The Active Network Design Space

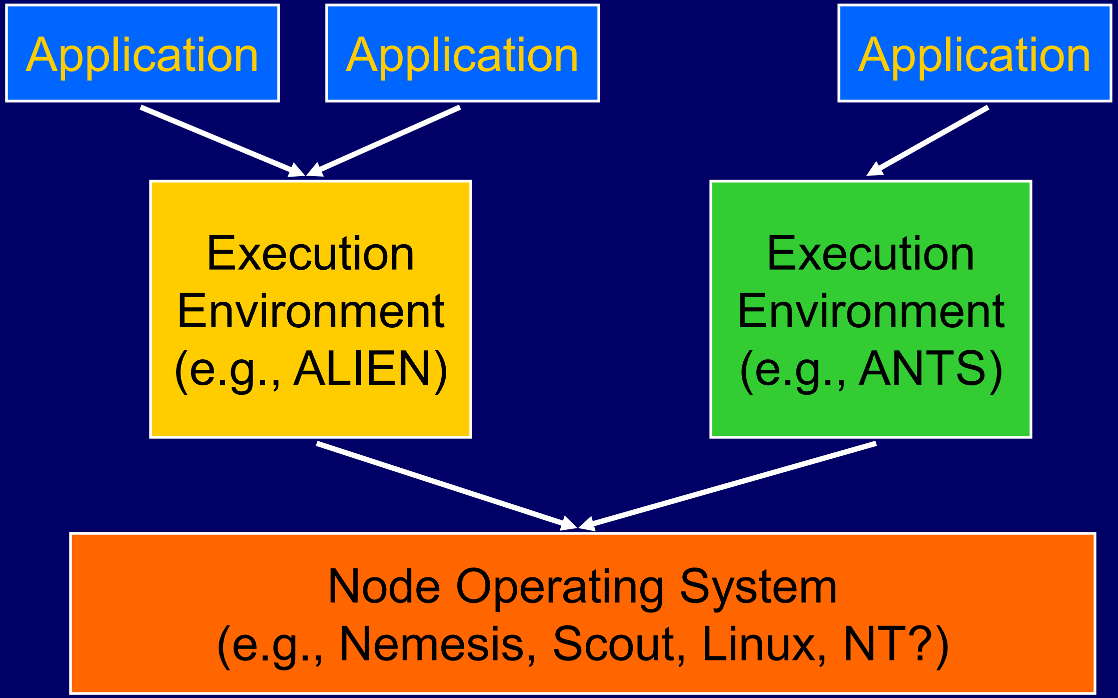
Mini-conference, Paris, FRANCE
May 17th, 2000

