NAT66: IPv6-to-IPv6 NAT

draft-mrw-behave-nat66-01.txt

Margaret Wasserman
mrw@sandstorm.net
Fred Baker
fred@cisco.com
IETF 73, Minneapolis
November 2008

What is NAT66?

- The NAT66 specification defines an IPv6-to-IPv6 Network Address Translation function that:
  - Is considerably less problematic than IPv4 NA(P)T
    - But it doesn't eliminate all of the problem associated with NAT44
  - Requires no per-host or per-connection state
    - Uses two-way, algorithmic address mapping
  - Requires no changes to transport layer headers
    - Uses only 1:1 address mapping, so no need for port mapping
    - Uses checksum-neutral mapping, so no need to change checksum

“I’ve never seen, heard, nor smelled an issue that was so dangerous it couldn’t be *talked* about.”

-- Attributed to Stephen Hopkins, Rhode Island representative to the Continental Congress, “1776”

Motivations for NAT66

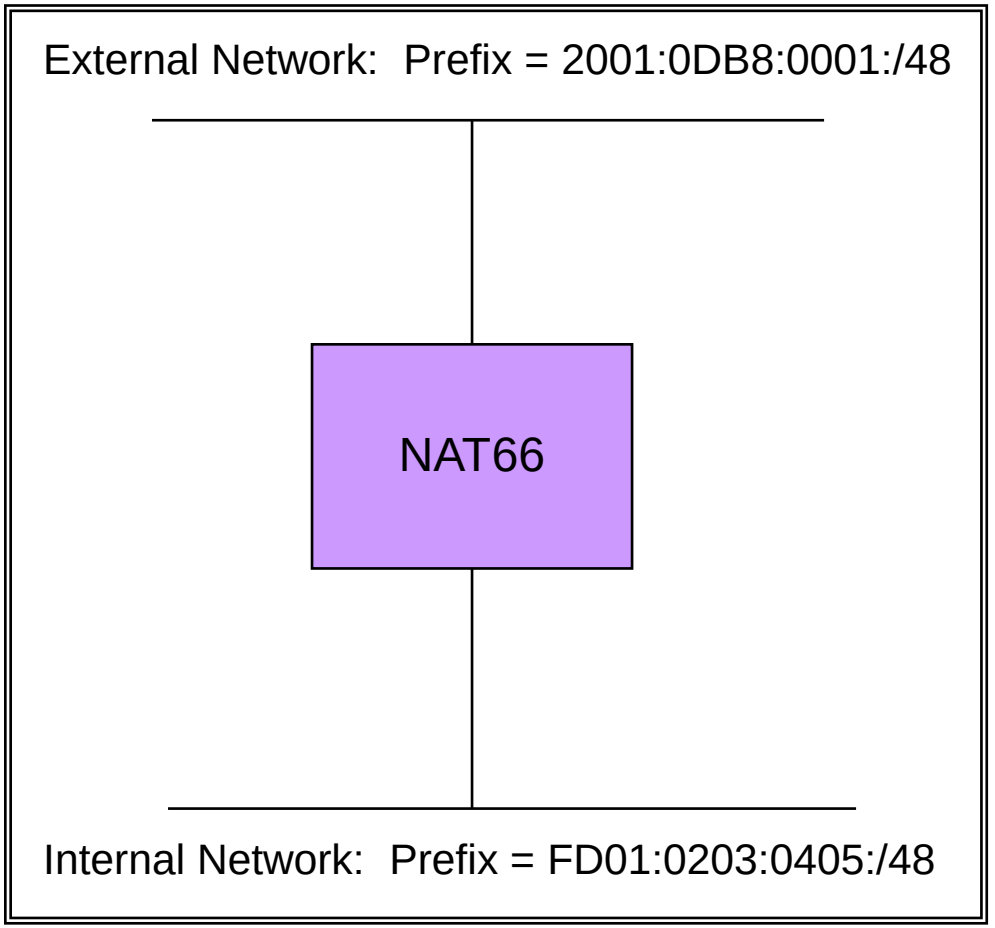
- A few facts..
  - There is demand from enterprise network operators for IPv6 NAT
  - Vendors are implementing IPv6 NAT products to meet that demand
  - There *will* be IPv6 NAT, and the IETF cannot do anything to prevent it
- Therefore, we have two choices…
  - Refuse to document IPv6 NAT, don't offer advice about how to do it
    - Some vendors will simply build IPv4 NA(P)Ts with longer addresses
    - Others will try to make improvements, causing inconsistency
  - Document an IPv6 NAT mechanism (such as NAT66)
    - Share our understanding of how to build a less problematic IPv6 NAT
    - Minimize negative impacts of IPv6 NAT
    - Promote consistency in how IPv6 NATs will work

