Ecdysis: Open-Source DNS64 and NAT64

Jean-Philippe Dionne, Simon Perreault and Marc Blanchet

Viagénie

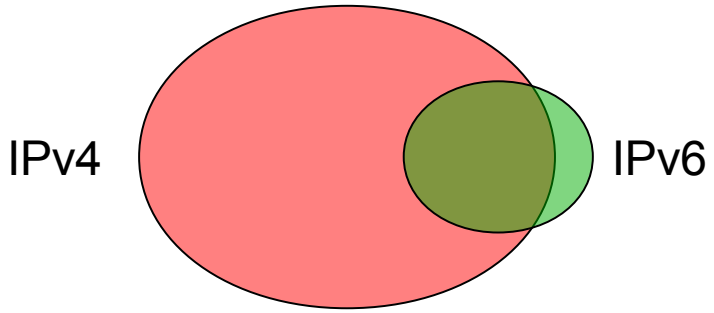
Partner: NLnet Foundation

Plan

- IPv4 and IPv6 coexistence
- Ecdysis Project
- DNS64 and NAT64 specifications
- Use Cases
- Implementations
- Improvements
- Lessons learned

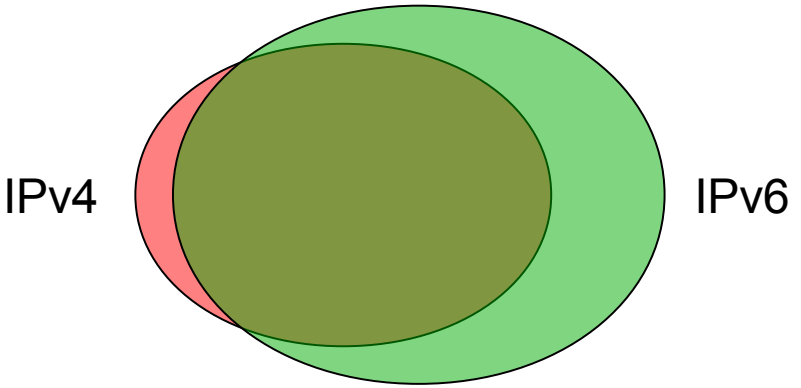
VIAGÉNIE

Today



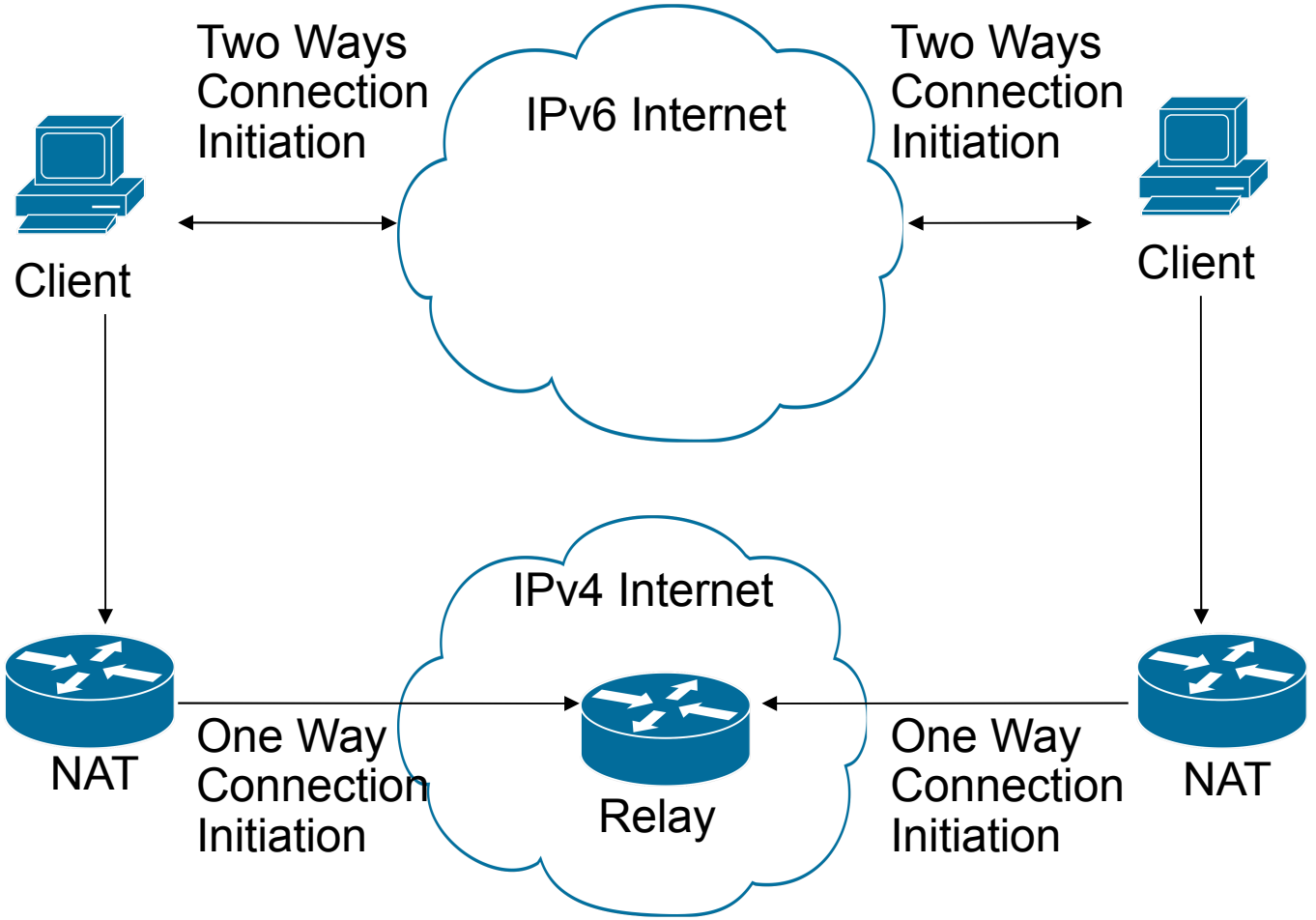
- Most users are using a single IPv4 stack
- Few dual-stack users
- No IPv6 single stack users!

