

Analysis of Solution Candidates to Reveal the Origin IP Address in Shared Address Deployments

draft-boucadair-intarea-nat-reveal-analysis-01

(INTAREA WG, BEHAVE WG)

IETF 80-Prague, March 2011

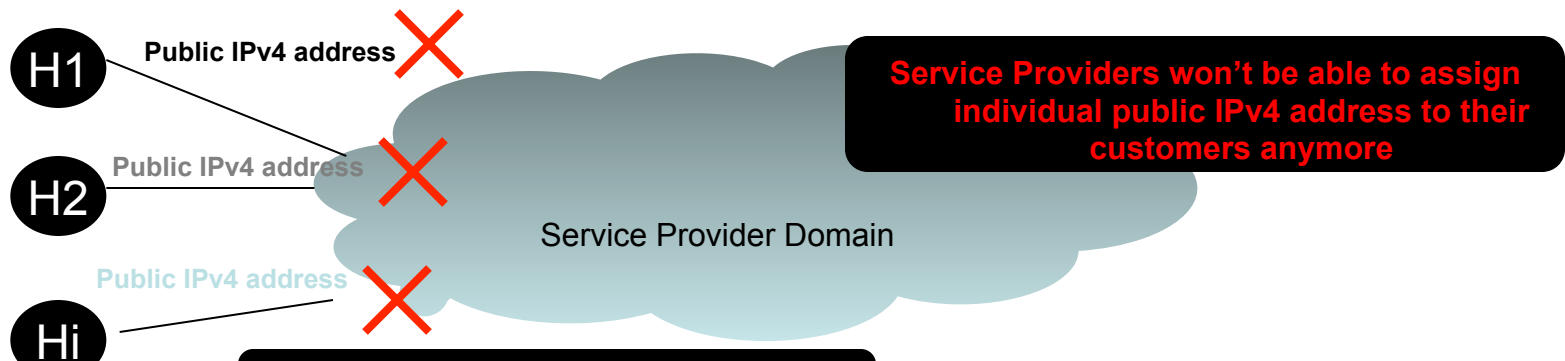
M. Boucadair, J. Touch, P. Levis and R. Penno

Agenda

- Reminder about some address sharing issues
- Why Host_ID is needed?
- How to insert a HOST_ID?
- Solution analysis
- Next steps

IPv4 Service Continuity

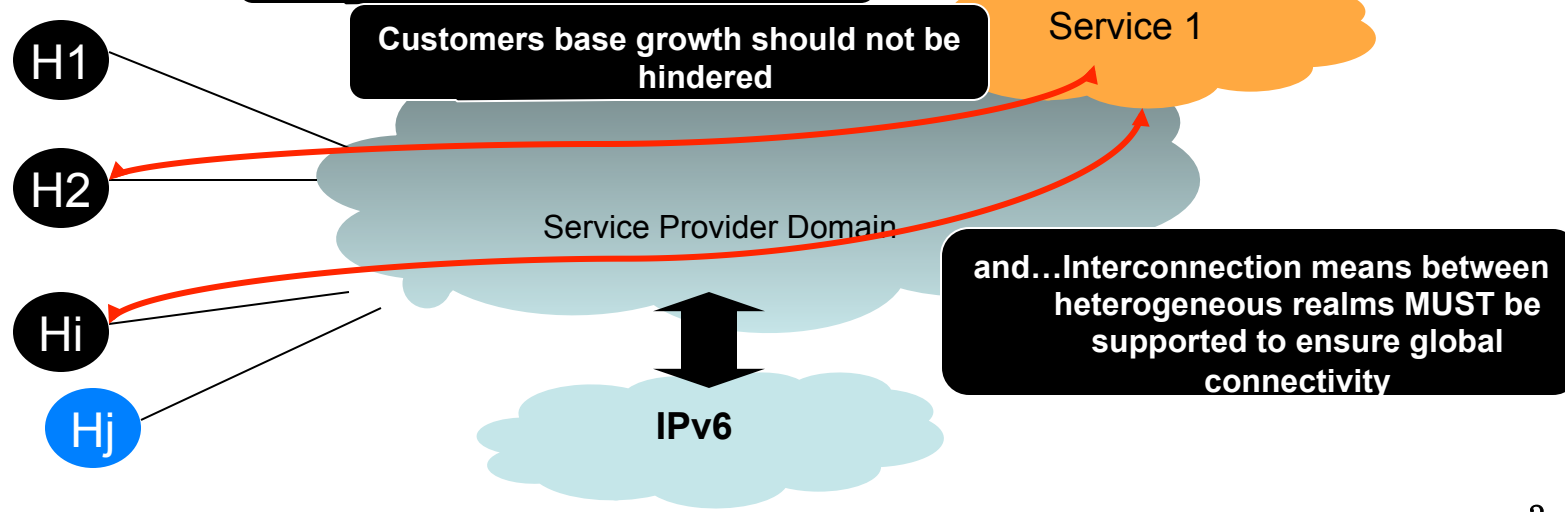
Public IPv4 address will be exhausted soon
Need to rationalize the use of IPv4 addresses