Striking a Balance

- The document states that the IETF does not recommend the use of IPv6-to-IPv6 NAT
  - Anyone who is considering implementing or deploying NAT66 should first read the IPv6 Local Network Protection document (RFC 4864), and consider alternatives
- However, we understand that some people *will* choose to implement or deploy NAT66 for a variety of reasons
- So, our message could be summarized:

We *do not recommend* that you implement IPv6-to-IPv6 NAT, but if you *do* choose to implement it, do it *this* way!

Simple NAT66 Example



- Only the IP address prefixes are mapped
  - Source prefix on outbound traffic
  - Destination prefix on inbound traffic
- No per-host/connection state on NAT66 device
  - Prefixes configured
- Port numbers and transport checksum are not changed

NAT66 Scenarios

- The draft describes 3 scenarios for NAT66 deployment
  - Leaf network connected to the Internet via a single NAT66 device
  - NAT66 used between two private networks
  - More than one NAT66 device attached to a single network

Mapping Mechanisms

- Two-way algorithmic mapping
  - Checksum correction is performed to make the resulting IPv6 header checksum-neutral (for TCP/UDP pseudo-header checksums)
  - Can be reversed by any system that knows internal and external prefixes and prefix lengths.
- Topology Hiding mapping
  - Version in draft is broken, and wasn't all that great, anyway
  - New version (on later slide) provides cryptographic protection of subnet information and also includes checksum correction.
  - Can be reversed by any system that knows internal and external prefixes, prefix lengths, and the crypto key.
- Both mappings avoid need for per-host or per-connection state on the NAT66 device. Both mappings are checksum neutral.

Two-Way Algorithmic Mapping

- On outbound packets:
  - The source address prefix is overwritten with the external prefix
  - Checksum correction is performed as follows:
    - Calculate checksum of the old prefix (cP)
    - Calculate checksum of the new prefix(cP')
    - Take the ones complement difference (cP' + ~cP)
    - The difference is subtracted (using ones complement addition) to 16 non-prefix bits in the address
      - Bits 49-64 if the prefixes are /48 or shorter
      - *New* Bits 113-128 if the prefixes are /49 or longer

Two-Way Mapping Example *Improved*

Internal Prefix: FD01:0203:0405:/48

External Prefix: 2001:0DB8:0001:/48

} Configured on NAT66 Device

Outbound Example:

ORIGINAL SOURCE ADDRESS: FD01:0203:0405:0001::1234

cP = 0xFCF5

External prefix is copied into the address, cP' = 0xD245

~cP' = ~0xD245 = 0x2DBA

Diff = cP + ~cP' = 0xFCF5 + 0x2DBA = 0x2AB0

~Diff = ~0x2AB0 = 0xD54F

Bits 49 - 64 => 0x0001 + 0xD54F = 0xD550

MAPPED ADDRESS = 2001:0DB8:0001:D550::1234

Two-Way Mapping Example (Cont.)

Internal Prefix: FD01:0203:0405:/48

External Prefix: 2001:0DB8:0001:/48

} Configured on NAT66 Device

Inbound Example:

ORIGINAL DESTINATION ADDRESS: 2001:0DB8:0001:D550::1234

cP = 0xD245

External prefix is copied into the address, cP' = 0xFCF5

~cP' = ~0xD245 = 0x030A

Diff = cP + ~cP' = 0xD245 + 0x030A = 0xD54F

~Diff = ~0xD54F = 0x2AB0

Bits 49 - 64 => 0xD550 + 0x2AB0 = 0x0001

MAPPED ADDRESS = FD01:0203:0405:0001::1234

Topology Hiding Concepts

- There are two related concepts that need to be picked apart
  - Topology Hiding: Hiding the internal network structure
    - Hiding subnet information from external attackers
  - Preventing Correlation: Eliminating host<=>connection correlation
    - Isn't provided by NAT66, because both mappings are 1:1
    - Would host use of RFC 4941 privacy addresses be sufficient?

