IP Transport, Routing and Control and the Software-Defined Networking era

The RouteFlow project
Agenda

• Introduction to OpenFlow/SDN
• Selected hot topics in IP networking
  • Multipath-TCP, ALTO, IETF RRG
• RouteFlow
  • Motivation
  • Architecture
  • Use cases
  • The Path ahead
Innovation and Ossification in the Internet

Source: Peter Stuckmann and Rainer Zimmermann, "European Research on Future Internet Design", IEEE Wireless Communications Magazine, October 2009
Tussles in Networking

Applications

Network
Traditional Tussles in Networking

Applications vs. Network
Software vs. Hardware
Centralized vs. Distributed
Tussles in Networking

Applications deriving network
• BitTorrent, Skype, etc.

Networks spying on Applications
• DPI, Netflow, IDS, etc.
Applications & Network interaction

**Intuiting Info**
- Application deriving network
  - Proprietary control channels, pingstats, GSM, check-ins, traceroute
- Network spying on application
  - DPI, Netflow, IPDR, IDS

**Communicating**
- Application programming network
  - OpenStack
  - CloudStack
  - OpenFlow
  - PCE
  - GenApp
  - Provider-based development platforms
- Network informing application
  - ALTO
  - BGP-TE, BGP-CDNI

Source: D. Ward, Juniper
ALTO (Application Layer Traffic Optimization)

• IETF ALTO Generalized Goal:
  • Provide applications with information to perform better-than random server selection
  • Key questions:
    • What information is provided to applications? E.g., cost-based rank of server IPs
    • And based on which network provider information? E.g., Policies, topology, congest. that determine server selection criteria (out of standard´s scope)

• Goals for CDN and Cloud networking
  • Optimize CDN traffic delivery
  • Improve global and local server load balancing
  • Facilitate topology & status information sharing between domains (e.g., CDN / Cloud – ISP)
ALTO

Application Layer

CDN

IPTV Servers / Caches

OTT Overlay

Cloud / *aaS

ALTO

ALTO Server:
- Information Collector
- Algorithms
- Databases

Policy Database

Routing / Control Plane Protocols Databases

ISIS, OSPF and BGP

Geo-location information

State and performance information

IETF ALTO

Standard Protocol (API: Request / Reply Model)

Layer Separation – No topology information leaking
IETF Multipath TCP (MPTCP)

- **Multipath transport**
  - Allows multiple links to be treated as a single pooled resource.
  - Traffic moves away from congested links.
  - Larger bursts can be accommodated.

- **Fairness and Resource Pooling**
- **Improved Resilience**
- **No modification to applications**

Source: Mark Handley, UCL
OpenFlow: Towards a clean API to hardware
OpenFlow and Software-Defined Networks

Source: Martin Casado, Nicira
Driving forces: Convergence / Programmability / SDN

* Picture from NEC
An open-source project to provide IP routing & forwarding services in OpenFlow networks

http://go.cpqd.com.br/routeflow

- Unicamp
- Unirio
- Ufscar
- Ufes
- ....

- Indiana University
- Stanford
- Google
- Deutsche Telekom
- ....
RouteFlow Project Timeline

Jan / 2010

- Start MSc. Thesis by Marcelo Nascimento
  - First Prototype
  - First Short-Paper @ WPEIF
  - QuagFlow Poster @ SIGCOMM

Aug / 2010

- Open-Source Release
  - Evaluation on NetFPGA testbed

Dez / 2010

- Demos @ ONS11
  - Tutorial & Demo @ OFELIA/CHANGE SS
  - Indiana University - Pronto OF switches + BGP peering with Juniper MX

May / 2011

- Nation-wide field trials (Brazil, Internet2) @ ONS12
  - Demo @ SuperComputing 11
  - RouteFlow + OpenStack (Cloud RouteFlow)

Oct / 2011

Nov / 2011

... / 2012
... building a community

1,390 visits came from 333 cities

http://go.cpqd.com.br/routeflow/
Motivation v1

Original motivation around RouteFlow (formerly QuagFlow) (Seeded in experience building a Broadcom-based L2/L3 switch prototype)

• Current “mainframe” model of networking equipment:
  • Costly systems based on proprietary HW and closed SW;
  • Lack of programmability limits cutting-edge innovation and in-house innovation;
  • Ossified architectures.

