
IPv6 Specific Issues to Track States of Network Flows

Yasuyuki Kozakai (TOSHIBA Corporation)
Hideaki Yoshijuji (Keio University)
Hiroshi Esaki (The University of Tokyo),
Jun Murai (Keio University)

USAGI / WIDE Project

Table of Contents

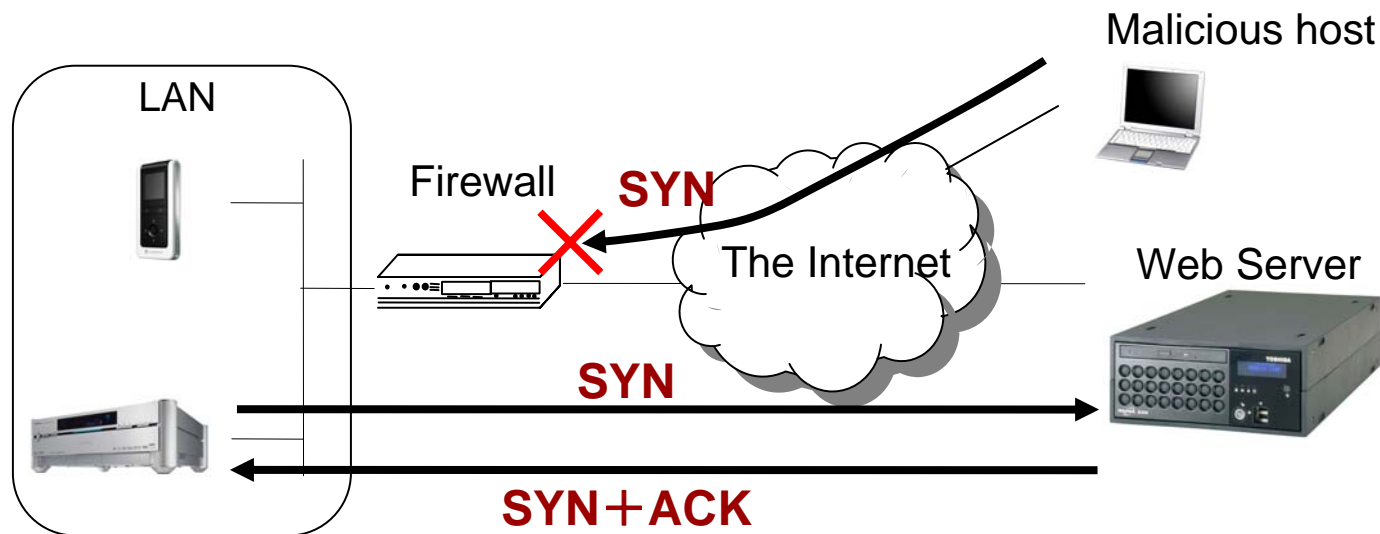
- **Background**
 - Stateful filtering and IPv6
 - Connection Tracking subsystem on Linux
- **Motivation**
- **Objectives**
- **Issues and our proposals**
 - A Changed path with Type 0 Routing Header
 - Address spoofing with headers used in Mobile IPv6
- **Conclusion**

Background – stateful filtering and IPv6

Stateful filtering

- Today it is widely deployed in IPv4 networks
- It can discard unsolicited incoming packets from the outer network
 - To protect LAN from straightforward attacks
 - To prevent hosts on LAN from consuming resources

We believe same demands also exist on IPv6 networks



Background – Connection tracking subsystem on Linux

- **We introduced support for IPv6 stateful filtering in Linux**

It consists of 2 parts

- Layer 3 protocol independent connection tracking subsystem (nf_conntrack)
 - It tracks states of network flows (called 'connection')
 - It supports TCP, UDP, ICMP, FTP
- Packet filter to make filter decision by state

Motivation

- **We solved some IPv6 specific issues on nf_conntrack**
 - Generalization of processing independence on address family
 - Special handling for fragmented packets
 - **But more improvements are required**
 - The number of IPv6 supported devices is increasing
 - We expect that many devices will support IPv6 related extensions in next step
- ⇒ **nf_conntrack is required to support them**

