

Scalable Site Multihoming and the separation of Identifiers and Locators

erik.nordmark@sun.com

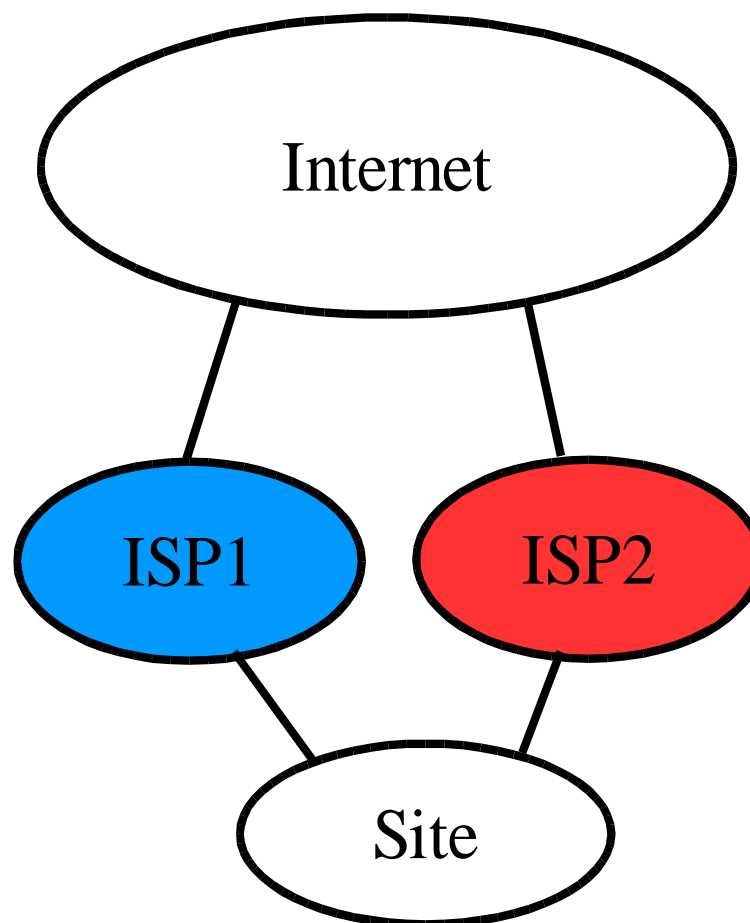
A distillation of the ideas of many people

Outline

- The scalable site multihoming problem
- Big questions:
 - Separate Identifiers and locators?
 - Is a new ID name space needed?
- Detailed questions:
 - Connection rehomeing
 - Ingress Filtering implications
 - Path selection
 - Failure detection
- Security
- Where you can help