• Goal: Open commodity routing solutions:
  + open-source routing protocol stacks (e.g. Quagga)
  + commercial networking HW with open API (i.e. OpenFlow)
  = line-rate performance, cost-efficiency, and flexibility!
Current router architectures

- **Management**
  - Telnet, SSH, Web, SNMP

- **Control Logic**
  - RIP
  - BGP
  - OSPF
  - ISIS

- **Driver**
  - O.S.

- **Proprietary IPC / API**

- **Hardware**
OpenFlow model

Management

Control Logic

Controller

RIP
BGP
OSPF

O.S.

API

Driver

Switch

O.S.

API

Hardware

Standard API (i.e. OpenFlow)
Motivation v2

• A transition path, incrementally deployable: from current IP networks to SDN
  • Hybrid modes of operations: traditional IP control planes along SDN

• Innovation around IP control planes
  • Simplified network mgm, protocol optimization, shadow networks

• Advancing IP Network Virtualization
  • From flexible Virtual Routers to IP Network-as-a-Service
Use Cases

- **Logical Split Router (1:1)**
  - Virtual Network Provider (Network Slices)
  - Infrastructure Provider (Physical Substrate)

- **Router Multiplexation (1:n)**

- **Router Aggregation (m:1 or m:n)**
Architecture

Architectural Features

• Separation of data and control planes;
• Loosely coupled architecture:
  • Three RF components: 1. Controller, 2. Server, 3. Slave(s)
• Unmodified routing protocol stacks;
  • Routing protocol messages can be sent 'down' or kept in the virtual environment;
• Portable to multiple controllers:
  - RF-Controller acts as a “proxy” app.
• Multi-virtualization technologies
• Multi-vendor data plane hardware
Demos @ ONS´11, SC´11

Indiana University

w/ commercial switches from IBM, NEC, Pronto
The Path Ahead

• OpenFlow 1.2, 1.X ... 2.0
• Controller API: Rest-API JSON & Apache Thrift
• Advancing the IP Network Virtualization
  – Protocol Optimization, Modes of operation, Router Migration
• Scalability and Resiliency
• System Limits and Stress testing
• Live Trials
  – Reality-Checks at Scale
• Embrace related work (past & ongoing)
  – RCP, SoftRouter, VROOM, DROP, FIBIUM, ONIX, etc.
• Build a community!
  – Student Projects corner (https://sites.google.com/site/routeflow/projects)
Protocol Optimization

- Separation of concerns between topology maintenance and routing state distribution
  - E.g. HELLOs sent “down” while LSA are kept “up”
  - E.g. BFD-like fault detection substitute HELLOs
Convergence in SDN Network

- Link detection
- Signal controller
- Path compute/lookup
- Push updates to network

OpenFlow Network
Resiliency and Scalability

- Distributed Virtual environment with distributed OVS for load balancing, replication, and advanced VM management (e.g., migration)
- NoSQL-like distributed database for core RouteFlow state
- Multi-controller environments
- Fault-tolerance: Master / Slave, Master / Master, ...?
System limits and Stress testing

- Increase network size
  - Increase flowmod/sec
    - Variable OpenFlow control packet handling / processing:
      Impact on Routing Protocol?
      Impact on topology maintenance protocol, e.g., LLDP-based?

- Scale limitation (Flow table size) of logical / large routing tables
  - Smart shared multiple table lookup in OF.1.1
  - Smart caching, hybrid software-hardware flow state
  - Related Work (e.g., ViAggre)
  - etc.
Advancing the IP Network Virtualization

- Variable mapping of physical and logical entities
  - Slice network hardware for multiple customers
  - Handle multiple network devices using a single control plane

- Live-migration of logical routers
  - Load balancing (capacity, routing tables, CPU)
  - Scheduled router maintenance
  - Energy conservation
Aggregated Router

• Scenarios:
  • BGP instances aggregating a number of OpenFlow switches
  • L3 services in multi-tenant data center distributed single virtual switch
Enabling Virtual networks as a Service

• Many open research questions and ongoing work
  • e.g., Quantum @ OpenStack
• CloudRouteFlow as a Service?
Enabling Virtual networks as a Service
**CPqD Dynamic Converged (Packet and Circuits) Services**

**Goal:** Common control plane for Layers 1 to 3 networks aiming at NaaS, RaaS, VNO

**Approach:** OpenFlow + RouteFlow + SPIDER (virtualization comes in a subsequent phase)
OpenFlow R&D activities

**OpenFlow Switch** – first in South America
- 24 x 10/100/1000
- 2 x 10Gb
- L2/L3
- ~1000 flow entries
- No protocol stack

