6LoWPAN: Incorporating IEEE 802.15.4 into the IP architecture

Internet Protocol for Smart Objects (IPSO) Alliance

White paper # 3

Jonathan Hui, PhD, Arch Rock Corporation

David Culler, PhD, University of California, Berkeley

Samita Chakrabarti, IP Infusion

January 2009

Executive Summary

IP for Smart Objects seeks to extend the use of IP networking into resource-constrained devices over a wide range of low-power link technologies – IEEE 802.15.4 represents one such link. Extending IP to low-power, wireless personal area networks (LoWPANs) was once considered impractical because these networks are highly constrained and must operate unattended for multiyear lifetimes on modest batteries. Many vendors embraced proprietary protocols, assuming that IP was too resource-intensive to be scaled down to operate on the microcontrollers and low-power wireless links used in LoWPAN settings. However, 6LoWPAN radically alters the calculation by introducing an adaptation layer that enables efficient IPv6 communication over IEEE 802.15.4 LoWPAN links.

Introduction

Several leading radio manufacturers have implemented IEEE 802.15.4, which specifies a wireless link for low-power personal area networks (LoWPANs). 802.15.4 is widely used in embedded applications, such as environmental monitoring to improve agricultural yields, structural monitoring to track building and bridge integrity, industrial control to provide more sense points and control points at lower cost, . These applications generally require numerous low-cost nodes communicating over multiple hops to cover a large geographical area, and they must operate unattended for years on modest batteries. Such requirements target a very different set of applications than do WPAN technologies such as Bluetooth, which eliminate wiring for headsets, game controllers, and personal devices. Accordingly, 802.15.4's capabilities are more limited than other WPANs and WLANs – they have small frame sizes, low bandwidth, and low transmit power. Additionally, the microcontrollers typically coupled with LoWPAN radios have limited memory and compute power. These constraints led many LoWPAN vendors to embrace proprietary protocols and link-only solutions (such as ZigBee), presuming that IP was too memory- and bandwidth-intensive for them to scale it down as necessary. While not following the IP standard, many of these technologies still have not proven their effectiveness in constrained environments.

6LoWPAN radically alters the landscape by introducing an adaptation layer between the IP stack's link and network layers to enable efficient transmission of IPv6 datagrams over 802.15.4 links, dramatically reducing IP overhead [3]. The adaptation layer is an IETF proposed standard and provides header compression to reduce transmission overhead, fragmentation to support the IPv6 minimum MTU requirement, and support for layer-two forwarding to deliver and IPv6 datagram over multiple radio hops [4]. 6LoWPAN achieves low overhead by apply cross-layer optimizations; it uses information in the link and adaptation layers to compress network- and transport-layer headers. Drawing on IPv6 extension headers, it employs the *header stacking* principle to separate the orthogonal concepts and keep the header small and easy to parse.

Here, we discuss key 6LoWPAN concepts to demonstrate how it enables efficient support for IPv6 over 802.15.4 links.

6LoWPAN Network Architecture

By communicating natively with IP, LoWPAN networks are connected to other IP networks simply by using IP routers. As shown in Figure 1, LoWPANs will typically operate on the edge, acting as stub networks. The LoWPAN may be connected to other IP networks through one or more *border routers* that forward IP datagrams between different media. Connectivity to other IP networks may be provided through any arbitrary link, including Ethernet, Wi-Fi, GPRS, or satellite. Because 6LoWPAN only specifies operation of IPv6 over IEEE 802.15.4, border routers may also implement Stateless IP/ICMP Translation [6] or other IPv6 transition mechanisms to connect 6LoWPAN networks to IPv4 networks [7]. These IPv6 transition mechanisms do not require 6LoWPAN nodes to implement IPv4 in whole or in part.

