

# SIPNAT (source\_IP NAT)

November 3, 2009  
charliep@wichorus.com



# Business pitfalls of moving to IPv6 today

- **Practically all of the customers are using IPv4**
- **So, business must serve IPv4 web accesses**
- **Web presence is required 24 x 7 x 52 x ...**
- **This is not compatible with today's NAT solutions, or today's IPv6 solutions**
- **Needed: “always on” NAT technology for v4→v6 translation**

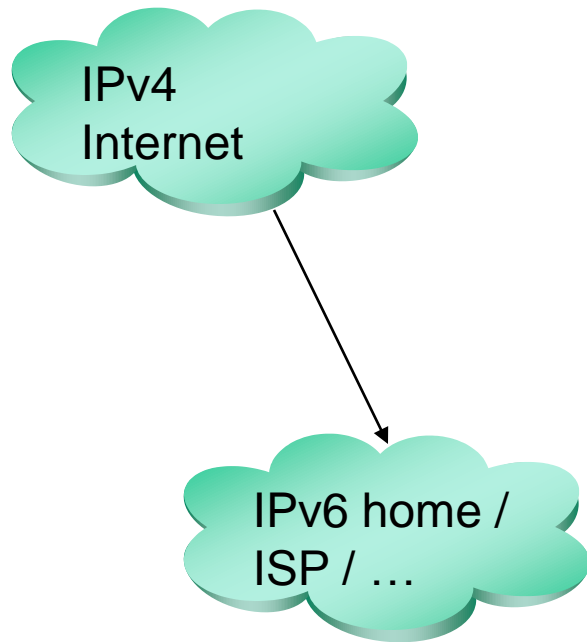
# NAT today: works, but running out of steam

- **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**
- **Requires per-function ALGs (e.g., TCP, FTP, ...)**
- **Always requires that inside host initiates app.**
- **Merging networks causes a renumbering nightmare**

# Other existing NAT solutions

- **Today, almost universally IPv4→IPv4 NAPT**
  - **Outgoing only**
  - **Port numbers required (so, e.g., GRE does not work).**
  - **Incompatibilities are confusing for application development**
  - **Poor results after non-participation by IETF or other SDOs**
- **IPv6 requires translation to work with Internet today**
  - **Many variations for IPv6 → IPv4 connections**
- **Many approaches require dual-stack (e.g., DS-Lite)**
- **Only IVI enables incoming session initiation**
  - **Not scalable; not deployed**

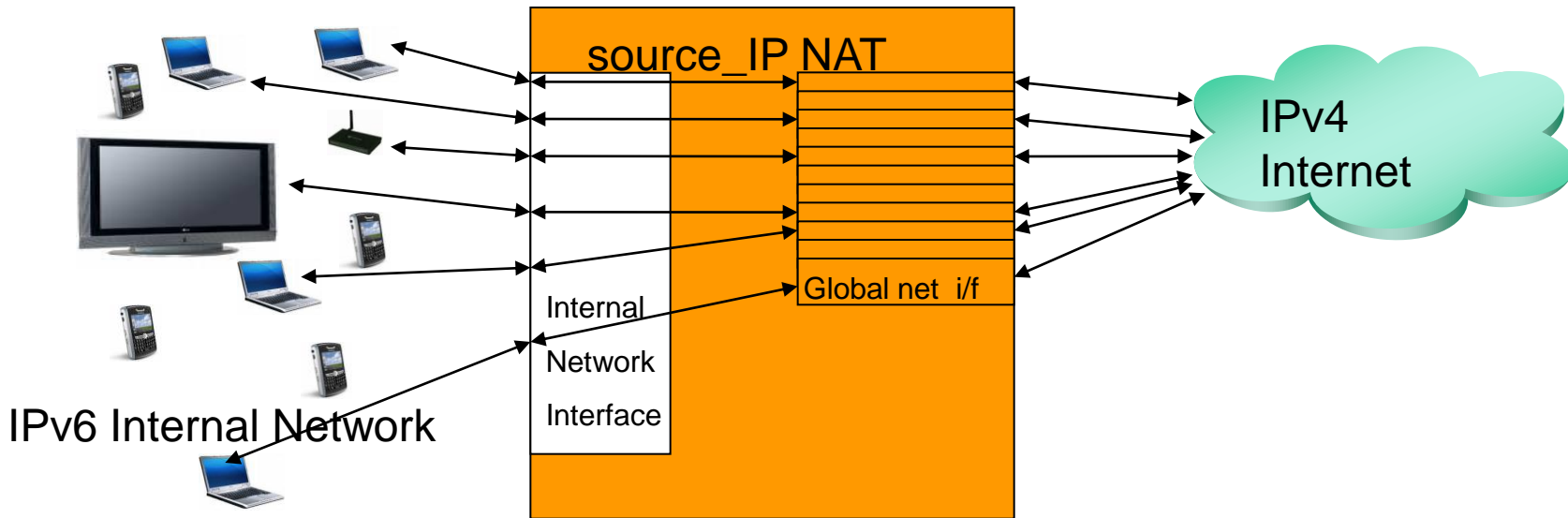
# Proposal allows IPv4 → IPv6 communication



