Integrating Security, Mobility, and Multi-Homing in a HIP way

Pekka Nikander

Ericsson Research Nomadiclab
&
Helsinki Institute for Information Technology

Outline

- The problem: TCP/IP is getting old
  - Locators and Host Identifiers
  - Combined locator–identifiers
- A Solution: Change the Architecture
  - The Basic Idea of HIP
  - Socket bindings revisited
  - Protocol walkthrough
- End-host Mobility & Multi-homing
  - End-Host Mobility & End-Host Multi-Homing
  - Virtual Interfaces
- Implementation status
- Summary

The problem: TCP/IP is getting old

- Computers are getting more *mobile* and more *connected*
  - E.g. a PDA with GSM / UMTS / WLAN / BlueTooth
- Internet addresses still are *addresses*,
  - bound to places in network topology
  - but working less well as host identifiers
- Proving mobility and multi-homing has turned to be hard
- Maybe it is a time to *rethink* the situation?

- Host Identity Payload (HIP) is a concrete attempt to provide a new approach — we may be right, we may be wrong, but we try, and hopefully we learn
  - WORK IN PROGRESS!

Locators and Host Identifiers

- IP addresses are bound to network *topology*
  - In the old network class (A / B / C) based system it was less so, but hosts were still bound to networks
  - Today, keeping the routing tables manageable requires CIDR and hierarchical address prefixes
    - Therefore the globally routable IP addresses are sometimes called as *Provider Assigned* (PA) addresses

→ IP addresses act as a *locators* (names of locations)

- Network connections are bound to IP addresses
  - TCP / UDP sockets are *identified* with IP addresses
    - SCTP is an exception to this, and indeed a comptetitor to HIP
  - DNS gives out addresses for new connections

→ IP addresses act as *host identifiers* (names of hosts)

Combined locator–identifiers

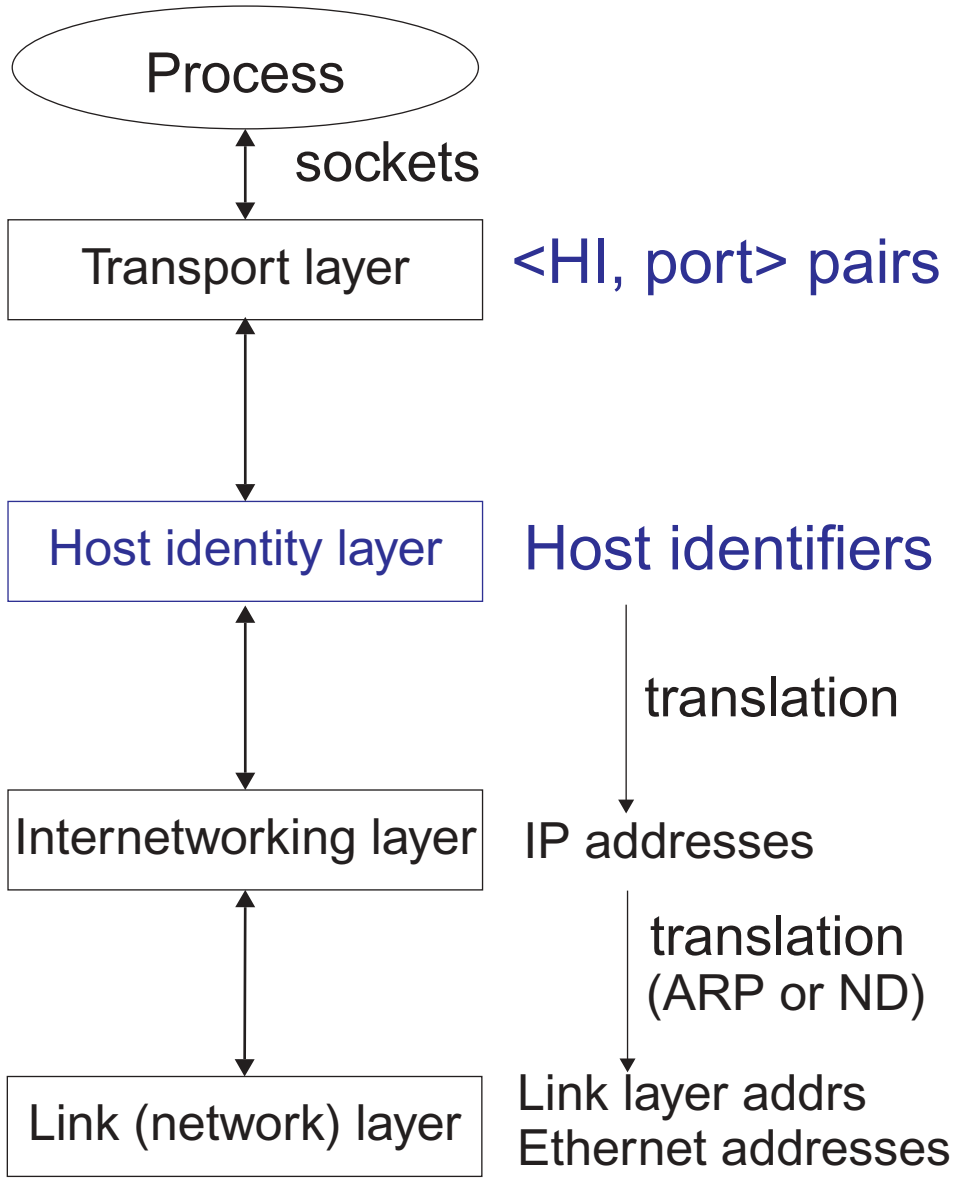
- IP addresses are *combined* locators and host identifiers
- Good from security point-of-view
  - Packet sent to Alice's address is indeed sent to Alice
  - Why? Because Alice is identified by the address!
    - The routing system is assumed to be secure in the sense that it either delivers the packets to their destination or not at all
    - This limits the potential attackers to those topological paths that make it possible to eavesdrop packets sent to a particular address
- Bad from mobility / multi-homing point-of-view
  - Host changes its location → must *change its identity*
    - Leads to the Home Address / Care-of-Address design in Mobile IP
  - Multi-homed must have *multiple identities*
    - Leads to multi-address sessions in SCTP
  - Managing multiple / dynamic addresses becomes harder than necessary

