

UDP Encapsulation for IPv4 and IPv6 in DS-MIPv6

MIP6 WG, IETF 68

Vijay Devarapalli (vijay.devarapalli@azairenet.com)

Use of UDP encapsulation in DS-MIPv6

- UDP encapsulation is used in DS-MIPv6 for NAT Traversal
- Two formats supported
 - IPv6-in-UDP-over-IPv4
 - IPv4-in-UDP-over-IPv4
- There is an issue when multiple protocol headers are encapsulated in UDP
 - Hard for the recipient to figure out what follows next
- Note: No port number associated with MIPv6 today

Use of UDP encapsulation in DS-MIPv6

- Indicating the type of protocol header in the UDP header would be desirable
- Multiple options are available
 - Nothing new, mostly re-using known concepts
- Pick one option for DS-MIPv6
 - Same solution would be applicable for PMIPv6 too

Alternative 1

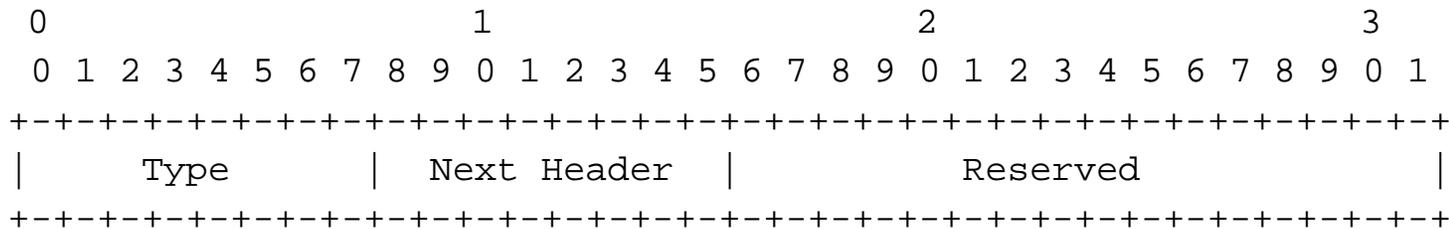
- Parse the protocol header that comes after the UDP header
 - For e.g look for the version field that says IPv4 or IPv6
 - This does not work for other encapsulated headers
- If DS-MIPv6 is not implemented in the kernel, then it is still an issue

Alternative 2

- One reserved UDP port per protocol header that is encapsulated
 - One UDP port for IPv6-in-UDP-over-IPv4
 - One UDP port for IPv4-in-UDP-over-IPv4
- If needed more ports can be reserved for each additional protocol header
 - For e.g., one for GRE later on
- Disadvantage is that the DS-MIPv6 application need to listen on each of these ports

Alternative 3

- One reserved UDP port and a DS-MIPv6 “tunnel type message”
 - The DS-MIPv6 tunnel type message will say what protocol header follows
 - Similar to MIPv4 NAT traversal (RFC 3519)



- Adds a 4 byte overhead for every tunneled packet

Alternative 4

- The encapsulated protocol header type is indicated in the BU
 - One reserved UDP port number is still needed
- A new mobility option in the BU/BAck
- Does not introduce per-packet overhead

Comments/Questions?