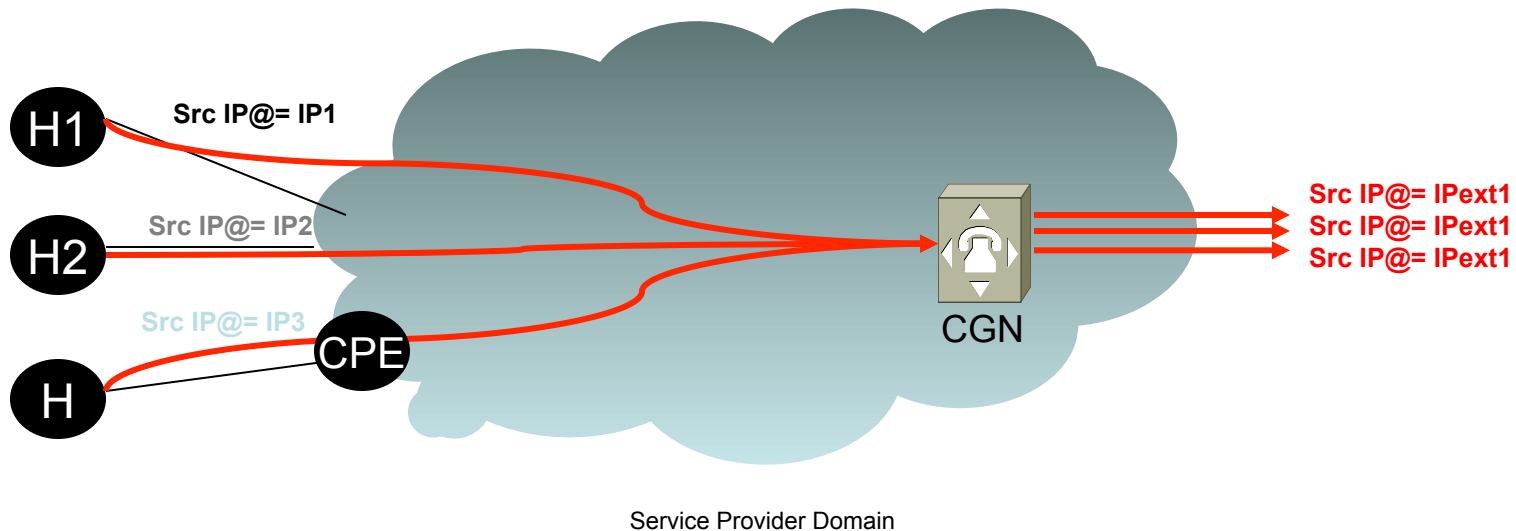
But...

Means to ensure service continuity is a MUST...