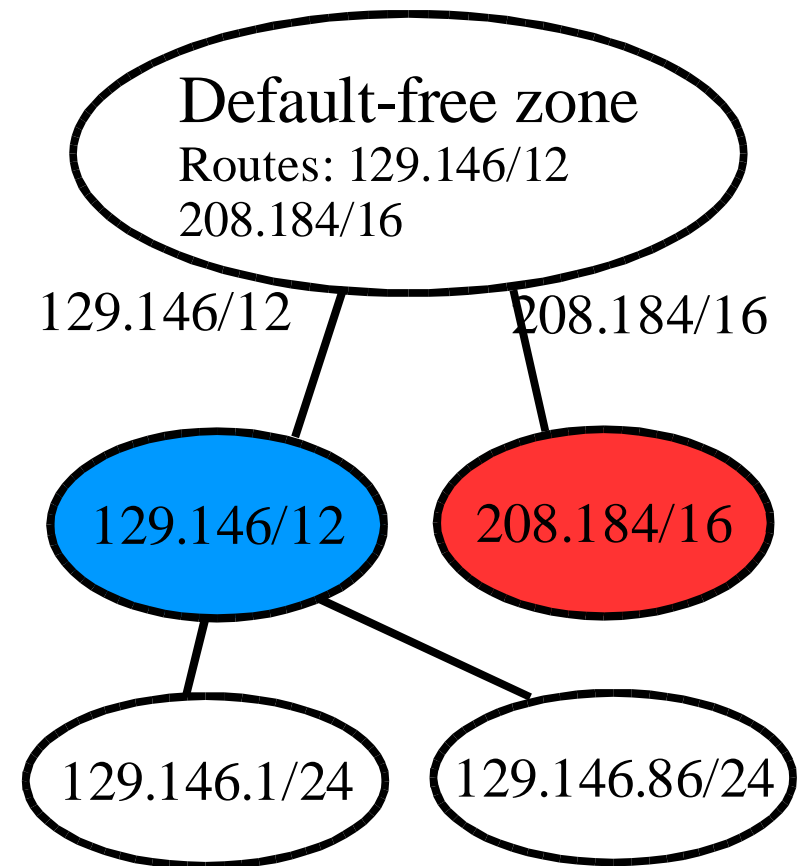
Site Multihoming: What and Why

- Connect site to multiple ISPs
- Improved failure resilience
 - Including total ISP failure
- Load balancing
- Better quality connectivity to customers of the connected ISPs



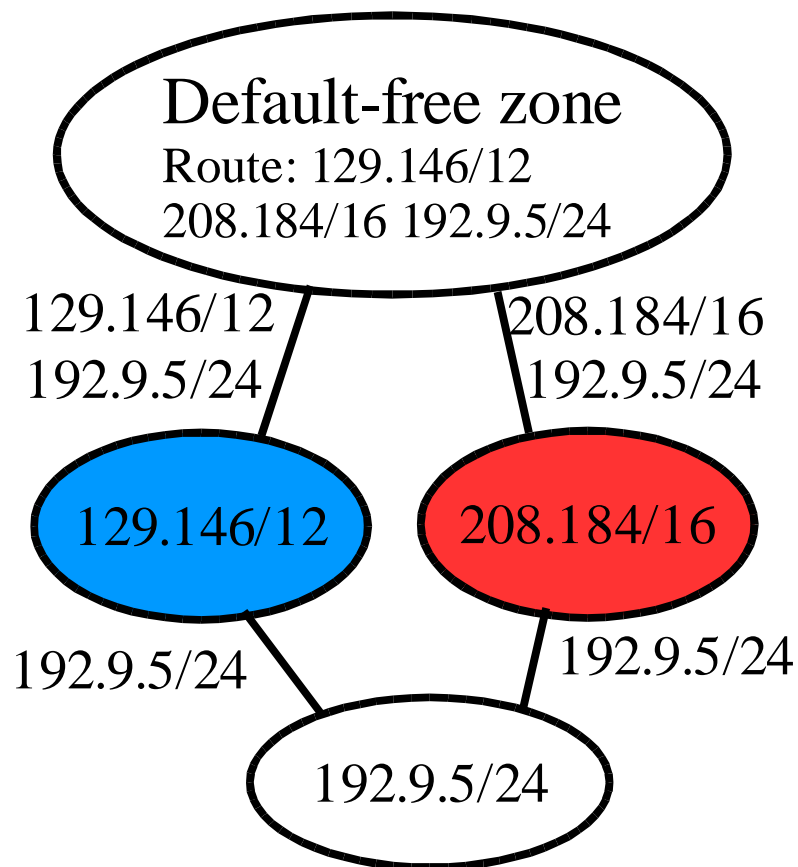
Today's Internet

- Normally sites are assigned addresses from their ISP
 - Addresses come from ISP's address space
- Routes aggregated by ISPs
 - DFZ has one route per ISP
- Provider Independent Addresses being used in some places