New Topology Hiding Mechanism

- Prefix is mapped, as in two-way mapping
- The subnet bits and part of the IID are encrypted using a reversible cipher
  - For /48 or shorter prefix lengths, the subnet bits and enough of the lowest order IID bits to make 64 bits are encrypted
    - Using a standard 64 bit cipher (perhaps DES?)
  - For prefix lengths from /49 to /64, the subnet bits (if any) and enough of the IID bits to make 48 bits are encrypted
    - Will require identification of a 48-bit reversible cipher
- Checksum correction is performed using the lowest order 16 bits of the IID

Topology Hiding Example (Outbound)

FD01:0203:0405:	0001:0000:0000:0000:	1234
#1	#2	#3
2001:0DB8:0001:	xxxx:xxxx:xxxx:xxxx:	nnnn

- #1: Map from internal to external prefix (/48)
- #2: Encrypt appropriate number of bits (64 in this case)
- #3: Perform checksum correction in lowest order bits

NAT66 vs. IPv4 NA(P)T

- One-to-One Two-Way Algorithmic Mappings
  - Allows inbound connections and direct peer-to-peer applications
  - External addresses can be configured in the DNS
- NAT66 doesn’t do port mapping or affect the TCP/UDP pseudo-header checksum
  - No need to traverse the IPv6 extension header chain
  - Compatible with security mechanisms that encrypt the transport header (IPsec ESP)
  - Allows for continued innovation at transport layer
- Both NAT66 and IPv4 NA(P)T change IP addresses en route
  - Causes problems if applications use IP addresses for referrals
  - Interferes with security mechanisms that rely on immutable IP addresses

Open Issues

- We've received quite a bit of feedback on NAT66
  - Thanks to everyone who has read and commented!
- In this presentation, we've focused on a few important issues that would benefit from discussion
  - Issues that aren't listed here are not being ignored. If we decide to go forward with this work, they will be addressed.

A NAT by any other name…

- There have been proposals to re-name the NAT66 specification (MAT, NAC, …)
  - Pros:
    - Highlights the difference between NAT66 and IPv4 NA(P)T
    - Doesn't directly contradict statements that IPv6 doesn't include NAT
  - Cons:
    - Somewhat obscure and misleading -- NAT66 *is* an IPv6 NAT proposal
    - Makes it harder for implementors who are working on an IPv6 NAT to find this work

Hairpinning, etc.

- The draft should be enhanced to cover Behave IPv4 NAT advice, to whatever extent that applies to NAT66
  - Hairpinning
  - What else?

Topology Hiding Requirements

- What are the actual requirements for topology hiding?
  - Does encrypting the subnet bits (and part of the IID) meet the needs?
    - IPv4 NATs make it look like all packets come from one host
  - What level of security is required?
    - How many samples could an attacker collect?
    - Is 48 bits enough? 64 bits?
  - Do we need to do something to obscure the original ports?
    - NAT66 doesn’t currently touch the transport-layer ports

Use of ULA Addresses

- Concerns have been raised about recommending the use of ULAs behind an NAT66 device
  - Changes the semantics of ULA addresses?
    - In other words, will users/applications be surprised if local addresses go global?
  - RFC 4193 indicates that ULAs have global scope but are locally routed
    - Not sure what that means in this context?
  - Applications are encouraged (but not required) to treat ULAs like global addresses, except they may be preferred over global addresses if both are present
  - What are the real-world assumptions about how these addresses will be used? Is using them behind a NAT66 box going to cause a conflict?
  - Should we also recommend the use of Link Locals behind NAT66?

Motivations/Applicability

- There have been a number of issues raised with the motivations and applicability described in the draft
  - Current text is not clear regarding what problem(s) NAT66 solves
  - Message is muddled regarding what we are and are not recommending
  - Discussion of moving applicability to a separate document