Objectives

Improvements of nf_conntrack to solve following issues

- **A changed path with Type 0 Routing Header(RT0)**
 - A path of communication might be changed by using RT0
 - If the path is changed and turned back, stateful filter drops packets
- **Address spoofing with headers used in Mobile IPv6**
 - A packet might include a spoofed home address in
 - Home Address Option in Destination Options Header, or
 - Type 2 Routing Header
 - If state filter does not check their headers, it would allow the packet to pass through the firewall

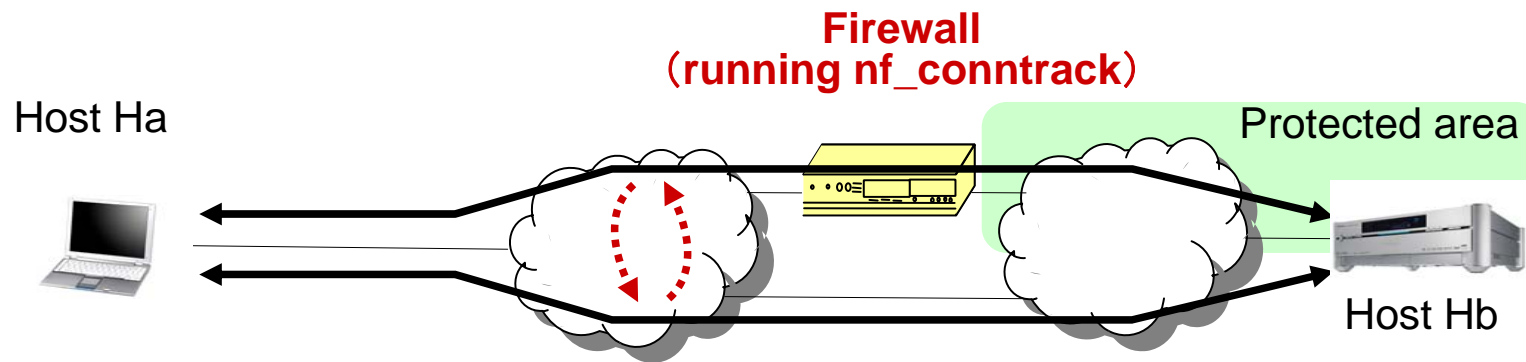
Objectives

Improvements of nf_conntrack to solve following issues

- **A changed path with Type 0 Routing Header(RT0)**
 - A path of communication might be changed by using RT0
 - If the path is changed and turned back, stateful filter drops packets
- **Address spoofing with headers used in Mobile IPv6**
 - A packet might include a spoofed home address in
 - Home Address Option in Destination Options Header, or
 - Type 2 Routing Header
 - If state filter does not check their headers, it would allow the packet to pass through the firewall

A Changed path with Type 0 Routing Header(RT0)

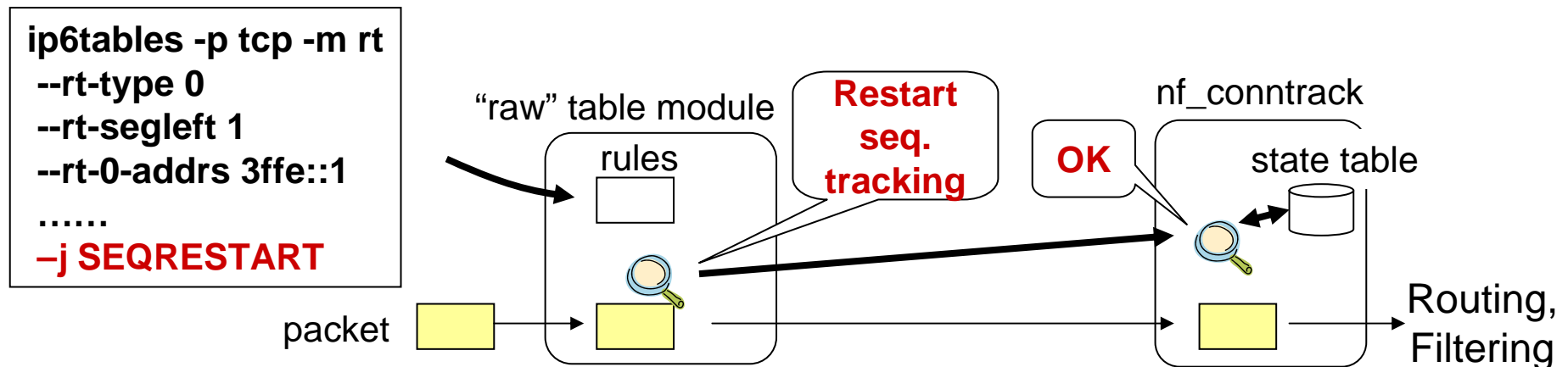
- **A path of connection might be changed by RT0**
e.g. A path of TCP connection was changed once and turned back



