Table of Contents

在 IEEE 802.15.4 网路上传送 IPv6 报文

在线阅读

Gitbook | Github

离线阅读

PDF | ePub | Mobi

英文原版

点击查看英文原版

强烈建议读完中文后再阅读英文。

当您认认眞眞读完一篇英文后，会发现原来读英文也如此轻松！

做贡献

如果您发现文档有任何错误，包括错别字，歧义等，您可以：

- 自己修改，提交 pull request，我会 merge 你的改动。
- 提问题(issue)，我会根据 issue 修改相应的内容。

更多关于物联网标准的中文文档，请移步：

物联网相关的标准文档(IEEE、IEFT RFC)中文版汇总

摘要

本文描述了在 IEEE 802.15.4 网络上传送 IPv6 报文的帧格式，和生成 IPv6 链路地址和无状态自动配置地址的方法。另外还包括了一个简单的使用共享上下文的报头压缩方法和在 IEEE 802.15.4 meshs 网路中传送报方的方法。

1 简介

IEEE 802.15.4 标准 [ieee802.15.4] 的目标是低功耗个人局域网。本文定义了在 IEEE 802.15.4 网路上传送 IPv6[RFC2460]报文的帧格式和 IPv6 链路地址和无状态自动配置地址等信息。 由于 IPv6 的报文长度比 IEEE 802.15.4 最大帧长还要大得多， 所以这里定义了一个适配层。 为了使 IPv6 能在 IEEE 802.15.4 网路上运行和在 IEEE 802.15.4 meshs 上进行报文传送， 本文 还定义了报文压缩机制。然而，一个完整的 mesh 路由定义（使用特定的协议，集成邻居发 现等）不在本文的讨论范围内。

1.1 符号说明

本文使用的关键词“ MUST”、“ MUST NOT”、“ REQUIRED”、“ SHALL”、“ SHALL NOT”、“ SHOULD”、“ SHOULD NOT”、“ MAY”和“ OPTIONAL”请参考[RFC2119]的描述。

1.2 术语使用

AES - Advanced Encryption Scheme

高级加密标准

CSMA/CA - Carrier Sense Multiple Access / Collision Avoidance

载波侦听多路访问/冲突避免

FFD - Full Function Device

全功能设备

GTS - Guaranteed Time Service

保护时隙

MTU - Maximum Transmission Unit

最大传输单元

MAC - Media Access Control

介质访问控制

PAN - Personal Area Network

个人局域网

RFD - Reduced Function Device

简化功能设备

2 IP 的 IEEE 802.15.4 模式

IEEE 802.15.4 定义了四种类型的帧：信标帧，MAC 命令帧，确认帧和数据帧。IPv6 报文必须放在数据帧里。数据帧可以选择是否需要被确认。为了符合 [RFC3819]，为了帮助链路恢复，建议在传送 IPv6 报文时设置该帧为需要确认。

IEEE 802.15.4 网络有两种模式：信标使能模式和非信标使能模式 [ieee802.15.4]。 后者是一个可选模式，网络中的设备通过协调器信标进行同步[注：据我目前的理解，这句话应该是不对的，后面再校正]。IEEE 802.15.4 允许使用超帧，在超帧内可以使用免竞争的 GTS。本文档不要求 IEEE 网络必须运行在信标使能模式下。在非信标模式网络中，传送数据帧（包括运载 IPv6 报文的数据帧）时是通过基于竞争的非时隙版 CSMA/CA 来访问信道的。

在非信标模式网络中，信标不是用来同步的，但是依然可用于链路层的设备发现，即用于设备与网络的关联和解关联。本文建议配置信标以实现这些功能。A further recommendation is for these events to be available at the IPv6 layer to aid in detecting network attachment, a problem being worked on at the IETF at the time of this writing.[注：英文原版写得狗屁不通，直接贴英文，不翻译了。看来 RFC 文档也不全干货，很多水分！啥叫建议配置信标？信标使能模式和非信标模式的 PAN 网络中都必须要有信标好不好！啥叫实现这些功能？哪些功能？很怀疑作者有没有认真阅读 IEEE 802.15.4 的文档]

IEEE 802.15.4 允许在帧中省略源地址或/和目标地址，但是本文档定义的机制要求在帧头中必须包含源地址和目的地址，以及可以包含 PAN ID 和目的 PAN ID 字段。

3 编址模式

IEEE 802.15.4 定义了两种编址模式：64 位扩展地址模式和（在设备与网络关联后）16 位地址模式（这个地址在 PAN 网络中是唯一的）。本文档既支持 64 位扩展地址，又支持 16 位短地址。本文对 16 位短地址的格式增加了一些限制（除了 IEEE 802.15.4 要求的之外），具体描述见第 12 章。短地址实质上是瞬态的，需要注意的是：由于短地址是由 PAN 协调器在进行关联操作时分配的，所以它们只在此次关联的生命周期内是有效的、唯一的。当关联周期到期，或者 PAN 协调器发生灾难时，短地址的生命周期就结束了。由于集中分配和 PAN 协调器的单点故障而产生的可伸缩问题，开发者在使用短地址部署网络时必须仔细权衡网络的增长（并实行必要的机制）。当然， IEEE 64 位扩展地址就不会有这样的缺点，但仍然会有一些可伸缩问题，比如路由、设备发现、配置等。

本文假定一个 PAN 映射到一个特定的 IPv6 连接。这符合共享网络支持链路层子网广播的建议 [RFC3819]。严格来说，在 IPv6 里存在多播而不存在广播。而 IEEE 802.15.4 不支持多播，所以 IPv6 层多播报文必须放在 IEEE 802.15.4 网路的链路层广播帧里。这是强制要求的，因为广播帧只被特定链路的 PAN 里的设备所接收。安装 [ieee802.15.4] 第 7.5.6.2 的要求，必须要实现如下两点：

1. 帧里包含一个目的 PAN 标识符，它必须与链路层的 PAN ID 相匹配。
2. 帧里包含一个目的短地址，它必须与广播地址（0xffff）相匹配。

另外，第 9 章的 IPv6 多播地址映射的支持必须只用在一个 mesh 配置里。本文对这个功能不做更具体的描述。

通常，主机从路由器广告里学习 IPv6 前缀 [RFC4861]。

4 最大传输单元 MTU

在 IEEE 802.15.4 上的 IPv6 报文最大传送单元是 1280 字节。然而， 一个 IEEE 802.15.4 帧里 容不下一个完整的 IPv6报文。802.15.4 协议数据单元的大小取决于报文头的大小 [ieee802.15.4]。由于一个物理层的最大帧长是 127 字节（aMaxPHYPacketSize）且一个最大帧头长度是25 字节（aMaxFrameOverhead），所以 MAC 层的最大帧长是 102 字节。链路安全协议还需要增加报头长度，最大的情况（对于 AES-CCM-128 来说是 21 字节， AES-CCM-32和 AES-CCM-64 分别是 9 和 13 字节），只剩下 81 字节可用。这明显远不够 IPv6 报文的最小长度 1280 字节，为了和第 5 章的 IPv6 规范 [RFC2460] 一致， IP 层下必须提供一个分片和重组的适配层。 适配层在第 5 节定义。