Customers base growth should not be hindered

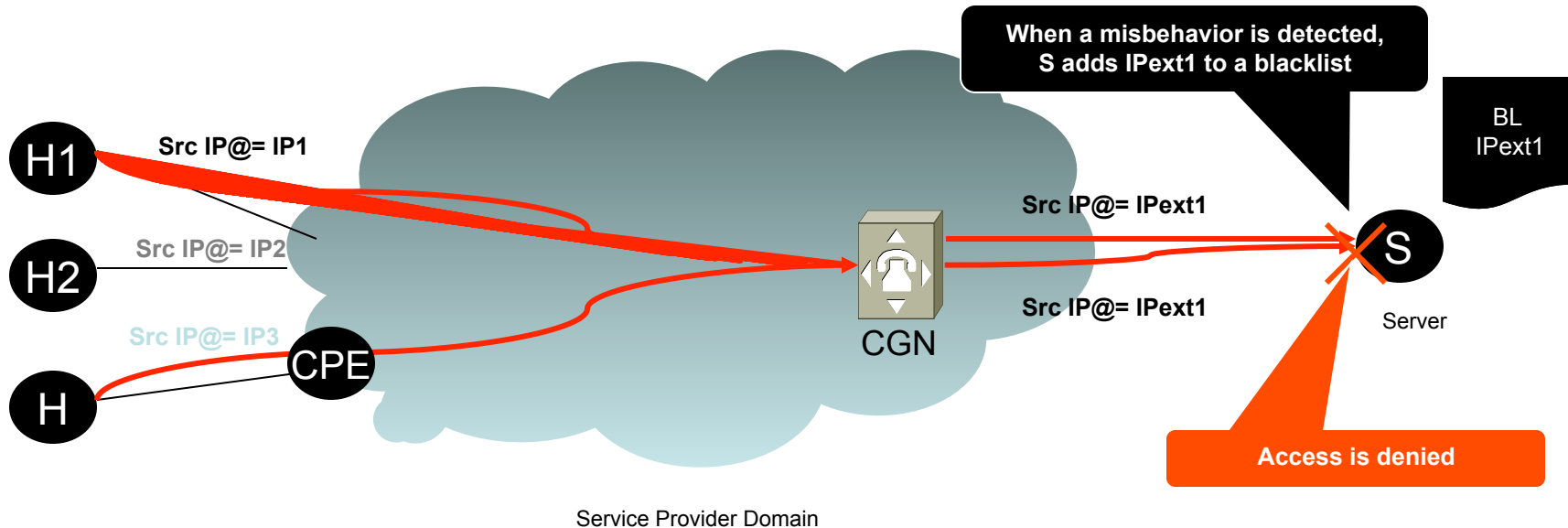


NAT-based Address Sharing



**The internal and the external IP addresses may be of distinct address families (e.g., IPv4, IPv6):
NAT44 or NAT64**

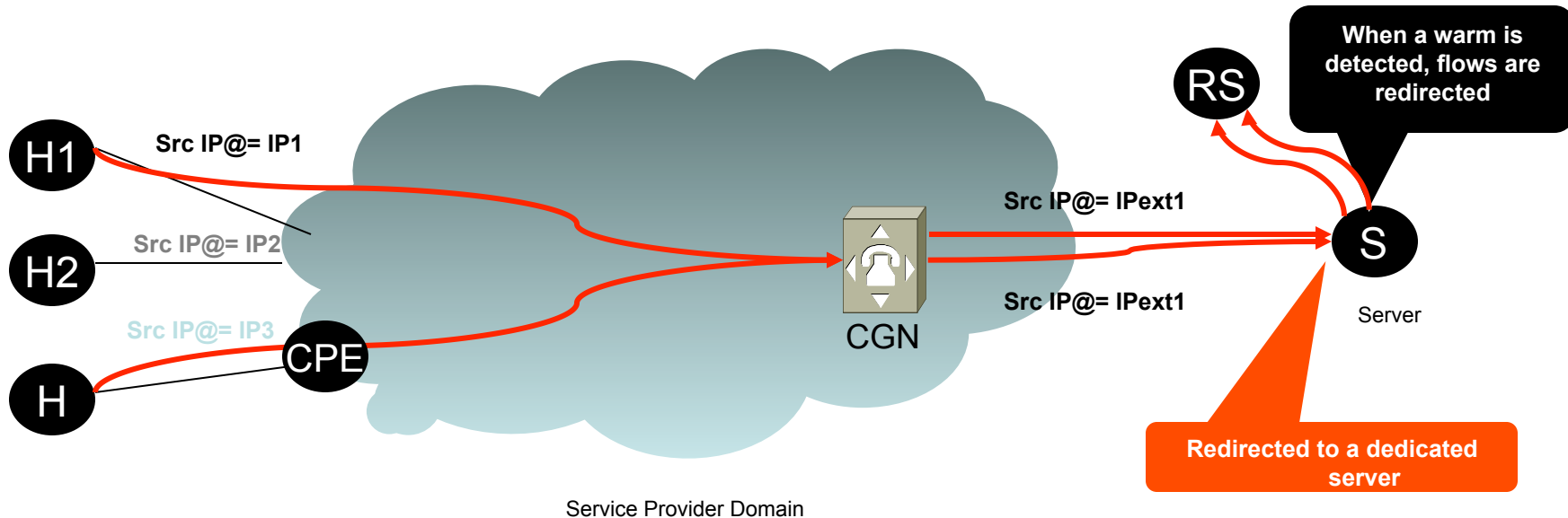
NAT-based Address Sharing



Blacklisting a misbehaving user:
The server relies on the source IP address

All subscribers using the same address will be impacted:
Loss of users for the content providers, calls to the hotline for the IP Network Provider (\$\$/mn, OPEX loss for the ISP) and unsatisfied customers

NAT-based Address Sharing



Infected machine

traffic redirection is based on the source IP address

All subscribers using the same address will be impacted:

Difficult to troubleshoot, calls to the hotline for the IP Network Provider (\$\$/mn, OPEX loss for the ISP) and unsatisfied customers

A more exhaustive list of issues are identified in
I-D.ietf-intarea-shared-addressing-issues

