SwitchWare: Lessons Learned, and Where Next?

Microsoft Research, Redmond, WA
December 19th, 2006

Jonathan M. Smith
University of Pennsylvania
http://www.cis.upenn.edu/~jms
Active Networks enable new distributed systems

Smart Hosts + Dumb Switches = Passive Nets

Smart Hosts + Smart Switches = Active Nets
Virtual Infrastructures, e.g., IP

- IP is a network interoperability layer
- Interoperable through minimality:

  - Packet Format, Addressing
  - Overlays (running at hosts)
  - Virtual Network Infrastructure (runs globally)
  - Subnetworks (run IP locally)

Idea: Make waist Programmable
Accelerate Network Evolution

- Create *programmable* network nodes; standardize the *programming model*, not the nodes
- Change from *Political Tempo* (standards) to *Technical Tempo* (code)
- Balance Usability, Flexibility, Performance and Security
SwitchWare Approach

- Modern Programming Language technology can help with safety and security, maybe performance (cntxt X)?
- Build flexible node executing programs written in such languages
- Use P.L. typing to restrict programs for safe multiplexing of node in a network
A Language-Oriented Solution in 3 Parts

- **Switchlet Language** for users (SL)
  - Formal semantics, restricted programs
  - E.g., restricted CAML or Domain Specific Lang.
- **Wire Language** for communicating (WL)
  - Enforce formal semantics across boundaries
  - Cryptographic signatures + hashes
- **Infrastructure Language** for Virtual Machine (IL)
  - Formal semantics supported on metal: run-time
  - Replace O.S. with P.L. runtime
SwitchWare Architecture

Part 1

ALIEN Library

ALIEN/Caml/OS

Dynamic Integrity Checks

Static Integrity Checks

Node-Node Authentication

Caml Switchlet

Caml Switchlet

ALIEN/Caml/OS

Part 1
ALIEN Active Loader

D. Scott Alexander’s Ph.D. thesis

- Libraries
- Core Switchlet
- Loader
- Runtime (Caml)
- OS (Linux)

Protection Boundary
Mutability Boundary
The ALIEN Loader

- startup routines
- active program loading
- system console
- mechanism only

Diagram:

- Libraries
- Core Switchlet
- Loader
The Core Switchlet

- language primitives
- OS access
- network access
- thread access
- loading support
- message logging
- mechanism and policy
The Library

- “Everything else”
- IP
- UDP
- Utility functions
Active Packets in ALIEN

- If ANEP header indicates ALIEN
  - SANE processing as part of ANEP
  - Code portion is loaded
  - `func` is called with code, data, and func name as arguments
saneping Performance (533 Mhz Alpha PC, 100M Ethernet)
Breakdown of Costs in Alien

- Kernel/wire: 26%
- Authentication: 25%
- Transmission related: 4%
- Caml overhead: 20%
- Information gathering: 10%
- Marshaling: 16%
SwitchWare Architecture

Part 2

Dynamic Integrity Checks

Static Integrity Checks

PLAN

Packet

ALIEN/Caml/OS

Node-Node Authentication

PLAN Packet
Packet Language for Active Networks (PLAN)

- Hicks, Kakkar, Moore, Gunter, Nettles
- Active Packet-based approach
- Highly-restricted domain specific language (safe “glue” language, like the UNIX shell), extensible via ALIEN
- Active extensions for restricted (“privileged”) things
PLANet: 2-level Architecture

PLAN : • safe • in packet

Services: • general • on router (e.g., ALIEN)

PLAN packet

Install

Routing

...
Trust Management

Joe

OK

Service install
Trust Management

Ed

Service install

not allowed
Form of Service Policies:

Access

- PLAN packet
- Routing
- Sending packets

default user
Form of Service Policies: Access

privileged user

Joe

Install
Routing
Sending packets
Form of Service Policies: Usage

default user

PLAN packet

Sending packets
Form of Service Policies: Usage

Joe

Sending packets

privileged user
Security Procedure

PLAN packet

Routing

Sending packets

arrival as default user
Security Procedure

authentication
Security Procedure

namespace adjustment

Joe

Install
Routing
Sending packets
Security Procedure

Joe

Sending packets

... if Joe then more bandwidth...

per-usage adjustment
Chunks - units of authentication

- **Unit of evaluation in PLAN**
  - like a suspended function call
- **First-class**
  - can be manipulated as data within PLAN programs
- **Useful programming construct**
  - encapsulation via `eval`
Chunks - in PLAN packets

Fixed fields

evalDest ...