另外，由于 IPv6 报头长度是 40 字节，这就只剩下 41 字节给上层协议，如 UDP。 UDP 使用 8 字节的报头，那应用数据就只剩下 33 字节了。并且，如上面提到的，还需要一个分片和重组的适配层，这将消耗更多的报头字节。

上述问题引起了如下两点：

1. 适配层必须满足 IPv6 对于最小 MTU 的要求。然而(a)多数 IEEE 802.15.4 应用并不会使用这么大的报文，(b)小量的应用数据和使用合理的报头压缩可以产生适合于一个 IEEE 802.15.4 帧的报文。 使用适配层的理由不仅是为了满足 IPv6 报文，极有可能一些应用交换（如配置或服务）产生的报文也需要少量的分片。
2. 即使上述计算显示的是最坏情况的场景，它的确显示出了事实上报文压缩几乎是不可避免的。我们希望多数（如果不是全部） IEEE 802.15.4 使用 IP 的应用会用到报头压缩，这在它 10 章定义。

译者注：

1. aMaxPHYPacketSize，aMaxFrameOverhead 是 IEEE 802.15.4 中定义的两个常量，其值分别是 127 和 25。

5 LoWPAN 适配层和帧格式

本章所说的封装格式（在后面也叫做“*LoWPAN* 封装”）是 IEEE 802.15.4 协议数据单元（PDU）的负载。这个封装头的后面是 LoWPAN 的负载（比如 IPv6 报文）。

所有通过 IEEE 802.15.4 传输的 *LoWPAN* 封装报文的前缀都来自于封装头部栈。封装头部栈中的每个头部都由头部类型和紧跟在头部类型后的零个或多个头部字段组成。对于 IPv6 的报文头部，其栈依次包含地址、逐跳选项、路由、分片、目的选项和负载 [RFC2460]；对于 LoWPAN 的报文头部，类似的序列是 mesh （L2）地址、逐跳选项（包括 L2 广播/多播）、分配和负载。以下的例子展示了在 LoWPAN 网络中可能用到的典型的报头栈结构。

一个 LoWPAN 封装的 IPv6 报文：

```
+---------------+-------------+---------+
| IPv6 Dispatch | IPv6 Header | Payload |
+---------------+-------------+---------+
```

一个 LoWPAN 封装的 LOWPAN_HC1 压缩 IPv6 报文

```
+--------------+------------+---------+
| HC1 Dispatch | HC1 Header | Payload |
+--------------+------------+---------+
```

一个 LoWPAN 封装的采用 LOWPAN_HC1 压缩的、需要 mesh 寻址的 IPv6 报文：

```
+-----------+-------------+--------------+------------+---------+
| Mesh Type | Mesh Header | HC1 Dispatch | HC1 Header | Payload |
+-----------+-------------+--------------+------------+---------+
```

一个 LoWPAN 封装的采用 LOWPAN_HC1 压缩的、需要分片的 IPv6 报文：

```
+-----------+-------------+--------------+------------+---------+
| Frag Type | Frag Header | HC1 Dispatch | HC1 Header | Payload |
+-----------+-------------+--------------+------------+---------+
```

一个 LoWPAN 封装的采用 LOWPAN_HC1 压缩的、需要 mesh 寻址的、需要分片的 IPv6 报文：

```
+-------+-------+-------+-------+---------+---------+---------+
| M Typ | M Hdr | F Typ | F Hdr | HC1 Dsp | HC1 Hdr | Payload |
+-------+-------+-------+-------+---------+---------+---------+
```

一个 LoWPAN 封装的采用 LOWPAN_HC1 压缩的、需要 mesh 寻址的、需要广播头以支持 mesh 广播/多播的 IPv6 报文：

```
+-------+-------+-------+-------+---------+---------+---------+
| M Typ | M Hdr | B Dsp | B Hdr | HC1 Dsp | HC1 Hdr | Payload |
+-------+-------+-------+-------+---------+---------+---------+
```

当一个报文里出现多个报头时它们必须以以下的顺序出现：

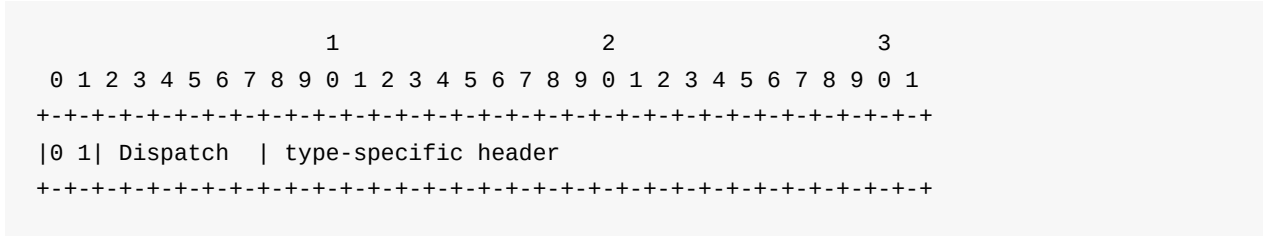
- mesh 寻址头
- 广播头
- 分片头

如上面所给出的例子中那样，所以的协议头（比如 IPv6、压缩的 IPv6 头等）都应该在一个有效的 LoWPAN 封装报头后面。这可以统一软件对报文的处理而不管传送方式是什么。

The definition of LoWPAN headers, other than mesh addressing and fragmentation, consists of the dispatch value, the definition of the header fields that follow, and their ordering constraints relative to all other headers. Although the header stack structure provides a mechanism to address future demands on the LoWPAN adaptation layer, it is not intended to provided general purpose extensibility. This format document specifies a small set of header types using the header stack for clarity, compactness, and orthogonality.