- **Issue in such situation**
 - nf_conntrack detects the skipped TCP sequence numbers in packets
 - nf_conntrack handles them as invalid
 - ⇒ Stateful filter drops all packets after the path turned back

Our proposal

- **nf_conntrack restarts tracking TCP sequence numbers**
 - User specifies the condition to restart tracking
 - e.g. - The address in RT0 is trusted by user
 - The user limits frequency of restarting
- **How to implement it on Linux**
 - We can utilize the table of rules evaluated before nf_conntrack
 - We need to implement a rule module to signal nf_conntrack
 - ⇒ User can make various rules by combining it and traditional rules



Objectives

Improvements of nf_conntrack to solve following issues

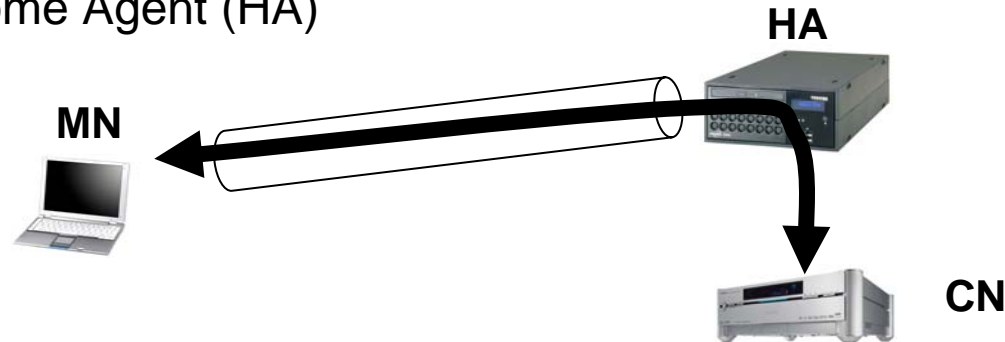
- **A changed path with Type 0 Routing Header(RT0)**
 - A path of communication might be changed by using RT0
 - If the path is changed and turned back, stateful filter drops packets
- **Address spoofing with headers used in Mobile IPv6**
 - A packet might include a spoofed home address in
 - Home Address Option in Destination Options Header, or
 - Type 2 Routing Header
 - If stateful filter does not check their headers, it would allow the packet to pass through the firewall

Mobile IPv6

There are 2 modes

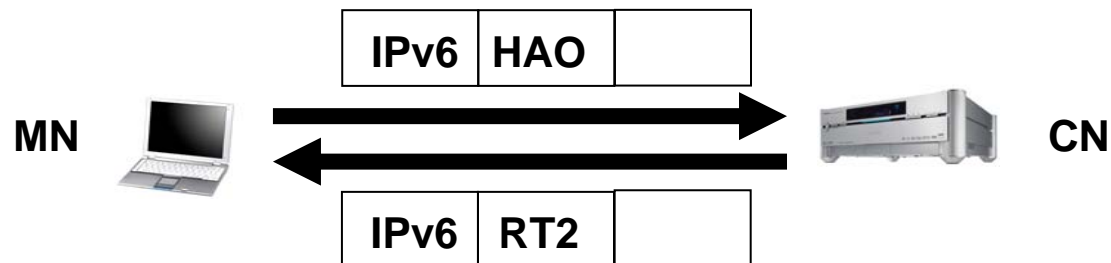
- Bidirectional tunneling

All traffic between Mobile Node (MN) and Corresponding Node (CN) passes through Home Agent (HA)



