Software-Defined Networking
Applications in Network Science and Engineering

Deniz Gurkan, PhD
February 21, 2014

Networking Lab: http://sites.tech.uh.edu/networking-lab/
dgurkan@uh.edu

Funded by Dell, Infoblox, vArmour Networks, NSF.
Active collaborations with Juniper, Cavium Networks and Intel.
Separation of control and data planes = SDN

Innovation opportunities unleashed

NOW

- Network Management
- Feature_1
- Feature_2
- Feature_3
- Controller_X
- Switch_X
- Controller_Y
- Switch_Y

NEAR FUTURE

- Network Management
- Application
- Application
- Application
- Controller
- E.g., OpenFlow
- Protocol Agent
- Protocol Agent
- Protocol Agent
- Switch
- Switch
- Switch
Research Directions

1. Network management
2. New network abstractions with capability-based nodes
3. Network functions virtualization and distribution
   1. On-demand network programmability
   2. Traffic steering
4. Switching/forwarding as a software construct
5. Future directions
Motivation

Deep programmability of management, control, and data plane

E.g., OpenFlow
Network Management

- Control plane: program forwarding elements
- Data plane: forward data packets/flows
- Management plane: discover/monitor/manage resources

**Research on:**
1. Managed object models
2. Management primitives
3. Share control/management
OF-Config Protocol: Visualization of the Management Plane in OpenFlow/SDN Networks

Anatoliy Malishevskiy and Deniz Gurkan

- Implemented on ProtoGENI
- OF-Config: Network Management for OpenFlow networks
- DEMO: Displaying LINC Switch configurations through NETCONF using OF-Config specification
- OF-Config Specification uses NetConf as transport
- YangCli (YUMA) mechanism is used to get switch configurations
- GREE 2013 work in progress paper

The data model for OF-Configurations Protocol is structured into classes and attributes of classes. Each class is described in a separate sub-section by XML, UML, and YANG. The core of the model is an OF Capable Switch that is configured by OF Configuration Points. Instances of OF logical switches are contained within the OF Capable Switch. A set of OF Controllers is assigned to each OF logical switch. When issuing a NETCONF “get-request” all elements in the requested sub-class or sub-tree must be returned in the result and then manipulation can be done.
Outline

1. Network management
2. New network abstractions with capability-based nodes
3. Network functions virtualization and distribution
   1. On-demand network programmability
   2. Traffic steering
4. Switching/forwarding as a software construct
5. Future directions
Motivation

Leverage hardware capabilities

Network Management

Application

Application

Application

Controller

Protocol Agent

Switch

Protocol Agent

Switch

E.g., OpenFlow

Ctrl

App

Protocol Agent

Switch
Application-Network Interfaces

• Network abstraction for application development
  • Socket API: connect/send/receive/listen
  • ?
• Manipulate flows on their way from source to destination
  • Not at end points
  • Leverage hardware: Acceleration, buffering, storage, DPI, etc.

**Research on:**
1. Taxonomy of functions
2. Control/routing with sub-units
Split Data Plane SDN: Development of Applications for Network Programming at the Network Nodes

Rajesh Narayanan (Dell), Fahd Gilani (XFlow Research), Praveen Mala (UoH) and Deniz Gurkan (UoH)

Network Programmability

**Definition:** Configure, provision, and specialize an underlying network according to the service application and user needs

**Current:** program end point behavior, take network as quasi-rigid setup

**OpenFlow/SDN:** dynamically control flow definitions by a controller

**Split Data Plane (SDP) SDN:** dynamically control and change flows at network nodes while high-level flow definitions are still governed by the controller

**GOAL**

Creation of applications that leverage SDN technologies and OpenFlow protocol to deliver network processor capabilities into network programmability.

**DEMO SETUP**

An emergency application sends secure-sensitive announcement over the network: network-level encryption and decryption is achieved.

**Emergency Management Portal:** Encrypt network data when emergency communication in effect

Flow setup forwards emergency messages to the SDP for encryption

Special thanks to Michael Blair Wever from Cavium

Diagram:
- OF Macro-flows on TCAMs
- OF Micro-flows on special processors
- XAU Interface
- Encryption
- Compression
- Deep Packet Inspection
- Did not fit into TCAMs
- Exotic flow definitions
- Split Data Plane SDN
- OpenFlow Switch
- Non-SDP Switch/Router
- Emergency (secure-sensitive) announcement SOURCE
- Emergency (secure-sensitive) announcement DESTINATION
**What is SDN Innovation Platform?**

- OF Macro-flows on TCAMs
- OF Micro-flows on special processors

**Developing Applications**

- As easy as socket programming – *not at end points but on network nodes*
- Popular programming languages
- Hands-on experimentation
- *Sky is the limit*

**Deploying on GENI**

- Booting capabilities are similar to GENI
- Better way to generate applications and experiments on the physical network nodes
- Network visualization
- Opening up new research fields

**ICMP-Proxy**

- Controller Apps
- Flow State/Counters/Data Store
- Node Monitor
- PostHilo World
- PostHilo World
- PostHilo World
- PostHilo World
ICMP-Proxy

Controller Apps

Controller

Switch

SDP Apps

Flow Stats/Counters Data Store

Post1Hello World
Post2Hello World
Post3Hello World
SDP-Network Processing Unit (NPU) on ExoGENI

Levent Dane, Deniz Gurkan (UoH)

