Multihoming Support based on Mobile Node Protocol LIN6

*Arifumi Matsumoto, *Kenji Fujikawa, *Yasuo Okabe

**Fumio Teraoka, **Mitsunobu Kunishi

***Masataka Ohta, ****Masahiro Ishiyama

*Kyoto University, **Keio University

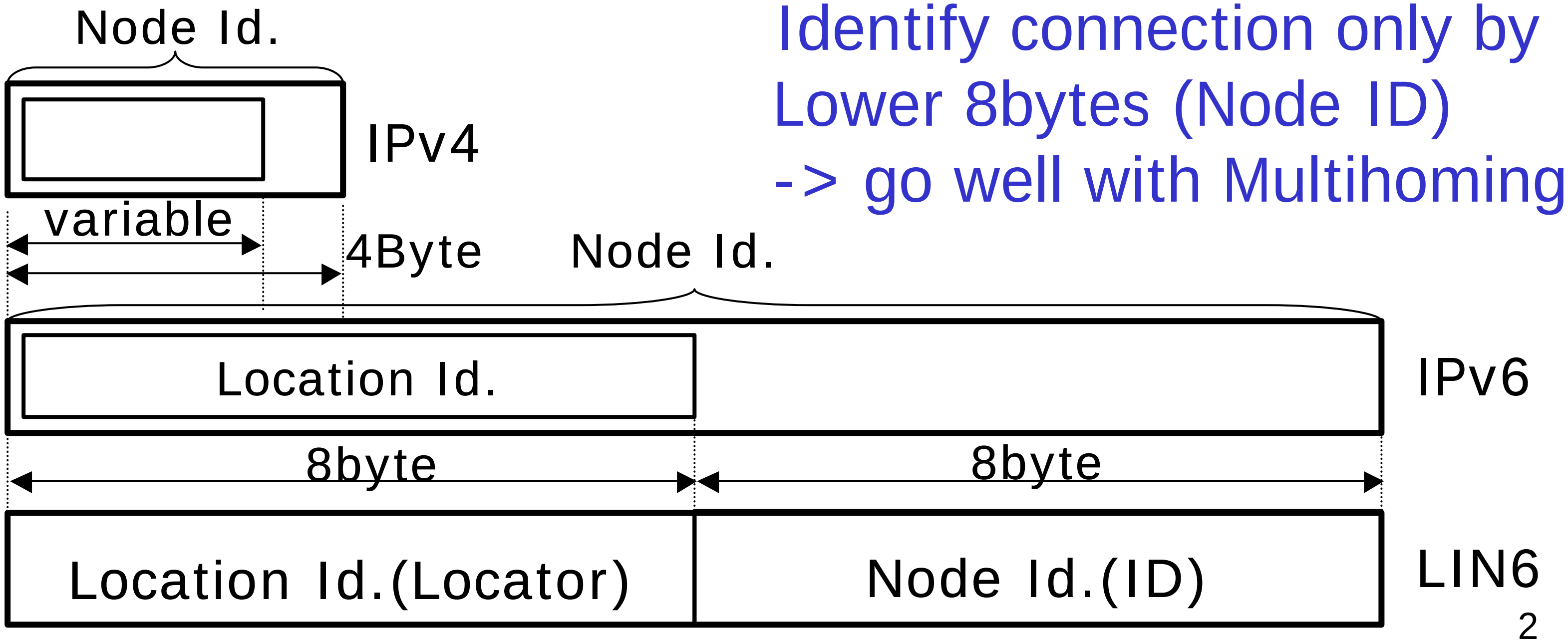
***Tokyo Institude of Technology

****Toshiba Corporation

Jan. 28, 2003
SAINT

What is LIN6 ?

