MIPv6 CN-Targeted Location Privacy and Optimized Routing

draft-weniger-mobopts-mip6-cnlocpriv-02

Kilian Weniger

IETF #69, Chicago, July 2007

Outline

- Scope of this draft
- Scenario and problem definition
- Proposed solution
- Assumptions and applicability
- Changes in new draft version
- Conclusion

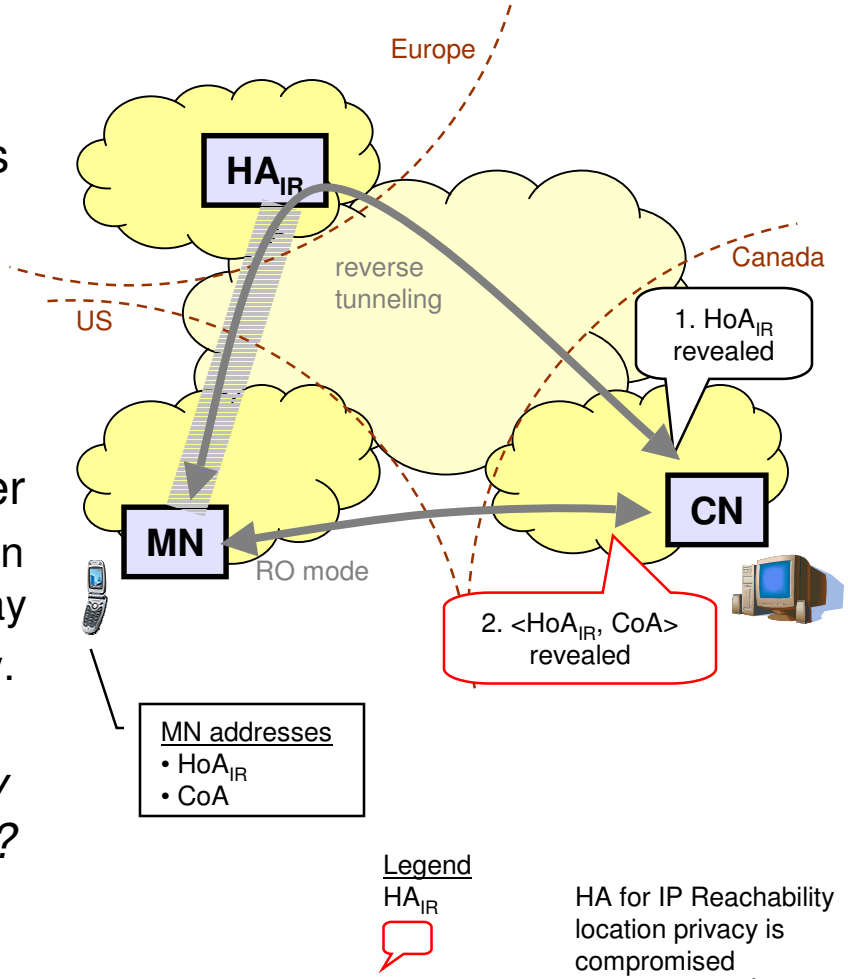
Scope of this draft

- *“CN-targeted location privacy”* = Preventing disclosure of the MN‘s topological location to a CN
  - see Mobile IPv6 location privacy problems [RFC4882]
- Problem of disclosing location to eavesdroppers is out of scope

Scenario and problem definition

- MN is reachable at public HoA_{IR}
  - associated with MN‘s public identity
- MN-CN communication session requires short packet delays (e.g., Skype)
- MN wants to hide its location from CN
  - i.e., <HoA_{IR}, location> must be hidden