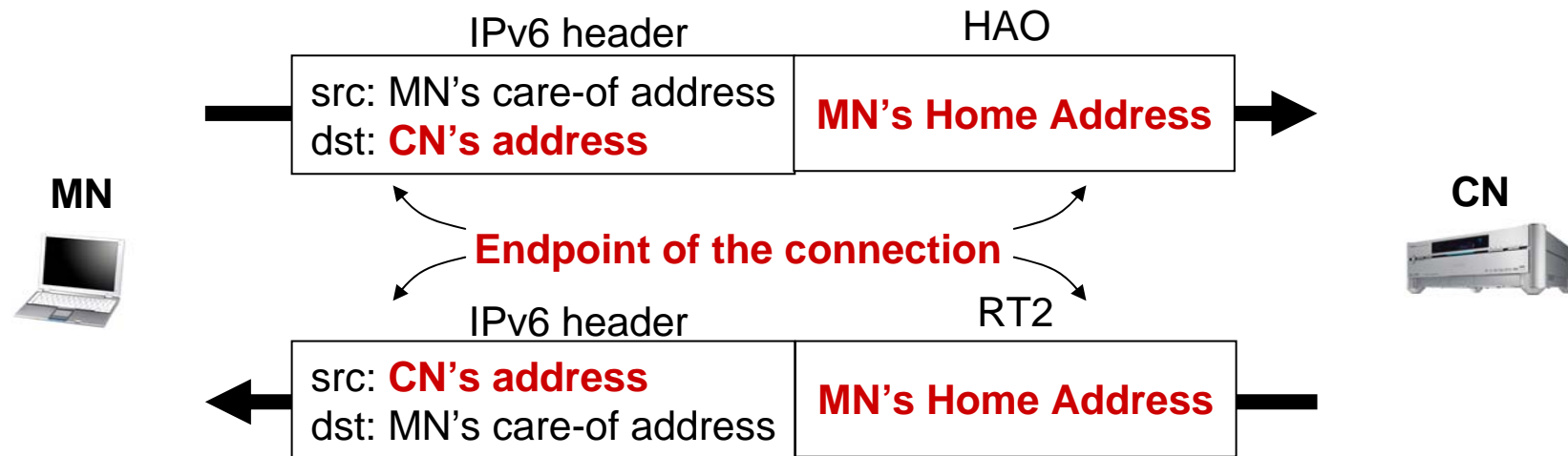
- Route optimization

- After some MIPv6 signaling, MN can directly communicate with CN by
 - Type 2 Routing Header (RT2) and
 - Home Address Option (HAO) in Destination Options Header



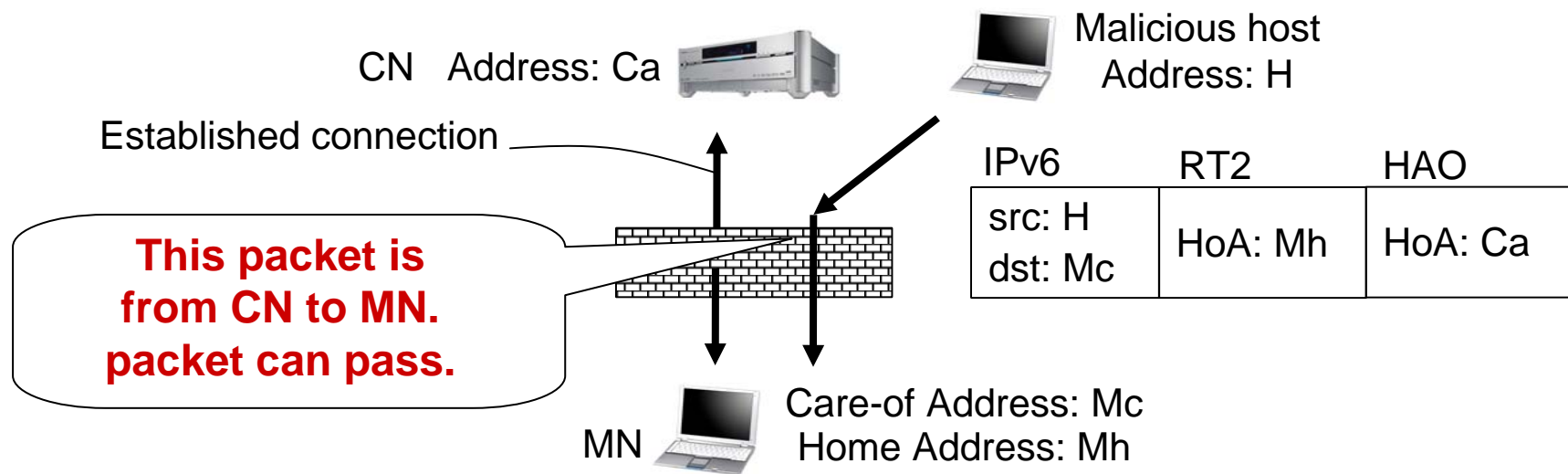
Home address of Mobile Node in HAO and RT2

