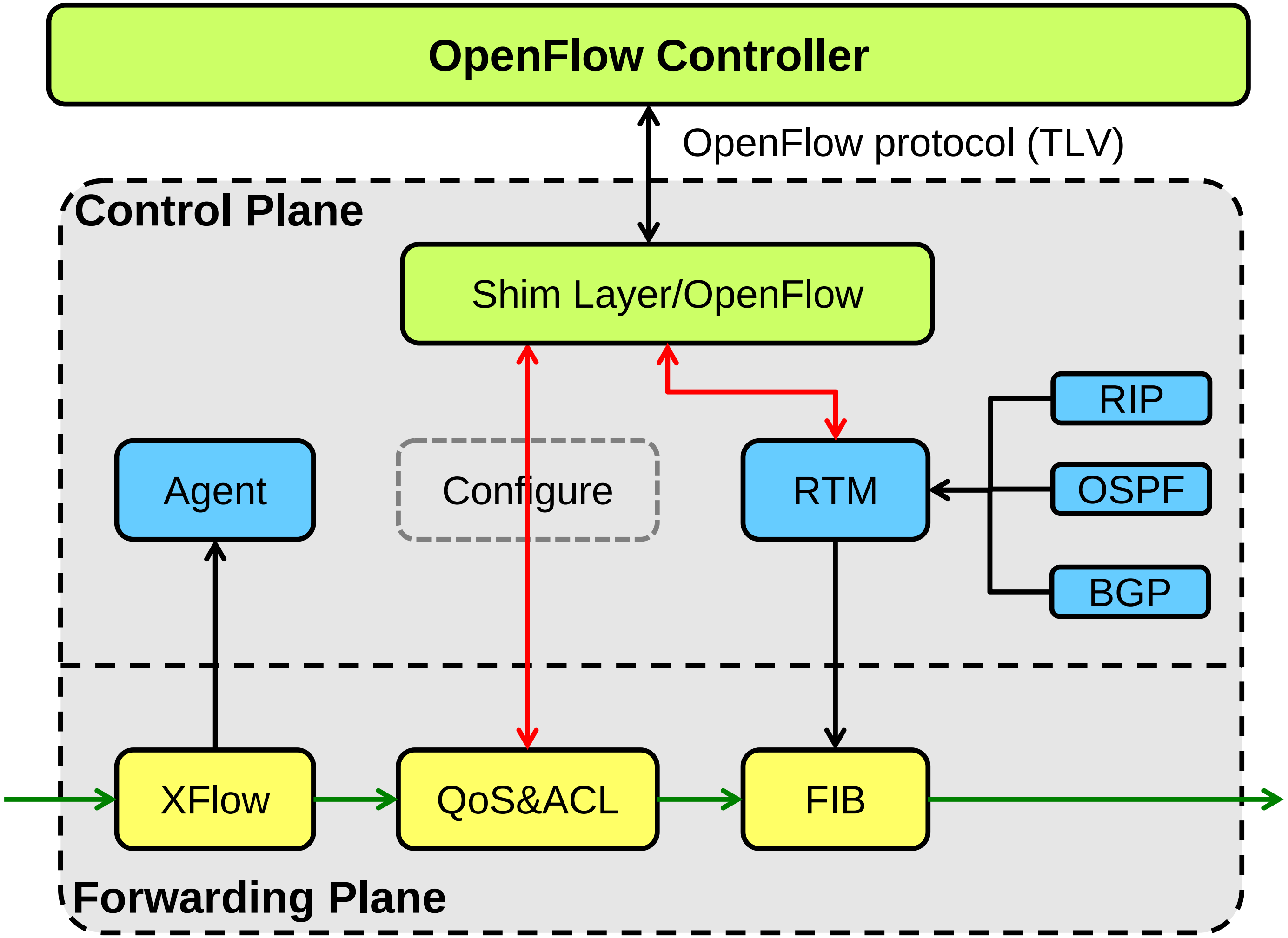
Choosing OpenFlow for CPF

- Architecture consideration
  - CPF's central control architecture
  - Flexible for deployment of innovative but long tail new functions, esp. for universities' research
- Interfaces standardization consideration
  - OpenFlow protocol to unify multiple protocols between control and device – shown in last page
- Implementation consideration
  - Easy for upgrade and deployment at legacy routers in a operational campus network – "*OpenRouter*"
  - Forwarding abstraction based on legacy hardware by adding a "OpenFlow shim layer" in software
- Network cognizance consideration
  - RIB changes/packet sampling– "*OpenFlow+*" (with extension)