**OpenFlow ROADM** – first in South America
- WSS for mesh networks
- 3 / 5 degree
- Directioned / Coloured
- Virtualization

**Towards OpenFlow 1.2** – collaboration with Ericsson
- IPv6 support via extensible flow matching (NXM)
- Software switch prototype
Conclusions

- RouteFlow proposes a commodity routing architecture that combines the line-rate performance of commercial hardware with the flexibility of open-source routing stacks (remotely) running on PCs;

- Allows for a flexible resource association between IP routing protocols and a programmable network substrate:
  - Multiple use cases around virtualized IP routing services.
  - IP routing protocol optimization
  - Migration path from traditional IP deployments to software-defined networks
"SDN has won the war of words, the real battle over customer adoption is just beginning...."  
- Scott Shenker
Christian Esteve Rothenberg, PhD
esteve@cpqd.com.br
+55 19 3705-4479
RF-Controller application

- Shim application on an OpenFlow controller
- Mainly acts like a proxy for the OpenFlow API
- Interacts with the OpenFlow datapaths
- Filters relevant events to the RF-Server
- Receives flow mod commands
- Delivers traffic to/from VM interfaces via OVS
RouteFlow Server

• The “brain” of RouteFlow;
• Manages available virtual machines (VM);
• Configures the virtual environment
• Receives events from the RF- controller
  – Switch join/leave, packet-in;
• Associates VMs and OpenFlow switches;
• Determines packet delivery from/to VMs
• Requests flow installation / modification in OpenFlow switches.
RouteFlow-Slave

- Runs as a daemon in Linux-based VM
- Registers the VM with the RF-Server
- Configures the VM (e.g., interfaces)
- Listens to ARP and IP table updates via Linux Netlink events
  - Linux Routing stack independent (Quagga, XORP)
- Translates routing updates into flow rules;
  - Match: DST_MAC + DST_IP + MASK
  - Actions: Re-write MACs + port-out
- Translates ARP entries into flow rules
  - Match: DST_MAC + DST_IP
  - Actions: Re-write MACs + port-out
- Sends flow update commands to RF-Server
- Runs VM-OVS attachment discovery protocol
Virtual Environment

- V1 used TUN/TAP devices and payload encapsulation in the RF-Protocol
- V2 manages VM connectivity through an OpenFlow-capable soft-switch
- Routing engines (e.g. Quagga) exchange routing protocol packets
  - Two modes of operation for VM packet exchange:
    - UP: Directly through the OVS (requires Topology Disc)
    - DOWN: Through the physical switches
- Centralized but logically distributed
  - Can be physically distributed
- Support of different virtualization technologies
  - From QEMU to LXC
- VM-OVS Attachment Discovery Protocol
IP Forwarding Rules in OpenFlow

RF-Slave info from the Linux network stack
- Route = IP + MASK [Rede] + IP[Gateway] + Interface
- ARP = IP[Host] + MAC[Host] + Interface

OpenFlow 1.0 entry:
- Match: DST_MAC + DST_IP + SUBNET_MASK
- Actions:
  - Re-Write [SRC_MAC (Interface)], Re-Write [DST_MAC (NextHop)]
  - Forward [Port-out(Interface)]

Longest Prefix Match (LPM)
- Add priority to flow entry based on the length of the subnet mask

In OpenFlow 1.1:
- Additional actions: TTL decrement, checksum update
- Multiple-Table: Table[0] Matches DST_MAC, Table[1] Matches DST_IP
SRS Demo @ SC11

SCinet Research Sandbox Testbed (SC11)

NOC
DNOCs

Quagga
OSPF

Flowvisor
OpenFlow
NOX
RF-Server
Main goals of CPqD

- Carry out R&D and bridge the gap between university research and product technology to innovate and (mainly, but not necessarily) help local companies

- Support government in strategic technology decisions in telecommunications and information technology
(Typical) Model of Operation

Government funds
Own funds

Government funds
Own funds

Also work as independent contractor for national and international companies

Licensing of technology to partners
Direct provider of services in the market
Future Internet R&D

• Innovation in the Internet architecture, technology and services to help (mainly, but not exclusively) Brazilian companies and society

• Evolutionary and Revolutionary

• Forwarding plane and Control plane

• Network Operating System

• Services and applications