A Solution: Change the Architecture

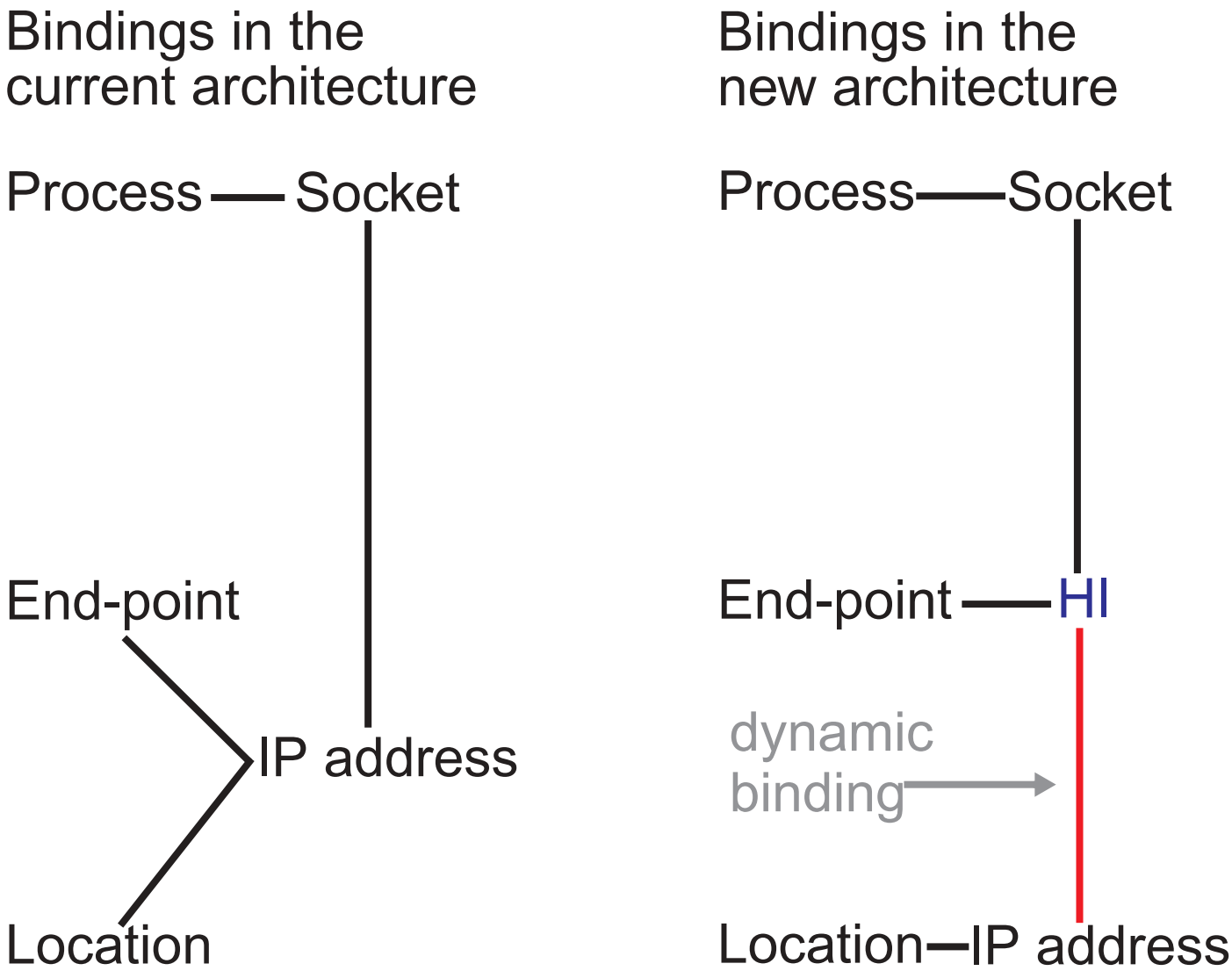
- Separate locators from host identifiers
- Let IP address continue to function as locators
  - No changes to the routing infrastructure
  - Mobile host still needs to keep changing its address
  - Multi-homed host still has multiple addresses
- Create a *new name space* for host identifiers
  - Use *public keys* as primary identifiers
    - How to get a host's public key is beyond our current scope
- Provide a secure binding between a host's public key and its IP address(es)
  - Sign the address(es), optionally use CGA
  - Check that the key is available at the address(es)
    - This step is essential for DoS protection; cf. Mobile IPv6 RO

The Basic Idea of HIP

- A new layer
- A new Name Space: Host Identifiers
  - Public keys
  - Represented as key hashes, Host ID Tags (HIT)
- Sockets bound to the Host IDs, not IP addresses
- Kernel translates outgoing HI into an real IP address
  - cf. Bellovin's Host NAT