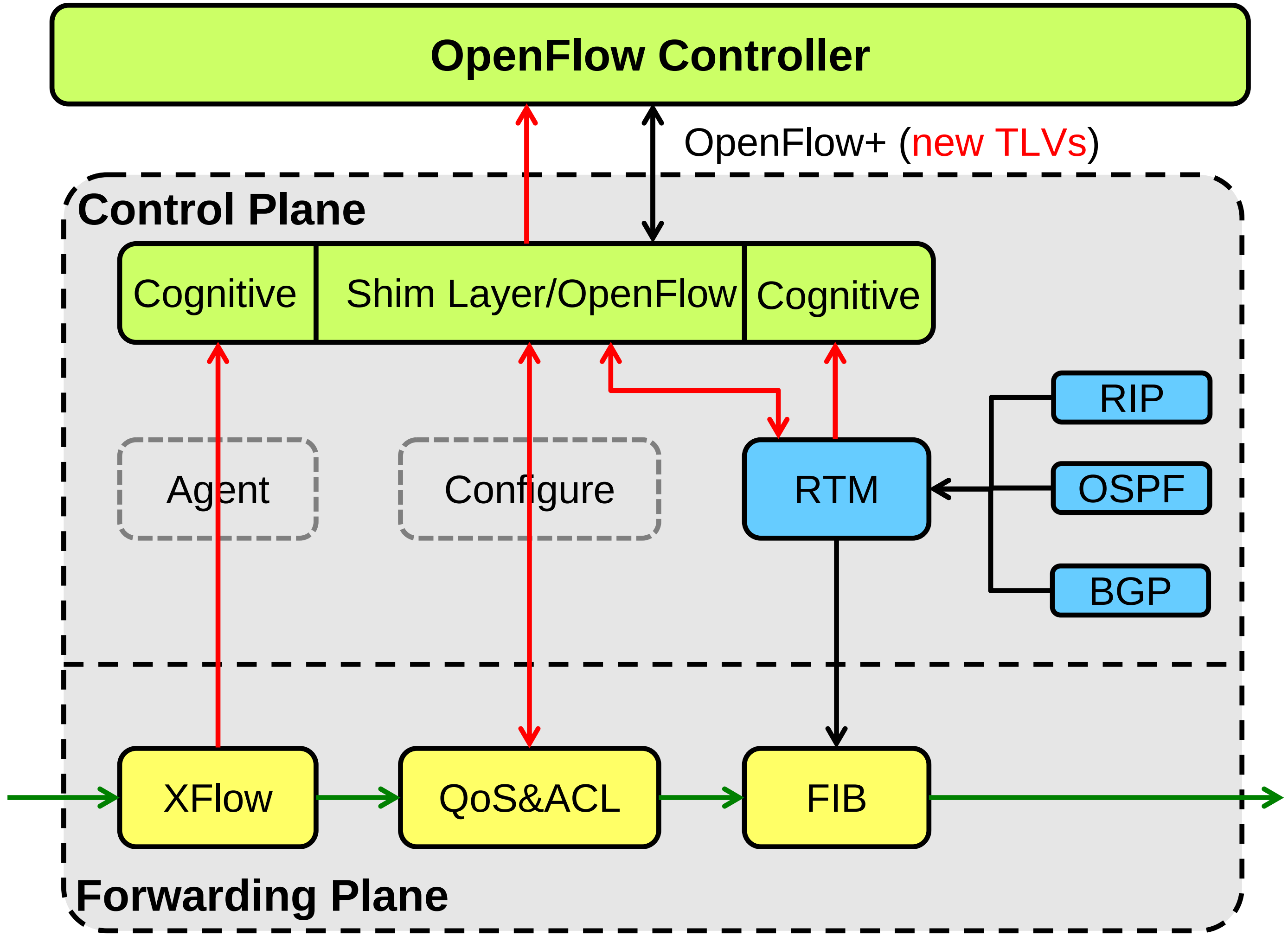
Current Router Architecture



Forwarding Abstraction Based on Existing Hardware



Some Autonomic Functions



Xflow: Packet Cognitive

RTM: L3 IPv6 Cognitive

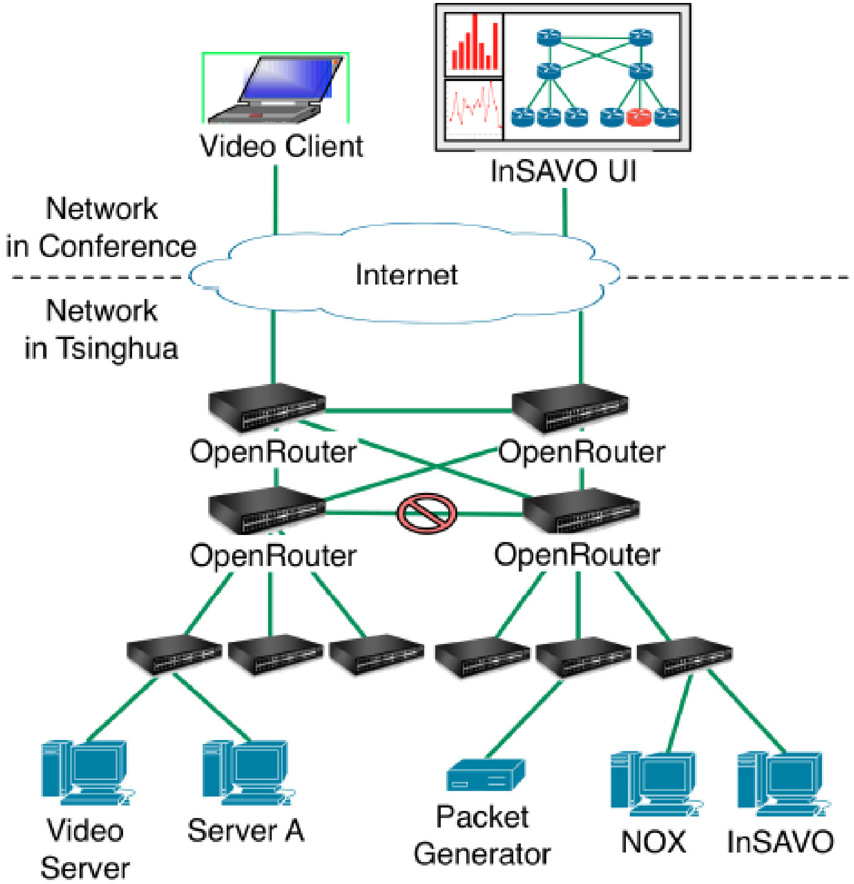
Implementation and TestBed

- Implementation
  - OpenRouter implementation: based on a commercial router: DCN DCRS5980
  - Controller implementation: APP/NOX loose couple mode for scalability (by socket communication)

