# NAT and Firewall Traversal with STUN / TURN / ICE

Simon Perreault

Viagénie

{mailto|sip}:Simon.Perreault@viagenie.ca

http://www.viagenie.ca

# Credentials

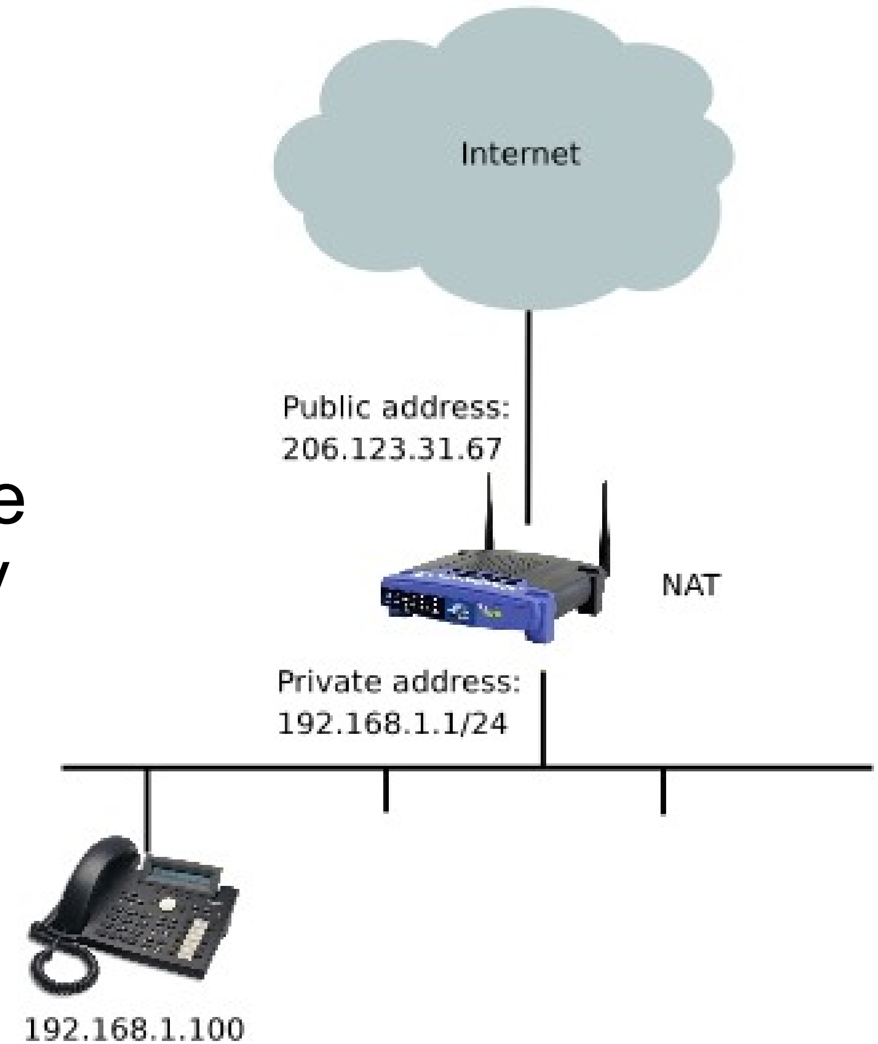
- Consultant in IP networking and VoIP at Viagénie.
- Developed Numb, a STUN / TURN server.
- Ported FreeSWITCH to IPv6.
- Co-ported Asterisk to IPv6.
- Developed many custom VoIP applications.

# Plan

- The problem of NAT and firewalls in VoIP
- How STUN, TURN, and ICE solve it
- Wireshark traces

# The Problem of NAT and Firewalls in VoIP

- Network address translators (NATs) are common devices that “hide” private networks behind public IP addresses.
- Connections can be initiated from the private network to the Internet, but not the other way around.
- The situation is made worse by the fact that SIP controls separate media streams and thus transports addresses.



# The Problem of NAT and Firewalls in VoIP

- Many firewalls only allow connections to be initiated from the private network, thus having the same effect as NATs.
- Moreover, firewalls commonly deny access to port numbers associated with VoIP.
- Some even inspect the packet contents to identify and reject VoIP traffic.
- **Result**: VoIP users behind NATs and firewalls do not benefit from the end-to-end connectivity necessary for VoIP.

# Server-Reflexive Address

- A NAT device works by associating a public address and port with a private destination address and port.

| Public | | Private |
|---|---|---|
| 206.123.31.67:55123 | ↔ | 192.168.1.2:5060 |

- The public address and port together are known as the *server-reflexive address*.
- This mapping is created when a TCP SYN packet is sent from inside the NAT or when a first UDP packet is sent.
- It is maintained for as long as the TCP connection or UDP flow are “alive.” Flow timeout is implementation-dependent.