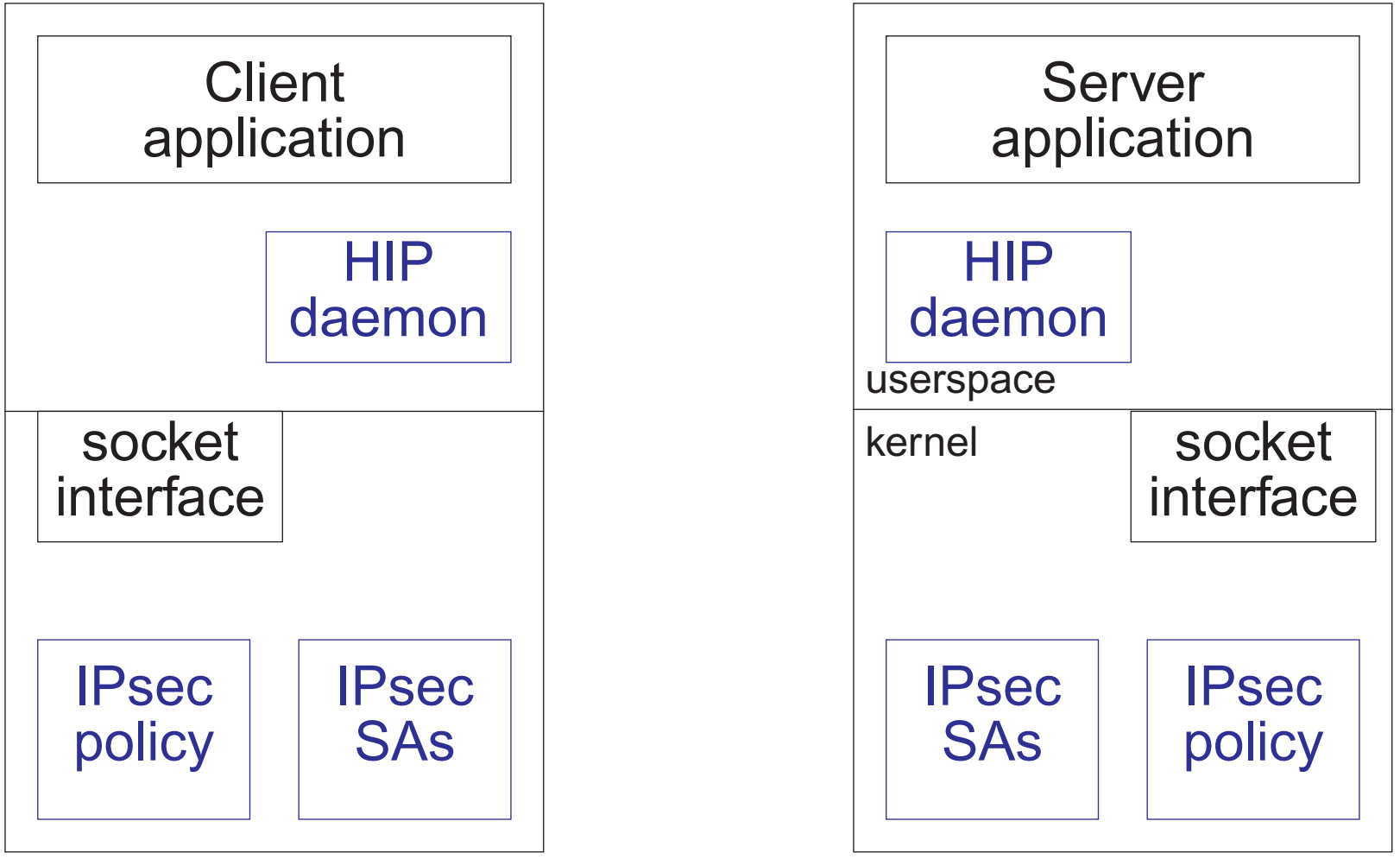


Socket bindings revisited



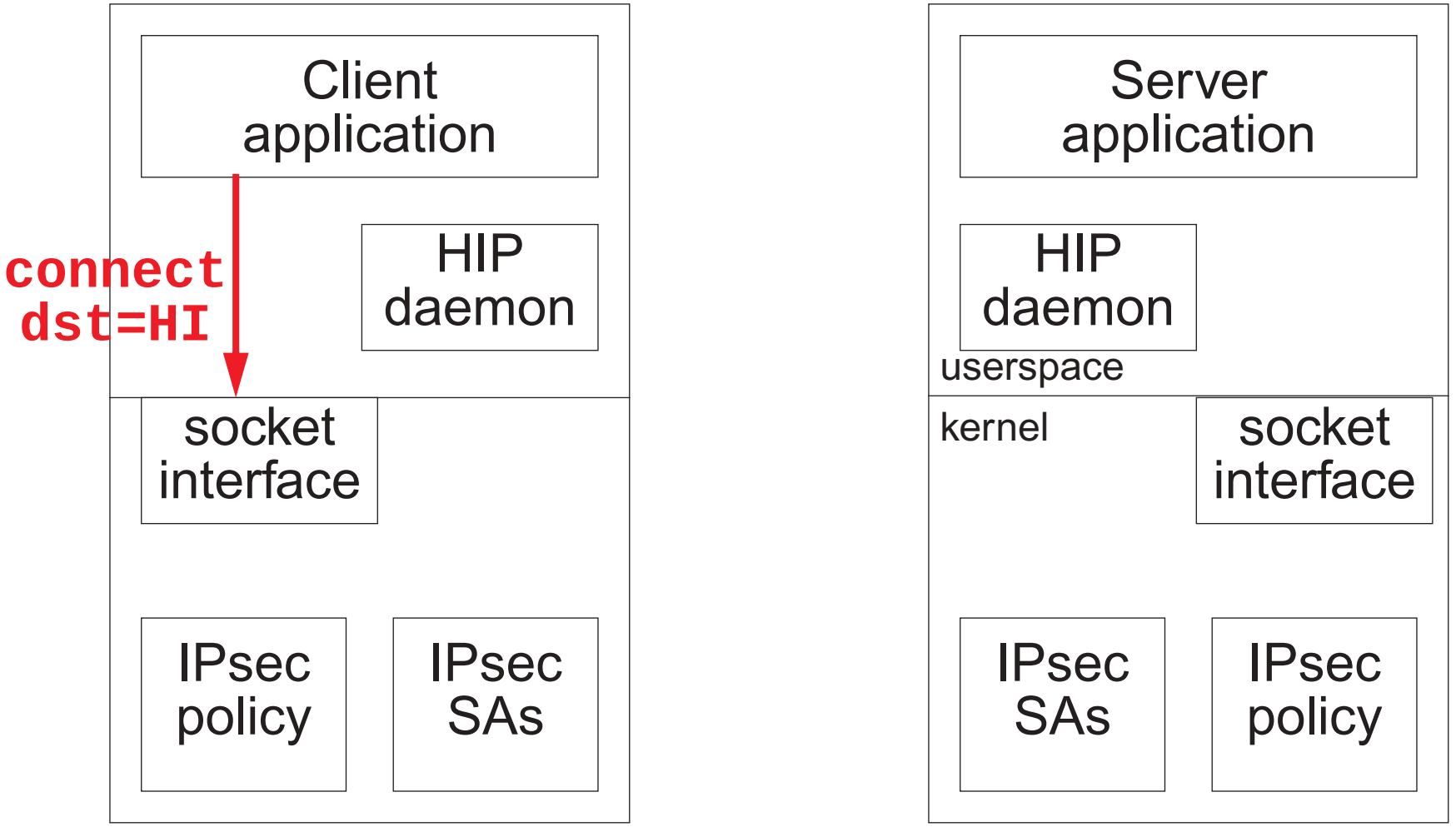
Protocol walkthrough

- A new key management daemon
- New IPsec transformation type: HIP transform



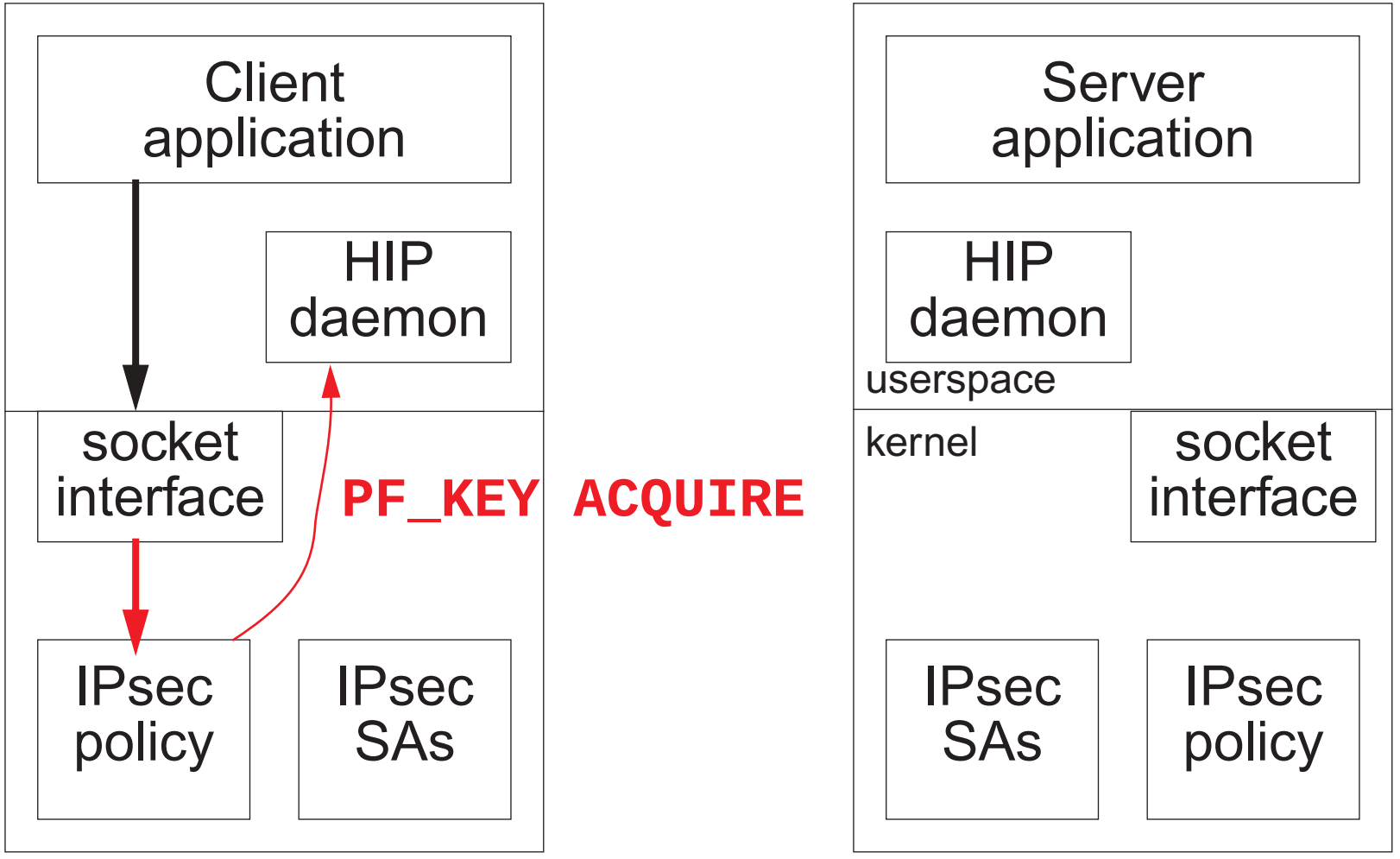
Protocol walkthrough

- Client application connects to a server
- **`connect(dst = server's Host Identity)`**