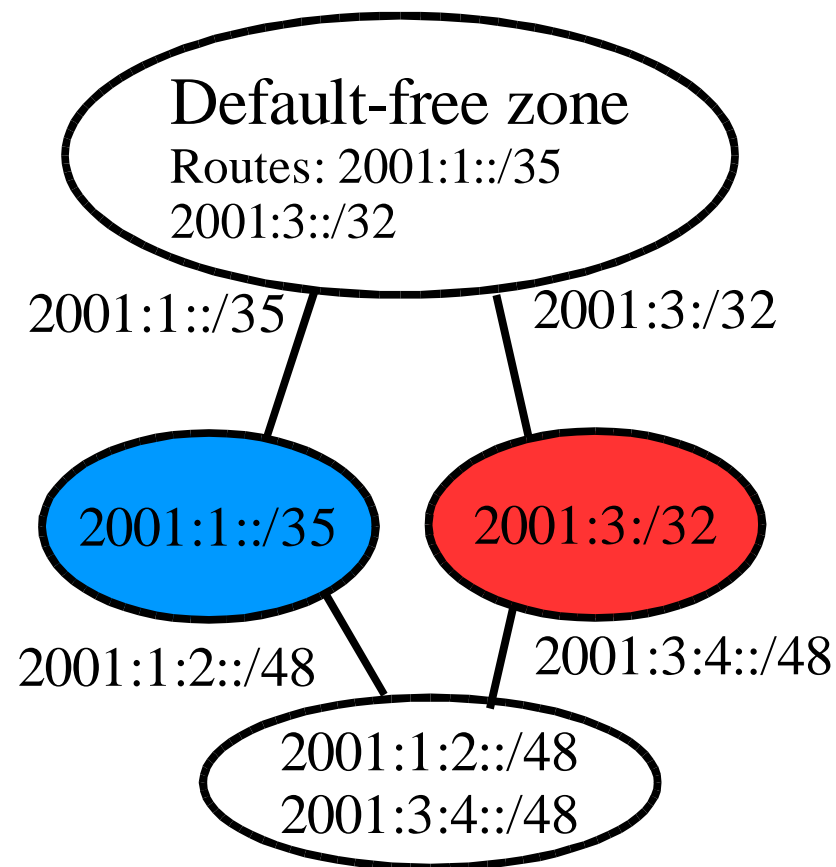
IPv4 Site Multihoming Today

- Multihomed site gets Provider Independent address space
- Routes can't be aggregated by ISPs
 - Fundamental to get resilience
 - One route per multihomed site in the world – doesn't scale
- Variants exist