5.1 分派类型和报头

分派类型定义为第 1 位为 0 第 2 位为 1。分派类型和报头如下图所示： The dispatch type is defined by a zero bit as the first bit and a one bit as the second bit. The dispatch type and header are shown here:



- 分派（Dispatch）- 一个 6 位选择器。确定报头类型，后面是分派报头。
- 类型相关的头（type-specific header）- 一个由分派报头确定的报头

图 1. 分派类型和头部

分派值可以看作一个非正式的命名空间，我们只需要少量的符号就可以表示当前 LoWPAN 的功能。尽管将附件功能编码到分派类型中可以节省空间，但这种方法可能对未来扩展功能有所限制。

Pattern	Header Type	
00 xxxxxx	NALP	- Not a LoWPAN frame
01 000001	IPv6	- Uncompressed IPv6 Addresses
01 000010	LOWPAN_HC1	- LOWPAN_HC1 compressed IPv6
01 000011	reserved	- Reserved for future use
...	reserved	- Reserved for future use
01 001111	reserved	- Reserved for future use
01 010000	LOWPAN_BC0	- LOWPAN_BC0 broadcast
01 010001	reserved	- Reserved for future use
...	reserved	- Reserved for future use
01 111110	reserved	- Reserved for future use
01 111111	ESC	- Additional Dispatch byte follows
10 xxxxxx	MESH	- Mesh Header
11 000xxx	FRAG1	- Fragmentation Header (first)
11 001000	reserved	- Reserved for future use
...	reserved	- Reserved for future use
11 011111	reserved	- Reserved for future use
11 100xxx	FRAGN	- Fragmentation Header (subsequent)
11 101000	reserved	- Reserved for future use
...	reserved	- Reserved for future use
11 111111	reserved	- Reserved for future use

图 2. 分派值位模式

- NALP：指明接下来的比特位不是 LoWPAN 封装的一部分，所有的 LoWPAN 节点应该丢弃收到的分派值为 00xxxxxx 的报文。其它想要与 LoWPAN 节点兼容的非 LoWPAN 协议应该使 802.15.4 头部后面的第一个字节与这种模式相匹配。
- IPv6：指明接下来的头是无压缩的 IPv6 报头 [RFC2460]。
- LOWPAN_HC1：指明接下来的报头是一个 LOWPAN_HC1 压缩的 IPv6 报头。这个报头格式定义如图 9 所示。
- LOWPAN_BC0： 指明接下来的报头是一个 LOWPAN_BC0 报头，它支持 mesh 广播/多播，在 11.1 节进行描述。
- ESC：指明接下来的报头是一个用于分派值的单独的 8 位字段。它允许支持分派值大于 127。

5.2 Mesh 寻址类型和报头

前两位为 `10` 表示 Mesh 类型。 Mesh 类型和报头如下图所示：

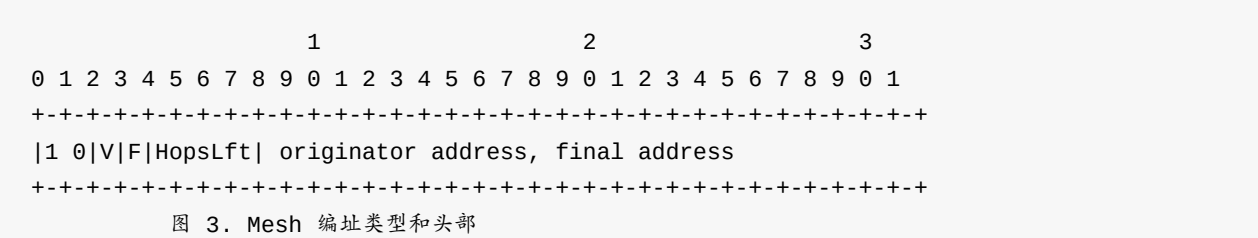


图 3. Mesh 编址类型和头部

各字段定义如下：

- V：如果源地址（最初始节点的地址）是一个 IEEE 扩展 64 位地址（ EUI－64），这位就是 0；如果是16 位短地址，这位就为 1。
- F：如果最终目标地址是一个 IEEE 扩展 64 位地址（ EUI－64），这位就是 0；如果是 16 位短地址，这位就为 1。
- 剩余跳数（Hops Left）：该字段占 4 位。转发节点将报文转发给下一跳之前将该字段减 1。如果值减至 0，报文将不会被转发。值 0xF 是保留的，表示接下来的一个字节是一个 8 位剩余跳数字段，这允许源节点指定一个大于 14 的跳数限制。
- 原始地址（Originator Address）：原始节点的链路层地址。
- 最终目的地址（Final Destination Address）：最终目的节点的链路层地址。

注意到‘ V’和‘ F’位允许 16 位和 64 位地址的混合。这至少对于允许 mesh 层“广播”是有用的，因为 802.15.4 广播地址定义的是 16 位短地址。

关于在 mesh 网路中传送帧的内容将会在第 11 节进行具体的讨论。

5.3 分片类型和报头

If an entire payload (e.g., IPv6) datagram fits within a single 802.15.4 frame, it is unfragmented and the LoWPAN encapsulation should not contain a fragmentation header. If the datagram does not fit within a single IEEE 802.15.4 frame, it SHALL be broken into link fragments. As the fragment offset can only express multiples of eight bytes, all link fragments for a datagram except the last one MUST be multiples of eight bytes in length. The first link fragment SHALL contain the first fragment header as defined below.

如果一个完整的负载（如 IPv6）数据报装在一个单独的 802.15.4 帧里，它就是不分片的，那么 LoWPAN 封装就不必包含一个分片报头。如果数据报无法封装在一个单独的 IEEE 802.15.4 帧内，它应该在链路层进行分片。因为碎片偏移量只能表示为 8 字节的倍数，所以除了最后一片外的所有分片大小都必须为 8 字节的整数倍。。第一个链路分片应该包含第一个分片报头，其定义如下图所示：

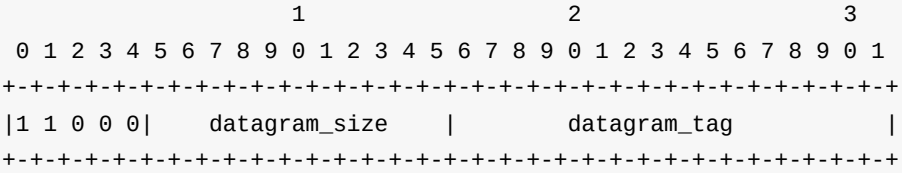


图 4. 第一个分片

The second and subsequent link fragments (up to and including thelast) SHALL contain a fragmentation header that conforms to the format shown below.

第二个和以后的链路分片（直到包含最后一个）应该包含的分片报头，如下图所示：

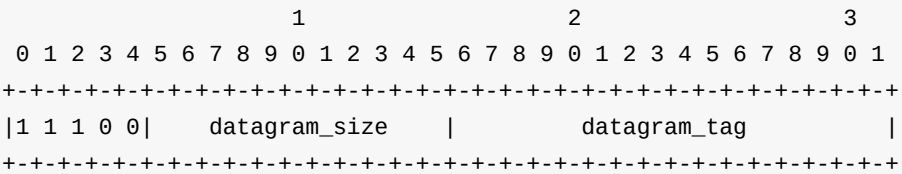


图 5. 后续的分片

- datagram_size: This 11-bit field encodes the size of the entire IP packet before link-layer fragmentation (but after IP layer fragmentation). The value of datagram_size SHALL be the same for all link-layer fragments of an IP packet. For IPv6, this SHALL be 40 octets (the size of the uncompressed IPv6 header) more than the value of Payload Length in the IPv6 header [RFC2460] of the packet. Note that this packet may already

be fragmented by hosts involved in the communication, i.e., this field needs to encode a maximum length of 1280 octets (the IEEE 802.15.4 link MTU, as defined in this document).

