A Secure PLAN (extended version)

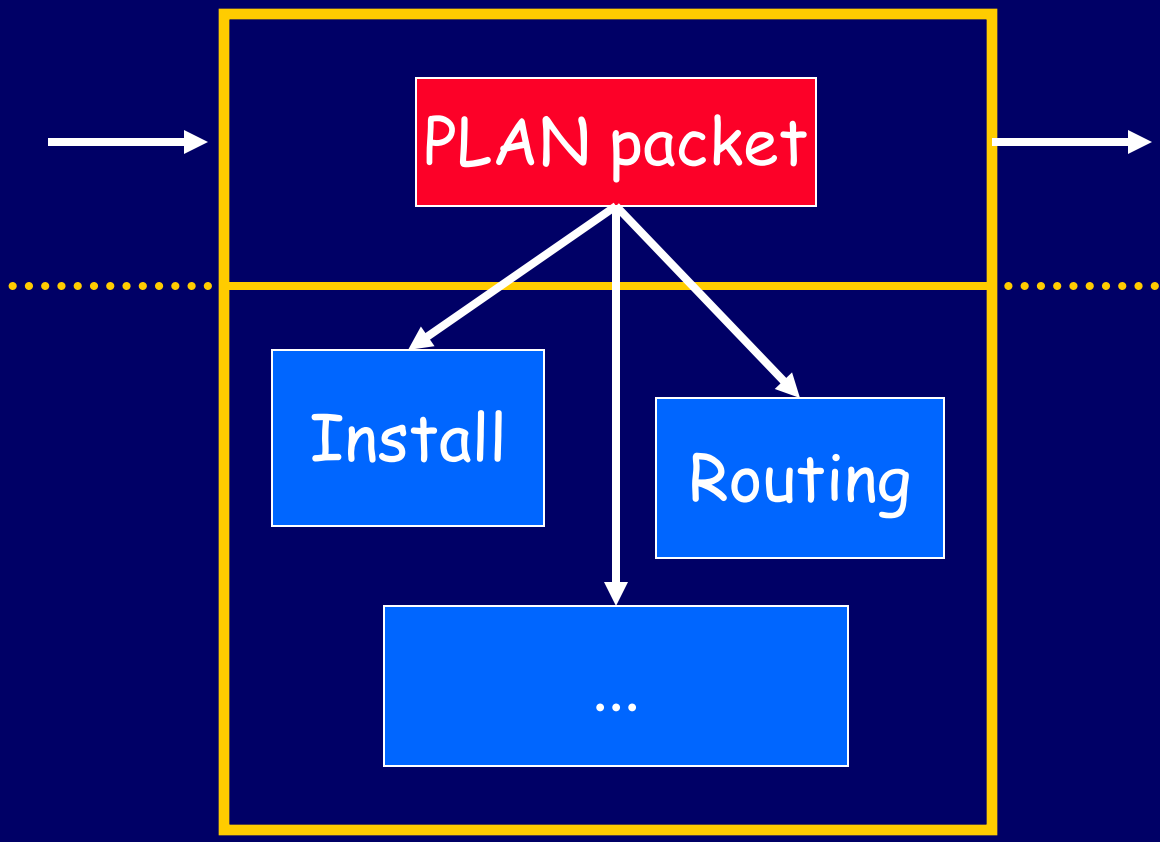
Michael Hicks, Cornell University
Angelos D. Keromytis, Columbia University
Jonathan M. Smith, U. Penn

DARPA Active Networks Conference and Exposition (DANCE), San Francisco CA, May 29-30, 2002

Nugget to take home:

- A careful separation between language protections for active packets and cryptography-based authorizations for active extensions can lead to a system with flexibility, performance *and* security
- http://www.cis.upenn.edu/~switchware/PLAN

