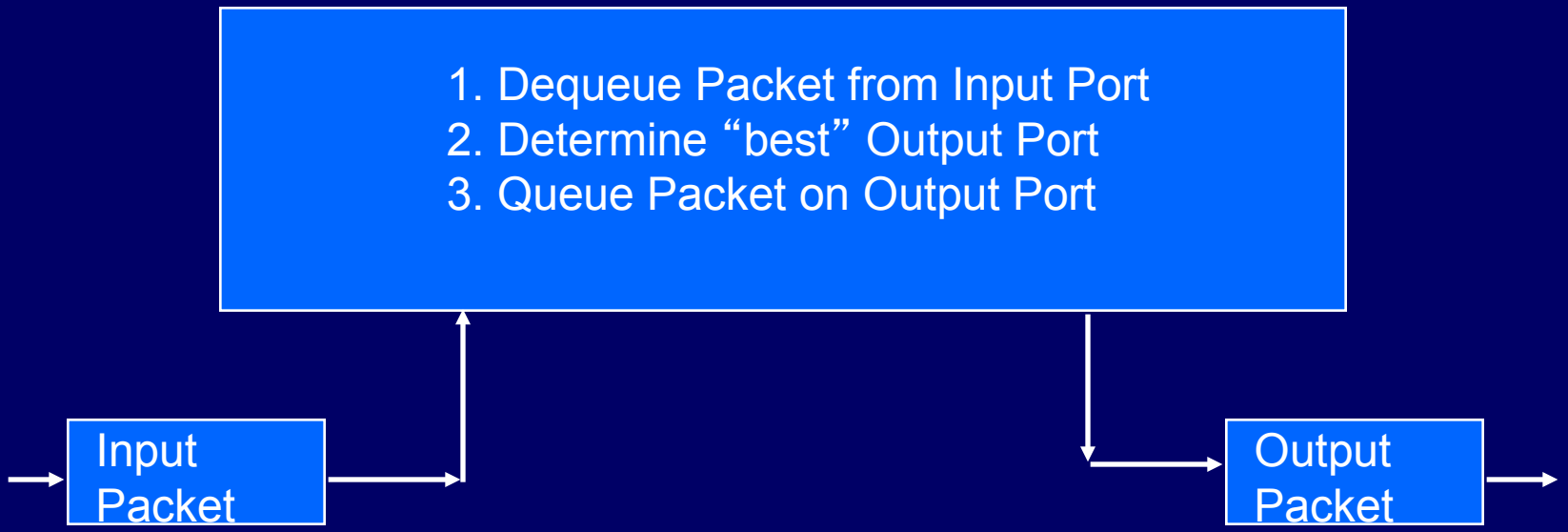
# Active Networks: Myths and Measurements