- 数据报大小（datagram_size）：该字段占 11 字节，表示在链路分片之前（但在 IP 层分片之后）的整个 IP 报文的大小。对于一个 IP 报文来说，所有分片的 datagram_size 的值都应该是相同的。对于 IPv6，这个值应该是比报文里 IPv6 报头[RFC2460]的负载长度多 40 字节（无压缩的 IPv6 报头的长度）。注意，这个报文可能已经由参与通信的主机进行分片了，也就是说，这个字段需要设置最大长度是 1280 字节（ IEEE 802.15.4 链路 MTU，如在本文所定义的那样）。

NOTE: This field does not need to be in every packet, as one could send it with the first fragment and elide it subsequently.However, including it in every link fragment eases the task of reassembly in the event that a second (or subsequent) link ragment arrives before the first. In this case, the guarantee of earning the datagram_size as soon as any of the fragments arrives tells the receiver how much buffer space to set aside as it waits or the rest of the fragments. The format above trades off implicity for efficiency.

注意：不是每个报文都需要这个字段。如第一个分片里含有这个字段，后面分片就可以省略该字段。然而，如果每个链路分片都包含这个字段，能简化当第二个分片（或更后面的分片）比第一个分片早达到时进行重组的过程。在这种情况下，接收端在收到任一个分片后就可以知道 datagram_size 的值，可以更早地知道需要设置多大的缓存空间。

- datagram_tag: The value of datagram_tag (datagram tag) SHALL be the same for all link fragments of a payload (e.g., IPv6) datagram.The sender SHALL increment datagram_tag for successive, fragmented datagrams. The incremented value of datagram_tag SHALL wrap from 65535 back to zero. This field is 16 bits long, and its initial value is not defined.

- 数据报标签：一个负载（比如 IPv6）数据报的所有链路分片的数据报标签的值是相同的。发送者应该对连续的分片数据报递增 datagram_tag 的值。当 datagram_tag 的值递增到大于 65535 时应该变为 0。这个字段占 16 位，它的初始值没有定义。

- datagram_offset: This field is present only in the second and subsequent link fragments and SHALL specify the offset, in increments of 8 octets, of the fragment from the beginning of the payload datagram. The first octet of the datagram (e.g., the start of the IPv6 header) has an offset of zero; the implicit value of datagram_offset in the first link fragment is zero. This field is 8 bits long.

- 数据报偏移（datagram_offset）：这个字段只出现在第二个和之后的链路分片里，表示从负载数据报开始的以 8 字节为增量的分片偏移量。 数据报的第一个字节（例如 IPv6 报头的开始）的偏移量是 0；第一个链路分片的 datagram_offset 隐含值是 0。这个字段占 8 位。

The recipient of link fragments SHALL use (1) the sender's 802.15.4 source address (or the Originator Address if a Mesh Addressing field is present), (2) the destination's 802.15.4 address (or the Final Destination address if a Mesh Addressing field is present), (3) datagram_size, and (4) datagram_tag to identify all the link fragments that belong to a given datagram.

链路分片的接收者应该使用（1）发送者的 802.15.4 源地址（或 Originator Address 如果出现了 Mesh Addressing 字段），（2）目标的 802.15.4 地址（或 Final Destination Address 如果出现了Mesh Addressing 字段），（3）datagram_size 和（4）datagram_tag 来确定一个给定数据报的所有链路分片。

Upon receipt of a link fragment, the recipient starts constructing the original unfragmented packet whose size is datagram_size. It uses the datagram_offset field to determine the location of the individual fragments within the original unfragmented packet. For example, it may place the data payload (except the encapsulation header) within a payload datagram reassembly buffer at the location specified by datagram_offset. The size of the reassembly buffer SHALL be determined from datagram_size.

在收到一个链路分片后，接收者开始构造大小为 datagram_size 的未分片的原始报文。它使用 datagram_offset 字段来确定每一个分片在原始报文中的位置。例如，它把数据载荷（除了封装报头）放在一个位置由 datagram_offset 确定的载荷数据报重组缓存里。重组缓存的大小应该由 datagram_size 来确定。

If a link fragment that overlaps another fragment is received, as identified above, and differs in either the size or datagram_offset of the overlapped fragment, the fragment(s) already accumulated in the reassembly buffer SHALL be discarded. A fresh reassembly may be commenced with the most recently received link fragment. Fragment overlap is determined by the combination of datagram_offset from the encapsulation header and "Frame Length" from the 802.15.4 Physical Layer Protocol Data Unit (PPDU) packet header.

如果收到与另一个分片重叠的链路分片，如上面所确定的，并且与重叠的分片的大小和 datagram_offset 都不同，那么已经累积在重组缓存里的分片应该要丢弃。一个新的重组可能从最近接收到的链路分片开始。分片重叠是由封装报头的 datagram_offset 和 802.15.4 物理层协议数据单元（ PPDU）报文头部的“帧长”组合来确定的。

Upon detection of a IEEE 802.15.4 Disassociation event, fragment recipients MUST discard all link fragments of all partiall reassembled payload datagrams, and fragment senders MUST discard all not yet transmitted link fragments of all partially transmitted payload (e.g., IPv6) datagrams. Similarly, when a node first receives a fragment with a given datagram_tag, it starts a reassembly timer. When this time expires, if the entire packet has not been reassembled, the existing fragments MUST be discarded and the reassembly state MUST be flushed. The reassembly timeout MUST be set to a maximum of 60 seconds (this is also the timeout in the IPv6 reassembly procedure [RFC2460]).

在检测到一个 IEEE 802.15.4 分裂事件时， 分片接收者必须丢弃所有部分重组的载荷数据报所有的链路分片，分片发送者必须丢弃部分发送的载荷（如 IPv6）数据报的未发送的链路分片。类似地，当一个节点接收到第一个给定 datagram_tag 的分片，它开始一个重组定时器。当定时时间到，如果整个报文没有重组完成，现有的分片必须丢弃，并且重组状态必须重设。重组超时的最大时间必须为 60 秒（这也是 IPv6 重组程序的最大超时[RFC2460]）。

6 无状态地址自动配置

This section defines how to obtain an IPv6 interface identifier.