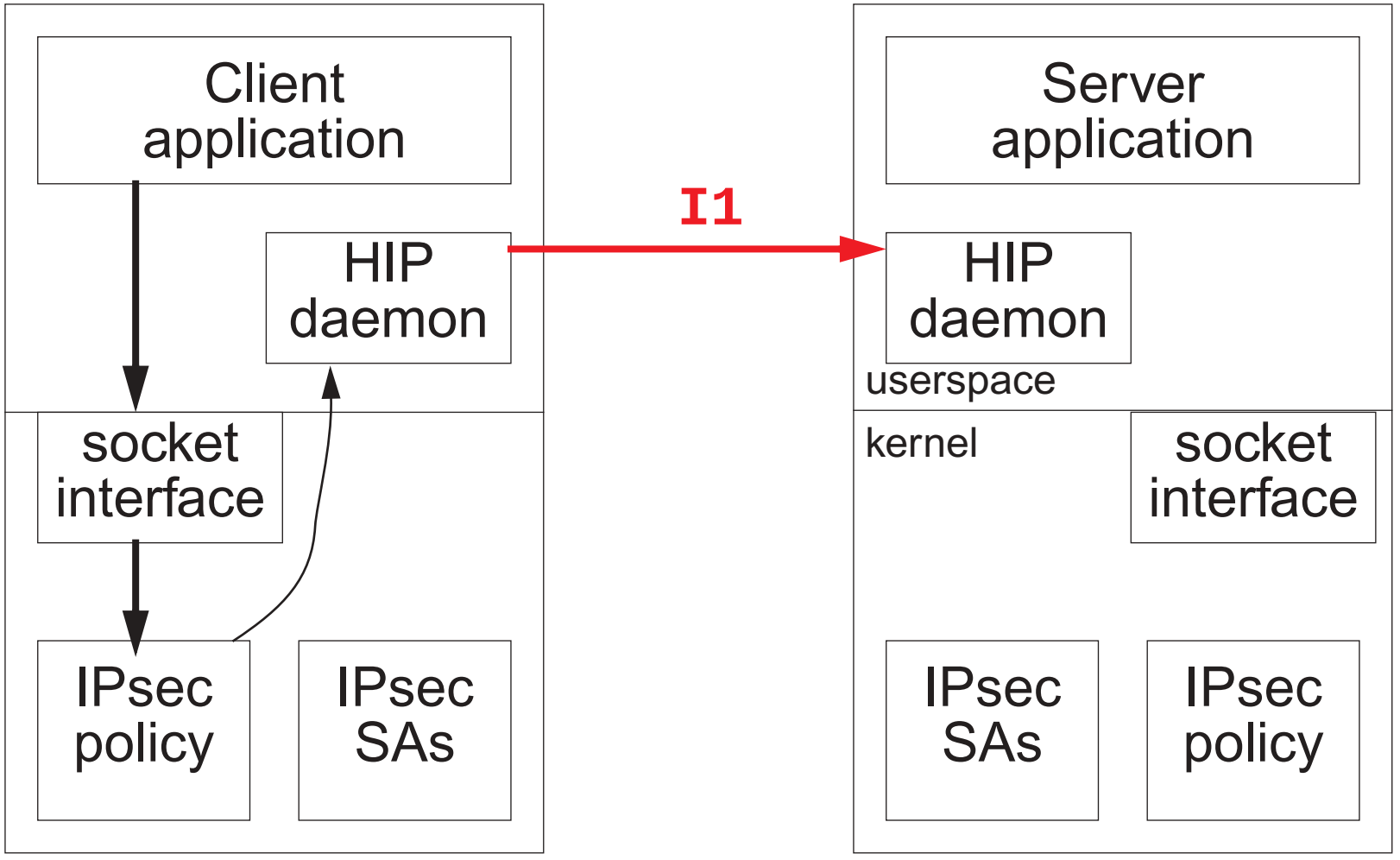
Protocol walkthrough

- IPsec policy engine traps the Host Identity
- Since there is no SA, a **PF_KEY ACQUIRE** is passed