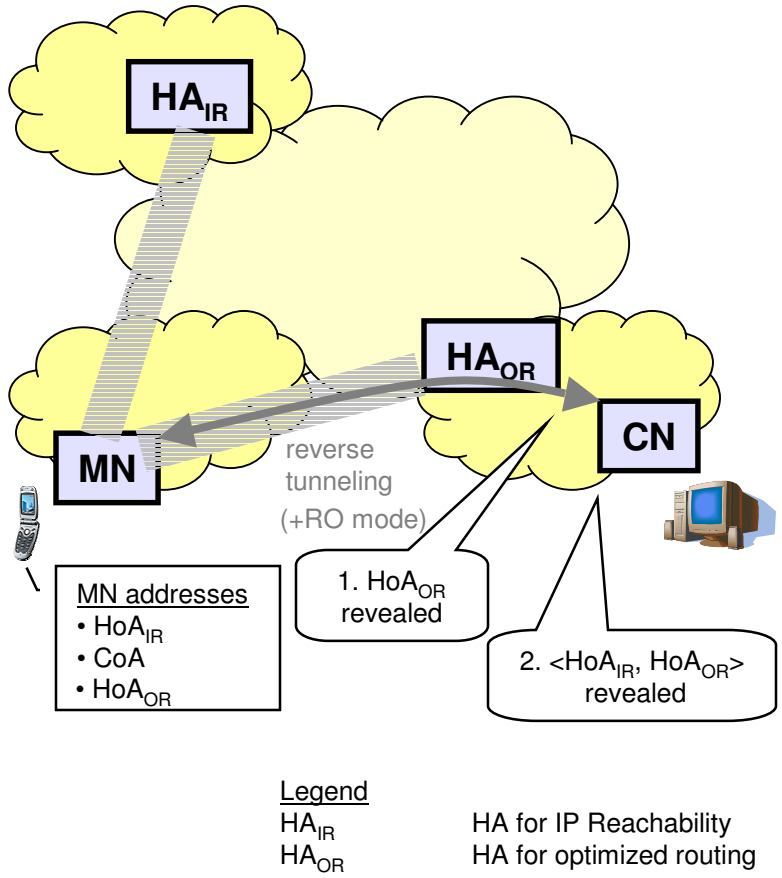
- If MN is far away from home, it can either
  1. Use reverse tunneling to hide its location from CN. But this increases packet delay
  2. Use RO mode to get short packet delay. But this reveal the location to CN
- *But how to achieve both location privacy and short packet delays simultaneously?*



