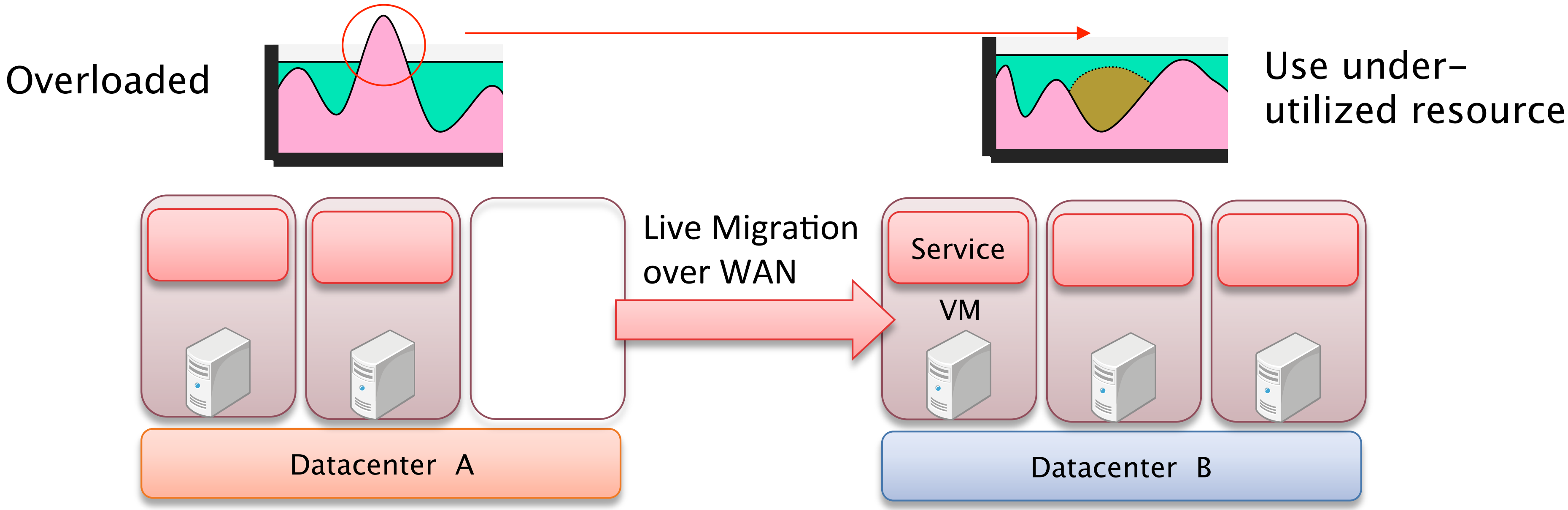
Kagemusha: A Guest-Transparent Mobile IPv6 Mechanism for Wide-Area Live VM Migration

Takahiro Hirofuchi, Hidemoto Nakada,
Itoh Satoshi, and Sekiguchi Satoshi
National Institute of Advanced
Industrial Science and Technology (AIST)

Inter-datacenter load balance

- Optimize VM locations among datacenters
- Save energy and improve performance



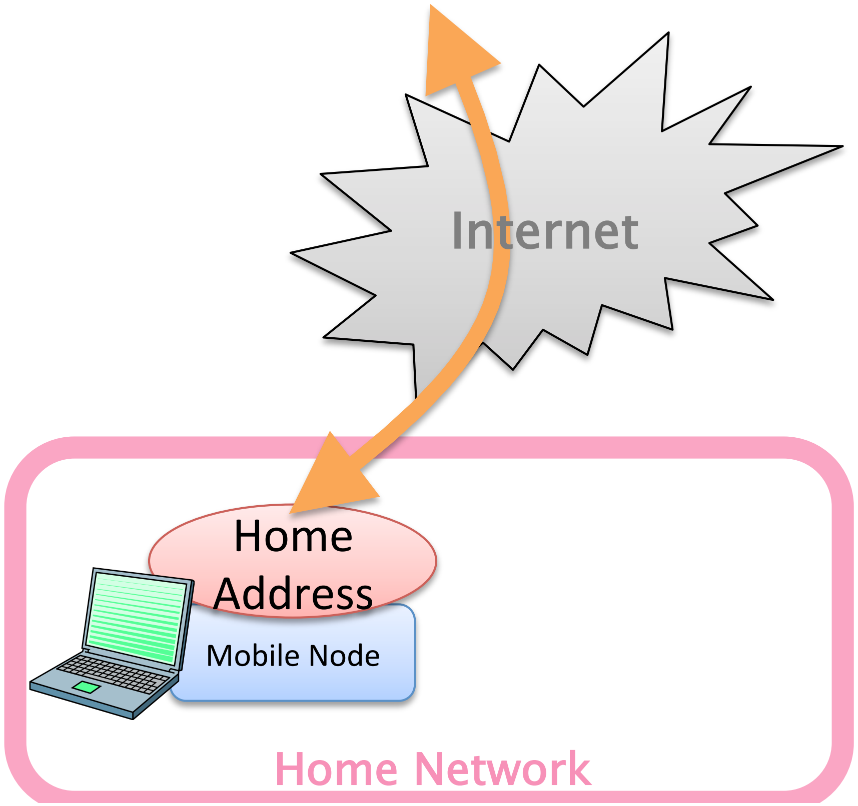
Need feasible live VM migration technology in WAN environments

Our Ongoing Projects for Wide-Area VM Live Migration

- Live VM migration (Yabusame)
  - Postcopy Live Migration for Qemu/KVM [CCGrid'10]
  - Extend Yabusame to be suitable for WAN
    - Add a proactive precache mechanism
- Live storage migration(xNBD)
  - Postcopy storage migration for any VMMs [VTDC'09]
  - Precache important disk blocks in advance
- Network support for wide-area live migration
  - MIPv6-based tunneling mechanism [CloudMan'12]
  - Allow a guest OS to keep using Home Address in any place
  - Transparent to the guest OS of a VM

