Distributed Systems and Network Architectures

NetArch 2009, Ascona, CH
March 18th, 2009

Jonathan M. Smith
University of Pennsylvania
http://www.cis.upenn.edu/~jms

Design Space for Distributed Applications

- Application Requirements
  - →Delay, delay variance, reliability, privacy,…
- Network Conditions
  - →Error, loss, congestion, topology,…
- Protocol Elements
  - →Links, multiplexers, headers, ACKs,…
- Clark/Tennenhouse SIGCOMM 90

Protocol Design (in the e2e world)

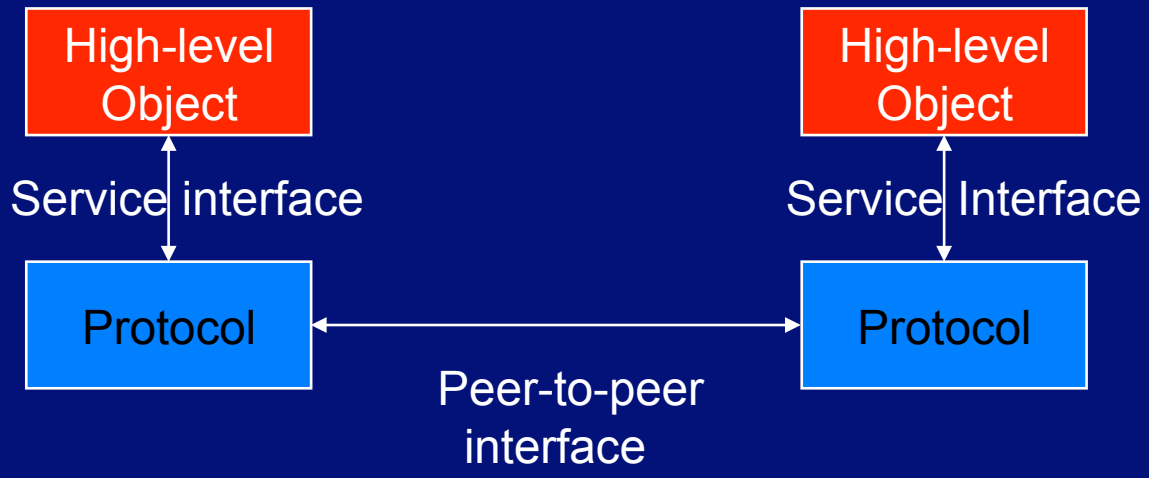
- Begins with problem to be solved, including assumptions
  - →e.g., TCP's "reliable bytestream", over IP
- Optimization:
  - →Measure
  - →Identify common case
  - →Make it fast
  - →Repeat until satisfied.....

Critique of Methodology

- Pessimistic Design Style
  - → Assume worst-case
  - → Pare away functions to get "fast-path"
- Optimizations Fragile
  - → Environment Changes (WWW)
  - → Common Cases Change (delay, loss, ...)
  - → Things can break BADLY! (try at home :-)

Layered Network Protocols

- Fixed service and peer interfaces
- Static functions / algorithms

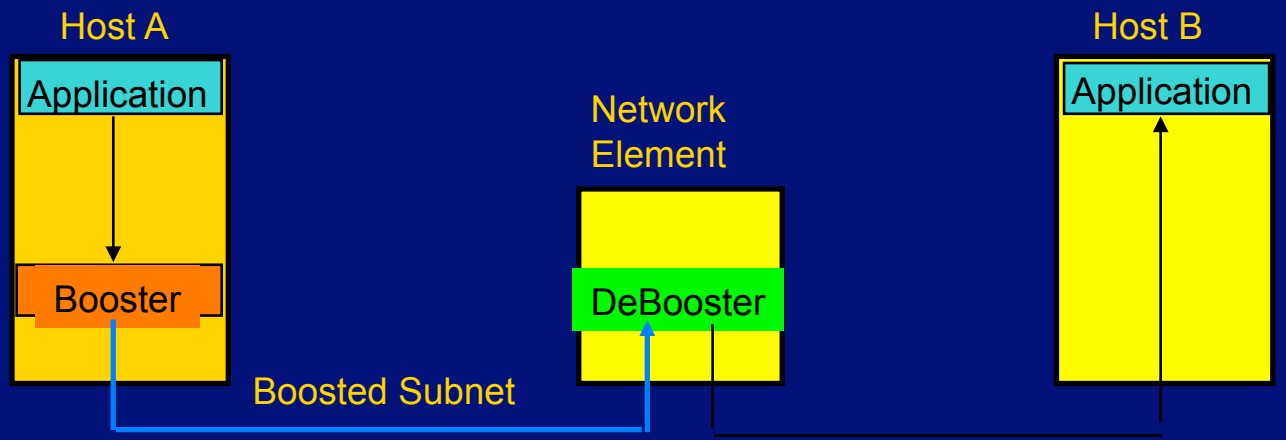


An alternative methodology

- Assume things are working well
- Detect when they are not (policy)
- Add functions (mechanism) to fix
- Functions are called "protocol boosters"
- An optimistic approach to transparently achieving high end-to-end performance