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

Outline: 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

Active Network Architecture



Application

Application

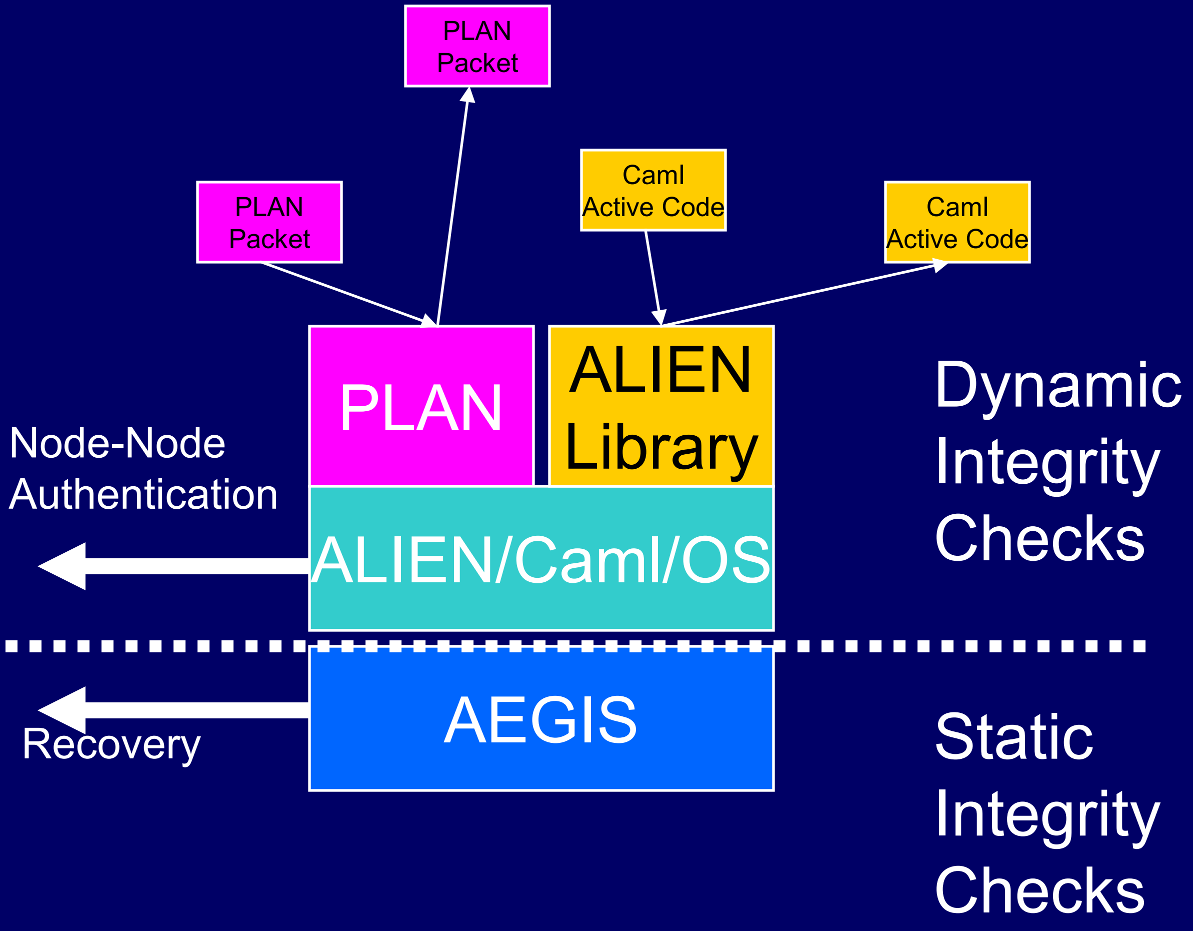
Application

Execution Environment (e.g., ALIEN)

Execution Environment (e.g., ANTS)

Node Operating System
(e.g., Nemesis, Scout, Linux, NT?)

Example: SwitchWare Architecture



The ALIEN Approach

- Achieved by *restricting* a general computing model
- Realized in ALIEN, an active loader for Caml
  - →General computing model
  - →Interface to OS
  - →Interface to active code
- Only privileged portions of the system can directly access shared resources

Decisions in the Design Space

- Usability *vs.* Flexibility *vs.* Security *vs.* Performance
- A General-Purpose Language gets the first two for free; other two are hard!
- Domain-specific Languages (such as PLAN) may achieve different tradeoffs