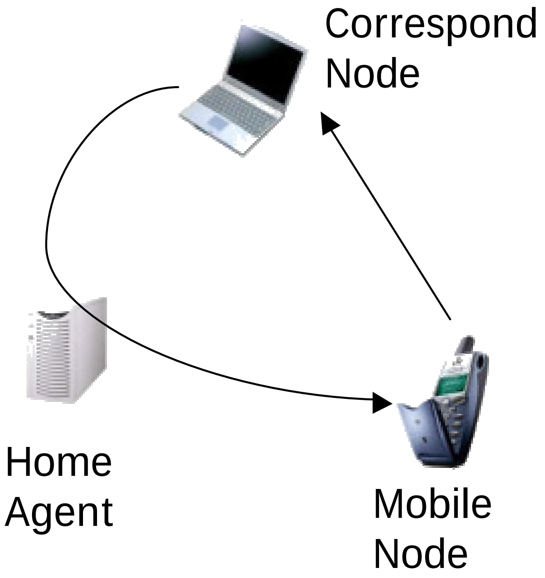
- LIN6:Location Independent Network Architecture for IPv6
- 8+8 model: Divide an IPv6 address into 2 parts
  - Upper 8bytes: Location Id.(Locator)
  - Lower 8bytes: Node Id.(Identifier)



Identify connection only by Lower 8bytes (Node ID)
-> go well with Multihoming

LIN6 solves many problems in…

- IETF MobileIP
  - Triangular routing(CN→HA→MN)
  - -> Connection overhead (delay)
  - -> Security problem (Ingress/EgressFilter)
- IETF MobileIPv6
  - Longer address and more option headers(HAO, RH)
  - -> More header overheads
  - -> Network Delay (serious for VoIP)



Needs for Multihoming

- Needs for more robust, fast Internet Connection
- Access-line is getting cheaper, broader

⬇

Multihoming using multiple up-stream lines

- connection gets more Redundant/Reliable
- connection gets Faster/Broader

By making LIN6 to support multihoming, we propose a new network architecture for both Mobility and Multihoming

Multihoming used to be…

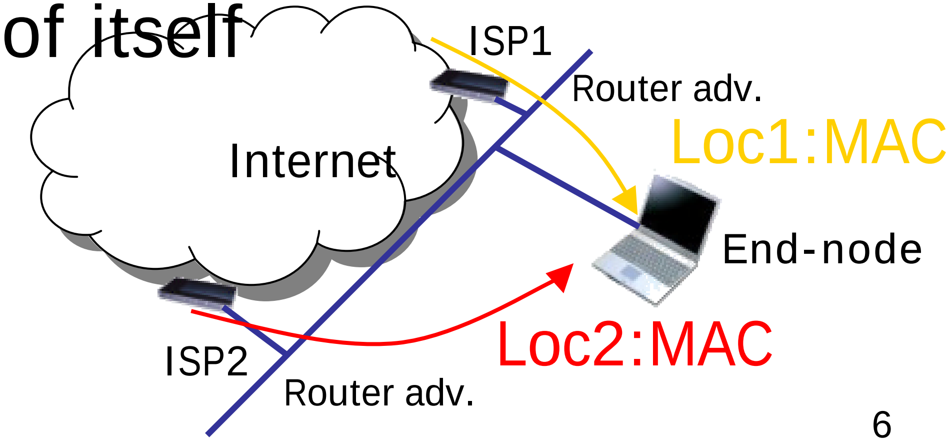
- Router Dependent Multihoming
  - Site border router advertises one prefix to multiple up-stream routers
  - -> spoils the hierarchical structure of IPv6 addressing
  - -> makes Global Routing Table too large
  - -> shortage of AS ID

