Generic Referral Objects

draft-carpenter-behave-referral-object-01

Brian Carpenter
Mohamed Boucadair
Joel Halpern
Sheng Jiang
Keith Moore

November 2009

Status of this draft

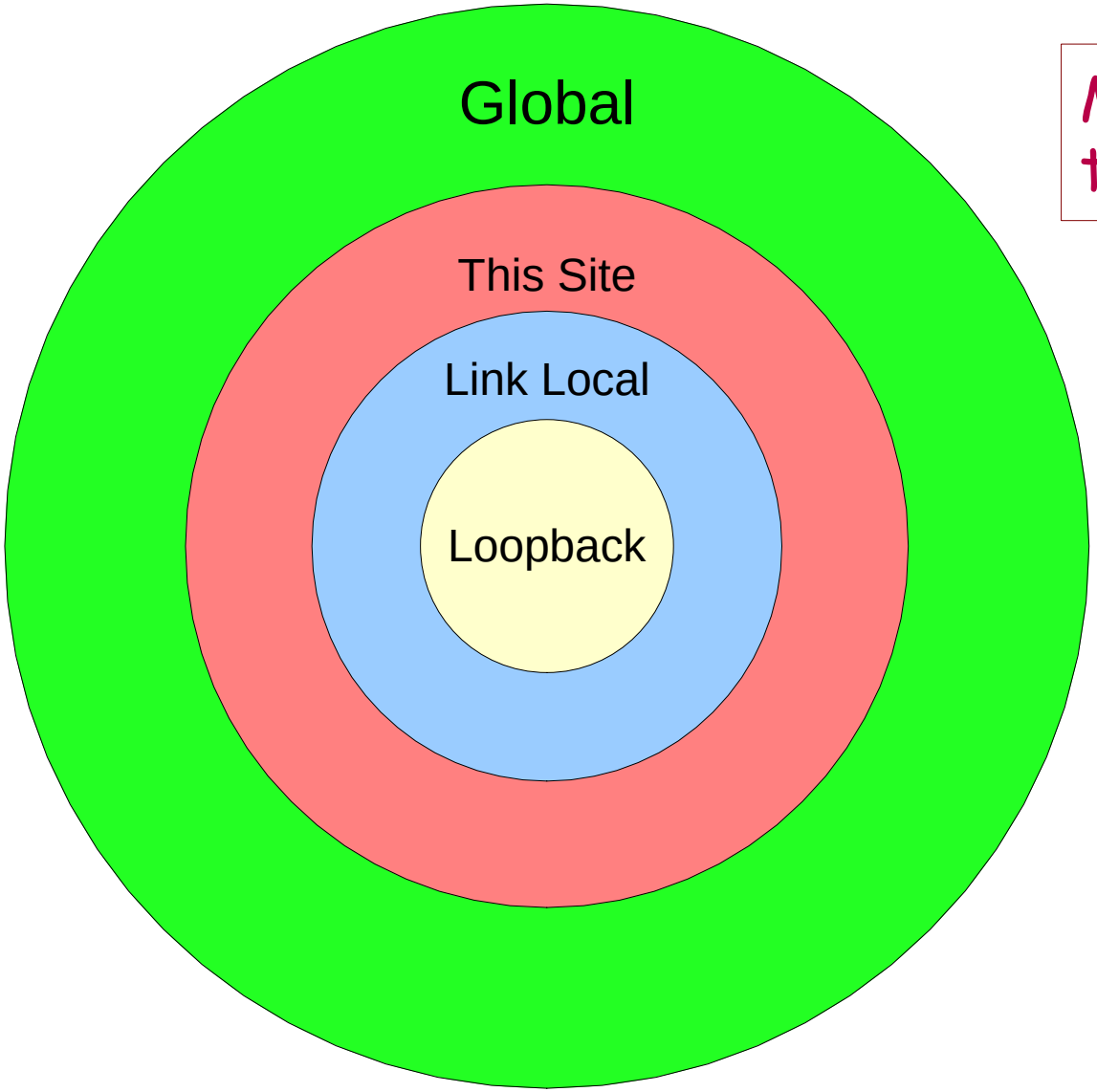
Why not just use DNS names?