IWAN '99, Berlin, July 2nd, 1999

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

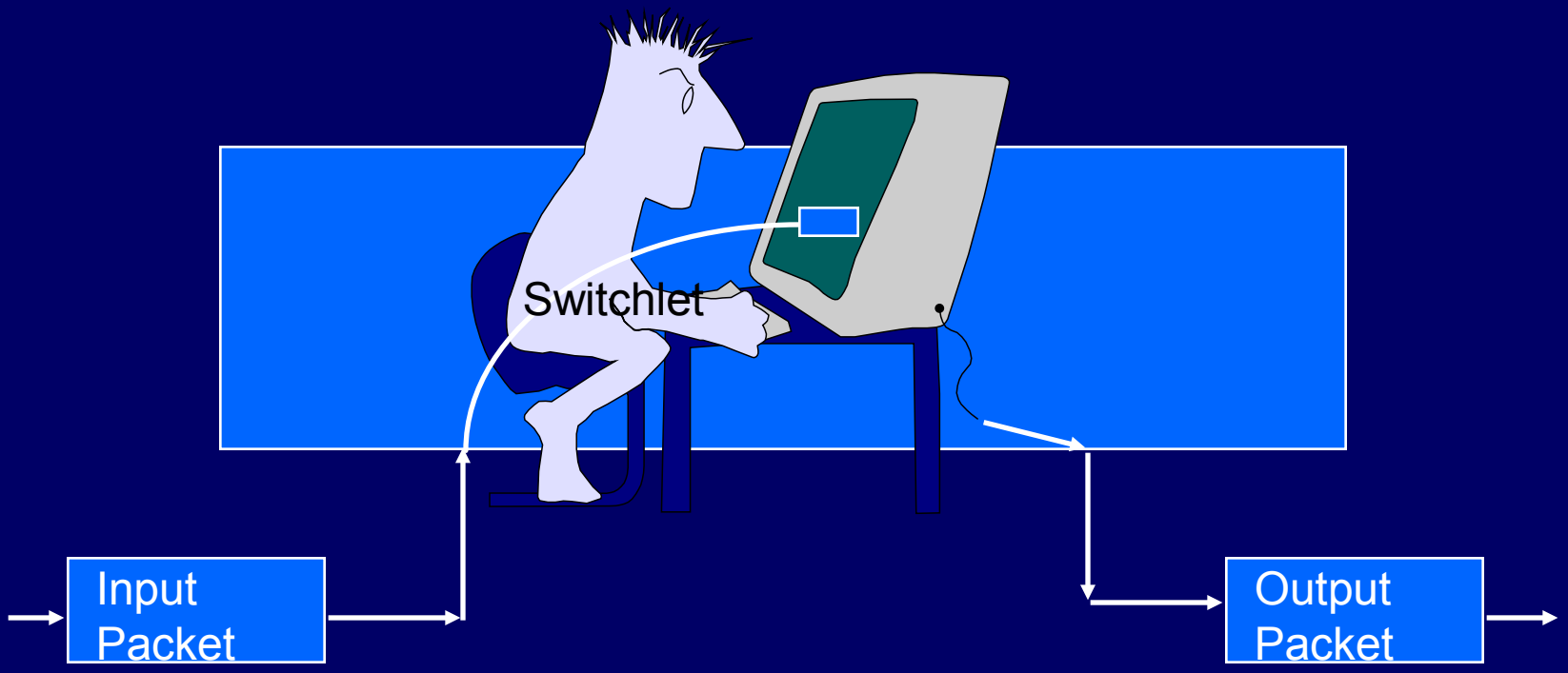
# IP Routing Infrastructure

- Model: Store and Forward



# Active Networking Nodes

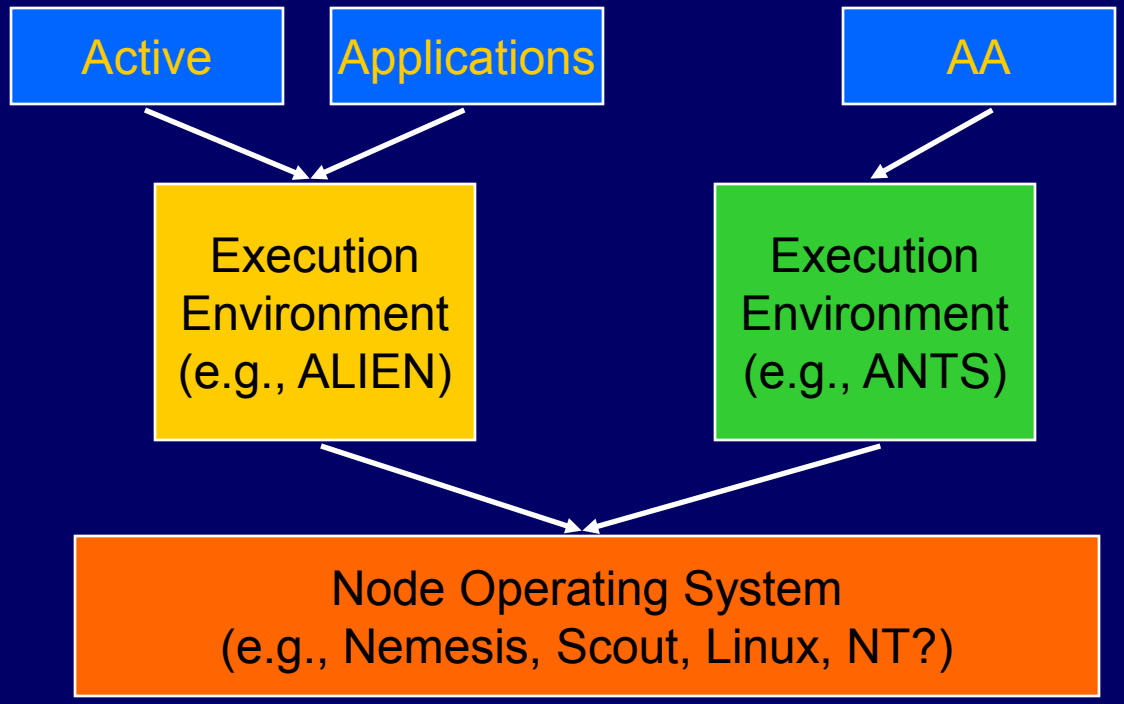
- Store, COMPUTE and Forward!