- TestBed in Tsinghua campus.
  - OpenRouters
  - Openflow switches: PCs with NetFPGA Cards
  - Controller/App: NOX&CPF
  - Packet generator

CFP at Openflow Testbed

- CPF as a application example at Openflow testbed
  - Intra-AS source address validation based On OpenFlow (INFOCOM2012 Demo)
- Results
  - Easy for implementation
  - Easy for CPF function revision
  - Easy for deployment
  - Reduce filtering false positive caused by dynamic network change
- Can we do more like this?
  - Introducing new services
  - Introducing new net arch



We are thinking the next steps

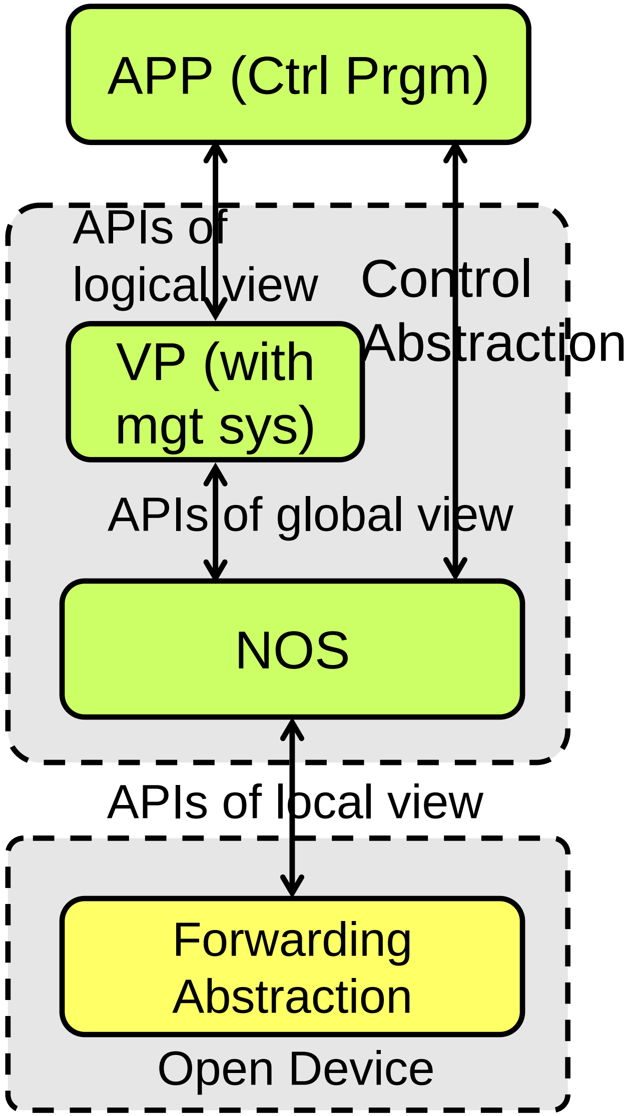
- Large-scale network
  - 100 campus networks of CNGI-CERNET2
  - 25 ASes (core nodes)
- May try multiple APPs
  - intra-AS SAV (CPF)
  - inter-AS SAV
  - NDN/CCN
  - New IP services or
  - New network architectures

Planning a open innovation platform for new net arch (FINE)