- Experience shows that an application cannot reliably use an FQDN to find the address(es) of an arbitrary peer.
- FQDNs work fairly well to find the addresses of servers. But DNS records are not as reliably maintained for arbitrary hosts such as those in peer-to-peer applications.
- An FQDN may not be sufficient to establish successful communications involving heterogeneous peers (i.e. IPv4 and IPv6) .
- An application does not have a reliable way of knowing its own domain name.
- Which is why referrals often use IP addresses.

The problem with simple address referrals

- Entity A needs to tell entity B how to reach entity C
  - “entity” is typically an application instance in a host
- But the address of C viewed from B is not the same as the address of C viewed from A
  - A, B and C are potentially in different addressing scopes separated by NATs, firewalls, VPNs and/or address families (v4/v6)
- Therefore referrals by simply passing an address are liable to fail

Imaginary scopes
(or, each host is the centre of the Universe)

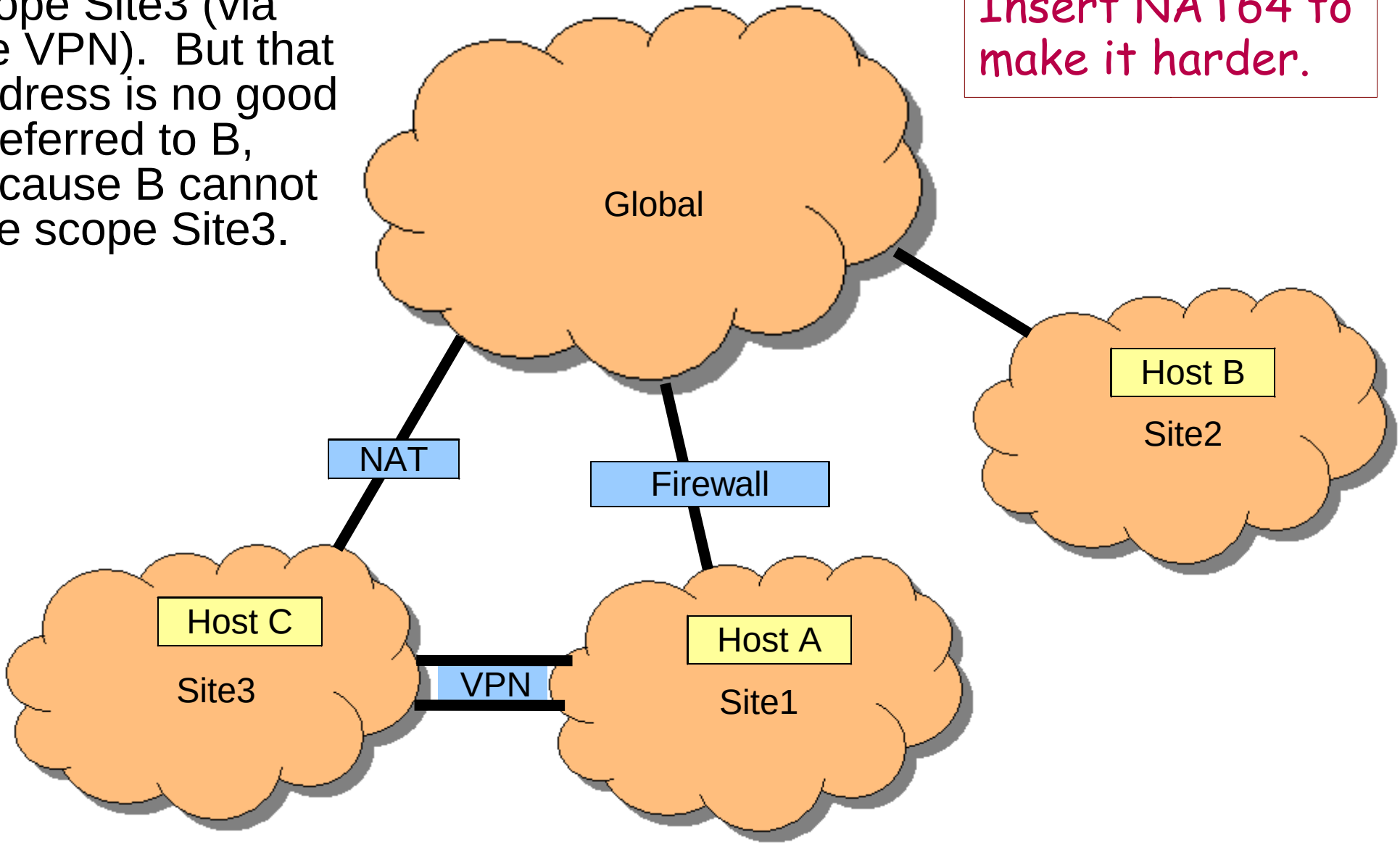


Maybe it was like that in 1995?

Real Scopes

A can see C in scope Site3 (via the VPN). But that address is no good if referred to B, because B cannot see scope Site3.

Insert NAT64 to make it harder.



Identifying real scopes

- We need to define a better way of identifying address scopes
  - “site-local” and “global” don’t capture the A-B-C problem
  - In particular, you’d need to know which site was relevant
  - Tunnels or VPNs can join scopes in arbitrary ways
  - Scopes can overlap
- Naming the scopes is the only way to make this explicit.

Names for address scopes

- We consider that a scope can be:
  - Null (e.g. loopback)
  - Link-local
  - Limited (e.g. VPN, behind NAT, RFC1918, ULA, DMZ)
  - Global
- The entity receiving a referral needs to be able to know whether a limited scope is reachable.
  - This requires the ability to *name* scopes
  - Hosts need to know which named scopes they can reach

Flexibility

- Given that we have at least two different forms of reference already (IP Address and FQDN),
- and that an IP Address is actually two different types itself (IPv4 and IPv6),
