SwitchWare

Accelerating Network Evolution

Jonathan M. Smith, David J. Farber, Carl A. Gunter and Scott M. Nettles
University of Pennsylvania

W. David Sincoskie and Mark E. Segal
Bell Communications Research, Inc.
The Problem

- Pace of network service development **SLOW**
  e.g., IETF->Cisco->ISPs (5-8 years)
- **NEED** for standardization (interoperability)
- IP Packet format **WRONG** level of abstraction!
Approach: “30,000 Foot Level”

- Programmable interoperability layer
- Infrastructure-provided, e.g., switches, ...
- Programmable (to some degree) by users
- Or, one way to “Active Networks”
- MANY challenges: security, performance, ...
Routing IP Packets

Model: Store and Forward

1. Dequeue Packet from Input Port
2. Determine “best” Output Port
3. Queue Packet on Output Port
SwitchWare switching

- Store, COMPUTE and Forward!
Applications, or Why bother?

- Self-Paying Information Transport
  - Routing by economics; policy with $$$

- Network Management:
  - in-band OR out-of-band
  - inject diagnostics code *as-needed*
  - e.g., Morris worm code patches

- Dynamic bandwidth aggregation (striping)
Problems

- Performance: Well, yes but Correctness FIRST!
- Safety: Good guys can make mistakes...
- Security: Bad guys can program too...
- Network Infrastructure is *shared*
  - it MUST work (telephony as example)
- Can we get **FLEXIBILITY** *and* **SECURITY**?
Security IS NOT Cryptography!!!

- **Security is:**
  - *Right information to*
  - *Right people at*
  - *Right place at*
  - *Right time*

- **This is policy**

- **Insecure systems exhibit policy failures**
Security: Enforcing Policy in 3 Parts

- Identification
- Access Control
- Quality of Service
  » versus “Denial of Service” attacks
A Language-Oriented Solution in 3 Parts

- Switchlet Language for users (SL)
  - formal semantics restrict programs
- Wire Language for communicating (WL)
  - formal semantics across boundaries
- Infrastructure Language for Virtual Machine (IL)
  - formal semantics supported on metal: run-time
What DOESN’T T work...

<table>
<thead>
<tr>
<th>Java/TCL</th>
<th>SL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Java bytecodes</td>
<td>WL</td>
</tr>
<tr>
<td>C</td>
<td>IL</td>
</tr>
</tbody>
</table>
Penn/Bellcore Active Router

<table>
<thead>
<tr>
<th>CAML</th>
<th>SL</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAML bytecodes</td>
<td>WL</td>
</tr>
<tr>
<td>CAML</td>
<td>IL</td>
</tr>
</tbody>
</table>
Penn/Bellcore SwitchWare Target

<table>
<thead>
<tr>
<th>Verifiable ML--</th>
<th>SL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Encrypted Verified Intermediate Language</td>
<td>WL</td>
</tr>
<tr>
<td>ML++</td>
<td>IL</td>
</tr>
</tbody>
</table>
Target Platforms

- Shared Memory MP as “Switch”
  - HP Netserver LS (Pentium)
  - SGI Challenge (MIPS R4000)

- ATM and Ethernet Line Cards

- Bellcore OPCv2 ATM cell buffer/mux
Accelerating Network Evolution

- Programmable services
- Extensibility of infrastructure
- Security by design, not afterthought
- Partitioning resources under policy
- Portability and technology independence

http://www.cis.upenn.edu/~jms/white-paper.ps
Sharing and Security

Application Modules

Protection

Level Boundary

Scheduler/Multiplexer

Traditional Operating System
### SwitchWare Contributions to Active Nets

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Formal Model</td>
<td>***</td>
<td>*</td>
<td></td>
<td></td>
<td>*</td>
<td>***</td>
</tr>
<tr>
<td>2. Runtime Env.</td>
<td>**</td>
<td>**</td>
<td>***</td>
<td></td>
<td>**</td>
<td></td>
</tr>
<tr>
<td>3. Router</td>
<td>**</td>
<td>**</td>
<td>**</td>
<td>**</td>
<td>**</td>
<td>**</td>
</tr>
<tr>
<td>4. Security</td>
<td>*</td>
<td>**</td>
<td>**</td>
<td>***</td>
<td>**</td>
<td>**</td>
</tr>
<tr>
<td>5. OPCv2</td>
<td>***</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>**</td>
</tr>
</tbody>
</table>

* = relative importance
Project Tracks and Timeline

**Formal SwitchWare Semantics**

- Idealized Formal Language
  - Specify SwitchWare Language

**SwitchWare Run-Time System**

- Prototype Run-Time on SGI
  - Support Active Router
  - Measure and Extend to OPCv2

**Applications and Active Routing**

- Active Router
  - SwitchWare Applications
  - Extend Applications

**Timeline**

- Year 1
- Year 2
- Year 3