New Multihoming method is required

End-to-End Multihoming

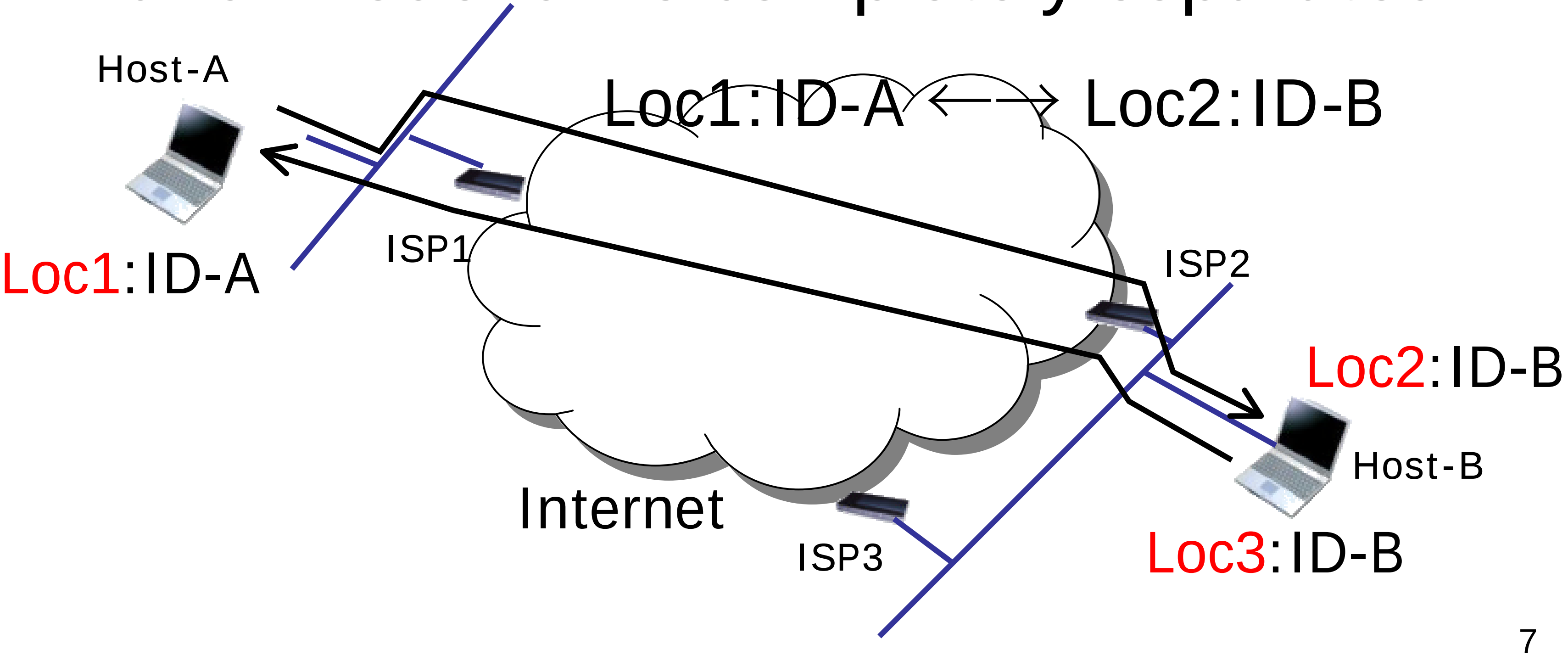
- draft-ohta-e2e-multihoming-03.txt
- Multihoming not by intelligent routing protocols but by a pair of end-system
- Each node has multiple addresses correspinding to the # of up-stream ISPs.
- TCP/UDP/App. identify each connection using multiple addresses of itself
- No change is

necessary for routers



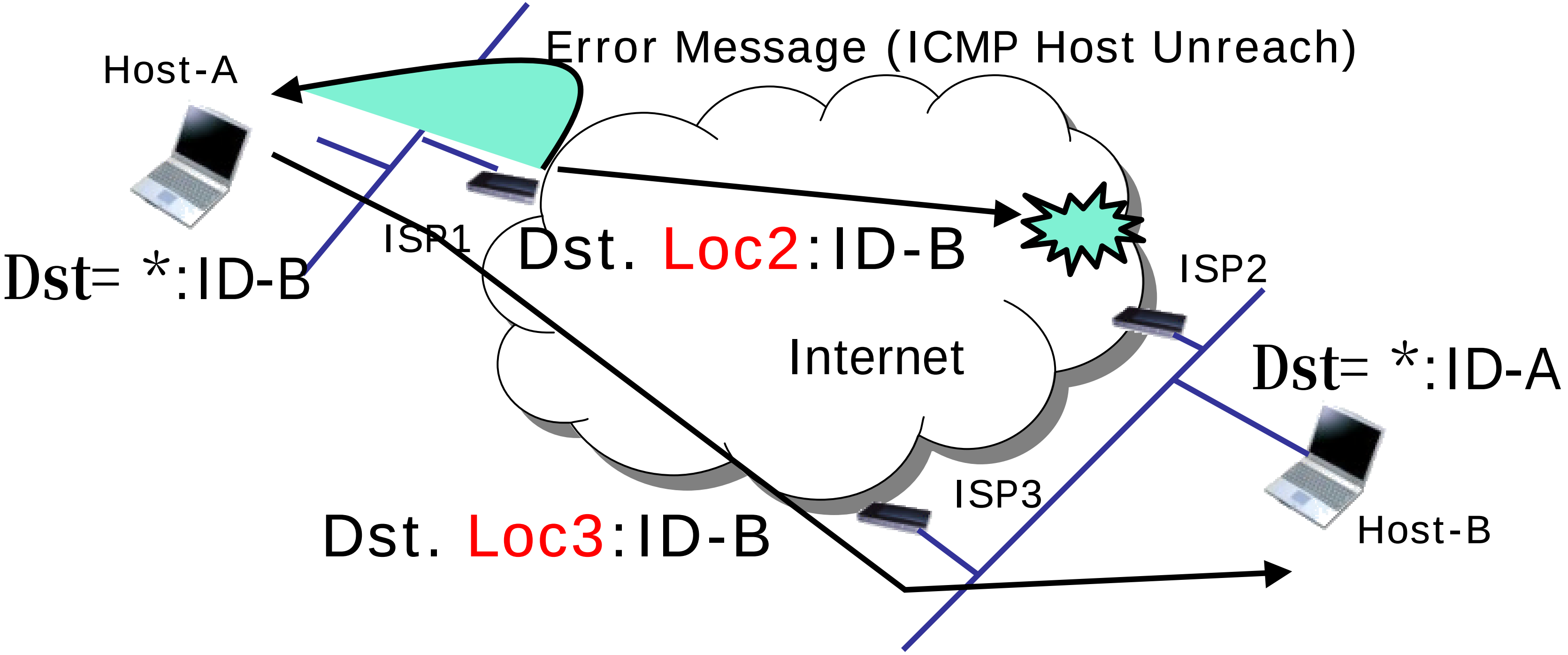
E2E Multihoming with LIN6 model(1/2)