The ALIEN Active Loader

□D. Scott Alexander

□CAML runtime

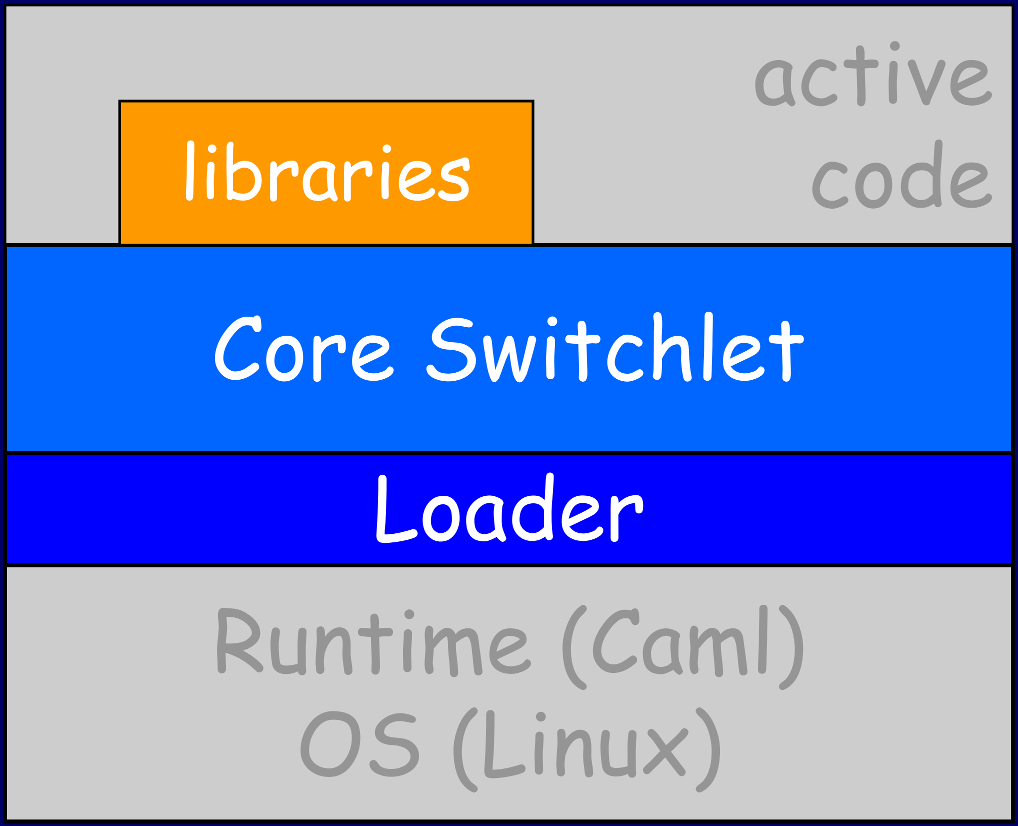
□CAML capsules restricted via module thinning

□Digitally-signed certificates for remote accesses to resources

□Will use for detailed case study

ALIEN in an Active Element

□Three layer architecture



Implementation of Active Code

- □Active Extensions
  - →Loaded from disk or network (TFTP)
  - →We use queues for communication
  - →Could use upcalls…
    - ✦Security?
  - →…or blocking downcalls
- □Active Packets
  - →ANEP encapsulated (over UDP or link layer)
  - →Can use SANE for security
  - →Linker/ procedure call for communications