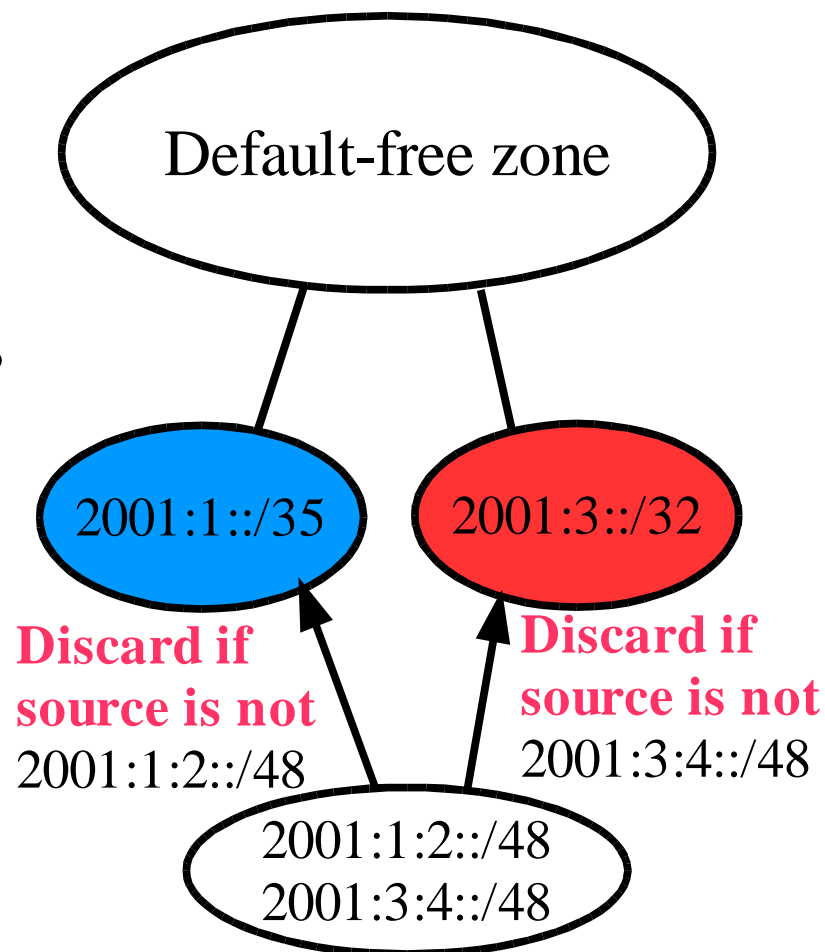
IPv6 Site Multihoming Today

- Could use IPv4 approach
 - But doesn't scale
- Or multiple addresses per host
 - Multihomed site uses addresses from each ISP
 - Each host gets multiple addresses
 - Has limitations



Multiple Addresses per Host

- Resilience provided in the application layer
 - TCP connections do not survive change in addresses
 - Moves the problem to the applications
- Needs address selection
 - RFC 3484 a start
- Ingress filtering issue
- Limited Applicability?



Outline

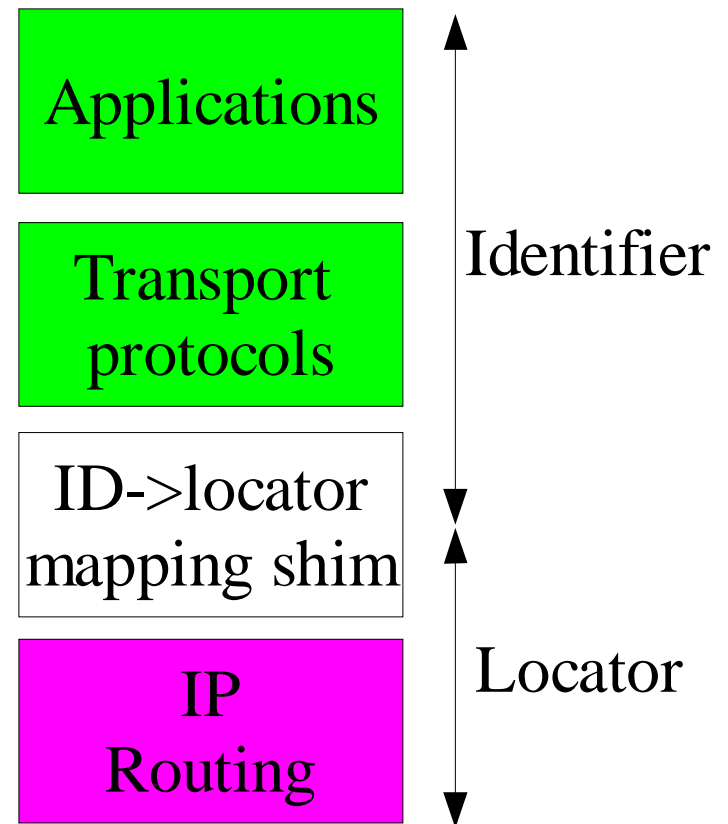
- The scalable multihoming problem
- **Big questions:**
 - **Separate Identifiers and locators?**
 - **Is a new ID name space needed?**
- **Detailed questions:**
 - Connection rehomeing
 - Ingress Filtering implications
 - Path selection
 - Failure detection
- Security
- Where you can help

Key Question: Separate Identifiers and Locators?