Tomorrow



- Increasing number of IPv6 single stack users.
- Need to access valuable content on the IPv4 Internet

IPv4 Internet: NATs and Relays



The problem

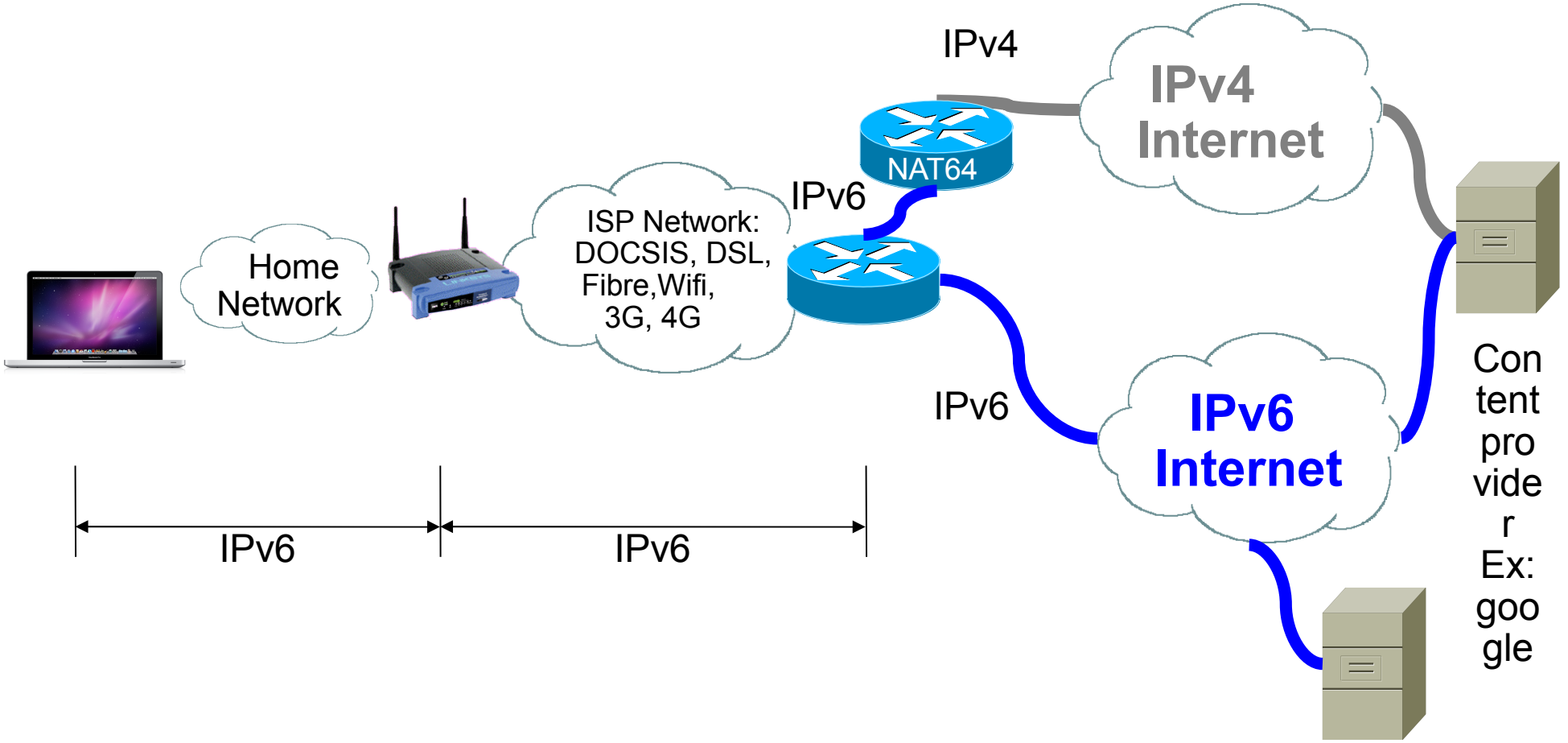
- We are soon going to have a new category of users using only IPv6.
- IPv6 in IPv4 tunnels (such as freenet6) doesn't enable new protocol users to communicate with old protocol users without dual-stack hosts.

