Software Defined Networking

Olaf Hagemann
SE Director for DACH and HGM
ohagemann@extremenetworks.com
SDN Drivers

- User, Device, App Mobility
- Cloud-Sourcing
- Consumerization of IT
- Silo’d Workflows

- Dynamism
- Scale
- Complexity
- Vendor Dependence
- Cost
Remember the Mainframe?

- Specialized Applications
- Specialized Operating System
- Specialized Hardware

- Vertically integrated
- Closed, proprietary
- Slow innovation
- Controlled industry

Programmable Open Interface

- Open interfaces
- Rapid innovation
- Broad industry Participation

Open Interface

- Windows (OS)
- Linux
- Mac OS

Commercial Off The Shelf Microprocessor
What is SDN: Networking

- **Specialized Features**
  - Tight Vertical integration
  - Closed, proprietary
  - Slow innovation
  - Tightly controlled

- **Specialized Control Plane**

- **Specialized Hardware**

- **Open Interface**
  - De-coupling of Layers
  - Open interfaces
  - Rapid innovation
  - Broad Industry Opportunity

**Control Plane**

**Merchant Switching Chips**
Software Defined Networking (SDN) Model

- Make Control and Management Plane Programmable
- Centralize Network Intelligence
- Abstract Network Infrastructure for Applications
- Separate Control Plane from Data Plane
SDN Economics: Applications Automation & Simplicity

SDN Enables Comprehensive Seamless Networking

Legacy Network Services

Ethernet-based SDN

- Applications Drive Productivity
- More System Interoperability
- More Complete Automation
- Less Dependence on CLI
- Fewer “Experts” Required

Network Operational Revolution

Application-Based Networking
Drives Lower TCO
High Level Architecture

Adaptive Routing

Quality of Service

Traffic Engineering

Metering & Billing

Custom Applications

Centralized Control and Management

Open Programmable OS

Single Consolidated Network based on High Capacity Open Fabric

© 2011 Extreme Networks, Inc. All rights reserved.
Today – Closed Boxes, Fully Distributed Protocols
Another Approach – Centralized Control

Network Operating System

Specialized Packet Forwarding Hardware

Operating System

Open

App

App

App
The “Software-defined Network”

1. Open interface to hardware

2. At least one good operating system
   Extensible, possibly open-source

3. Well-defined open API
Virtualization or “Slicing” Layer

Isolated “slices”

Many operating systems, or Many versions

Open interface to hardware

Simple Packet Forwarding Hardware

Simple Packet Forwarding Hardware

Simple Packet Forwarding Hardware

Simple Packet Forwarding Hardware

Network Operating System 1

Network Operating System 2

Network Operating System 3

Network Operating System 4

App

App

App

App
Software Defined Network

Well Defined Open API

Open Interface to Hardware (e.g. OpenFlow)

APP

SDN Controller

APP

APP

APP
Extreme Networks SDN Approach

SDN Apps
- VM Lifecycle Management (XNV)
- User Identity Management
- BYOD
- Application Performance Management
- ...

Management Platforms
- Ridgeline
- OpenFlow
- OpenStack Quantum Plugin

XOS – Extensible, Open Secure
- XML
- Scripts
- External App SDK
- OpenStack Quantum Plugin
- OpenFlow
- Modular
- Predictable Performance
- Memory Protected
- Hardware Abstracted

High Performance Converged Open Fabric
- Low Latency
- High Capacity
- MLAG
- DCB
Programmability: New Open Interfaces For SDN

- Support for OpenFlow across all Ethernet Switch products
- Powered by ExtremeXOS
- Will be available as a dynamically downloadable module
- Hybrid mode of operation

**Extending SDN Strategy with OpenFlow**

**Open Multi-Vendor Approach to SDN**

- Supporting multiple OpenFlow controllers

**Extending SDN Strategy with Openstack**

- Support for OpenStack Quantum with Plug-In

**Creating SDN Developer Community Portal with XKit**

- Launching Xkit for collaborative
- Programmable Network Applications
What is OpenFlow?

OpenFlow is a protocol that allows an external node to control the forwarding tables of a switch.
What is OpenFlow?

• Started 2008 at Stanford University
• Is now driven by Open Networking Foundation (ONF)
• ONF has over 70 members including:
  – Extreme Networks
  – Broadcom
  – Google
  – Microsoft
  – Yahoo
  – Facebook
  – Deutsche Telekom
  – …..
What is OpenFlow?

- The OpenFlow protocol defines an API to the forwarding plane of a network device.
- A central OpenFlow controller can use this API to manipulate the forwarding on the device data plane.
- The OpenFlow controller virtualizes the network (network hypervisor).
What is OpenFlow?

