Transmission of IPv6 Packets over IEEE 802.15.4

Tuesday, March 8, 2005

62 IETF, Minneapolis

Salon E, 0900-1100

Gabriel Montenegro – Sun

Nandu Kushalnagar - Intel

Why this document?

- *Basic requirement to run IPv6*
  - *“IPv6 over Foo”*
  - *“Sub-IP” section in node-requirements document*
- *Other Examples of “IPv6 over Foo” documents*
  - *Ethernet – RFC 2464*
  - *PPP – RFC 2472*
  - *ATM - RFC 2492*
  - *IEEE 1394 – RFC 3146*
- *MAY 2005: working group last call*
- *JULY 2005: Send to IESG as Proposed Standard*

Topics

- *IEEE 802.15.4 mode for IP*
- *MTU*
- *Frame format and Adaptation Layer*
- *Stateless Address Autoconfiguration*
- *IPv6 Link-local Addresses*
- *Unicast Address Mapping*
- *Header Compression*
- *Packet Delivery in a mesh*
- *Security considerations*

IEEE 802.15.4 mode for IP

- *IP packets to be carried on Data Frames*
- *Non-beacon enabled network*
  - *No beacons for synchronization, but certainly for device discovery*
  - *Contention-based channel access (csma/ca)*
  - *No Guaranteed Time Service (GTS)*
- *Addressing mode:*
  - *both source and destination addresses included (PAN IDs also possible)*
  - *Two types of MAC addresses:*
    - *64 bit Extended Addresses*
    - *16 bit short addresses are possible (TO BE DONE)*
- *PAN ID (16 bits) <==> IPv6 Link (i.e., IPv6 Prefix)*
  - *Current assumption: a given PAN ID maps to a given IPv6 link*
  - *Not specified: how to do the mapping.*
  - *One possibility: /48 + 16 bits of PAN ID ==> IPv6 prefix*

