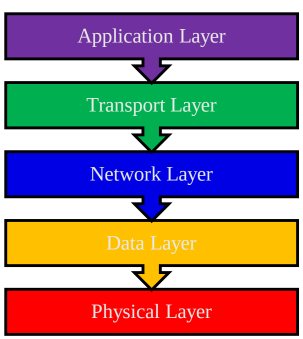
A Crash-Course in Software-Defined Networking (SDN)

Jon Aho & Kailash Joshi

Content

1. Issues with Traditional Networking
2. SDN Architecture
3. Technologies within the Architecture (*OpenFlow*)
4. Applications
5. Traffic Engineering
6. Current Research Topics & Goals

Issues with Traditional Networks



- ***Logically-discrete layers***
- Network behaviors based on hardcoded router policies and internal assumptions about user needs
- Applications have extremely limited ability to control network behavior. Network is essentially "application ambivalent"
- Network does not expose internal diagnostic or state information to applications
- **Simple, but Inefficient**

Issues with Traditional Networks

- *Traditional networks are extremely ossified*
  - **Difficult to perform real world experiments on large scale production networks.**
  - **Research stagnation** - huge costly equipment to be procured and networks to be setup by each team for research
  - **Rate of innovation in networks is slower** as protocols are defined in isolation- lack of high level abstraction.
  - **Inconsistent Policies**
  - **Closed systems**
    - Hard to collaborate meaningfully due to lack of standard open interfaces.
    - Vendors starting to open-up but not meaningfully.
    - Innovation is limited to vendor/vendor partners
    - Huge barriers for new ideas in networking.

What is SDN?

- ***Software Defined Networking*** (**SDN**) is an evolutionary approach to network design based on the ability to programmatically modify the behaviour of network devices.

- SDN is a framework to allow network administrators to automatically and dynamically manage and control a large number of network devices, services, topology, traffic paths, and packet handling (quality of service) policies using high-level languages and APIs.

How does SDN address this?

- **Specification Goal:** "*…provide open interfaces enabling development of software that can control the connectivity provided by a set of network resources and the flow of network traffic though them…*" [1]

- **What does this mean?**
  - **Decouple** the network control from the network forwarding nodes, and **centralize** network intelligence
  - Allow **applications to govern network resources** to maximize efficiency, flexibility, and scalability
  - Make network diagnostics and statistics **accessible**

What does SDN bring to a network?

- **Virtualization**
- **Orchestration**
- **Programmability**
- **Dynamic Scaling**
- **Visibility**
- **Automation**:
  - Troubleshooting
  - Reduce downtime
  - Policy enforcement

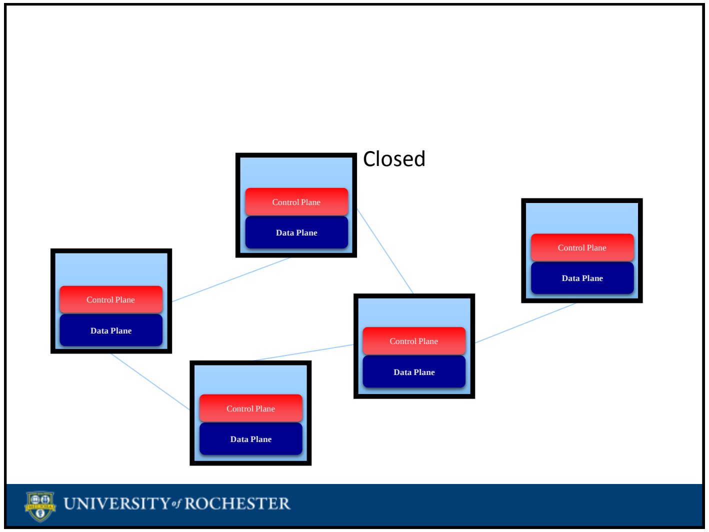
How is this implemented?