- E2E multihoming has an affinity to LIN6 addressing model, in which “location id” and “node id” is completely separated.



E2E Multihoming with LIN6 model(2/2)

- It’s simple: each connection is identified by Node-Id.
- The mere change of “location part” can avoid connection-down even when a line gets in trouble



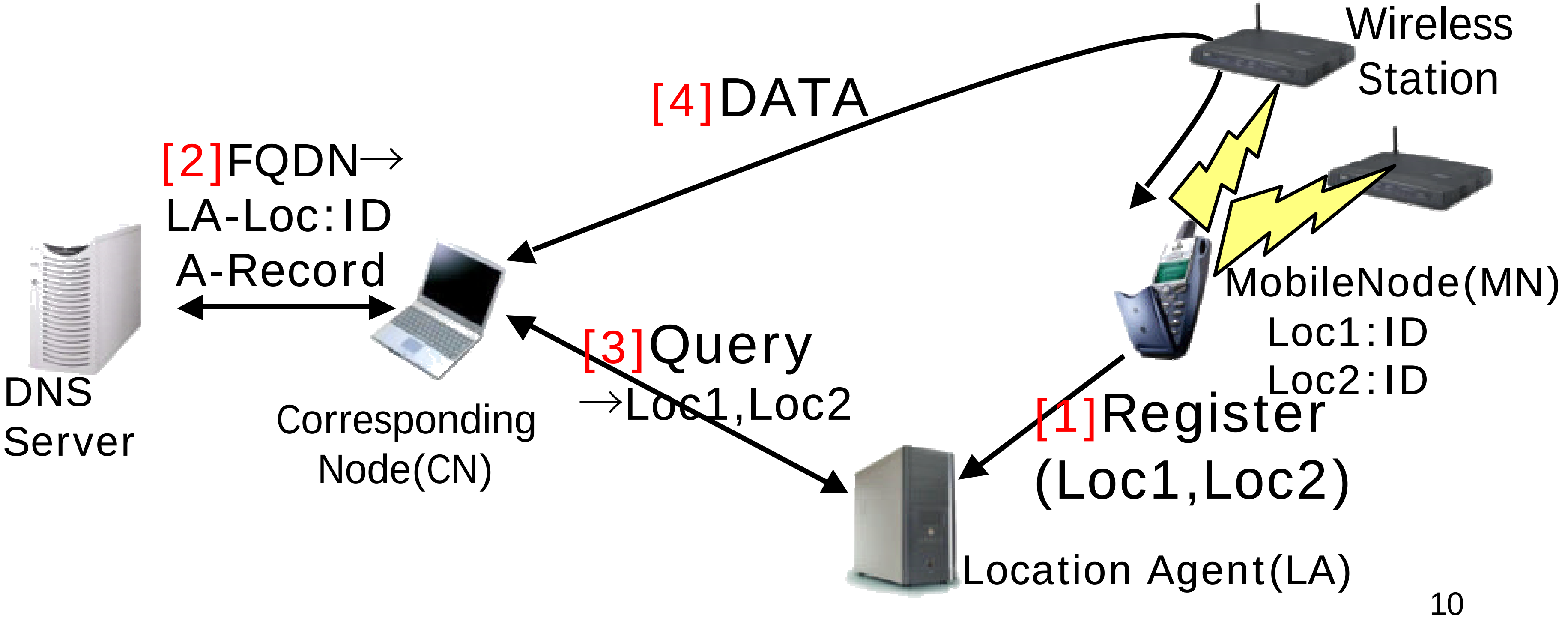
What to do ?