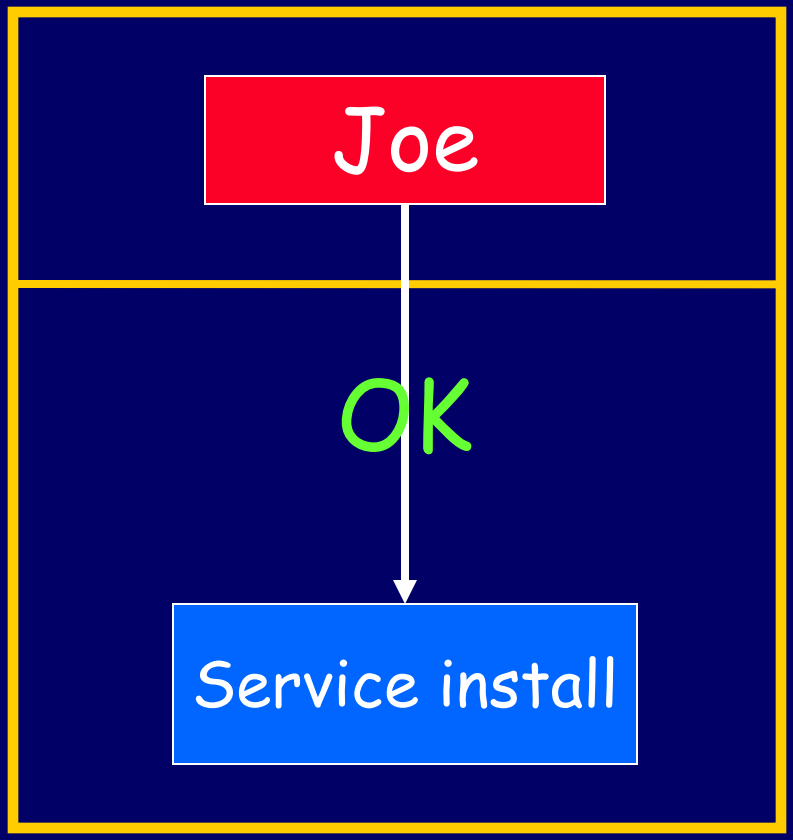
PLANet: 2-level Architecture



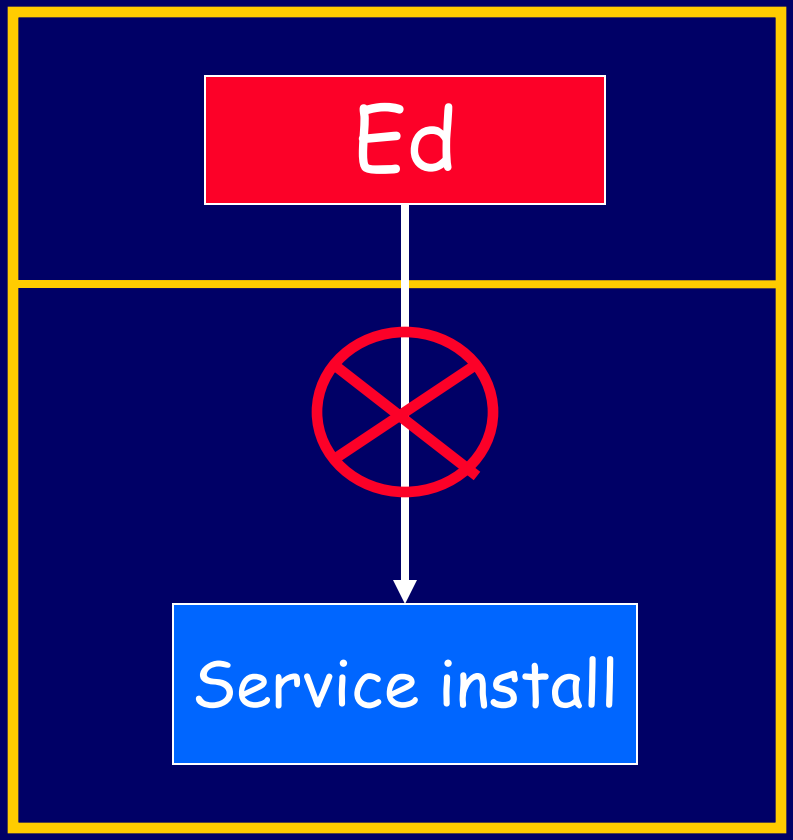
PLAN :
- safe
- in packet

Services:
- general
- on router

Trust Management

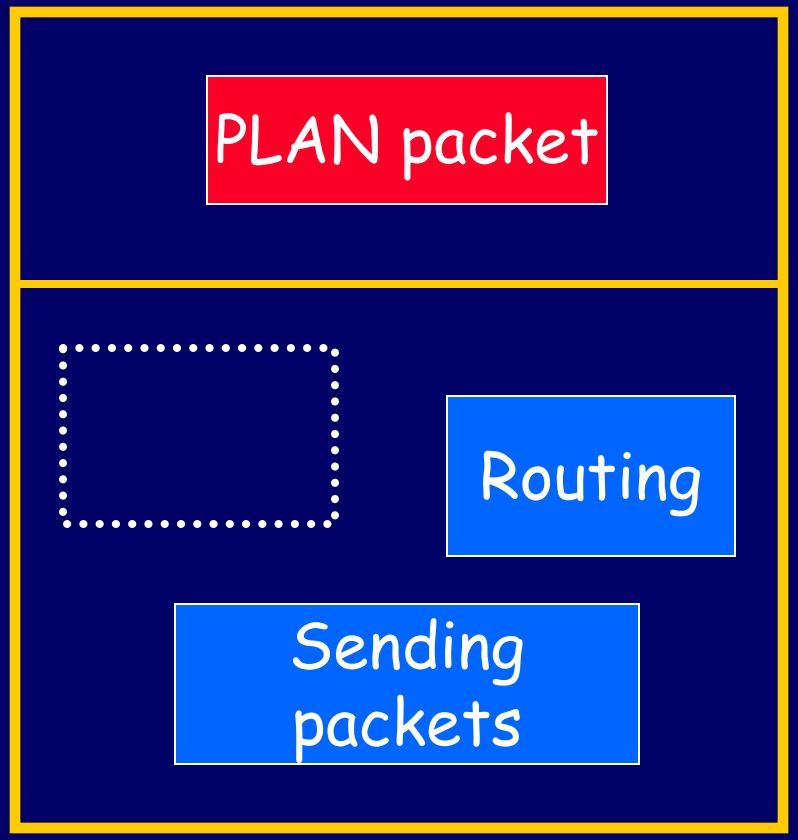


Trust Management



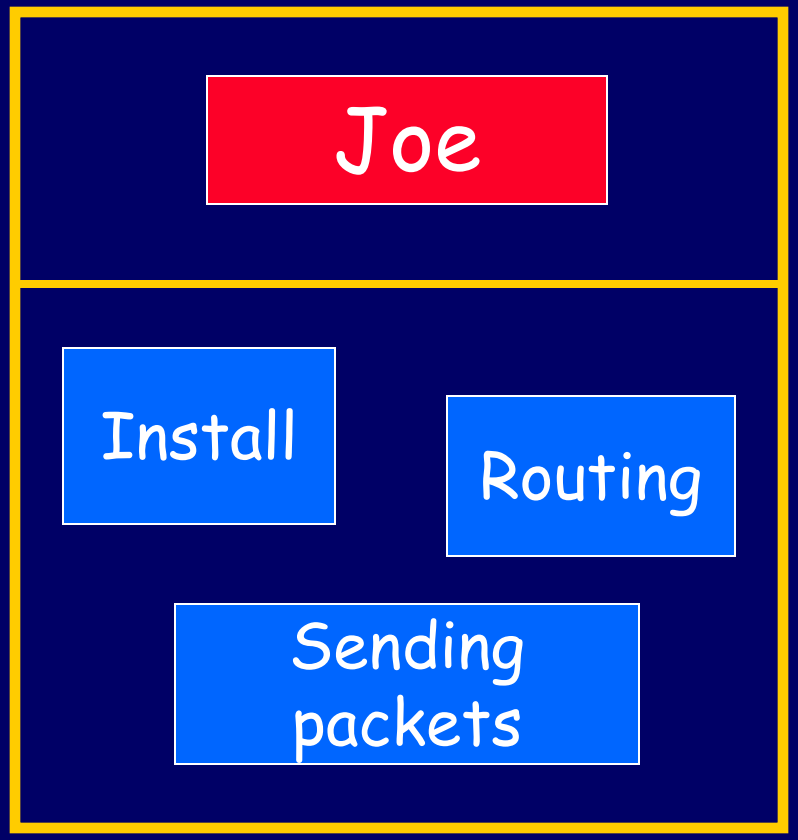