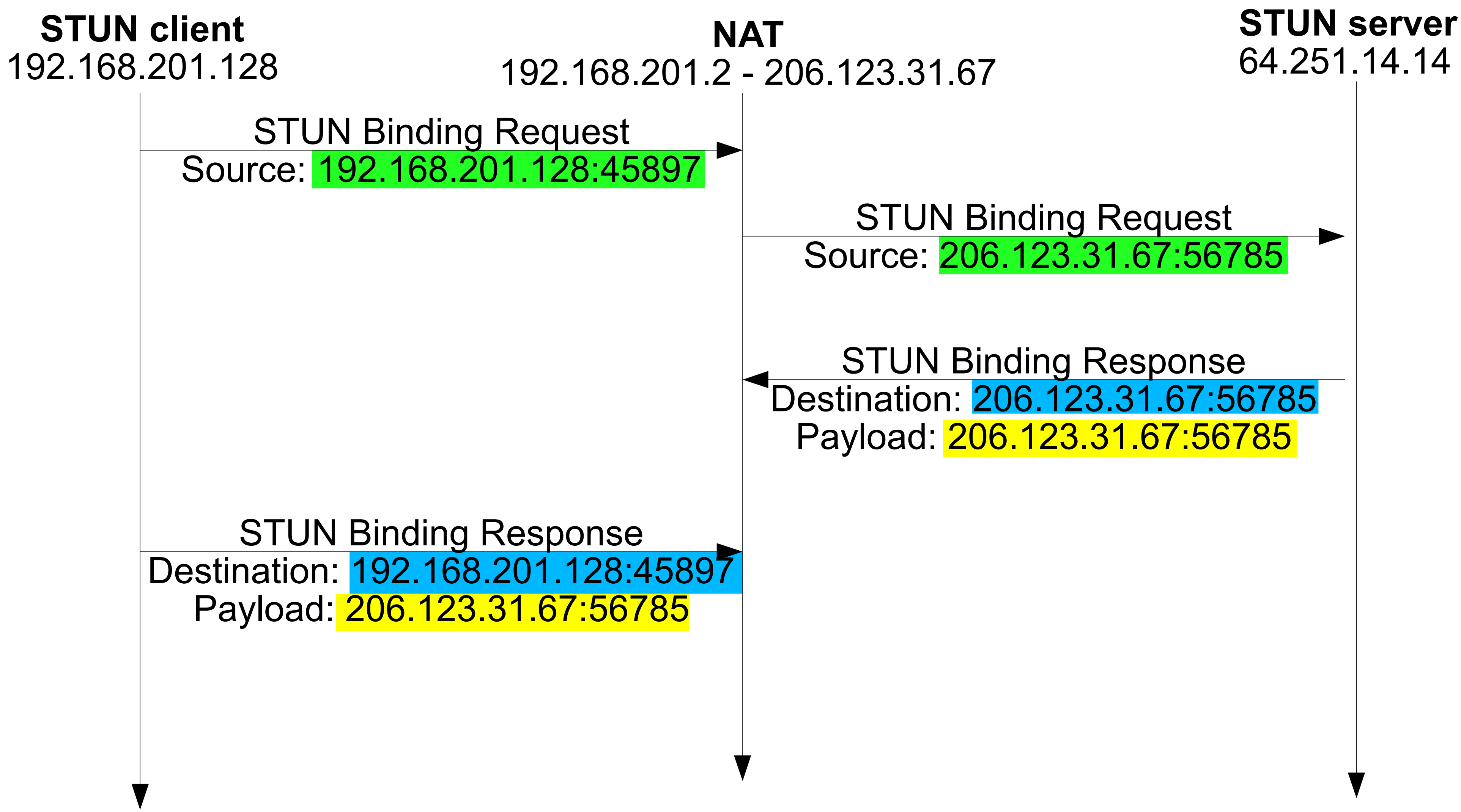
# Server-Reflexive Address

- For the majority of NAT devices (mostly home routers), any device on the Internet may contact the NATed party by sending packets to the server-reflexive address, even if they are not the receiver of the connection-initiating packet.
- A mean for discovering the server-reflexive address and communicating it to the other party is therefore needed.

# STUN

- Session Traversal Utilities for NAT (STUN) is a simple protocol for discovering the server-reflexive address.
- A STUN server is located in the public Internet or in an ISP's network when offered as a service.
- The NATed peer initiates a connection to the STUN server, thus creating a binding in the NAT device.
- The STUN server receives the query and inspects the sender address, which is the server-reflexive address.
- It sends back a reply containing the server-reflexive address in its payload.
- The client thus learns its server-reflexive address.

