sNATPT
Simplified NAT-PT

H.Miyata/M.Endo

IETF Co-existence Interim Meeting @ Montreal

First of all

- We submitted three documents
  - Draft-miyata-v6ops-trans-approach
    - Insisting the requirement URGENT SOLUSION
    - Not technical document
  - Draft-miyata-v6ops-snatpt
  - Draft-endo-v6ops-dnsproxy

Why we make separate documents

sNATPT, Translator Friendly DNS Proxy (TFDP)?

- Not Every Application uses DNS to resolve the address
  - DNS is just an (but typical) example
  - Translator principle must be able to work individually

sNATPT

- DNS is a key component of Internet
  - DNS behavior must be considered

TFDP

- Consistency must be guaranteed
  - Co-operate with DNS should be considered

TFDP

What is sNATPT

- Purpose
  - Enabling IPv6 Client to connect IPv4 Server
  - Enabling IPv4 Client to Connect IPv6 Server
- Approach
  - Recycling NATPT(2766)
  - Eliminating DNSALG Dependability
    - Focusing on packet translation
  - Enhancing NATPT(2766)
  - Don't require any change to both IPv4 and IPv6 node for minimum access
  - Require advanced feature for advanced access
    - E.g. Detect synthetic address
  - Don't beyond the IPv4-IPv4 NAT

sNATPT Features(1/2)

- Bi-Directional Translation
  - (inc. Unidirectional, of course)
  - Stateful Translation
- Address Type
  - Unicast, Multicast
- Protocol
  - TCP, UDP, ICMP(Information, Error)
  - RFC2765(SIIT) compliant

sNATPT Features(2/2)

- Fragmented Packet
  - Ordered, Out-of-Ordered
  - TCP, UDP(w/ Checksum, w/o Checksum)

Is it really required?
RFC2765 clearly stated it is not worth to translate…

- Dummy Prefix and Synthetic Address
  - Basic
    - A Part of unassigned administrative prefix
    - Normal unicast prefix
  - Advanced
    - IDENT Bit in can indicate the synthetic address

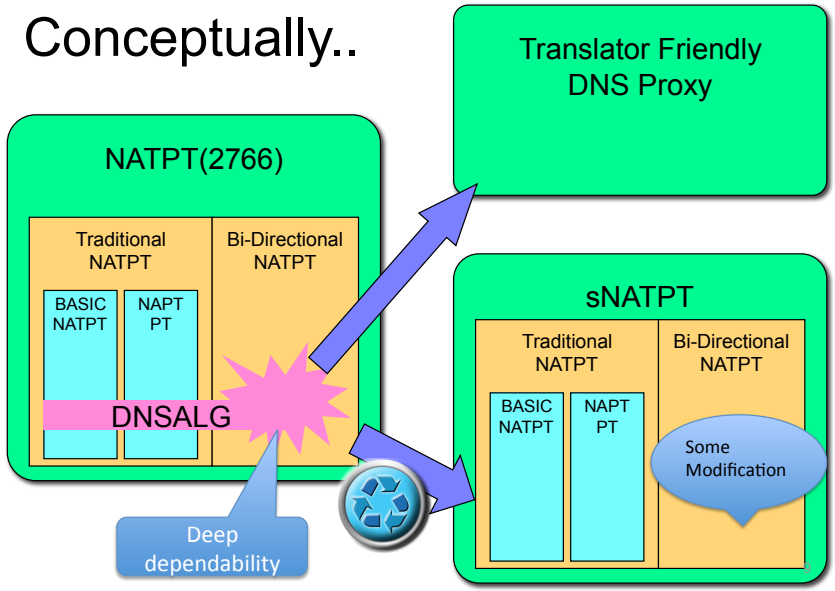
What is TFDP?