• Definitions:
  – **OpenFlow-only switch**: supports the mandatory client features of the OpenFlow protocol.
  – **OpenFlow-capable switch**: supports the mandatory client features of OpenFlow in addition to normal switch functions (e.g., STP, EAPS).
  – **Controller**: supports the server feature of the OpenFlow protocol. Manages the forwarding behavior of one or more OpenFlow switches.
  – **Flow entry**: the basic unit of forwarding management.
  – **Flow table**: consists of a set of priority ordered flow entries. A switch may support more than one flow table.
  – **Emergency flow table**: flow table that the switch uses in case connectivity with all configured controllers fails.

• A switch may simultaneously connect to multiple Controllers, but only one controller is master.
  – **Flowvisor**: can be used to slice a switch into multiple logical switches, each managed by a different Controller.
OpenFlow Flow Table Entry

<table>
<thead>
<tr>
<th>Rule</th>
<th>Action</th>
<th>Stats</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1. Forward packet to port(s)</td>
<td>Packet + byte counters</td>
</tr>
<tr>
<td></td>
<td>2. Encapsulate and forward to controller</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3. Drop packet</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4. Send to normal processing pipeline</td>
<td></td>
</tr>
</tbody>
</table>

Ingress Port | VLAN ID & Priority | Ethernet | IP | TCP |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>SA</td>
<td>SA</td>
<td>SRC</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DA</td>
<td>DA</td>
<td>DST</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Type</td>
<td>Proto</td>
<td></td>
</tr>
</tbody>
</table>
Flow Entry

- An OpenFlow flow entry consists of the following components/properties:
  - Header fields
    - the set of header match conditions used to match a packet. Each header field can either be fully matched or wildcarded. IPv4 source and destination addresses can also be subnet masked.
  - Priority
    - the relative priority amongst the set of flow entries in a flow table. Flow entries with no wildcarded header fields have implicit highest priority. The highest priority matching flow entry is selected as the matching flow entry.
  - Counters
    - matching byte/packet counters; duration (nanosecond).
  - Actions
    - forwarding operations to perform on a packet:
      - Forward, Drop, Enqueue, Modify Field
Flow Entry Actions

• Forward: forward packet to the following physical or virtual ports:
  – Output port: forward to a specified physical output port
  – ALL: forward to all ports, excluding the incoming port
  – CONTROLLER: encapsulate packet (Packet-In message) and send it to the controller
  – LOCAL: forward via the switch’s local networking stack
  – TABLE: perform actions in the flow table (for Packet-Out messages only)
  – IN_PORT: forward the packet back out of the incoming port
  – NORMAL (optional): forward according to the traditional forwarding path (e.g., generated by STP + learning)
  – FLOOD (optional): flood the packet along the minimum spanning tree, excluding the incoming port
• Drop: drop the packet
• Enqueue (optional): enqueue the packet in a specified output port queue.
• Modify-Field (optional): modify a header field
Flow Entry Modify Actions

- Set VLAN ID
- Set VLAN Priority
- Strip VLAN header
- Modify Ethernet MAC SA
- Modify Ethernet MAC DA
- Modify IPv4 SA (and update IPv4/TCP/UDP checksum)
- Modify IPv4 DA (and update IPv4/TCP/UDP checksum)
- Modify IPv4 ToS (DSCP)
- Modify TCP/UDP source port
- Modify TCP/UDP destination port
OpenFlow Startup Example

- Switch connects to controller. Both exchange Hello’s. Version negotiation.
- Controller requests features with Features-Request.
- Switch responds with Features-Reply to indicate supported features and OpenFlow-enabled ports.
Controller discovers topology of the network by forwarding LLDP’s out to the switching and inspecting the replies.
Each switch forwards the LLDP’s out the indicated ports.

Each switch forwards the LLDP’s to the controller as Packet-in’s.

Controller now knows the network topology.
Reactive OpenFlow Application Example

- System A needs to communicate with B.
- No path exists.
- OpenFlow controller reactively establishes path through the switching network.
Reactive OpenFlow Application Example

Packet-In ARP Reply

OpenFlow Controller

ARP Reply
Reactive OpenFlow Application Example

OpenFlow Controller

Flow-Mod Add

Flow-Mod Add

Flow-Mod Add

A

B

© 2011 Extreme Networks, Inc. All rights reserved.
Reactive OpenFlow Application Example

OpenFlow Controller

Packet-Out
ARP Reply

ARP Reply

IP Packet

A

B
SDN Technology Leadership *Continues*….