How do we connect an IPv6-only network to the IPv4 Internet?

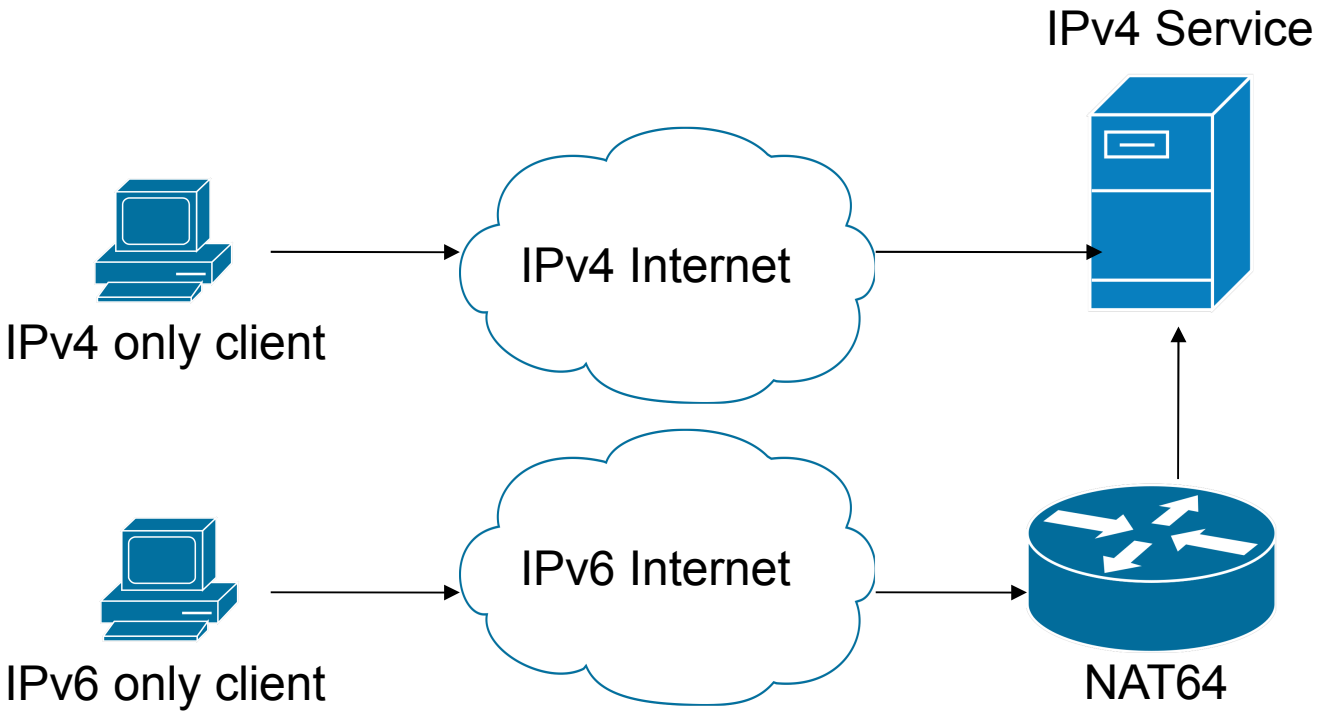
Ecdysis: Open-Source DNS64 and NAT64

- Funded by NLnet and Viagénie.
- Ecdysis refers to the moulting of the cuticula in arthropods, as an analogy of IPv4 molting into IPv6. After molting, the arthropod is fresh and ready to grow! Arthropods is also the expertise of the 5 years old son of one of the project leads...
- http://ecdysis.viagenie.ca