Protocol boosters* for links

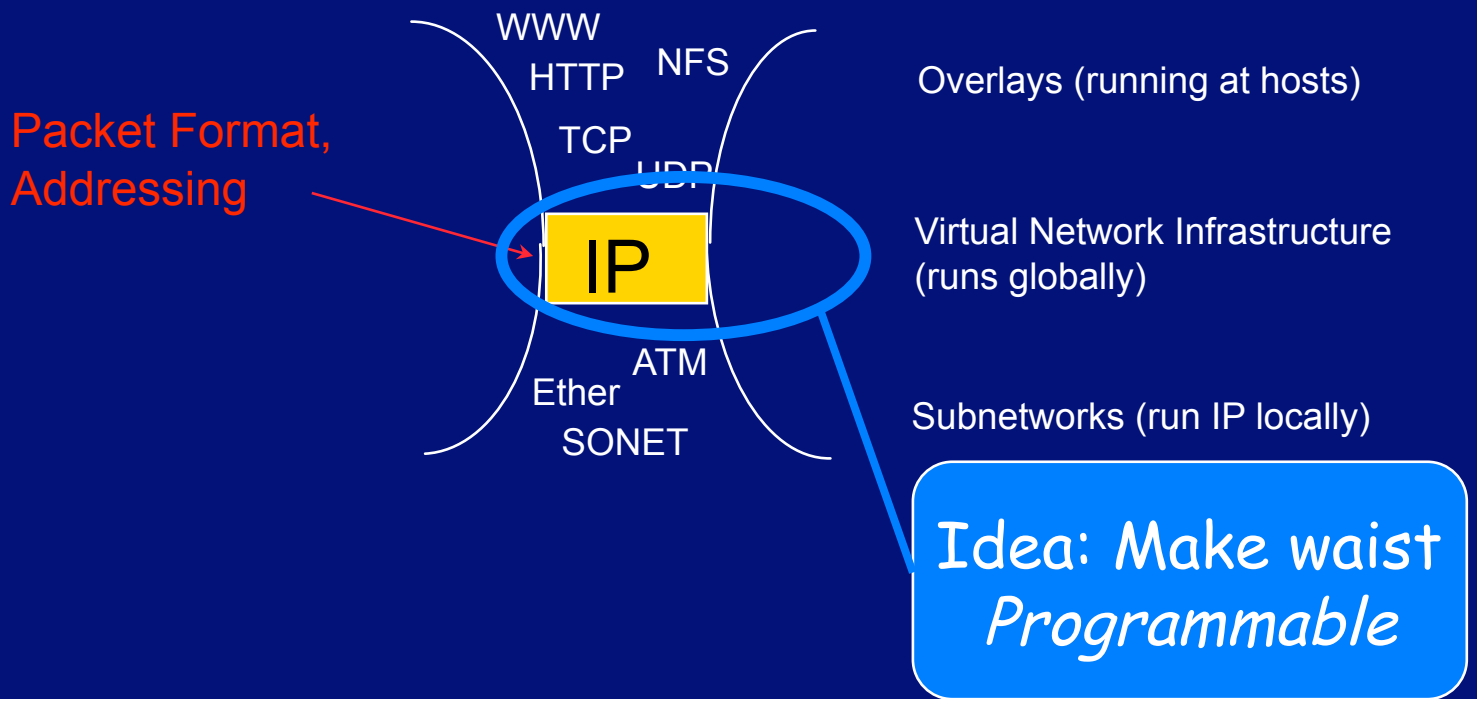
- Earliest work, RFC 5, "Decode-Encode Language", Rulifson
- Protocol Elements added "as-needed"
  - → D. Ritchie "A Stream I/O System", BSTJ '84
- Useful to meet dynamic requirements
  - → Tschudin, "Flexible Protocol Stacks", SIGCOMM '91
  - → O'Malley & Peterson, "A Dynamic Network Architecture", ACM ToCS, '92