本章定义如何获得 IPv6 接口标识符。

The Interface Identifier [RFC4291] for an IEEE 802.15.4 interface may be based on the EUI-64 identifier [EUI64] assigned to the IEEE 802.15.4 device. In this case, the Interface Identifier is formed from the EUI-64 according to the "IPv6 over Ethernet" specification [RFC2464].

IEEE 802.15.4 接口的接口标识符 [RFC4291] 可能是基于分配给 IEEE 802.15.4 设备的 EUI-64 标识符 [EUI64] 生成的。在这种情况下，先按照“以太网上的 IPv6 ”规范 [RFC2464] 生成一个 EUI-64，然后根据这个 EUI-64 构成一个接口标识符。

All 802.15.4 devices have an IEEE EUI-64 address, but 16-bit short addresses (Section 3 and Section 12) are also possible. In these cases, a "pseudo 48-bit address" is formed as follows. First, the left-most 32 bits are formed by concatenating 16 zero bits to the 16-bit PAN ID (alternatively, if no PAN ID is known, 16 zero bits may be used). This produces a 32-bit field as follows:

- `16_bit_PAN:16_zero_bits`

Then, these 32 bits are concatenated with the 16-bit short address.This produces a 48-bit address as follows:

- `32_bits_as_specified_previously:16_bit_short_address`

The interface identifier is formed from this 48-bit address as per the "IPv6 over Ethernet" specification [RFC2464]. However, in the resultant interface identifier, the "Universal/Local" (U/L) bit SHALL be set to zero in keeping with the fact that this is not a globally unique value. For either address format, all zero addresses MUST NOT be used.

所有的 802.15.4 设备都有一个 IEEE EUI－64 地址，同时也可能有一个 16 位短地址（第 3 章和第 12 章）。在这种情况下，一个“假的 48 位地址”的产生过程如下：

首先，左 32 位由 16 位 0 和 16 位 PAN ID（可选，如果不知道 PAN ID， 就用 16 位 0）组成。这样产生的 32 位字段如下所示：

- `16 位 PAN ID : 16 位 0`

然后，将这这生成的 32 位和 16 位短地址串接在一起，就产生了 48 位地址：

- `如上所示的 32 位 : 16 位短地址`

由这 48 位则构成了类似于“以太网上的 IPv6”规范 [RFC2464] 中规定的接口标识符。不过，这样形成的接口标识符并不是全球唯一的，所以要将“全局/本地”（U/L）位置为 0。

A different MAC address set manually or by software MAY be used to derive the Interface Identifier. If such a MAC address is used, its global uniqueness property should be reflected in the value of the U/L bit.

也可以使用由人工或者软件设置的 MAC 地址来构成接口标识符。在这种情形下，它的全局唯一性应该反映到“全局/本地”（U/L）位上。

An IPv6 address prefix used for stateless autoconfiguration [RFC4862] of an IEEE 802.15.4 interface MUST have a length of 64 bits.

IEEE 802.15.4 接口上用于无状态自动配置[RFC4862]的 IPv6 地址前缀长度必须是 64 位。

7 IPv6 链路本地地址

The IPv6 link-local address [RFC4291] for an IEEE 802.15.4 interface is formed by appending the Interface Identifier, as defined above, to the prefix FE80::/64.

前缀 FE80::/64 与 IEEE 802.15.4 接口标识符共同构成 IPv6 链路本地地址[RFC4291]，如下所示:

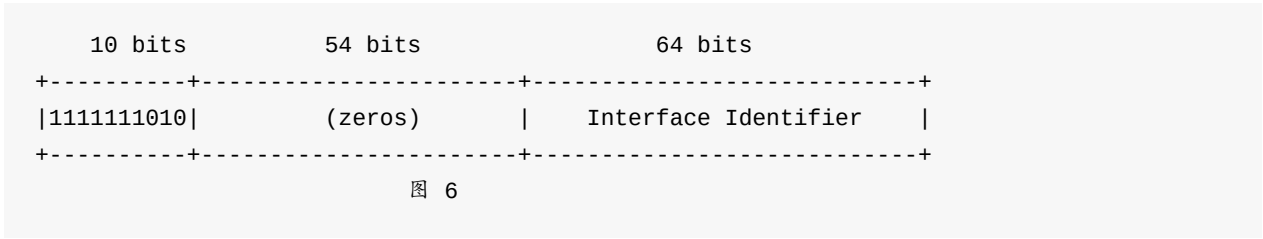


图 6

8 单播地址映射

The address resolution procedure for mapping IPv6 non-multicast addresses into IEEE 802.15.4 link-layer addresses follows the general description in Section 7.2 of [RFC4861], unless otherwise specified.

除非特别说明，用于将 IPv6 非多播地址映射到 IEEE 802.15.4 链路层地址的地址解析程序（ARP）遵循 [RFC4861] 7.2 节的通用描述。

The Source/Target Link-layer Address option has the following forms when the link layer is IEEE 802.15.4 and the addresses are EUI-64 or 16-bit short addresses, respectively.

当链路层是 IEEE 802.15.4，地址是 EUI-64 或 16 位短地址时，源/目标链路层地址选项分别有以下的形式：

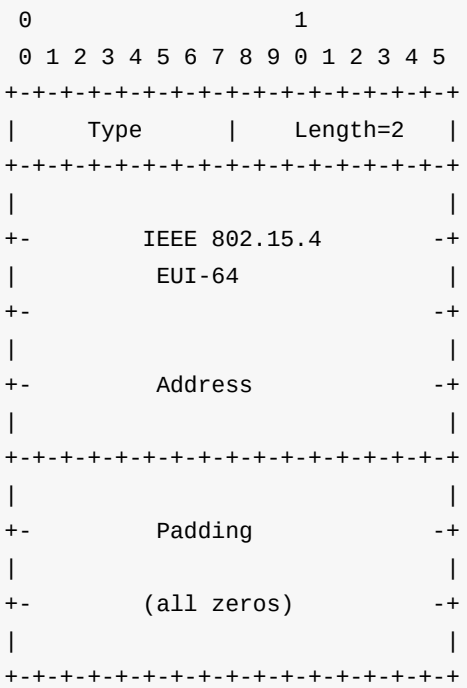


图 7

Option fields:

- Type:

  1: for Source Link-layer address.

  2: for Target Link-layer address.

- Length: This is the length of this option (including the type and length fields) in units of 8 octets. The value of this field is 2 if using EUI-64 addresses, or 1 if using 16-bit short addresses.

- IEEE 802.15.4 Address: The 64-bit IEEE 802.15.4 address, or the 16-bit short address (as per the format in Section 9), in canonical bit order. This is the address the interface currently responds to. This address may be different from the built-in address used to

derive the Interface Identifier, because of privacy or security(e.g., of neighbor discovery) considerations.