# STUN Flow Diagram



# STUN

- It turns out that some NAT devices try to be clever by inspecting the payloads and changing all references to the server-reflexive address into the private address.
- To address that issue, the new version of STUN (known as STUN 2, still an IETF draft) obfuscates the address by XORing it with a known value.
- TCP and UDP are supported over IPv4 and IPv6.

# Server-Reflexive Address

- A client who knows its server-reflexive address may use it in place of its private address in the SIP headers.
- The same process must be carried out for the RTP ports in the SDP, each one having its own NAT binding and needing a separate STUN request.

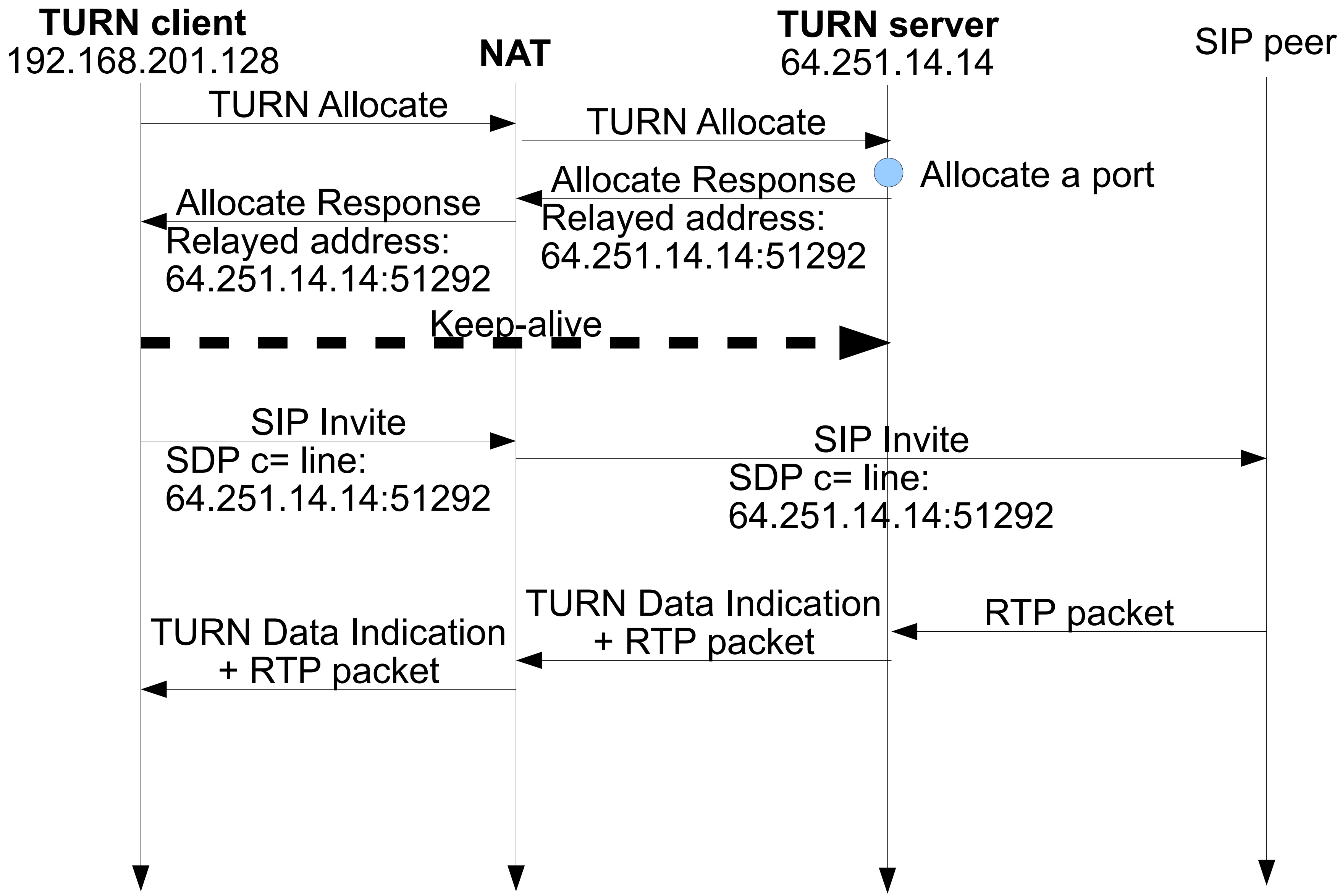
# Symmetric NATs

- Some NAT devices only allow packets from the remote peer to reach the NATed peer.
- Thus a STUN request is useless because only the STUN server could reach the NATed peer through the server-reflexive address.
- These NAT devices are called *symmetric NATs*.
- They are often “enterprise” NATs that hide more devices on average.
- Thus, their presence is significant and must be worked around.