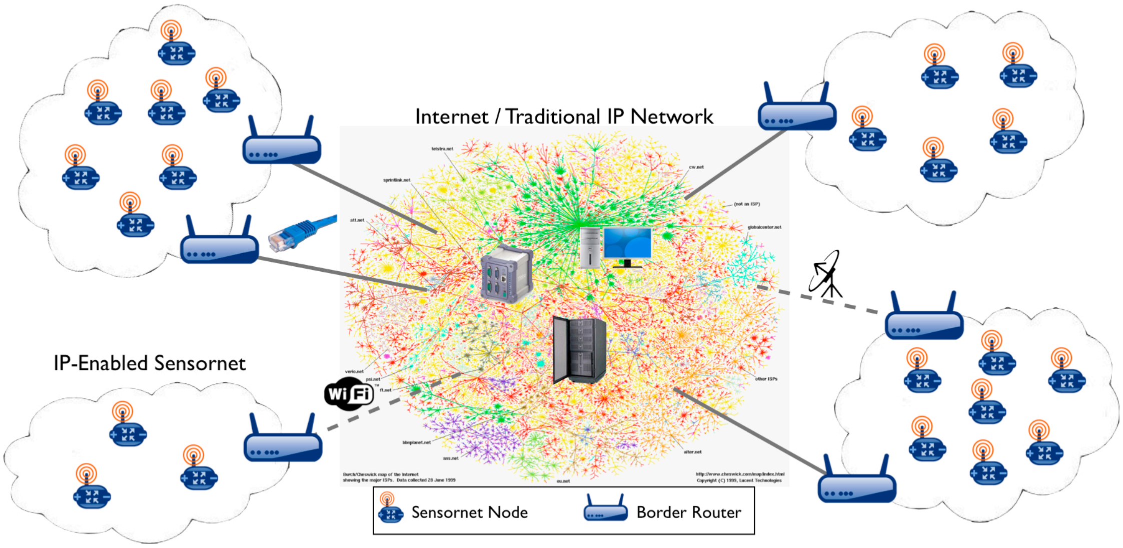
Internet / Traditional IP Network
telstra.net
cw.net
sprintlink.net
(not an ISP)
att.net
globalcenter.net
IP-Enabled Sensornet
verio.net
psi.net
Wi Fi
ft.net
other ISPs
bbnplanet.net
ans.net
alter.net
Burch/Cheswick map of the Internet
showing the major ISPs. Data collected 28 June 1999
eu.net
http://www.cheswick.com/map/index.html
Copyright (C) 1999, Lucent Technologies
Sensornet Node
Border Router

Figure 1. Extending the Internet Architecture.

Because border routers forward datagrams at the network layer, they do not maintain any application-layer state. Other ad-hoc network architecture (such as ZigBee) require stateful and complex application gateways to connect LoWPANs to other networks. These application gateways must understand any application profiles that may be used in the LoWPAN, and any changes to application protocols on the wireless nodes must also be accompanied by changes on the gateway [1]. In contrast, IP-based border routers remain agnostic to application protocols used in the LoWPAN. Note, however, that the IP architecture does not preclude the use of proxies and caches to optimize network performance, both of which are widely used in the Internet today.

IPv6 over IEEE 802.15.4

The IPv6 protocol is designed as the successor to IPv4 and enables the Internet to scale for decades to come. To overcome dwindling unallocated address space – and in anticipation that networked appliances and instruments will vastly outnumber conventional computer hosts – IPv6 expands the IP address space from 32 to 128 bits. Recognizing the growth in link bandwidth, IPv6 increases the minimum MTU requirement from 576 to 1,280 bytes. To simplify routers and increase performance, IPv6 implements fragmentation at the endpoints, rather than in intermediate routers. To increase protocol efficiency and eliminate the need for ad hoc link-level services to bootstrap a subnet, IPv6 includes scoped multicast as an integral part of its architecture. Core IPv6 components, such as Neighbor Discovery (ND), use link-local scoped multicast for address resolution, duplicate address detection (DAD), and router discovery. Stateless address autoconfiguration (SAA) simplifies configuration and management of IPv6 devices by enabling nodes to assign themselves meaningful addresses.

IPv6 also reflects the advances in link technologies the Internet uses. Ethernet has prevailed as the dominant link, and its throughput has increased at an extraordinary rate.