* "Protocol Boosters", Feldmeier, et al., IEEE JSAC, 1998

Virtual Infrastructures, e.g., IP

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

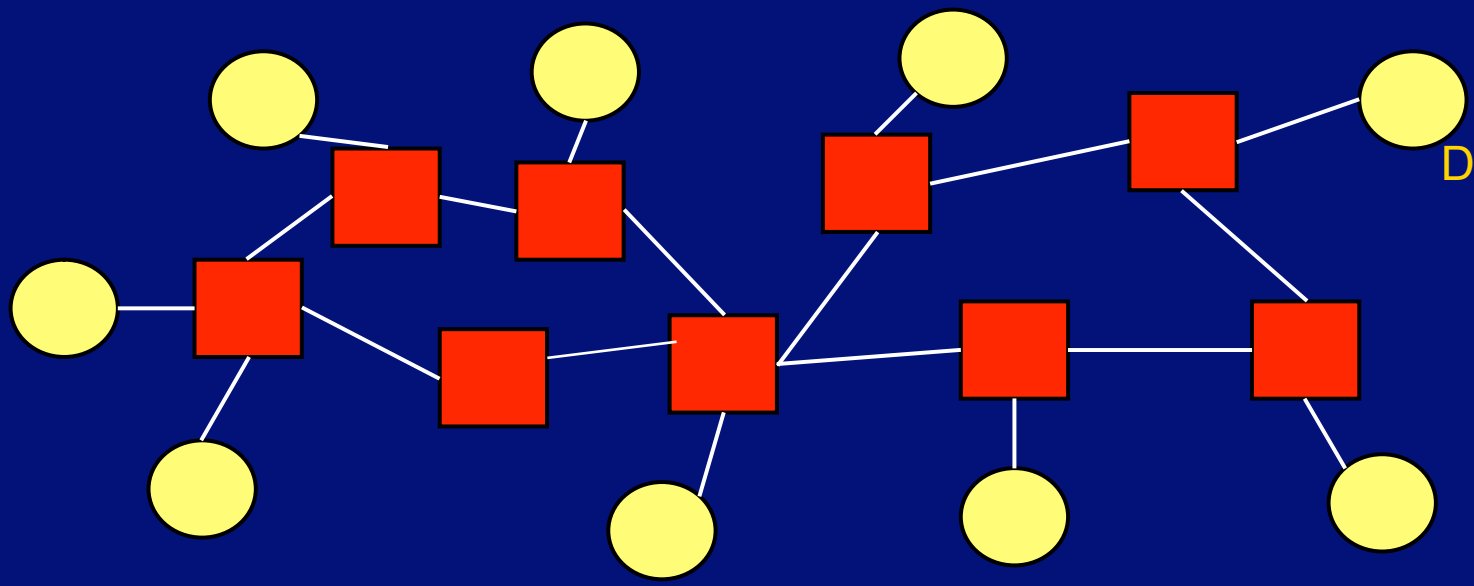


Accelerate Network Evolution*

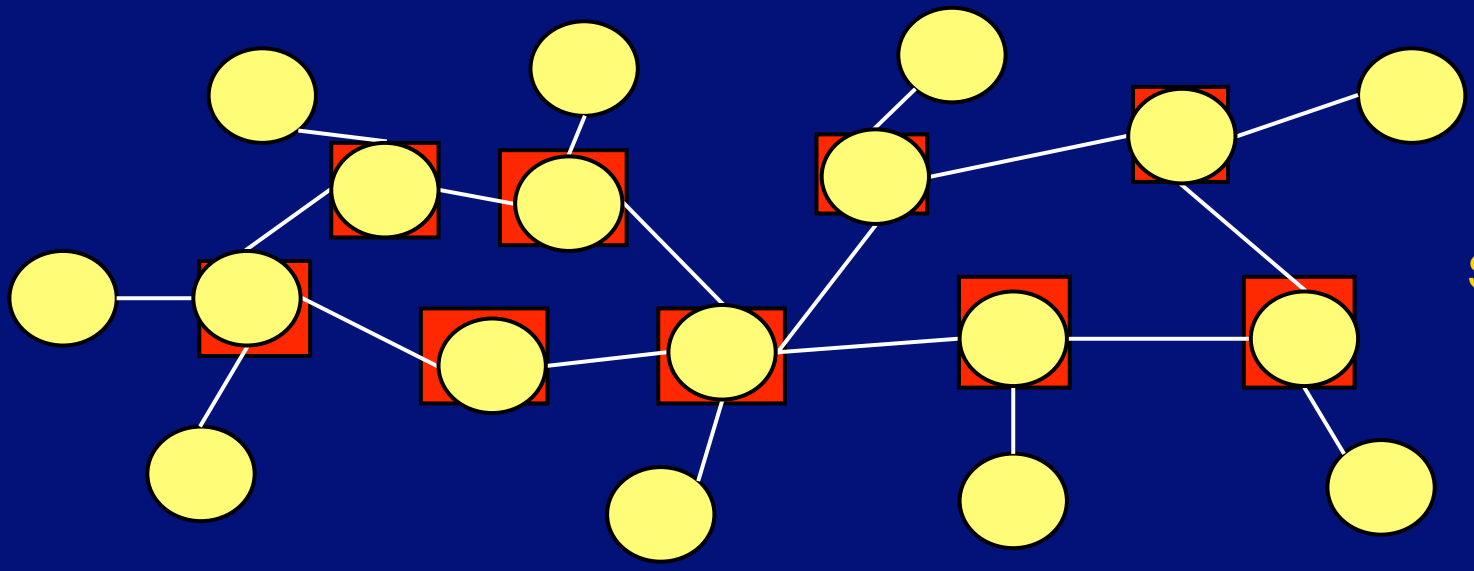
- Create *programmable* network nodes+; standardize the programming model, not the nodes
- Change from Political Tempo (standards) to Technical Tempo (code)
- Balance Usability, Flexibility, Performance and Security

*"SwitchWare: Accelerating Network Evolution (White Paper)", 1996
+ "Softnet – Packet Radio in Sweden", J. Zander, Proc. ARRL, 1981

Active Networks enable new distributed systems



Smart Hosts
+
Dumb Switches
are limited



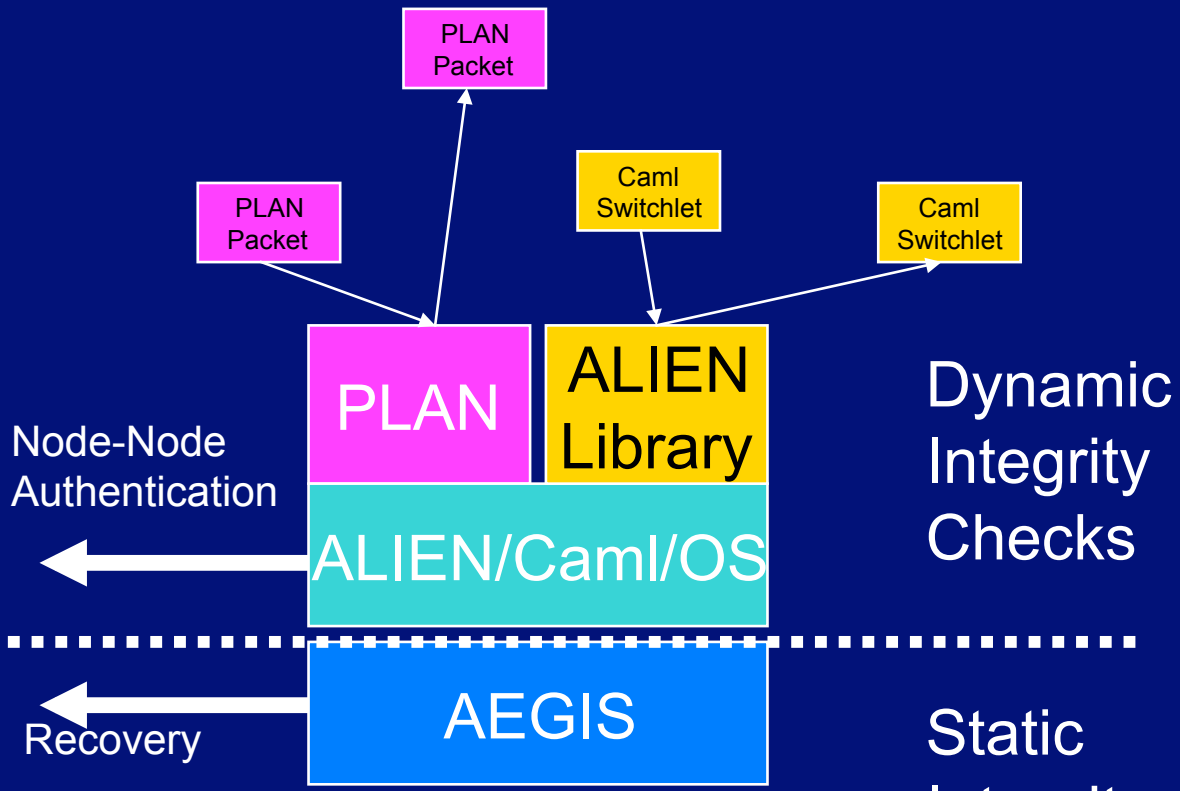
Smart Hosts
+
Smart Switches
are not limited

