Openet: Open Networking through Programmability

Tal Lavian
tlavian@ieee.org
Nortel Network, Advanced Technology Lab
Agenda

❖ Two Evolutions: computer vs network
❖ Openet: open networking approach
❖ DARPA-Funded Project: Openet/Alteon & Research Platform
❖ EE CS Collaboration
❖ Openet Features and Applications
❖ Summary
Think of computer evolution ...

1980s - Vertical Industry

IBM  CDC  Digital  Amdel

2000s - Horizontal Industry

Applications
OSs
Peripherals
Hardware
What’s network evolution?

The inflection point is quickly approaching ...
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

Add local intelligence to network devices

while (true)
doLocalProcessingOnDevice();

Move Turing Machine onto Device
Challenges and Solution

✓ 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

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

✓ Our solution
  - Openet
    - A programmable networking platform across devices
  - Active Services through Openet
  - Wire-speed data plane, powerful computation in the control plane
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
  • MITRE, TASC, and CSIRO
Openet Architecture

Control Plane

Data Plane
(Wire Speed Forwarding)

Traffic Packets
DARPA-Funded Project

Active Nets Technology Transfer through High-Performance Network Devices

❖ Exploring new commercial network hardware as a research platform
  • L2-L7 filtering
  • Fast content filtering and redirection
  • Strong and extensible CPU capability
  • Secure partitioning hardware and software

❖ Server and network collapse
  • Getting computation inside the network
  • Explore new ideas
Introducing the Alteon 780 Series
Large-Scale Data Center Content Switch/Router

❖ Alteon Webworking integrated with Nortel switching technology

• Distributed Alteon WebICs
• Alteon WebOS services
• Layers 2-7 switching
• 128G switch capacity
• 300+ FE; 60 GbE
• Data center class redundancy
• Future:
  – iSD and PCD integration
  – ATM and PoS connectivity
  – NEBS-3 compliance platform
Dynamic L2-L7 Filtering

L2-L7 Filtering Capability

- Source Address
- Source Port
- Destination Address
- Destination Port
- Protocol
- Diffserve Code Points
- Content Filtering
- Cookies Filtering

JFWD 5-tuple Filtering

- Divert the packet to the control plane
- Don't forward the packet
- Change DSCP field
- Set VLAN priority
- Adjust priority queue
- Modify session table
- Parsing request header
- Parsing application contents
The value of Alteon:
Alteon = Control + Processing + Storage + Programmable Services

Alteon intercepts selected flows and do some intelligent processing based on L2-L7 filtering.

An API is needed for filtering.

The emphasis is on interception and processing transparently. Entities at both ends may not be aware of the existence of the Alteon in the path.
What does Alteon do that cannot be done by another processor? (X=Processor+1Gbyte+SWs)

- Before X can do any processing X has to do filtering and/or redirecting the intended flows (flows in general sense, i.e. L2 – L7). Alteon does this within its architecture.

- Some intended flows require Ln processing. X processor has to process L2 – L(n-1) before Ln level processing can be done. Alteon prepares up to and including L(n-1) level processing within its architecture.

- X processor can be an iSD or any general processor as long as there is an Alteon API.
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

High-speed Link Layer between Alteon Switch and iSDs

Alteon

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

Others
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
Any interest?

Looking for a grant?

- Interested in summer internship?
  - Talk with me later
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
Openet on Alteon is a powerful programmable networking platform
- Great Research platform to explore new ideas
- Commutation embedded within the network
- Linux development environment

Gigabit speed data-plane with programmability on the control-plane

Openet-Alteon is a sophisticated platform for developing real applications and for introducing services on-demand

Openet-Alteon SmartNets requires your collaboration!
Q & A

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

Thank You!
Backup
Openet Features and Applications
Openet Architecture

End Apps

Switch

Repository Server

Download codes, policies, configs

Router

Control & Configuration

Control Console (Net Mgr)
- service initiation and policies
- network configuration
- resource administration
- repository maintenance

Control

Data path

Download

User request

End Apps
- use of service
- request on the fly

Openet DARPA UC Berkeley
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

Diagram:
- Oplet
- Service
- ORE
- Java Virtual Machine
- API Extensions
- Hardware
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**
**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

Openet DARPA UC Berkeley
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
On-the-fly configuration
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**
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
Experimental Setup

Source 1
\text{tcp\_send()}

100 Mbps

Acclar 1100B Routing Switch

100 Mbps

Destination
1. \text{tcp\_recv()}
2. \text{tcp\_recv()}

Source 2
\text{tcp\_send()}

100 Mbps

Openet DARPA UC Berkeley
Throughput Results

Throughput Results graph showing the network performance with time in seconds on the x-axis and Mbps on the y-axis. The graph distinguishes between Low Priority and High Priority flows, with specific events labeled as Start 2nd Flow, Change Priority, and End 2nd Flow.