# TURN

- To be reachable, a device behind a symmetric NAT needs to initiate and maintain a connection to a relay.
- Traversal Using Relays around NAT (TURN) is a protocol for communicating with the relay.
- Built on top of STUN.
- The TURN server is located outside the NAT, either on the public Internet or in an ISP's network when offered as a service by the ISP.
- A NATed TURN client asks the server to allocate a public address and port and relay packets to and from that address.

# TURN Flow Diagram



# Relayed Address

- The address allocated by the TURN server is called the *relayed address*.
- The TURN server communicates that address to the TURN client.
- The TURN client may use it in the SIP headers.
- Separate allocations must be made for each RTP port, and the relayed addresses may be used in the SDP.

- TURN **guarantees** communication in all NAT cases unless there is an explicit firewall policy to prohibit its use.

# Disadvantages of TURN

- TURN server is in forwarding path.
  - Requires a lot of bandwidth.
  - Server must remain available for the whole duration of the allocation.
  - Triangle routing results in longer path.
- Encapsulation.
  - Lowers MTU (not so much a problem for VoIP packets).
  - Additional headers consume a bit more bandwidth.
  - Firewall must inspect payload to discover real sender.
- Allocation must be kept alive.

# Disadvantages of TURN

- ICMP not relayed.
  - No path MTU discovery.
- TTL not properly decremented.
  - Possibility of loops.
- DiffServ (DS) field not relayed.
- As of now only IPv4 and UDP.

# Mitigating Mechanisms

- Availability and scalability provided by anycast.
  - Only used for discovery, server must remain up for the duration of the allocation.
- Channel mechanism for minimizing header size.
  - 4 bytes only.
- Permission mechanism enforced by TURN server.
  - Only peers previously contacted by client may send data to relayed address.
  - Firewall may “trust” the TURN server, no payload inspection.
- Keep TURN server close to NAT device.
  - Offered as a service by ISPs.

# IPv4 and IPv6 Interoperability

- TURN will also be used to relay packets between IPv4 and IPv6.
- Alleviates load from the B2BUA.
  - Designed for relaying performance.
  - Anycast ensures scalability and reliability.
- TURNv6 draft still in progress.

# Numb

- Numb is a STUN and TURN server developed by Viagénie.
  - Supports IPv4 and IPv6 in mixed scenarios.
  - Supports anycast.
- Free access at http://numb.viagenie.ca
- To install it in your own network, contact us: info@viagenie.ca

# Connectivity Establishment

- Many addresses may be available:
  - Host addresses.
  - Server-reflexive address.
  - Relayed address.
  - Each in IPv4 and IPv6 flavour!
  - Each in UDP and TCP flavour!
- Which one to choose?
- Need for an automatic *connectivity establishment* mechanism.