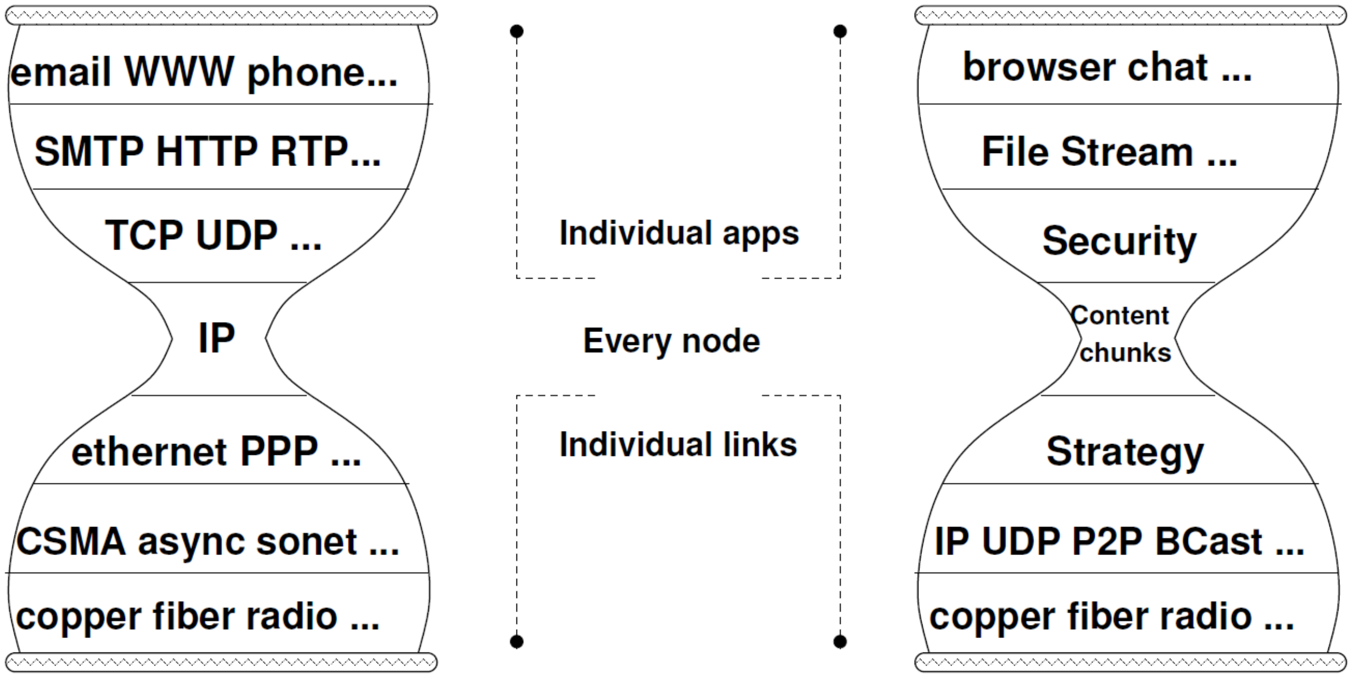
Considering SDN as the fundamental infrastructure

Some SDN Design Challenges in an Operational and Large-scale Network

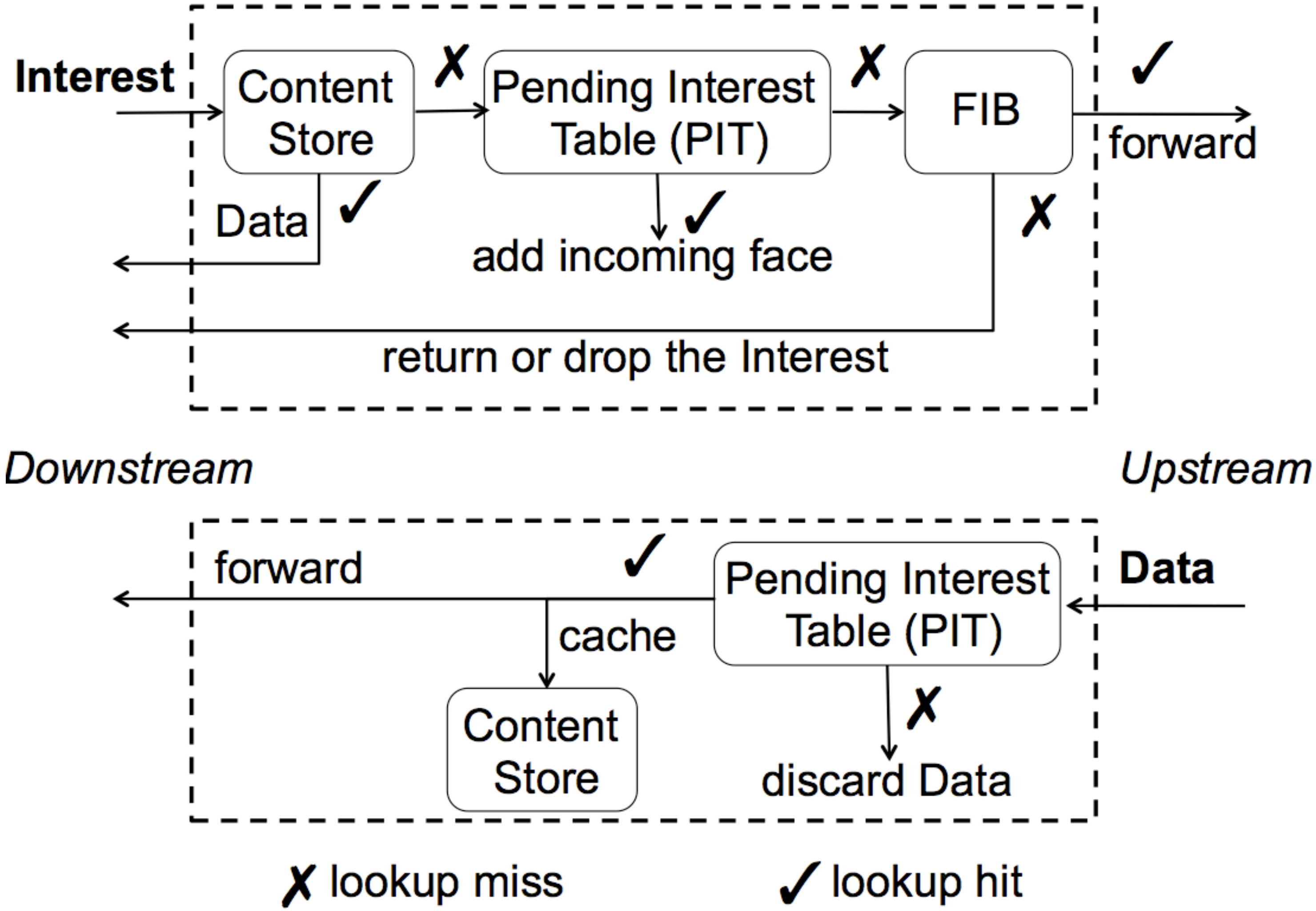
- For *Intra-domain* (abstractions for programmable control)
  - – *Forwarding Abstraction* providing APIs local or device view
  - ✓ Post-IP forwarding abstraction (taking NDN as an example)
  - – *Control Abstractions*
  - ✓ *Network Operation System* (NOS) providing global physical view
  - ✓ *Virtualization Platform* (VP) with mgmt sys and development tool, providing APIs of logical view
- For *Inter-domain* (abstraction for programmable negotiation)
  - – Standard *Inter-domain policies negotiation* (IPN) abstraction