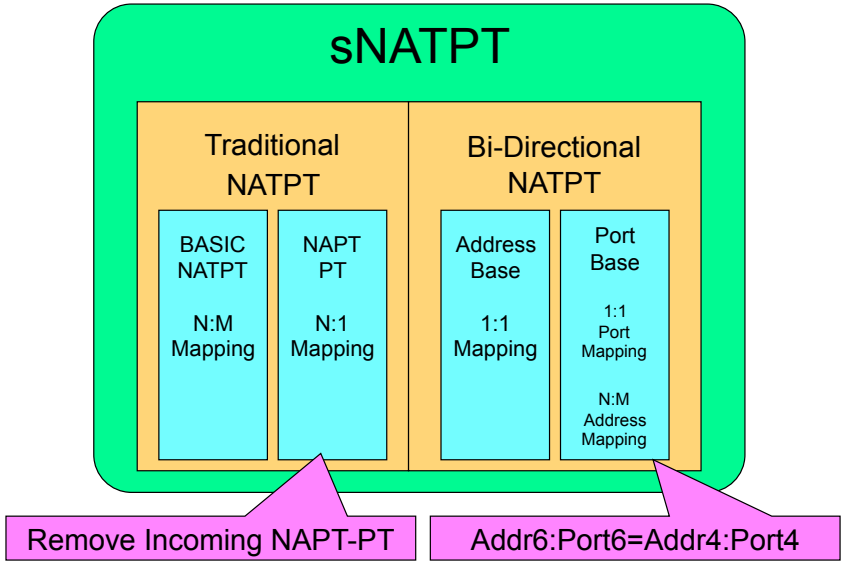
- Purpose
  - Resolve some DNS-ALG related issues described by RFC4966
  - adopt additional functionalities
- Approach
  - Translator independent
  - Transport independent
    - The behavior doesn't depend either transport and network protocol
  - Sequential resolving
    - At first, try to get a real RR.

TFDP Features

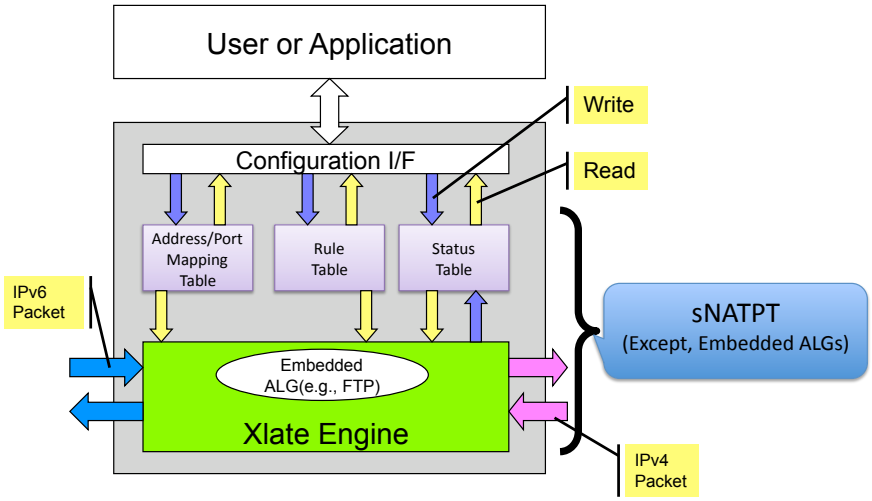
- Synthesize Resource Record
  - A/AAAA/PTR(option)
- Dynamic Address mapping (IPv4->IPv6)
  - Pre-configured IPv4 address pool
  - One by One mapping
- Translator load balancing
  - Selecting a prefix means selecting a translator.
- DNSSEC
  - EDNS ORR option

Conceptually..

