

What Identifiers Do We Need

Scott Brim

RRG @ IETF 74 SFO

Acknowledgments

- The Dagstuhl participants
- Harald Alvestrand
- Brian Carpenter

Motivation 1: Identifiers

- Weaning identification from location-dependent information opens up possibilities
- We could name new entities and enhance communication about a connection's intentions
- What needs identifiers? Where do those identifiers need to be available for use?

Motivation 2: “stack ID”

- Some people in Dagstuhl liked “stack ID”:
 - “A stack ID identifies a functional instance of an IP protocol module and the protocols on top of it.”
- Didn’t determine attributes except that it is persistent
- I have yet to see the usefulness → discussion
- Identifiers are needed for different purposes
 - Not clear which of those “stack ID” covers
- Our architecture needs to consider all identification needs that loc/id separation impacts.

When do we need identifiers (between IP and application)?

- **Discovery** to get detailed info including “where”
- **Connection** to the packet consumer you want
- **Authentication**
- **Authorization** (ffs)
- **Session control**
- **Referrals**
and
- **Abstractions**

Example endpoint scenarios

- HIP
- Trilogy Multipath TCP
- MIP
- HTTP
- Add your favorite scenario here

-> what identifiers with what attributes

HIP

- Discovery: FQDN -> Locator and HIT
 - HIT persistent, locator temporary
 - can be two lookups
- Connection: Locator, HIT, protocol, port
 - HIT persistent, not necessarily any others
- Authentication: HIT and trusted 3rd party
 - all persistent
- Session control: HIT
- Referral: FQDN or HIT persistent, not other

Trilogy Multipath TCP

- Discovery: FQDN -> Locator
 - temporary
- Connection: Locator, session ID, protocol, port
 - all temporary
- Authentication: ?? 3rd party (Erik?)
 - something persistent
 - devolves to something like HIP with extra session ID
- Session control: session ID (temporary)
- Referral: tuple (some persistent, some not)

MIP

- Discovery: FQDN -> Locator (HoA)
- Connection: HoA (locator), protocol, port
- Authentication: based on HoA, server HA
 - HoA (locator) must persist to be used as identifier
- Session control: HoA (identifier), HA
- Referral: HoA

- HoA assumed persistent for all uses

HTTP

- Discovery: URI -> Locator
- Connection: Locator, protocol, port, FQDN (some persistent)
- Authentication: server CERT, client login (persistent)
- Session control: cookies (temporary)
- Referral: URI (persistent)

What is persistent?

- FQDN or URI
 - even they can be volatile (conficker)
 - Used only in Discovery and Referral
 - particularly not in session control
- *Sometimes* HoA (locator/identifier), protocol and port persist
- A&A require other stable identifiers

How Many Identifiers?

- HIP:
 - FQDN, protocol, port ... HIT
- Multipath TCP:
 - FQDN, protocol, port ... session ID, auth ID
- MIP:
 - FQDN, protocol, port ... HoA
- HTTP:
 - FQDN, protocol, port ... cookie, login, URI

Endpoints Already Have Names

- Different entities communicate at different layers
 - IP \leftrightarrow IP, transport \leftrightarrow transport, ...
- A lower layer may carry an identifier of an upper layer entity for reference, but does not address it directly
- After establishment, a particular identifier is used by peers at that level, in that context.
 - e.g. protocol, port

Are Referrals Special?

- A referrer may need to give detailed flexible, variable information, optionally including locator, FQDN, URI, protocol, port, pw ...
- Given scoping turmoil, what can a referrer assume is stable?
- What can a referrer require a referee to look up itself, based on the reference?

Abstractions 1: Shared Machines

- Does a peer need an identifier for a virtual host accessible only on some interfaces?
- That is not the peer's business. It is told how to send packets. The nature of the recipient is the recipient's business.
- The name for the entity reachable on a set of interfaces is just like the name for any other endpoint.

Abstractions 2: Spanned Machines

- Receiver is an entity on multiple (virtual) machines.
- A stack ID could name the abstraction.
Receiver diverts internally based on stack ID?
- Currently packets are diverted based on *whole packet*, not just destination header. This will not change.

Stack IDs

- “A stack ID identifies a functional instance of an IP protocol module and the protocols on top of it.”
- If it is persistent, it *might* match HIP and MIP. Every scenario is different.
- Communication endpoints already have names ... in their contexts.
- Is it useful to refer to a “stack”?
 - Or, continue to refer to communicating entities by their names in their contexts
 - We need lots of flexibility
 - Philosophy of ignorance: make room for the future
- Discuss.