Searching for the scaly NAT-eater

(draft-carpenter-referral-ps-02, 2010)

Brian Carpenter
Sheng Jiang*
Bo Zhou**
Zhen Cao**

Manis tetradactyla munching a NAT
Wikimedia Commons

*Huawei
**ChinaMobile

Introductory remark: Internet infrastructure still matters

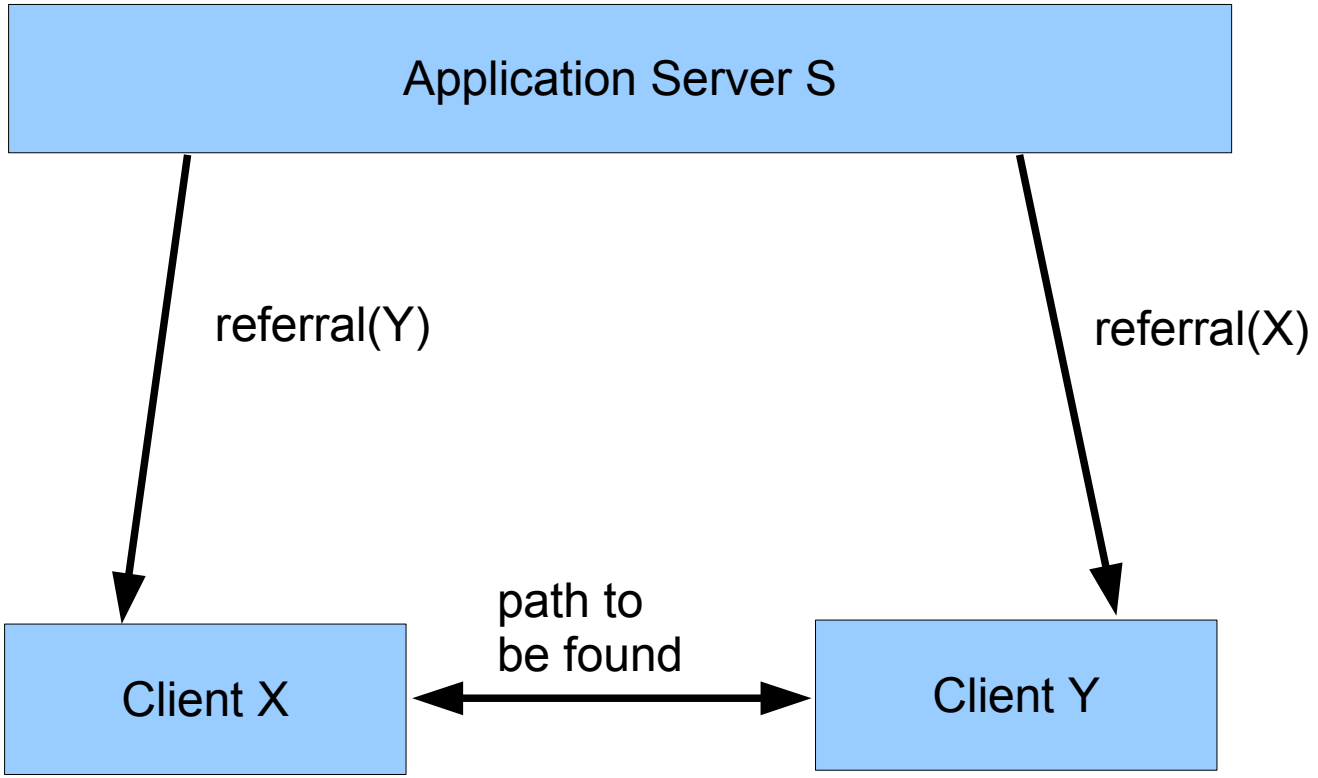
- On 2012-05-06, Hannes Tschofenig wrote to the IETF:

  "you will not find interest from young engineers to work on 10 year old topics. You can try it yourself: give a talk at a university and see the reaction from the students. Pick a lower-layer topic and a topic from the application layer (some Web stuff)."

- As in any major technical system, neglect of the infrastructure is a Very Bad Idea. Consider what happens to a city if it ignores the sewers. The IETF (and the operators who read RFCs) are in the same position as municipal utilities. It's hard to get students interested in sanitary engineering.