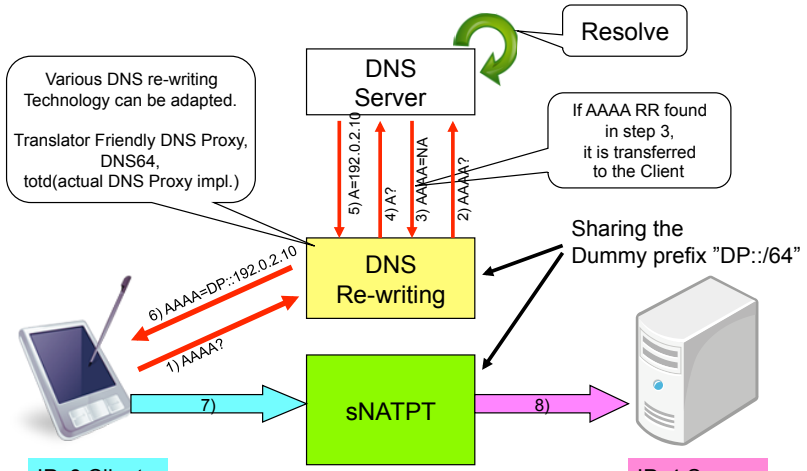




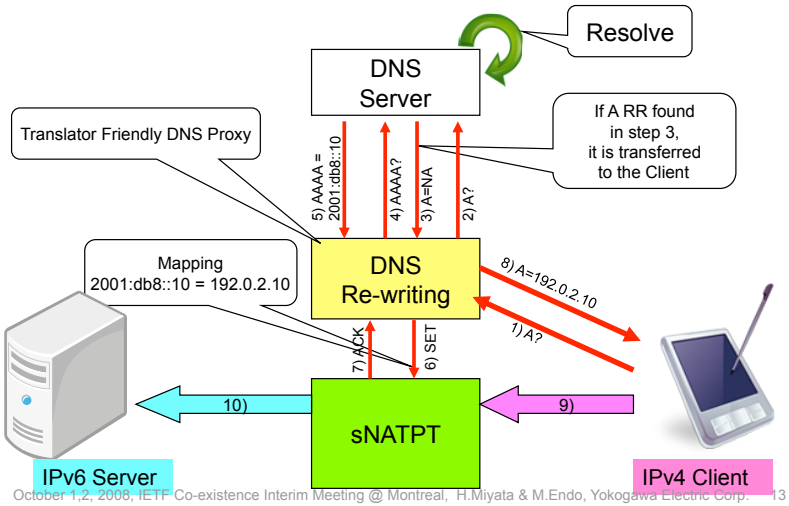
Architecture



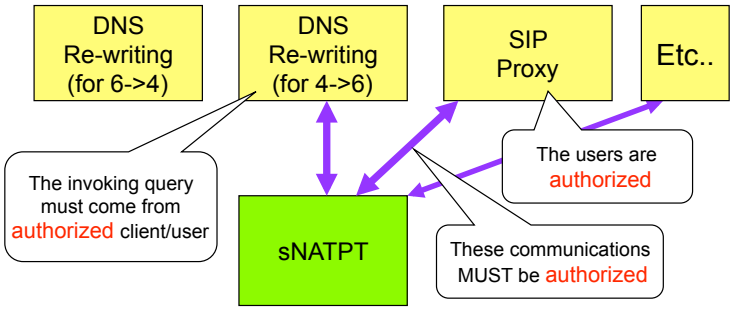
How to co-work (1/2)
IPv6 Node initiates connection(6->4)



How to co-work (2/2)
IPv6 Node initiates connection(4->6)

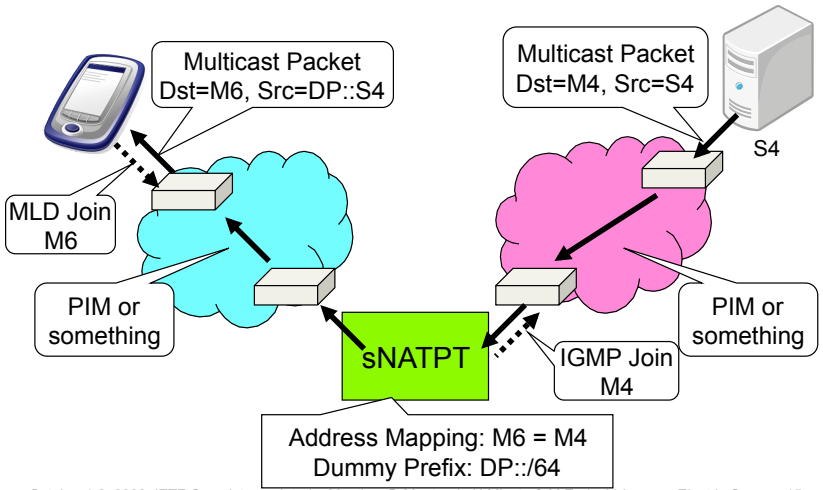


sNATPT friendly ALGs (Proxies)

