

# **RFC3484 & shim**

**draft-arifumi-ipv6-policy-dist-00.txt**

**draft-fujisaki-dhc-addr-select-opt-01.txt**

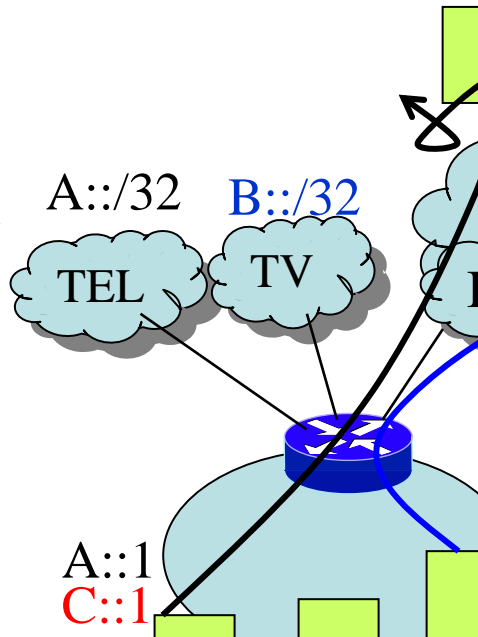
**Arifumi Matsumoto**

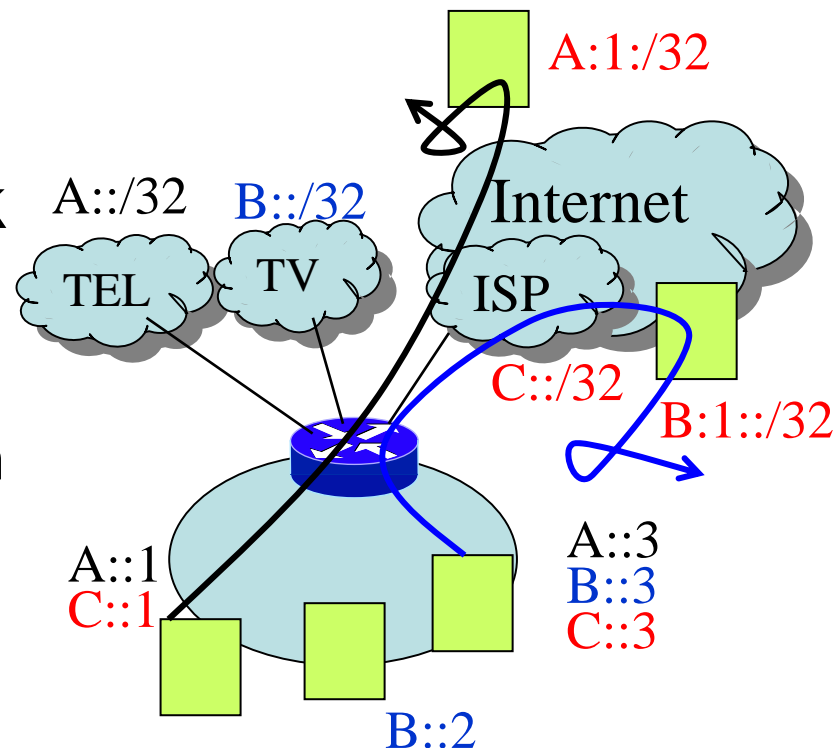
**NTT Information Sharing Platform Labs.**

# Initial Contact Is Important

- Initial contact must succeed if we use deferred shim context setup.
- Even if the corresponding host isn't shim-enabled, initial contact should succeed.
  - This is necessary for deployment
  - You can't re-write addresses at mid-path routers.
- Long connect loop in initial contact annoys users.
- **What can we do for initial contact issues ?**