- 2010
  - Extreme starts discussions with Big Switch

- 2011
  - Extreme demos OpenFlow switch at INTEROP 2011
  - Extreme XOS supports Stanford software implementation of OpenFlow 1.0
  - Extreme enters partnering discussions with controller vendors (BigSwitch, Nicira etc.)
  - Extreme XOS OpenFlow distributed to over 20 customers

- 2012
  - Extreme demo OpenFlow at ONS
  - Extreme demo OpenFlow at Interop 2012
  - OpenFlow & Openstack GA in Extreme's Open Fabric 1QCY2013

- 2013
  - OpenFlow Rel. 2 with Open vSwitch extensions
  - Openstack Rel. 2
Vision: 3-Tier SDN Architecture
Next Generation Approach to Building Networks

SDN Applications

SDN Controller

SDN Data Plane
BlackDiamond® X8 - Introduction

Highest Consolidation
- 14.5 RU - 1/3rd of Rack
- 768 x 10G wire-speed
- 192 x 40G wire-speed
- 32 x 100G wire-speed

Ultra-Low Latency
- 2.3 uSec – Unicast*
- 2.4 uSec – Multicast*

Unmatched Capacity
- 20+ Tbps Capacity/Switch
- 2.56 Tbps Bandwidth/Slot
- 1 Million L2/L3 Entries

Server Virtualization
- 128K Virtual Machines
- VM Lifecycle Management
- VEPA, VPP, XNV™
- VR, MLAG, VPLS

High Availability
- 1+1 Management
- N+1 Fabric, Power & Fan
- N+N Power Grid
- EAPS, LAG, VRRP

Storage Convergence
- iSCSI, NFS, CIFS
- DCBx (PFC, FS, ETS)
- FCoE Transit

Power & Cooling
- Front-to-Back Cooling
- Variable Fan Speed
- 5.6W per 10GbE port *
- Intelligent Power Mgmt.

* Based on Lippis Test Report
BlackDiamond X8 Port Density & Diversity

- 48-Port 100/1000/10000MbE RJ45 Module
- 48-Port 10GbE SFP+ Module
- 12-Port 40GbE QSFP+ Module
- 12-Port 40GbE-XL QSFP+ Module
- 24-Port 40GbE QSFP+ Module
- 4-Port 100GbE-XL CFP2 Module
BlackDiamond X – Virtualized Multi-Tenant Open DC

Data Center 1

BDX8

VR1

VR2

VR3

X670

40G

10G

100G

VM

VM

VM

VM

Inter-DC Leased Line

OpenFlow Controller

VPLS1

VPLS2

VPLS3

VPLS PW

Data Center 2

BDX8

VR1

VR2

VR3

X670

40G

100G

VM

VM

VM

VM

© 2011 Extreme Networks, Inc. All rights reserved.
SDN with OpenFlow could:

- Use any networking topology
  - With little regard for ‘traditional CLOS Trees’
- Remove VLAN limitations
- Allow selective multipathing
  - By using switch hardware based hashing
- Dynamic End-to-End VM service provisioning
- Scale MAC and IP tables well beyond current TCAMs

SDN is becoming a viable Alternative
Software-Defined Networking: Myths vs. Reality

“SDN is just a research topic”
⇒ Seeing real production deployments for early adopters, moving to early majority

“OpenFlow can (can’t) solve any real-world network problem”
⇒ OpenFlow is at version 1.0 and can solve a few problems well, but is expanding
⇒ Key point: it is a *control* (not a data) protocol

“SDN == OpenFlow”
⇒ OpenFlow is an open, standard protocol between the control & data planes in an SDN architecture

“SDN == Network Virtualization”
⇒ Network Virtualization is an important, key application for SDN, but others are possible

“We’ve been doing SDN with scripts for years…”
⇒ There has been a trend toward central controllers, but a programmable dataplane is different from configuration automation

“OpenFlow can’t scale”
⇒ Google’s entire datacenter backbone runs on OpenFlow 1.0

“SDN/OpenFlow deployment is rip-and-replace”
⇒ All our current deployments have SDN/OpenFlow interoperating with existing networks
SDN Technology in the Adoption Cycle

* Terminology From Wikipedia – Gartner Technology Hype Cycle
Architectural Disruptions
Open architectures bring waves of innovation

Disruption

Open Architectures
- Drupal
- WordPress
- Joomla!
- php
- perl
- python
- Apache
- MySQL
- Linux OS
- X86 Architecture

Closed, Proprietary Systems

Compute

Mobile

Networking
Looking to the Future of SDN and OpenFlow

OpenFlow becomes the Android of networking

- Open network OS and controller for Ethernet switches and routers
- OpenFlow and related specifications all available as Open Source

Apps for every need

- From QoS, to PBR, to Identity Management, to Mobility Management, to multi-tenancy and so on

Controller vendors eventually become application developers

- Apps come in free, premium, and freemium models

Switch vendors continue to build and sell switches

- Much like phone vendors continue to sell phones

App Stores and Marketplaces for OpenFlow and SDN Apps
THANK YOU