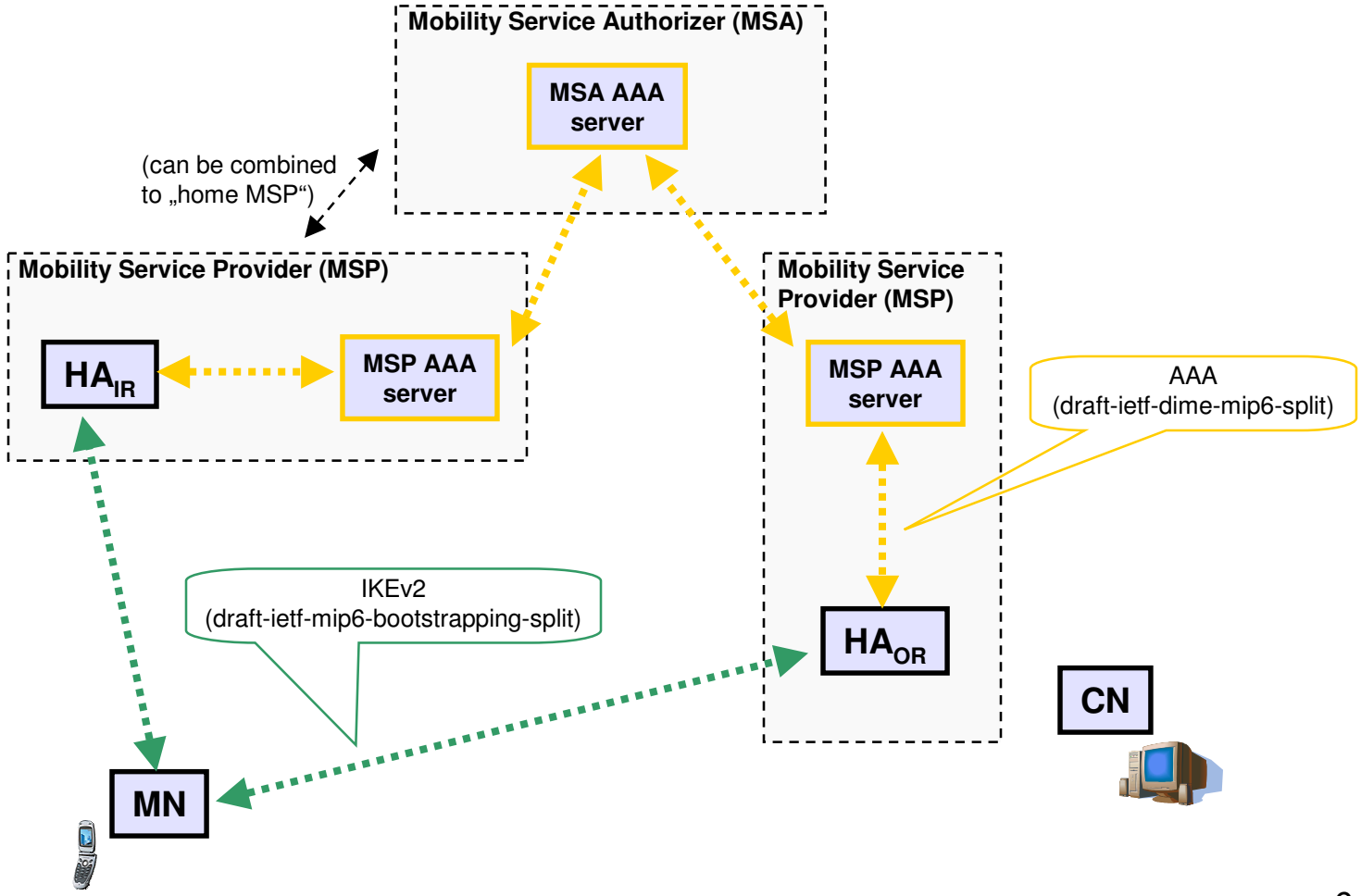
Proposed solution

- Approach
  - Bootstrap and reverse tunnel to another HA (HA_{OR}) in or nearby to CN domain

  → *HA_{OR} can provide optimized routing and HoA_{OR} has no location information*
- Case 1: MN-initiated session
  - MN reverse tunnels data to HA_{OR} with HoA_{OR} as source address
- Case 2: CN-initiated session
  - MN starts return routability and uses RO mode with HoA_{IR} as HoA and HoA_{OR} as CoA, i.e., CoT/i, BU, data is reverse tunneled to HA_{OR}



Mapping to MIPv6 bootstrapping architecture



HA_{OR} discovery

MN can obtain HA_{OR} address/name using

- DNS-based HA address discovery [draft-ietf-mip6-bootstrapping-split]
  - MN includes CN's prefix or domain name in QNAME, e.g., "ORHA.<CNdomain>" or "CNdomain.ORHA.<MSAdomain>"
- DHCP-based HA address discovery [draft-ietf-mip6-bootstrapping-integrated-dhc, ietf-mip6-hiopt]
  - MSA AAA server transmits all authorized HA addresses to NAS during network authentication
  - MN puts CN's domain as target network in Home Network Identifier Option of DHCP Information request msg
  - DHCP reply contains HA_{OR} address