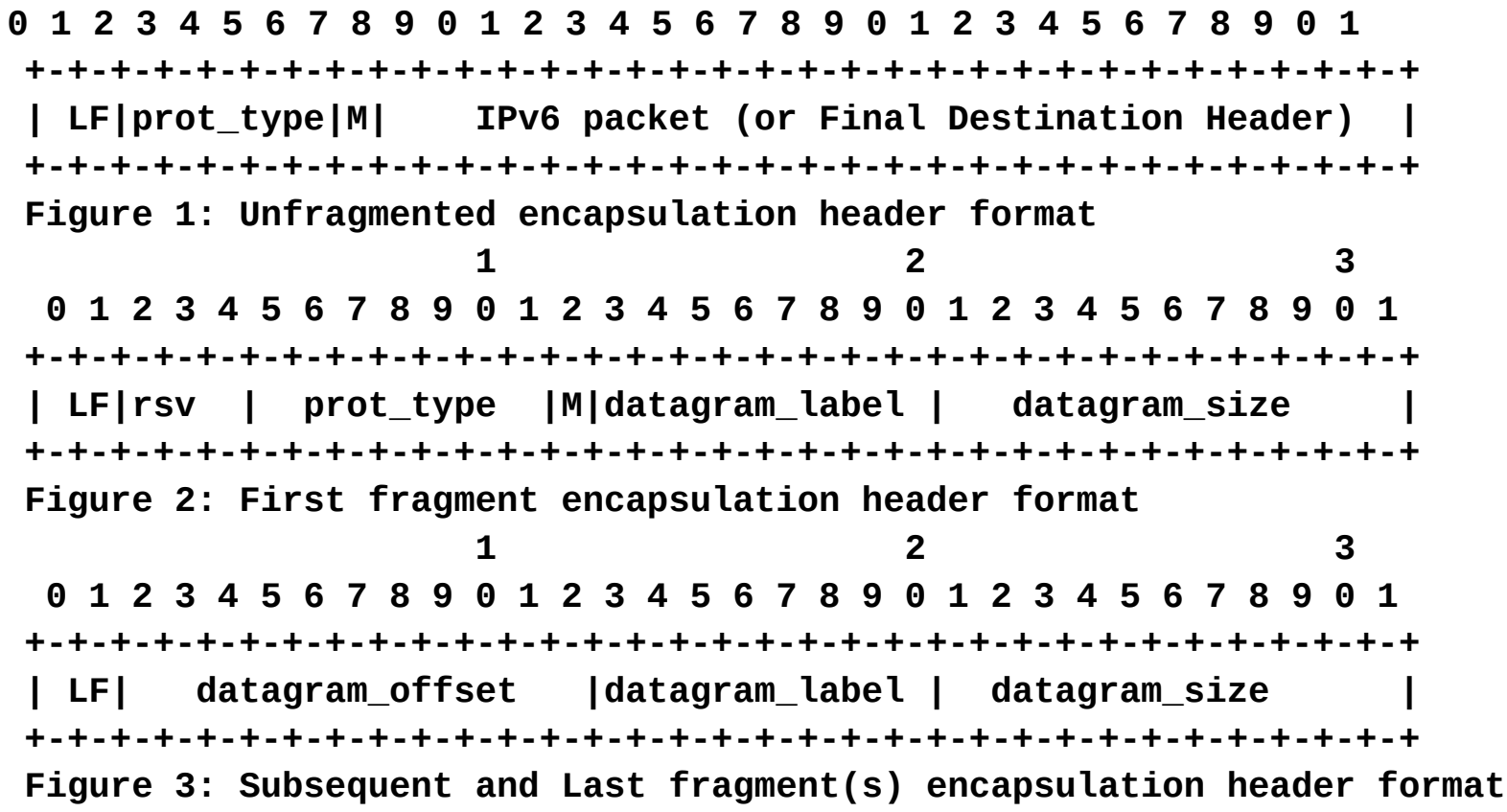
IPv6 MTU

- ***IPv6 MTU over IEEE 802.15.4 SHALL BE 1280 (minimum IPv6 MTU)***
- ***802.15.4 MTU is***
  - ***102 octets w/out security***
  - ***81 octets with AES in max CCM mode***
- ***This means***
  - ***Need an adaptation layer***
  - ***Need header compression for IP (not a luxury)***

LoWPAN Service Layer

- *Adaptation layer*
- *Allows "larger" (than IEEE 802.15.4) packets to be sent*
- *Allows packet delivery in a mesh*
- *Allows protocol multiplexing via protocol type*
- *Current protocol types are:*
  - *IPv6 (prot_type = 1)*
  - *IPv6 compressed headers (prot_type = 2)*
  - *AODV for 802.15.4 (prot_type = 4)*
  - *Other protocols*
  - *Some applications themselves*

Frame format and Adaptation Layer (1/2)



- These are the payload in the IEEE 802.15.4 MAC protocol data unit (PDU)

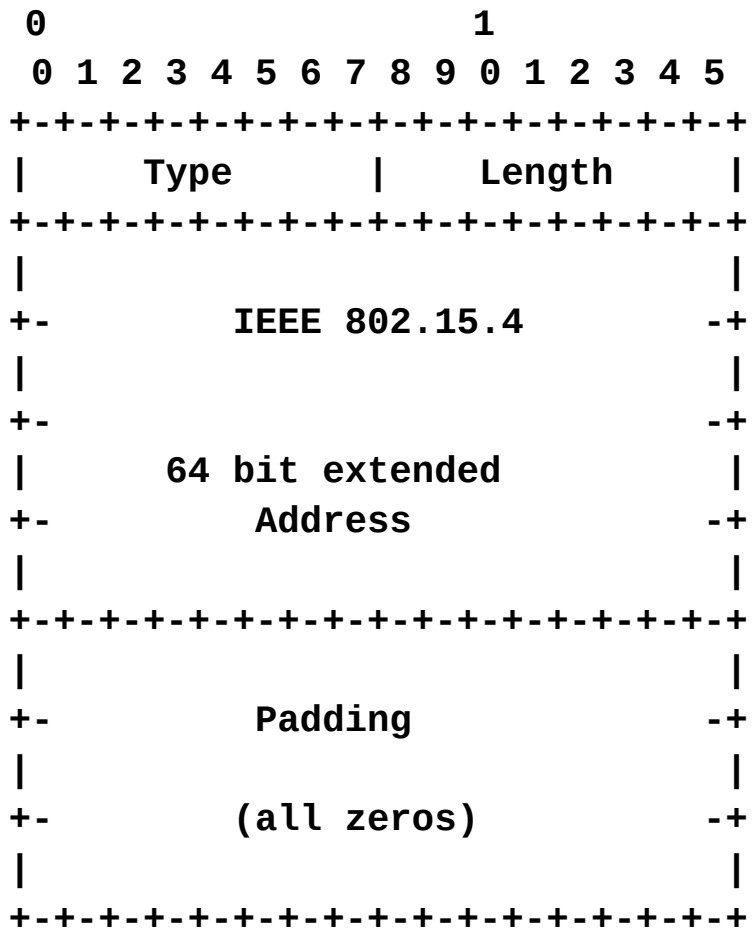
Frame format and Adaptation Layer (2/2)