SwitchWare* Approach

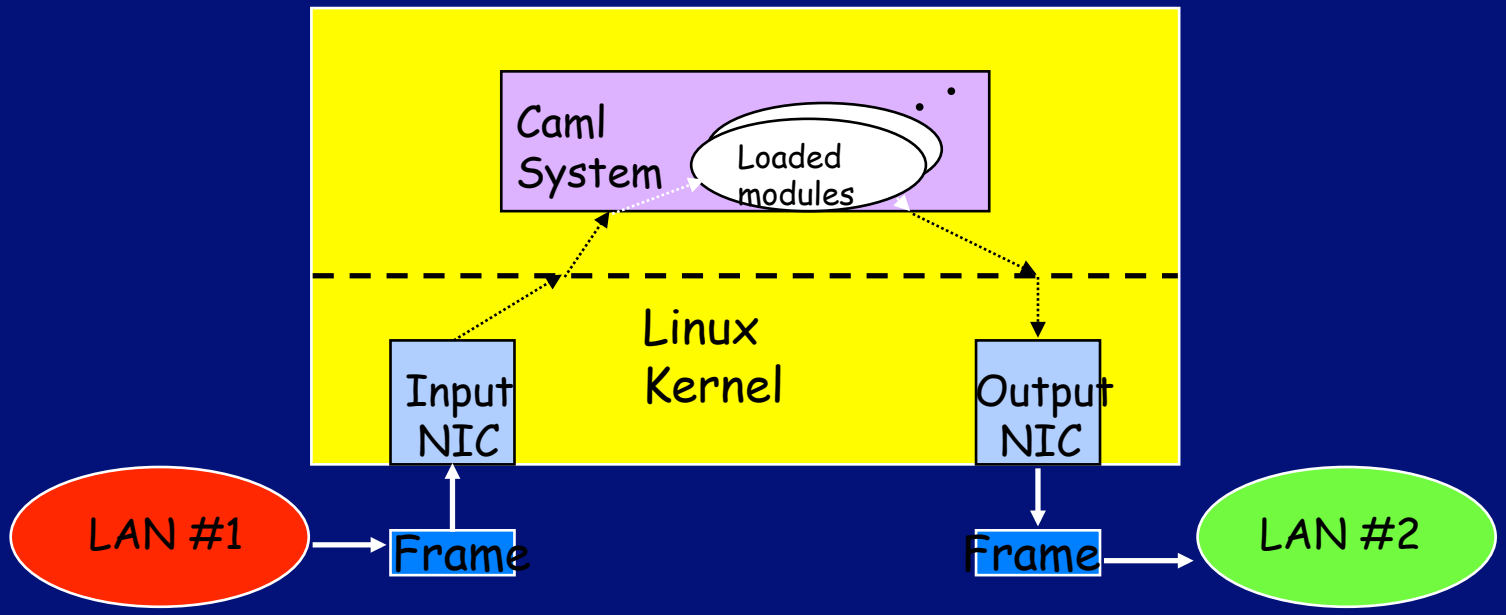
- Modern Programming Language technology (CAML) can help with safety and security+, maybe even performance
- Build flexible node executing programs written in such languages
- Use language mechanisms to restrict programs for *safe multiplexing of nodes* in a network

* "The SwitchWare Active Network Architecture", Alexander, et al., IEEE Network, May/June 1998
+ "A Secure Active Network Environment Architecture: Realization in SwitchWare, Alexander, et al., IEEE Network, May/June 1998