Chunk

code args fun
Ping packet

```
fun ping(source, dest) = 
  ... 
  fun ack() = 
    print("Success")
```

dest

evalDest  ...  code  args  fun

source, dest  ping
Core Service

\[
\text{authEval: } \text{‘a chunk } \times \text{ sign } \rightarrow \text{ ‘a}
\]

- Takes a chunk and an HMAC digital signature and authenticates the chunk
- If successful, performs namespace adjustment and evaluates the chunk
Application: An Active Firewall

- Rather than filter external packets, restrict their privilege.
- Accomplished by encapsulating incoming packets with service-restricting chunk.

```haskell
fun wrap(c, sign) =
  (zeroRB(); authEval(c, sign))
```
Experimental Setup

Trusted network

Untrusted network
Outgoing Ping

S -> F -> D

ping
Returning Acknowledgement

Firewall signs as and encapsulates packet chunk
Firewall-wrapped Ping packet

```plaintext
fun wrap(c, sign) =
  (zeroRB();
   authEval(c, sign))
```

```
dest
```

```
ping chunk, ...
```

```
wrap
```
Firewall Performance

![Bar Chart]

<table>
<thead>
<tr>
<th>Elapsed Time (ms)</th>
<th>no payload</th>
<th>max payload</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ping</td>
<td>2.5</td>
<td>3.5</td>
</tr>
<tr>
<td>+Firewall</td>
<td>3.0</td>
<td>4.0</td>
</tr>
</tbody>
</table>
Firewall Overhead Breakdown

Elapsed Time (ms)
SwitchWare Architecture

Node-Node Authentication → ALIEN/Caml/OS

Recovery → AEGIS

Dynamic Integrity Checks

Static Integrity Checks

Part 3
Integrity of the Runtime/O.S.: AEGIS Secure Bootstrap

● Integrity Guarantees for Dynamic Integrity Checking

- Trusted Repository
- SwitchLets
- O.S. (BSD)
- ROMs, Boot Block, ...
- BIOS Sec. 1
- POST
Resource Controlled AN Environment (RCANE):

Application

Execution Environment A

Application

Execution Environment B

Node Operating System (e.g., Nemesis, XP, Linux, Vista?)

“A” share of machine

“B” share of machine
SwitchWare Architecture

- PLAN Packet
- Caml Switchlet

ALIEN/Caml/OS

- PLAN Library
- Node-Node Authentication
- Recovery

AEGIS

Dynamic Integrity Checks
Static Integrity Checks
Lessons Learned

- Interoperability problems not removed; just moved.
- Performance acceptable for access networks
- CAML technical win, marketing lose
- Restricted language for packets a win
  - May need to augment with cryptographic tools
- Did not allow enough time for network versus node work (should have been 5-6 year project, not 3+)
- Convincing (not ping) Active Applications hard
Next: Active Router Control (Active Border Gateways?)

- IP Router/Forwarders co-located with Active Elements:

- Forwarding Tables
  - IP
  - IP
  - IP

- Routing Policies and Decisions (and New Services)

- Active Element
  - LAN
Virtualized Transport Infrastructure

Xen Programmable Edge Router Technology (XPERT): flexible & sliceable

Acknowledgments:

● **SwitchWare** was a joint project of Penn and Telcordia, supported by DARPA
  → Extensive literature
    ✫ Responsible parties named there!
  → **RCANE** was a collaboration with Cambridge University, described in Paul Menage’s Ph.D., and supported at Penn by NSF

● **Hewlett-Packard, Intel, 3Com & Nortel**
Questions and Discussion
Backup
ARC: Internet Control Plane

Policies

Route Updates

Router A

Router B

Measure

Measure
Active Network Architecture

(see April 1999 “IEEE Computer”)
Research Issue: Restricting Programs in the Network

- Node safe versus network safe
Netwide Sense Data Selection

- Nets and computers improving exponentially. Humans not.
- Active nodes contain “delegates”
  - select information (watching a million cameras at once......)
  - forward towards you for consumption
  - your senses extended into the network