NDN/CCN Architecture

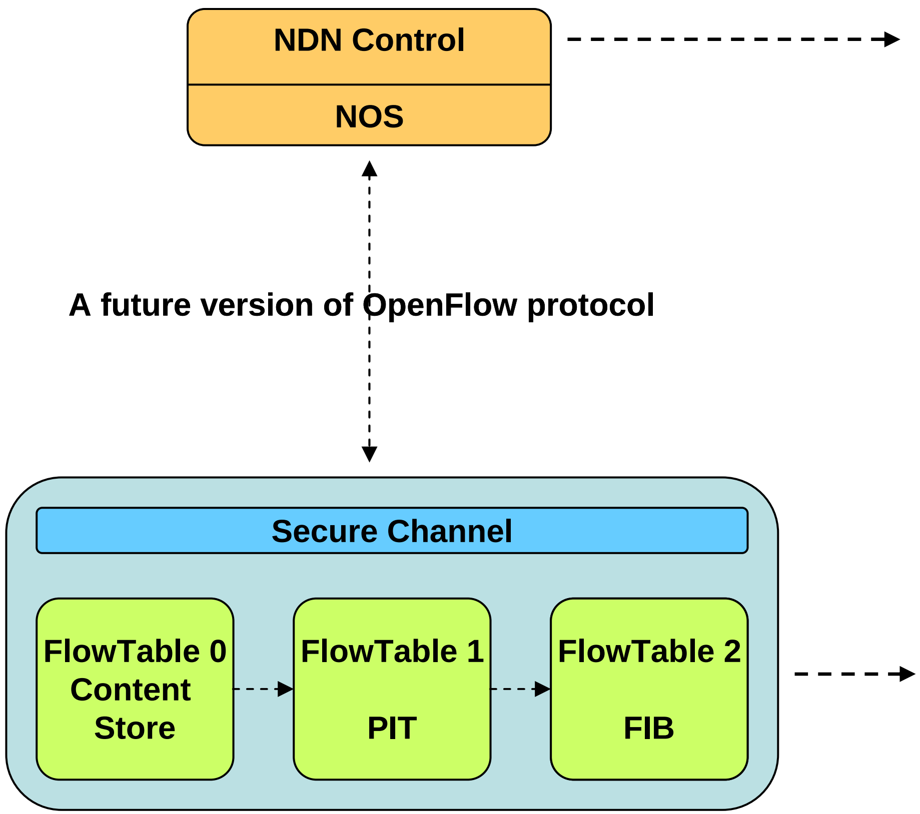


NDN/CCN Forwarding



Challenge 1: Post-IP Forwarding (NDN over SDN)