SwitchWare Architecture

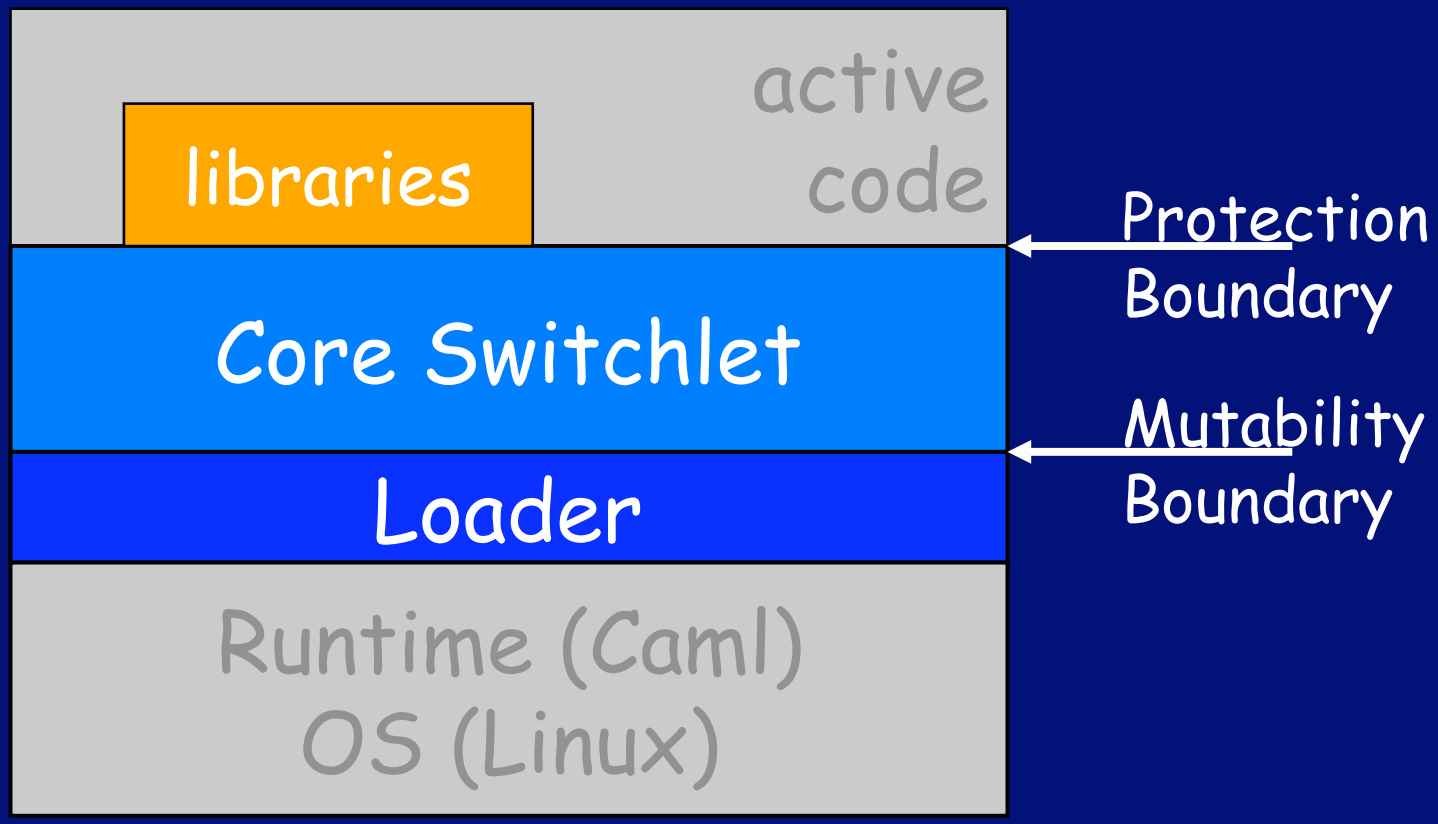


Active Bridging*



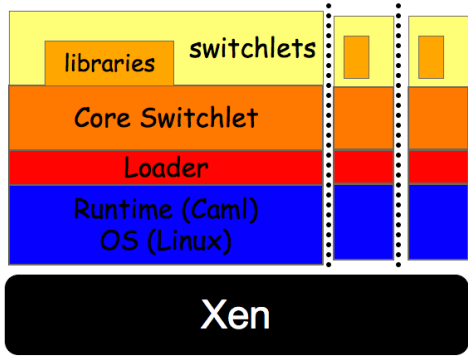
* Alexander, et al., Proc. SIGCOMM 1997

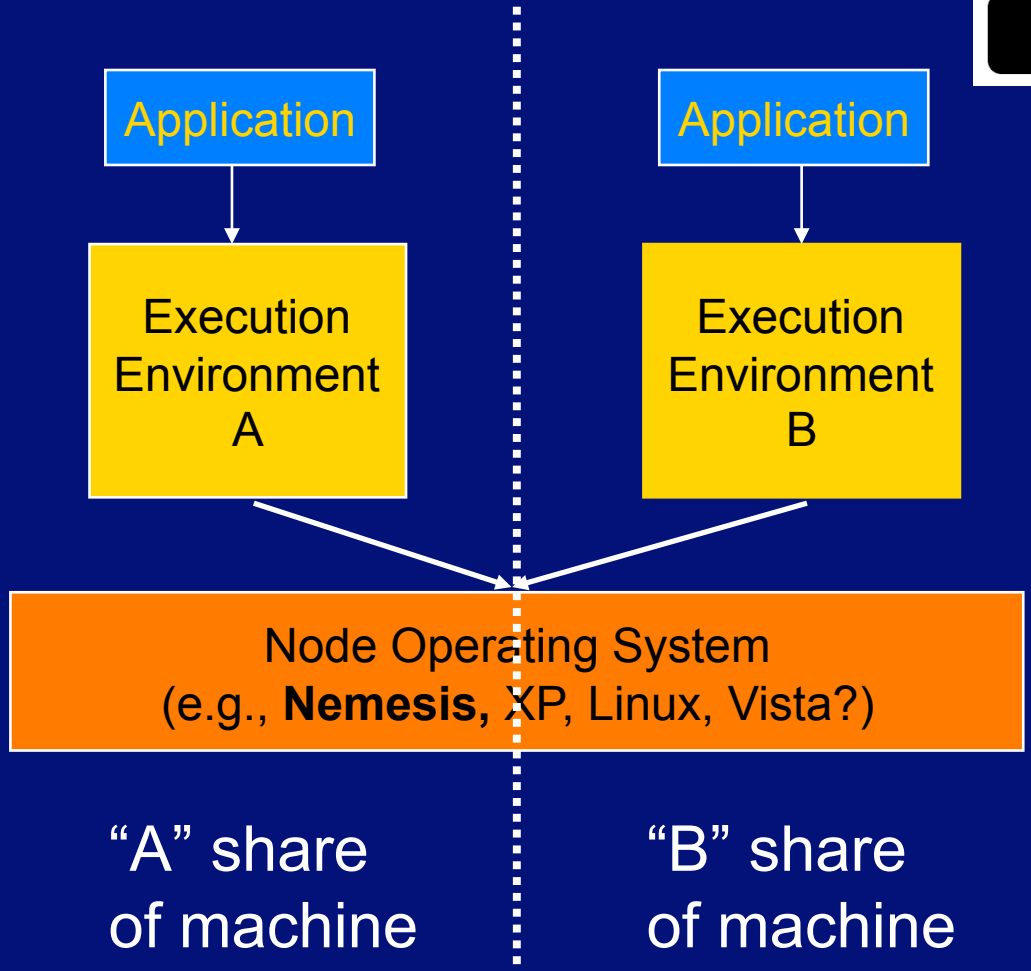
ALIEN Active Loader*



* "The Price of Safety in an Active Network", Alexander, et al., Journal of Communications and Networks, Marrch 2001