Referral example: the trapezoid scenario

- One server, two clients, three paths



Definitions

- Referral: the act of one entity informing another entity how to contact a specific entity.
  - A tells B how to contact C
  - or A tells B how to contact A
- Entity: any software component embedded in a host that sends, receives or uses referrals.
  - An entity might migrate between hosts, for load sharing or failover
- Reference: the actual data (name, address, identitifier, locator, pointer, etc.) behind a referral.
- Scope: the region(s) of the Internet within which a given reference is applicable to reach the referenced entity.

Goals of a referral

- The principal purpose of a referral is to enable one entity in a multi-party application to pass information to another party involved in the same application.
  - No assumptions about whether entities act as clients, servers, peers, super-nodes, relays, proxies, etc.
  - No assumptions about how entities become aware of the need to send a referral; this depends on the application.
  - Referral does not guarantee reachability, since the referring entity has no general way of knowing which paths exist between the receiving entity and the referenced entity. Path selection isn't part of the referral problem.

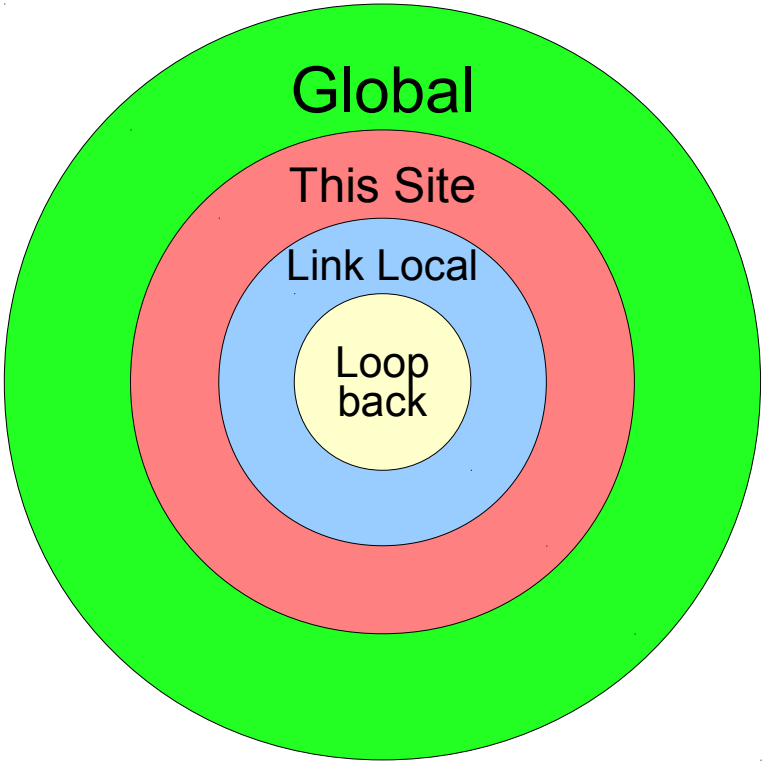
Problem statement (1)

- Referring an IP address often fails in today's Internet.
- Cannot assume that an address by which you reach a host from location A also works from location B.
  - IP addresses no longer all have global scope, they often have limited reachability, and may have a limited lifetime.
  - Can no longer assume that a host with a fixed location has a single fixed IP address, or even a stable IP address.
  - A public IPv4 address often no longer identifies a single customer/user/host, without knowing the port number.
  - A private IPv4 address is meaningless out of the private network.
  - Addresses and port numbers may be different on either side of a NAT, and firewalls may block them.
  - The Internet has two address formats (IPv4 and IPv6).

Problem statement (2)

- IP addresses today may have an implied "context" (VPN, VoIP VC, IP TV, etc.): the reachability of such an address depends on that context.
- Thus there is no clean definition of the scope of an address (especially an IPv4 address, due to the prevalence of NAT).
  - It is impossible to determine algorithmically the scope of reachability of an address by inspecting the bits.
  - Resolving the scope problem would greatly clarify the general problem of referrals.