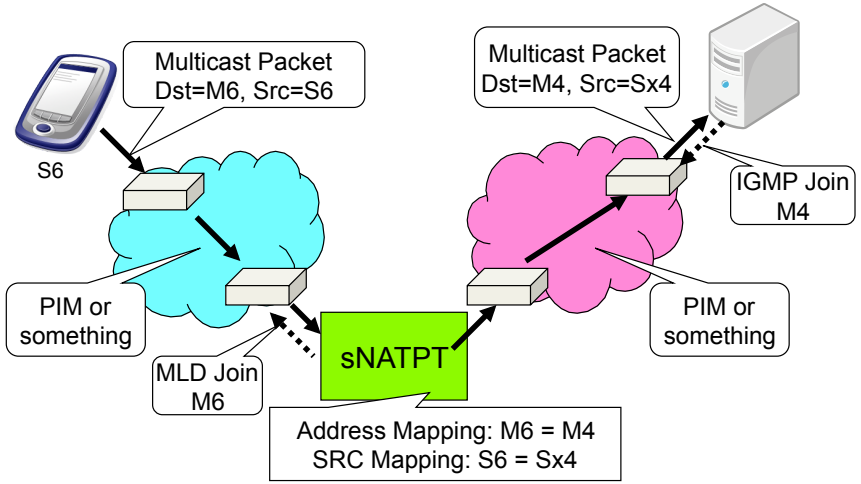


Since, sNATPT is configurable,
it can cooperate with other systems
like DNS Proxy, SIP Proxy, etc..
The dynamic allocation should not be used in Anonymous service.

Multicast Translation (IPv4 -> IPv6)



Multicast Translation (IPv6 -> IPv4)



ICMP Error translation

- IPv4 -> IPv6
  - No change
- IPv6 -> IPv4
  - Destination Unreachable (Type 1)
    - Code 2 - Beyond scope of source address
      - Set Code to 1 (host unreachable).
    - Code 5 - Source address failed ingress/egress policy
      - Set Code to 1 (host unreachable).
    - Code 6 - Reject route to destination
      - Set Code to 1 (host unreachable).

Fragment (1/2)

- NAT-PT(RFC2766) challenged to support UDP w/o Checksum packets
- SIIT(RFC2765) abandons supporting UDP w/o Checksum packets

If we gave up to support UDP w/o Checksum packet.
Basic-NAT-PT and Basic-Bi-Directional-NAT-PT
do NOT NEED to REASSEMBLE.
NAPT-PT and Port-Mapping still NEED to REASSEMBLE.

Fragment (2/2)

- Reassembling IPv4 fragmented packet solves issues
- But, we still have remaining issue
  - Resource consumption
  - Delay
  - DoS attack

No way, if reassembling is required…

Must be prevented

- We can introduce optional behavior to be light
  - We have good experiences in IPv4 NAT

Solutions for DNS issues (1/2)

- Issues Exacerbated by the Use of DNS-ALG
  - Network Topology Constraints Implied by NAT-PT
    - Separating DNS-ALG from a translator
  - Scalability and Single Point Failure Concerns
    - Separating DNS-ALG from a translator
    - Some redundancy mechanism are needed
  - Issues with Lack of Address Persistence
    - No solution
  - DoS Attacks on Memory and Address/Port Pools
    - SHOULD make a limitation or authorize clients

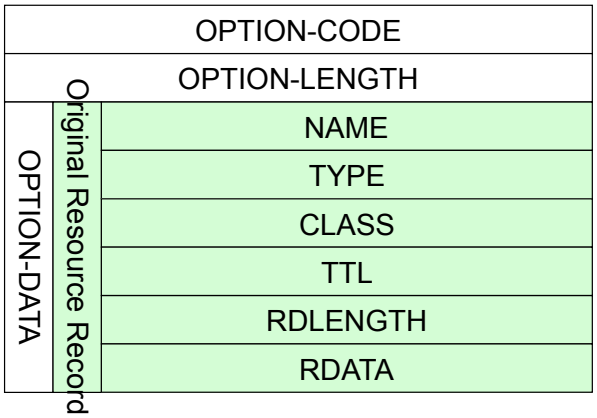
Solutions for DNS issues (2/2)

- Issues Directly related to Use of DNS-ALG
  - Address Selection Issues when Communicating with Dual-Stack End-Hosts
    - This DNS Proxy doesn't reply both real address and synthesized address
  - Non-Global Validity of Translated RR Records
    - No solution
  - Inappropriate Translation of Response to A Queries
    - This DNS proxy is stateful.
  - DNS-ALG and Multi-Addressed Nodes
    - No solution
  - Limitation on Deployment of DNS Security Capabilities
    - ORR option