LF	Position	
00	Unfragmented	(Figure 1)
01	First Fragment	(Figure 2)
10	Last Fragment	(Figure 3)
11	Interior Fragment	(Figure 3)

Stateless Address Autoconfiguration

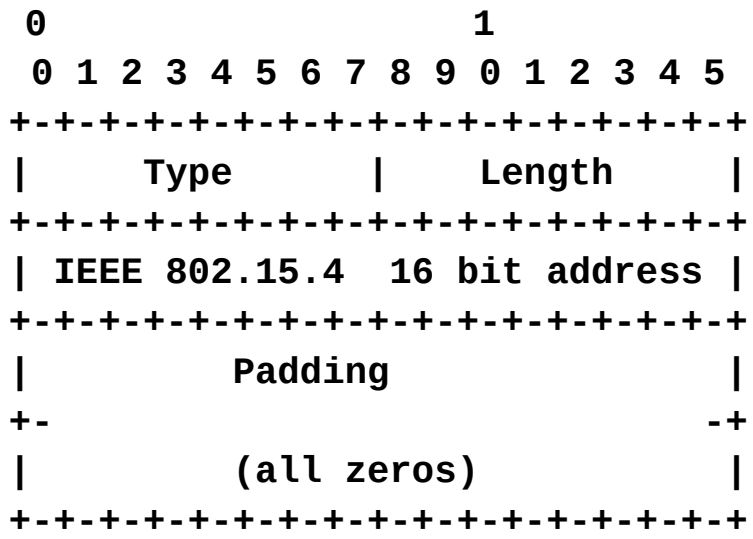
- *EUI-64 addresses for IEEE 802.15.4*
- *Interface Identifier (IIF) obtained via IPv6 over Ethernet (RFC 2464)*
- *Prefix must be 64 bits*
- *But also: mapping to 16-bit short addresses possible (to be done)*
- *Link-local addresses as usual: FE80::IIF*

Unicast Address Mapping (1/2)



- Length in units of 8 octets: 2 (64 bit extended addresses)

Unicast Address Mapping (2/2)



- Length in units of 8 octets: 1 (16 bit extended addresses)
- To be done...

Header Compression in a Mesh

- *Very simple (one flow perhaps between peers), very low context*
- *Integrate Layer 2 and Layer 3 compression to gain space*
- *Only if necessary use pairwise context-building as per usual header compression*
- *Shared context is preferred (no need to build context)*

LoWPAN Header Compression

- Most common IPv6 header:
  - IP version = IPv6
  - IPv6 source and destination both are link local
  - Length inferred from IEEE 802.15.4 header
  - Traffic Class and Flow Label both are 0
  - Next Header = UDP, ICMP or TCP
  - Only need to carry Hop Limit
- This common case is highly compressible
  - 2 octets instead of 40
  - Alternate encoding allows further compression to 1 octet (to be done)
- UDP header compression
  - Allows use of 4 bit (potentially 8 bit) port ranges (P+range)
  - UDP header compressible from 8 octets to 4
- TCP header compression can use existing specs with proper prot_type (to be done)

Packet Delivery in a Mesh

- *IEEE 802.15.4 networks are expected to implement meshes*
- *But IEEE 802.15.4 does not specify how to*
- *Achieved via a simple extra header with this info*
  - *Final Destination (addresses are 64 bit, but 16 bit possible)*
  - *Hops Left*
- *Allows any upper layers (via protocol type) to benefit from this layer two mesh*