- and that folks tend to invent new ways of talking about entities or applications
  - (HIP identities are an example),
- it seems necessary to handle more kinds of references than just the obvious ones.

Multiplicity

- Since the sender may not know which type of reference the receiver of the referral can best use, it should send as many as it knows accurately.
  - Any or all of IPv4, IPv6, FQDN, HIT, HIP ID... that it actually knows
- Since there may be multiple possibly applicable scopes, and again the sender cannot know which apply to the receiver, it should send information for all the scopes it knows.

Solution approach

- Define a standardised abstraction known as a *Generic Referral Object (GRO).*
- Assume for the purposes of this talk that we have a namespace for address scopes (*ScopeID*)

GRO strawman

- In the draft, we describe a binary GRO format as a sequence of optional TLVs
  - Some TLVs are references; others can qualify them
  - In particular, *ScopeID* can qualify an address
  - Intentionally not giving any details today

GRO sender's job

- To construct the most complete GRO it can from what it knows about the referenced host, i.e. always include all known addresses and FQDNs, with all known qualifiers such as lifetimes
  - While respecting privacy and security policies that are known and apply to the sender.
- Where an address is known to have limited scope, supply the *ScopeID*
  - Therefore, the sender needs to be aware of the *ScopeID* for each correspondent address (for example, use the site's *ScopeID* for RFC1918 addresses or ULAs)

GRO receiver's job

- To interpret the data in the GRO appropriately before trying to contact the referenced host
  - For limited scope addresses, check whether the *ScopeID* is known to be reachable
    - Therefore, the receiver needs to be aware of the *ScopeID*s it can reach
  - If not, look for something else useable in the GRO, such as an FQDN or HIT or HI.

Questions? Discussion?

- Note that the draft goes into quite a bit more detail, but the first question is whether the idea has any merit.
- Acknowledgement: there is much history that we have learned from, including multiple application efforts and TURN / ICE.