Transmission of IPv6 Packets over IEEE 802.11 OCB Networks

draft-ernst-its-ipv6-over-80211ocb-00.txt
draft-lee-its-ipv6-over-80211ocb-00.txt

Thierry Ernst, Alex Petrescu (speaker),
Jong-Hyouk Lee

Acknowledgements:
Albin Severinson, Marios Makassikis, Michelle Wetterwald,
Dick Roy, Robert Moskowitz, Alexey Voronov, José Santa Lozano
and “starkj” on itsforge.net

IETF Berlin, July 19th, 2016

Contents of a “IPv6-over-80211 OCB” document at IETF

- The Title
- Maximum Transmission Unit:
  - What is the size of the MTU on 802.11-OCB?
  - How to communicate the MTU between 802.11-OCB nodes?
  - IP or MAC fragmentation?
- Frame format:
  - What is the value of the Ethertype on 802.11-OCB?
  - What is the frame format on 802.11-OCB?
- Interface ID: how to build the Interface ID?
- Address mappings: how to map certain unicast and multicast addresses between 802.11 and IP layers?

Title

- Transmission of IPv6 Packets over IEEE 802.11 Networks Outside the Context of a Basic Service Set

- Transmission of IPv6 Packets over the IEEE 802.11p OCB Mode

- Transmission of IPv6 Packets over IEEE 802.11-OCB Networks

Maximum Transmission Unit size

- Currently MTU for 802.11-OCB == 1500 bytes as wired Ethernet

- ETSI CAM specifications are not TLV but ASN.1 “containers”/”matrioska” – no size limit

- 802.11-OCB can transmit >1500byte MTU
- 802.11-OCB MTU is 2308 bytes [RR] ⟹ Should we use this MTU?

- Size of common packets captured on 802.11-OCB:
  - WSMP (non IPv6):
    - Unsigned BSM 171 bytes
    - Signed BSM 335 bytes
    - SPAT 350 bytes
    - MAP 663 bytes
  - CAM (non IPv6):
    - At ITS Congress demo: 108 and 524 bytes
    - At GCDC 2016 demo: 102 bytes
  - IPv6:
    - CAM ‘heresy’ as UDP/IPv6 payload: 76 bytes
    - RA: 125 bytes

- MTU bigger than 1500bytes may be useful for:
  - List of Points of Interest (POI) advertised by Road-Side Units
  - Multiple authentication signatures added to large messages

- Determine the real MTU:
  - PMTUD open source?
  - Algorithm: send an increasing UDP packet until flag “802.11 More Fragments” flips.

Fragmentation

- Not a typical topic in IPv6-over-foo, but seems favorable to include.
- To respect the MTU of 802.11-OCB:
  - IP layer fragments, _and_ MAC layer fragments
  - IP layer does not fragment, MAC layer fragments
  - IP layer fragments, MAC layer does not fragment
  - ␣ which one to use and write?

Frame Format

4. Frame Format

IPv6 packets can be transmitted as "IEEE 802.11 Data" or alternatively as "IEEE 802.11 QoS Data“:

3 independent sources

IEEE 802.11 Data
Logical-Link Control
IPv6 Header

IEEE 802.11 QoS Data
Logical-Link Control
WSMP Header - *if* replaced by IPv6 header