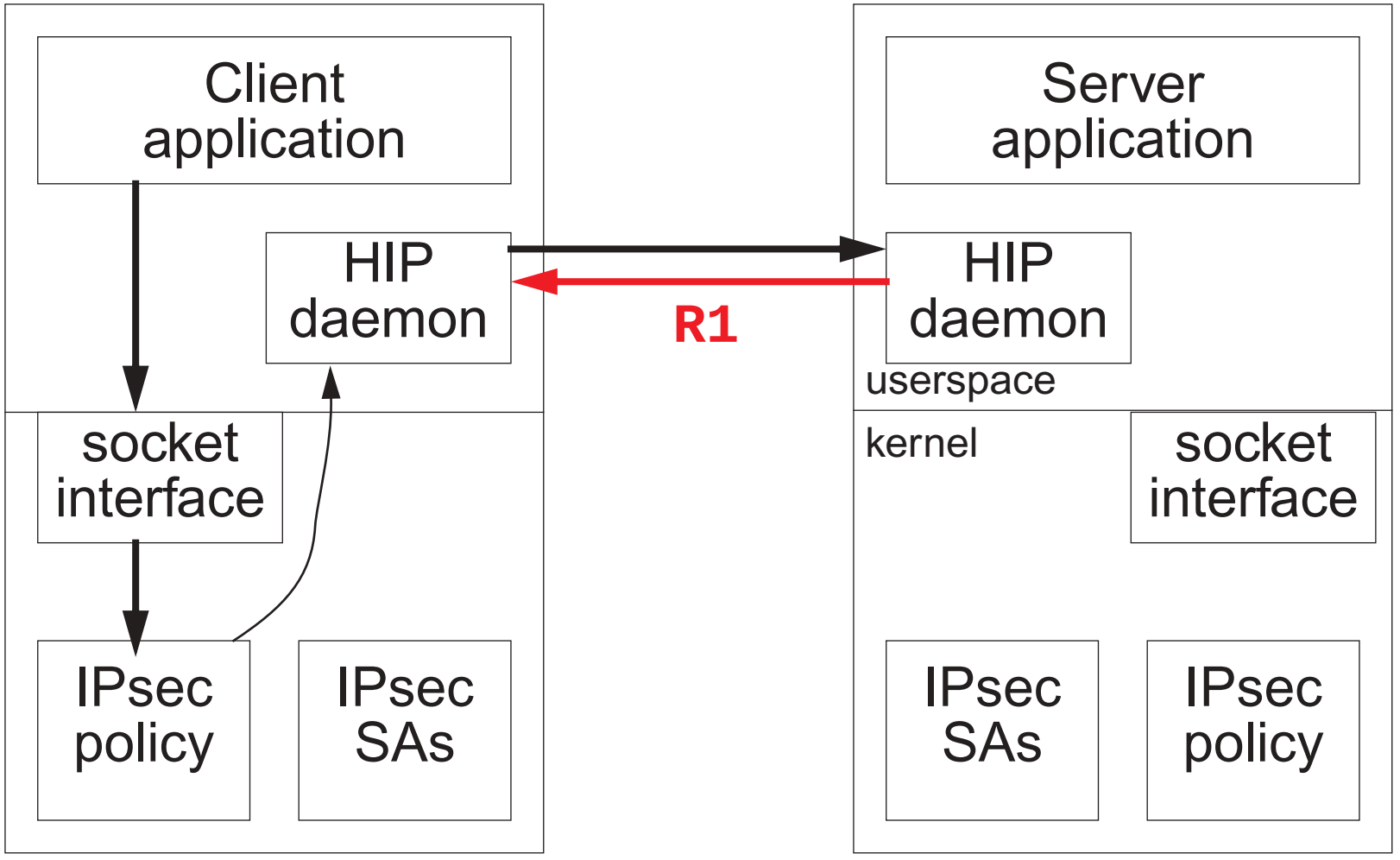
Protocol walkthrough

- The HIP daemon initiates a key negotiation
- **I1: HIT_I HIT_R**



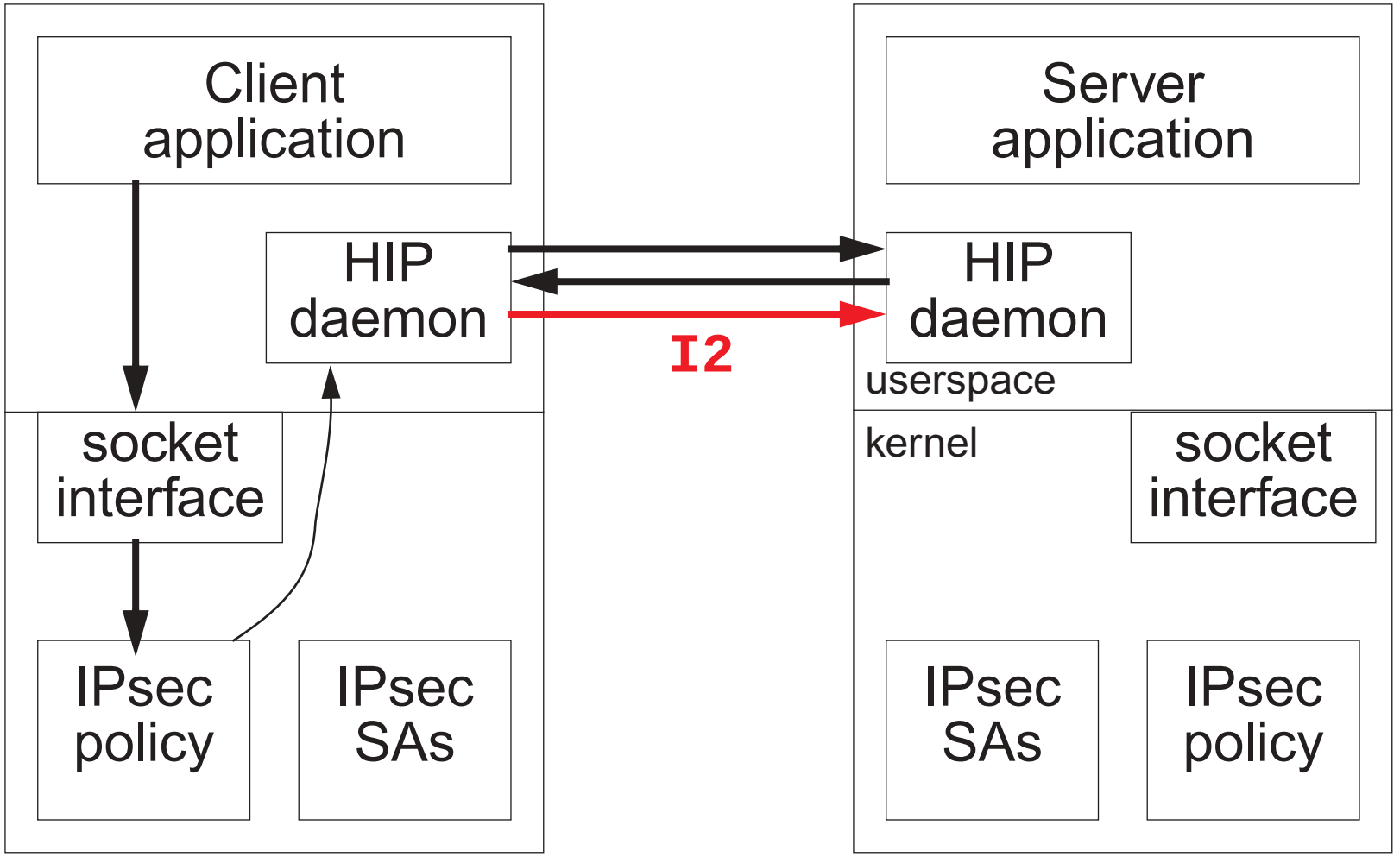
Protocol walkthrough

- Responder replies with a *canned* R1
- **R1: HIT_I $SIGN_R(HIT_R$ *Puzzle* g^x params $HI_R)$**



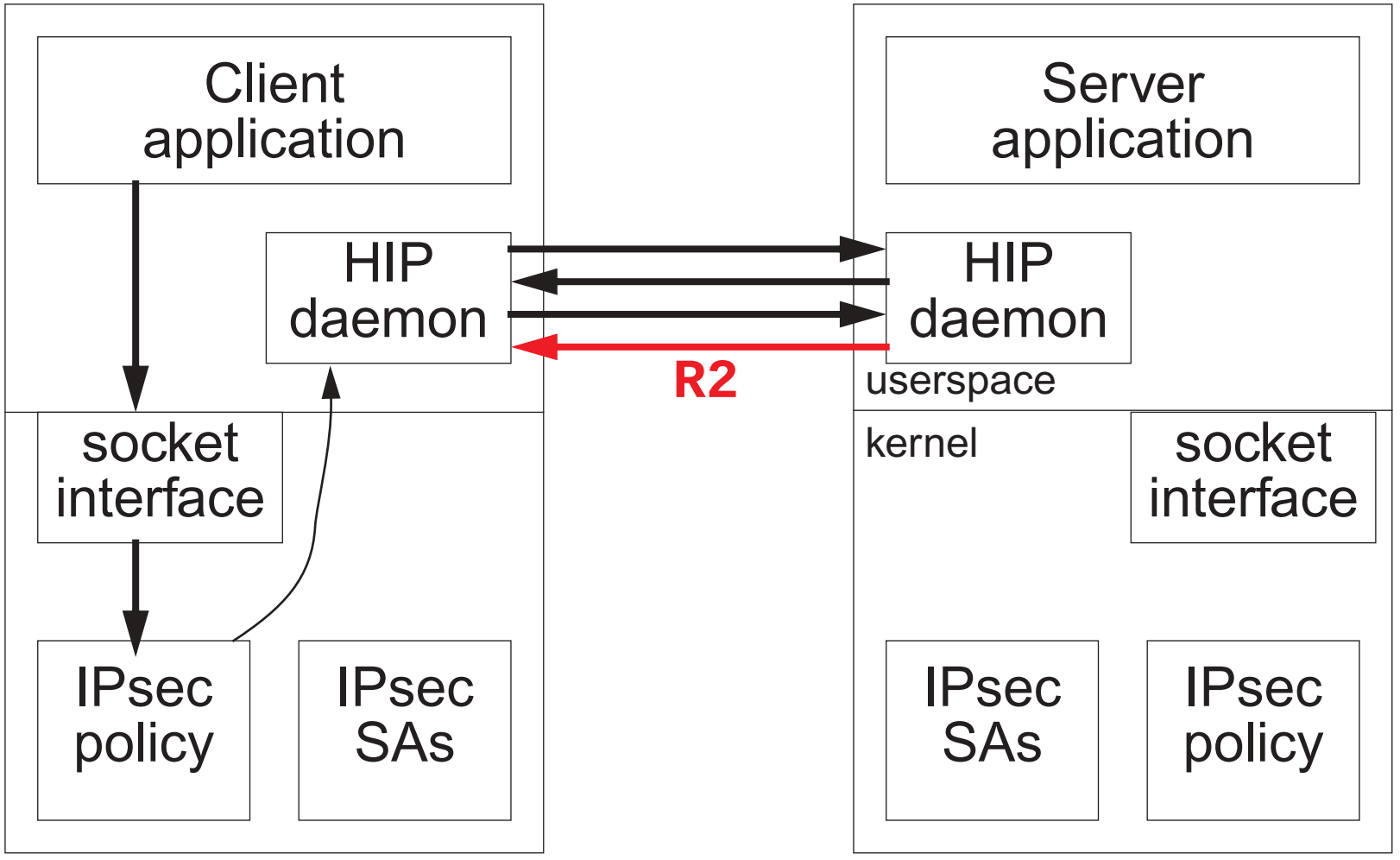
Protocol walkthrough

- Initiator solves the puzzle and sends I2
- **I2: $SIGN_I(HIT_I\ HIT_R$ *Result* $g^y\ ENC_{sess}(HI_I))$**



