BIER powered Homenet multicast routing

Markus Stenberg <fingon+ietf92@iki.fi>
(with some handholding from others)

The idea

- ‘The party’:
  - BIER needs unique bits for (egress) routers
  - HNCP has well ordered set of indexes for routers
  - Babel is a cool unicast routing protocol
  - => marriage made in heaven
- No changes to HNCP needed (0 TLVs)
- How to encode the BIER bits in IPv6?
  - Extension header (elegant, slow, hard)
  - MPLS (not done here)
  - Steal IPv6 bits (who needs 112 bit group ID anyway? RFC3306, embedded RP do this too.. ~64 bits stealable?)

BIER is .. “hard”

- ~200 lines in Linux kernel (new C module)
    - netfilter module (pre-routing IPv6 hook) that does
        - BIER unicast forwarding (=unicast route lookups, duplication on per-unique-next-hop and BIER bitmask changes) and
        - local delivery (if bit matches) / drop (no bit match)
- 46 lines of Python
    - which configures the kernel module based on HNCP state
- Few lines of change to hnetd multicast script
    - to invoke the Python script (too often) when something changes
- => win, forwarding path prototype done in a couple of hours

What is missing?

- Currently client has to encode the address BIER bits manually
- However, MLD / IGMP proxy could do subscription handling
    - On reception convert e.g. ff05::x ==> ff05:0:1234:5678::x as needed given subscription state
    - (inverse is already handled)
- IPv4 support, but it could be added by encoding it as IPv6 group within the network
- Could also have e.g. HNCP BIER support TLV so only create indexes for routers that care, or are egress routers
- Lots of bug fixes - quick-n-dirty prototype :)

Demo + Questions?

- (Minimal network topology demo with 10 nodes but just 5 of them BIER routers..)