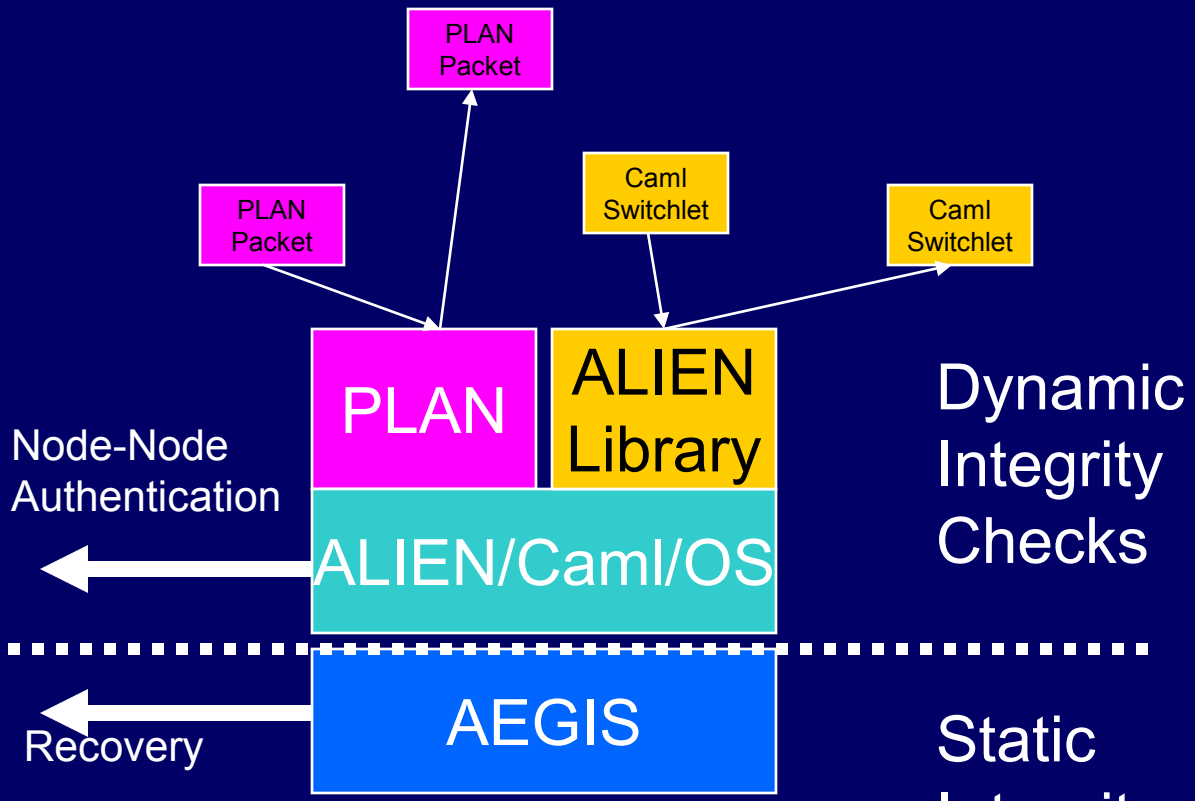
# Three Big Myths

- Active Networks will not perform well
- Active Networks cannot be secured
- Active Networks are an increment on current thinking