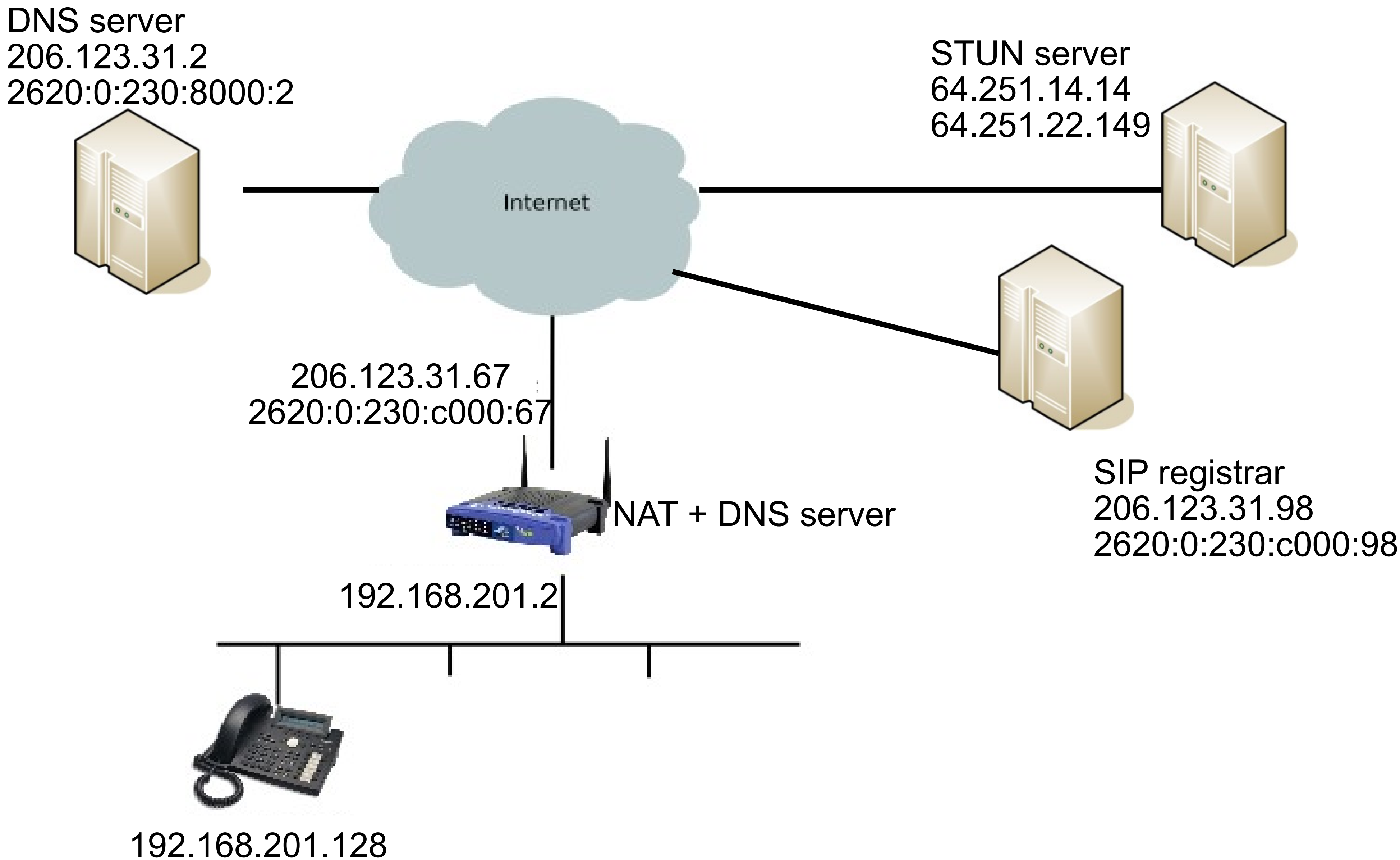
# Interactive Connectivity Establishment (ICE)

- Conceptually simple.
  - Gather all *candidates* (using STUN/TURN).
  - Order them by priority.
  - Communicate them to the callee in the SDP.
  - Do connectivity checks.
  - Stop when connectivity is established.
- Gnarly details:
  - Keep candidates alive.
  - Agree on priority.
  - Reduce delays and limit packets.

# Peer-Reflexive Address

- Server-reflexive address useless with symmetric NAT.
- Address as seen from peer (instead of STUN server) is *peer-reflexive address* and does work even with symmetric NAT.
- During ICE connectivity checks, peer-reflexive candidates are gathered and prepended to check list.
- TURN relay still necessary when both peers are behind symmetric NATs.
- STUN requests need to be multiplexed with RTP.
- Information reuse between ICE instances.

# Examples


DNS server
206.123.31.2
2620:0:230:8000:2
Internet
STUN server
64.251.14.14
64.251.22.149
206.123.31.67
2620:0:230:c000:67
NAT + DNS server
SIP registrar
206.123.31.98
2620:0:230:c000:98
192.168.201.2
192.168.201.128

# Deployment

- ISPs are deploying STUN / TURN servers within their network.
- TURN a part of the IPv6 migration.
- SIP client vendors are implementing ICE.
- B2BUAs also should implement ICE.

# Conclusion

- Discussed
  - The problem of NAT and firewalls in VoIP
  - How STUN, TURN, and ICE solve it
    - Obtaining a server reflexive address via STUN
    - Obtaining a relayed address via TURN
    - Telling the other party about these addresses via ICE
    - Making connectivity checks
    - Obtaining peer reflexive addresses
- STUN / TURN / ICE stack too thick? Use IPv6!

# Questions?

Simon.Perreault@viagenie.ca

This presentation: http://www.viagenie.ca/publications/

STUN / TURN server: http://numb.viagenie.ca

References:

STUNv1 RFC: http://tools.ietf.org/html/rfc3489

STUNv2 draft: http://tools.ietf.org/html/draft-ietf-behave-rfc3489bis

TURN draft: http://tools.ietf.org/html/draft-ietf-behave-turn

ICE draft: http://tools.ietf.org/html/draft-ietf-mmusic-ice