- IP addresses used for
 - Identifying purposes (“who”; in TCP connections)
 - Locating purposes (“where”; lookup routes)
- Intentionally designed overloading
 - To avoid a system that maps between them
- Time to separate into identifiers and locators?
 - Current IP addresses would be locators
 - Must add a layer of indirection between the applications/ transports and the routing system
- Doesn't imply a new ID name space

Possible Approach to Separation

- DNS, apps, transports operate on IDs
 - ID handle could be 128 bits
- IP operates on locators
 - Multiple locators per host
- New layer on hosts map between identifiers and locators
 - Shim layer above IP
- Need protocol to setup mapping on hosts



New Layer of Indirection

- Transparent to (most) applications
 - Referrals and callbacks might need to be done differently
 - Doesn't affect existing IPv6 APIs
- Possible to lookup identifier to find locators?
 - Depends on type of Identifier
- Exact location of the shim?
 - Above IP routing layer
 - Below IP endpoint layer (fragmentation, reassembly, IPsec)

Key Question:

A new ID namespace or not?

- Could add the layer of indirection without a new name space
 - Fully Qualified Domain Name as the ID?
 - Sets of locators as the ID?
 - Ephemeral ID tied to a purpose-built key?
- New Identifiers could be long-term stable
 - Independent of ISPs used
 - Could have other properties, help with security, etc.
- What benefits do they have?
 - Application usage?
 - Requires careful understanding of requirements

What could new ID look like?

- Allocate using managed hierarchy?
 - Rooted ID space akin to domain name space and IP address space
 - Provides globally unique identifiers
 - Simplifies mapping from IDs
 - Requires defining allocation hierarchy and control of the root
- Or self-allocate?
 - Each host forms a private, public key pair
 - ID = hash of host's public key
 - Provides probabilistically unique identifiers
 - Harder to build mapping system from IDs

The Mapping System

- Allow applications to lookup identifier to find locators
 - Needed for application referrals in e.g., multi-party applications
 - Needed for callbacks
- We only know how to make this scale for hierarchical name spaces
- Can it be made to scale in the self-managed case?
 - Research idea: distributed hash table, ID is hash of site public key + hash of delegation certificate to host

Outline

- The scalable multihoming problem
- Big questions:
 - Separate Identifiers and locators?
 - Is a new ID name space needed?
- Detailed questions:
 - Connection rehomeing
 - Ingress Filtering implications
 - Path selection
 - Failure detection
- Security
- Where you can help

Connection Rehoming

- Able to change locators
 - Without affecting transports and applications
 - Locators on the wire different than what the applications/transport see as IDs
- Needed whether or not a new ID name space is used
- Need a protocol to “rehome” the traffic
 - Easy except for security issues – just pass the new locator to the peer
- Multi6 WG has many proposals for this

Ingress Filtering

- ISPs can be told of all the address prefixes
 - ISP A allows ISP B's addresses as source
 - Unlikely for the consumer-class service
- Different approaches
 - Make packets take exit path which matches the source address of the packet – source-based routing
 - Border routers inform the host of the source address prefix to use with the destination
 - Or ID/locator separation so that source locators don't matter to the receiver
 - Allowing the source locators to be rewritten by routers along the path instead of packet being filtered out

Path/Locator Selection

- The shim layer needs to select the source and destination locators to use
 - When communication starts
 - In response to a path failure
- This is largely independent of
 - Whether or not there is a new ID name space
 - Mechanisms used for connection rehomeing
- Is there experience with SCTP which we can leverage?

Path/Locator Failure Detection

- Time constraints are application dependent
 - VoIP traffic vs. web vs. TCP giving up
- Transport protocols can provide hints
 - Knows when there are timeouts
- Role of the routing system?
 - Too slow BGP convergence – build above the routing system?
 - Improve BGP convergence?
 - Receive BAD news faster than GOOD news?
 - As hint to try an alternate locator
- Shim layer could exchange heartbeats end-to-end

Time line/Ambition

1. Something short term to make hosts with multiple prefixes work well
 - Cope with Ingress filtering
 - No changes required on peers
 - Applications see multiple IP addresses
 2. Hide multiple IP addresses without a new name space
 - Provide connection survivability
 3. Introduce a new name space
 - Might be useful for the applications
 - Mapping from this name space is an issue
- Might need 1 for 2 or 3 (details, details)