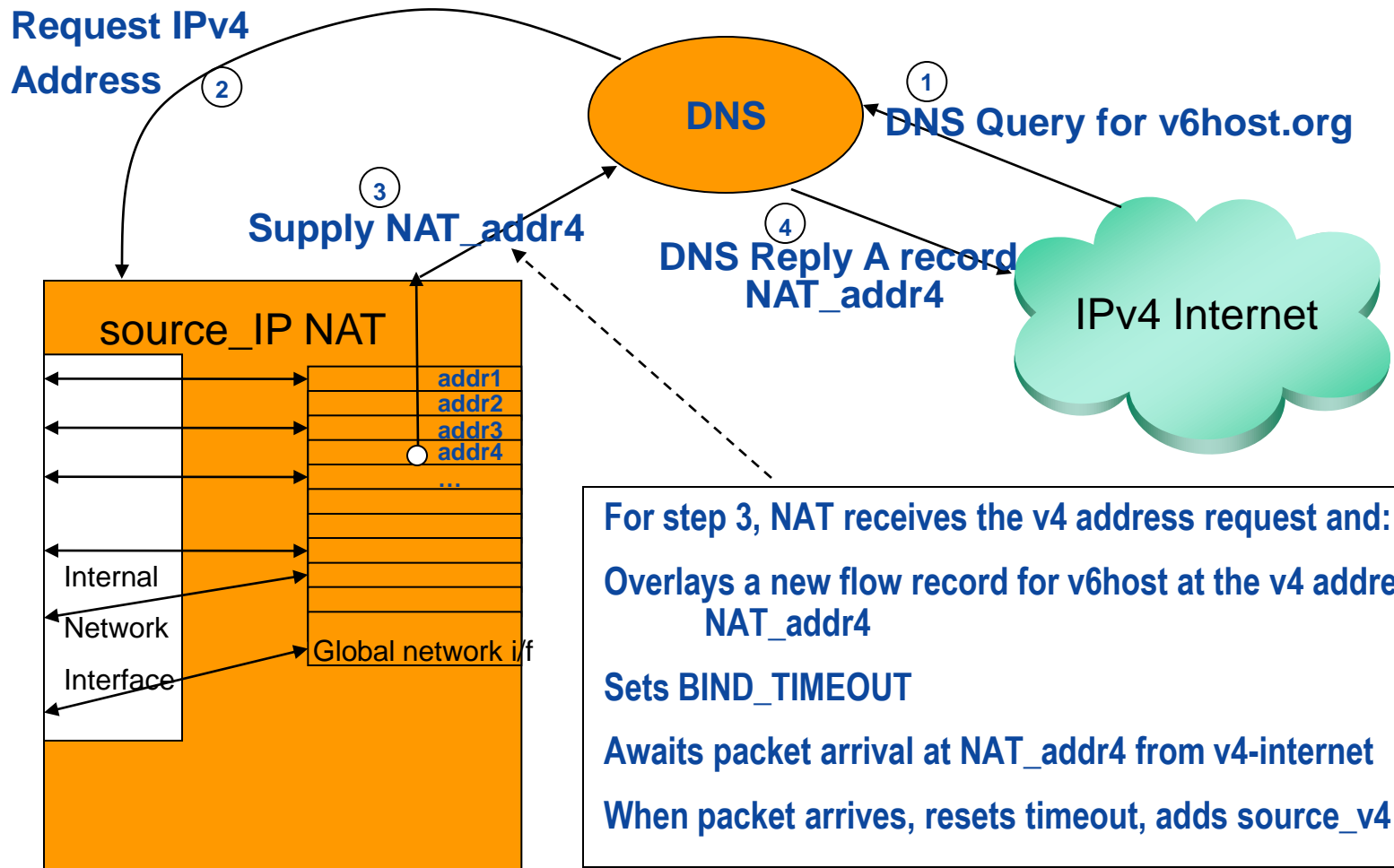
- Designed for IETF [behave] wg.
- For incoming packets, usual NAT needs the dest port # to find the destination → communication already started
- New proposal uses source IP address to “select” the IPv6 destination
  - May use s-port # for finer control
- DNS-based setup phase provides an IPv4 address for communication with the IPv6 device
- Allocation completed using source IP