Overview of Mobile IPv6

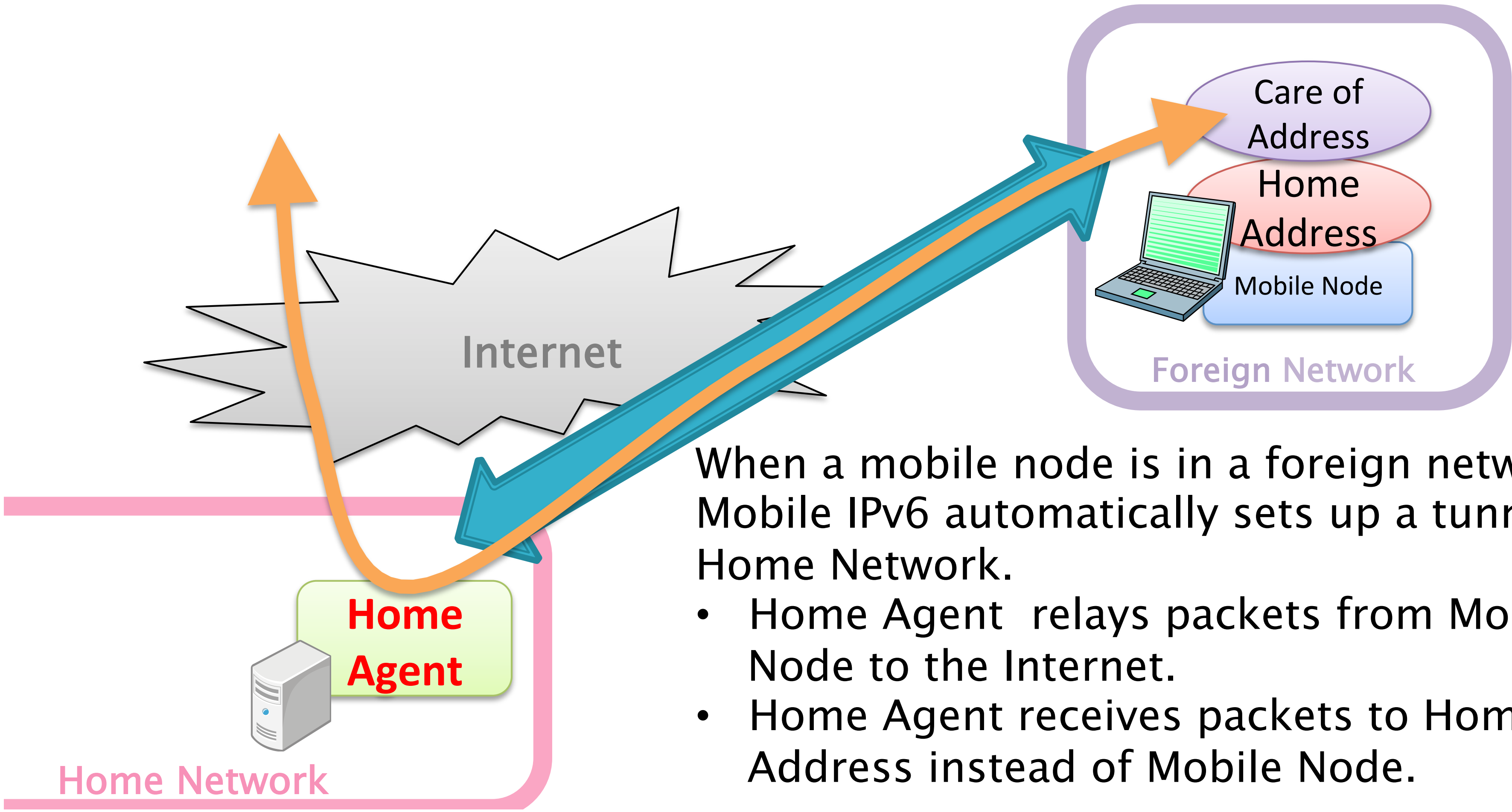
- Keep network reachability with Home Address in any place



In Home Network, No Mobile IPv6 mechanism is necessary. Mobile Node gets Home Address.

Overview of Mobile IPv6

- Keep network reachability with Home Address in any place

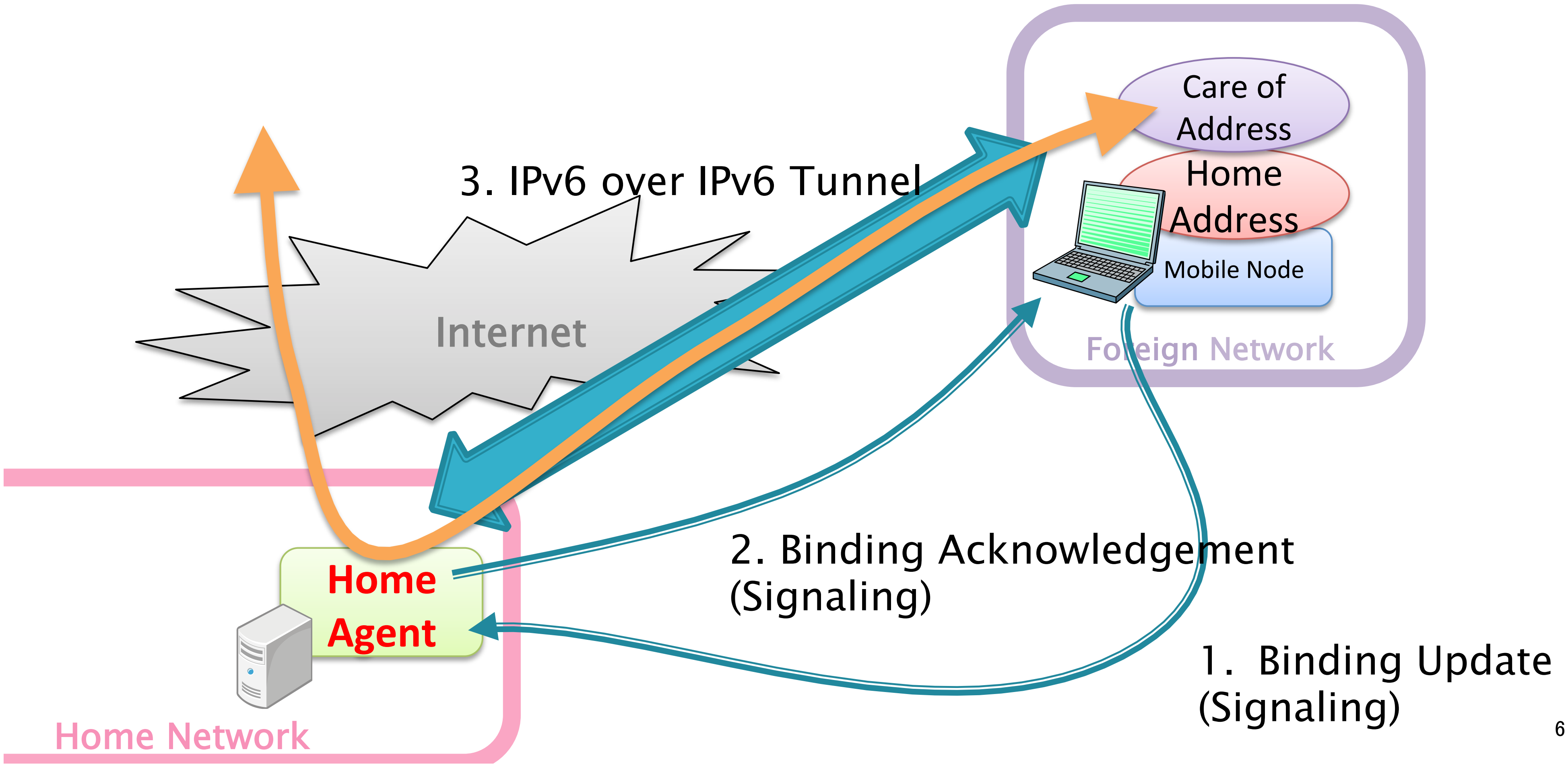


When a mobile node is in a foreign network, Mobile IPv6 automatically sets up a tunnel to Home Network.

- Home Agent relays packets from Mobile Node to the Internet.
- Home Agent receives packets to Home Address instead of Mobile Node.

Overview of Mobile IPv6

- Keep network reachability with Home Address in any place



Mobile IPv6 Advantages

- Open protocol standardized in RFCs
- Strong security mechanisms based on IPSec
- Open source implementations
- Fast hand-over mechanism
- Home Agent redundancy mechanism

- Mobile IP in the wild
  - Wireless LAN in Train, WIMAX

- MIPv6 provides strong mobility support for mobile nodes and is *supposedly* promising.