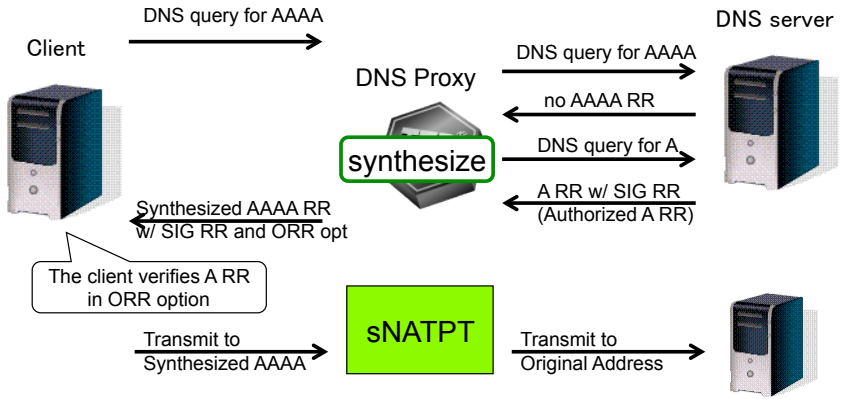
DNSSEC Compliant(1/3)
EDNS Original Resource Record option

- ORR option indicates that the answer section has synthetic RRs.
  - If RR was authorized and synthesized, DNS Proxy adopts this option with Original Resource Record.
    - e.g., In A -> AAAA case, ORR option has A RR.
- RR verification can be finished in one DNS exchange.
  - RR included ORR option can be used to verify.
  - DNSSEC implementation should be modified.

DNSSEC Compliant(2/3)
ORR option format



DNSSEC Compliant(3/3)
DNS exchange w/ ORR Option

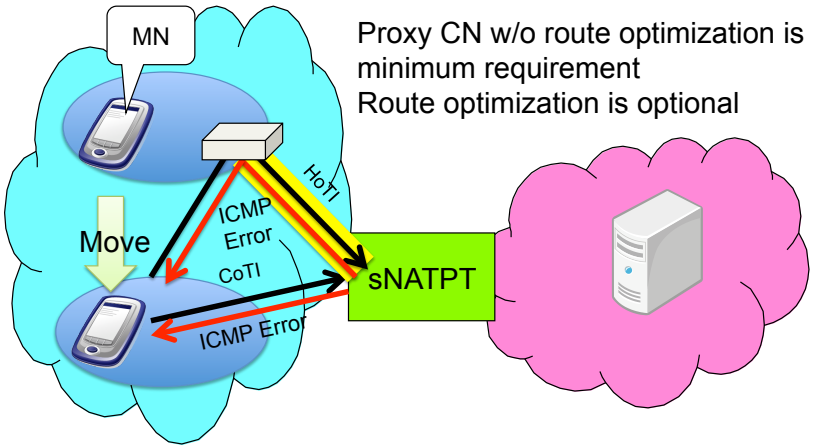


EDNS0 or IDENT Bit

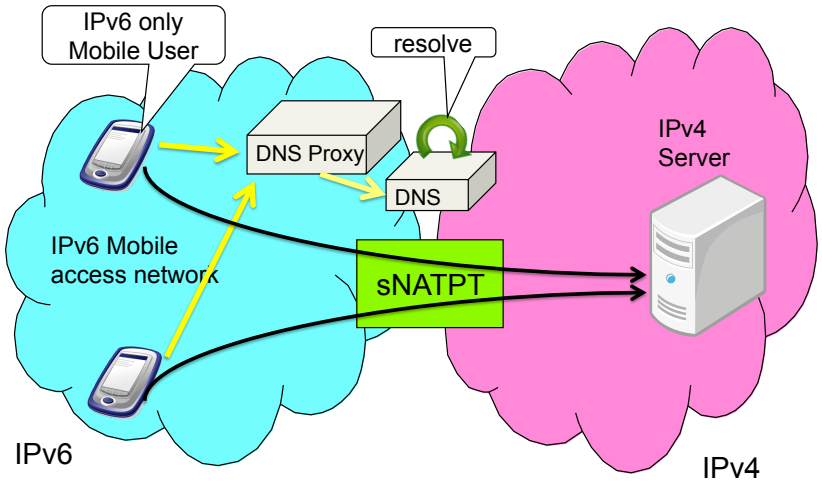
- Translator Friendly DNS Proxy and DNS64 proposed EDNS0 to transfer DNSSECed RR
- sNATPT Proposed IDENT bit to indicate Synthesized address

