source_IP NAT

March 24, 2009
charliep@wichorus.com

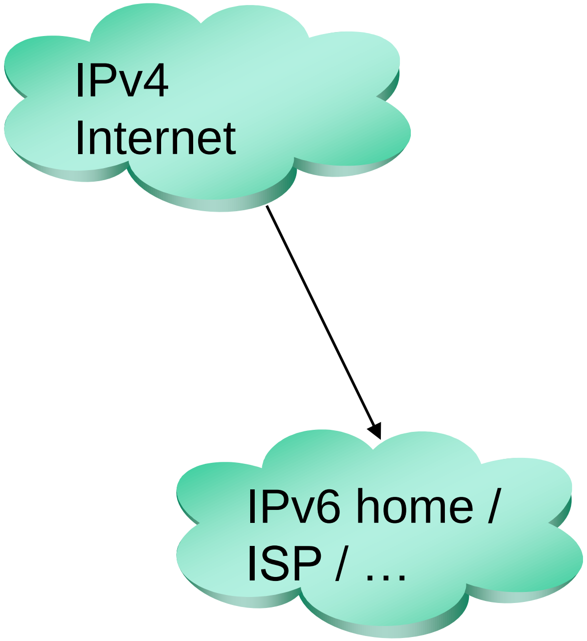
NAT today

- Network Address Translation – typically between globally unique IP addr. and “private” IP addr.
    - Net 10.0.0.0 provides a million private addresses per site
    - Net 192.168 provides 65,536 such private addresses
    - Provides topology hiding; typ. bundled with firewall
- Often, translation relies on port translation
- Requires per-function ALGs (e.g., TCP, FTP, …)
- Works only when inside host initiates application
- Many variations also for IPv6 → IPv4 connections

Can new businesses use IPv6 after the runout?

- Not unless they can serve their customers!
- They <u>must</u> have presence on the web all the time, not depending on getting a port allocated by a flow initiated by the company website
- Without this, businesses will fight very hard against using IPv6 – otherwise they lose 99% of their potential customer base
- As it is now, even with continuous growth, IPv6 will take a long time (?decades?) to catch up with IPv4
- A better model for transition: run v6, serve everyone, but serve IPv6 “slightly” better.

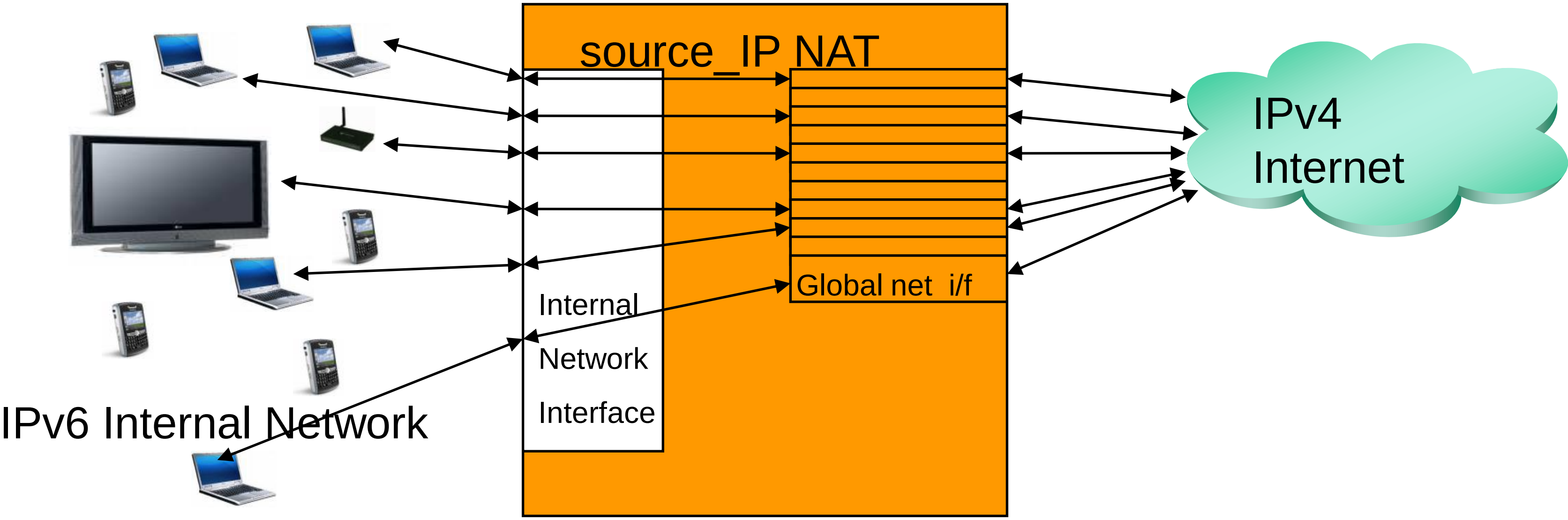
Proposal allows IPv4 → IPv6 communication