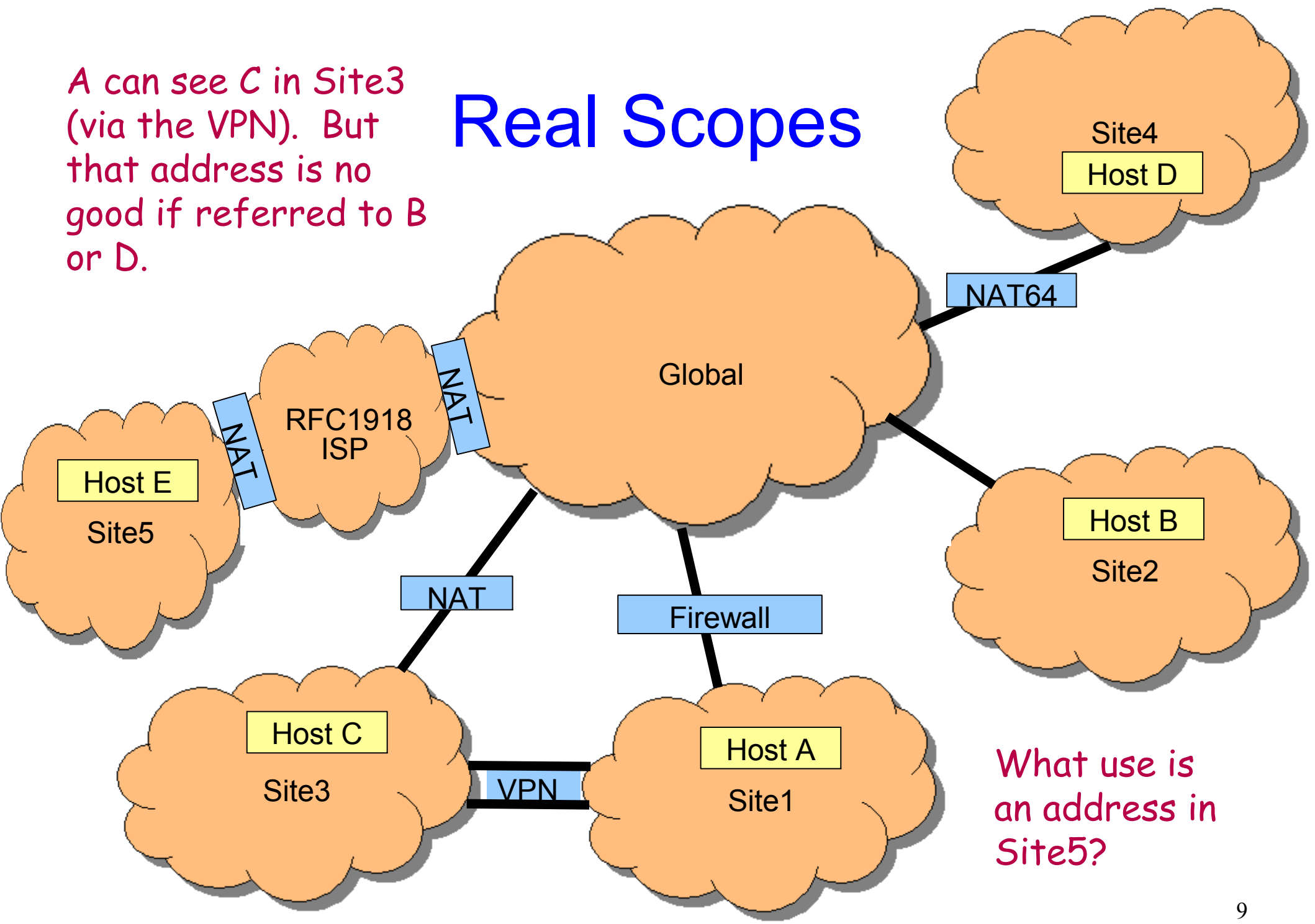
Old view of scopes
(each host as the centre of the Universe)



This is the view of scope assumed by the (IPv6) socket API, but it has been far too simple for at least ten years.

A can see C in Site3 (via the VPN). But that address is no good if referred to B or D.