Mobile IPv6 Families vs VM Migration

- Client MIPv6
  - Mobile Node (Guest OS) handles the MIPv6 protocol.
  - Not transparent to customers of an IaaS cloud
- NEMO (Network Mobility)
  - Mobile Router handles the MIPv6 protocol
  - Transparent to VMs in a sub-network
  - Need to move all VMs in the sub-network at once
- Proxy MIPv6
  - Access network handles the MIPv6 protocol
  - Transplant to VMs
  - All traffic from/to Mobile Node always goes through Home Agent

Requirements

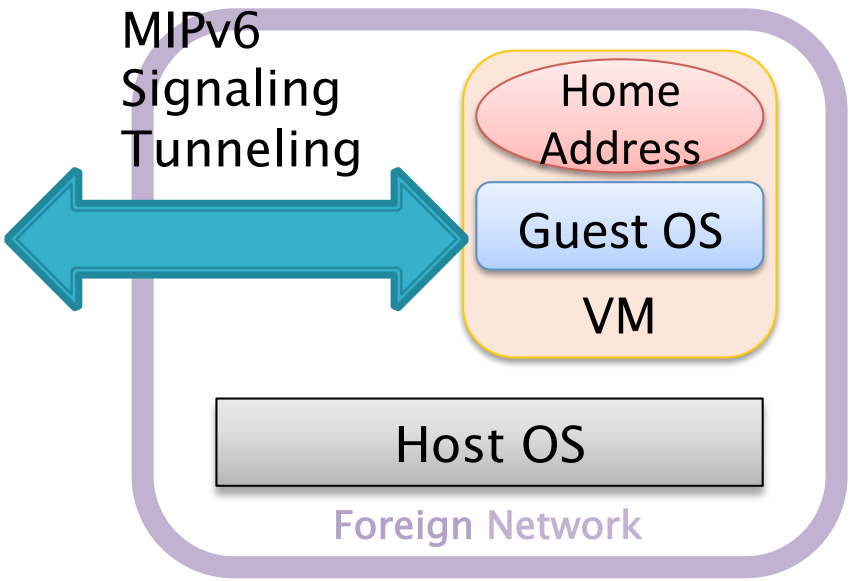
- Allow fine-grained load-balance among datacenters
  - Can migrate VMs in a one-by-one manner
  - Client MIPv6
- Transparent to customers of an IaaS cloud

- Develop a Client MIPv6-based, but guest-transparent mechanism

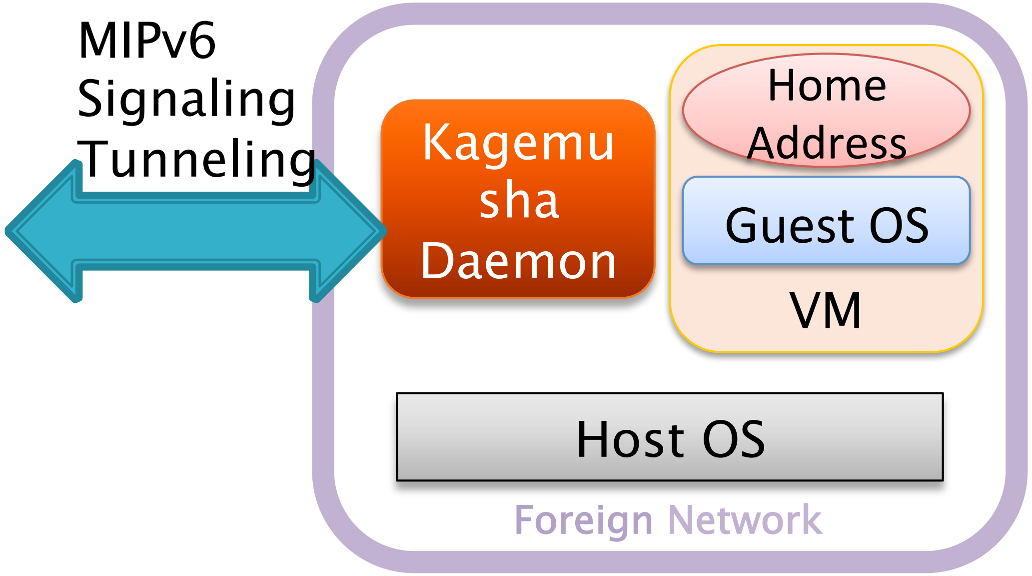
Proposed Mechanism(Kagemusha)

- Handle the signaling and tunneling of Client MIPv6 on a host OS; not in a guest OS
- Allow a guest OS to keep using Home Address

CMIPv6 without Kagemusha

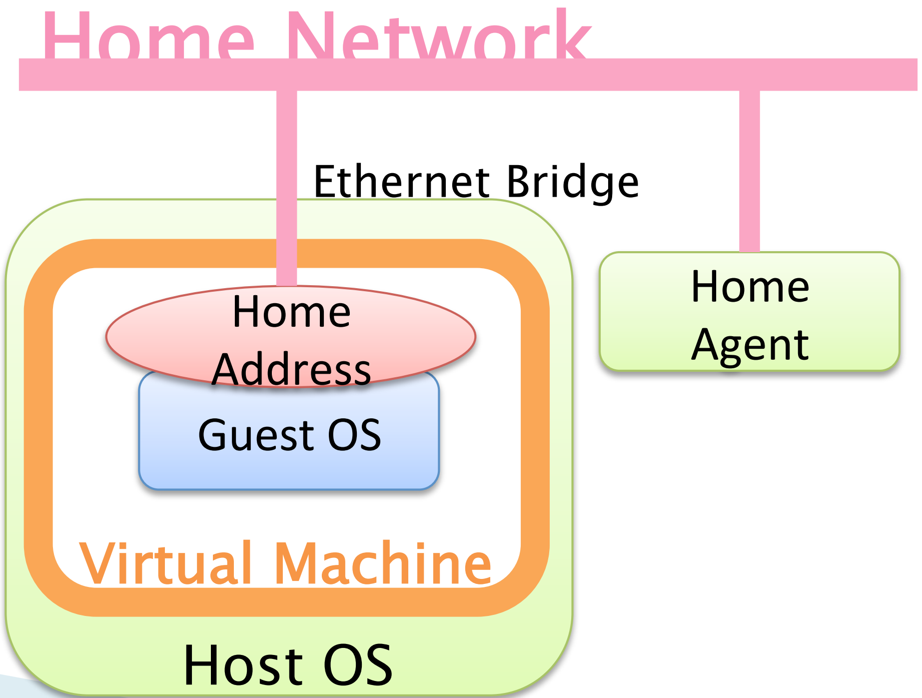


CMIPv6 with Kagemusha



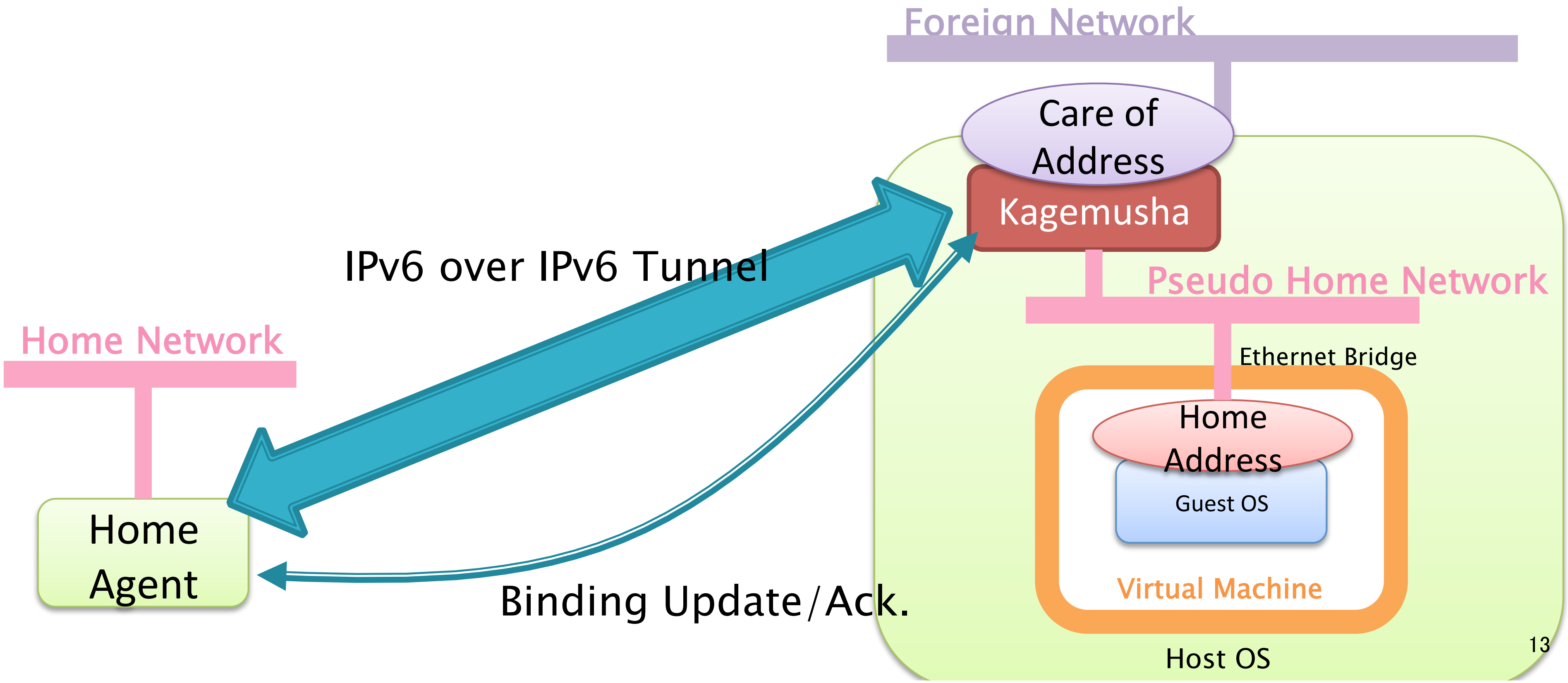
Overview (Home Network)