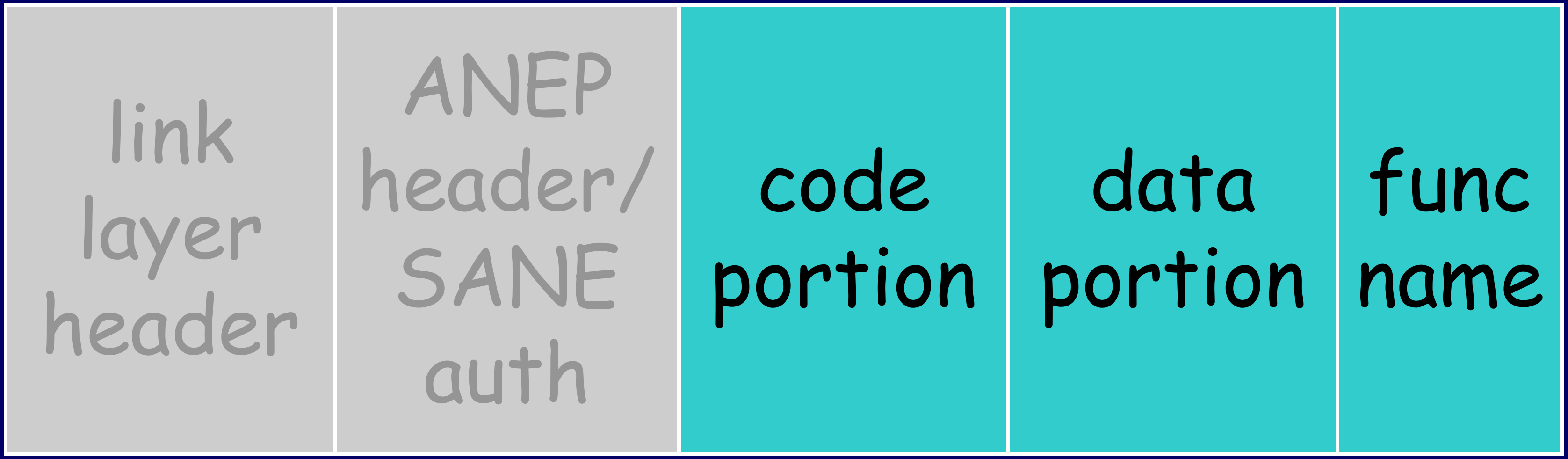
Active Packets in ALIEN

□If ANEP header indicates ALIEN

→SANE processing as part of ANEP

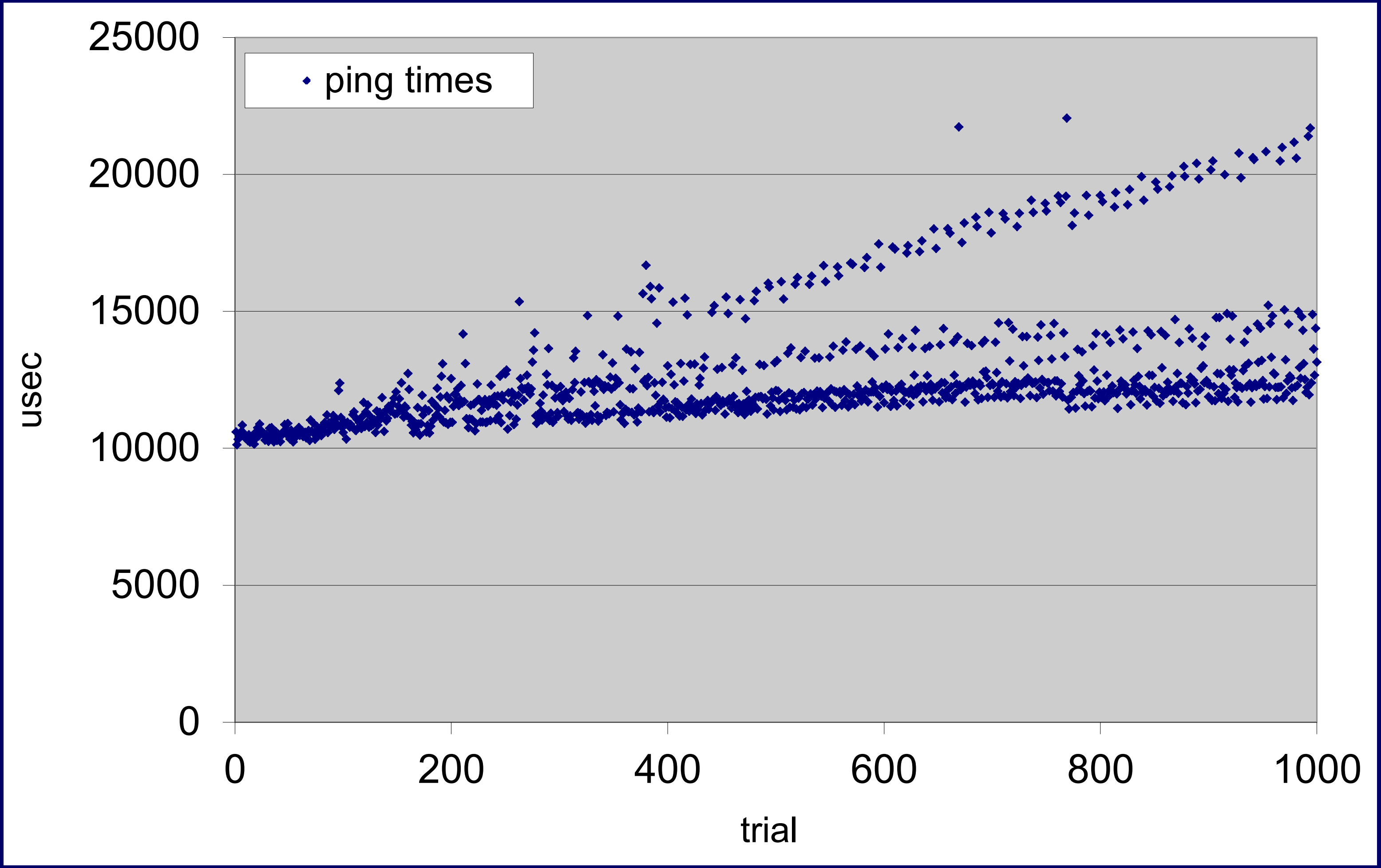
→Code portion is loaded

→*func* is called with code, data, and func name as arguments

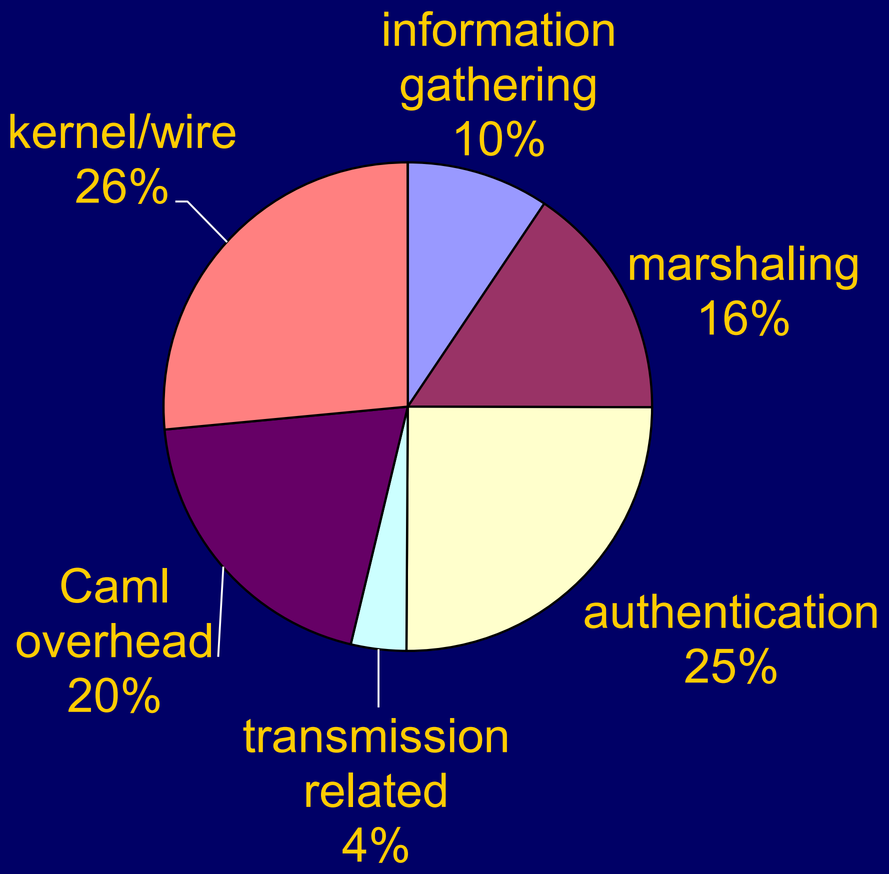


link layer header	ANEP header/ SANE auth	code portion	data portion	func name

saneping Performance



Overall Breakdown of Costs



Major Costs

- □Kernel/Wire (26%, 3078 μs)
  - →Kernel time + transmission time
  - →To avoid
    - ✦Reduce size of packet
    - ✦Reduce or avoid kernel boundary crossing cost
- □Authentication (25%, 2910 μs)
  - →Mostly cost of performing SHA-1 (4 times)