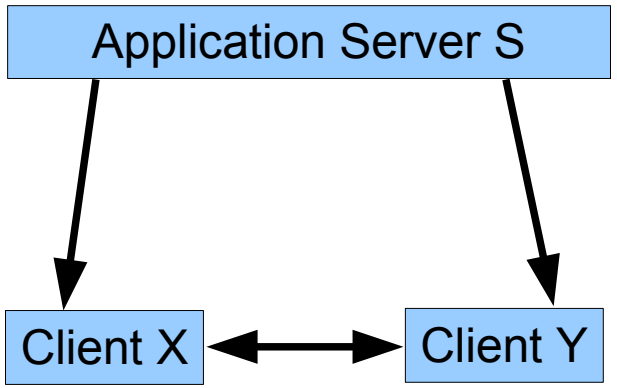
Real Scopes

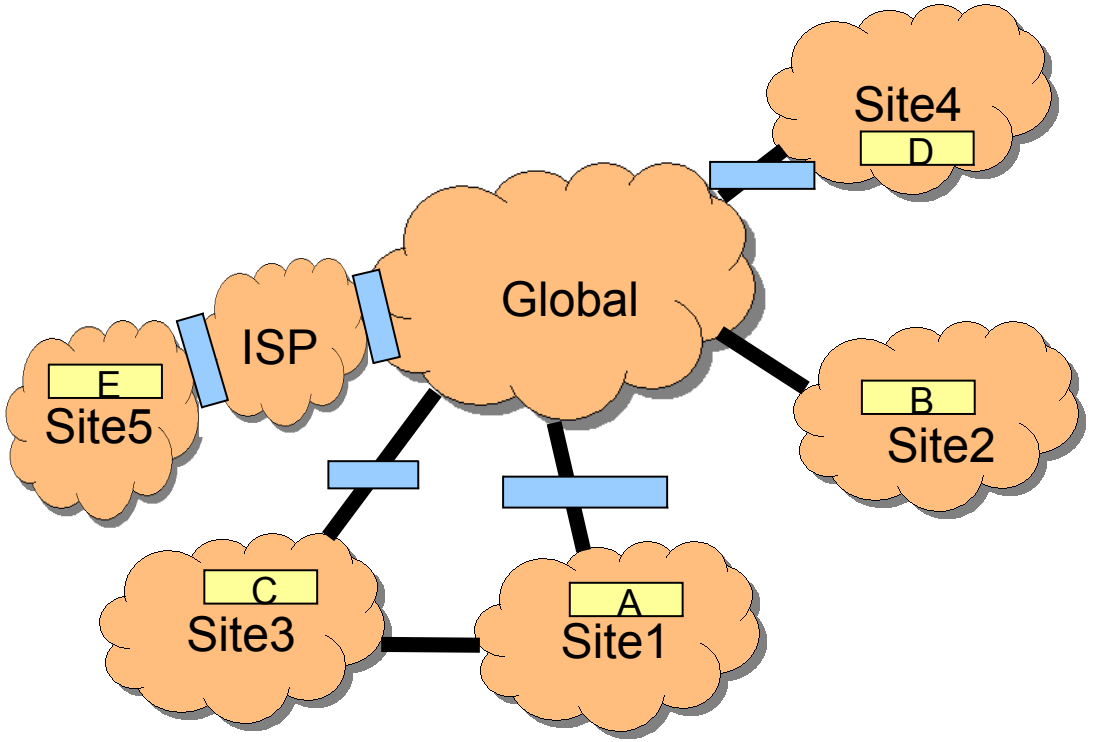


What use is an address in Site5?

Exercise for the Reader

- Take S, X and Y from slide 3 and map them each possible way onto A, B, C, D and E on slide 9.





Problem statement (3)

- DNS names are not enough
  - Applications 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 may not exist for arbitrary hosts (such as subscribers).
  - FQDN isn't used by existing p2p applications
    - example: SIP, RTSP, BitTorrent, H.323
    - because endpoints generally have no way to create an FQDN
  - An FQDN may be insufficient to establish sessions involving heterogeneous peers (i.e. IPv4 and IPv6) .
  - An application does not have a reliable way of knowing its own FQDN.

Problem statement (4)

- Neither an IP address nor an FQDN gives complete information about the referenced entity.
  - e.g., lifetime and scope missing
- ID-Locator Split Mechanisms will increase complexity
  - e.g. with Name-based Sockets, if a referral is based on the IP address used at a given instant for a socket, that address might be useless by the time the referral was received, because the socket migrated to a different IP address.

Problem statement (5)

- Application view of all of the problems above is constrained by the socket interface. Most of the complexity is invisible, yet where is the intelligence to deal with that complexity?
- Application developers are left on their own to reinvent special solutions (like ICE), trial and error methods, or heuristics.
- Other application issues?