Security Considerations

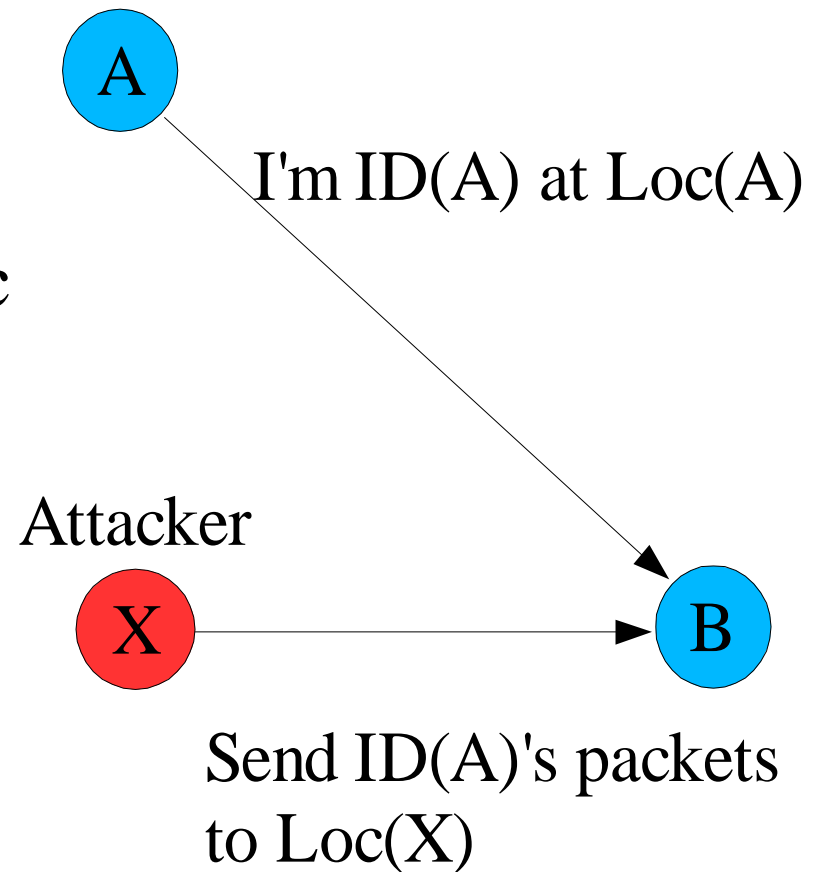
- Not discussed in this memo

Security Ambition

- The new multihoming protocols should not make the Internet less secure than today
- As things like DNS and BGP are secured, we don't want the multihoming protocols to become the weakest link
- If there is some end-to-end infrastructure like PKI for IPsec, would like to take advantage of this to minimize threats