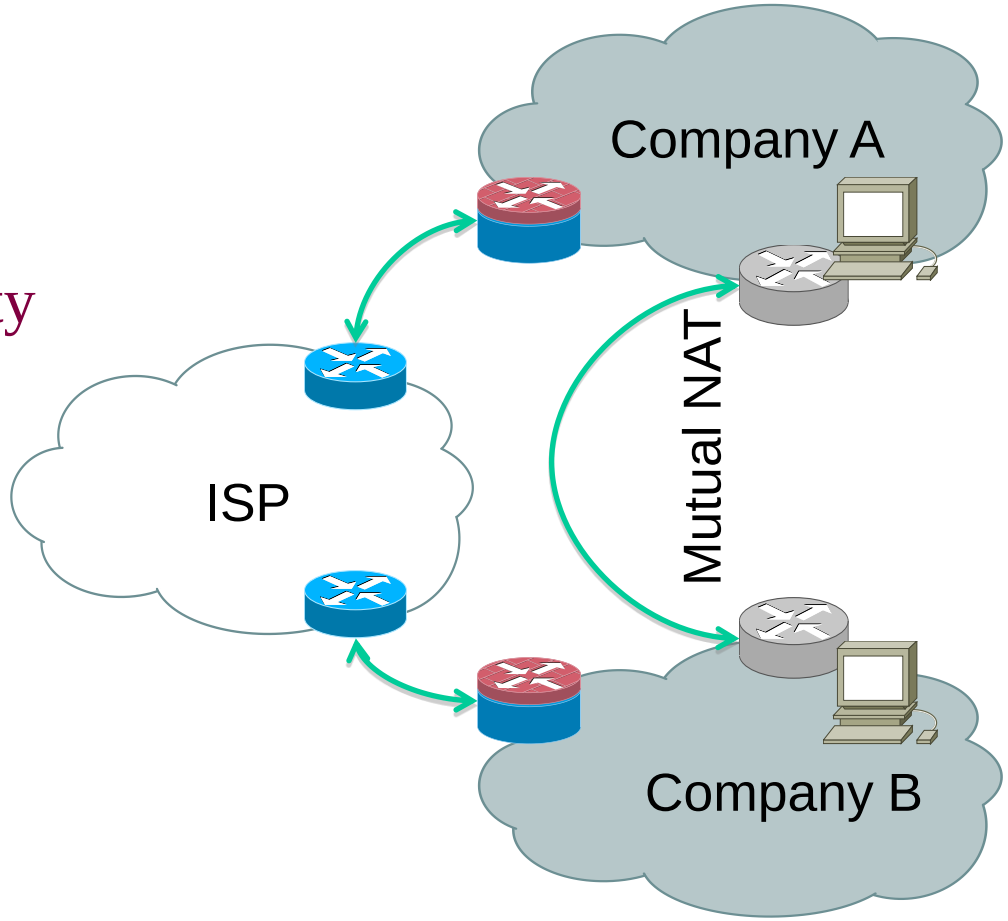
Reasons to use translation: IPv6/IPv6

Why do I care?

- I have customers telling me that NAT is important to them for “topology hiding”
  - Can someone tell me what about “topology” is important to hide?
- NAT66 attributes:
  - Obviates question of TCP/UDP checksum
  - Enables 1:1 address/host interface mapping
  - Therefore resolves several major issues with the end to end principle that IPv4/IPv4 NAT did not
- Therefore
  - I’m interested in helping the Internet scale better and trying to figure out what the remaining issues are

Business-to-business VPN

- Business-to-business connectivity
  - Company A uses services of company B under contract and has private security/connectivity relationship
- Issues:
  - Connectivity management
  - Mutual exposure – limiting information revealed
- Problem discussed in
  - http://tools.ietf.org/id/draft-baker-v6ops-b2b-private-routing