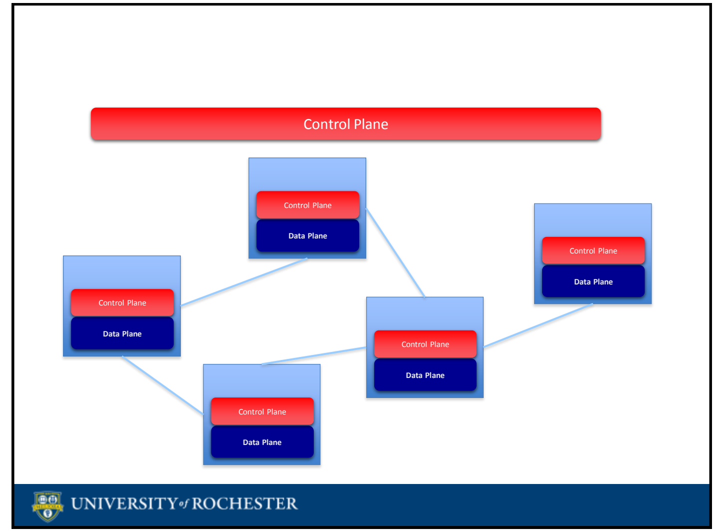
Centralized Intelligence

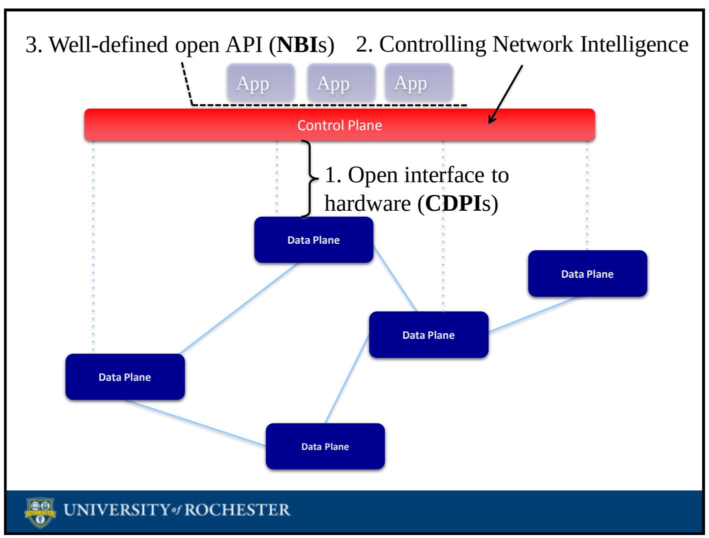
- Create a **logically-centralized** network controller that communicates with both ***applications*** and ***forwarding nodes***, and will be responsible for implementing application needs at the network's composite nodes and reporting information back to the applications
- Essentially a **network operating system**

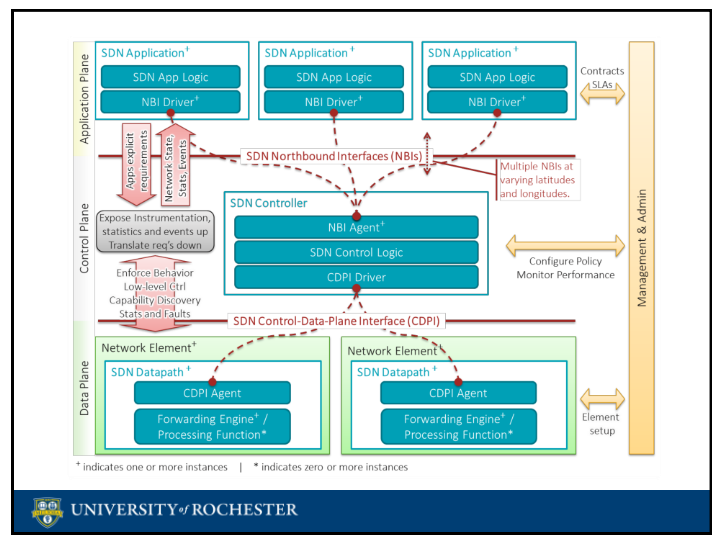
Cross-Planar Communication

- Create interfaces between the ***application***, ***controller***, and ***forwarding*** planes, to allow network control instructions to propagate "down", and state and diagnostic information to propagate "up"
- **North-Bound Interfaces (NBIs)** and **Control-Data Plane Interfaces (CDPIs)**