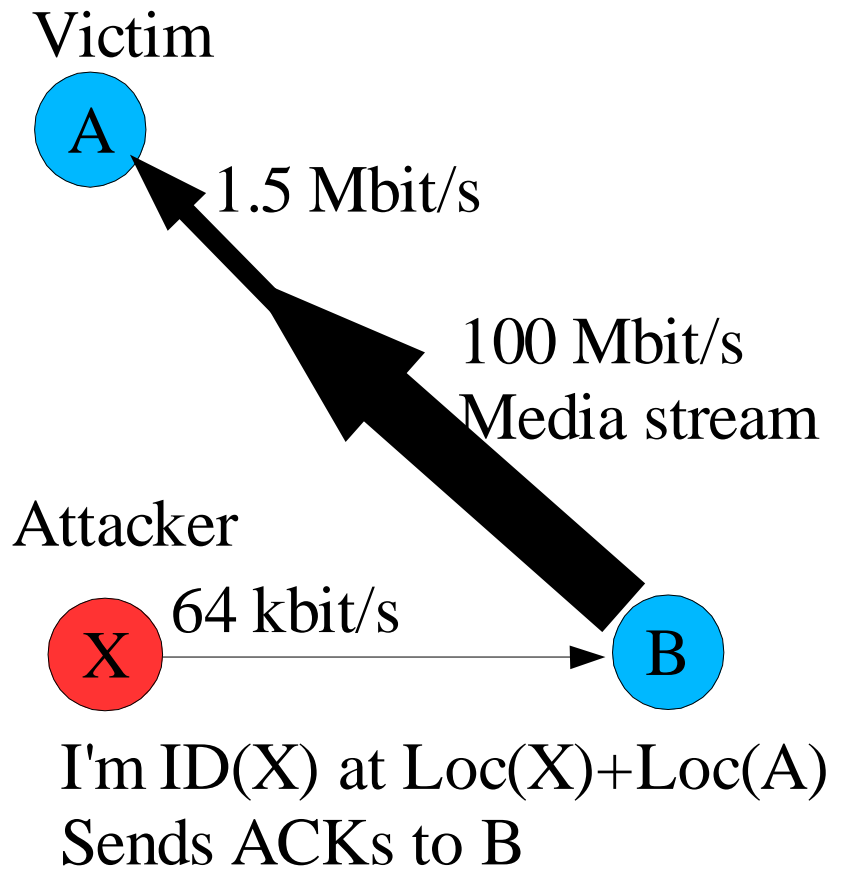
Security Threat: Redirection Attack

- Connection rehoming implies changing locators
 - But can be used to maliciously redirect traffic
- If (IPsec, TLS, etc.) used, attack limited to (selective) Denial-of-Service attacks
- Solution?
 - Verify identity is A



Security Threat: 3rd party flooding

- Today flooding capability of attacker is limited by its bandwidth
- Being able to specify the locator of a 3rd party changes this
 - Fool B to send packets to A
 - Amplification
- Solution?
 - Need to verify that X “owns” $\text{Loc}(A)$, or that X is present at $\text{Loc}(A)$?



Security Approaches

- Don't depend on some not-yet-deployed infrastructure, such as a global PKI
- Different approaches have been suggested
 - Use existing DNS – forward + reverse lookups
 - Define the ID as a hash of a public key – use public-key crypto to prove ownership of private key
 - Leap-of-faith (like ssh first time) to create some shared secret – perhaps this is too weak?
- Security implies more work for the hosts
 - Potential for introducing new DoS attacks

What about Mobile IP?

- Mobile IPv6 work a baseline for security understanding
- Effectively has a stable locator (home address) and a temporary, smaller latency locator (care-of address)
 - Plus the home address being the identifier
- This can be generalized for multihoming
- Scaling: solutions for **site** multihoming might not scale up to support **host** multihoming and mobility

Current Status

- **MULTI6 WG** www.ietf.org/html.charters/multi6-charter.html
 - Starting architectural analysis
 - Looking at proposals with different time scale and ambition
 - Lots of proposals
- **HIP WG** www.ietf.org/html.charters/hip-charter.html
 - Specifying Host Identity Protocol as experimental RFCs
- **Proposed HIP-related research group**
 - Longer-term research problems

How can you help

- How useful would a new ID space be to applications? What properties would it need to satisfy?
- What are the issues if FQDNs and/or sets of locators are the identifiers?
- Can BGP/IGPs provide faster hints that a locator might not work?
- How to perform e2e failure detection?
- How to intelligently select locators?
- Contribute to multi6's architectural analysis

Credits

- Mike O'Dell's 8+8/GSE – proposing ID/locator separation
- SCTP – first transport protocol handling multiple IP addresses
- Mobile IPv6 – understanding of the threats and weak (return routability) security
- HIP – a worked out approach for a self-managed ID name space and security mechanisms

Questions?

A few observations

- Transport protocols have an internal layer boundary
 - Path-dependent vs. path-independent
- Source locators are almost not needed
 - Return address for 1st packet; ICMP errors
 - Can be used to carry very limited “traceback” info
- ID/locator separation is akin to architected NAT
 - Allows rewriting of locators independent of IDs
- Protocols using multicast already have IDs?
 - Is this only true for RTP?
 - Sufficient to say that multicast uses locators?