

Bound End-to-End Tunnel (BEET)

Presentation at 58th IETF, Minneapolis

Pekka Nikander

Ericsson Research Nomadiclab

`draft-nikander-esp-beet-mode-00.txt`

Presentation outline

- Background
- BEET in a nutshell
- Motivation
- Answers to common objections
- Summary



Background

- mobike proposing mobility extensions to IKEv2
- nsrg, multi6, and hip discussing id/locator split
- Separate **end-point identifier** and **locator** roles of IP addresses
- Avoid transport protocol reconnection when underlying IP addresses change

BEET in a nutshell

- Transport header but tunnel semantics
 - A fixed pair of inner addresses
 - Address ranges **not** allowed
- = Transport mode + Bellovin's hostNAT



Motivation I: save bytes

- “This is useless, just use tunnel mode!”
- Counter-argument: sometimes bytes matter

Headers	Uncompressed	ROCH
Baseline: IPv4 + TCP	20 + 20	2
IPv4 + ESP + IPv4 + TCP	80	58
IPv4 + ESP + TCP	60	38
IPv6 + ESP + IPv6 + TCP	120	78
IPv6 + ESP + TCP	80	38

51% saving

Motivation 2: Id/loc split

- Inner addresses work as **end-point identifiers**
 - Visible to upper layer protocols
 - No change with mobility / multi-addressing
- Outer addresses work as **locators**
 - Bound to the topological location
 - Change with mobility / multi-addressing
- Difference to tunnel mode is **architectural**
 - Inner addresses internal, not visible on wire

Common objections (and answers to them)

- “Adds complexity”
 - Does 98 lines of code really matter?
- “Hard to add to existing implementations”
 - Make optional, use tunnel mode if not there
- “Optional features are bad for portability”
 - Easy to check whether supported (PF_KEY)
- “Not needed”
 - NAT traversal, HIP, multi6, ...

Summary



- New mode to ESP
 - Tunnel semantics, inner and outer addresses
 - Fixed inner addresses, no address ranges
 - Transport mode header structure
- PF_KEY support via SADB_IDENTTYPE
- Up to 51% header savings when ROCH is used
- Facilitates id/locator separation
- Minimal added complexity: 98 lines of code