Generic Referral Objects

draft-carpenter-behave-referral-object-00

Brian Carpenter
Mohamed Boucadair
Scott Brim
Joel Halpern
Sheng Jiang
Keith Moore

July 2009

Status of this draft

Problem

- Entity A needs to tell entity B how to reach entity C
  - “entity” is typically an application 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
- Therefore referrals by simply passing an address are liable to fail

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

Flexbility of referal form

- Given that we have at least two different forms of reference already (IP Address and FQDN)
  - And an IP Address is actually two different types itself (IPv4 and IPv6 addressess)
- Given that folks tend to invent new ways of talking about entities or applications
- It would seem necessary that any mechanism handle more kinds of identities than just the ones we can obviously see
  - HIP identities are another relevant example.

Solution approach

- Define a standardised abstraction known as a *Generic Referral Object (GRO).*
- To do that, we first need to define a better way of dealing with address scopes
  - “link-local”, “site-local” and “global” don’t capture the A-B-C problem
  - In particular, you’d need to know which link or site was relevant
  - VPNs can join scopes in arbitrary ways

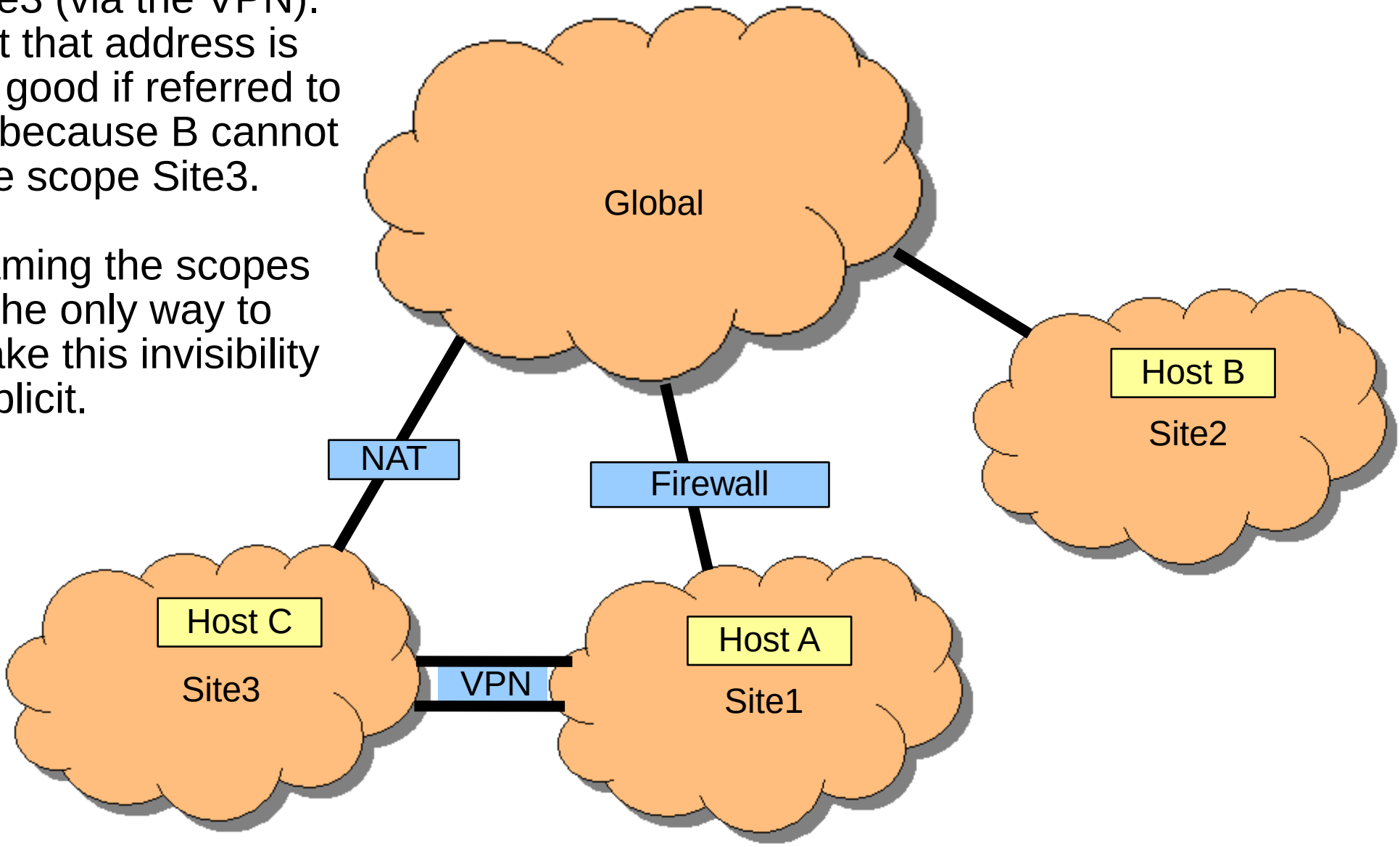
Names for address scopes

- 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
  - Let’s skip the details of Scope IDs for now

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.

Naming the scopes is the only way to make this invisibility explicit.



Multiple references

- The preceding implies that when sending referral information, a sender needs to send multiple pieces of information.
- Obviously, it can only send what it knows.
  - How a referrer gets that information is out of scope for this draft
- The referrer may have policy or security restrictions on what kinds or scopes of information it can send
  - This is not the target or subject policy, but the sender’s policy

Kinds of 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, ...
  - That it actually knows
- Since there may be multiple possibly applicable scopes, and again the sender can not know which apply to the receiver, it should send information for all the scopes it knows.

GRO strawman (1)

- A GRO is a sequence of optional TLVs
  - Some TLVs are references; others can qualify them
  - Reference TLVs:
    - IPv4_address
    - IPv6_address
    - FQDN
    - HIT
    - HI (HIP identifier)

GRO strawman (2)

- Qualifier TLVs
  - IPv4_mask
  - IPv6_mask
  - Ref_lifetime
  - Ref_source (configured/DNS/DHCP/SLAAC/relayed/translated)
  - Ref_scope (null/link/limited/global)
  - ScopeID
  - Port_number
  - Transport_protocol
  - Port_source (direct/relayed/translated)

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 ScopeIDs 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.