Conclusion:
A Generic Referral Mechanism is needed

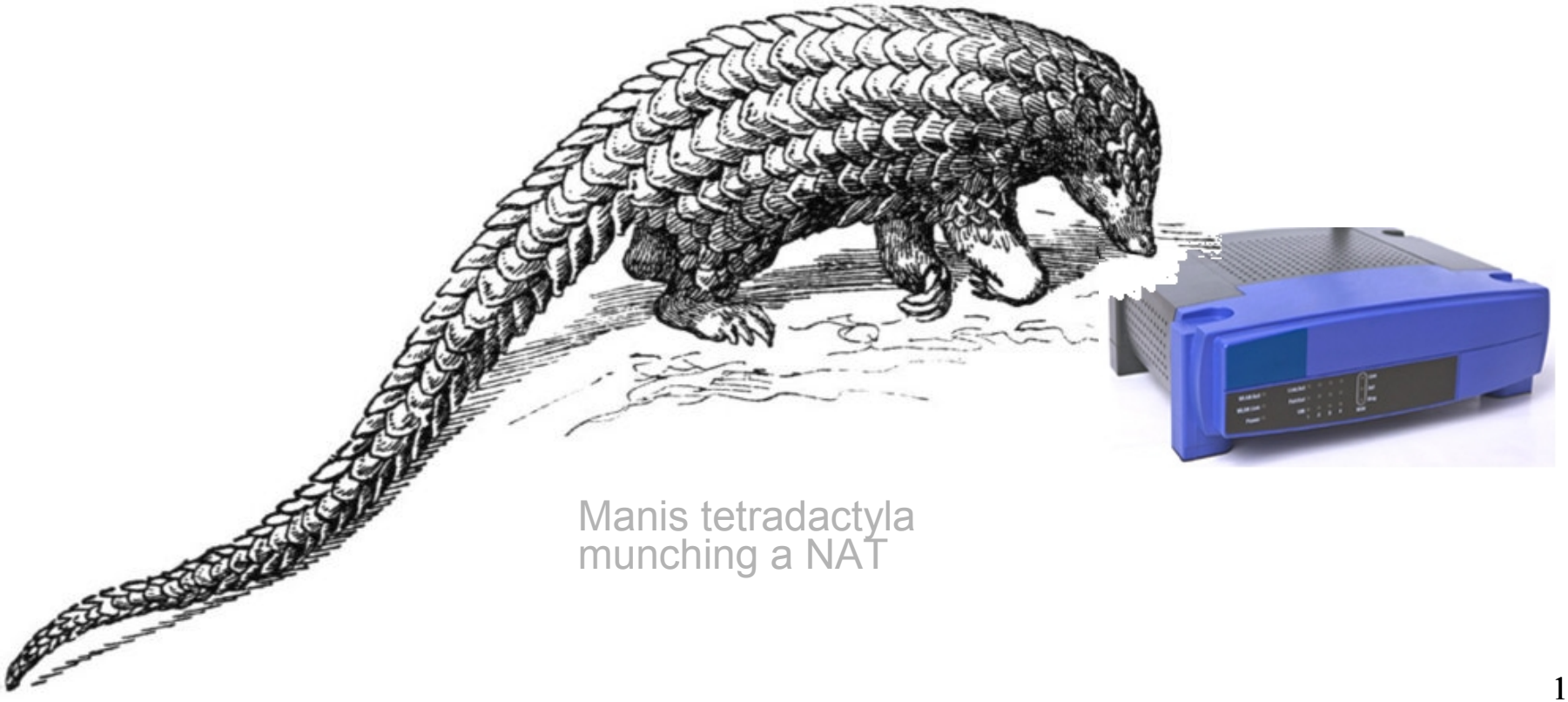
- Motivations for this conclusion:
  - Unless the parties have agreed on the scope, lifetime, and format of the elements in a referral through some other means, that information must be passed with the referral.
  - It may be helpful to the entity receiving a reference to also receive information about the source of the reference, such as an FQDN, to help it recover from any failure.
  - A reference should contain alternatives to an IP address or an FQDN, when any such alternatives exist.
- We also identified the need to define address scope more precisely.
- Partial or application-specific solutions to these problems abound, because any multi-party distributed application must solve them.

Footnote:
Path selection isn't part of the referral problem

- A reference might carry multiple references for the same target. These may lead to multiple possible paths from the receiving entity to the referenced entity.
- The receiving entity will need to make a choice of path, possibly by local policy (e.g. RFC3484) or possibly by trial and error. This choice is out of scope for the referral mechanism itself.
- Complicated by multi-homing and multi-interface scenarios.

Questions? Discussion?

- Acknowledgements
  - There is much history that we have learned from, including multiple application efforts and TURN / ICE.
  - Input from several people, especially Dan Wing.



Manis tetradactyla
munching a NAT