not allowed

Form of Service Policies: Access



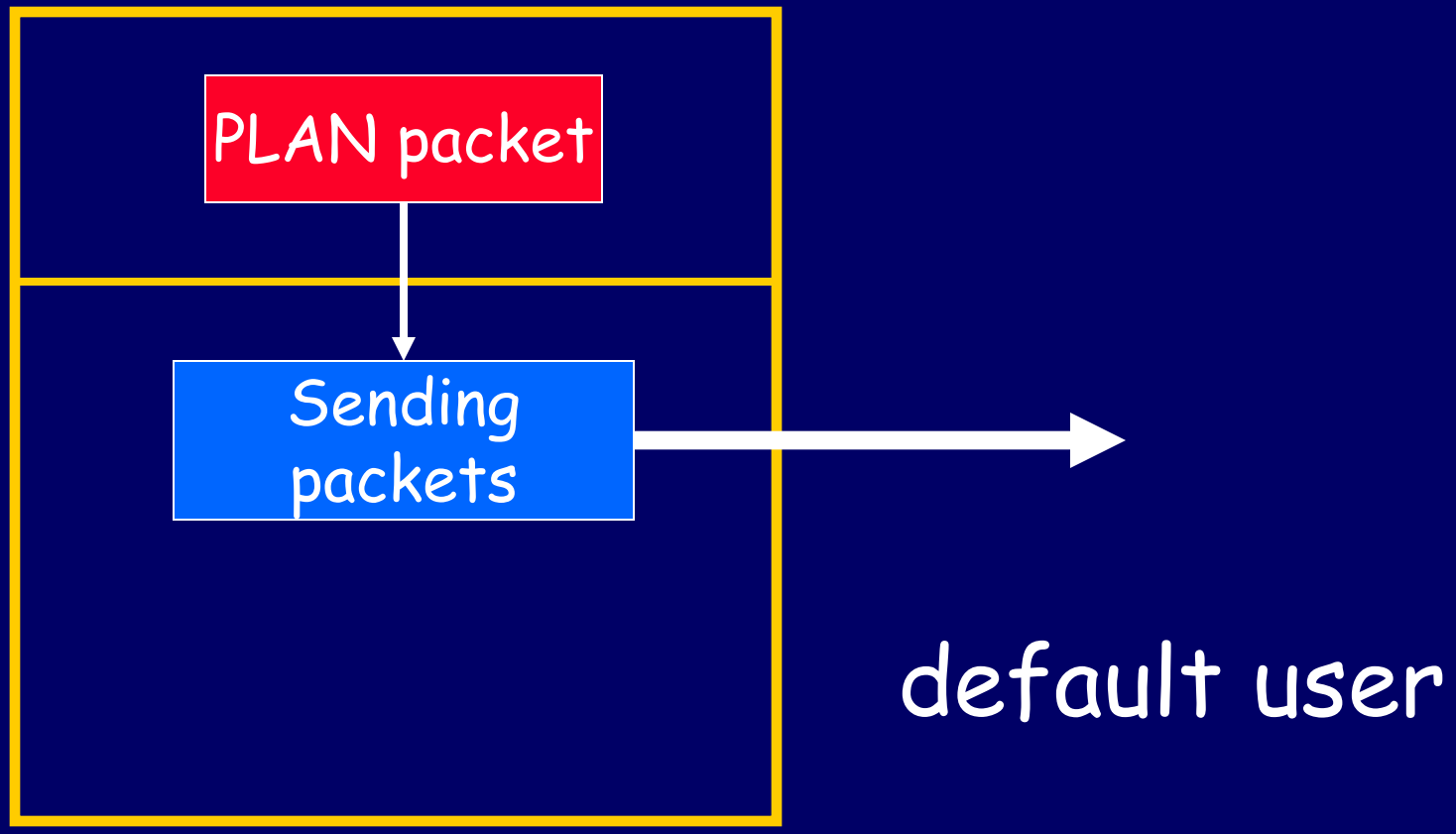
default user

Form of Service Policies: Access



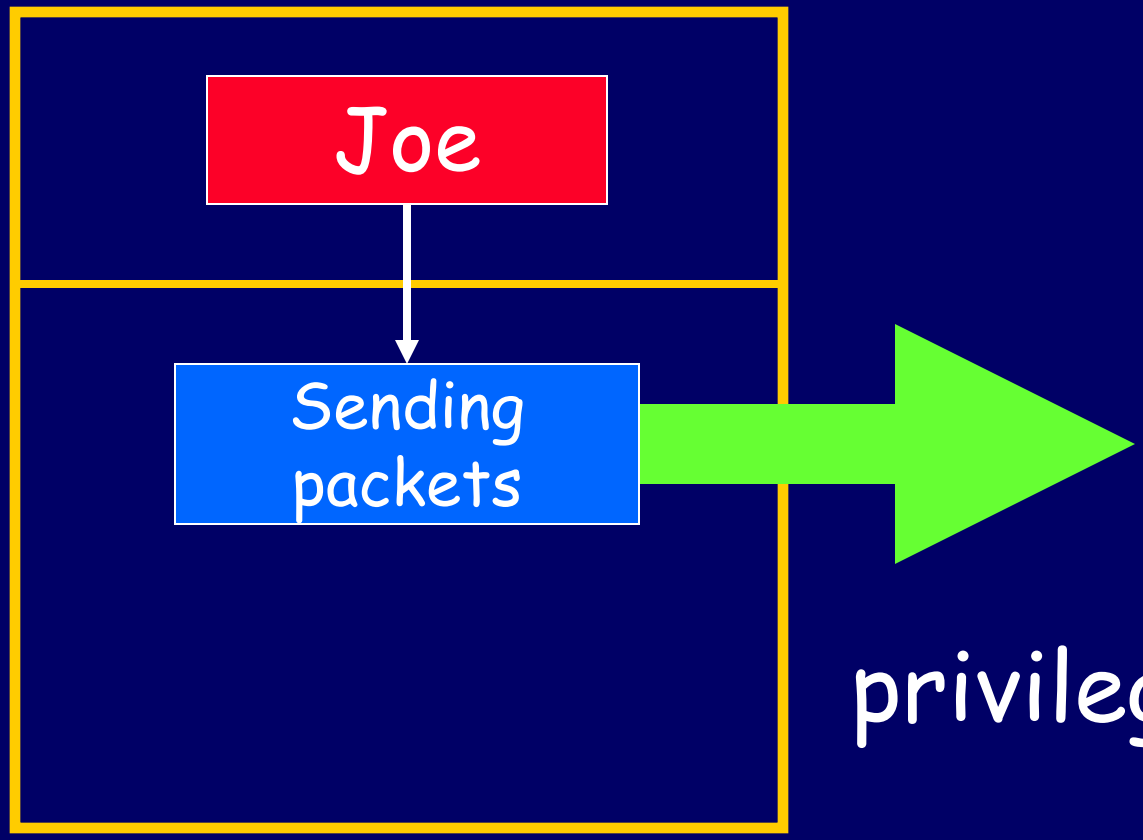
privileged user

Form of Service Policies: Usage

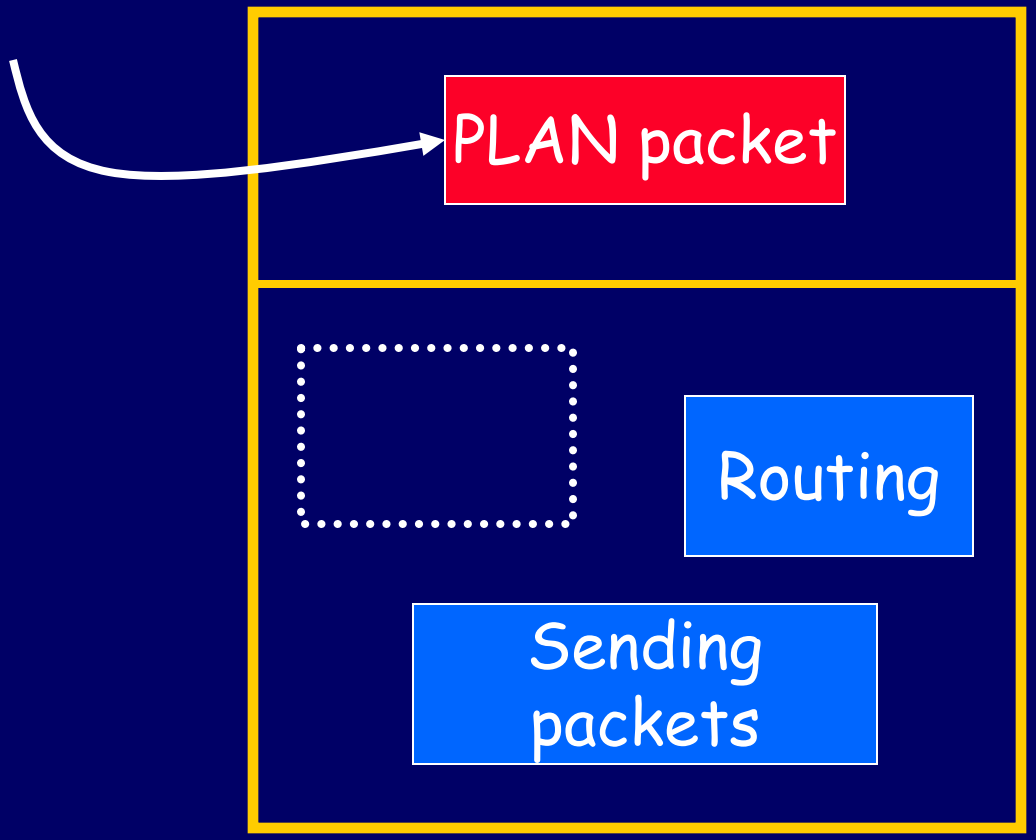


Form of Service Policies:

Usage

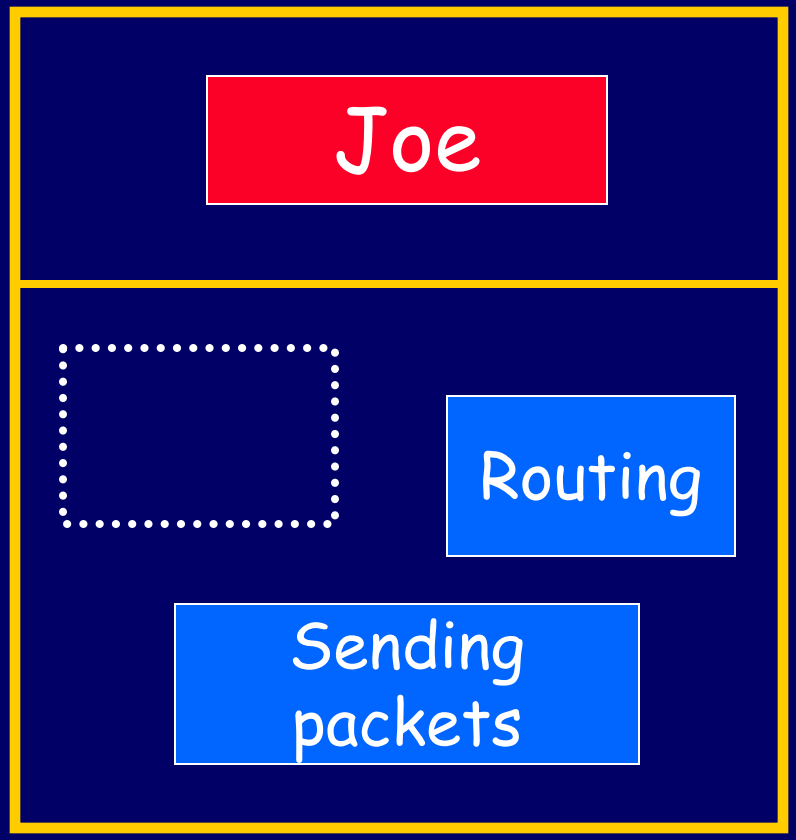


Security Procedure



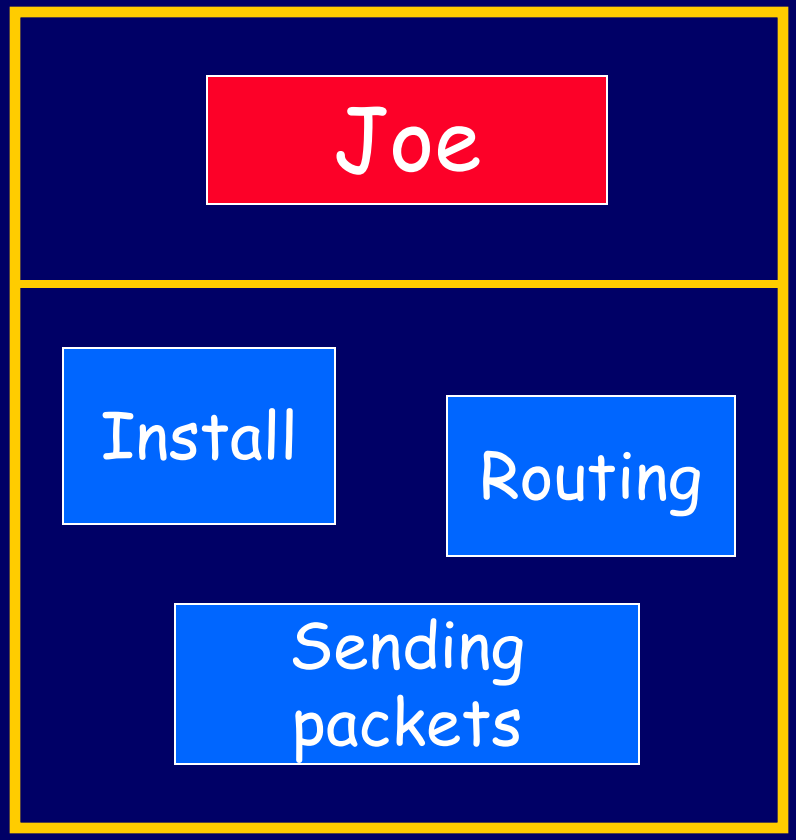
arrival
as default user

Security Procedure



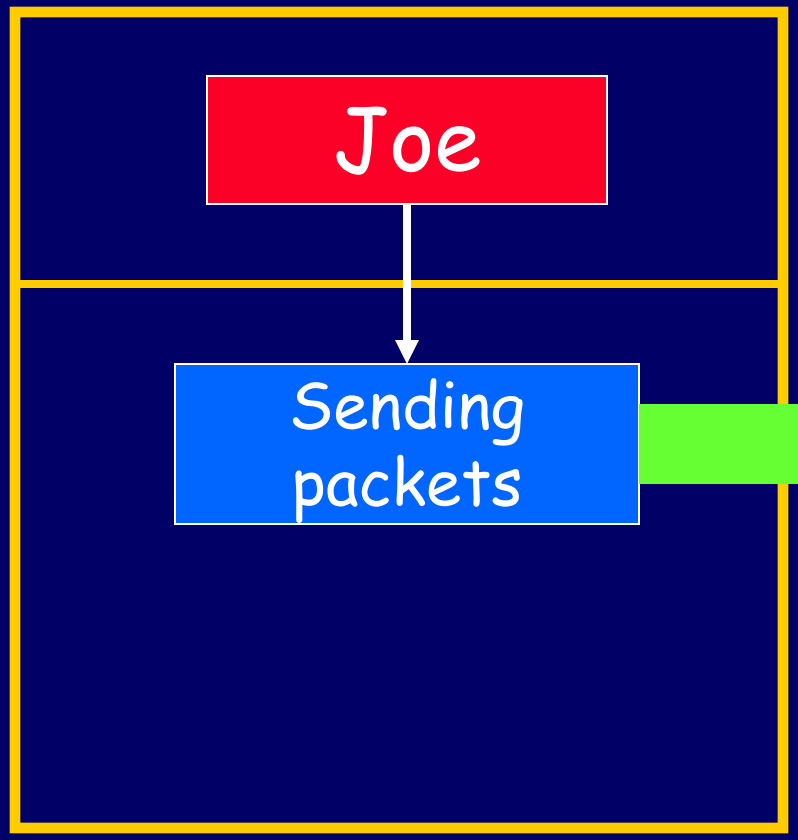
authentication

Security Procedure



namespace adjustment

Security Procedure




```
...
if Joe then
   more bandwidth
...
```

per-usage adjustment

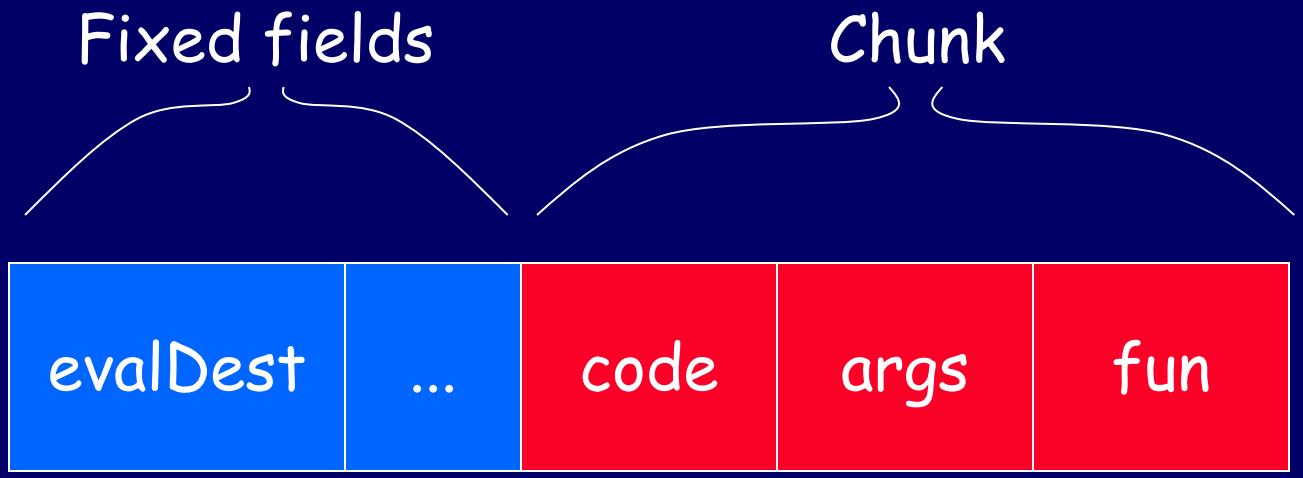
Security Mechanisms

- Authentication via HMAC-SHA1
  - →signed Diffie-Hellman, as with IPsec
- Authorization Policies - Query Certificate Manager (QCM)