Basic Framework for NDN running over SDN

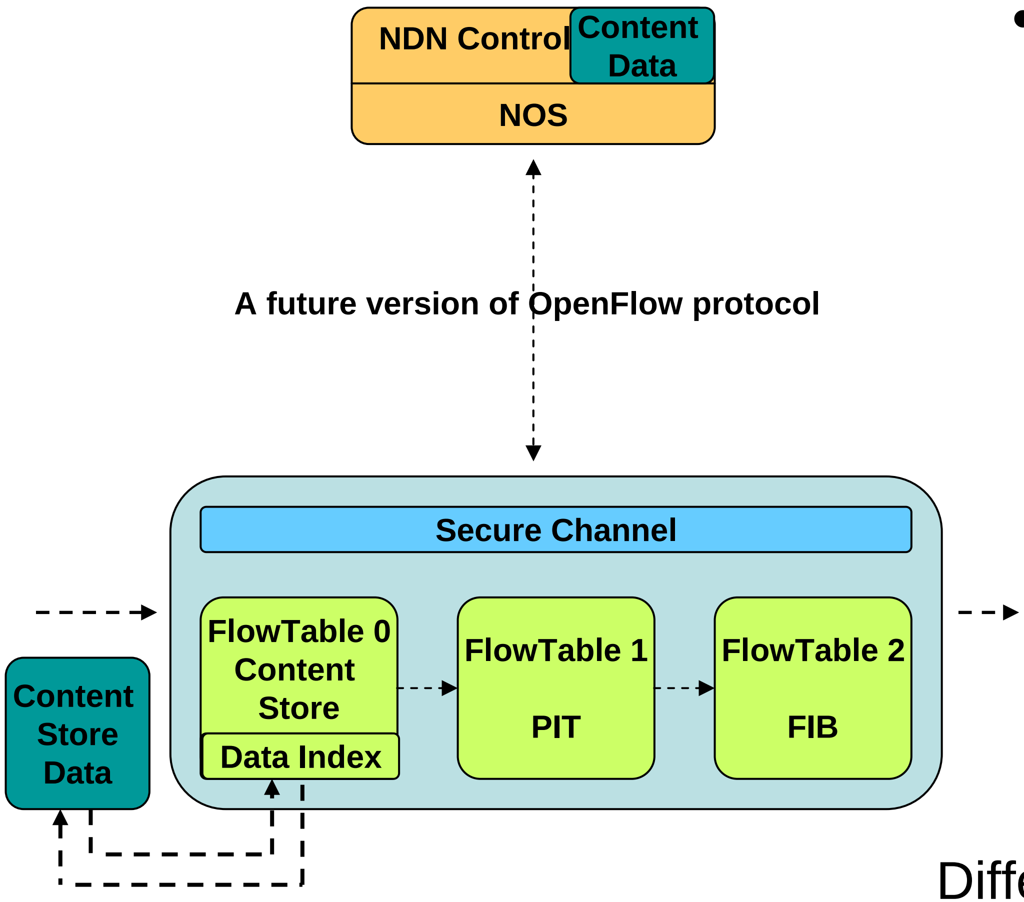


1. NDN Routing Algorithm;
2. NDN Cache Strategy;
3. NDN Management;

1. Three Tables: CS, PIT, FIB;
2. FIB is generated by Controller;
3. PIT and CS are generated automatically by users' interest/data packets;
4. PIT and CS may also be monitored and changed by controller;

Challenge 1: Post-IP Forwarding (NDN over SDN)

Discussion 1: How to add new *forwarding components*



- E.g. where to store the Content Data?

Position1: inside OF switch,
directly stored in FlowTable 0;

Position2: outside OF switch,
stored in a bypass memory devices;

Position3: stored in NOS
(storage in the cloud);

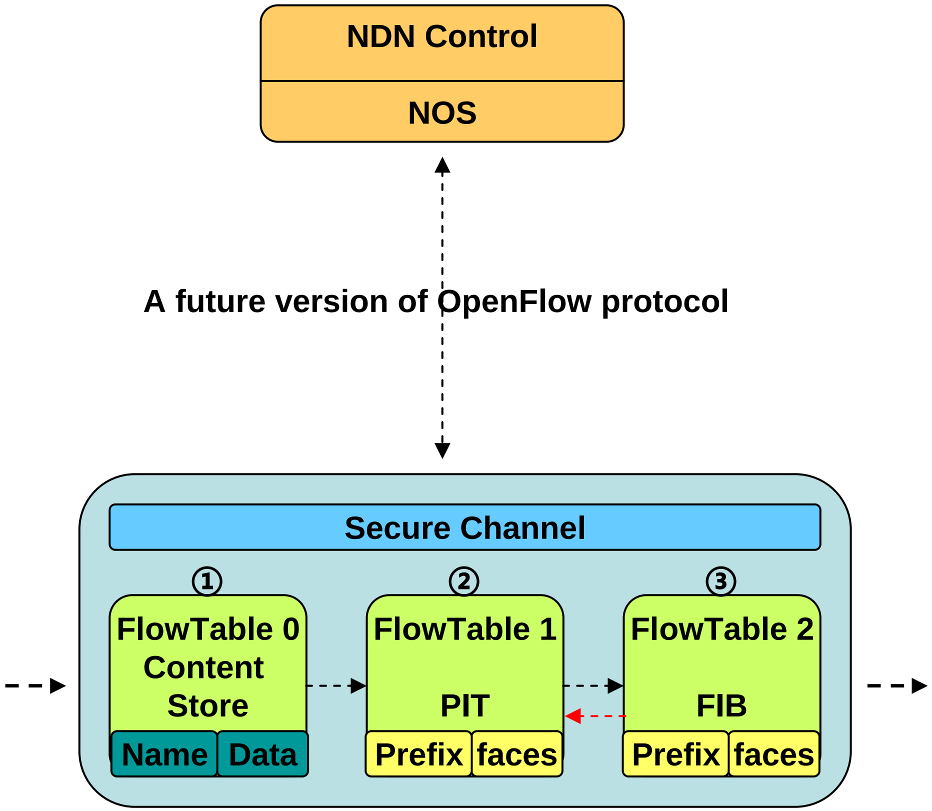
Different store positions, different forwarding abilities

Challenge 1: Post-IP Forwarding (NDN over SDN)

- Forwarding abstraction needs extension (*more autonomic*?)
- Cache update policy
- SDN device for Position1
  - Needs the ability to *generate* new packets of the data
- SDN device for Position2
  - Lookup the data in pass-by storage devices by data index
  - Fetch the data to OpenFlow switch
  - Generate a new packet of the data and send back
- SDN device and controller for Position3
  - OpenFlow switch send a request packet to controller with data index
  - Controller lookups the data by data index
  - Controller generates a new packet of the data and send to the user

Challenge 1: Post-IP Forwarding (NDN over SDN)