可选字段：

- 类型（Type）：

  1：源链路层地址。

  2：目的链路层地址。

- 长度（Length）：这是这个选项（包含 type 和 length 字段）的长度，单位是字节。如果使用 EUI-64 地址，这个字段的值为 2，如果使用 16 位短地址，这个字段的值为 1。
- IEEE 802.15.4 地址：64 位的 IEEE 802.15.4 地址，或 16 位短地址（如第 9 章所描述的格式），使用正则位顺序。这是接口当年可响应的地址。缘于隐私或安全（如邻居发现）的考虑，这个地址可能与用于产生接口标识符的内建的地址不同。

9 多播地址映射

The functionality in this section MUST only be used in a mesh-enabled LoWPAN. An IPv6 packet with a multicast destination address (DST),consisting of the sixteen octets DST[1] through DST[16], is transmitted to the following 802.15.4 16-bit multicast address:

这一节里的功能必须在一个 mesh 使能的 LoWPAN 里使用。一个有多播目标地址（DST）的 IPv6 报文，，将会传输到以下 802.15.4 的 16 位多播地址：

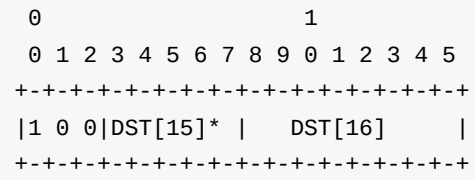


图 8

Here, DST[15]* refers to the last 5 bits in octet DST[15], that is, bits 3-7 within DST[15]. The initial 3-bit pattern of "100" follows the 16-bit address format for multicast addresses (Section 12).

这里， DST[15]*表示 DST[15]的后 5 位，那就是， DST[15]里的 3－7 位。开始的 3 位序列“100”紧跟在 16 位短地址格式后面用于多播地址（节 12）。

This allows for multicast support within 6LoWPAN networks, but the full specification of such support is out of the scope of this document. Example mechanisms are: flooding, controlled flooding, unicasting to the PAN coordinator, etc. It is expected that this would be specified by the different mesh routing mechanisms.

这允许在 6LoWPAN 网路中支持多播，但具体的多播方式不在本文的描述范围内。机制示例是：泛洪、受控泛洪、单播到 PAN 协调器等。这应该由不同的 mesh 路由机制来描述。

10 报头压缩

There is much published and in-progress standardization work on header compression. Nevertheless, header compression for IPv6 over IEEE 802.15.4 has differing constraints summarized as follows:

- Existing work assumes that there are many flows between any two devices. Here, we assume a very simple and low-context flavor ofheader compression. Whereas this works independently of flows (potentially several), it does not use any context specific to any flow. Thus, it cannot achieve as much compression as schemes that build a separate context for each flow to be compressed.
- Given the very limited packet sizes, it is highly desirable to integrate layer 2 with layer 3 compression, something traditionally not done (although now changing due to the ROHC (RObust Header Compression) working group).
- It is expected that IEEE 802.15.4 devices will be deployed in multi-hop networks. However, header compression in a mesh departs from the usual point-to-point link scenario in which the compressor and decompressor are in direct and exclusive communication with each other. In an IEEE 802.15.4 network, it is highly desirable for a device to be able to send header compressed packets via any of its neighbors, with as little preliminary context-building as possible.

尽管存在很多已被发行和正在制定的头部压缩标准，但是我们可以将对运行在 IEEE 802.15.4 之上的 IPv6 进行头部压缩时面临的各种限制总结如下：

- 现有标准都假设在任何两个设备之间存在很多流量。我们假设存在一个简单的、与上下文不太相关的报头压缩。这种压缩与流量无关，不使用与任何流量相关的任何上下文，因此其压缩效果没有为每个流量构建独立上下文的效果好。
- 由于报文尺寸极其有限，所以非常有必要对第 2 层和第 3 层进行压缩。这是传统压缩没有做的事（尽管现在有了 ROHC（ RObust 报头压缩）工作组而正在改变）。
- 尽管 IEEE 802.15.4 设备应该部署在多跳网路中，但是在传统的点对点链路场景下，压缩者和解压缩者是直接与对方通信并独占链路的。在 IEEE 802.15.4 网络中，设备最好能（使用尽量少的初始化工作）通过它的任何邻居发送头部压缩过的报文。

Any new packet formats required by header compression reuse the basic packet formats defined in Section 5 by using different dispatch values.

通过使用不同的分派值，报头压缩所需要的任何新报文格式都可以重用第 5 章中所定义的基本报文格式。

Header compression may result in alignment not falling on an octet boundary. Since hardware typically cannot transmit data in units less than an octet, padding must be used. Padding is done as follows: First, the entire series of contiguous compressed headers is laid out (this document only defines IPv6 and UDP header compression schemes, but others may be defined elsewhere). Then, zero bits SHOULD be added as appropriate to align to an octet boundary. This counteracts any potential misalignment caused by header compression, so subsequent fields (e.g., non-compressed headers or data payloads) start on an octet boundary and follow as usual.

报头压缩可能导致不能字节边界对齐。由于硬件通常不能传输小于一个字节的数据单元，所以必须使用填充位。填充的步骤：首先，将所有连续的压缩报头按位排序（本文只定义了 IPv6 和 UDP 报头压缩机制，但其它的可能在别处定义）。然后，用若干位 0 进行填以与字节边界对齐。这消除了所有由报头压缩产生的不对齐，所以后续字段（例如非压缩报头、数据载荷）就从正常的字节边界开始了。

10.1 IPv6 报头字段压缩

By virtue of having joined the same 6LoWPAN network, devices share some state. This makes it possible to compress headers without explicitly building any compression context state. Therefore, 6LoWPAN header compression does not keep any flow state; instead, it relies on information pertaining to the entire link. The following IPv6 header values are expected to be common on 6LoWPAN networks, so the HC1 header has been constructed to efficiently compress them from the onset: Version is IPv6; both IPv6 source and destination addresses are link local; the IPv6 interface identifiers (bottom 64 bits) for the source or destination addresses can be inferred from the layer two source and destination addresses (of course, this is only possible for interface identifiers derived from an underlying 802.15.4 MAC address); the packet length can be inferred either from layer two ("Frame Length" in the IEEE 802.15.4 PPDU) or from the "datagram_size" field in the fragment header (if present); both the Traffic Class and the Flow Label are zero; and the Next Header is UDP, ICMP or TCP. The only field in the IPv6 header that always needs to be carried in full is the Hop Limit (8 bits). Depending on how closely the packet matches this common case, different fields may not be compressible thus needing to be carried "in-line" as well (Section 10.3.1). This common IPv6 header (as mentioned above) can be compressed to 2 octets (1 octet for the HC1 encoding and 1 octet for the Hop Limit), instead of 40 octets. Such a packet is compressible via the LOWPAN_HC1 format by using a Dispatch value of LOWPAN_HC1 followed by a LOWPAN_HC1 header "HC1 encoding" field (8 bits) to encode the different combinations as shown below. This header may be preceded by a fragmentation header, which may be preceded by a mesh header.