# Active Network Model



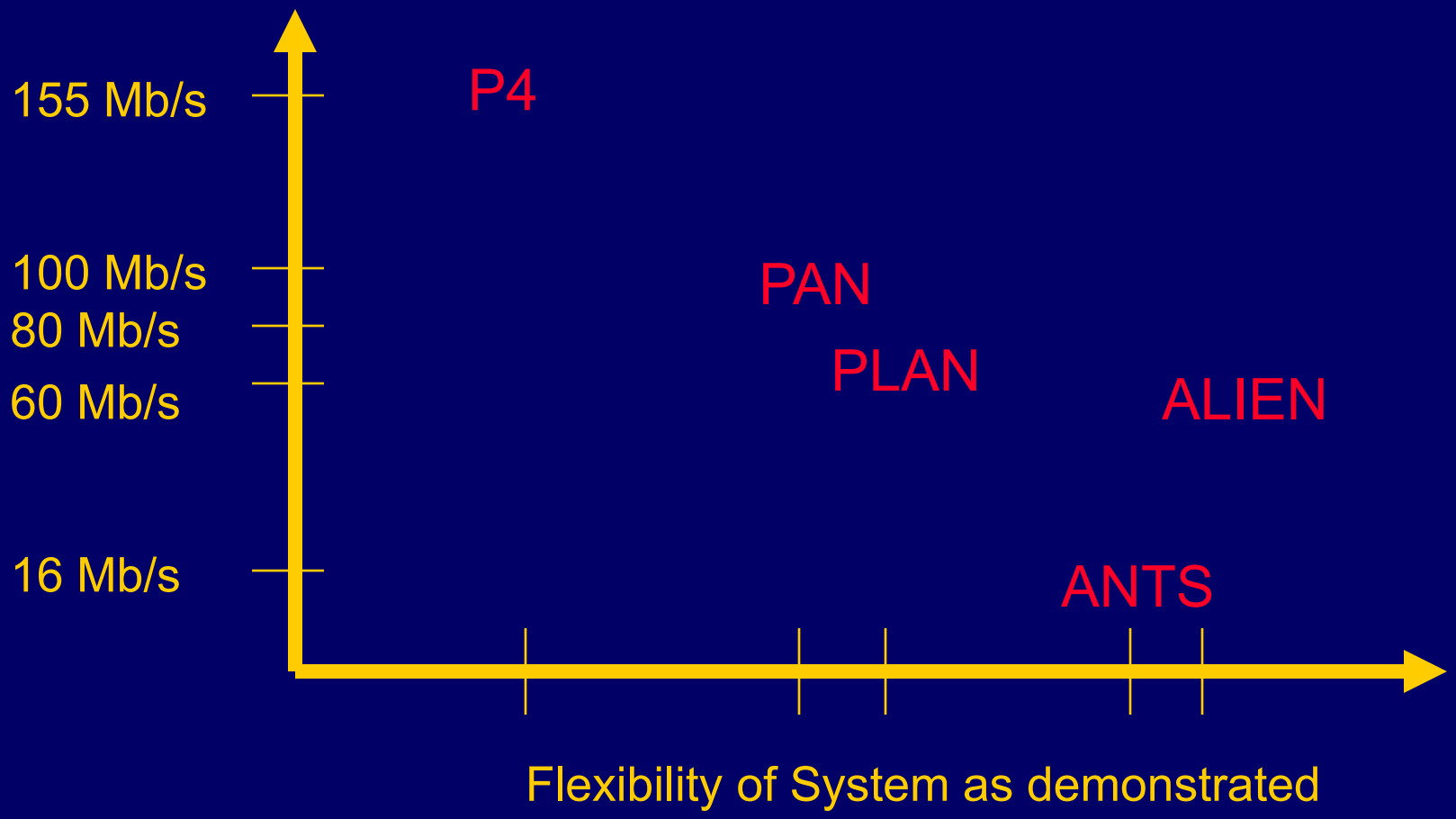
# Example: SwitchWare Architecture



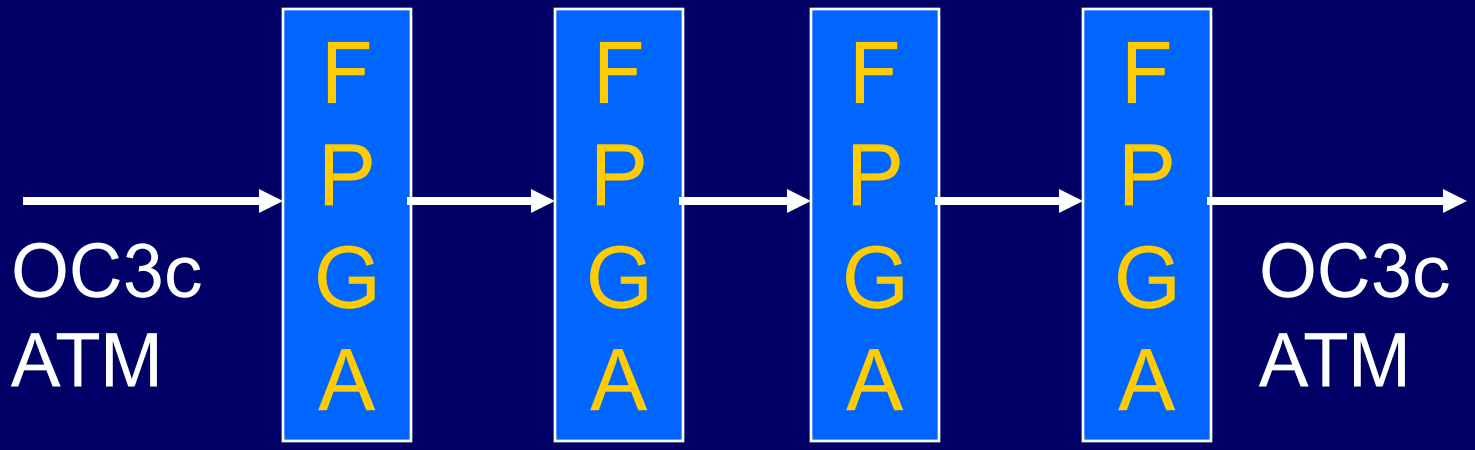
# The Design Space

- Usability *vs.* Flexibility *vs.* Security *vs.* Performance
- There may be unattractive tradeoffs, e.g., Performance and Security may be inversely related! (also Usability?)
- Usability and Flexibility can (mostly) be obtained with a general-purpose language such as Java, Caml or Forth