Assumptions and Applicability

- If the MN is not able to discover and bootstrap with a trusted HA_{OR}, this optimization cannot be used
  - e.g., if no roaming relationship between MSA and MSP of HA_{OR} exists or if MN is not authorized to use this HA
- This optimization should only be used for sessions requiring simultaneous CN-targeted location privacy and optimized routing
  - for other sessions reverse tunneling to HA_{IR} or RO mode can be used
- To allow optimized routing to many or even any CN, MSA must have roaming relationships with MSP(s), which together offer HA services from various topological locations
  - this is also required for wide applicability of local HA service as specified in draft-ietf-mip6-bootstrapping-integrated

Changes in new draft version

- Clarified details for HA_{OR} discovery using DHCP/AAA
  - MSA must send potential HA_{OR} addresses to NAS during network auth
- Added section about mode selection
  - reverse tunneling or RO mode should be used if session is not delay-sensitive or no location privacy is required
- Added some text about scalability
  - MN should limit number of simultaneous HA_{OR} registrations
- Clarified HA_{OR} trust verification
  - MSA/MSP only assigns trusted HAs or MN verifies trust by itself
- Added section about home/source address selection
  - policy table defined in RFC 3484 can be used

Conclusion

- Currently, MIPv6 doesn't support scenarios where MN needs both location hiding from CN and optimized routing
- Proposed optimization achieves that with the existing MIPv6 bootstrapping extensions and without changes to HA or CN or to MIPv6 protocol msgs

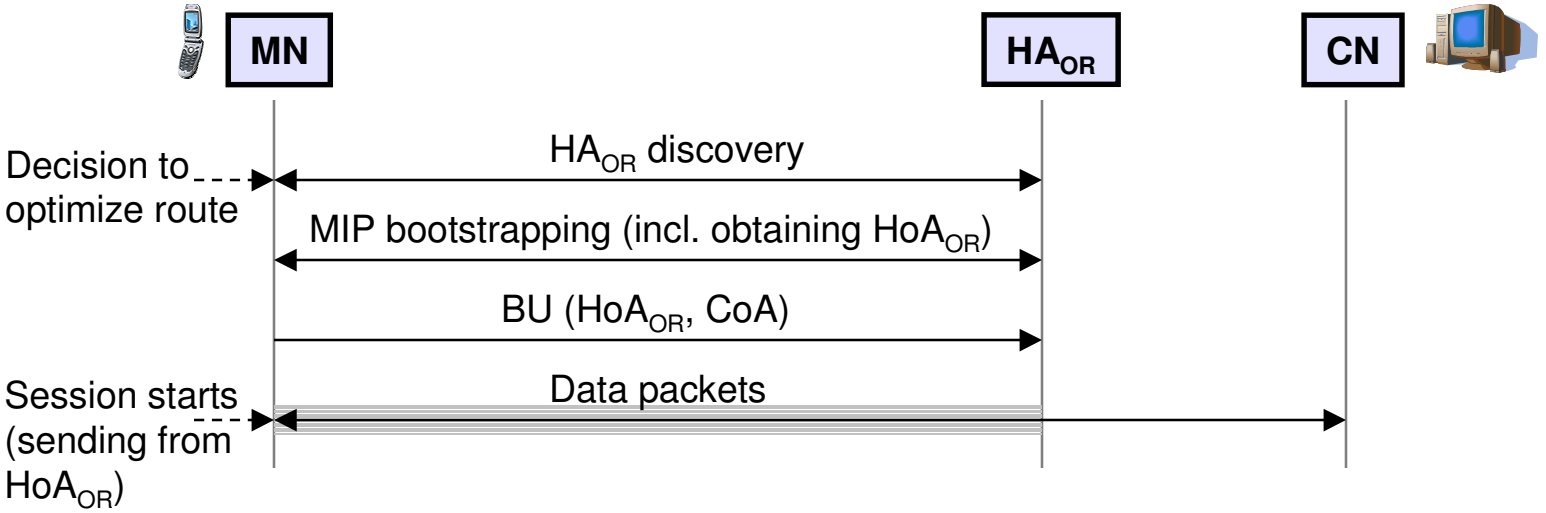
Thanks!

Questions/Comments?