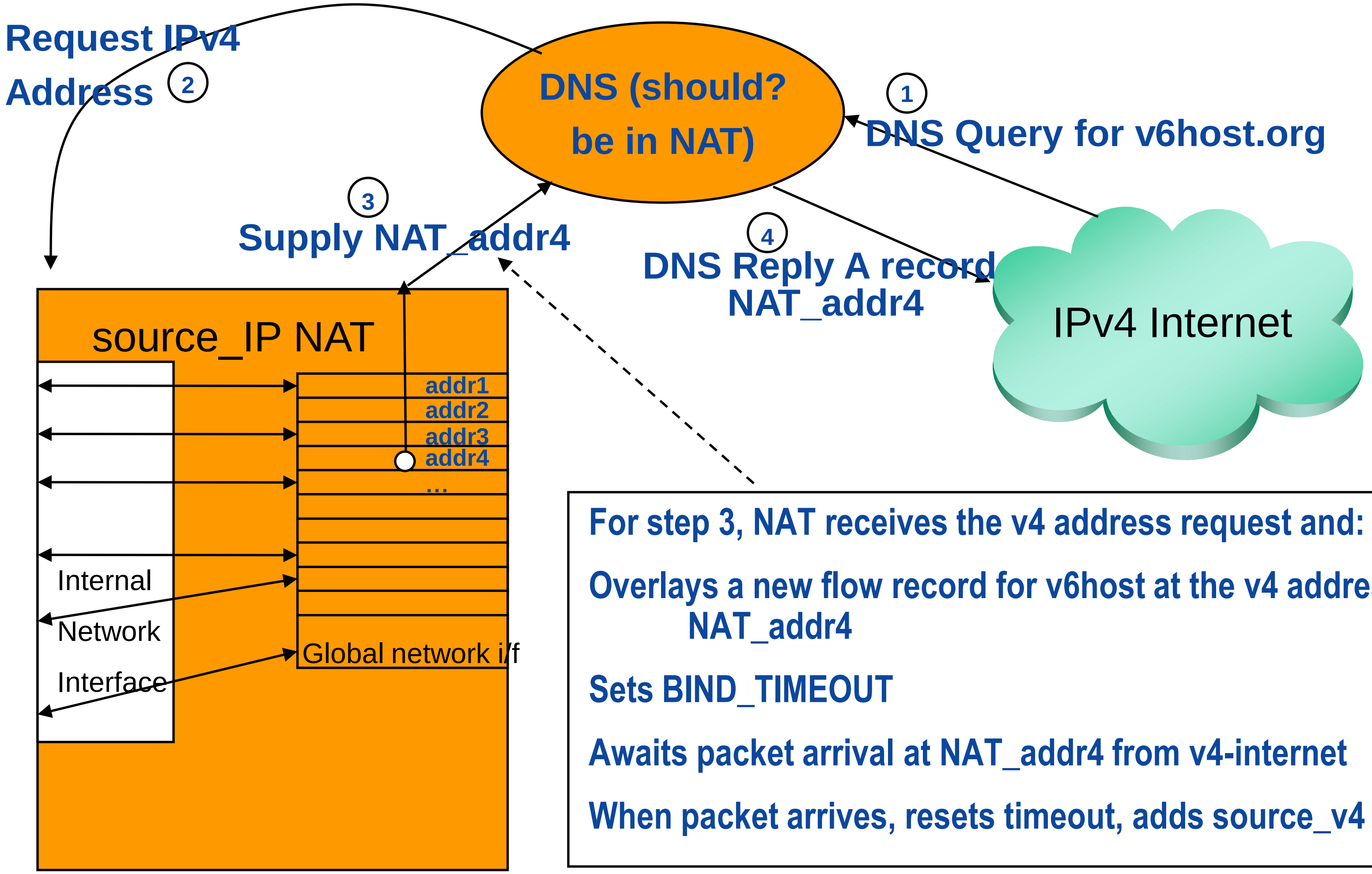
- DNS-based setup phase dynamically assigns a flow and an IPv4 address for communication with the IPv6 device
- When packet arrives at the newly allocated IPv4 address, the source IP address is then associated with the flow
- For established flows, source IP address "selects" the IPv6 destination
  - May use s-port # for finer control
- Designed for IETF [behave] wg, to be an easy step from where we are today
  - **It's not perfect!**

Bidirectional NAT v4 ←→ v6 (uses DNS)

- No changes to IPv6-only hosts or IPv4-only hosts
- No dual-stack
- No tunneling
- Easiest to delegate special domain to NAT box
- Modeled as a flow-management problem



Operation of system…



For step 3, NAT receives the v4 address request and:

Overlays a new flow record for v6host at the v4 address NAT_addr4

Sets BIND_TIMEOUT

Awaits packet arrival at NAT_addr4 from v4-internet

When packet arrives, resets timeout, adds source_v4

Two failure modes

- **The system will fail if there are too many new flow requests at about the same time**
  - Since the DNS Request does not have the source IP address, a newly allocated flow at a NATv4 address blocks that address “momentarily”
- **The system will fail if a specific source tries to access too many destinations**
  - At each IPv4 address of the NAT, a source IP address (and, possibly, source port) _identifies_ the flow
  - Can have one flow per source per NATv4 address, if lucky

Testing

- First try: www.wichorus.com [not “varied” enough”]
- Second try: HP’s 85 million access records for World Cup 1998
- By preprocessing input, can adjust many parameters
  - DNS response time (but not fine-grained enough control yet)
  - Arrival rate for DNS request == flow allocation request
  - WAIT_TIME
  - BIND_TIMEOUT
  - Number of destinations; number of sources
- Crucial need for more real-world data
- Have run thousands of scenarios; results available
- Website: http://www.psg.com/~charliep/sourceIP_NAT

Is it really like flow management?

- Incoming <v4dev, sport, NATaddr, dport, TOS> → <v4mapped, sport, v6dev, dport, TOS>
- Could use DPI as required
- Gradually move more functions to hardware?
  - Checksums
  - Pattern recognition
- Have to search overlapping flow records per v4addr
  - Determine maximum degree of overlap?
  - This is what provides scalability for the solution

Results: 0.05 flows/sec, #dests=1040, wait_time 60s

# of NATv4 addresses	Percentage of flows not served
1	10.45%
2	3.86%
4	1.9%
8	0.93%
16	0.45%
32	0.15%
64	0.02%
128	0.01%