The Tempest

Practical Experience in Control Architecture Development

Sean Rooney, May 1998
The Tempest

Control Architectures

- a control architecture is ... ;
- examples: SS7, ATMF UNI/NNI, RSVP;
- congruent with the services that the network they control is expected to support;
- so how many will we need?
The Tempest

Control Architectures

• one (big one) according to one school of thought;
• how can we know about what the control needs of future services?
• suppose we could; would this be a wise approach?
The Tempest

Control Architectures

- the Tempest environment permits multiple control architectures:
  - Q.Port (A portable imp. of UNI signalling)
  - RSVP (Reservation Protocol for IP)
  - Hollowman (A proof-of-concept open control architecture)
- others candidates:
  - B-ISUP, XBind, TINA, MPLS, UNITE, OPENET etc.
- which is “the best”?
The Tempest

Control Architectures

- an ATM control architecture contains:
  - addressing scheme;
  - signalling transport;
  - a UNI, NNI;
  - a switch interface;
  - switch controller (Routing, CAC, OAM);
  - host controller;
  - an API.

- their design and implementation are influenced by the nature of the services the control architecture supports.
The Tempest

Control Architectures

• adding extra functionality carries a cost;
• requirement trade-offs:
  – Security/Ease of use,
  – Reliability/Efficiency,
  – Generality/Simplicity.
• do you want to pay for what you don’t need/use/want?
The Tempest

Case Study: Q.Port

Application

Host Signalling
- Host Manager
- Host Call Control
- H-Q.93B/Q.SAAL
- AALCP

Switch Signalling
- Switch Manager
- Switch Call Control
- AALCP
- AALCP
- Fabric Control

Application

Host Signalling
- Host Manager
- Host Call Control
- H-Q.93B/Q.SAAL
- AALCP

Ariel Interface

Prospero

University of Cambridge
Computer Laboratory
The Tempest

Case Study: Hollowman
The Tempest

Control Architectures

- Q.Port & Hollowman offer similar functionality (e.g. pt-to-pt, pt-to-mt signalling)
- Hollowman is simpler, Q.Port more general;
- Example differences:
  - addressing;
  - signalling transport;
  - switch interface;
- Hollowman extensions for “advanced features”, connection caching, soft PVCs, anycast, groupcast, connection closures ...
## The Tempest

### Control Architectures

<table>
<thead>
<tr>
<th>Name</th>
<th>Latency (ms)</th>
<th>Environment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q.Port/GSMP</td>
<td>20</td>
<td>Off-switch on Linux PC</td>
</tr>
<tr>
<td>Hollowman/GSMP</td>
<td>5+2</td>
<td>Off-Switch on Solaris</td>
</tr>
<tr>
<td>UNI, Fore ASX-200WG</td>
<td>10</td>
<td>On Switch</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>Size</th>
<th>Install</th>
<th>Caveat</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q.Port</td>
<td>60M</td>
<td>With docs, Tcl, passMT</td>
<td>Not inc. FORE, switch interface.</td>
</tr>
<tr>
<td>Hollowman+Tempest</td>
<td>2M</td>
<td></td>
<td>Not inc. SNMP, ACE, GSMP, FORE</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>Footprint</th>
<th>Caveat</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hollowman switch manager</td>
<td>4M</td>
<td>90%+ due to the DPE</td>
</tr>
<tr>
<td>Q.Port switch manager</td>
<td>7M</td>
<td>Using Tcl !</td>
</tr>
</tbody>
</table>
The Tempest

What about ubiquity?

• one solution; *networks without frontiers*:
  – all switches run a divider server;
  – run any control architecture anywhere;
  – however ...

• another solution; use a “glue” control architecture;

• P-NNI seems like a good candidate;

• Hollowman and Q.Port can interoperate.
The Tempest

Conclusion

- switches will need several CAs (MPLS, ATMF ...);
- multiple instances of ATMF signalling:
  - having different function subsets;
  - privileging different requirements;
- also role for more advanced ATM CAs.