Cryptography is Expensive

- Implemented in C because too slow in Caml
- Times to hash 4MB of data

	bytecode	native
Caml Int32	86.45s	61.99s
Caml int	36.03s	2.48s
C		0.33s

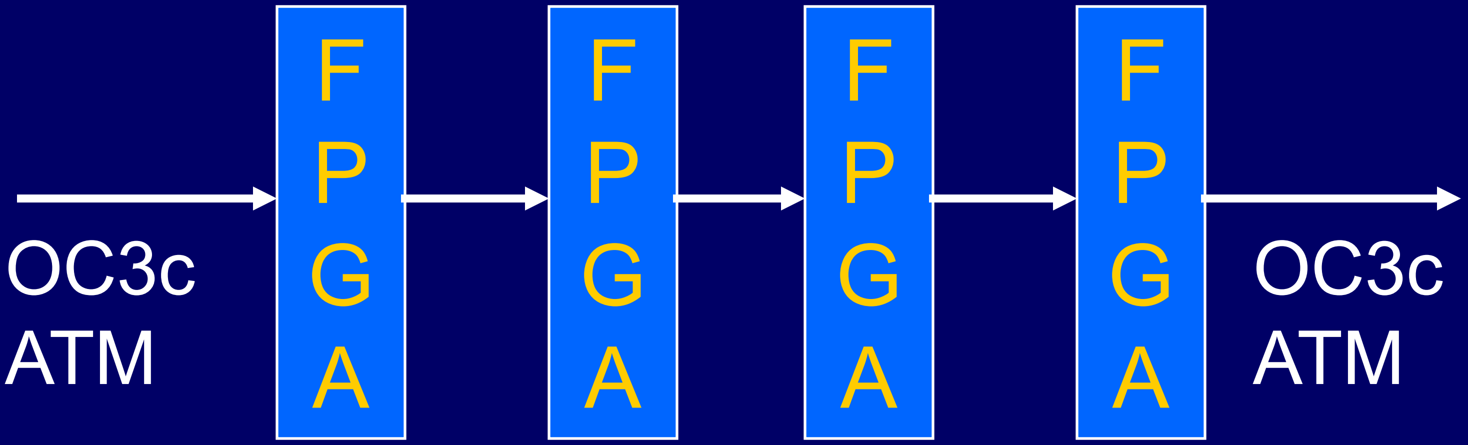
The take-home lesson:

- Must reduce per-packet crypto costs:
  - → Active extension amortizes costs
  - → ANTS caching amortizes costs
  - → Smaller packets (Dense CISC, a la BBN)
- Or, find another way to avoid crypto in the common case…

Packet Language for Active Networks (PLAN)

- Hicks, Kakkar, Moore, Gunter, Nettles
- Capsule-based approach
- CAML runtime
- Highly-restricted domain specific language (a safe “glue” language, like the UNIX shell), extensible via ALIEN
- Active extensions do restricted things

The Programmable Protocol Processing Pipeline (P4)