  - →language based on sets
  - →set descriptions may be distributed
  - →Use of certificates for push-based policy

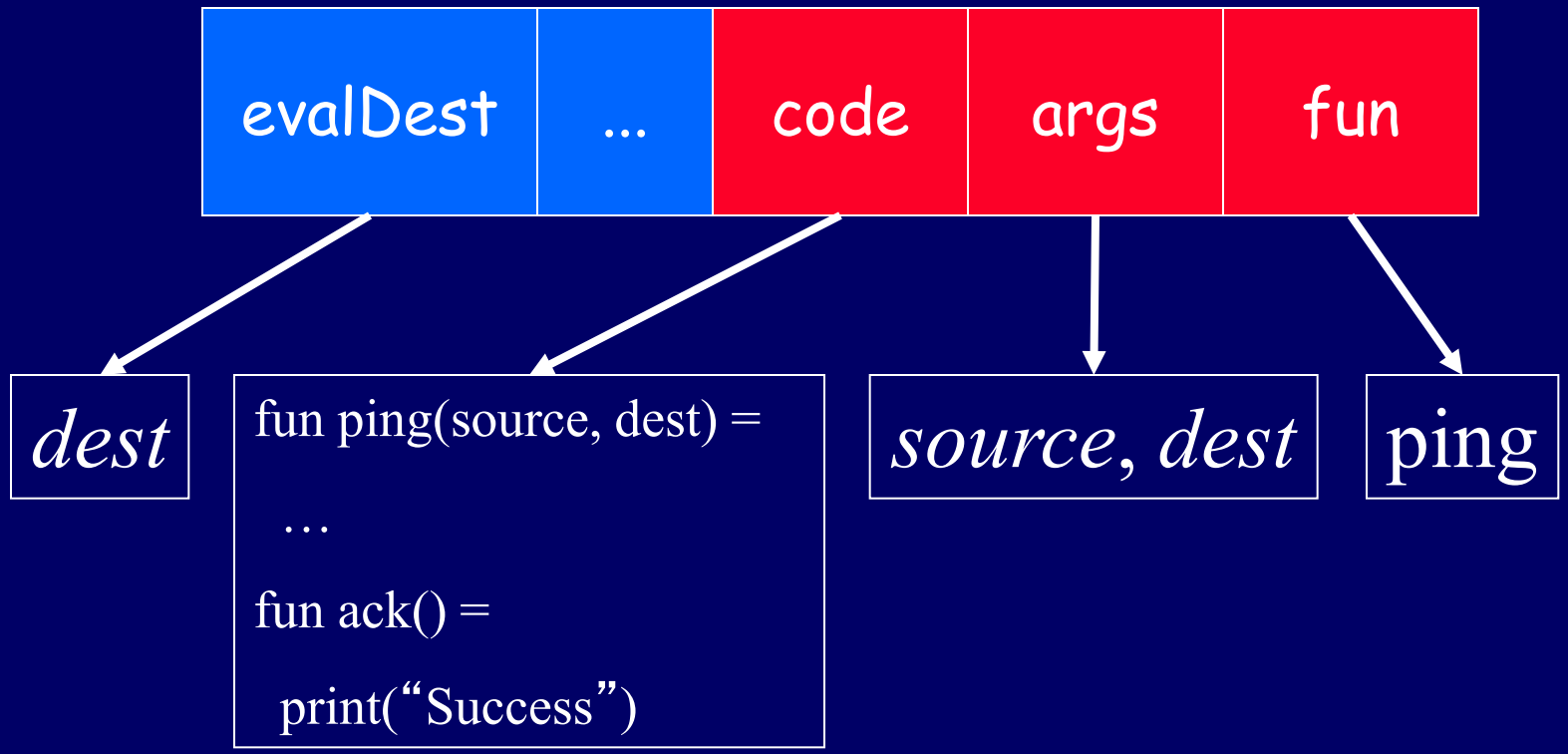
Chunks - units of authentication

- Unit of evaluation in PLAN
  - → like a suspended function call
- First-class
  - → can be manipulated as data within PLAN programs
- Useful programming construct
  - → encapsulation via `eval`

Chunks - in PLAN packets



Ping packet


evalDest
...
code
args
fun
dest
fun ping(source, dest) =
…
fun ack() =
print("Success")
source, dest
ping