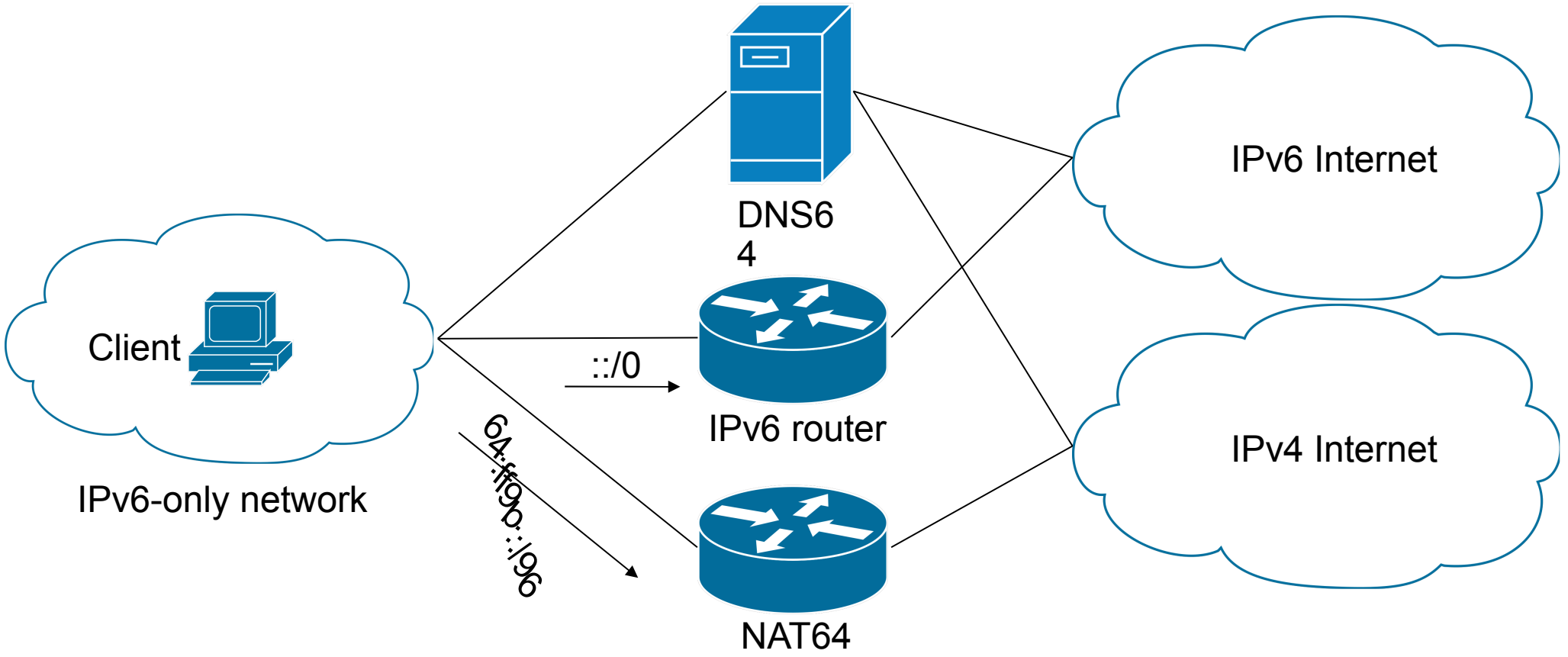
ISP Use Case


IPv4
IPv4 Internet
NAT64
IPv6
ISP Network: DOCSIS, DSL, Fibre,Wifi, 3G, 4G
Home Network
IPv6
IPv6 Internet
Content provider Ex: google
IPv6
IPv6
Content Provider