# Bidirectional NAT v4 $\leftrightarrow$ v6 (uses DNS)

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



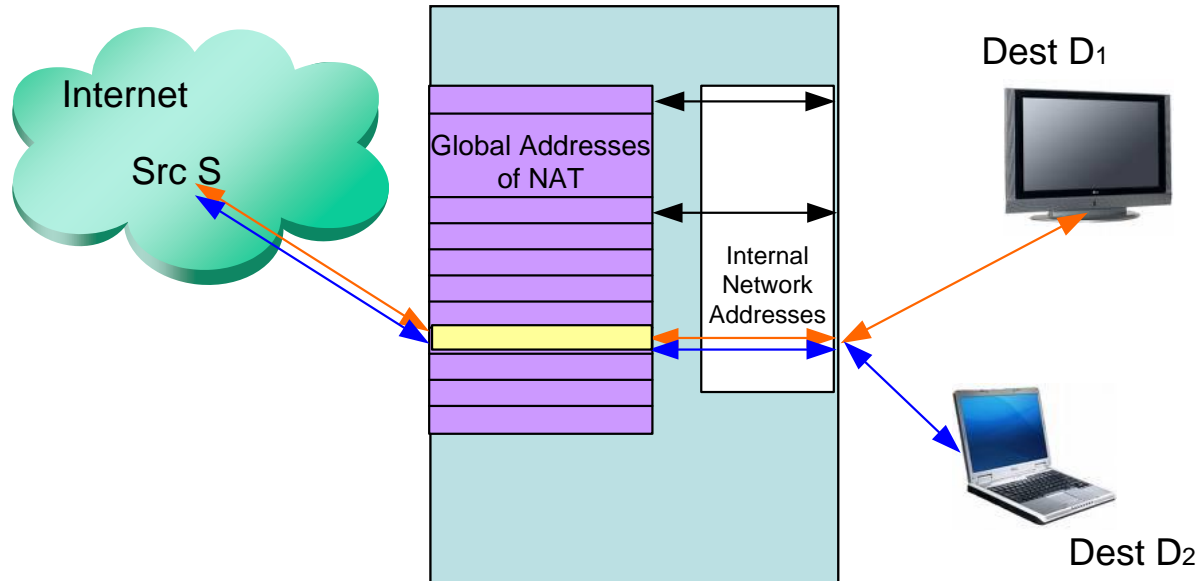
# Operation of system...



# Unassisted mode: two failure scenarios

- **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, the allocated flow will go to the source of the first packet to arrive that is not already deliverable**
- **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**

# Unassisted mode: one source $\leftrightarrow$ one dest.



- **One source cannot use the same NAT address for two different IPv6 destinations**

# Testing

- **Started with HP's 85 million access records for World Cup 1998**
- **By preprocessing input, can adjust many parameters**
  - **DNS response time**
  - **Arrival rate for DNS request == flow allocation request**
  - **WAIT\_TIME**
  - **BIND\_TIMEOUT**
  - **Number of destinations... sources**
- **Crucial need for more real-world data**
- **Have run thousands of scenarios; results available**
- **Old website [www.psg.com/~charliep/sourceIP\\_NAT](http://www.psg.com/~charliep/sourceIP_NAT)**

# Is it really like flow management?

- Incoming <v4dev, sport, NATaddr, dport, TOS> → <v4mapped, sport, v6dev, dport, TOS>
- Use DPI to figure out which ALG to use
- 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

# Payload assist for higher scalability / robustness

- **Base v4→v6 NAT system works well**
- **Can improve scalability and robustness using known payload fields (for certain protocols)**
- **Good example: http GET contains “http.host” field, identifying the destination**
- **Also: works for SIP (e.g., VoIP, presence, instant messaging, ...)**
- **Additional techniques to enable peer-to-peer**

# Pattern Matching techniques

- **A large majority of website pathnames are unique to specific destinations**
- **A pattern matching machine could identify the correct destination based only on HTTP pathname**
- **This could even be guaranteed for cooperative clients**
  - **For example: <http://www.wichorus.com/wichoruspages/...>**