Appendix

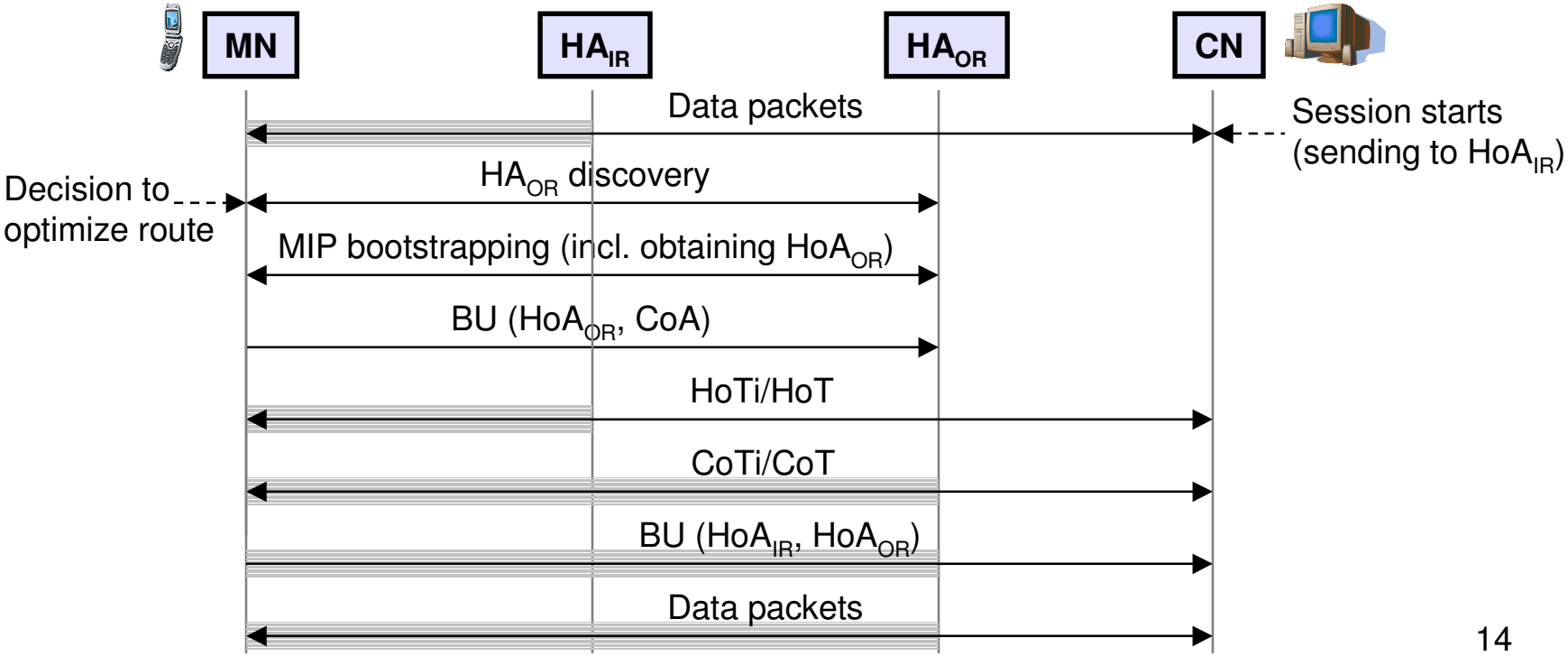
Signaling flow for case 1 (MN-initiated session)

- Before sending packets to CN, MN discovers HA_{OR}
- MN bootstraps with HA_{OR} and obtains HoA_{OR}
- MN uses HA_{OR} in bi-directional tunneling mode and HoA_{OR} for the session with CN
  - MN keeps registrations with other HAs, such as HA_{IR}



Signaling flow for case 2 (CN-initiated session)