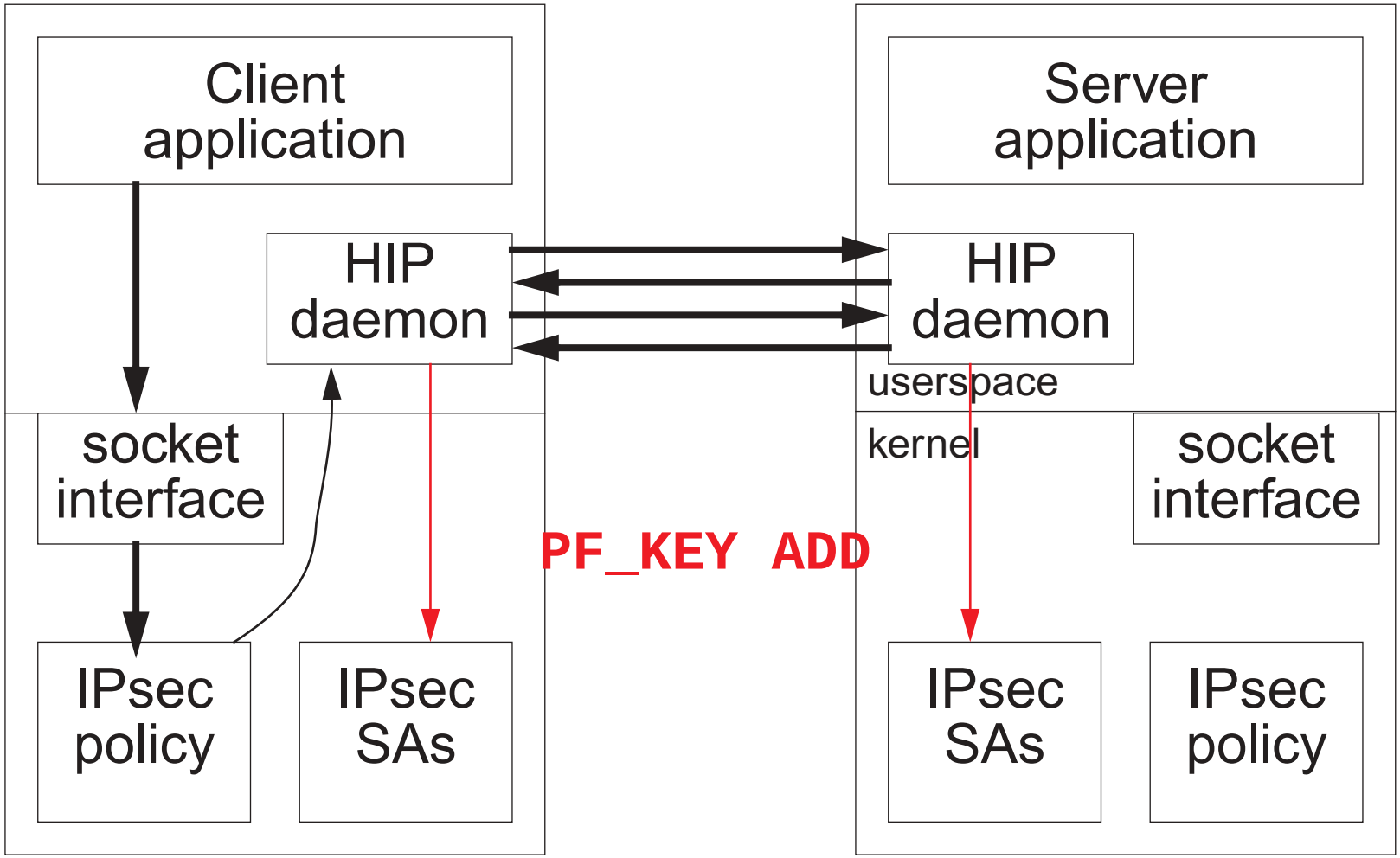
Protocol walkthrough

- The key negotiation is completed with R2
- **R2: $SIGN_R(HIT_R\ HIT_I\ SPI)$**



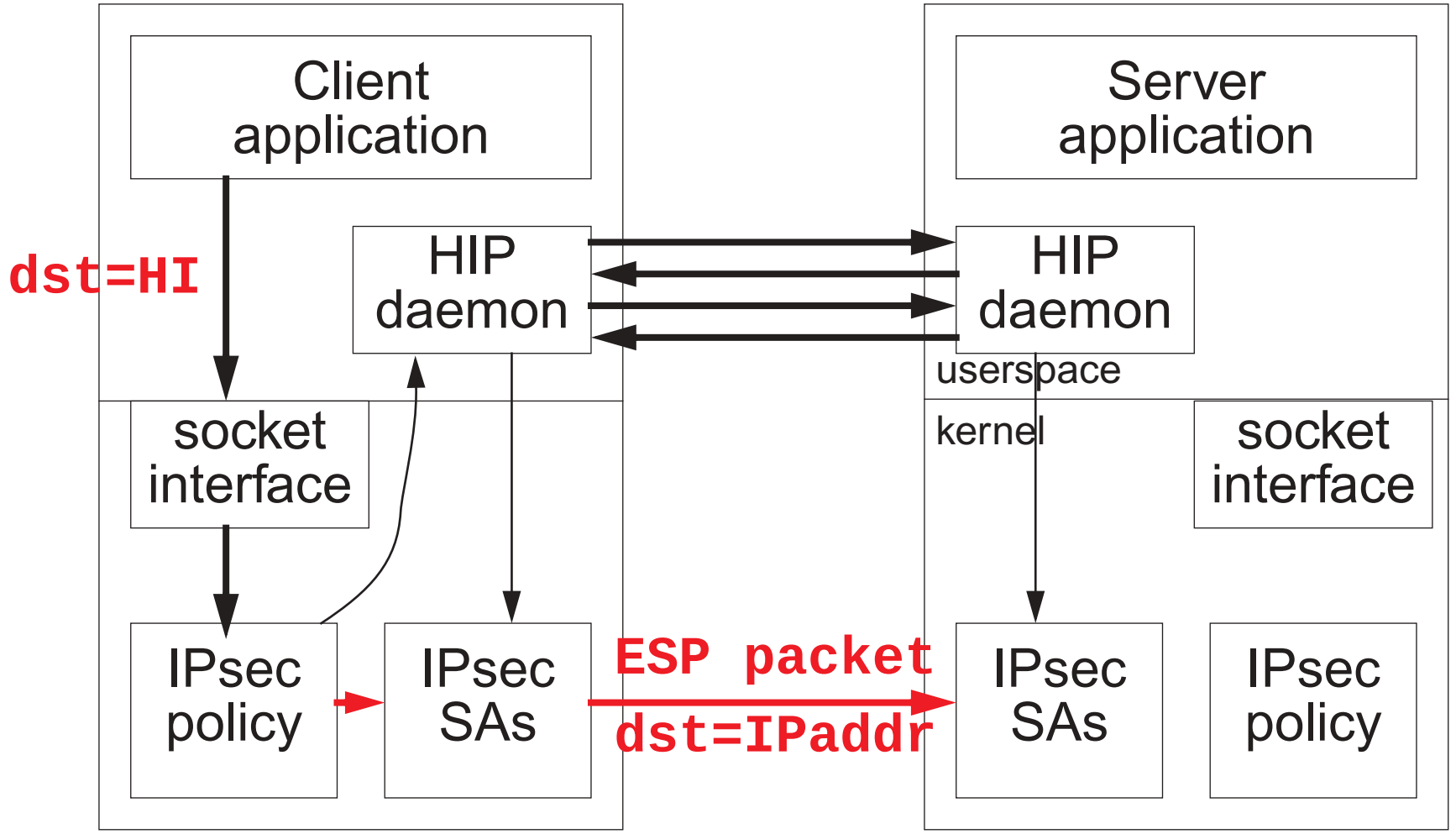
Protocol walkthrough

- HIP mode IPsec SAs are established
  - (Actually, the server does this before sending R2)