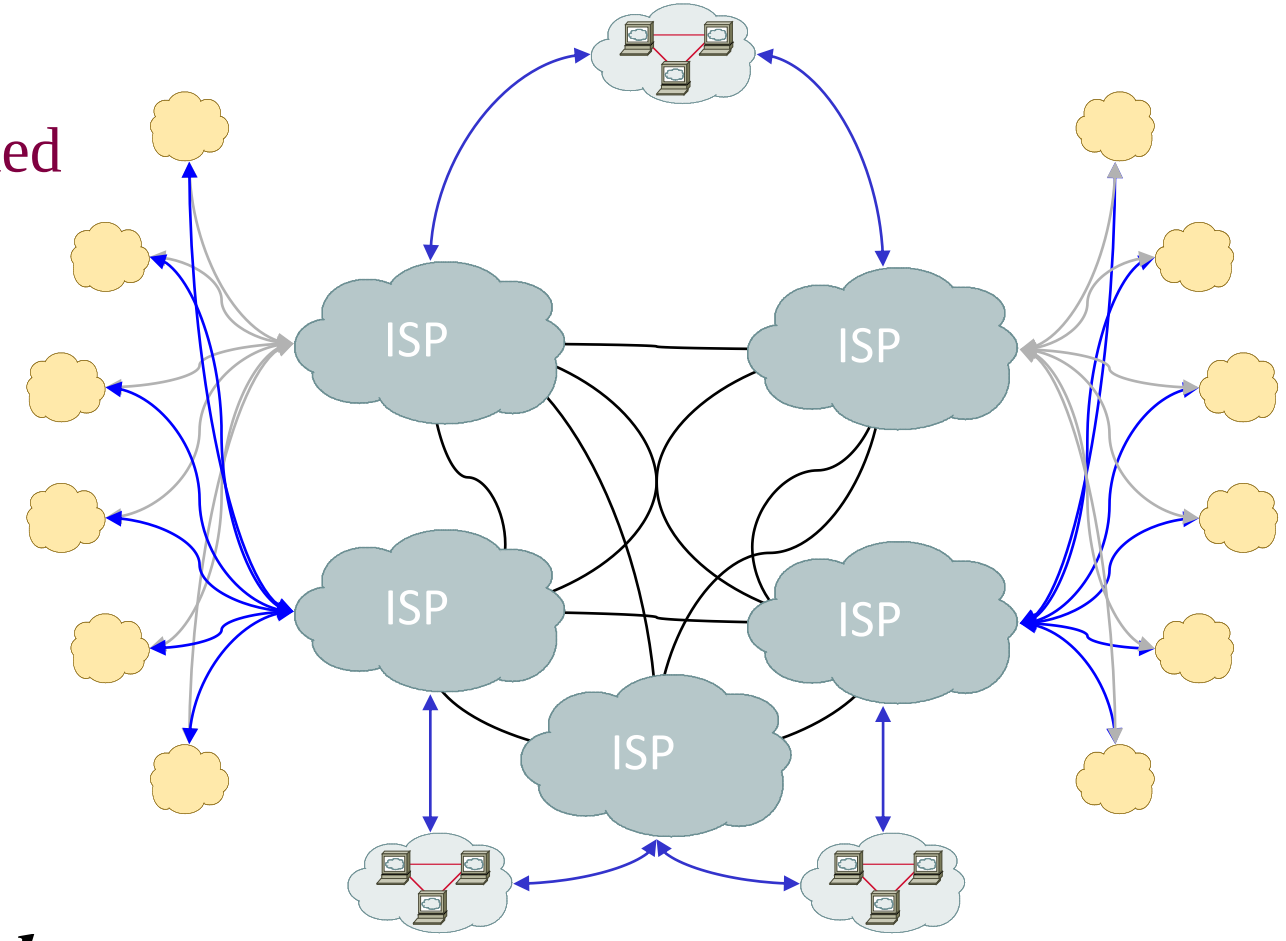


The multihoming problem…

SCALING INTERNET ROUTING

Present model - PI/PA multihoming

- Current statistics:
  - US: about one multihomed network per 18,000 population
  - World: about 1:50,000
- Expected 2050 density:
  - About 1:1000?
- Implication:



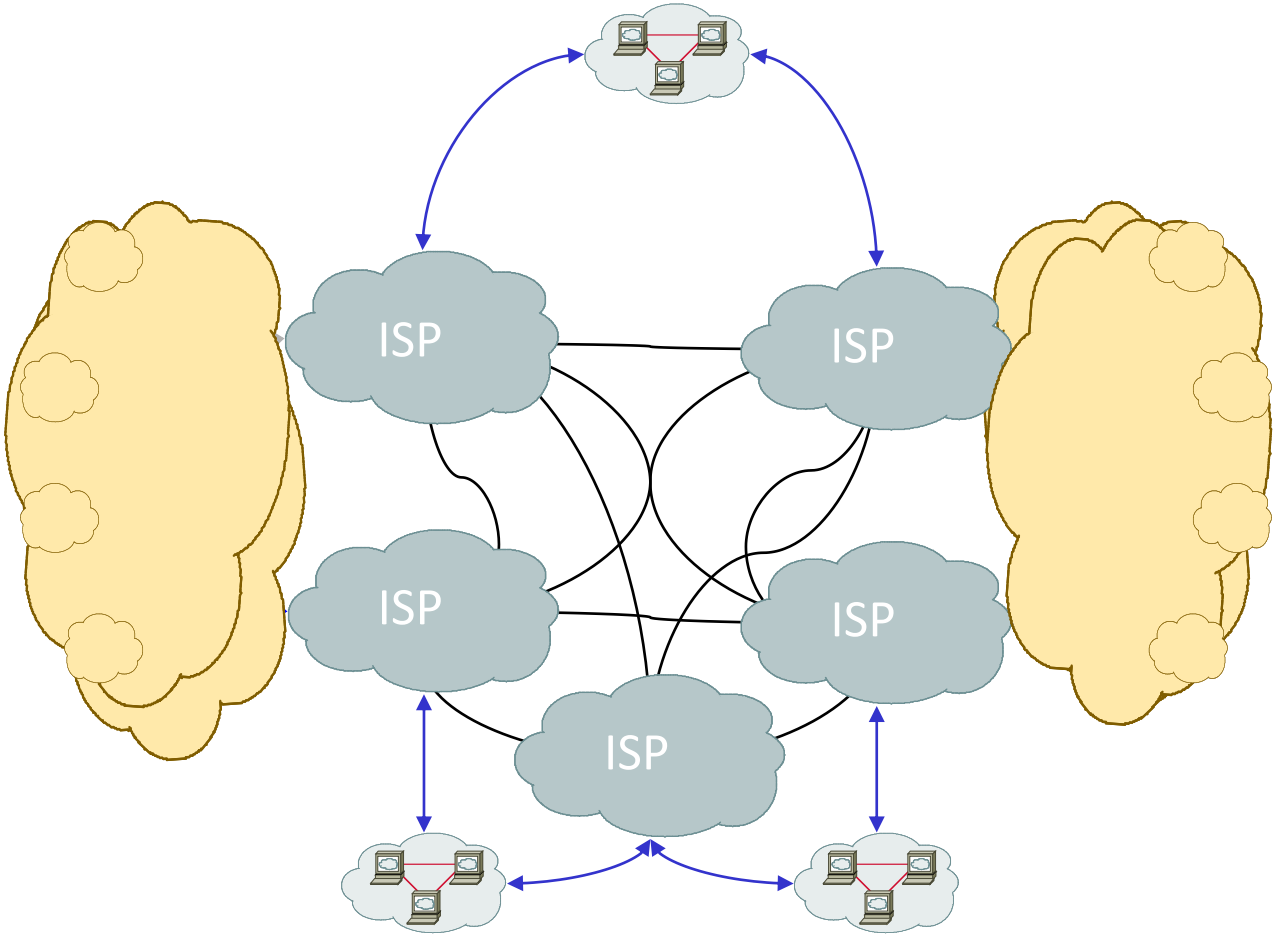
$$\frac{10,000,000,000\ people}{1000\ prefixes/capita} \approx 10,000,000\ prefixes$$

RFC 3582 analysis of PI/PA multihoming

\	PI	PA like PI
Redundancy	✓	✓
Address portability	✓	no
Load sharing	✓	✓
Performance	✓	✓
Policy	✓	✓
Simplicity	✓	✓
Transport session survivability	✓	✓
Impact on DNS	✓	✓
Datagram filtering	✓	Issues
Scaling: impact on routers	$O(10^{7})$ prefixes	$O(10^{7})$ prefixes
Scaling: impact on hosts	✓	✓
Scaling: host/router interaction	✓	✓
Scaling: network management	✓	✓
Scaling: ISP cooperation	✓	Issues

Shim6 viewpoint: PA multihoming

- Premise:
  - ISPs have prefixes
  - Edge networks inherit prefixes from ISPs
  - Only the ISP's prefix is advertised in BGP, not the inherited network prefix
- Prefixes in the internet core:
  - O(tens of thousands of prefixes)