由于加入到同一个 6LoWPAN 网络，所以设备间可以共享一些状态。这使得在压缩报头时可以不用明确创建任何内容状态。因此，6LoWPAN 报头压缩并不保留任何流量状态，反而依赖于与整个链路相关的信息。以下的 IPv6 报头值对于 6LoWPAN 网络是通用的，所以 HC1 报头从一开始用于高效压缩：

- 版本是 IPv6；
- IPv6 源和目标地址都是链路本地地址；
- 源地址和目标地址的 IPv6 接口标识符（后 64 位）可以从第二层的源和目标地址（当然，这是只能由一个 802.15.4 MAC 地址来产生接口标识符）推断出；
- 报文长度可以从第二层（ IEEE 802.15.4 PPDU 里的“帧长”）或分片报头（如果有）的“datagram_size”字段来获得；
- 业务类型字段和流量标签字段都是 0；
- 下一个报头是 UDP， ICMP 或 TCP；

IPv6 头部字段中唯一一个不需要压缩的字段是跳数限制字段（8 位）。依赖于报文与这种通用情形的匹配程序，一些字段可能不需要压缩而直接内嵌传输（第 10.3.1 节）。通用 IPv6 头部（如上面提到的）可以被压缩到 2 字节（1 字节的 HC1 编码和 1 字节的跳数限制），而不是 40 字节。这样的报文可通过 LOWPAN_HC1 格式来进行压缩，即通过使用 LOWPAN_HC1 的分派值以及后面的 LOWPAN_HC1 报头里“ HC1 编码”字段（8 位）来编码不同的组合，如下图所示。这个报头前面可能是一个分片报头，再前面可能是一个 mesh 报头。

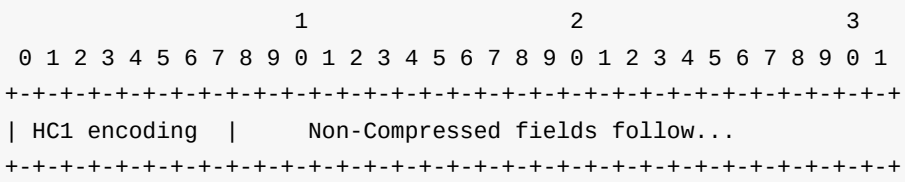


图 9: LOWPAN_HC1 (通用压缩报头编码)

As can be seen below (bit 7), an HC2 encoding may follow an HC1 octet. In this case, the non-compressed fields follow the HC2 encoding field (Section 10.3).

如下面所示（比特 7），一个 HC2 编码后面可能是一个 HC1 字节。在这种情况下，HC2 编码字段后面是一个非压缩字段（节 10.3）。

The address fields encoded by "HC1 encoding" are interpreted as follows:

- PI: Prefix carried in-line (Section 10.3.1).
- PC: Prefix compressed (link-local prefix assumed).
- II: Interface identifier carried in-line (Section 10.3.1).
- IC: Interface identifier elided (derivable from the corresponding link-layer address). If applied to the interface identifier of either the source or destination address when routing in a mesh (Section 11), the corresponding link-layer address is that found in the "Mesh Addressing" field (Section 5.2).

由“ HC1 编码”产生的地址字段解释如下所示：

- PI：内嵌的前缀(第 10.3.1 节)。
- PC：压缩的前缀（ 假定是链路本地前缀）。
- II：内嵌的接口标识符（第 10.3.1 节）。
- IC：省略的接口标识符（从相应的链路层地址获得的）。如果在一个 mesh 路由中使用源或目标地址的接口标识符（第 11 章），相应的链路层地址是“Mesh Addressing”字段里的地址（第 5.2 节）。

The "HC1 encoding" is shown below (starting with bit 0 and ending at bit 7):

- IPv6 source address (bits 0 and 1):
  - 00: PI, II

    - 01: PI, IC
    - 10: PC, II
    - 11: PC, IC
- IPv6 destination address (bits 2 and 3):
    - 00: PI, II
    - 01: PI, IC
    - 10: PC, II
    - 11: PC, IC
- Traffic Class and Flow Label (bit 4):
    - 0: not compressed; full 8 bits for Traffic Class and 20 bits for Flow Label are sent
    - 1: Traffic Class and Flow Label are zero
- Next Header (bits 5 and 6):
    - 00: not compressed; full 8 bits are sent
    - 01: UDP
    - 10: ICMP
    - 11: TCP
- HC2 encoding(bit 7):
    - 0: No more header compression bits
    - 1: HC1 encoding immediately followed by more header compression bits per HC2 encoding format. Bits 5 and 6 determine which of the possible HC2 encodings apply (e.g., UDP, ICMP, or TCP encodings).

“HC1 编码”如下所示（比特 0 到比特 7）：

- IPv6 源地址（比特 0 和 1）：
    - 00： PI， II
    - 01： PI， IC
    - 10： PC， II
    - 11： PC， IC
- IPv6 目标地址（比特 2 和 3）：
    - 00： PI， II
    - 01： PI， IC
    - 10： PC， II
    - 11： PC， IC
- 交通等级和流量标签（比特 4）：
    - 0：非压缩的；全 8 位的交通等级和 20 位的流量标签都会发送
    - 1：交通等级和流量标签都为 0
- 下一报头（比特 5 和 6）：
    - 00：非压缩的；全 8 位都会发送
    - 01： UDP
    - 10： ICMP

    - 11： TCP
- HC2 编码（比特 7）：
    - 0：没有更多的报头压缩位
    - 1： HC1 编码后面有更多 HC2 编码格式的报头压缩位。位 5 和 6 确定了 HC2 编码的用途（例如，UDP，ICMP 或 TCP 编码）。

10.2 UDP 报头字段编码

Bits 5 and 6 of the LOWPAN_HC1 allows compressing the Next Header field in the IPv6 header (for UDP, TCP, and ICMP). Further compression of each of these protocol headers is also possible. This section explains how the UDP header itself may be compressed. The HC2 encoding in this section is the HC_UDP encoding, and it only applies if bits 5 and 6 in HC1 indicate that the protocol that follows the IPv6 header is UDP. The HC_UDP encoding (Figure 10) allows compressing the following fields in the UDP header: source port, destination port, and length. The UDP header's checksum field is not compressed and is therefore carried in full. The scheme defined below allows compressing the UDP header to 4 octets instead of the original 8 octets.