Since some application/environment does not use DNS to resolve the address, both complement each other.
The administrator can register synthetic address to original AAAA record, it is DNSSEC compliant and don't need DNS rewriting.
IDENT is minimum requirement as a common solution.
EDNS0 would be helpful to support DNSSEC.

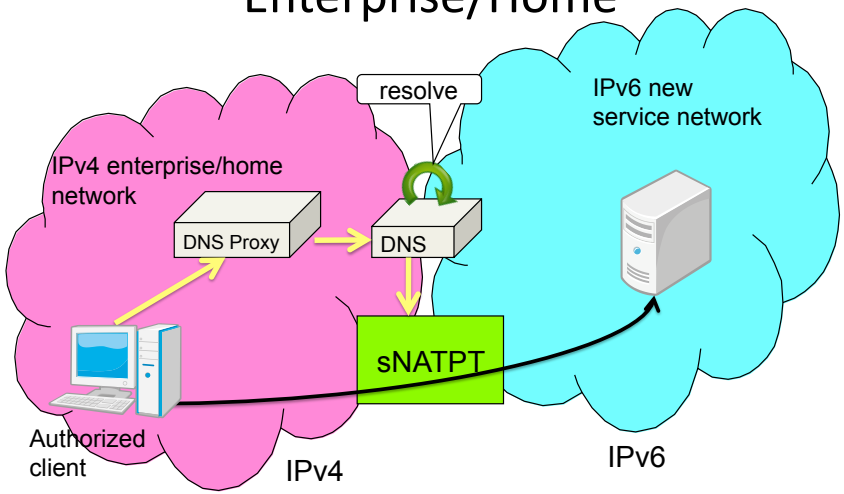
MIPv6 Proxy CN



Use Case (client side)
Mobile Access service

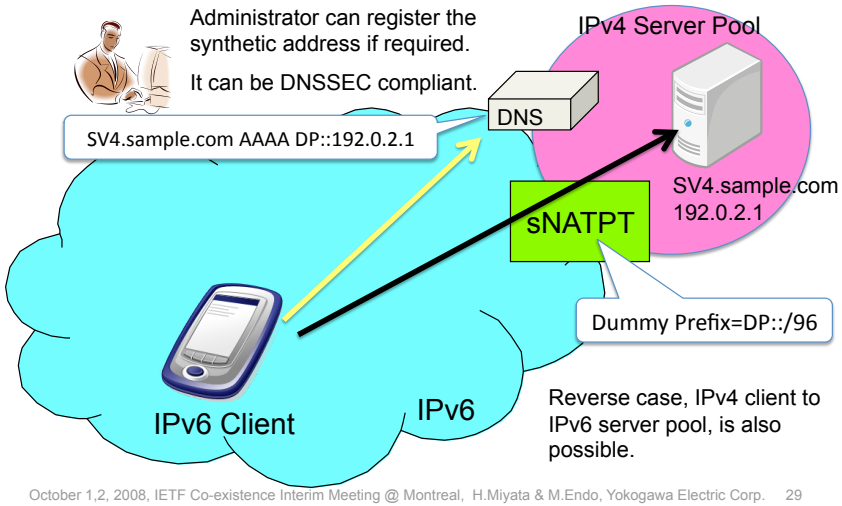


Use Case (client side)
Enterprise/Home



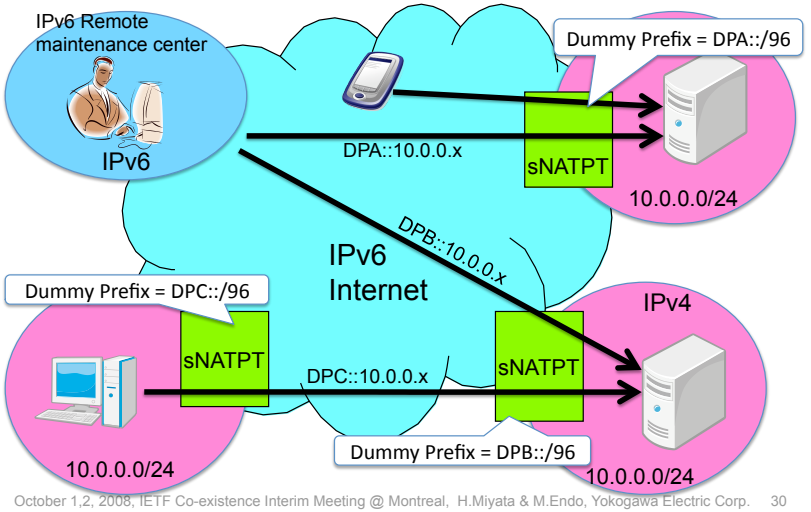
Use Case (Server Side)

In front of server pool