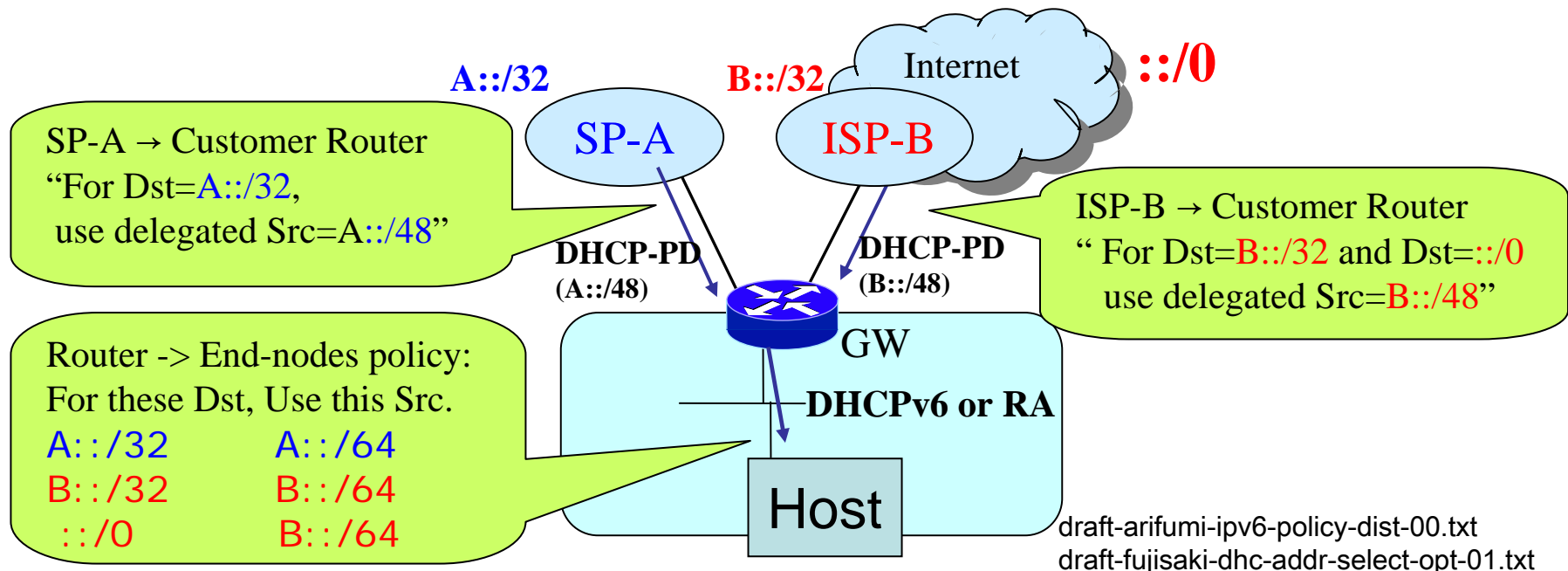
# Initial Contact Failure Reasons

1. Network outage somewhere in network
    - Dynamic problem.
  2. Ingress Filtering
    - Rather **static problem**.
  3. ISP and Closed-Network multihome environment
    - **Static problem**.
    - Deployed rapidly in Japan
    - You can't implement a large SP with ULA.
- 
- The diagram illustrates a multihome network environment. At the top, three clouds represent external networks: 'A::/32' (containing 'TEL'), 'B::/32' (containing 'TV'), and 'C::/32' (containing 'I'). These are connected to a central blue router. Below the router, a large light blue cloud represents the local network, with a red label 'A::1' and a red label 'C::1' indicating specific addresses. A black line with a curved arrow indicates a path or connection from the router to a green box on the right. A blue line also connects the router to the green box. The green box is labeled 'I'.



# Address Selection Policy for initial contact

- xSP distributes **Address Selection Policy** to its customers
  - Ingress Filtering(2) can be solved if GW converts policy into routing table or does src address routing.
  - Closed Network problem(3) can be solved.



# Redundancy for initial contact

- **An Application should cycle dst & src.**
  - An application needs ordered list of src addrs.
  - RFC3484 can choose better address first if policy is given.
  - How about **a hack for *getaddrinfo()***
    - Returns dst & src pair, like:  
dst1(fl=1),dst1(fl=2),dst2(fl=1),dst2(fl=2),... fl=flowlabel
    - *connect(dst1(fl=1));* binds src1  
*connect(dst1(fl=2));* binds src2
    - **No need to modify applications** if not statically linked.
    - Limit the number of src-dst pairs for usability ?
      - Only the addrs on the outgoing IF can be candidates ?

# TE for initial contact

- Outgoing TE (client side)
  - Depends on src addr selection
  - Better traffic control will be possible if policy table has ***Preference*** field also for src addr.
    - Or **DNS SRV** record like field
- Incoming TE (server side)
  - By DNS SRV record

# Summary

- In shim, RFC3484 is usable, but not enough for redundancy and TE.
- Policy distribution (or its modification) is necessary.
  - For network specific address selectionn policy.
  - To prevent unnecessary retry.
- So,
  - Modify RFC3484 to be shim-ready ?
    - **Preference** for src addr selection or **SRV RR** like value ?
    - Suggestion for Redundancy
    - TE capability.
  - Or do it in another place ?