Resource Controlled AN Environment (RCANE*):

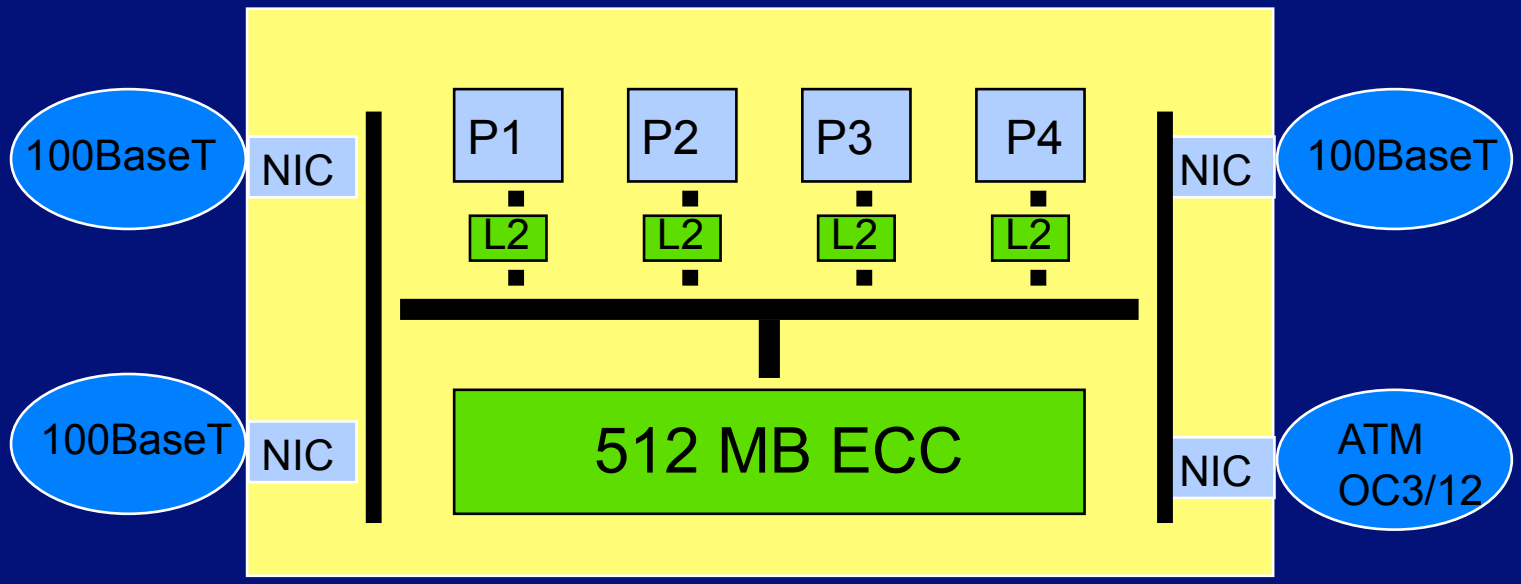




* "The Price of Safety in an Active Network", Alexander, et al., Journal of Communications and Networks, Marrch 2001

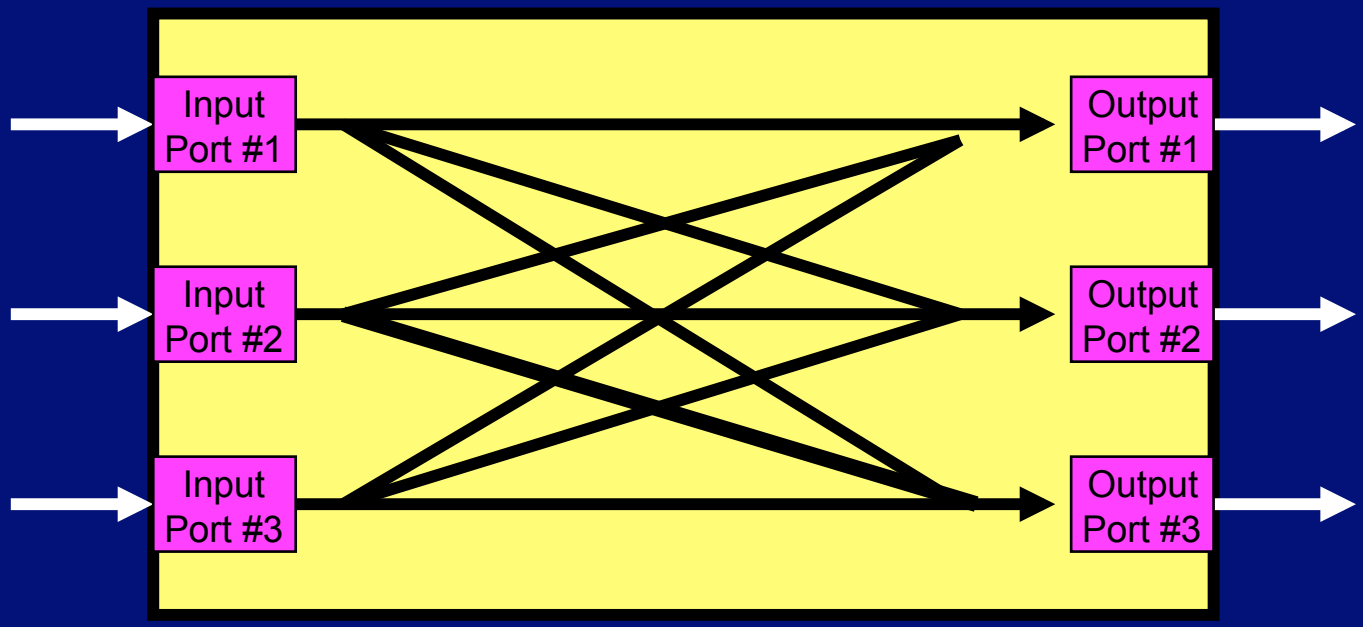
AN node hardware: multi-proc?