1 source

Details

```
IEEE 802.11 Data, Flags: ........C
    Type/Subtype: Data (0x0020)
    Frame Control Field: 0x0800
        .... ..00 = Version: 0
        .... 10.. = Type: Data frame (2)
        0000 .... = Subtype: 0
      ⊞ Flags: 0x00
    .000 0000 0000 0000 = Duration: 0 microseconds
    Receiver address: IPv6mcast_01 (33:33:00:00:00:01)
    Destination address: IPv6mcast_01 (33:33:00:00:00:01)
    Transmitter address: WistronN_e6:47:99 (30:14:4a:e6:47:99)
    Source address: WistronN_e6:47:99 (30:14:4a:e6:47:99)
    BSS Id: Broadcast (ff:ff:ff:ff:ff:ff)
    .... .... .... 0000 = Fragment number: 0
    1110 1100 1001 .... = Sequence number: 3785
    Frame check sequence: 0xd20e2270 [correct]
        [Good: True]
        [Bad: False]
Logical-Link Control
    DSAP: SNAP (0xaa)
        1010 101. = SAP: SNAP
        .... ...0 = IG Bit: Individual
    SSAP: SNAP (0xaa)
        1010 101. = SAP: SNAP
        .... ...0 = CR Bit: Command
    Control field: U, func=UI (0x03)
        000. 00.. = Command: Unnumbered Information (0x00)
        .... ..11 = Frame type: Unnumbered frame (0x03)
    Organization Code: Encapsulated Ethernet (0x000000)
    Type: IPv6 (0x86dd)
⊞ Internet Protocol Version 6, Src: fe80::3214:4aff:fee6:4799, Dst: ff02::1
⊞ User Datagram Protocol, Src Port: 39419 (39419), Dst Port: 50000 (50000)
⊞ Data (41 bytes)
```

```
IEEE 802.11 QoS Data, Flags: ........C
    Type/Subtype: QoS Data (0x0028)
    Frame Control Field: 0x8800
        .... ..00 = Version: 0
        .... 10.. = Type: Data frame (2)
        1000 .... = Subtype: 8
      ⊟ Flags: 0x00
            .... ..00 = DS status: Not leaving DS or network is operating in AD-HOC mode (To DS: 0 From DS: 0) (0x00)
            .... .0.. = More Fragments: This is the last fragment
            .... 0... = Retry: Frame is not being retransmitted
            ...0 .... = PWR MGT: STA will stay up
            ..0. .... = More Data: No data buffered
            .0.. .... = Protected flag: Data is not protected
            0... .... = Order flag: Not strictly ordered
    .000 0000 0000 0000 = Duration: 0 microseconds
    Receiver address: Broadcast (ff:ff:ff:ff:ff:ff)
    Destination address: Broadcast (ff:ff:ff:ff:ff:ff)
    Transmitter address: 00:cf:8f:90:82:6a (00:cf:8f:90:82:6a)
    Source address: 00:cf:8f:90:82:6a (00:cf:8f:90:82:6a)
    BSS Id: Broadcast (ff:ff:ff:ff:ff:ff)
    .... .... .... 0000 = Fragment number: 0
    0001 1100 0010 .... = Sequence number: 450
    Frame check sequence: 0xeb02f593 [correct]
        [Good: True]
        [Bad: False]
    Qos Control: 0x0022
        .... .... .... 0010 = TID: 2
        [.... .... .... .010 = Priority: Spare (Background) (2)]
        .... .... ...0 .... = EOSP: Service period
        .... .... .01. .... = Ack Policy: No Ack (0x0001)
        .... .... 0... .... = Payload Type: MSDU
        0000 0000 .... .... = TXOP Duration Requested: 0 (no TXOP requested)
Logical-Link Control
    DSAP: SNAP (0xaa)
        1010 101. = SAP: SNAP
        .... ...0 = IG Bit: Individual
    SSAP: SNAP (0xaa)
        1010 101. = SAP: SNAP
        .... ...0 = CR Bit: Command
    Control field: U, func=UI (0x03)
        000. 00.. = Command: Unnumbered Information (0x00)
        .... ..11 = Frame type: Unnumbered frame (0x03)
    Organization Code: Encapsulated Ethernet (0x000000)
    Type: (WAVE) Short Message Protocol (WSM) (0x88dc)
⊞ Wave Short Message Protocol(IEEE P1609.3)
```

Frame Format

- Reasons to use 802.11 Data Headers
  - Most prototypes do
  - Small and fast
- Reasons to use 802.11 QoS Data Headers
  - Map flows of IP data on separate DSRC/ITS-G5 channels
  - IP diffserv may apply
  - draft-ietf-tsvwg-ieee-802-11 proposes mappings of Differentiated Services Code Points (DSCP) to 802.11 User Priorities (UP) for Telephony, Multimedia Conferencing, Low-Latency Data ␣ why not DSRC Control Channel?
- Which frame format to use and to write?

Interface ID

- See the 6man Interface ID
- See the privacy MAC addresses

Address Mapping – unicast

- Work under Way.

Address Mapping – multicast

- ff02::1 ␣ 33-33-0-0-0-1
- ff02::XXXX:XXXX ␣ 33-33-XX-XX-XX-XX
- Destination alternatives:
  - 33-33-0-0-0-1
  - ff-ff-ff-ff-ff-ff “broadcast” for IPv4
  - 01-80-C2-00-00-1D reserved by IEEE for ISO/IEC JTC1/SC6 ISO/IEC 9542 as “All Multicast Capable Intermediate Systems Address”
  - 01-00-5E- reserved by IEEE for multicast addresses under IANA OUI (they are not Local).
  - ␣ Which one to use?

Address Mapping – multicast

For Link scoped IPv6 multicast addresses:

A Group ID TBD of length 112bits may be requested from IANA; this Group ID signifies "All 80211OCB Interfaces Address". Only the least 32 significant bits of this "All 80211OCB Interfaces Address" will be mapped to and from a MAC multicast address.

For other than link- and interface- scoped IPv6 multicast addresses:

Application-specific multicast addresses may be requested later for use in vehicular communications use-cases (i.e. “all Uber cars in the street”).