Core Service

```
authEval: 'a chunk * sign -> 'a
```

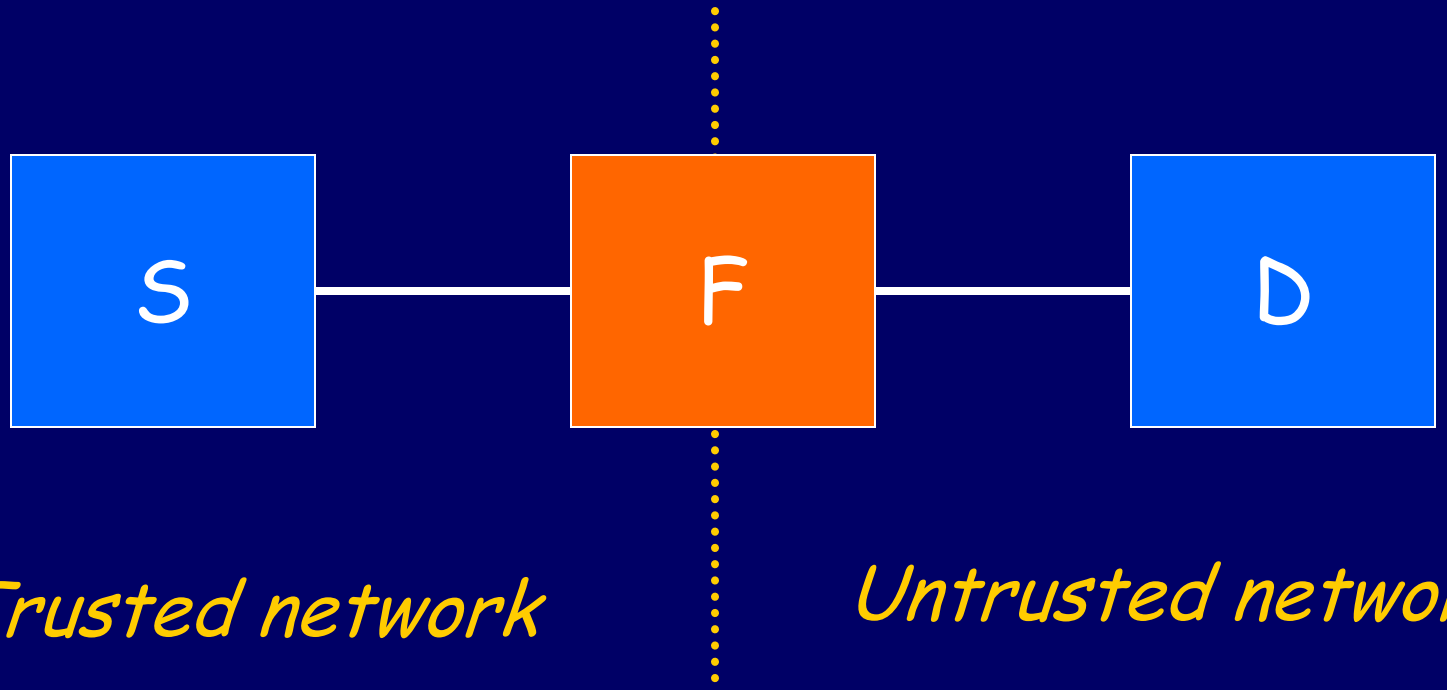
- takes a chunk and an HMAC digital signature and authenticates the chunk
  - → if successful, performs namespace adjustment and evaluates the chunk

Application: An Active Firewall

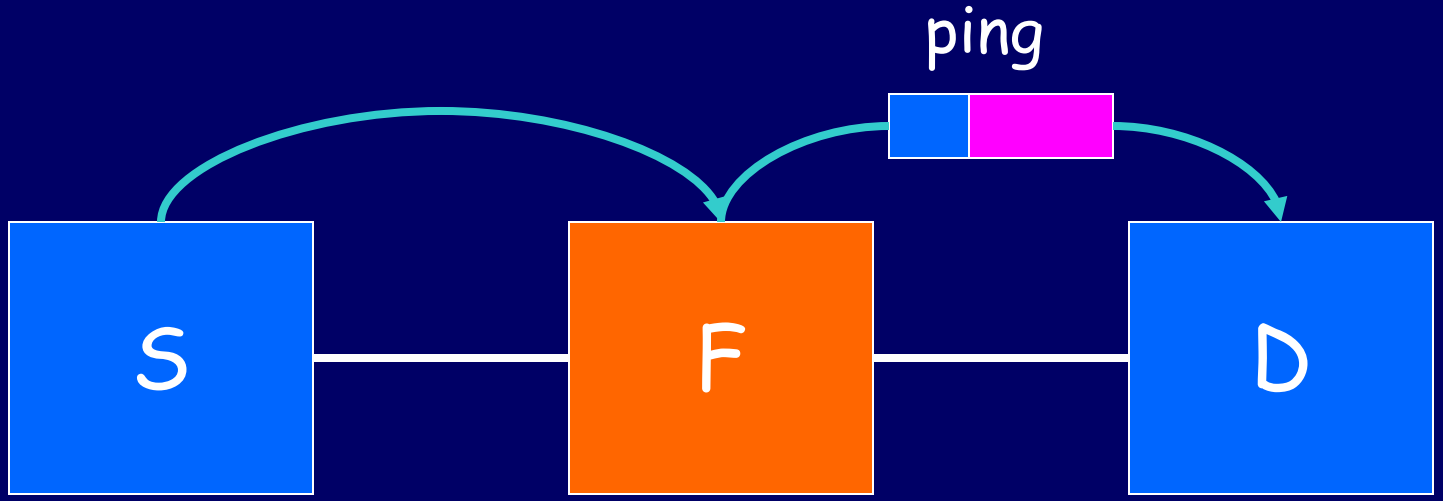
- Rather than *filter* external packets, *restrict their privilege*
- Accomplished by encapsulating incoming packets with service-restricting chunk

```
fun wrap(c,sign) =
   (zeroRB(); authEval(c,sign))
```