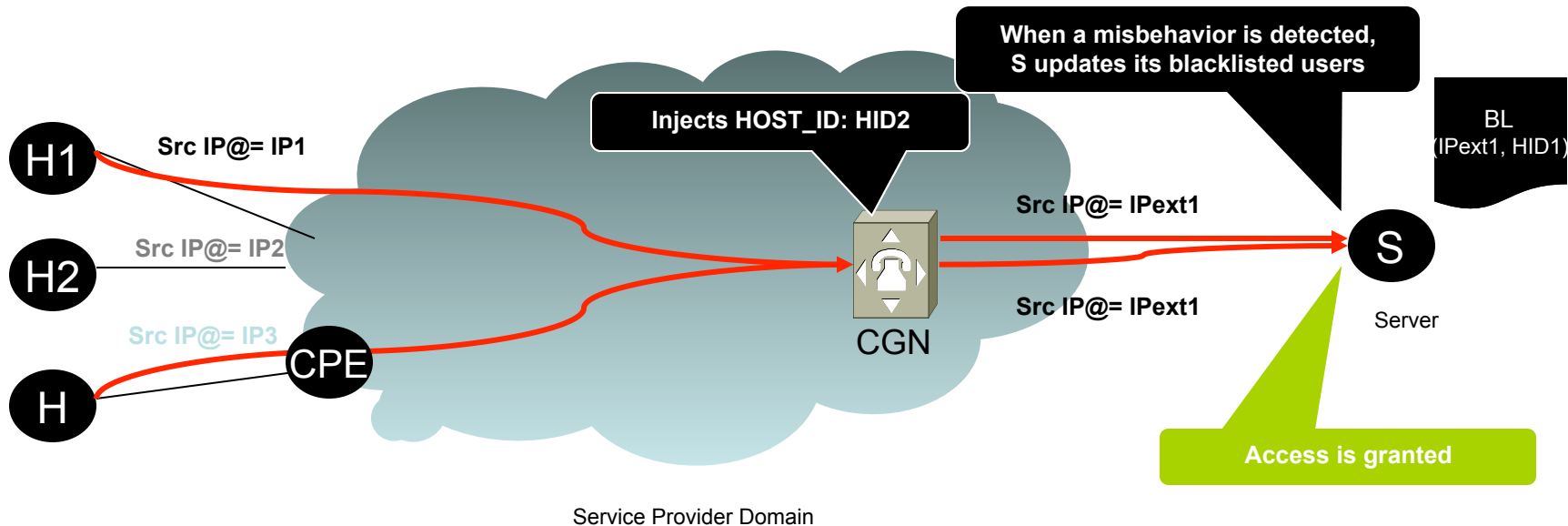
Generalizing the issue

- Observation
 - Today, servers use the source IPv4 address as an identifier to treat some incoming connections
 - **Tomorrow**
server does not know which host is the sending host
- Objective
 - sending host (not only based on the source IP @)
- Requirement
 - The server must have extra information than the source IP address to differentiate the sending host: We call **HOST_ID** this information

HOST_ID: Rationale

- **What is the HOST_ID?**
 - It must be unique to each user under the same address
 - Adding a HOST_ID does not “break” the privacy of the user, it reveals the same information as the source IP address when there is not CGN in the path
 - E.g., first bits of an IPv6 address, private IPv4 address, etc.
- **Who puts the HOST_ID?**
 - The address sharing function injects the HOST_ID when it translates IP packets
 - The CPE can put the identification in the packet and the CGN checks it instead of doing the actual writing. The performance impact would be distributed/shared between CPE and CGN
- **Where is the HOST_ID?**
 - If the HOST_ID is put at the IP level, all packets will have to bear the identifier
 - If it is put at a higher connection-oriented level, the identifier is only needed once in the session establishment phase
 - for instance TCP three-way-handshake

NAT-based Address Sharing (revisited)



Blacklisting a misbehaving user:

The server relies on the source IP address & **HOST_ID**

The server needs to be updated to:

(1) be able to extract the HOST_ID, (2) Enforce policies based on the HOST_ID, (3) log the HOST_ID

Solutions to reveal the HOST_ID

	UDP	TCP	HTTP	Encrypted traffic	Success Ratio	Possible performance impact	Modify OS TCP/IP stack is needed (*)	Deployable	Notes
IP Option	Yes	Yes	Yes	Yes	30%	High	Yes	Yes	
TCP Option	No	Yes	Yes	Yes	99%	Med to High	Yes	Yes	
