The Tempest

Practical Experience in Control Architecture Development

Sean Rooney, May 1998

University of Cambridge
Computer Laboratory

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 ?

University of Cambridge
Computer Laboratory

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 ?

University of Cambridge
Computer Laboratory

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” ?

University of Cambridge
Computer Laboratory

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.

University of Cambridge
Computer Laboratory

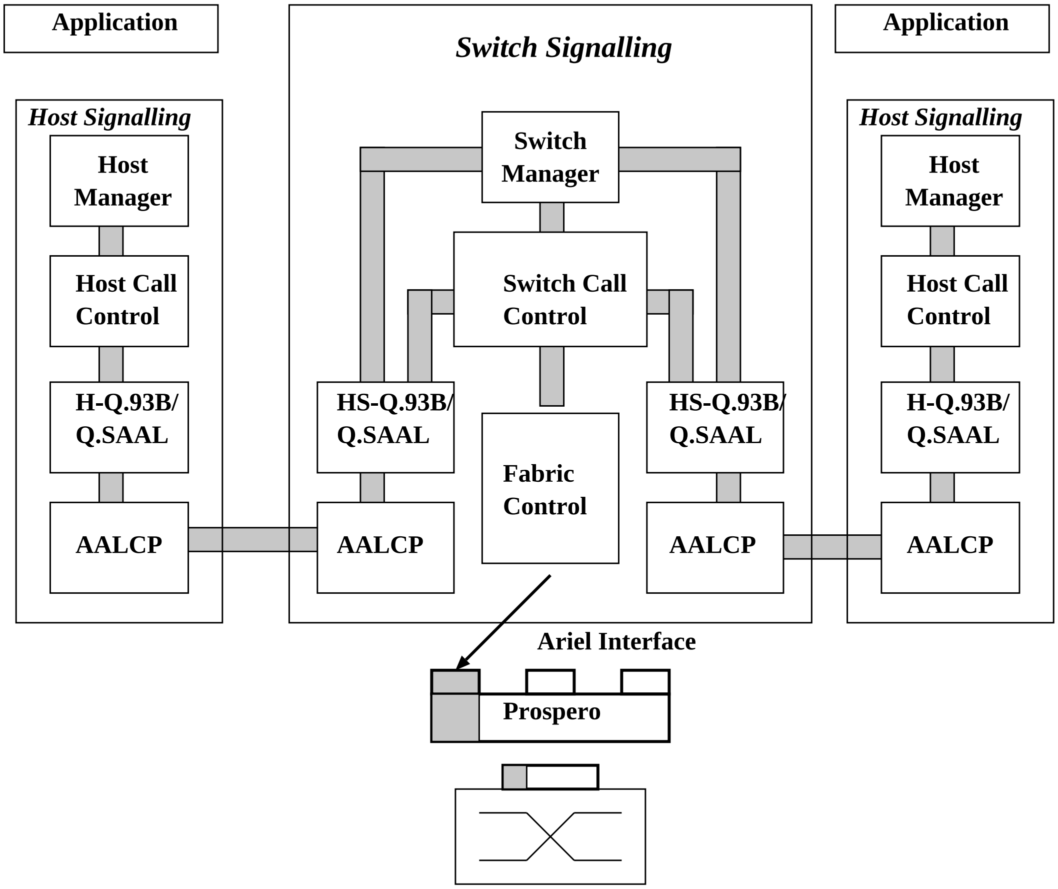
Control Architectures

- 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 ?

University of Cambridge
Computer Laboratory

The Tempest

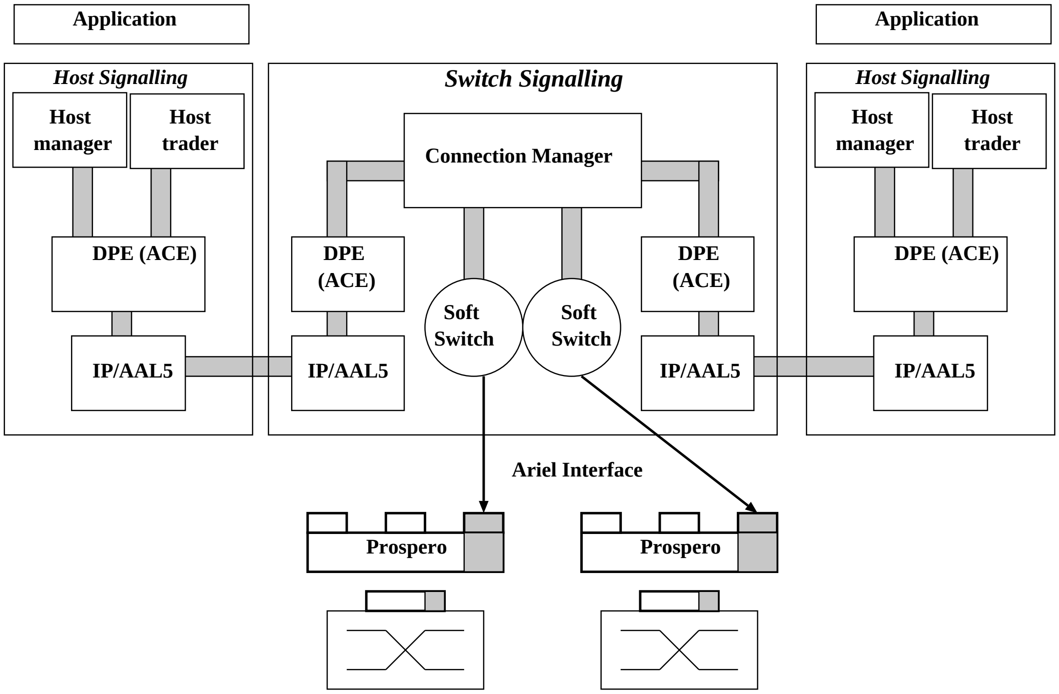
Case Study: Q.Port



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 ...

University of Cambridge
Computer Laboratory

The Tempest

Control Architectures

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

Name	Size Install	Caveat
Q.Port	60M	With docs, Tcl, passMT Not inc. FORE, switch interface.
Hollowman+Tempest	2M	Not inc. SNMP, ACE, GSMP, FORE

Name	Footprint	Caveat
Hollowman switch manager	4M	90%+ due to the DPE
Q.Port switch manager	7M	Using Tcl !

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.

University of Cambridge
Computer Laboratory