- **HAO and RT2 include the home address of Mobile Node**
 - nf_contrack needs to handle it as the endpoint of connection



Address spoofing with headers used in Mobile IPv6

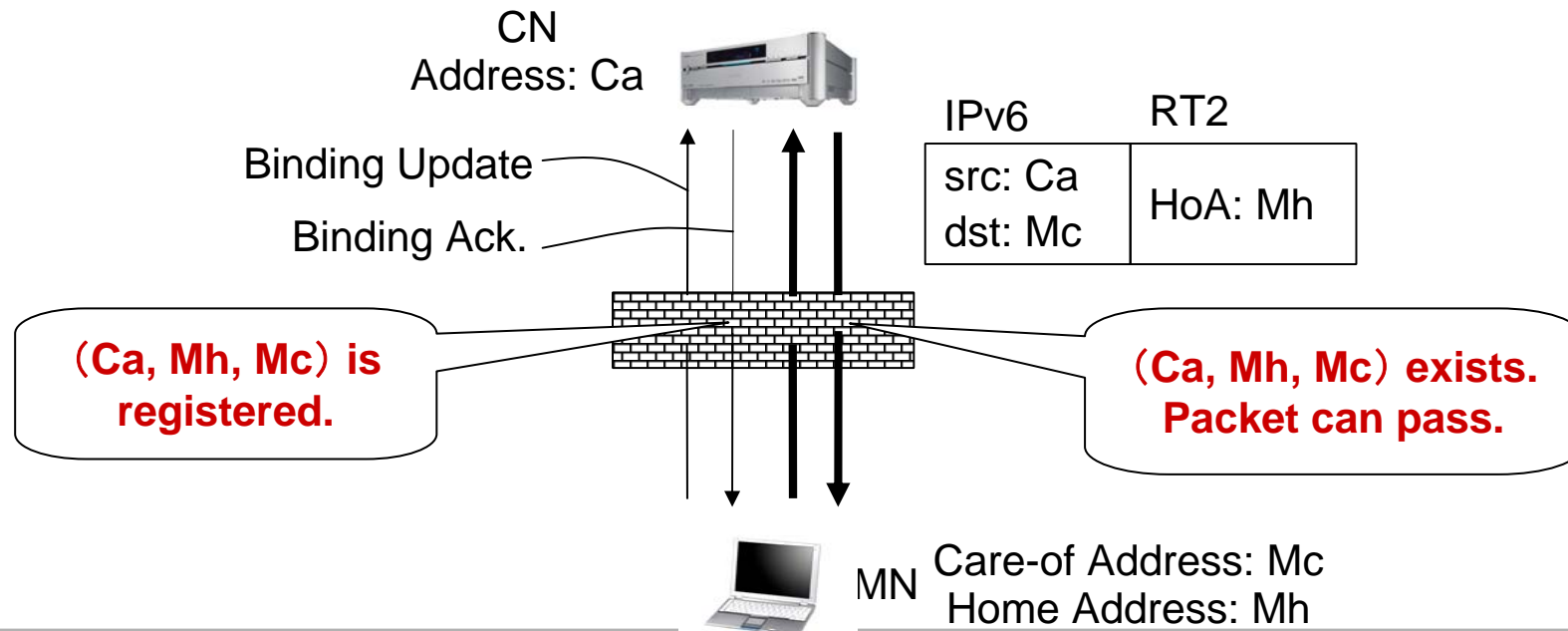
- It is easy for malicious host to send packets including spoofed home address
- ⇒ **If stateful filter does not check HAO and RT2, it would allow them pass through the firewall**