IP-ID	Yes	Yes	Yes	Yes	100%	Low to Med	Yes	Yes	1
HTTP Header (XFF)	No	No	Yes	No	100%	Med to High	No	Yes	2
Proxy Protocol	No	Yes	Yes	Yes	Low	High	No	No	
Port Set	Yes	Yes	Yes	Yes	100%	NA	No	Yes	1,3
HIP					Low	NA	--	No	4,5

- (1) Requires mechanism to advertise NAT is participating in this scheme (e.g., DNS PTR record) (*) Server side
- (2) This solution is widely deployed
- (3) When the port set is not advertised, the solution is less efficient.
- (4) Requires the client and the server to be HIP-compliant and HIP infrastructure to be deployed
- (5) If the client and the server are HIP-enabled, the address sharing function does not need to insert a user-hint. If the client is not HIP-enabled, designing the device that performs address sharing to act as a UDP/TCP-HIP relay is not viable.

IP option, IP ID and Proxy Protocol are **broken**

HIP is not “widely” **deployed**

Port Set requires **coordination**

XFF is **largely deployed** in operational networks but still the address sharing function **needs to parse all applications messages**

TCP Option is superior to XFF since it is not specific to HTTP but what about **UDP**? Update the Servers OS **TCP/IP is required**

What to do with this analysis?

- Recommend a solution?
 - Of course, individual solutions needs to discuss potential impact on performance, mis-usage of the solution to reveal other “sensitive” information, etc.
- Add a conclusion to say: “IETF has documented the issues and has analyzed solution candidates but IETF believes CGN should stay “evil””?
 - Risk of emergence of proprietary solutions
- Add a statement to say: “IPv6 will solve this?”
 - Yes, this is a strong signal but this does not mitigate the service brokenness to be encountered by subscribers when address sharing will be deployed at large
 - The issues are also valid for NAT64

Next steps?

- Please advise