- Bridge the NIC of a VM to a Home Network
- Assign a Home Address to the guest OS



Overview (Foreign Network)

- Kagemusha handles the MIPv6 protocol
- VM is connected to Pseudo Home Network

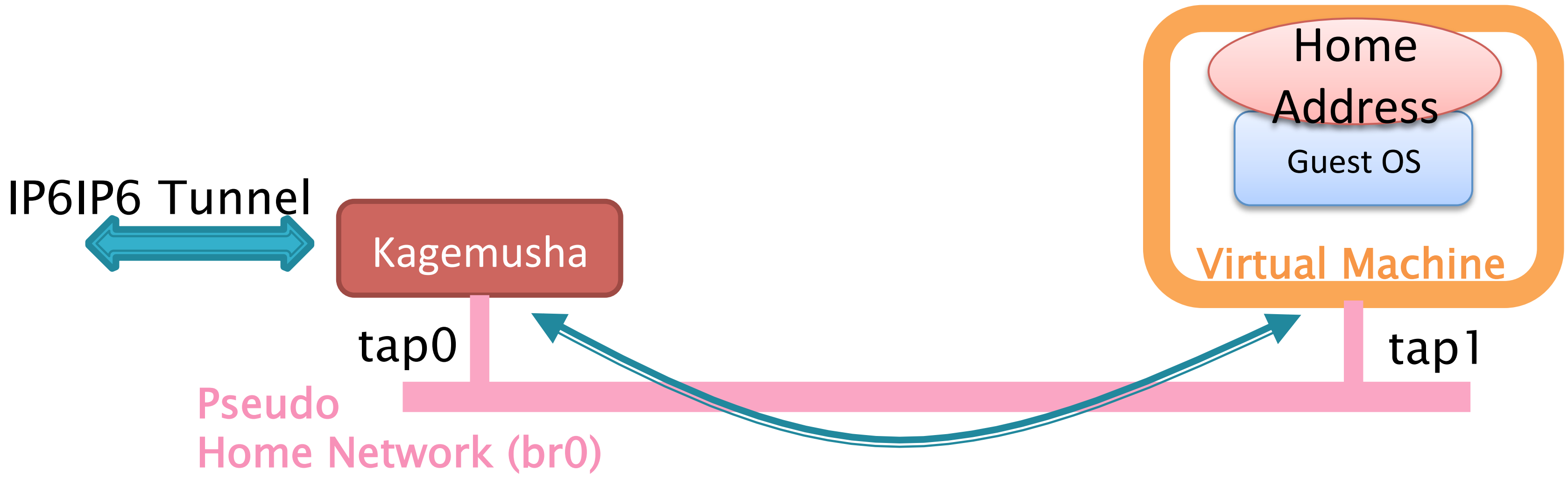


Components

- Kagemusha daemon
  - MIPv6 Signaling
  - MIPv6 Tunneling
  - Pseudo Home Network
- Assumptions
  - Kagemusha daemon knows the following information in advance.
    - Home Address of the VM
    - MAC Address of the VM

Pseudo Home Network Mechanism(1)

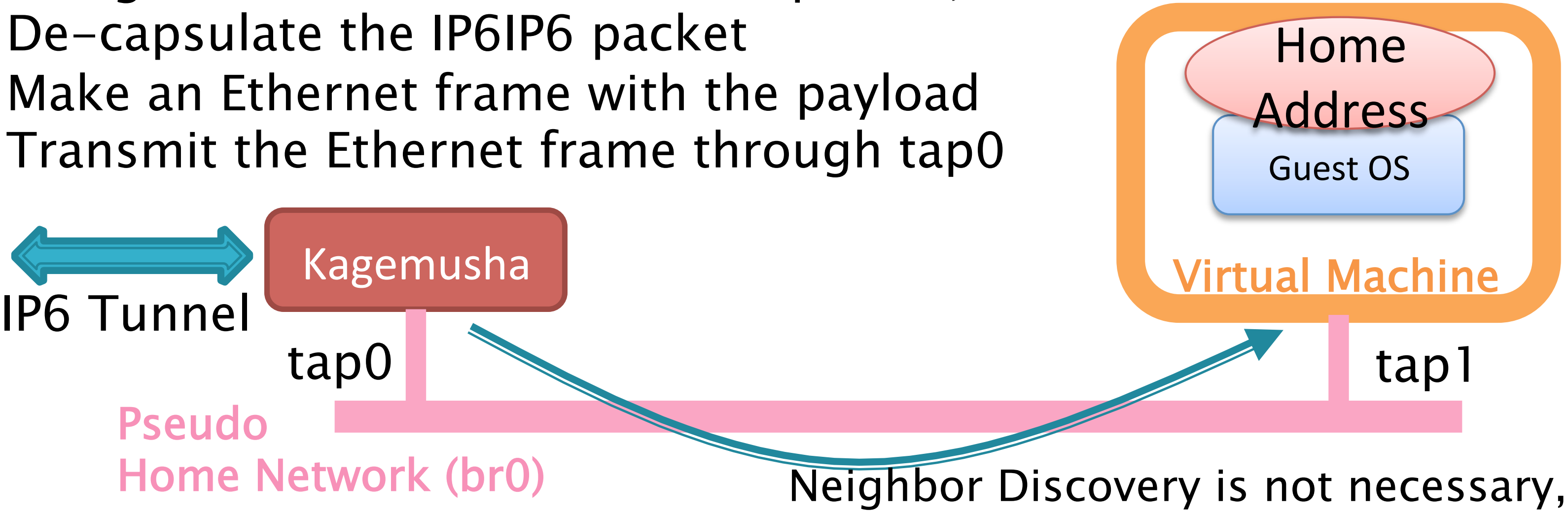
- Redirect packets between the VM and the IP6IP6 tunnel in a manner transparent to a guest OS
- Kagemusha acts like the other nodes in a home network
  - Receive packets from the VM as if Kagemusha is the default router or another host in the home network
  - Transmit packets to the VM as if Kagemusha is the default router or another host in the home network



Pseudo Home Network Mechanism(2)

