Programmable Network Node: Applications

Tal Lavian\textsuperscript{1} tlavian@IEEE.org

Rob Jaeger\textsuperscript{1,2} rojaeger@NortelNetworks.com rfj@cs.umd.edu

\textsuperscript{1} Technology Center, Enterprise Solutions, Nortel Networks
\textsuperscript{2} Department of Computer Science, University of Maryland
Accomplishments

- JVM on a silicon-based L3 Routing Switch
- ORE - Oplet Run-time Environment
- Java-enabled Device Architecture
- SNMP MIB API
- Network Forwarding API
- Active Networks applications:
  - dynamic control and modification of ASIC forwarding
Nortel Java-Enabled Device Architecture

Nortel Java-Enabled Device Architecture

Device HW

Operating System

JVM

Oplet Runtime Env

ORE Service

Java API

JNI

Device Drivers

Device Code

C/C++ API

Oplet

AN Packet Interception
Tools

- **MIB API**
  - Monitor device Management Information Base variables
    - MIB
    - RMON and RMON-II
    - DiffServ

- **Network API (JFWD)**
  - Interface to Filters
    - set packet drop filters
    - intercept packets
    - carbon copy packets while forwarding at line-speed
MIB API

• API uses a MIB Map to dispatch requests to variable access routines
  • Different parts of the MIB tree can be serviced by different mechanisms
  • Two main schemes:
    • An ad hoc interface to the SNMP instrumentation layer
    • A generic SNMP loopback
Network API: Dynamic Configuration of Forwarding Rules

[Diagram showing the interaction between Java Services and Forwarding Rules]

Forwarding Rules
Forwarding Processor

SW

HW
Network API: Control-vs-Forwarding Plane

Centralized, CPU-based Router

Routing SW

CPU

Control + Forwarding Functions combined

Slow

Forwarding-Processors based Router

Control Plane

CPU

Forwarding Processor

Forwarding Processor

Forwarding Processor

Wire

Speed

Control separated From forwarding

Java-enable Network Devices
Network API: Packet Capture

AN Apps → CPU

CPU → Forwarding Processor

Forwarding Processor → Forwarding Processor

Forwarding Processor → Forwarding Processor

Forwarding Processor → Forwarding Processor

JFWD to Divert or Copy

Wire Speed
Applications

- **Active Network Management**
  - Proactive Network Management
  - Diagnostic Agents

- **Dynamic DiffServ Classifier**

- **Active Intrusion Detection**

- **Multicast Caching**

- **IP Accounting**

- **Application-Layer Router-Server Collaboration**

- **Pseudo Default Drop Capability**
Active Network Management

- Download Oplet Service to the device.
- Monitor MIB variables
  - Might be complex conditions
  - Trend analysis
  - DiffServ, RMON-II, etc... MIBs
- Report “events” to NMS
  - drop rate, packets/second
- Allow Service to take action
- Download application
- Adjust parameters based on direction from NMS

![Diagram showing processes involving NMS, router, monitor, and appropriate application with download arrows and no more polling note.](image-url)
Proactive Network Management

- **Device-based Intelligence is Dynamic**
  - Static Management
    - SNMP set/get mechanisms
    - Telnet, User Interfaces (cli, web, etc...)
  - Dynamic Closed-loop Management at Network Node
    - capable of dealing with new and difficult situations
    - autonomous and rational properties.
    - dynamically system monitoring & modification
    - report status and trends
  - Monitor MIB to identify poor performance and notify NMS prior to failures
  - Downloaded service can instantiate new services
Diagnostic Agents

• **Automatic traceroute from edge router where problem exists.**
  — Each node reached generates a report to NMS
  — Traceroute code “moves” to next node in path
  — Mobile agents identify router health
  — Create logs for NMS
Dynamic DiffServ Classifier

- Set router filters to sample packets from edge device host ports
- Identify real-time traffic (RTP flows)
- Set filter on port to adjust DS-byte value based on policy
- Keep track of filters set
- Remove filters no longer in use
Dynamic DiffServ Classification

- Sample packets, set filters to modify DS-byte for Per-Hop-Behavior modification
Active Intrusion Detection

- Intruder is identified by Intrusion Detection software
- Intruder signature is identified
- Mobile agent is dispatched in direction of intruder (based on physical port of entry)
- Mobile agent “chases” intruder and terminates him (shuts down link, reboot host, notify NMS)
Multicast Caching

- Reliable Multicasting
- Distribute error control throughout multicast tree
- Retransmission a local node keeps control close to lossy links
- Balances processor load away from multicast source
IP Accounting

- Project ABCD (Active Bean Counter in Device)
- Perform usage accounting at edge node
- PreCorrelate/aggregate/reduce accounting record on-site
- $1 rule for billing
- Real-time billing can be realized
- Customize billable resources
Application Layer Collaboration Among Routers and Servers

- **Server farm load balancing**
  - server state monitored; rerouting based on congestion/load

- **Auctioning Applications**

- **Bandwidth Broker**
Server Collaboration

- Supports **distributed** computing applications in which network devices participate
  - router to router
  - server to router
- Supports **Intelligent Agents**
- Supports **Mobile Agents**
Bandwidth Broker Collaboration

- Routers Monitor RMON and DIFFSERV MIB
- Report Per-IP Address, Per Protocol statistic to resource broker
- Adjust DS-byte and Per Hop Behavior based on Bandwidth Broker directions
Real-time forwarding Stats and Monitors

Diagram showing the flow of data from AN Apps to CPU and vice versa, with modules for Forwarding Rules, Forwarding Processor, Statistics & Monitors.
Dynamic - On the Fly Configuration

- From downloadable Java application, we can modify the behavior of the ASICs
Mixed Topology of AN system

NO need to know the AN topology ahead of time

- AN Node
- None AN Node
Active Node Topology Discovery

NO need to know the AN topology ahead of time