- Packets are sent to/from MN's public HoA_{IR}
- MN discovers HA_{OR} and bootstrap with it
- MN performs return routability over reverse tunnel to HA_{OR} and registers HoA_{OR} as CoA at CN



Headers in case 2 (CN-initiated sessions)

- Data packets and BU sent by MN to CN

IPv6 header (source = care-of address,
destination = HA_{OR})
ESP header in tunnel mode
IPv6 header (source = *HoA_{OR}*,
destination = correspondent node)
Destination Options header
Home Address option (HoA_{IR})
Any protocol

- CoTi sent by MN to CN

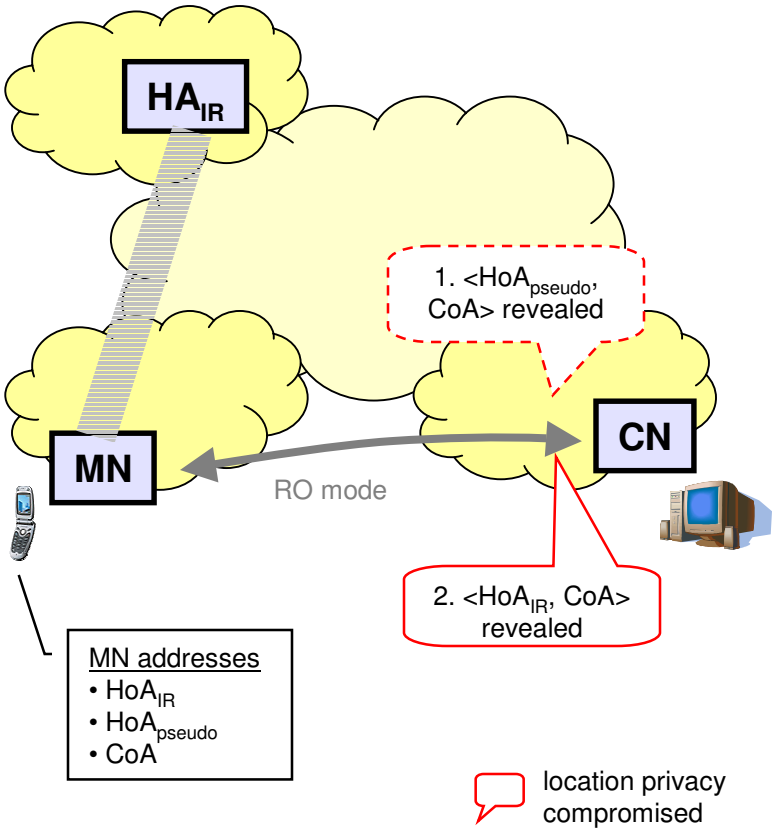
IPv6 header (source = care-of address,
destination = HA_{OR})
ESP header in tunnel mode
IPv6 header (source = *HoA_{OR}*,
destination = correspondent node)
Any protocol

How draft-irtf-mobopts-location-privacy-solutions addresses this problem

- Approach
  - MN discloses location to CN, but hides its identity by using pseudo HoA
- Case 1: MN-initiated session
  - MN uses RO mode with HoA_{pseudo} as HoA

    → *Issue*: *location privacy is compromised if CN figures out MN's identity during session*
- Case 2: CN-initiated session
  - Since CN initiated session using HoA_{IR}, it already knows MN's identity

    → *Issue: no solution to the problem in this case*



Location privacy issues when local HA is used

- Approach
  - Disclose location and identity, but hide fact that HoA_{local} contains location information
- Case 1: MN-initiated session
  - MN bootstraps with local HA_{local} and uses reverse tunneling mode

    → *Issue: location privacy is compromised if CN knows identiy associated with HA_{local} and knows that HoA_{local} is anchored at local HA*
- Case 2: CN-initiated session
  - To be reachable, MN publishes <HoA_{local}, identity>

    → *Issue: location privacy is compromised if CN knows that HoA_{local} is anchored at local HA*