- **What can stateful filter do to such packets ?**

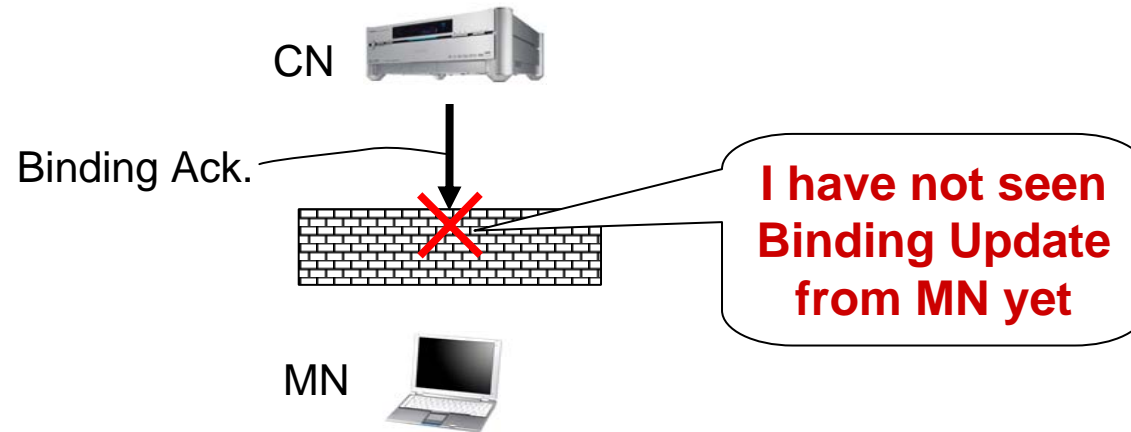
Our proposal

- nf_contrack inspects MIPv6 signaling
 - Binding Update to CN and Binding Acknowledgement from CN
 - It gets CN's address, MN's care-of address, MN's home address
- nf_contrack checks the addresses in RT2 and HAO
- Stateful filter blocks packets including unknown address in RT0 and/or HAO