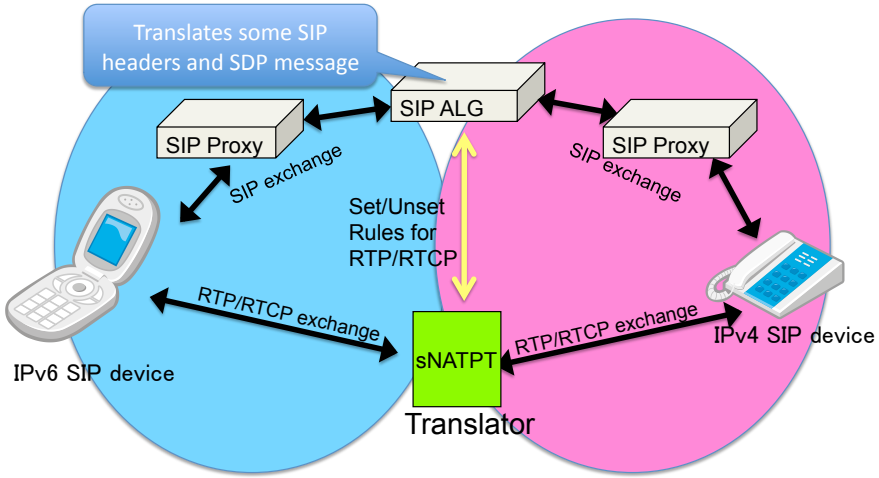
Use Case (Server Side)

Virtual Global Network



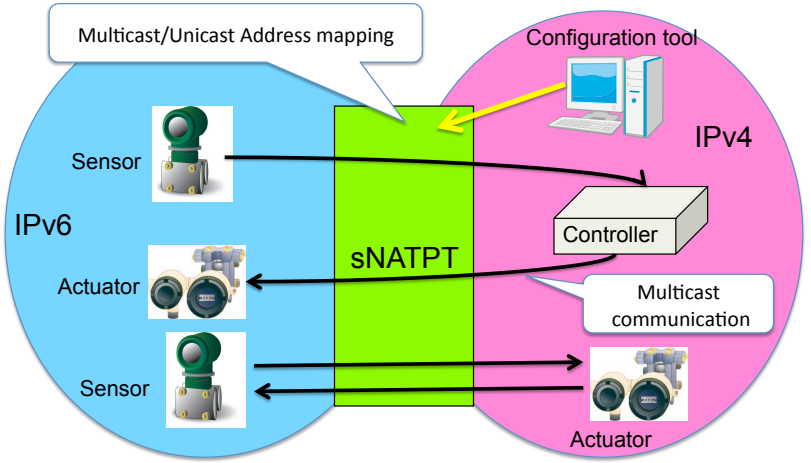
Use Case (boundary)

SIP translation



Use Case (end-to-end)

Fieldbus(sensor and control) Protocol



Proposal from TFDP

- Some applications cache the DNS response
  - Even though the RR has TTL, it does not care TTL
  - It is not load balance and dynamic allocation friendly behavior
- There was an discussion…
  - TTL is not for application but for DNS Server
  - It might be true when application does not cache the DNS response

Any implementation SHOULD take care the TTL, when caching DNS Response. PLEASE!