ExoGENI Integration

ExoGENI Rack

- Head Node
- Bare Metal Node
- Bare Metal Node
- NPU
- Dell SW
- Compute Nodes
- Compute Nodes
- Compute Nodes

USB Serial
- Ethernet

IBM G8052

Current Status
- Physically in between two bare metal nodes
- Contact UH for application deployment

Future Work
- Dynamic Authorization
- ORCA integration
- Dynamic VLAN pass through
Outline

1. Network management
2. New network abstractions with capability-based nodes
3. Network functions virtualization and distribution
   1. On-demand network programmability
   2. Traffic steering
4. Switching/forwarding as a software construct
5. Future directions
Motivation

Specialized controller/app coupling

Network Management

Application
Application

Controller

Controller

Protocol Agent

E.g., OpenFlow

Protocol Agent

Switch

Switch

Switch
On-demand Network Programmability

- Application-triggered network connectivity and path setup
- Emergency first responder assistance
  - Bandwidth
  - Priority
  - Best path (or all)

Research on:
1. Network data model
2. Application-aware control
3. Centralized control policy
Emergency Management: Interoperable Data Exchange and Dynamic Network Setup

University of Houston: Huy Vo, Deniz Gurkan, Anand Arun Daga
Infoblox: Sandhya Narayan, Stuart Bailey

Enter all data into SharePoint
Communicate with the entities involved
Make a decision on public safety
Communicate with the university administration
Communicate with university community

EMERGENCY OFFICER
View emergency management portal, publish police dispatch emergency events on the MAP server
Update/Publish other information on events and buildings related to emergency on the MAP server
Delete emergency events when necessary from the MAP server

PUBLIC SAFETY OFFICER
Subscribe to emergency events and related information feeds on the MAP server:
e.g., view video camera feeds around an emergency site as soon as there is a police dispatch of an event at the site

OPENFLOW CONTROLLER
Subscribe and/or search events to program and provision the network accordingly
Push flows to the OpenFlow switches to assign priority or pick best paths

DEMO SETUP
1 Emergency event is published to a building site
2 Controller is subscribed to events at flow-controlled sites
3 Controller pushes flow definitions for high quality video transmission
4 Public safety officers view video feeds at high quality
Network Functions Virtualization

- Application-triggered “network function call”
- Firewall rule offload to forwarding plane
  1. DPI the flow → identify
  2. If safe → offload to network as flow rule
  3. Track only state of session

Research on:
1. Time/energy savings
2. Network abstraction
3. Network measurements
Service Chaining

Enterprise Traffic + Scientific Data firewalled!

OpenFlow Switch

Flow rules to pass traffic through the firewall

DPI check for all incoming traffic

Scientific Data Transfer Initiated!

OpenFlow Controller

Project Sponsored by vARMOUR
Service Chaining cont.

ScienceDMZ application fast path configured!

Firewall Session Info Received!

Application Identified through DPI

Flow rules to pass traffic through the firewall

Scientific Data Transfer Initiated!

OpenFlow Switch

Project Sponsored by vARMOUR
Service Chaining cont.

If \{A \in U; \text{ Push Static Flows for fast path}\}

Offloading Scientific Data Transfer!

OpenFlow Controller

Firewall Session Info Received!

Fast path for ScienceDMZ app’s configured!

Flow Mod’s for Fast Path

OpenFlow Switch

Application Identified through DPI

New fast path flow rules for ScienceDMZ

Flow rules to pass traffic through the firewall

Scientific Data Transfer Initiated!

Project Sponsored by vARMOUR NET WORKS
Measurement Points

1. Firewall to AppServer N/W Delay
2. AppServer Processing Time
3. Application Steering Server to Controller N/W Delay
4. Controller Processing Time
5. Controller to OF Switch N/W Delay

Project Sponsored by vARMour
Offload - Justification?

Diagram:

- **XEN virtualization platform host**
  - **$OVS_c$** – traffic steering control
    - **AppServer**
    - **Firewall**
      - Arrival of last data packet \( @T_3 \)
      - Arrival of first data packet bypassed the firewall \( @T_4 \)
    - Arrival of flow_mod \( @T_2 \)
  - **$OVS_d$** – data plane
    - **Floodlight**
      - Arrival of a static flow push \( @T_1 \)
      - Flow_mod Sent Out \( @T_{1a} \)
  - **Client**
  - **Server**
Better measurement scenario
On GENI
Outline

1. Network management
2. New network abstractions with capability-based nodes
3. Network functions virtualization and distribution
   1. On-demand network programmability
   2. Traffic steering
4. Switching/forwarding as a software construct
5. Future directions
Motivation

Switching as a *software construct* for applications

![Diagram showing network management, application, controller, switch, protocol agent, and OpenFlow](image)
Switch as a Software Construct

• Server industry ~--- switch industry
  → approaching an understanding of the forwarding elements as a “software construct” rather than a “vendor box”

Research on:
1. Network abstraction
2. Programmable header parsing
3. Fault-tolerant switch fabric
Packet loss during failover and takeover on Distributed OpenFlow Switch Architecture

Gandhimathi Velusamy and Dr. Deniz Gurkan

In collaboration with Dr. Sandhya Narayanan

![Diagram showing the failover process between hosts and controllers.](image-url)

- S1 failed, S2 takes over
- S2 failed, S3 takes over
- Take over by S1 when S3 is running

Lost Pings Normalized with Avg.RTT during Failover and Takeover

![Graph showing the normalized lost pings with average RTT during failover and takeover.](image-url)
Thank you

dgurkan@uh.edu