IPv4 Content Provider Use Case



Example

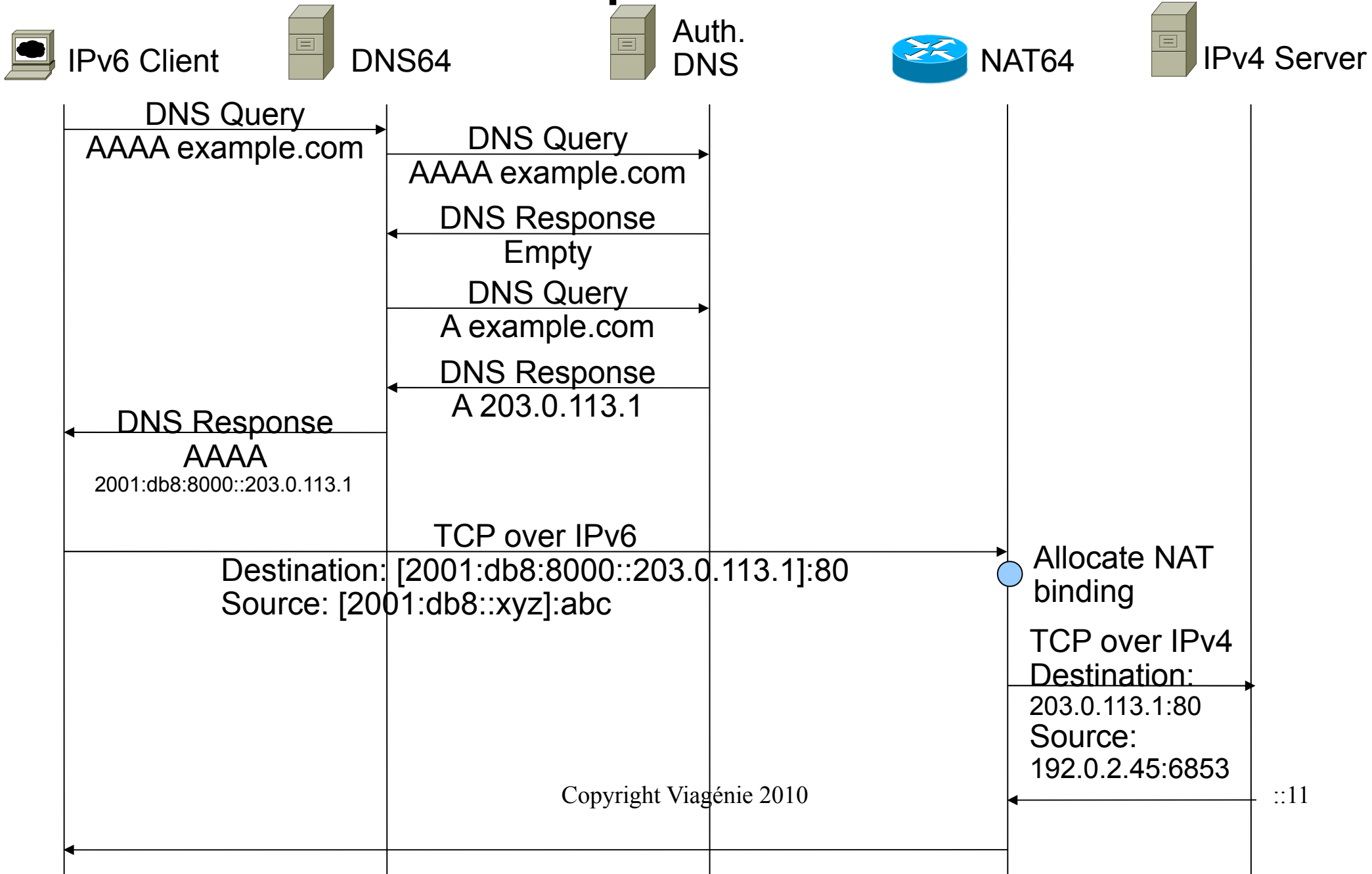


- DNS64 and NAT64 are completely separate.
- No shared state.

Example

- IPv6 network → 2001:db8::/64
- NAT64 → 2001:db8::1
- DNS64 → 2001:db8::2
- NAT64 public IPv4 pool → 192.0.2.0/24
- Pref64::/n → 2001:db8:8000::/96

Example



DNS64

- All connections need to start with a DNS query.
  - No DNS, no translation.
- Because DNS64 synthesizes records on the fly, it interferes with DNSSEC.
  - If client trusts DNS64, no problem.
  - If client knows about the special DNS64 prefix, no problem.
  - Otherwise, DNSSEC validation by the client fails.

NAT64

- Uses a special prefix, called Pref64::/n
  - Well-known prefix: 64:ff9b::/96
    - Starts with “64”.
    - Checksum-neutral.
  - Can use part of your own allocation.
    - Multiple NATs, multiple prefixes.
  - Can be shorter than 96 if you want to route on part of the IPv4 address.
- Precisely specified. Tries to make NAT traversal easy.