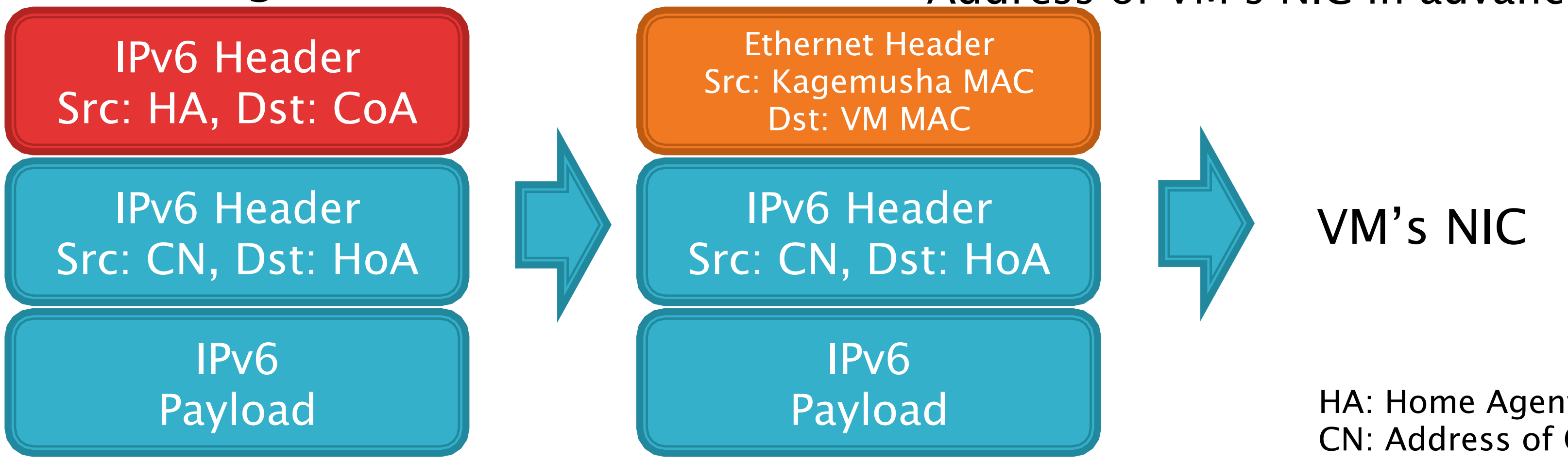
When Kagemusha received an IP6IP6 packet,

1. De-capsulate the IP6IP6 packet
2. Make an Ethernet frame with the payload
3. Transmit the Ethernet frame through tap0



Neighbor Discovery is not necessary, because Kagemusha knows Home Address and MAC Address of VM's NIC in advance.

Tunneling Packet

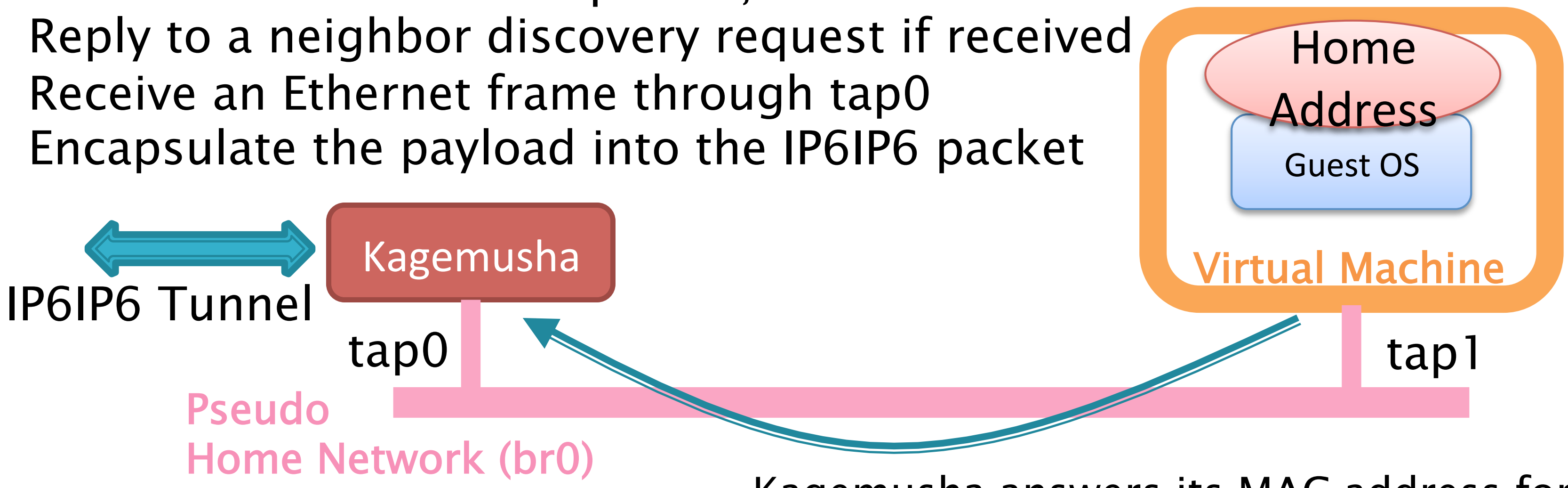


HA: Home Agent's Address
CN: Address of Correspondent Node
HoA: Home Address

Pseudo Home Network Mechanism(3)

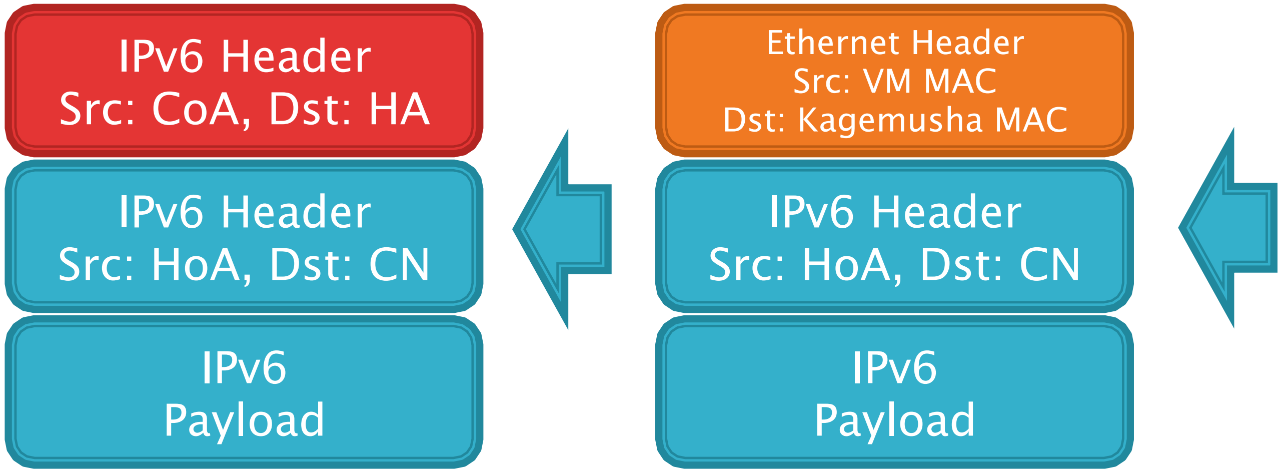
When VM transmits an IPv6 packet,

1. Reply to a neighbor discovery request if received
2. Receive an Ethernet frame through tap0
3. Encapsulate the payload into the IP6IP6 packet



Kagemusha answers its MAC address for neighbor discovery requests to get packets from the VM.

