Naming

15-744
Dave Andersen

Lecture warning

- Think “lots of in-class paper discussion” today

Re-thinking Naming and Binding

- One of the fundamental aspects of network architecture: How do you name services and endpoints?
- Today: IP addresses (hierarchical topological identifiers) and DNS names (hierarchical human-readable names).
  - Loose binding between the two
- What properties might we want in names?

Scope

- Global scope / validity / reachability
  - Why? Transitive use. A says to B “for a good time, call C”. Only works if A and B see the same name for C.
    - Real world example: p2p routing and searching
    - contacting hosts behind NATs
    - Note distinction between a NAT and a firewall, though they actually couple the two

Coupling

- Is the “name” involved in every packet, or is it resolved once into an IP address?
  - DNS -> IP once
  - Consequences:
    - Efficient! Naming gets out of the way
    - Loses control after connection establishment
      - Can't do Mobile IP by just changing DNS names

Control, Timescales, Granularity

- Who can update a name binding?
- What's the timescale of updates?
- What's the granularity of naming?
- Old DNS:
  - DNS admin; human; one or a few machines
- Dynamic DNS
  - Owner of machine; seconds?; one or a few machines

Context

- “Middleboxes” -- NATs, firewalls, etc.
  - Fragmented address space (lots of hosts in private space)
  - No inbound connections
  - May interpose on actual communication (and restrict or break it accidentally)
- Tough evolution
  - IP is popular: blessing and curse!
  - One change to make future change easier?

DHT reminder

- i3 and DOA both suppose the existence of a flat name resolution architecture
  - Massive scale (billions of entries)
  - Fast, Scalable, *secure*, cost effective...
- DHTs *suggest* that we can build such an infrastructure (networking sci-fi...), but
  - Serious security challenges
    - Malicious participants, secure node ID assignment, dropping messages...
  - Let's forget about them for today!

Flat names?

- No semantic meaning: 160 bit numbers
- No hierarchy
  - Nice: Doesn't restrict things it names
  - Doesn't have DNS-like politics
- Assumption: The infrastructure exists
  - And is funded, etc., etc.
- But:
  - Not very user-friendly (probably still want DNS... or Google)
  - Sacrifice locality

Self-Certifying Names

- Common combination:
  - Flat naming
  - Self-certifying naming
- Self-foo?
  - ID = Hash(Public Key)
  - Can verify
    - Ownership, uniqueness, etc.
    - With no external infrastructure binding ID->Key
- DOA uses, i3 doesn't (no apparent reason)

i3: Rendezvous-based

- Tightly coupled: resolve “name” *on every packet.*
  - Think back to Mobile IP discussion
- Some extra tricks that we'll talk about in a second

DOA: Delegation

- Two parts, really:
  - A HIP-like (host identity protocol) Endpoint ID (EID) mechanism
    - Globally unique
    - Resolution infrastructure resolves EID -> IP
  - Other cool stuff for delegating
- Major difference: *i3* sits on every packet
  - Fundamental? trade: Can do more to ongoing connections, increases stretch

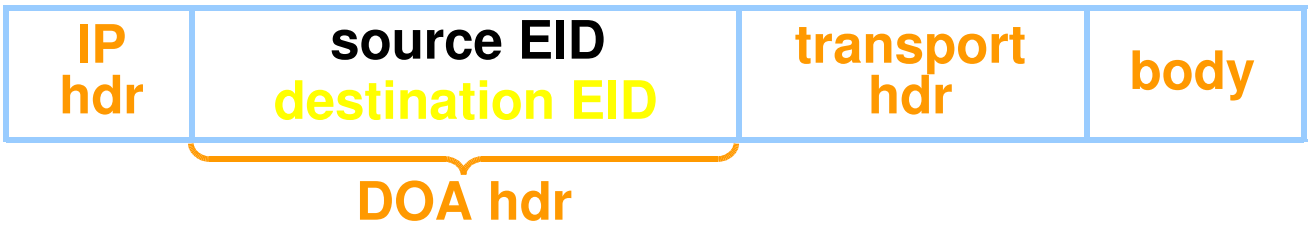
Delegation & Stacks

- Fundamentally different thing:
  - *Stacks* of addresses
- Resolve nameA -> nameB -> nameC -> EID
- i3 throws in a twist:
  - NameA -> {B, C} ...
- (Note that these are used in other contexts and previous systems, but these systems integrate them nicely with Internet. MPLS at below-IP layer.)

