Openet: Open Networking through Programmability

Tal Lavian tlavian@nortelnetworks.com
Nortel Network, Advanced Technology Lab
Agenda

- Two Evolutions: computer vs network
- Openet: open networking
- ORE and Openet Compositions
- Typical Applications
- Related Projects
- Summary
Think of computer evolution ...

Think of computer evolution ...

IBM  CDC  Digital  Amdel

1980s - Vertical Industry

2000s - Horizontal Industry

Applications
OSs
Peripherals
Hardware
What’s network evolution?

The inflection point is quickly approaching ...

06/28/18
Why Open Networking?

- Open network boxes to public
  - Current network devices are close systems

- Intelligence to network nodes because
  - Internet infrastructure evolves slow
  - Customers can not add new services

- Better use of network resources
  - Abundant bandwidth
  - Diversified clients’ needs

Move Turing Machine onto Device

Add local intelligence to network devices

while (true)

doLocalProcessingOnDevice ();
The Openet Approach

- Open networking through programming
  - A Service-enabled networking platform
  - Intelligence to commercial network devices
    - Network control and management
    - Packet forwarding and processing
  - Not impeding network performance and reliability
    - Forwarding
    - Security
- Enabling service creation and deployment
  - Value-added services across network elements
  - Dynamic and downloadable
- Standards and Partners
  - IEEE, IETF, Active Networks and FAIN
  - Columbia U., UC Berkeley and UPenn
  - MITRE, TASC, NetFuel and CSIRO
Control Console (Net Mgr)
- service initiation and policies
- network configuration
- resource administration
- repository maintenance

Control & Configuration

Repository Server
- Download codes, policies, configs

Switch
- ORE

Router
- ORE

End Apps
- 06/28/18

Openet Architecture
Openet Compositions

- **ORE**
  - Service creation and deployment
  - Service lifecycle management
- **Services**
  - Every network function is a service
  - Every service provides object APIs
- **ODK**
  - Service development and encapsulation
- **Management**
  - Service mgmt: initiation, policy and configuration
  - Manager on console and Agents on nodes
Openet: a node’s view

Oplet is a program unit wrapping services
ORE: the Openet Core

- ORE
  - Object-oriented Runtime Environment
  - Run customized software on network nodes
  - Neutral to heterogeneous hardware
  - Secure downloading, installation, and safe execution inside JVM
  - Fully implemented using Java
System Services: JFWD

- **Java Forwarding**
  - IP forwarding and routing
    - Diffserv marking
    - Filtering and diverting
    - Forwarding priority
    - Routing

- **Platform-independent APIs**
  - Implemented on Passport/Accelar and Linux
Function Services

- Common use utility
- Public neutral APIs
- Examples
  - HTTP: HTTP service
  - Shell: ORE interactive shell
  - Packet: packet handling (IP, TCP, UDP)
  - Logger: service runtime printout
  - OreServlet: Java servlet
Typical Applications

- **JDiffserv**
  - Diffserv forwarding and DSCP marking on Passport 8600
- **OpeCfg**
  - Dynamic configuration of optical port interfaces
- **IP filtering**
  - Dynamic priority changes on Passport 1100
- **JSNMP and JMIB**
  - SNMP/MIB access
  - Passport 1100 and 8600
- **Regatta: Fault recovery**
Challenges and Solution

- **Active Networks requires**
  - Open boxes to users
  - Networking programmability

- **Commercial network devices have**
  - Ever more use of hardware acceleration
    - Static and well-defined protocols and services
  - Little flexibility to introduce users’ intelligence
    - Allowing configuration rather than value addition

- **Our solution**
  - Openet
    - A programmable networking platform across devices
  - Active Services through Openet
Openet Architecture

Control Plane

Applications

Openet Systems

System Services

CPU System

Monitor status

New rules

Switching Fabric

Data Plane

(Wire Speed Forwarding)

Forwarding Processor

Statistics & Monitors

Traffic Packets
JDiffserv

- Goals
  - DSCP marking and re-marking
  - Priority forwarding or dropping
  - Filtering
- Passport
  - Model: 8600
  - Java 2
- ORE
  - version 0.4.1
  - JFWD/JDiffserv service
JDiffserv on Passport

- UDP sender
- Passport 1100B
- Passport 8600
- UDP receiver
- Linux PC
- UDP
- JDiffServ
- Device Console
- HTTP server
- Diffserv Monitor

06/28/18
Openet 17
JDiffserv Features

- **Marking Types**
  - Admission marking
  - condition marking (a.k.a., remarking)

- **Filters**
  - IP headers: 5-tuple
    - Source address and port, destination address and port
    - Protocol type
  - DSField: DSCP
  - Interface ports

- **Traffic profile**
  - Average rate and bucket size
  - Peak rate and bucket size

- **Action**
  - Marking then forwarding
    - 3-color marker: R/Y/G, RFC 2697
    - new DSCP
  - Dropping
Dynamic Classification

- **Objectives**
  - Implement flow performance enhancement mechanisms
  - **without** introducing software into data forwarding path
  - Service defined packet processing in a silicon-based forwarding engine
  - packet classifier
On-the-fly configuration