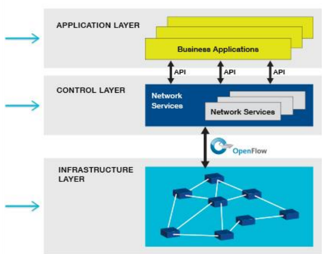

Closed
Control Plane
Data Plane
Control Plane
Data Plane
Control Plane
Data Plane
Control Plane
Data Plane
Control Plane
Data Plane
UNIVERSITY of ROCHESTER


Control Plane
Control Plane
Data Plane
Control Plane
Data Plane
Control Plane
Data Plane
Control Plane
Data Plane
Control Plane
Data Plane
UNIVERSITY of ROCHESTER


3. Well-defined open API (NBIs)
2. Controlling Network Intelligence
App
App
App
Control Plane
1. Open interface to hardware (CDPIs)
Data Plane
Data Plane
Data Plane
Data Plane
Data Plane
UNIVERSITY of ROCHESTER


Application Plane
Control Plane
Data Plane
SDN Application+
SDN App Logic
NBI Driver+
SDN Application+
SDN App Logic
NBI Driver+
SDN Application+
SDN App Logic
NBI Driver+
Contracts SLAs
Apps explicit requirements
Network State, Stats, Events
SDN Northbound Interfaces (NBIs)
Multiple NBIs at varying latitudes and longitudes.
Expose Instrumentation, statistics and events up Translate req's down
SDN Controller
NBI Agent+
SDN Control Logic
CDPI Driver
Configure Policy Monitor Performance
Enforce Behavior Low-level Ctrl Capability Discovery Stats and Faults
SDN Control-Data-Plane Interface (CDPI)
Network Element+
SDN Datapath+
CDPI Agent
Forwarding Engine+ / Processing Function*
Network Element+
SDN Datapath+
CDPI Agent
Forwarding Engine+ / Processing Function*
Element setup
Management & Admin
+ indicates one or more instances | * indicates zero or more instances
UNIVERSITY of ROCHESTER

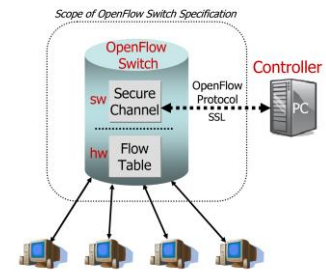
OpenFlow



- First SDN interfacing technology
- Specific protocol for **CDPI** operations: Controller – Data Forwarding nodes
- **Physically implemented** in the network nodes

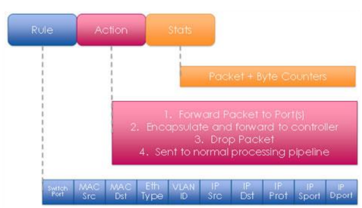
How OpenFlow Works

- Defines ***Flow Tables*** at each network element (switch, router).
  - These tables track message characteristics and tie specified identifiers to specified actions
- Table is defined by messages sent from the controller

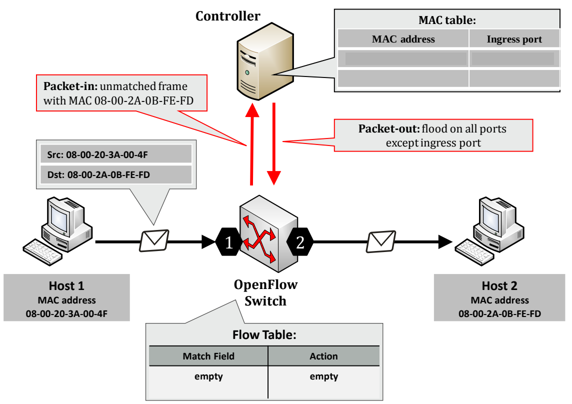


Flow Tables

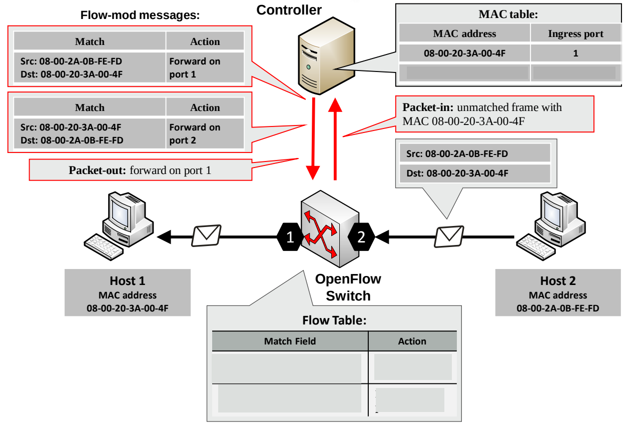
- Composed of three segments:
  - **Rule**: The characteristic of the incoming packet that defines it as this type
  - **Action**: What to do with packets specified by **Rule**
  - **Stats**: Tracking information for this type of packet (generally details history of use)