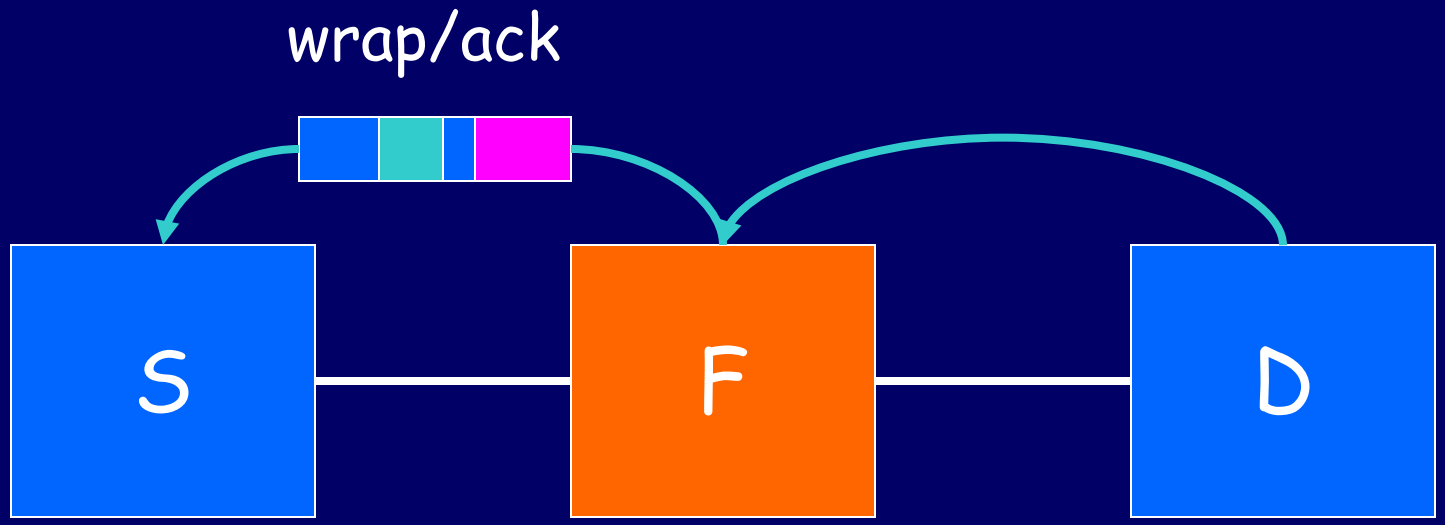
Experimental Setup



Outgoing Ping

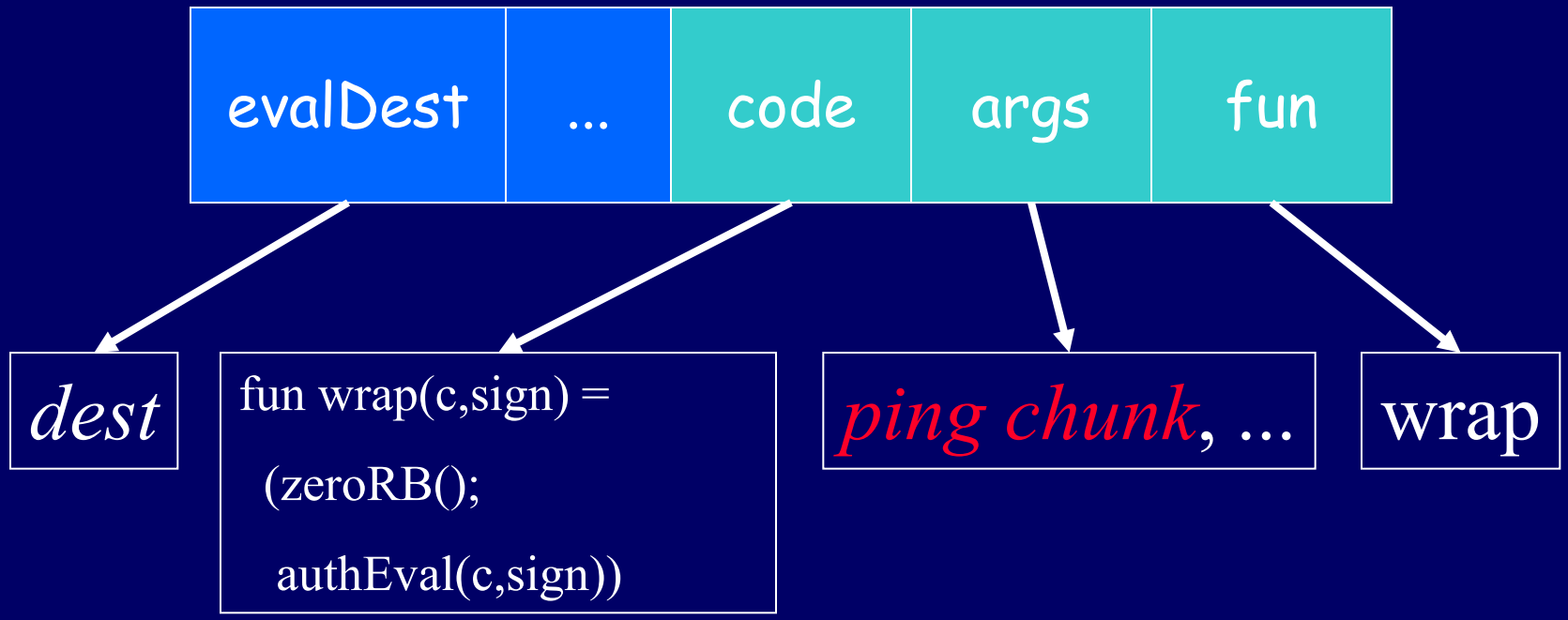


Returning Acknowledgement

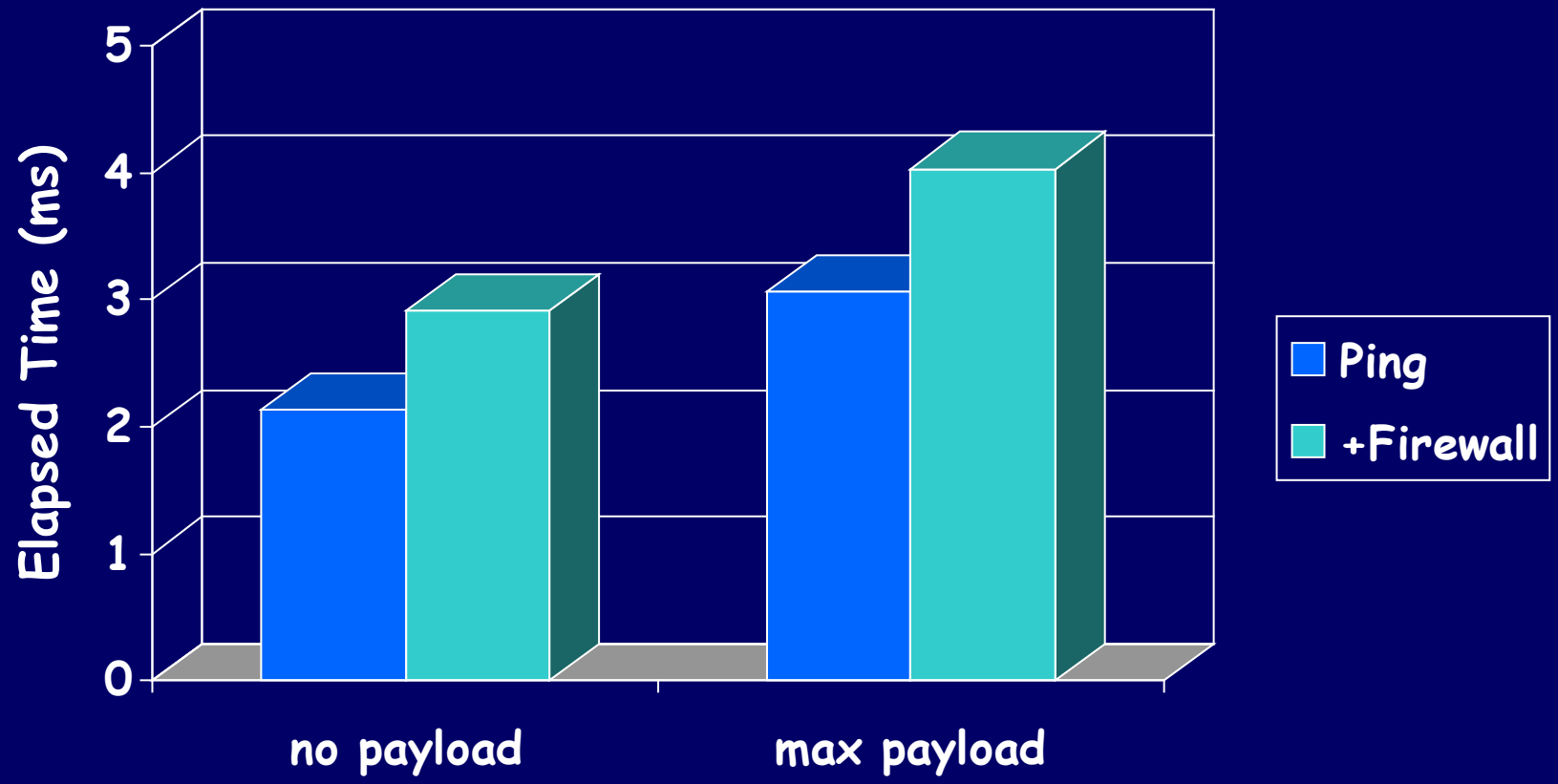


Firewall signs as and encapsulates packet chunk

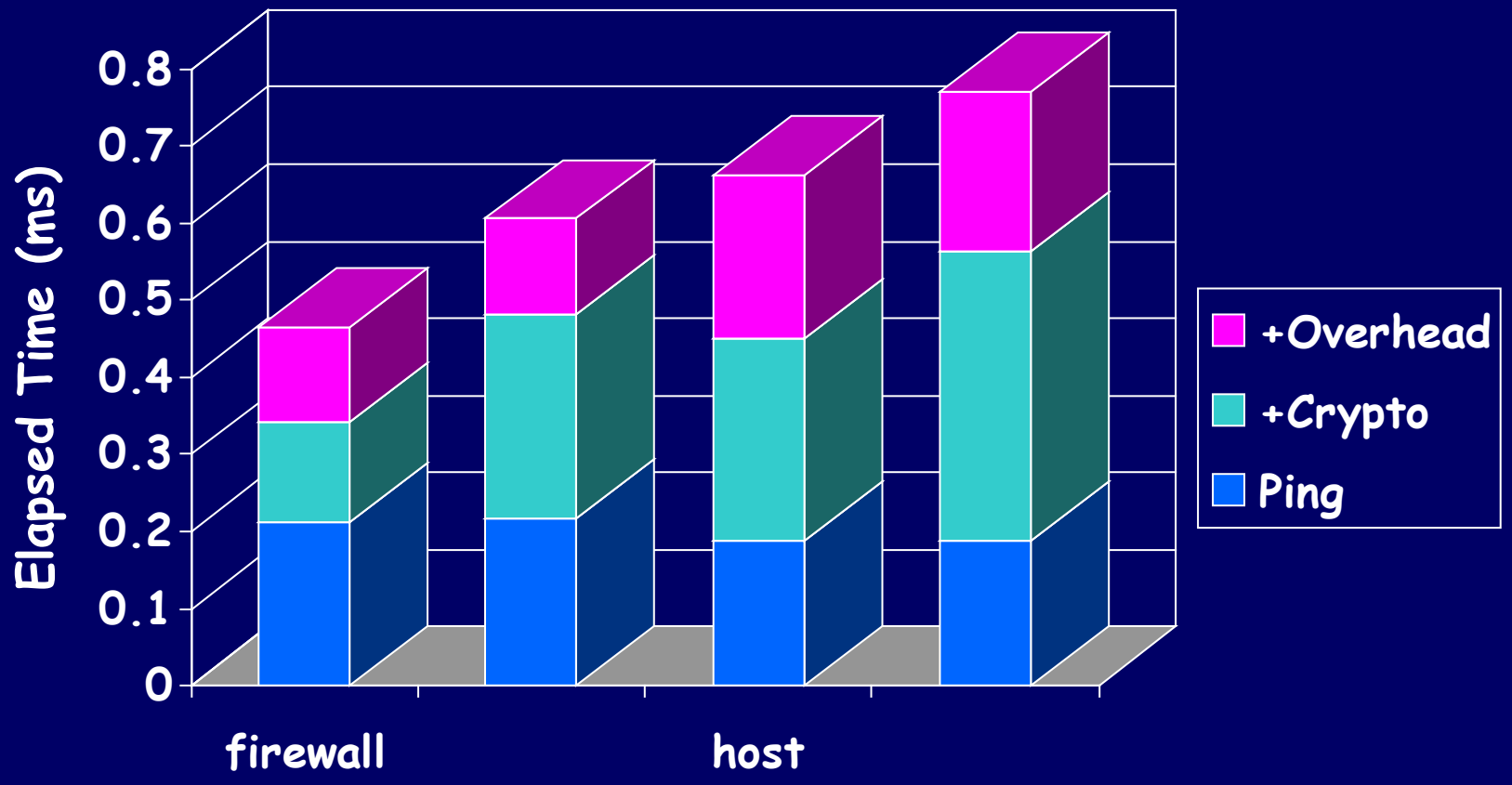
Firewall-wrapped Ping packet



Firewall Performance



Firewall Overhead Breakdown



Related Work

- AN Security
  - → Security Architecture for AN
  - → SANE
  - → SQoSH/RCANE
- Language-based protection schemes
  - → SPIN (Modula-3), MMM (Caml), J-kernel (Java), PCC and TAL (x86, Alpha assembly)
- Trust Management
  - → Keynote, PolicyMaker

Conclusions