Dynamic Apps

Policy

Filters

Packet

Filter

Forwarding Processor

Forwarding Processor

Packet
Experimental Setup

Source 1
\texttt{tcp\_send()}

\begin{center}
\begin{tikzpicture}
\node[draw,fill=black!20] (source) at (0,0) {Source 1 \texttt{tcp\_send()}};
\node[draw,fill=black!20] (destination) at (6,0) {Destination \texttt{1. tcp\_recv() 2. tcp\_recv()}};
\node[draw,fill=black!20] (switch) at (3,0) {Accloud 1100B Routing Switch};
\node[draw,fill=black!20] (source2) at (0,-2) {Source 2 \texttt{tcp\_send()}};
\draw[->,thick] (source) -- (switch) node[midway,above]{100 Mbps};
\draw[->,thick] (switch) -- (destination) node[midway,above]{100 Mbps};
\draw[->,thick] (source2) -- (switch) node[midway,above]{100 Mbps};
\end{tikzpicture}
\end{center}
Throughput Results

![Throughput Results Chart]

- **Start 2nd Flow**
- **Change Priority**
- **End 2nd Flow**

Throughput Results:
- **Low Priority**
- **High Priority**
OpeCfg: Optical Control

- Goals
  - Content-aware bandwidth allocation

- Features
  - Content flow interception
  - Dynamic optical control

- 2 Alteon switches, 3 iSD boxes and 2 OM3400

---

**iSD-1**

- CO2 App
  - Pckt Intercept
  - Forward
  - Optical BW
  - NA/SA
  - Opecfg-C

**App Server**

**Linux-1**

**Alteon184-1**

**OM3400-1**

**iSD-2**

- OpeCfg

- Openet

- SNMP

- Packets

- Filters

- Redir

- CLI

- MIB

- OPE Fiber
DARPA-Funded Project

Active Nets Technology Transfer through High-Performance Network Devices

- Exploring new commercial network hardware
  - L2-L7 filtering
  - Fast content filtering and redirection
  - Strong and extensible CPU capability
  - Secure partitioning hardware and software
  - Supporting heterogeneous EEs

- Research platform
- Server and network collapse
iSD Architecture

iSD

Web Switch

Switch Interface:

Operating System (Alteon Embedded Linux
Or
RTIK – Real time Internet Kernel)

SSI
- Clustering
- Management
- Registry, CLI, …

NAAP

Application code

iSD Hardware
Alteon API – differentiates itself from other boxes

Alteon API
Generic
Has interface functions from L2, L3, L4, and L5+
Object Oriented designed
Can be extended to include future sophisticated functionalities

iSD or other intelligent processing devices

Serial Link Layer between Alteon and external devices

Alteon

Layer 2 processing and filtering
Layer 3 processing and filtering
Layer 4 processing and filtering
Layer 5+ processing and filtering

Others

This slide and other previous slides sent to you illustrate my point.
You can also see it more clearly if you ask the question “Why don’t I just use a powerful iSD without the Alteon?”
How Does Application Acceleration Work?

Data Enters Alteon Web Switch

Data Tunnel 1

Control Tunnel

Data Tunnel 2

Application Program

Alteon Linux Appliance (iSD)

Return Path

Data Exits

• Use of NAAP interface eliminates TCP overhead between switch & server and allows large data blocks (complete HTTP request)

• Server is removed from return data path

• Switching functions are removed from Linux application code and transferred to switch hardware

• See notes section below for further detail
T1: Programmable content switch

- Openet on Alteon
  - L2-L7 filtering
  - Fast content filtering and redirection to active services
  - Enhanced closely with Alteon features

- Alteon: new generation of content switch
  - Multiple processors and ASICs
  - Programmable microcode
  - L2-L4 and application filtering and processing
T2: Research Platform

- **iSD**: powerful and extensible computational plane
  - Partitioning hardware and software resources
  - Close interfaces to Alteon
  - Cluster computations
- **Network Research Platform**
  - Openet: active service enabling
  - Alteon: content filtering in real-time
  - **iSD**: integral computation inside the network
Nortel- Berkeley Openet Project

OSPF Area 10.10.0.0

Alteon 184

10.10.100.2/24

Accelar 1100

10.10.110.2/24

OSPF Area 10.20.0.0

1Gbps Link

OSPF Area 0.0.0.0

1Gbps Link

Firewall

Millennium Network

10.1.1.1

10.1.1.2

MLT
Any interest?

Looking for grant?

- Interested in summer internship?
  - Talk with me latter
How Can we Collaborate?

- Corry is not far from Soda
  - Are we EE+CS or EECS?
  - How can we bridge CS and EE projects?
  - Can we create a virtual lab? How?

- Openet and SmartNet are supported by DARPA
Summary

- Industry supports open networking
  - Network evolutions
  - Programmable network devices
- Openet is a programmable networking platform
  - Java 1.1 and Java 2
  - Devices: Alteon/iSD, Passport 1100 and 8600
  - OS: VxWorks, Linux, Unix and Windows
Q & A

Visit us at
HTTP://www.openetlab.org

Thank You !
Backup
What’s an Oplet?

- **Oplet:** a *self-contained downloadable unit*
  - Encapsulates one or more service objects
  - Contains service attributes, e.g., names
  - Eases secure downloading and service installation
  - Use other service oplets
- **Examples**
  - Active Networks services: EE
  - Java Forwarding services: JFWD
  - Base services: ODK
How a service is deployed?

- Service design and coding
  - Regular Java programming
- Service package
  - Oplets by ODK
  - JAR files
  - Uploading to downloading servers
- ORE start at Passport
- Service activation by ORE
  - Downloading, start and stop
    - Startup service
    - Shell service
- Service execution