To make LIN6 support E2E multihoming ...

- Make extensions to LIN6 Protocol
  - To handle multiple locators for one node
- Prepare APIs to manipulate locators
  - API to make a socket that identifies each connection by lower 8bytes
  - API to change dst/src locator while connection is established
  - API to get locators of Corresponding Node

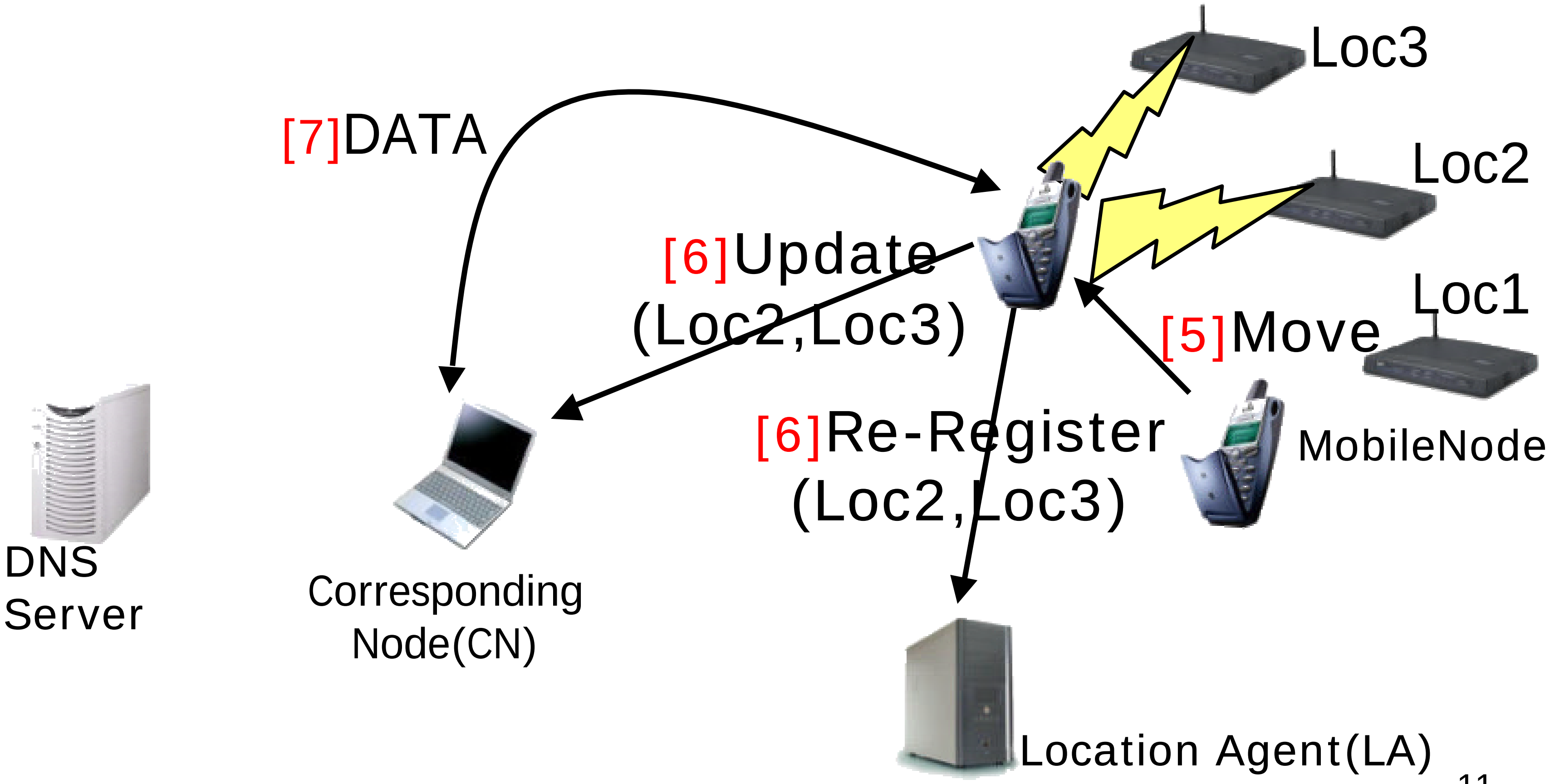
LIN6 Overview and Extension (1/2)