Off-Path Firewalls

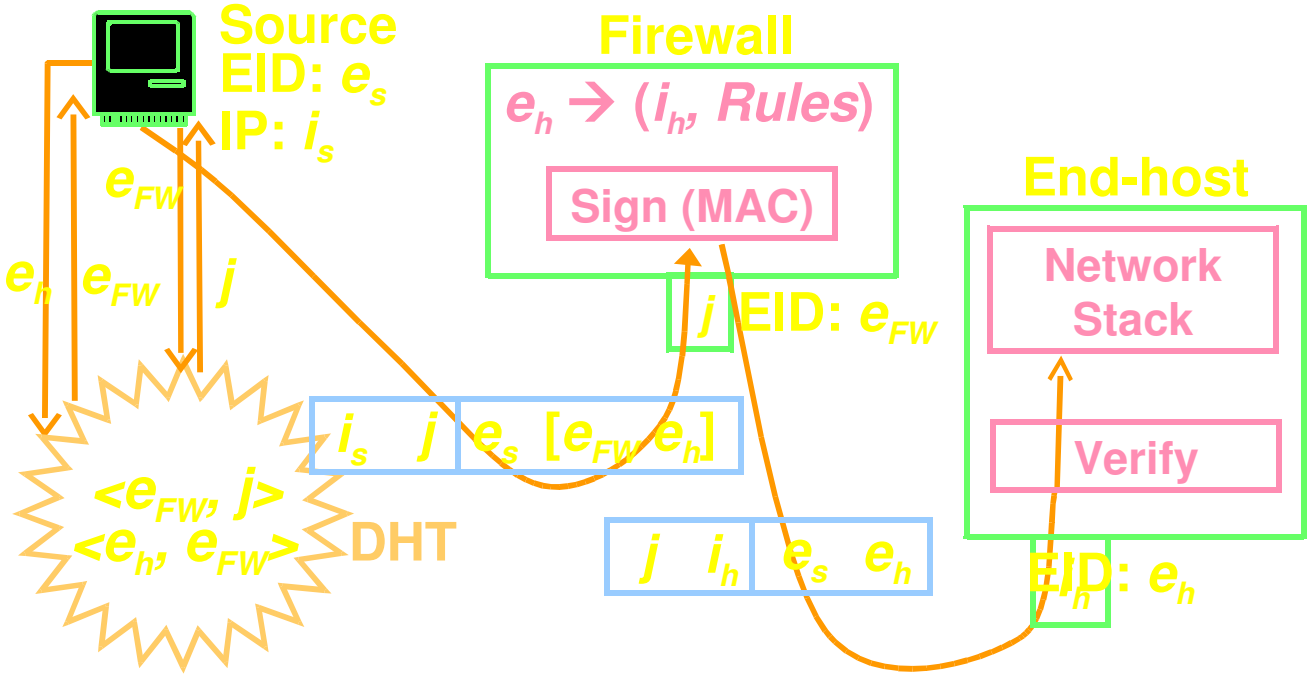
Globally Unique Identifiers for Hosts

- Location-independent, flat, big namespace
- Hash of a public key
- These are called EIDs (e.g., 0xf12abc…)
- Carried in packets



Slide Credit: Michael Walfish

Off-path Firewall



Slide Credit: Michael Walfish (but his is legible on a black background!)

Off-path Firewall: Benefits

- Simplification for end-users who want it
  - Instead of a set of rules, one rule:
  - *"Was this packet vetted by my FW provider?"*
- Firewall can be anywhere, leading to:
  - Third-party service providers
  - Possible market for such services
  - Providers keeping abreast of new applications

• Note: Many things that DOA enables can be done in other ways. Goal is a unified mechanism. (slide: mostly walfish)

DOA and i3 can...

- Service composition
  - Dave -> { transcoding proxy, Dave's EID }
  - Transcoding: Transformation and encoding changes, e.g., downsample images for handheld

i3: Dealing with Interposition

- Aggressive caching of DHT nodes
  - Only do log(N) hops on first lookup; 1-indirect after
- An i3 node must be on path to have all of the i3 benefits
  - Private triggers
  - Like picking an EID for a connection
  - Goal: Reduced stretch (could even do RON-like tricks)

Mobility via Interposition

- Mobility
  - Easy! Update trigger
    - Like mobile IP, really, but with ability to negotiate an intermediary near the path\

i3 multicast

- All members register trigger with same id
  - (security?)
- For scale, must use a hierarchy of triggers
  - (a -> {b,m} b->{c,z} ...)
  - Getting it right is pretty complicated
  - But it's a cool primitive to think about
- Anycast

So?

- Flat naming is an interesting hammer
  - Very worthwhile exercise to consider, even if not adopted
- Many *serious* challenges were one to really use i3: security, real usability, the real costs of all that indirection...
- Fewer with DOA, but many of the same security issues
- Worth it?