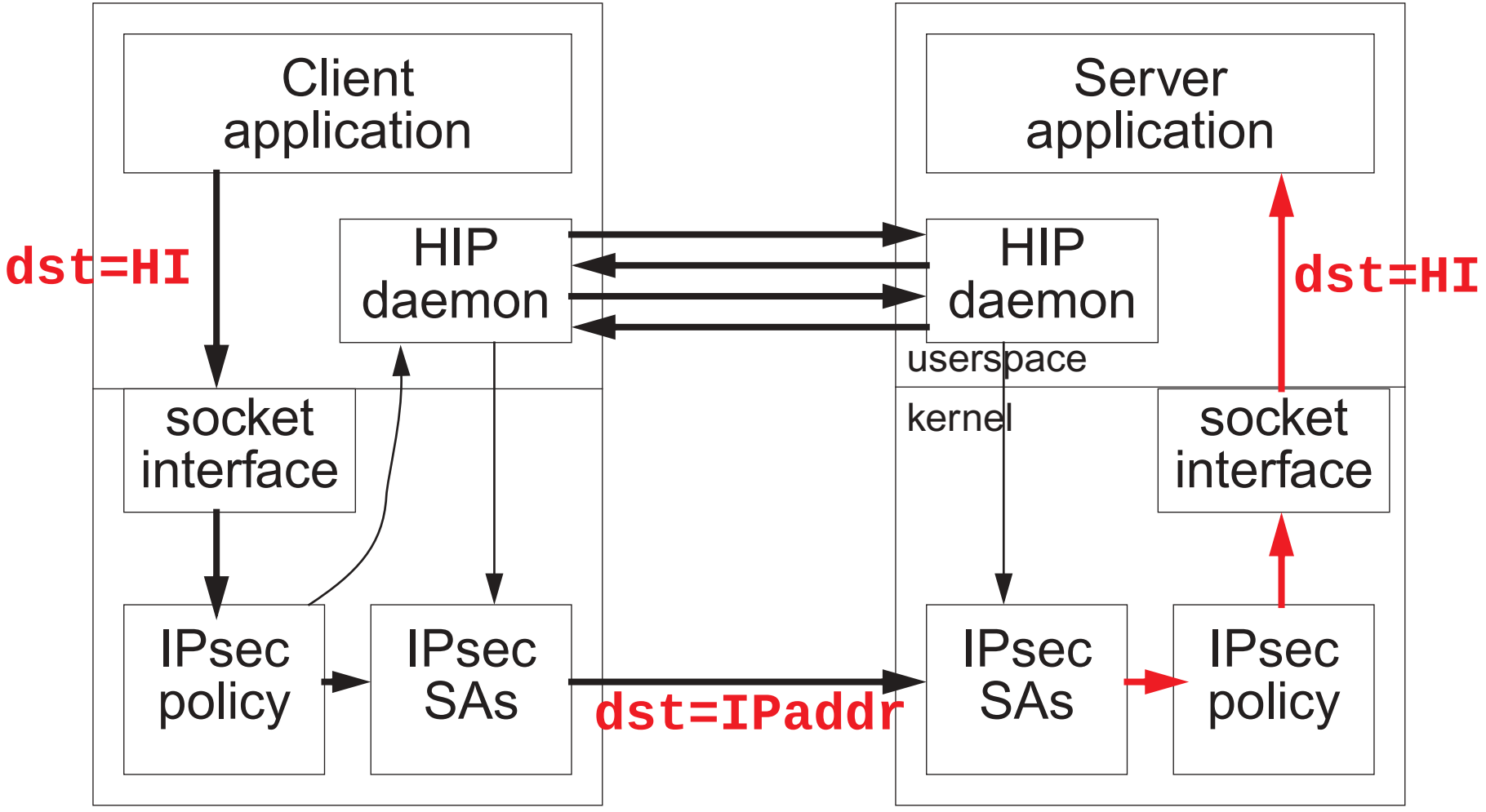
Protocol walkthrough

- The queued packet is sent over the ESP connection
- The SA *replaces* the server HI with a real IP address



Protocol walkthrough

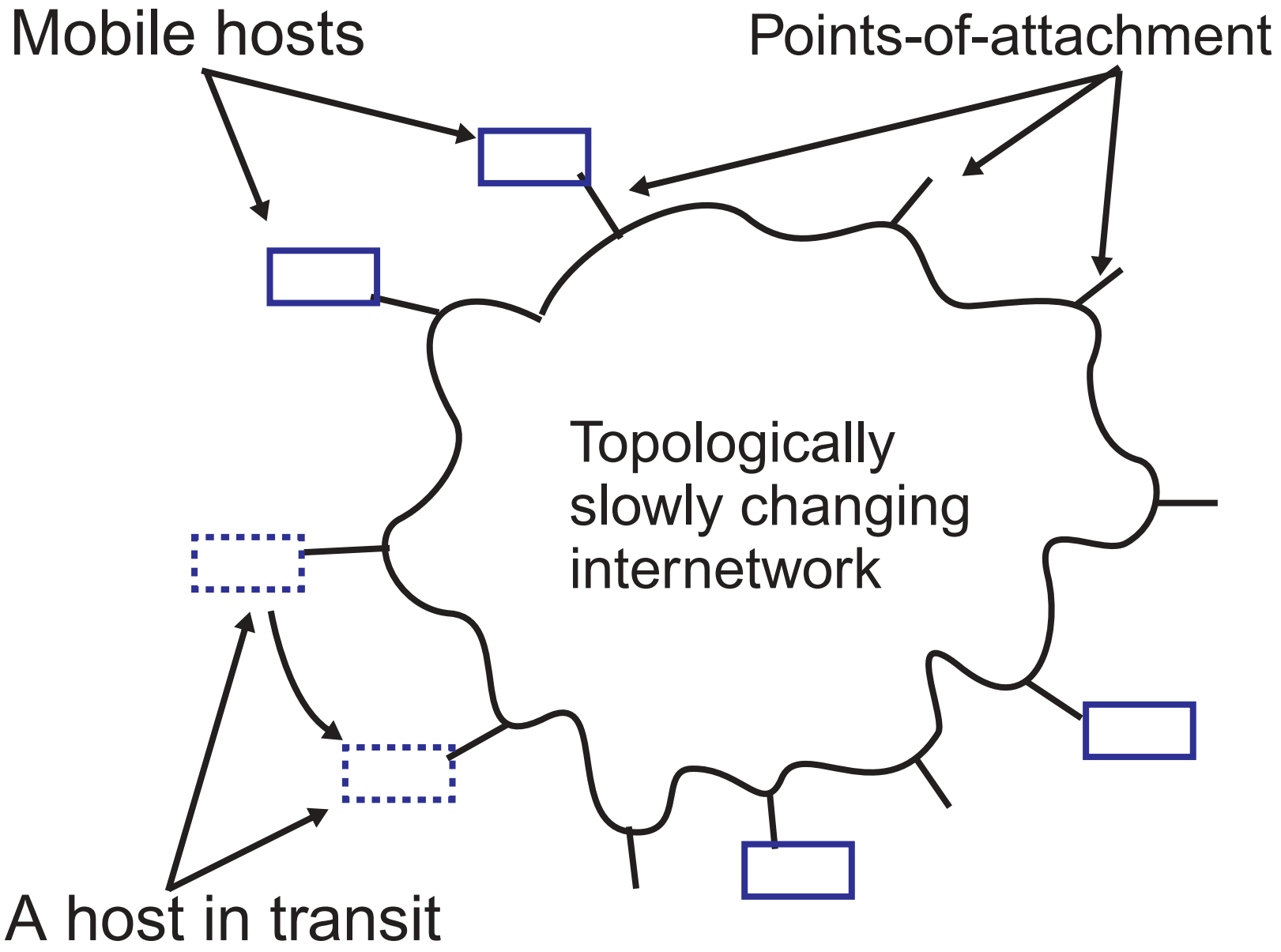
- The recipient SA replaces address with HI
- The packet is passed to the application