- Control or forwarding. Bus unrealistic



A.N. Switch* Architecture

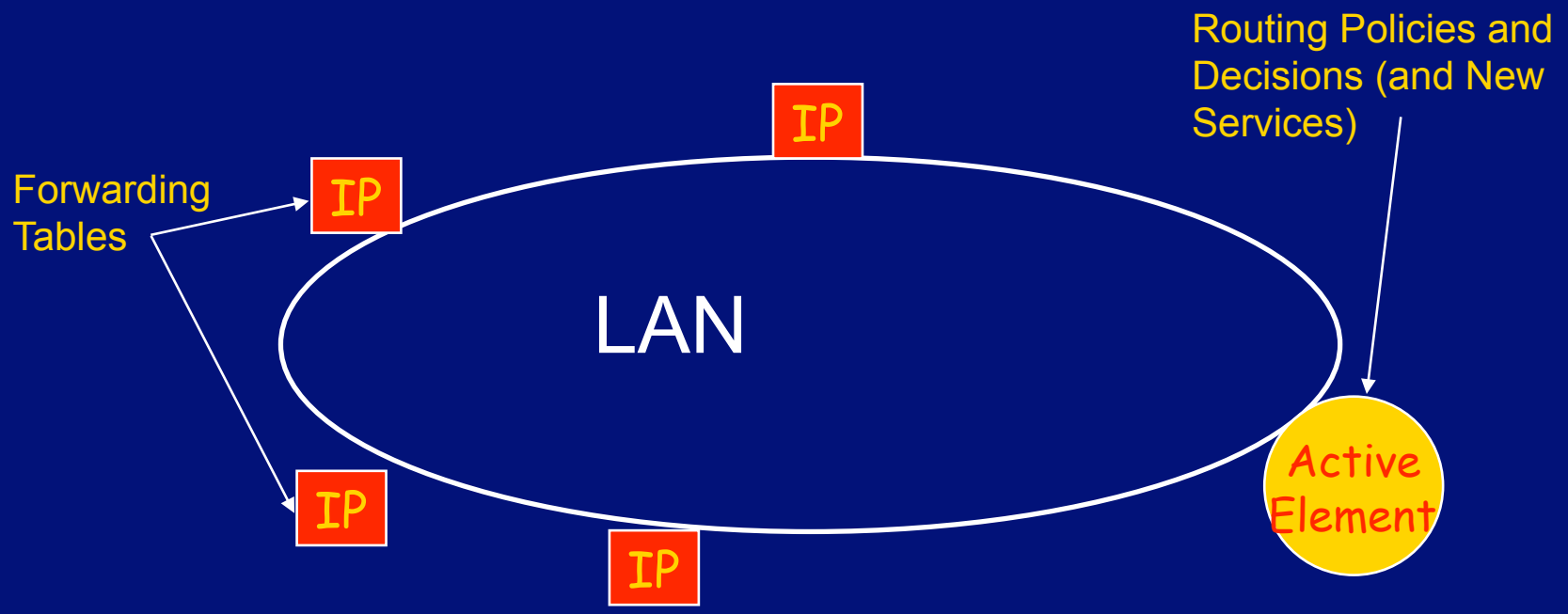
- Active Port Controllers, e.g., Intel IXP (original 1995 design was i960 OPCv2)



* "SwitchWare: Accelerating Network Evolution (White Paper)", 1996

Deployability?: Active Router Control*

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



* "Activating Networks", Smith, Calvert, Murphy, Orman, Peterson, IEEE Network, April 1999

Less ambitious approaches derived from AN are more deployable:

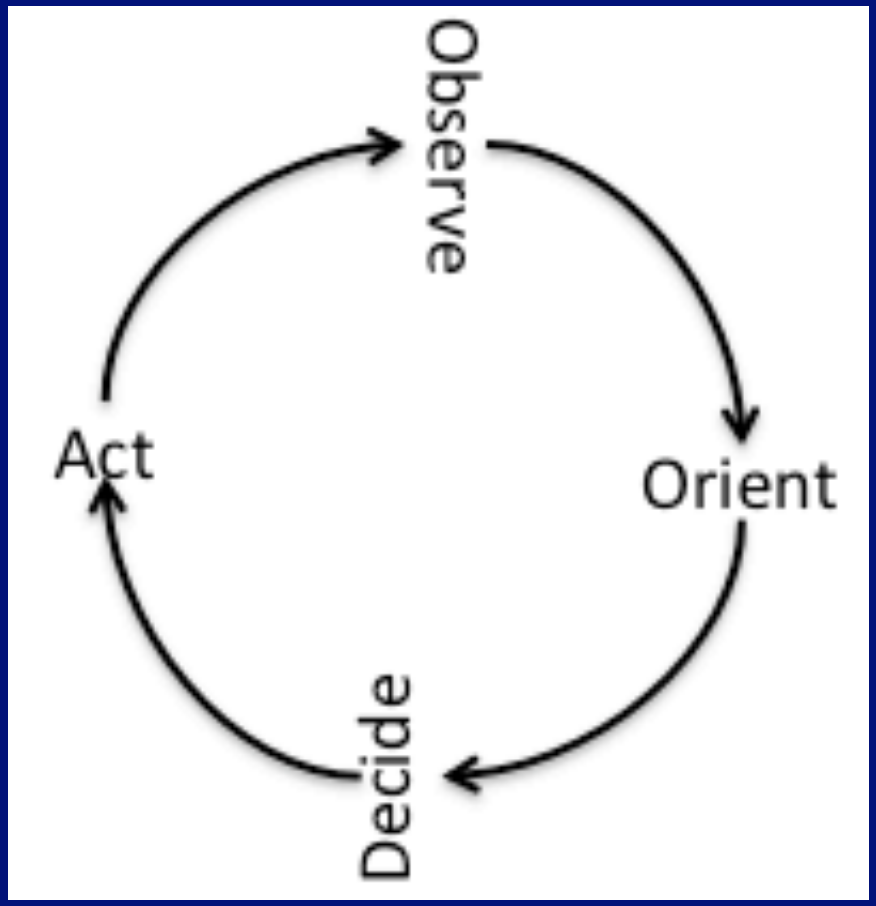
- Overlays (e.g., PlanetLab)
  - →No control of underlays (as noted in *)
- Network Virtualization (e.g., GENI)
  - →RCANE idea, with switch support
- OpenFlow
  - →Active Router Control with flow API