Discussion2: How to add new *forwarding actions*
e.g for NDN Interest packet processing (new action types shown in red)



①Interest packet has been matched in FlowTable 0--Content Store. The actions in CS are discussed;

②Otherwise, the packet is sent to FlowTable1- -PIT. If there is a match, the actions in PIT are:
- Add the arrival face to Flow Entry;
- Drop the packet;

③Otherwise, the packet is sent to FlowTable2- -FIB. If there is a match, the actions in FIB are:
- Forward the packet out from a face;
- Add a new Flow Entry in FlowTable1(PIT) with the Interest and forwarding face;

④Otherwise, the packet is
- Dropped;
- Or sent to the controller;

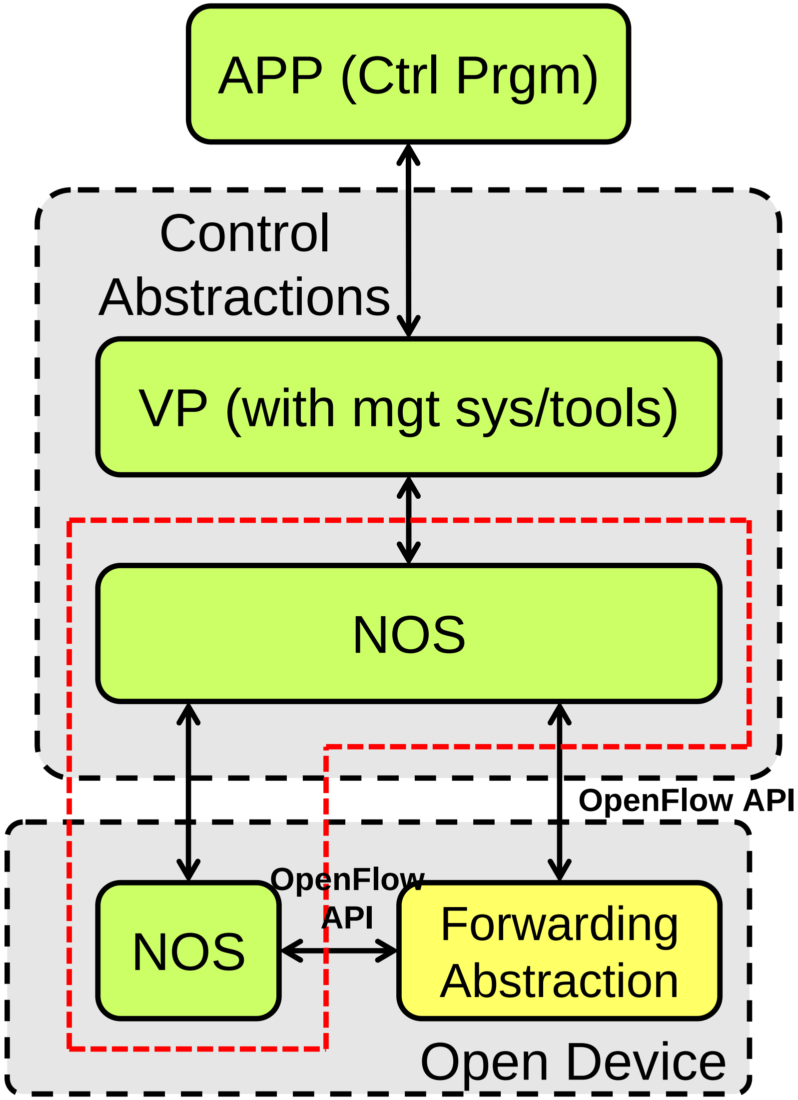
Challenge 1: Post-IP Forwarding (NDN over SDN)

- Summary of forwarding abstraction challenge for Post-IP running over SDN
  - How to define/add new *forwarding components* for Post-IP (e.g. content store in NDN)
  - How to define/add new *forwarding actions and sequence* of Flow Tables for new procedures of Post-IP (e.g. PIT processing in NDN)
  - How to extract forwarding abstractions for arbitrary *Post-IP architectures* co-existing at the same forwarding platform
  - Maybe, allows hybrid forwarding abstraction technologies but managed by the common NOS ?

Challenge 2: Network Operation System

The Combination of Centralized and Distributed Control

- To improve scalability, NOS may run over multiple physical servers
- NOS may also run inside network devices, which is good for performance /robustness of some protocols/architectures
- But for APPs, virtualization platform will provide a single global view (red box) to VP and APPs