Tunneling Packet



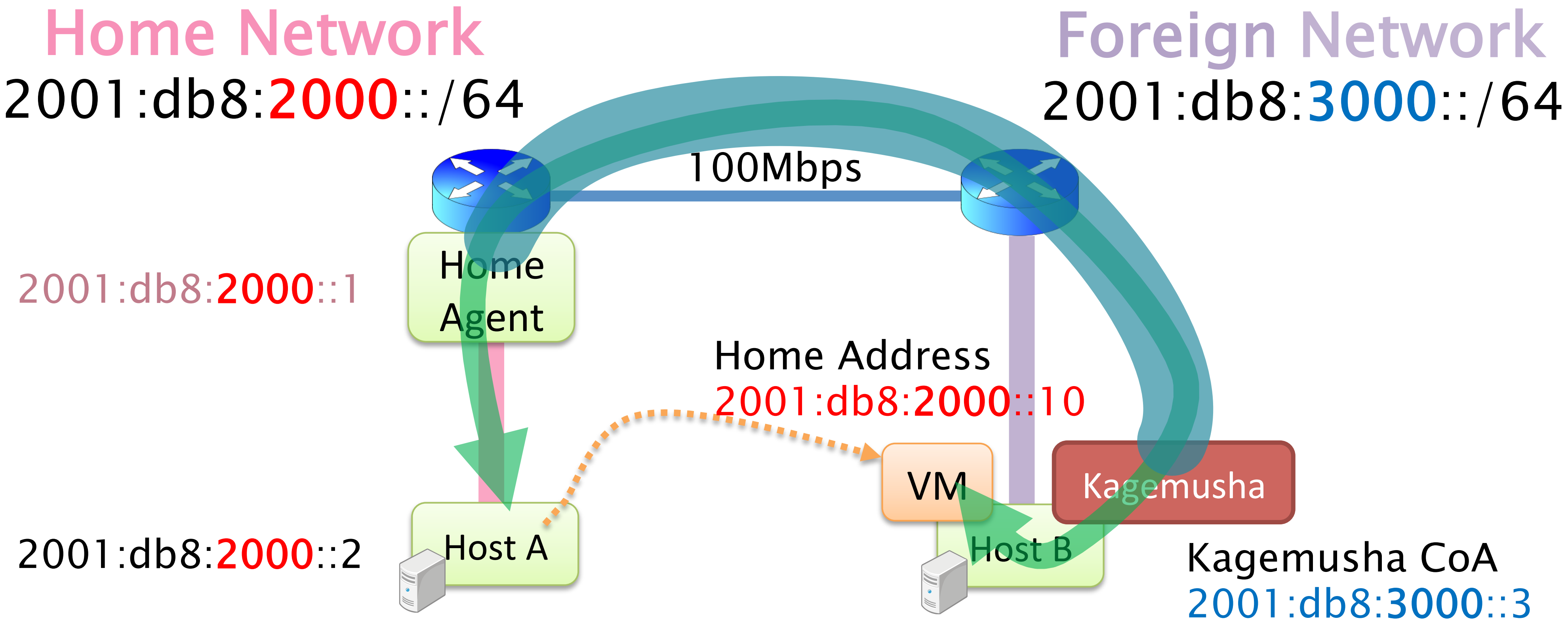
VM's NIC

HA: Home Agent's Address
CN: Address of Correspondent Node
HoA: Home Address

Prototype Implementation

- Kagemusha daemon
  - 1500LOC userland code
  - 1 daemon for 1 VM
- Portability for Unix-like OSes
  - Use raw socket of IPPROTO_{MH, IPV6}
  - Use the ancillary data of sendmsg/recvmsg() for the IPv6 extension headers
- Fix some Linux bugs (?)
  - Type2 routing header
  - IPSec (XFRM)

Experimental Network



- Home Agent: umip (git 74528e)
- VMM: qemu-kvm-{0.12.5, 0.14.1}
- {Host, Guest} OS: Linux Kernel 2.6.32

Ping6 (IPSec Disabled)

- Ping6 (from the guest OS on Host B to Host A)

Packet capture on Host B

1. IP6 2001: db8 :3000::3 > 2001: db8 :2000::1:
   DSTOPT mobility : BU seq #=13 AH lifetime =60

Binding Update

2. IP6 2001: db8 :2000::1 > 2001: db8 :3000::3:
   srcrt (len=2, type=2, segleft=1, [0]2001: db8 :2000::10)
   mobility : BA status =0 seq #=13 lifetime =60

Binding Ack.

3. IP6 2001: db8 :3000::3 > 2001: db8 :2000::1:
   IP6 2001: db8 :2000::10 > 2001: db8:2000::2:
   ICMP6 , echo request , seq 78, length 64

IP6IP6 Tunnel
(Echo Request)

4. IP6 2001: db8 :2000::1 > 2001: db8 :3000::3:
   IP6 2001: db8 :2000::2 > 2001: db8:2000::10:
   ICMP6 , echo reply , seq 78, length 64

IP6IP6 Tunnel
(Echo Reply

Ping6 (IPSec Enabled)

- Ping6 (from the guest OS on Host B to Host A)
- Protect MIPv6 signaling messages with IPSec ESP

Packet capture on Host B

1. IP6 2001: db8 :3000::3 > 2001: db8 :2000::1:
   DSTOPT ESP (spi=0x000003e8, seq=0xd), length 68

   Binding Update

2. IP6 2001: db8 :2000::1 > 2001: db8 :3000::3:
   srcrt (len=2, type=2, segleft=1, [0]2001: db8 :2000::10)
   ESP (spi=0x000003e9, seq=0x1b), length 52

   Binding Ack.

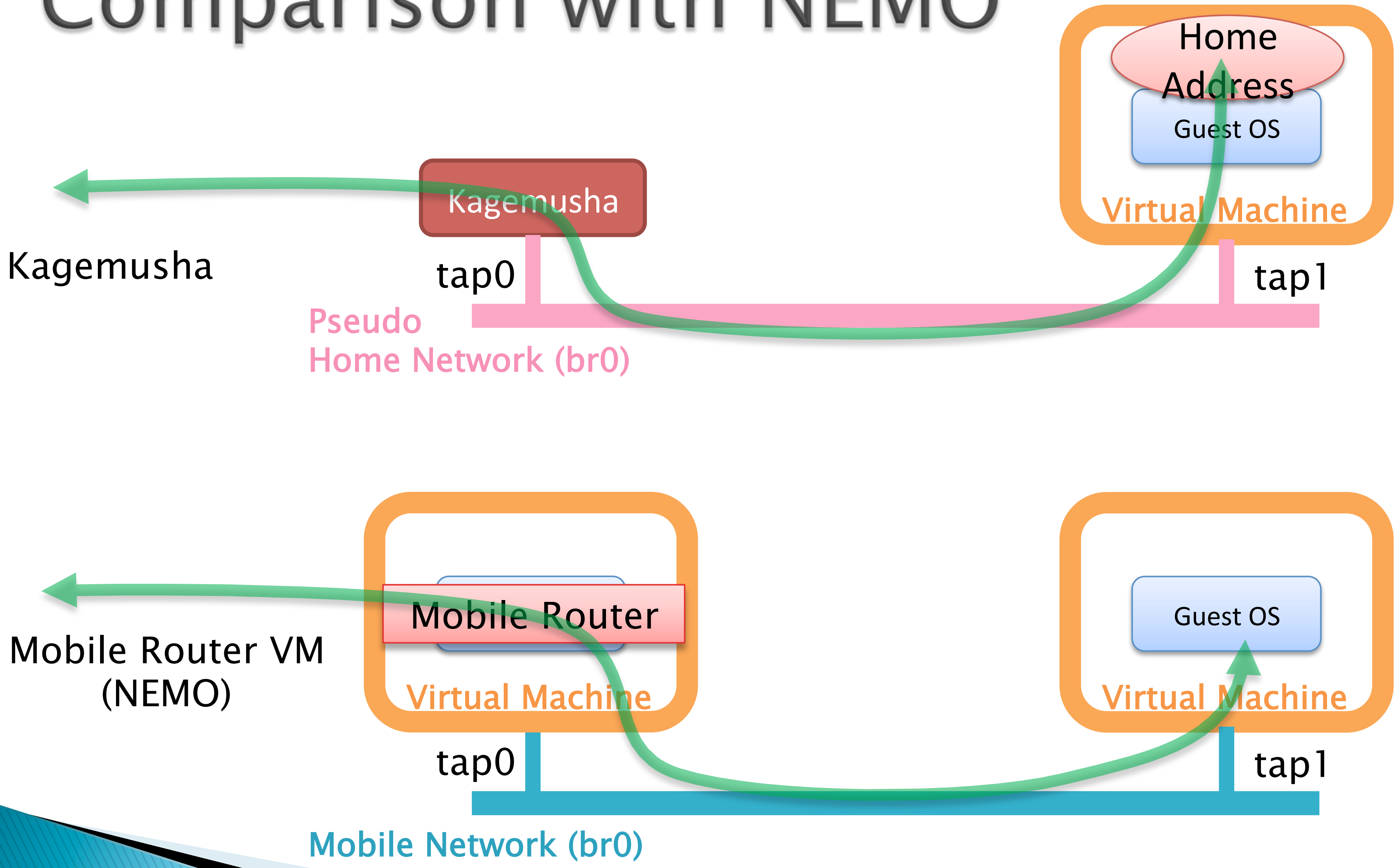
3. IP6 2001: db8 :3000::3 > 2001: db8 :2000::1:
   IP6 2001: db8 :2000::10 > 2001: db8:2000::2:
   ICMP6 , echo request , seq 394 , length 64