*"SwitchWare: Accelerating Network Evolution (White Paper)", 1996

Distributed Application #1: Content Selection

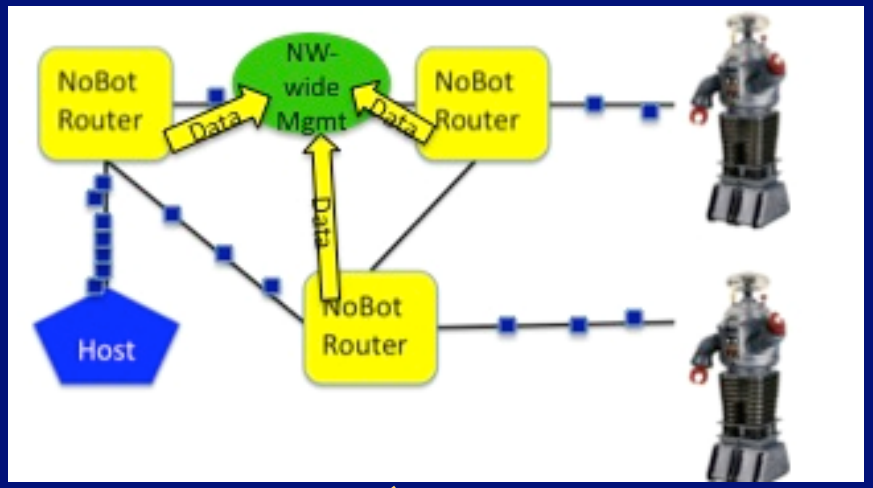
- Nets and computers improving exponentially. Sadly, humans not.
- Active nodes (perhaps content-centric?) contain "delegates"
  - → select information (watching a million cameras at once……)
  - → forward towards you for consumption
  - → your senses extended into the network

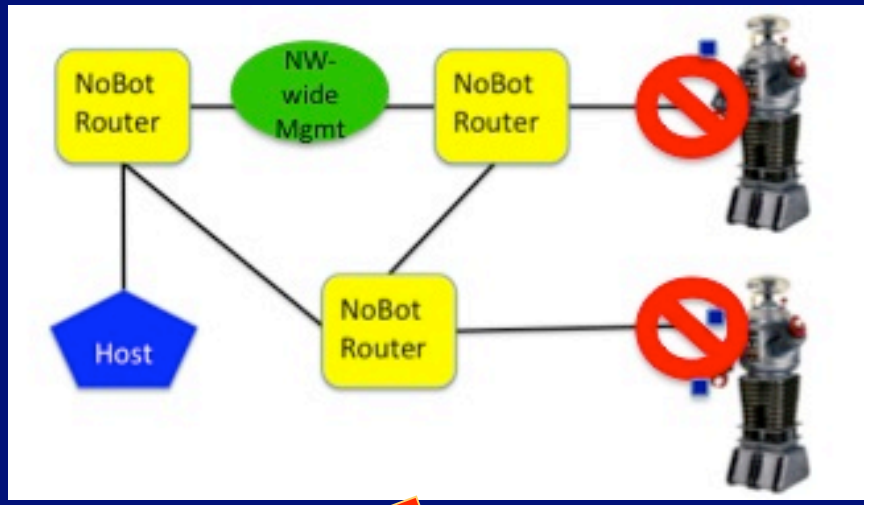
John Boyd's OODA Loop: how to win an arms race

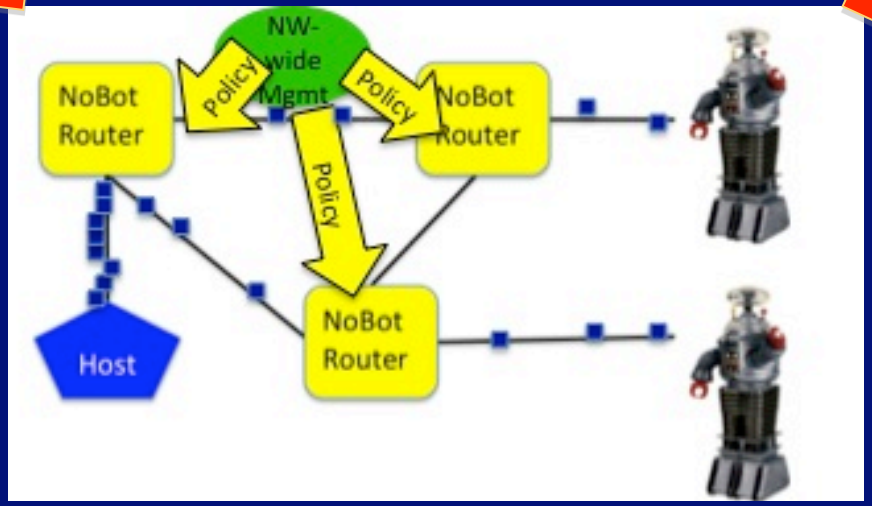


- Faster cycles than adversary: wins
- Technologies should therefore focus on accelerating OODA loop cycles
- Programmability is a key accelerator

Distributed App. #2: Networks Opposing Botnets (NoBot)*







*New work w/ Harvard & Princeton, to be supported by ONR

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+)
- Needed more focus on Active Applications

*"Active Networking: One View of the Past, Present and Future", Smith, Nettles, IEEE Trans. Sys., Man & Cybernetics, Feb. 2004

Acknowledgments:

- SwitchWare and Protocol Boosters were joints project of Penn and Bellcore (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