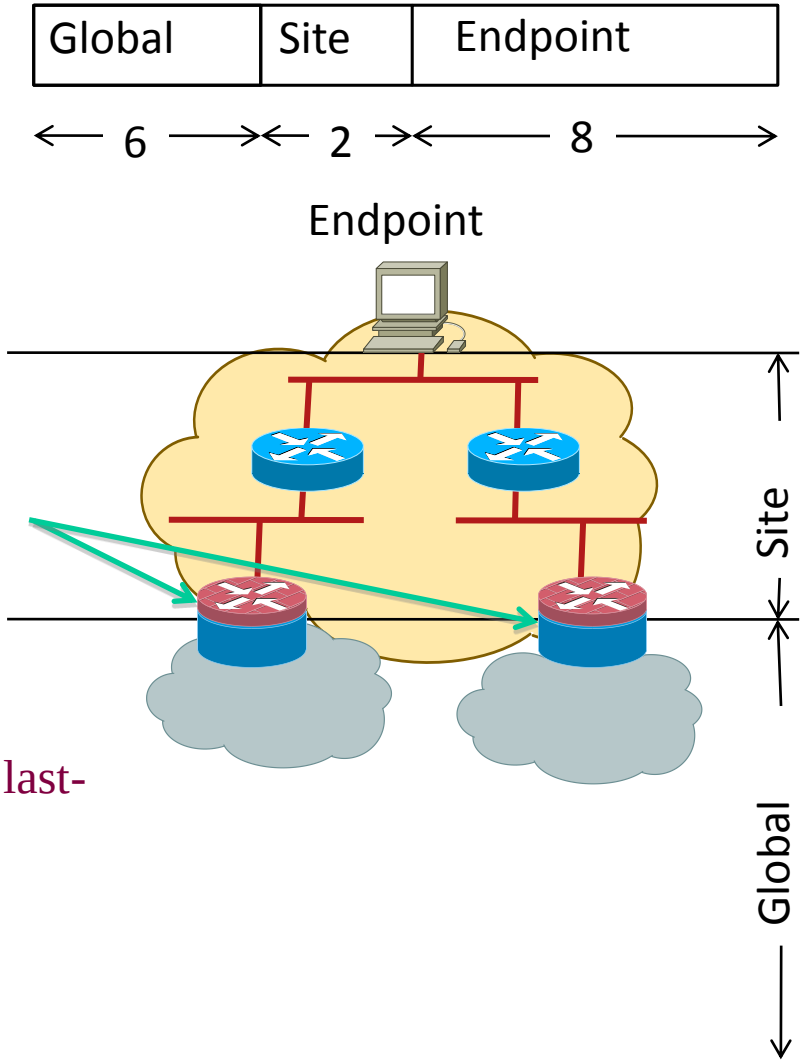


RFC 3582 analysis of shim6 multihoming

Redundancy	Multiple routes
Address portability	Addresses not portable
Load sharing	Host selects route by address pair
Performance	Performance only partially predictable
Policy	Address Pair policy is local
Simplicity	Not as simple as a single prefix
Transport session survivability	SCTP survives; TCP may with changes, UDP does not
Impact on DNS	✓
Datagram filtering	Ingress filtering affects routes
Scaling: impact on routers	$O(10^4)$ prefixes
Scaling: impact on hosts	Hosts must select address pair
Scaling: host/router interaction	✓
Scaling: network management	Choice of address pair not controlled in network routing but in host
Scaling: ISP cooperation	✓