Ecdysis Implementations

- DNS64
  - Stand-alone in Perl for quick experimentation.
  - Patch against ISC Bind.
  - Patch against NLnet Unbound.
- NAT64
  - User-space using tun(4). (not open-source)
  - Linux module (using Netfilter).
  - Patch against OpenBSD (pf).

DNS64 in Unbound

- Unbound is a non-authoritative DNS server with modular architecture
- DNS64 is in dns64.so module loaded at run-time.
- Other modules:
  - validator.so: DNSSEC processing.
  - iterator.so: Recursive DNS resolving.
- dns64.so is placed at front of chain. It modifies AAAA responses with empty answer section.

DNS64 in Bind

- Changes to core state machine, in query_find().
- New configuration variable, dns64-prefix.
- Example:

```
options {
        [...]
        dns64-prefix 64:ff9b::/96;
   }
```

- DNS64 is disabled if dns64-prefix is not specified.
- Currently running at TNC2010 (2001:778::37) !

nat64d

- User-space NAT64 using tun(4).
- Creates a tun interface to which you route Pref64::/n and the IPv4 address.
- Portable without too much effort.
- Uses sys/tree.h for its mapping table.
- SIGUSR1 --> print mapping table to STDOUT
- SIGHUP --> flush session table
- Easy development, easy experimentation, portability.

NAT64 in Linux

- Take nat64d, put it in kernel-space.
- tun(4) I/O replaced with Netfilter hooks and a dummy “nat64” interface.
- Currently running at TNC2010 (Pref64: 2001:778::ffff:64/64)

NAT64 in OpenBSD's pf

- Configuration syntax:

```
nat64 from any to 64:ff9b::/96 -> (em0)
```

- After 4.7, will become:

```
match from any to 64:ff9b::/96 nat64-to (em0)
```

- “rdr64” action is planned for future work.
- A new action keyword was preferred instead of implicit or explicit address family identification.
  - To ensure that “nat” always does what we're used to.
  - To not imply that nat from IPv4 to IPv6 is possible.

Comparison

- User-space implementation as close to IETF spec as possible.
- Linux implementation also, by inheritance.
  - But needs better integration with Netfilter facilities.
- pf implementation:
  - Works with user-space tools (e.g. pfctl(8), systat(1).
  - State synchronization with pfsync.
  - CARP.

Next Steps

- Use IPv4 address pool for NAT64.
- Scale with DNS load balancing and CARP
- Application Layer Gateways (ALG)

Load Balancing

- DNS-based Load Balancing
- Multiple CARPed pairs of NAT64 boxes.
- Each pair has its own Pref64::/n.
- DNS64 picks a prefix when synthesizing AAAA records.
  - Round-robin.
  - Random.
  - Hash of client's address.
  - Based on load feedback from NAT64.

Lessons Learned

- Things that work behind an IPv4 NAT do not necessarily work behind a NAT64.
  - Address literals in WWW hyperlinks.
  - MSN, Skype
- Strange behaviour with the resolver on Snow Leopard
  - In IPv6-only network, CNAMEs don't work.
  - Caused by a race condition while resolving A and AAAA simultaneously.
- Network managers think you're offline
  - Need to uncheck “Work offline” in Firefox.
  - Network manager sometimes automatically tries other SSID.

Lessons Learned

- Configuration of DNS in clients
  - Multiple options but not yet found in the default/automatic configuration of popular operating systems.
  - DHCPv6
    - stateful
    - stateless
  - IPv6 Router Advertisement Option for DNS Configuration
  - Site-local dns server address (fec0:0:0:ffff::1)
  - DHCPv4 server, just for DNS queries.

Demo at TNC2010

VIAGÉNIE

- Configuration steps
  - Disable IPv4 autoconfiguration
  - Enable IPv6 autoconfiguration
  - Manually configure dns server: 2001:778::37
  - Clear dns cache
    - Mac OS: “dscacheutil -flushcache”
- Give us feedbacks!

Conclusion

- NAT64 is usable now for “web & email” style of usage.
- Code is available now for you to try it and hack on it.
- We love patches!
  - Goal: clean up and push upstream.

Questions?

jean-philippe.dionne@viagenie.ca

http://ecdysis.viagenie.ca

Special thanks to:

- NLnet Foundation
- OpenBSD: Ryan McBride, David Gwynne

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