MIPv6 Binding Lifetime Extension

MOBOPTS RG
IRTF/IETF-60

Jari Arkko (jari.arkko@nomadiclab.com)
Christian Vogt (chvogt@tm.uka.de)
Outline of the Presentation

• Reasons for optimization
• RFC 3775 approach to lifetimes
• Our proposed alternative approach
  Simple - no config, no fancy crypto, one new option
  Based on exponentially earned lifetime credit
• Analysis
  Up to 70-fold decrease in amount of signaling
Reasons for Optimizations
Reasons for Optimization

• RFC 3775 RR efficiency:
  – Generally requires 6 messages (376 bytes)
  – These are per movement and per peer
  – And two round-trips

• Not a problem for current normal usage
  – Not issue upon movements because the rest of stack uses even more messages

• However, it can still be an issue when
  – Nodes don’t move that often
  – The rest of the stack becomes faster
Nodes that do not move often

• Movement frequencies
  – Movement is inherently infrequent on many link layers (GSM, UMTS, CDMA)
  – While frequent movements can happen on some link layers (WLAN), it is unlikely to be the most common case

• RFC 3775 RR causes 7.16 bits/s, if a node wishes to keep its RO state up

• This is not that significant, but waking up every few minutes may be
RFC 3775 Approach to Lifetimes
RFC 3775 Approach to Lifetimes

Movement

time
RFC 3775 Approach to Lifetimes

- RR protocol exchange
- Movement
RFC 3775 Approach to Lifetimes

- Movement
- RR protocol exchange
- Binding expires

→ time
RFC 3775 Approach to Lifetimes

Lifetime max 7 minutes

- Movement
- RR protocol exchange
- Binding expires

**time**
Why Have the Max Limit?

• It limits so called *time shifting* attacks
• If there was no limit, I could visit your network *today* and launch an amplified DoS attack on it *next month*
• With current RR, you have to have very recent *physical presence* to do it
Our Proposed Alternative Approach
The Basic Idea

• RFC 3775 rationale for limiting lifetimes is valid but there are other ways to do it besides the fixed limit

• We apply a “lifetime credit” based limit

• A node that just appeared for the first time gets a very short lifetime

• A node that has been on the same place for a long time will get a longer lifetime
The Exponentially Growing Lifetime
The Exponentially Growing Lifetime

Movement 1st RR run

Lifetime = 2 min

time
The Exponentially Growing Lifetime

Lifetime$_0$ = 2 min
Lifetime$_1$ = 3 min

Movement
1st RR run
2nd RR run
The Exponentially Growing Lifetime

Movement

1st RR run

2nd RR run

3rd RR run

nth RR run

Lifetime₀ = 2 min

Lifetime₁ = 3 min

Lifetimeₙ = 8 hrs

time
Protocol Details

• The **Lifetime Credit Authorization** mobility option (inside a BU) carries the request for using this type of lifetimes

• Includes an authenticator which shows knowledge of all past Kbm values at this location
  – $K_{credit} = \text{hash}(K_{bmN} \mid \text{hash}(K_{bmN-1} \mid \ldots))$

• **Movement resets the lifetime** back to its initial value
Analysis
Security

• We argue that this lifetime assignment -- even if different from RR -- is at least as fair and secure as in RR
  – First binding(s) after a movement have smaller lifetime than in RR -- less exposure to time shifting attacks
  – Subsequent bindings can have a large (up to 8 hrs) lifetime
  – But the involved nodes must have “invested” physical presence on the link to achieve this for much longer time (at least 24 hrs)
Efficiency

• For seldomly moving mobile nodes, there is less signaling
• 70-fold improvement in the steady state (from 7 bits/s to 0.1 bits/s)
• Nodes that expect to stay in one place at most 7 minutes should use the RFC 3775 method
Questions?