# Some Performance Tradeoffs



# The Programmable Protocol Processing Pipeline (P4)

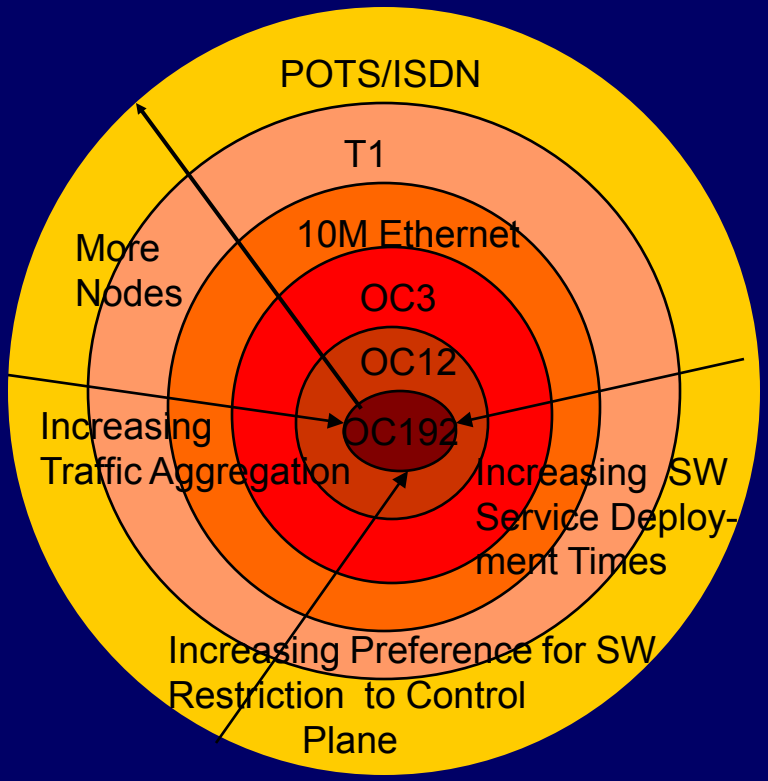



http://www.cis.upenn.edu/~boosters

# The P4 illustrates

- A restricted programming environment
  - Field-programmable gate arrays
- Very high performance; operates at OC-3c line rate with a 19.44Mhz clock
- Easily reaches to 300-400 Mbps with increases in clock rate and word size
- Can be integrated with software EE
  - A high-performance active HW/SW *hybrid*

# Activation potential at various commercially deployed rates:



# Take-Home Lesson Number 1:

- Access points are 14.4-10Mbps
- Peering Points are 1.5Mbps-155Mbps
- Almost all are near the slow ends
- Active Network *Prototypes* cover the entire range!
- This is probably the most sensible place to put value-added services in any case

# Security - not *entirely* there…

- ANTS uses MD5 hashes of programs to identify them at each active node
  - Namespace isolation
  - ANTS "virtual machines"
- ALIEN Active Loader
  - Namespace control with "module thinning"
  - Extend to net with cryptography (at some performance cost)

# *But* no worse than the Internet…

- Secure Active Network Environment
  - AEGIS Secure Bootstrap (EE integrity)
  - Node-node authentication
- Packet Language for Active Networks
  - Restricted "safe" base PLAN language
  - Controlled Access to Active Extensions

# *And* long-term, possibly better!

- Resource Controlled Active Net Environment (RCANE)
  - EEs/Caml on Nemesis => RCANE
  - Thwarts Denial-of-Service
- Research Underway to Specify Global Policy and translate to Local Actions
  - STRONGMAN trust management compiler
  - Netscript global firewalls

# Take-Home Lesson Number 2:

- Greater complexity of AN architecture, and programmability, inspires fear
- But it also stimulates *designed-in* security
- AEGIS and RCANE provide more broadly applicable results
- Programmability: from *nodes* to *nets!*

# Physics and Networks

- Speed of light limits propagation delay
- Bandwidth is increasing exponentially, and therefore bandwidth*delay
- How do we control networks?
  - Round-trip time paced control?
- Require network-embedded control!

# Biology and Networks

- We can probably handle 50 Mbps input
- Is that all we need? No!
- Want to find best of 10,000,000 video streams occurring simultaneously
  - finding
  - selecting
  - focus
- Network as Information Appliance!

# Take Home Lesson #3,……

- This isn't about improving TCP 0.0001%
- This isn't about selecting header fields
- It's about integrating networks and computing in a seamless and useful way!

# Three Big Truths

- Active Networks perform well
- Active Networks can be secured
- Active Networks will help address the problems of the future; think big - the past comes for free!

# Acknowledgments:


- All Penn work and most other work supported by DARPA ITO.
- Collaborators: Alexander, Arbaugh, Farber, Feldmeier, Gunter, Hadzic, Hicks, Keromytis, Marcus, McAuley, Menage, Moore, Nettles, Segal and Sincoskie…
- Hewlett-Packard, Intel and 3Com