Communication in OpenFlow Network



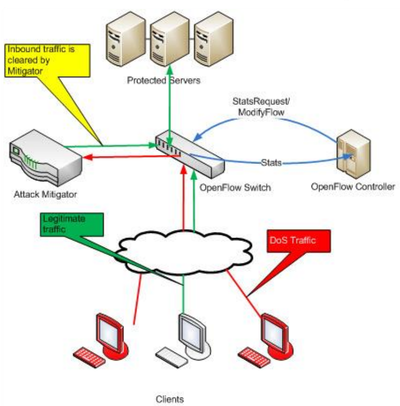
Communication in OpenFlow Network



Applications

- Enterprise Networks
- Data Centres
- Infrastructure-based Wireless Access Networks
- Cellular Networks
- Optical Networks
- Home and Small Business

SDN use case - Security



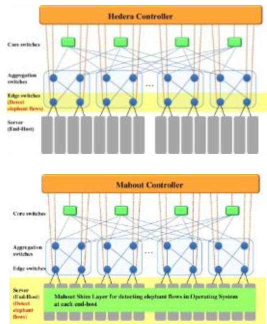
Traffic Engineering (TE)

- **Definition:** *"…optimize the performance of a data network by dynamically analyzing, predicting, and regulating the behavior of the transmitted data."*

- A large body of research exists for TE techniques on older ***ATM*** and ***IP*** networks
- While ***SDN*** shows great promise for advanced TE, research is still in the early stages of determining exactly how to best do so

Flow Management: Switch Load Balancing

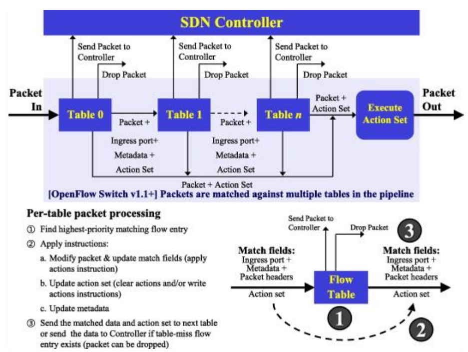
- Hash-based ***ECMP*** (Equal-Cost Multi-Path)
  - Each switch holds multiple equal-cost paths to a given destination
  - A ***hash*** from the *packet's headers* **modulo** *the number of paths* determines which path is used
- Two large, long-lived flows may end up on the same path, creating a bottleneck!
- Proposed solutions:
  - ***Hedera*** & ***Mahout***



Flow Management: Controller Load Balancing

- All new flows must be routed to the controller for processing
  - Huge bottleneck!
  - Not scalable with single controller
- **Four** main controller schemes for solving this:
  1. ***Logically-distributed***
  2. ***Physically-distributed***
  3. ***Hierarchical***
  4. ***Hybrid***
- **Proposed Approaches:**
  - Logically-distributed:
    - ***HyperFlow***
    - ***DIFANE***
  - Physically-distributed:
    - ***Onix***
    - ***BalanceFlow***
  - Hierarchical:
    - ***Kandoo***
  - Hybrid
    - ***SOX/DSOX***

Flow Management: Multiple Flow Tables



Fault Tolerance

- Network must be able to recover from infrastructure failures extremely quickly (< **50 ms**), so as to not affect users
- This is especially difficult for SDNs, which must:
  - Wait for the controller to identify a fault
  - Calculate a new route
  - Update the Flow Tables for each switch along the path.