   IP6IP6 Tunnel
   (Echo Request)

4. IP6 2001: db8 :2000::1 > 2001: db8 :3000::3:
   IP6 2001: db8 :2000::2 > 2001: db8:2000::10:
   ICMP6 , echo reply , seq 394 , length 64

   IP6IP6 Tunnel
   (Echo Reply

Comparison with NEMO



Performance (Kagemusha vs. NEMO)

Throughput (iperf)

	Throughput	CPU Usage
Kagemusha	84.4Mbps	14%
MR VM(NEMO)	87.8Mbps	92%

Network Latency (ping6)

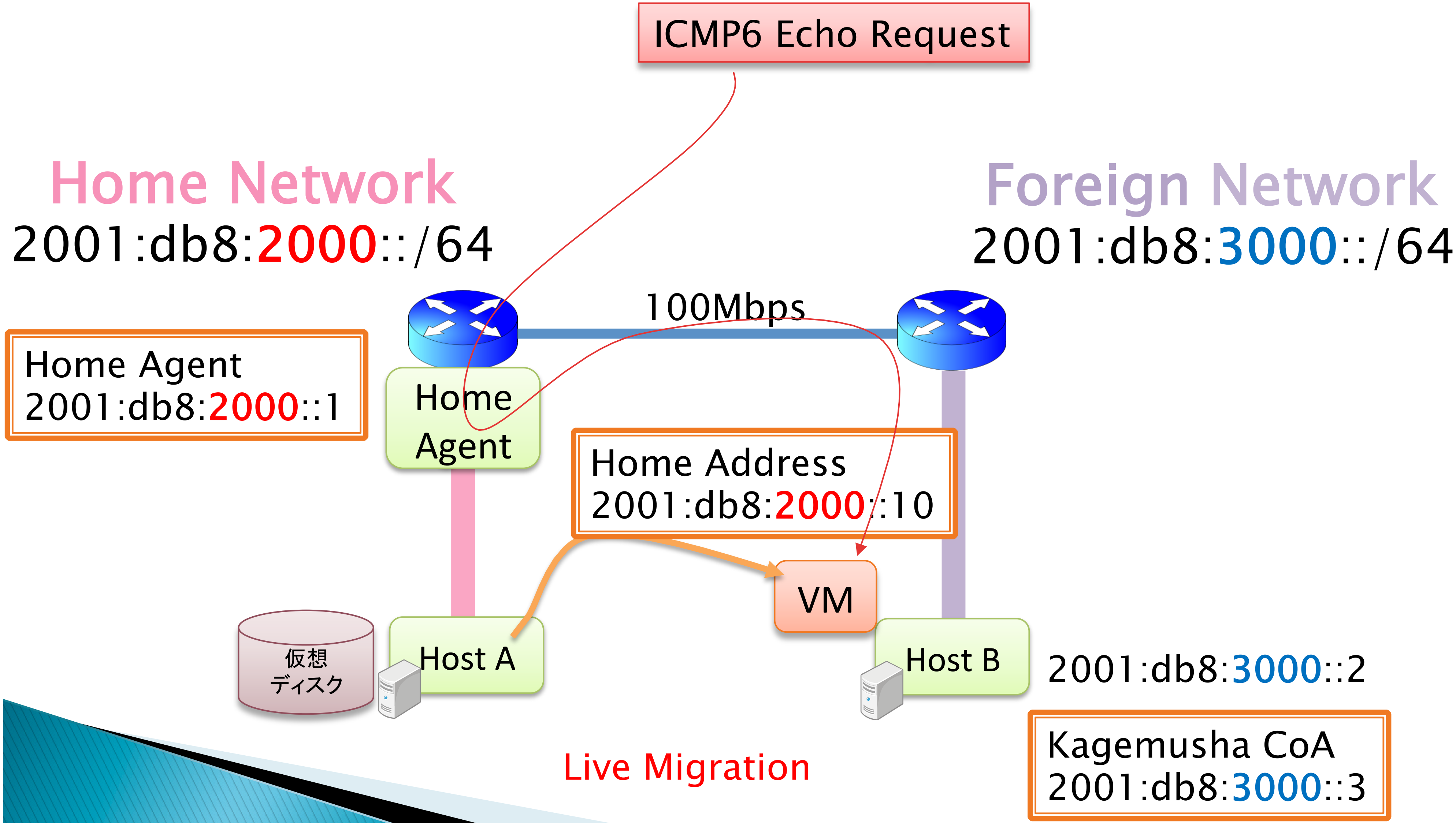
	RTT	RTT Jitter
Kagemusha	0.44ms	0.028ms
MR VM(NEMO)	0.77ms	0.046ms