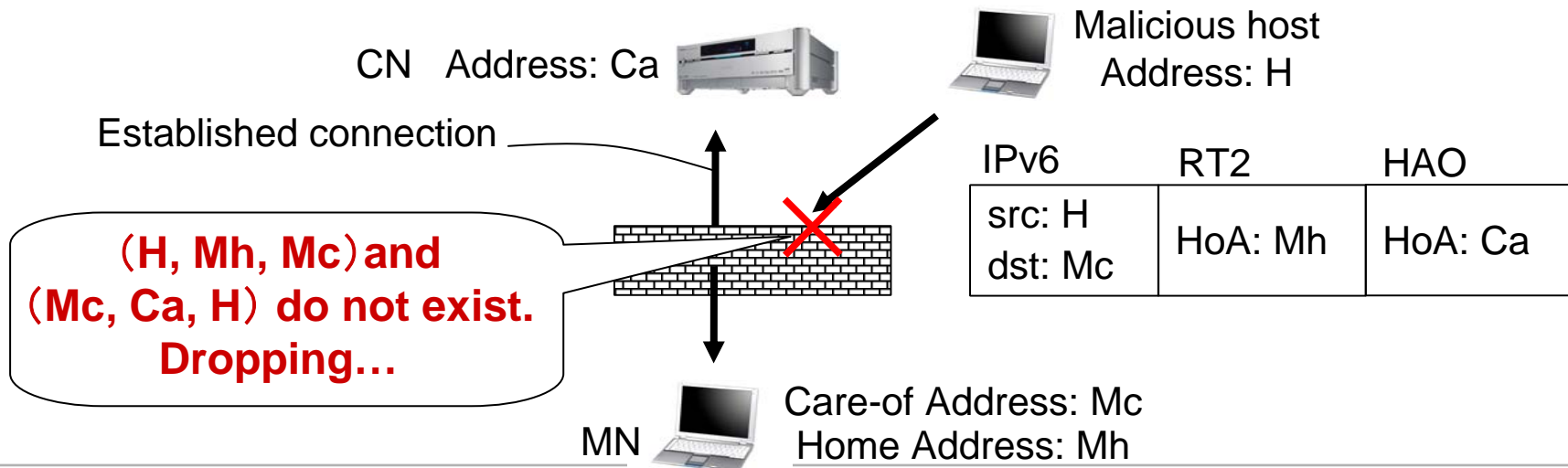


What stateful filter can block by our proposal(1/2)

- Unsolicited Binding Acknowledgement

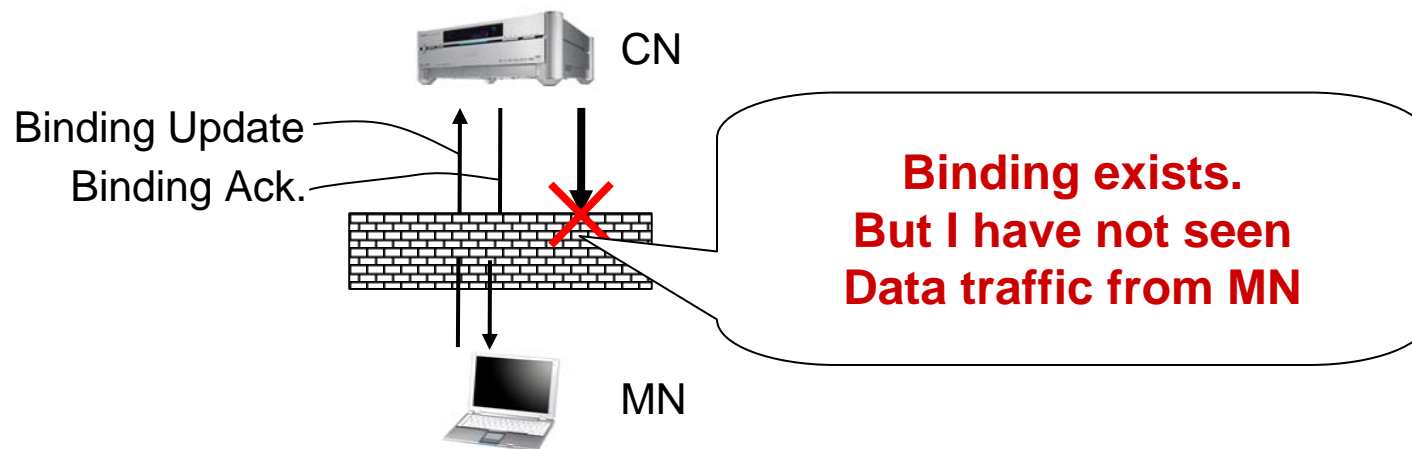


- Spoofed home address in HAO or RT2



What stateful filter can block by our proposal(2/2)

- Unsolicited data traffic



Optional behaviors

- Binding Acknowledgement can be omitted
 - ⇒ **An option :Stateful filter does not require to detect Binding Acknowledgement**
- CN can directly send packets to MN without MIPv6 signaling
 - ⇒ **An option :Stateful filter does not check CN's address**
 - It checks the home address and Care-of address of MN retrieved from Binding Update sent to other CN

Restrictions

- CN under a firewall running stateful filter
 - ⇒ MN cannot initiate MIPv6 signaling to CN
 - ⇒ pinhole on firewall is necessary
 - This is common restriction of stateful filter, not depending on MIPv6
- Encrypted Binding Update/Acknowledgement
 - draft-ietf-mip6-cn-ipsec-05
 - ⇒ There is no way of parsing them

Conclusion

We proposed 2 improvements to connection tracking

- So that `nf_conntrack` can restart tracking TCP sequence number
⇒ Stateful filter can allow packets pass firewall even if a path of connection is changed and turned back.

- So that `nf_conntrack` can check addresses in RT2 and/or HAO
⇒ stateful filter can block spoofed packets with RT2 and/or HAO

Future Work

- Implementation and evaluations

Fin.