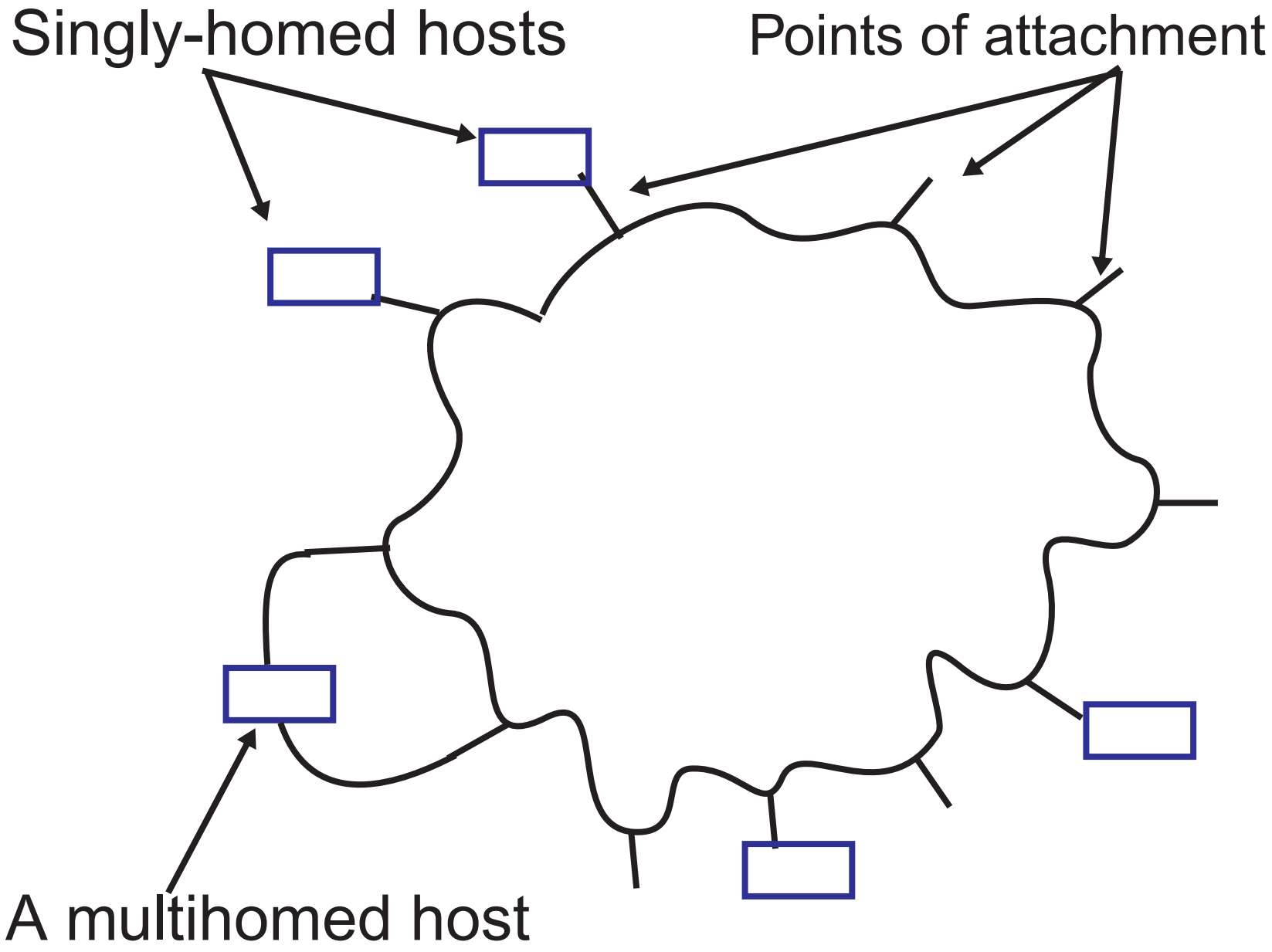
End-host Mobility & Multi-homing

- HIP seems to solve end-host mobility and multi-homing problems almost trivially
- Mobility and multi-homing become *duals* of each other
  - A mobile host has multiple addresses *serially*
  - A multi-homed host has multiple addresses *parallelly*
- The thinking can be folded into a *Virtual Interface Model*

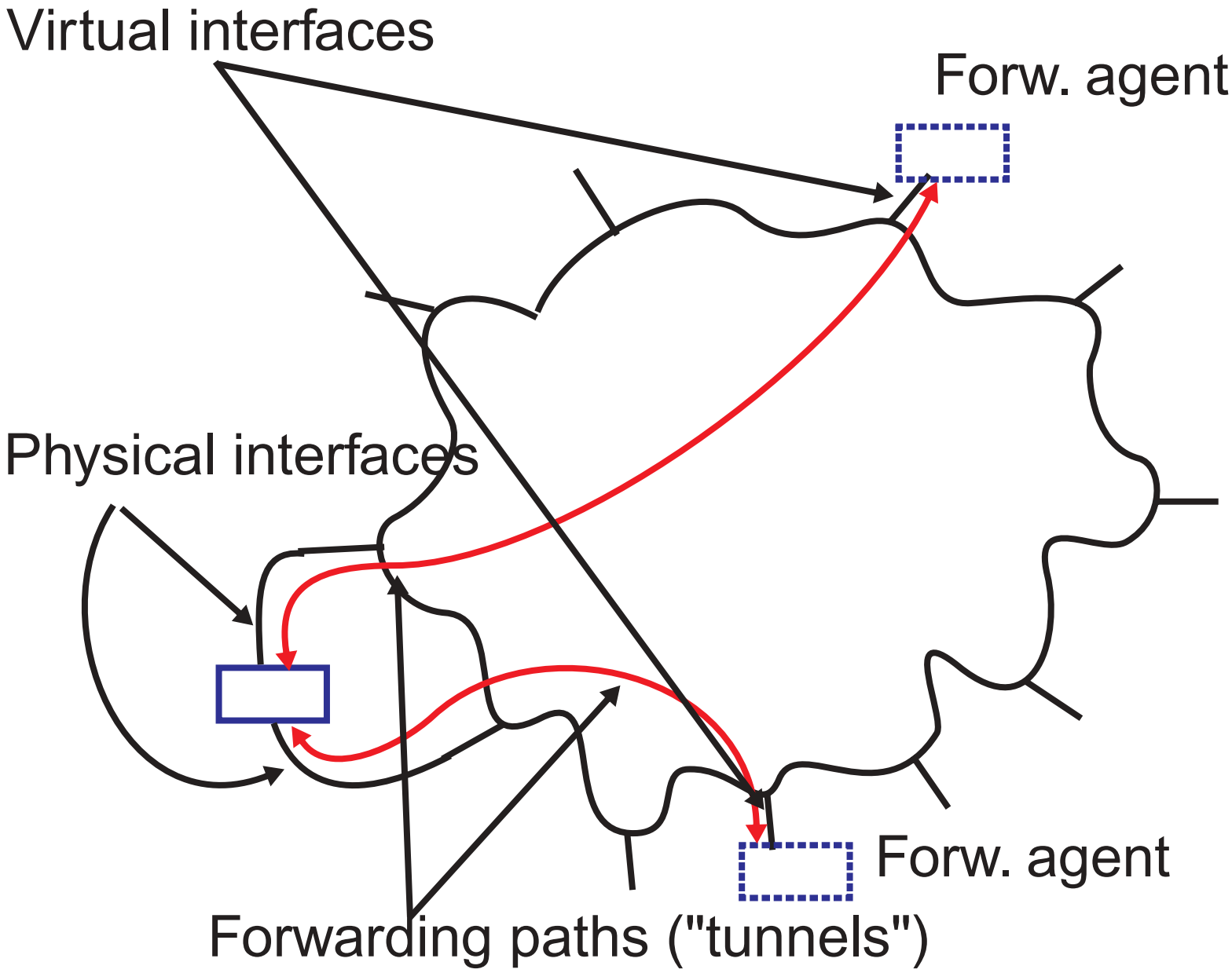
End-host Mobility



End-host Multi-Homing



Virtual Interfaces



Implementation status

- Four independent implementation efforts
  - Ericsson Research: NetBSD kernel
  - Helsinki Univ. of Tech.: Linux IPv6 kernel
  - Boeing Phantom Works: Linux IPv4 kernel
  - Andrew McGregor: Python user level
- All aim for an alpha version around the next IETF
  - Some versions already available, but with lacking functionality and small interoperability problems
  - Interoperability testing to happen at San Francisco

Summary

- A concrete, down-to-earth attempt to "fix" the Internet
  - Deployment can start at end-points
  - No changes required to routers
  - Can be made to work with firewalls
  - Supports NAT, but requires HIP-capable NAT boxes
  - Backwards compatibility can be provided with proxies
- Integrates IPsec key negotiation, end-host mobility, and end-host multihoming
- Work in progress
  - First usable implementations available soon
  - Internet-drafts to be updated before San Francisco

More information

- The "official" HIP site (slightly outdated)
  `homebase.htt-consult.com/HIP.html`
- HIP mailing list
  `lists.freeswan.org/mailman/listinfo/hipsec/`
- Ericsson Research HIP project
  `www.hip4inter.net`
- Helsinki University of Technology student project
  `gaijin.tky.hut.fi/hipl/`
- Boeing Phantom Works Linux implementation
  `Available upon request, see mailing list`
- Andrew McGregor's Python implementation
  `Available as a download, see mailing list`
- These slides: `www.tml.hut.fi/~pnr/publications/`