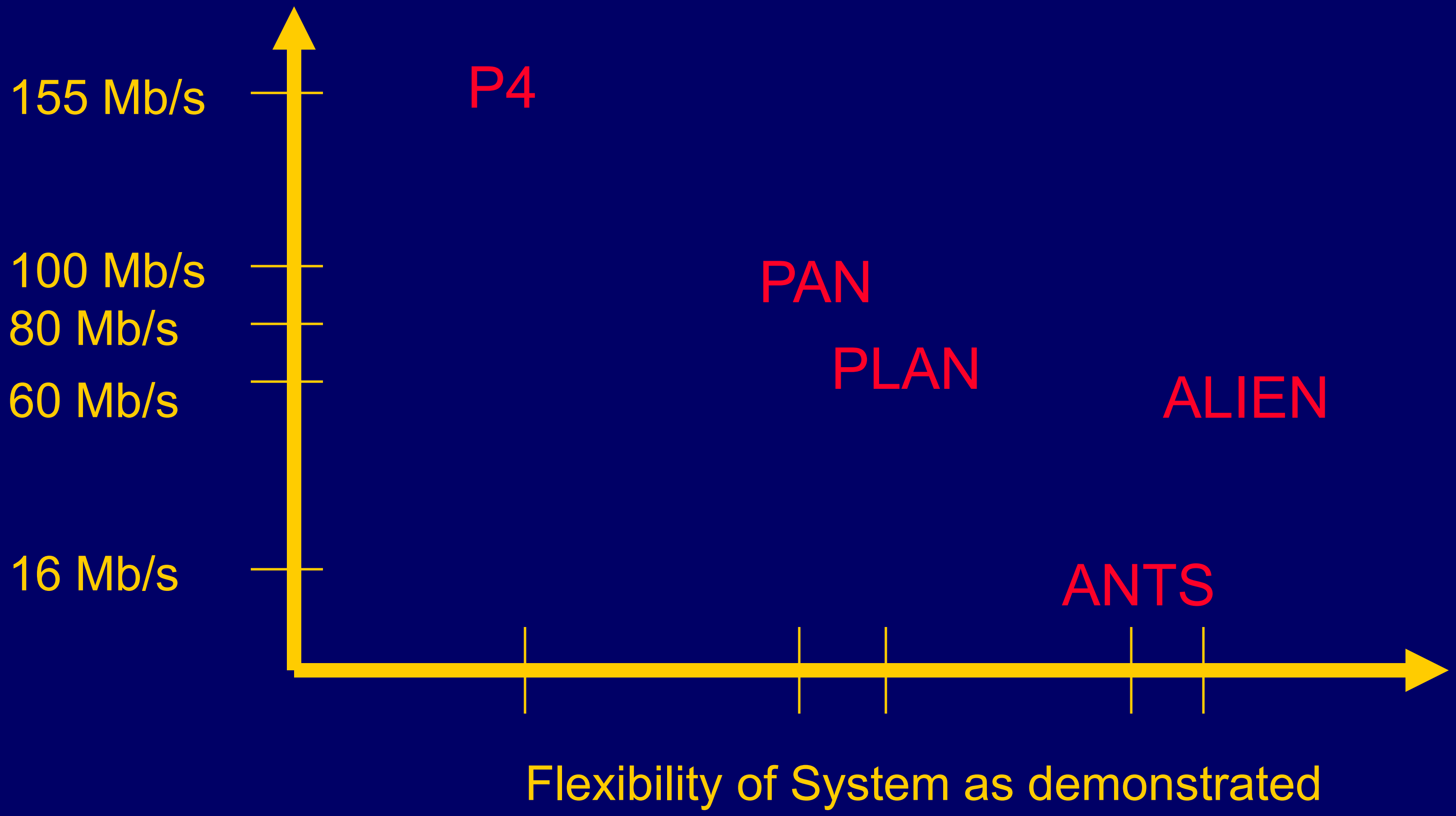


http://www.cis.upenn.edu/~boosters

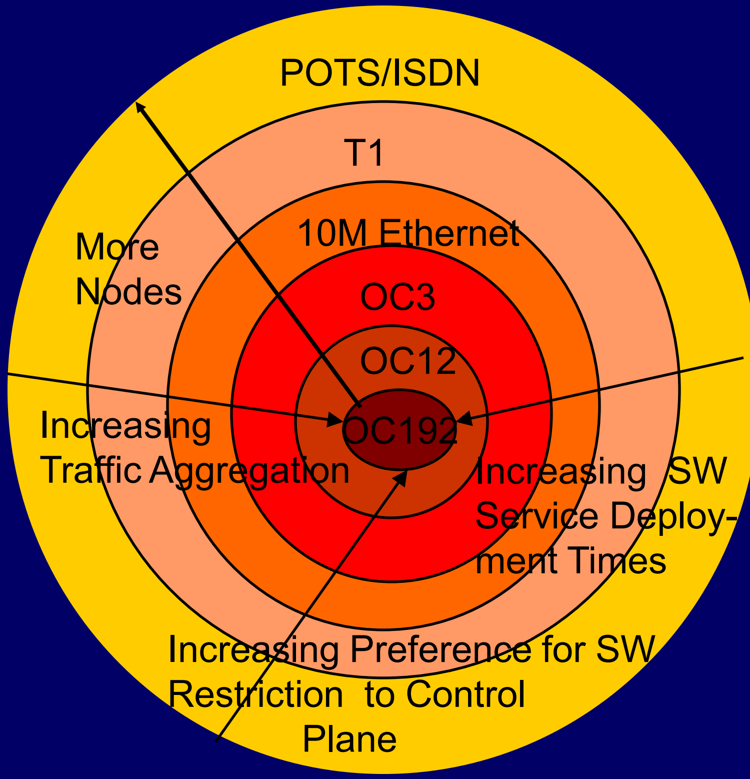
The P4 illustrates

- □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*

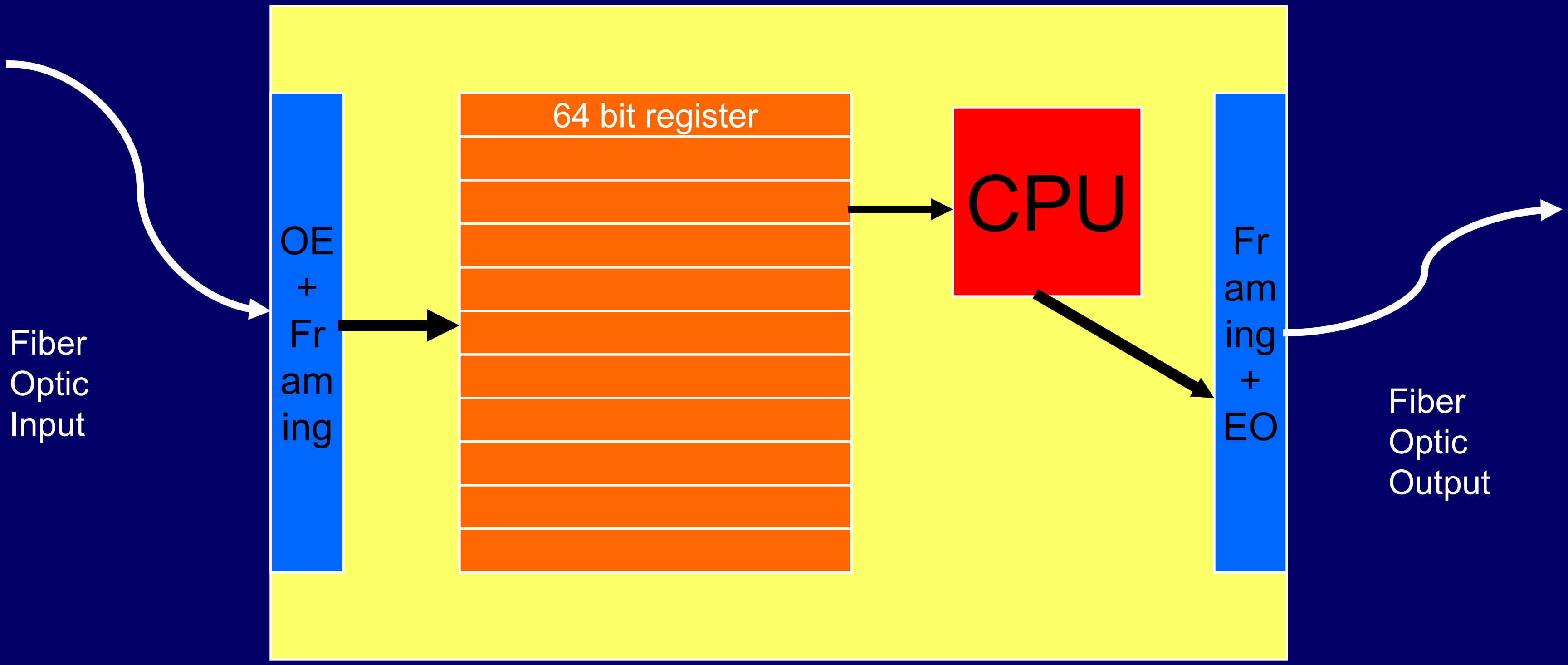
Some Performance Tradeoffs



Activation potential at various current line rates:


POTS/ISDN
T1
10M Ethernet
OC3
OC12
OC192
More Nodes
Increasing Traffic Aggregation
Increasing SW Service Deploy-ment Times
Increasing Preference for SW Restriction to Control Plane

Next Generation: in-Fiber A.N.



Register-Only Media Processor (ROMP)

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