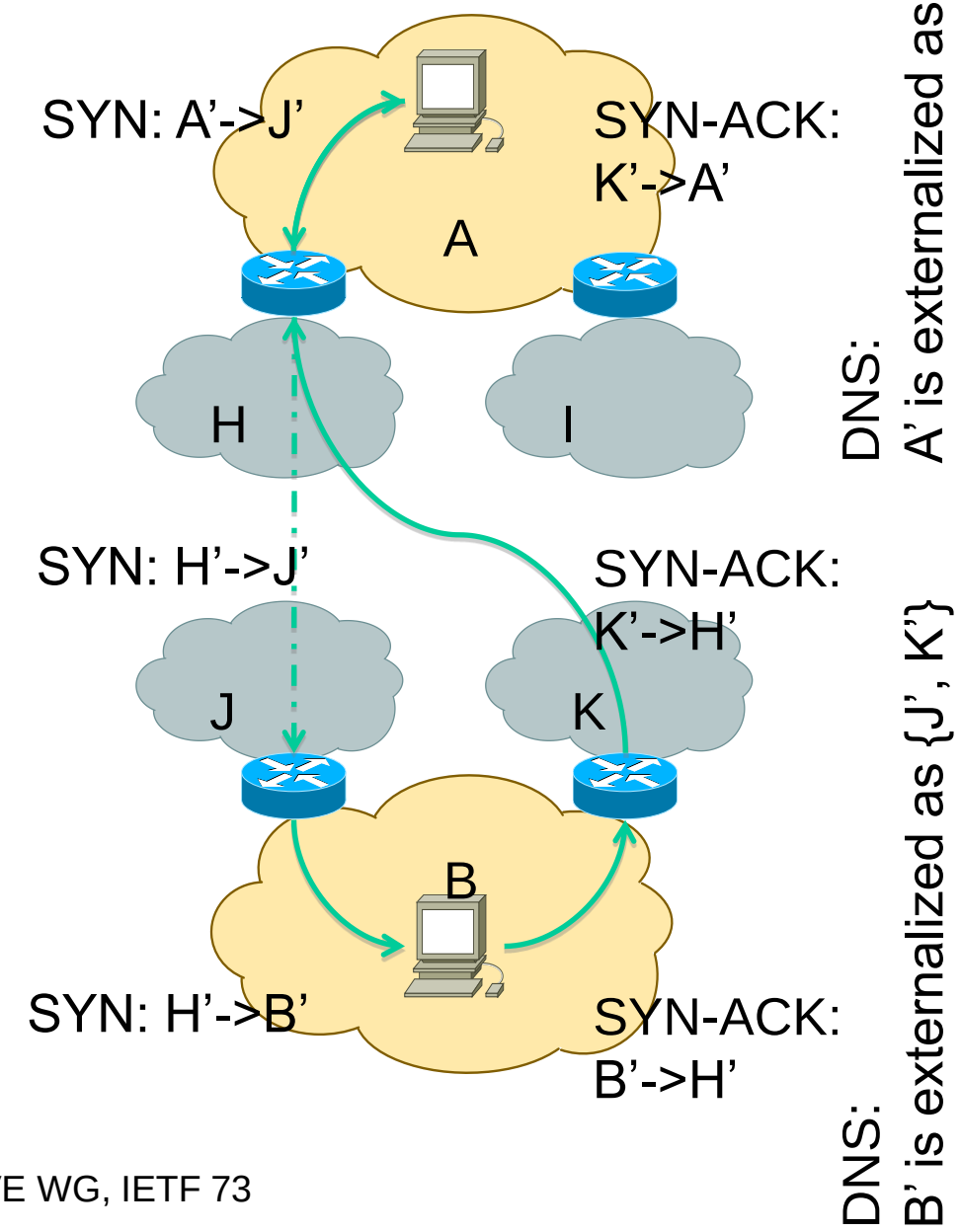
GSE Addressing Model: “8+8”

- Address components:
  - Global: /48
  - Site: 16 bit subnet (not /56 etc)
  - Endpoint: globally unique 64 bits
- Assumptions:
  - Global and perhaps Site parts are mutable
    - Only Global part used in core
    - Global and Site part can change at DMZ
    - Address in core is Provider-Assigned
    - Address in edge is Local in some form
  - Locator is relevant only to datagram routing/forwarding, including forwarding from last-hop router to host
  - Endpoint ID used to identify transport session
  - Host part of the address is Endpoint ID
- Prefixes in the internet core:
  - O(tens of thousands of prefixes)



Route Optimization in Multihoming

- Issue:
  - If address changes when crossing DMZ to a different provider, how does end system recognize the session on the SYN-ACK?
- Possible solutions
  - Recognize any or some of the addresses listed in DNS
  - Transport announces addresses
  - Loose Source Route inserted by DMZ
  - Host Identity Protocol (is that just IPsec ESP Null?)



RFC 3582 analysis of GSE

	GSE using NAT66
Redundancy	✓
Address portability	✓
Load sharing	✓
Performance	✓
Policy	✓
Simplicity	✓
Transport session survivability	✓
Impact on DNS	✓
Datagram filtering	✓
Scaling: impact on routers	$O(10^{4})$ prefixes
Scaling: impact on hosts	Endpoint Identification
Scaling: host/router interaction	✓
Scaling: network management	✓
Scaling: ISP cooperation	✓

Remaining real issue

- NAT66 is an address management solution, not a security solution
- Delusional, naïve, gullible network administrators confuse IPv4/IPv4 NAT with Stateful Firewalls and therefore with a “security solution”
- Delusional marketing people confuse IPv4/IPv4 NAT with Stateful Firewalls and therefore with a “security solution”
- Therefore, **people will sell and deploy IPv6/IPv6 NAT as a security solution**
- That will be bad unless products also implement a security solution

Next Steps

- Do we think that the IETF should define NAT66?
- If so, is the Behave WG the best place to do it?
- Is this document a good starting point for this work?
- How do we move forward from here?