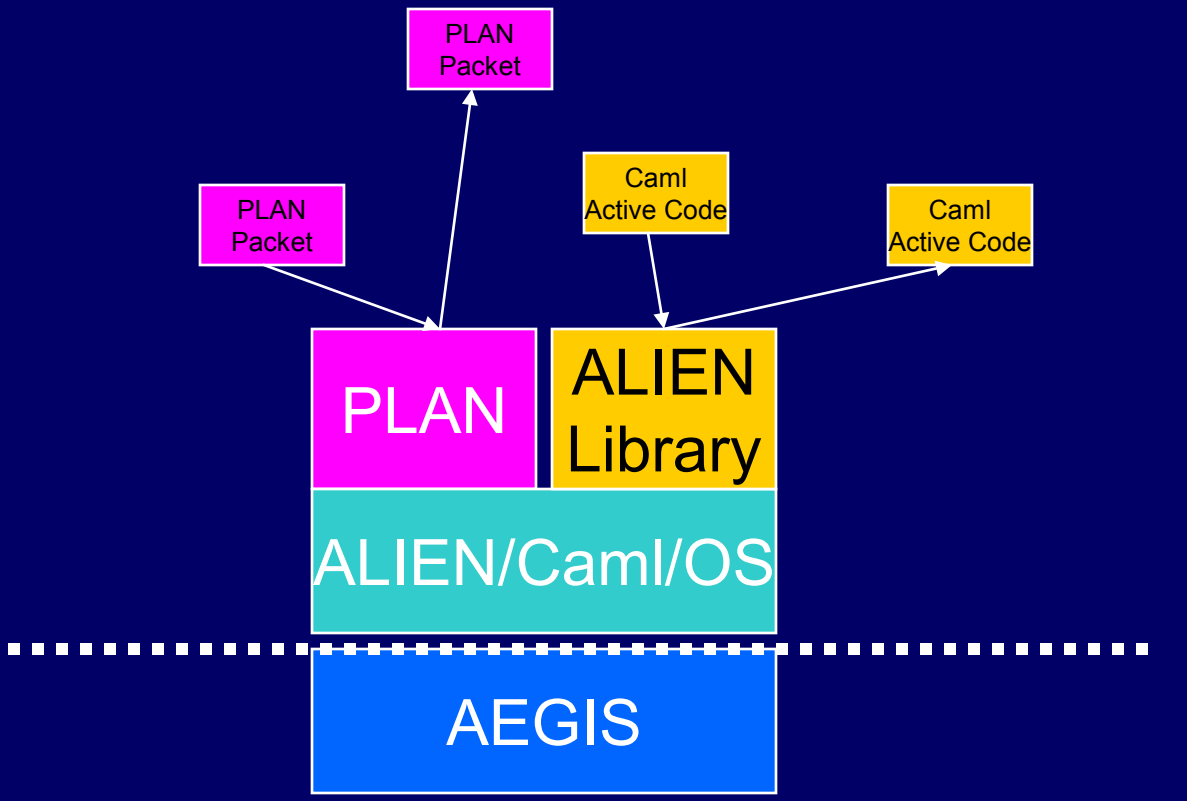
- Security in AN: PLANet
  - →while preserving performance, flexibility and usability
- Achieved with 2-level architecture
  - →language safety in the packets
  - →trust management for services
- Useful
  - →active firewall (active encapsulation)

Acknowledgments

- DARPA (of course!)
- Scott Nettles, Jonathan Moore, Scott Alexander, Bill Arbaugh, Trevor Jim, other SwitchWare team members
- An earlier (and highly abbreviated) version of this paper was presented at IWAN99 in Berlin, July 1999

Questions and Discussion

SwitchWare System Architecture



ALIEN Active Loader

- D. Scott Alexander's Ph.D. thesis