1. **Fault recovery at data plane:**
   1. ***Restoration*** (Reactive)
   2. ***Protection*** (Proactive)
   - ***Protection*** is more favorable for large-scale SDN networks
2. **Fault recovery at control plane:**
   - **Absolutely critical**
   - ***Primary Backup Restoration***
     - Must coordinate between primary and backup controllers
     - Must actually deploy the backup controllers

Topology Update

- How do we handle packet forwarding when our policies are dynamic?
  - ***Per-packet*** – Each packet will be individually processed
  - ***Per-flow*** – Each flow is guaranteed to be handled by the same version of policy

1. **Duplicate Table Entries**
   - Old policies are stored until all packets originally created during that policy are delivered
2. **Time-Based**
   - The controller delivers new policies with attached scheduled implementation, such that Switch 1 updates at time = t, Switch 2 at time = t + 1, etc, all along the intended route

Traffic Analysis

Tool	***Type***	***Technology***	***Analysis***
PayLess	Query-based monitoring	• Adaptive polling based on variable frequency flow statistics collection algorithm	• Accuracy and overhead dependent on polling interval
OpenTM	Query-based monitoring	• Periodically polling the switch on each active flow for collecting flow-level statistics	• High accuracy and high overhead
FlowSense	Passive push-based monitoring	• Using the PacketIn and FlowRemoved messages in OpenFlow networks to estimate per flow link utilization	• High accuracy and low overhead compared with the Polling method
OpenSketch	Query-based monitoring	• Wildcard rule at switches to monitor aggregate • Hierarchical heavy-hitter algorithm for high accuracy	• Low memory consumption with high accuracy.
MicroTE	Push-based monitoring	• Implemented on separate server • Scalable, low-overhead, proactive	• Low consumed network utilization.
OpenSample	Push-based monitoring	• Use packet-sampling tool sFlow and TCP sequence numbers • Quick detection of elephant flows	• Low latency measurement with high accuracy for both network load and elephant flows.

What to Take Away?

- Traditional networking has a number of significant limitations that slow innovation and prevent intelligent networking
- Software-Defined Networking is a recent system aimed at addressing these limitations by increasing openness, interconnectivity, and programmability
- With SDN, we can achieve greater flexibility, reactivity, and network awareness

Research Areas & Challenges

- **Scalability**:
  - Single controller is not sufficient to manage large scale network.
  - How many controllers are needed to support large scale network?
  - When to scale down?
- **Multi Controllers**:
  - Each controller is responsible to a subset of the network.
  - Concern with synchronization and communication between controllers
  - How to slice the network resources among controllers?
- **Latency between controllers and switches**

Questions?

UNIVERSITY of ROCHESTER

References

1. **A roadmap for traffic engineering in SDN-OpenFlow networks -** http://www.sciencedirect.com/science/article/pii/S1389128614002254
2. **OpenFlow: Enabling Innovation in Campus Networks** – www.openflow.org/documents/openflow-wp-latest.pdf
3. **SDN Networking Overview -** https://www.opennetworking.org/images/stories/downloads/sdn-resources/technical-reports/SDN-architecture-overview-1.0.pdf
4. **Software-Defined Networking: The New Norm for Networks -** https://www.opennetworking.org/images/stories/downloads/sdn-resources/white-papers/wp-sdn-newnorm.pdf
5. **OpenFlow/SDN Tutorial OFC/NFOEC -** http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6476319
6. **Considerations for Software Defined Networking(SDN):Approaches and Use Cases** – http://ieeexplore.ieee.org/iel7/6490096/6496810/06496914.pdf
7. **Open Networking Foundation -** https://www.opennetworking.org
8. **Are Vendors Closing OpenFlow? -** http://gigaom.com/2012/03/19/are-vendors-closing-openflow/
9. **Software-defined networking: Google leads the charge in making the internet faster -** http://www.extremetech.com/internet/140459-networking-is-getting-better-and-thats-partly-thanks-to-google
10. **A Survey of Software-Defined Networking: Past, Present, and Future of Programmable Network -** http://ieeexplore.ieee.org/document/6739370/?tp=&arnumber=6739370
11. **OPEN DATA CENTER ALLIANCE Master USAGE MODEL: Software-Defined Networking Rev. 1.0 -** https://www.opendatacenteralliance.org//docs/Software_Defined_Networking_Master_Usage_Model_Rev1.0.pdf
12. **Demonstrations – Open Networking Summit, April 2012 -** http://opennetsummit.org/archives/apr12/site/demonstrations.html

UNIVERSITY of ROCHESTER