LOWPAN_CH1 的比特 5 和比特 6 允许对 IPv6 头部（UDP,TCP 和 ICMP）中的下一个头部字段进行压缩。对这些协议的头部进行进一步压缩也是可能的，但是在本节主要解释 UDP 头部本身是如何进行压缩的。在本节中，LOWPAN_CH2 编码就是 HC_UDP 编码，且只在 HC1 中的比特 5 和比特 6 表示 IPv6 头部后的协议是 UDP 时才有效。HC_UDP 编码（图 10）允许压缩 UDP 头部中的如下字段：源端口、目的端口和长度。UDP 头部的检验和字段不能被压缩，因此它必须被完全传输。下面定义的机制可以让 UDP 的头部由原始的 8 字节压缩到 4 字节。

The only UDP header field whose value may be deduced from information available elsewhere is the Length. The other fields must be carried in-line either in full or in a partially compressed manner (Section 10.3.2).

在 UDP 头部中，唯一一个能出其它地方推断出的字段是长度字段。其它的字段必须被完全嵌入或者部分压缩的方式嵌入。

```
                    1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|HC_UDP encoding|     Fields carried in-line follow...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

图 10: HC_UDP (UDP 通用头部压缩编码)

The "HC_UDP encoding" for UDP is shown below (starting with bit 0 and ending at bit 7):

- UDP source port (bit 0):
  - 0: Not compressed, carried "in-line" (Section 10.3.2)
  - 1: Compressed to 4 bits. The actual 16-bit source port is obtained by calculating: P

+ short_port value. The value of P is the number 61616 (0xF0B0). The short_port is expressed as a 4-bit value which is carried "in-line" (Section 10.3.2)

- UDP destination port (bit 1):
  - 0: Not compressed, carried "in-line" (Section 10.3.2)
  - 1: Compressed to 4 bits. The actual 16-bit destination port is obtained by calculating: P + short_port value. The value of P is the number 61616 (0xF0B0). The short_port is expressed as a 4-bit value which is carried "in-line" (Section 10.3.2)
- Length (bit 2):
  - 0: not compressed, carried "in-line" (Section 10.3.2)
  - 1: compressed, length computed from IPv6 header length information. The value of the UDP length field is equal to the Payload Length from the IPv6 header, minus the length of any extension headers present between the IPv6 header and the UDP header.
- (bit 3 through 7)

UDP 的“HC_UDP”编码如下所示（从比特 0 到比特 7）：

- UDP 源端口（比特 0）：
  - 0: 非压缩的，“内嵌”运载（第 10.3.2 节）。
  - 1: 压缩到 4 位。实际的 16 位源端口通过计算得到：P + short_port。P 的值是 61616（ 0xF0B0）。 short_port 是一个“内嵌传输的 4 位值（第 10.3.2 节）。
- UDP 目标端口（比特 1）：
  - 0: 非压缩的，“内嵌”运载（第 10.3.2 节）。
  - 1: 压缩到 4 位。实际的 16 位目的端口通过计算得到：P + short_port。P 的值是 61616（ 0xF0B0）。 short_port 是一个“内嵌传输的 4 位值（第 10.3.2 节）。
- 长度（比特 2）：
  - 0: 非压缩的，“内嵌”运载（第 10.3.2 节）.
  - 1: 压缩的，长度从 IPv6 报头长度信息来计算而来。UDP 长度字段的值等于 IPv6 报头的载荷长度减去出现在 IPv6 报头和 UDP 报头之间的任何扩展报头的长度。
- 保留（比特 3 到 7）

10.3 非压缩字段

10.3.1 IPv6 非压缩字段

This scheme allows the IPv6 header to be compressed to different degrees. Hence, instead of the entire (standard) IPv6 header, only non-compressed fields need to be sent. The subsequent header (as specified by the Next Header field in the original IPv6 header) immediately follows the IPv6 non-compressed fields.

Uncompressed IPv6 addressing is described by a dispatch type containing an IPv6 dispatch value followed by the uncompressed IPv6 header. This dispatch type may be preceded by additional LoWPAN headers.

The non-compressed IPv6 field that MUST be always present is the Hop Limit (8 bits). This field MUST always follow the encoding fields (e.g., "HC1 encoding" as shown in Figure 9), perhaps including other future encoding fields). Other non-compressed fields MUST follow the Hop Limit as implied by the "HC1 encoding" in the exact same order as shown above (Section 10.1): source address prefix (64 bits) and/or interface identifier (64 bits), destination address prefix (64 bits) and/or interface identifier (64 bits), Traffic Class (8 bits), Flow Label (20 bits) and Next Header (8 bits). The actual next header (e.g., UDP, TCP, ICMP, etc) follows the non-compressed fields.

10.3.2 UDP 非压缩字段

This scheme allows the UDP header to be compressed to different degrees. Hence, instead of the entire (standard) UDP header, only non-compressed or partially compressed fields need to be sent.

The non-compressed or partially compressed fields in the UDP header MUST always follow the IPv6 header and any of its associated in-line fields. Any UDP header in-line fields present MUST appear in the same order as the corresponding fields appear in a normal UDP header [RFC0768], e.g., source port, destination port, length, and checksum. If either the source or destination ports are in "short_port" notation (as indicated in the compressed UDP header), then instead of taking 16 bits, the inline port numbers take 4 bits.

15 参考文献

15.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2434] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 2434, October 1998.
- [RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.
- [RFC2464] Crawford, M., "Transmission of IPv6 Packets over Ethernet Networks", RFC 2464, December 1998.
- [RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, February 2006.
- [RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, September 2007.
- [RFC4862] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", RFC 4862, September 2007.
- [ieee802.15.4] IEEE Computer Society, "IEEE Std. 802.15.4-2003", October 2003.

15.2. Informative References

- [EUI64] "GUIDELINES FOR 64-BIT GLOBAL IDENTIFIER (EUI-64) REGISTRATION AUTHORITY", IEEE http://standards.ieee.org/regauth/oui/tutorials/EUI64.html.
- [KW03] Karlof, Chris and Wagner, David, "Secure Routing in Sensor Networks: Attacks and Countermeasures", Elsevier’s AdHoc Networks Journal, Special Issue on Sensor Network Applications and Protocols vol 1, issues 2-3, September 2003.
- [RFC0768] Postel, J., "User Datagram Protocol", STD 6, RFC 768, August 1980.
- [RFC3756] Nikander, P., Kempf, J., and E. Nordmark, "IPv6 Neighbor Discovery (ND) Trust Models and Threats", RFC 3756, May 2004.
- [RFC3819] Karn, P., Bormann, C., Fairhurst, G., Grossman, D., Ludwig, R., Mahdavi, J., Montenegro, G., Touch, J., and L. Wood, "Advice for Internet Subnetwork Designers", BCP 89, RFC 3819, July 2004.