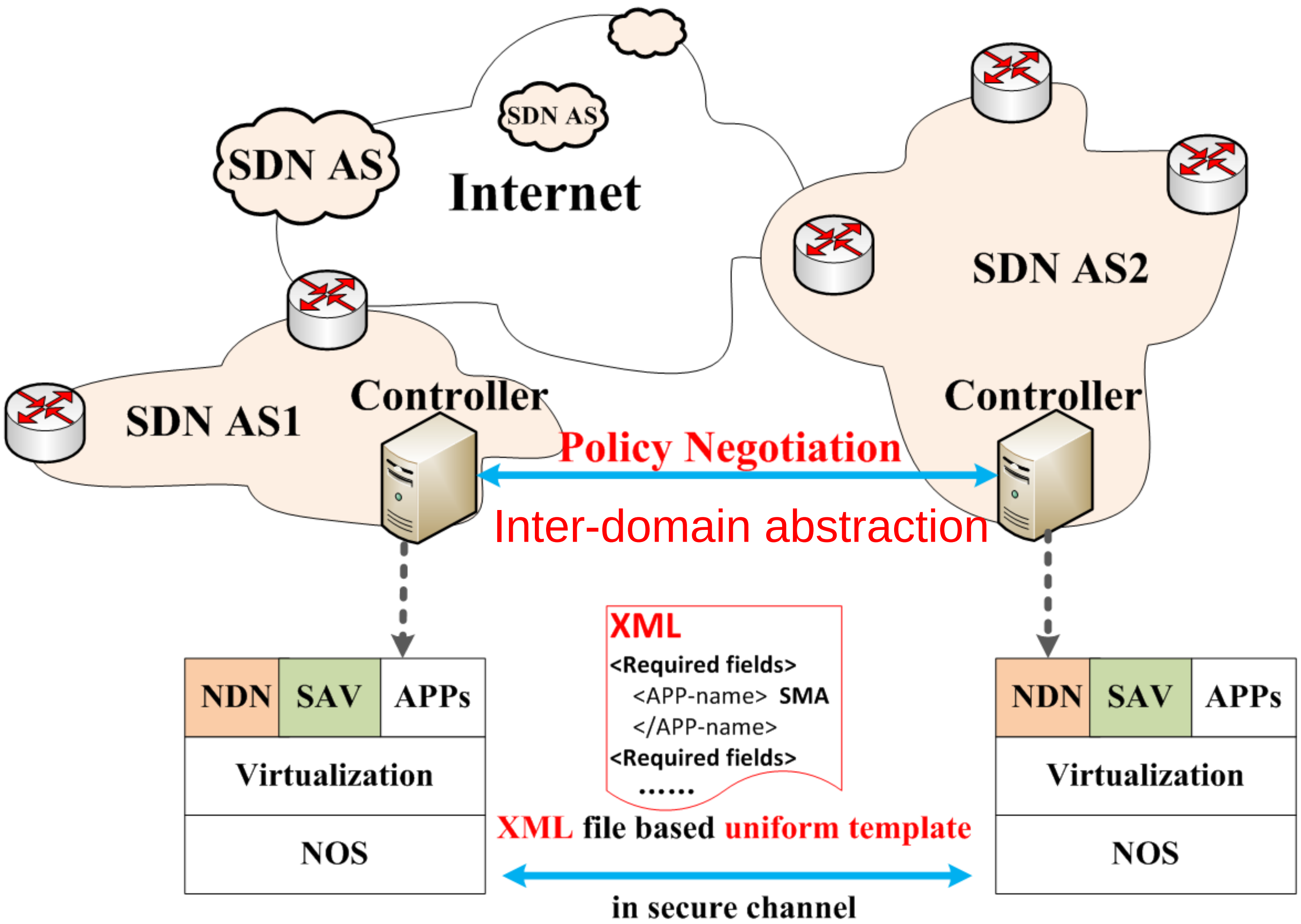
Challenge 2: Network Operation System

- An example of issues is that *large amount of APPs* (forwarding policies) may result in conflict of network control rules if NOS as a common platform
- Possible way to avoid conflict
  - Resource (virtual) isolation (e.g. VLAN-id/APP-id)
  - FCFS based (ACL-like)
  - Priority based (RTM-like)
  - Or more agility way?
  - Need some study

Challenge 3: Virtualization Platform

- Virtualization platform design goal
  - Mapping from logical resources/function requirements (topology, computation, routing, security, etc) to physical resources/functions
  - 1: N or N:1 resources mapping for
  - Devices, links, function elements (routing, security...), resources
- N:1 mapping
  - How to share forwarding resource with less conflicts from different APPs
- 1: N mapping
  - One issue is that APP may has flexibility to select specific resources, e.g. ask for running inside a specific network device
- *Do Need management system and tools !*

Challenge 4: Inter-domain Policies Negotiation within SDN Alliance



Challenge 4: Inter-domain Policy Negotiation within SDN Alliance - Example

- Packet processed by inter-domain negotiated policies, e.g.:
  - IP (routing path),
  - SAV-SMA (signature)...
- Policy negotiation are done by controllers in each AS.
- An design example: APPs use uniform XML template for policies negotiation abstraction
- Three fields in the template: Mandatory, Optional, and User-defined

<Mandatory fields>
<APP-name> **SMA** </APP-name>
<version> 2 </version>
<reachability type> IPv6 </ra type>
<reachability length> 128</ra length>
<reachablity value> 2001:xx<ra value>
……
</Mandatory fields>

<Optional fields>
<signagure len> 64bits </signagure len>
<signagure> xxx </signagure>
……
</Optional fields>

<User-defined fields>
<description> algorithm: KISS-99 64-bit Joint</description>
……
</User-defined fields>

Conclusion

- Inter-AS IPv6 Source address validation using OpenFlow
  - OpenFlow upgradable from legacy routers
  - OpenFlow extension for autonomic functions
  - Good results so far, discussing next step
- SDN design challenges
  - Intra-AS: forwarding abstraction for Post-IP
  - Intra-AS: control abstractions - NOS and virtualization platform
  - Inter-AS: standard inter-domain policies negotiation abstraction

SDN is “FINE”

Thanks!