- Multihomed node has more than one locators
- Make extension to LIN6 to register/query/update multiple locators for each node



LIN6 Overview and Extension (2/2)

- Mobile Node moves seamlessly using multiple locators



APIs for Multihoming

- socket(AF_LIN6)
  - Make a socket for LIN6
  - LIN6 socket identifies each connection using only lower 8 bytes
- getaddrinfo2()
  - Gets corresponding node(CN)'s locators
  - Queries Location Agent and gets CN's latest locators
- getsockopt()/setsockopt()
  - Get/Change the dst/src locator of a connected socket
  - When connection error is detected, another locator can be used
- We modified Existing APIs also
  - getaddrinfo() etc…
  - -> Existing Applications works well on a LIN6 node.

Example Program

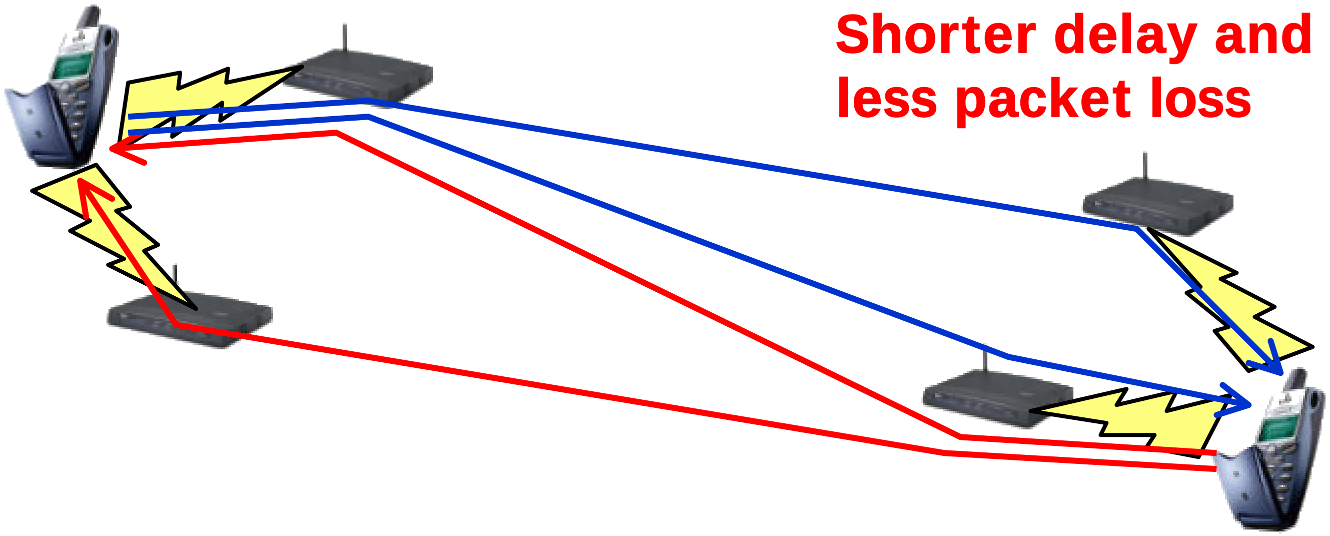
```
struct addrinfo2
{    ai_family
     ai_socktype
     ai_addr ..
     ai_ntloc
     ai_tloc ..    }
```

```
struct addrinfo2 hints, *res, *res2;
getaddrinfo2(“hoge.com”,”http”,&hints,&res);
/* get addresses、make connection */
for(con=0;con!=1&&res->ai_next!=NULL; res=res->ai_next) {
  sock = socket(res->ai_family, res->ai_socktype, res->ai_protocol);
  if (res->ai_family == AF_LIN6) {
    for (con=0,i=0;i<res->ai_ntloc;i++)
      if (connect(sock, res->ai_tloc[i], res->ai_addrlen)==0) con = 1;
  } else { connect(sock, res->ai_addr, res->ai_addrlen)…}
}

void sig_urg(int signo) {  /* error signal handling */
  setsockopt(sock,IPPROTO_IPV6,FOREIGNLOCATOR,
                   res->ai_tloc[++i], sizeof(struct lin6_prefix));
…}
```