Memory Usage

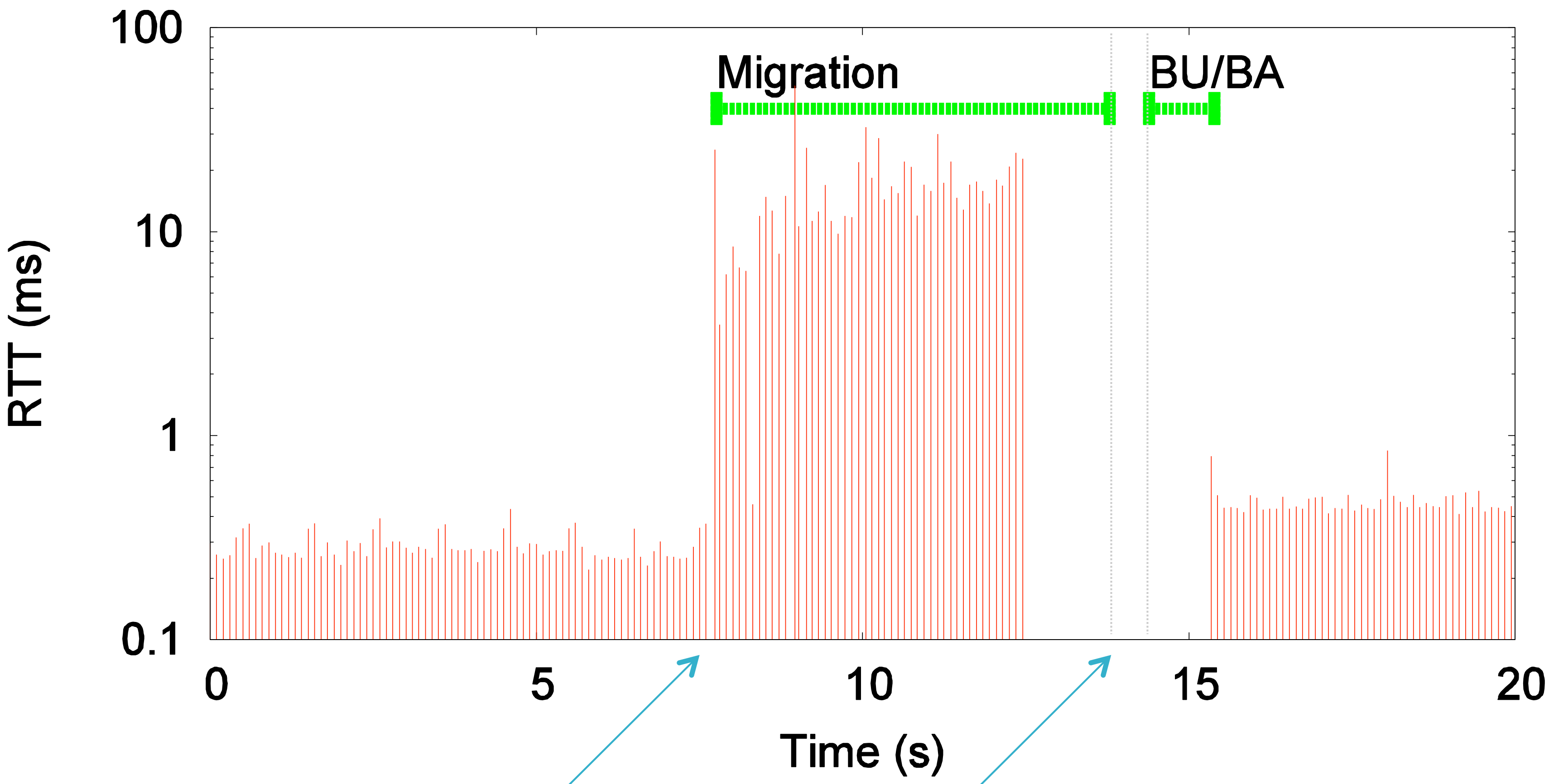
	Memory Usage
Kagemusha	1.5MB
MR VM(NEMO)	Dozens MB

The memory usage of Kagemusha is much smaller than MR VM.

Live VM Migration



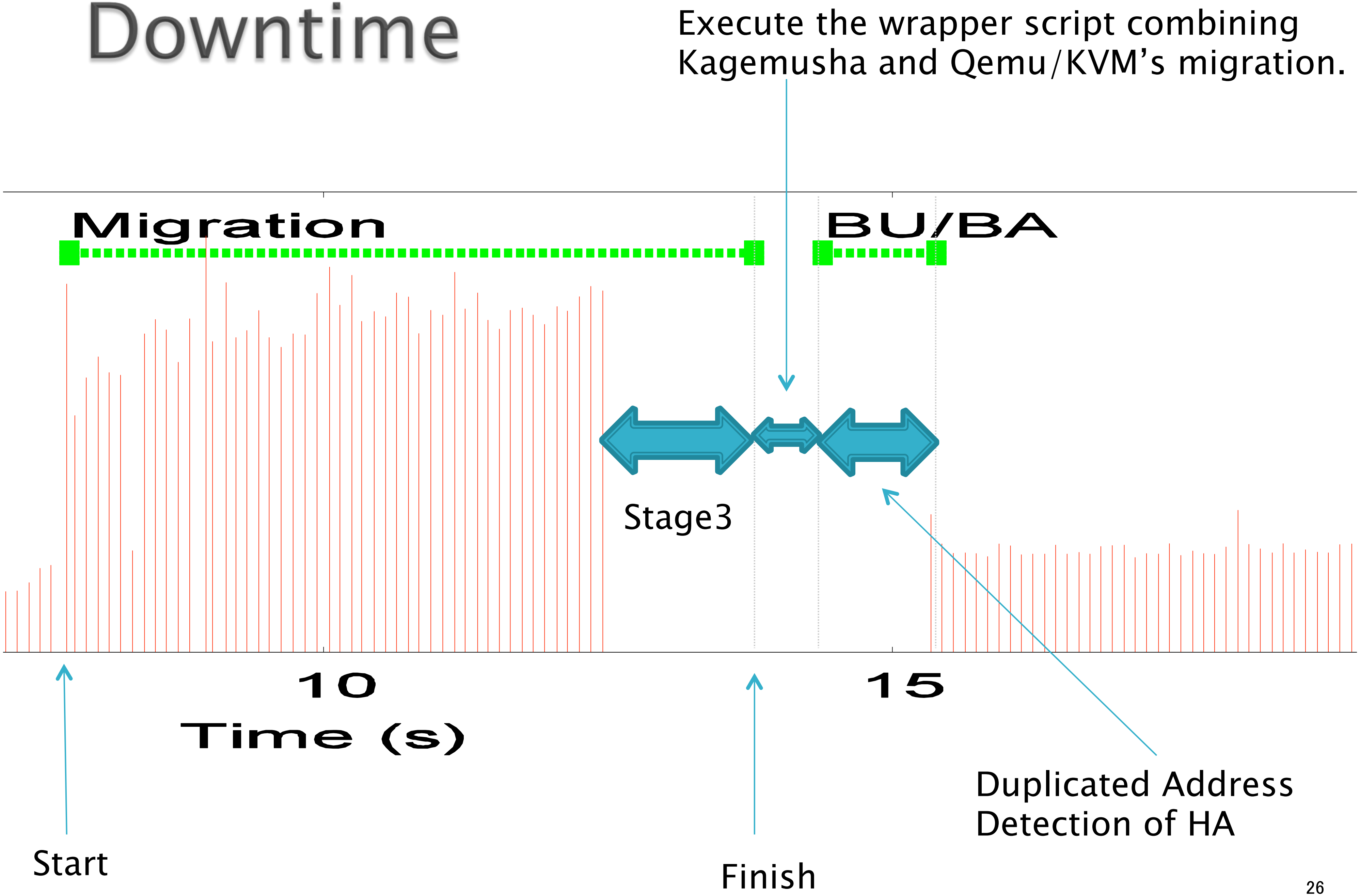
RTT by ping6



A live migration starts

Finish

Downtime


Execute the wrapper script combining
Kagemusha and Qemu/KVM's migration.
Migration
BU/BA
Stage3
10
15
Time (s)
Duplicated Address
Detection of HA
Start
Finish

Related Work

- Client MIPv6 support on guest OS [Harney 2007]
  - No MIPv6 program outside VM
  - Not transparent to guest OS
- NEMO（MR VM [Ishibashi 2010], MR host OS [Shima 2009]）
  - Transparent to guest OS
  - Suitable for live migration of a cluster of VMs
- HyperMIP [Li 2008]
  - Proxy MIPv4 support at Xen Hypervisor
  - Transparent to guest OS
  - All network traffic goes though the HA.

Conclusion

- Kagemusha
  - Client MIPv6 mechanism transparent to guest OS
  - Perform MIPv6 tunneling and signaling instead of guest OS
  - No MIPv6 support inside guest OS
  - Fully-compatible with HAs and VMMs
- Prototype Implantation
  - Worked correctly with live migration
- Future Work
  - Support IPv4 (Dual-Stack MIPv6)
  - Support NEMO