Sample Application

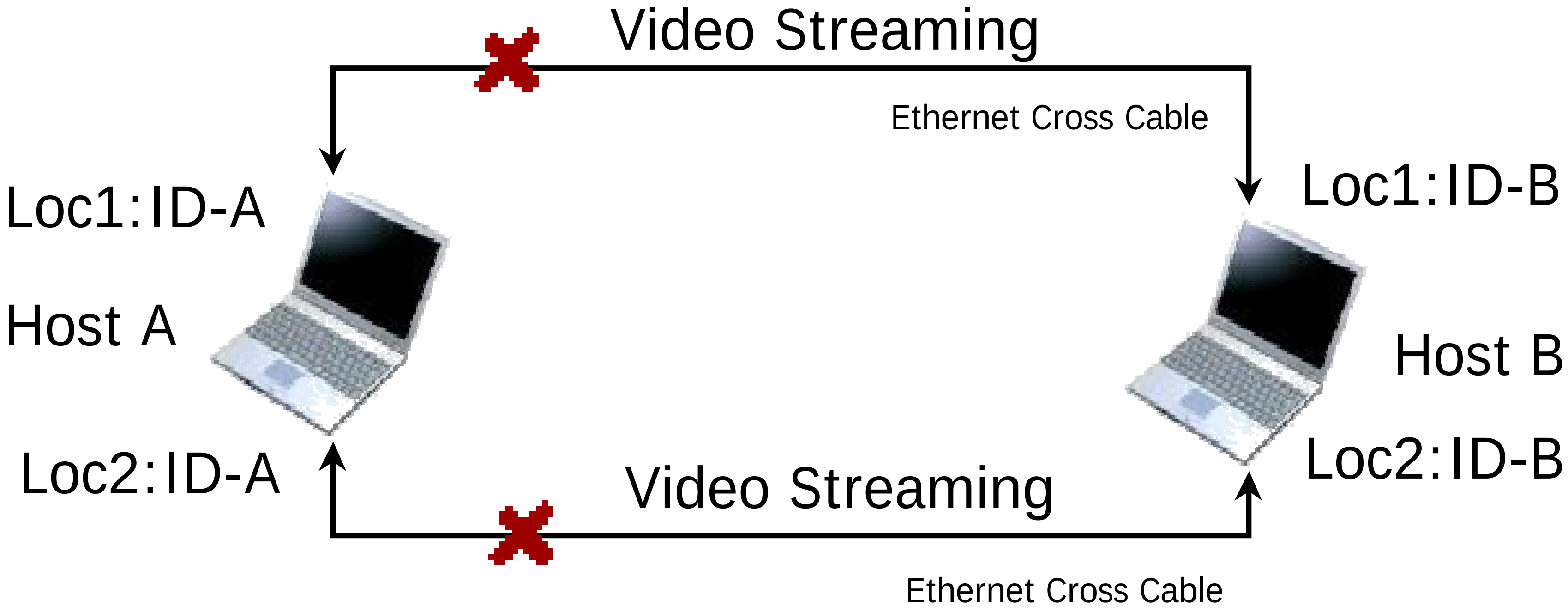
- Multihoming support for NOTASIP telephone application
- Ex. Sends each packet(UDP) to all the locators of corresponding node.



Shorter delay and less packet loss

⇒Easy to make an application multihoming-ready

Demo. of Multihome-ready App.



Conclusion

- We think much of the friendliness of LIN6 model and E2E Multihoming
- We extend LIN6 Protocol and prepare APIs to support E2E multihoming
- As a future work,
  - Commit a field test in Kyoto Pref.
  - Auto-support for end-to-end multihoming in TCP layer