Current WLAN technologies, such as Wi-Fi, mirror Ethernet capabilities by supporting similarly sized MTUs and high link rates. Both links operate in the context of ample power and highly capable devices. WPAN technologies, on the other hand, operate with lower power. IEEE 802.15.4 was designed specifically for long-lived application domains that require numerous low-cost nodes, and these constraints limit the capability of LoWPAN links and the microcontrollers to which they're attached. Throughput is limited to 250 kbps. The frame length is limited to 128 bytes to ensure reasonably low packet error rates when bit-error rates are non-negligible and reflects microcontrollers' limited buffering capabilities. 802.15.4 defines short 16-bit link addresses, in addition to IEEE EUI-64 addresses, to reduce header overhead and memory requirements. Communication range is short (tens of meters) because transmission power increases polynomially with range. Unlike most typical WPAN and WLAN installations, LoWPANs communicate over multiple hops. Finally, the associated microcontrollers typically have about 8 Kbytes of data RAM and 64 Kbytes of program ROM.

Due to these resource constraints and LoWPANs' multihop nature, supporting IPv6 over LoWPAN networks presents several challenges. First, IPv6 datagrams aren't a natural fit for LoWPANs. Low throughput, limited buffering, and frames that are one-tenth the size of the IPv6 minimum MTU requirement make datagram fragmentation and compression a necessity for efficient operation. For example, link headers can limit effective link payload to 81 bytes, making the IPv6 (40 bytes), UDP (8 bytes), and TCP (20 bytes) headers seem exceedingly large. Second, because 802.15.4 is both low-power and low-throughput, it's more prone to spurious interference, link failures, dynamic link qualities, and asymmetric links. Such characteristics require the network layer to be responsive and adaptive while remaining energy efficient, and they affect all aspects of networking, including fragmentation, compression, forwarding, and routing. Third, a LoWPAN's expected topology is a mesh of short-range connections. This negates the assumption that the link is a single broadcast domain on which a core of IP architectural components – such as IPv6 ND and SAA – relies. The IETF 6LoWPAN working group addressed these issues with RFC 4944 [4]. In the remainder of this article, we provide a basic overview of RFC 4944 and touch on the issues that remain to be addressed.

6LoWPAN Adaptation Layer

The 6LoWPAN format defines how IPv6 communication is carried in 802.15.4 frames and specifies the adaptation layer's key elements. 6LoWPAN has three primary elements:

- *Header compression*. IPv6 header fields are compressed by assuming usage of common values. Header fields are elided from a packet when the adaptation layer can derive them from link-level information carried in the 802.15.4 frame or based on simple assumptions of shared context.
- *Fragmentation*. IPv6 packets are fragmented into multiple link-level frames to accommodate the IPv6 minimum MTU requirement.
- *Layer-two forwarding*. To support layer-two forwarding of IPv6 datagrams, the adaptation layer can carry link-level addresses for the ends of an IP hop.

Alternatively, the IP stack might accomplish intra-PAN routing via layer-three forwarding, in which each 802.15.4 radio hop is an IP hop.

The key concept applied throughout the 6LoWPAN adaptation layer is the use of stateless or shared-context compression to elide adaptation-, network-, and transport-layer header fields – compressing all three layers down to a few bytes, combined. We can see that it's possible to compress header fields to a few bits when we observe that they often carry common values, reserving an escape value for when less-common ones appear. Common values occur due to frequent use of a subset of IPv6 functionality (such as UDP, TCP, and ICMPv6 as Next Header values) and simple assumptions of shared context (for example, a common network prefix assigned to the entire LoWPAN). 6LoWPAN also elides redundant header information across protocol layers (for instance, UDP and IPv6 length fields and IPv6 addresses are derived from lower-layer headers).

Traditional IP header compression techniques are stateful and generally focus on optimizing individual flows over a highly constrained link. These methods assume that the compressor and decompressor are in direct and exclusive communication and compress both network- and transport-layer headers together. They optimize for long-lived flows by exploiting redundancies across packets within a flow over time, requiring the endpoints to initially send packets uncompressed. Flow-based compression techniques are poorly suited for LoWPANs. Traffic in many LoWPAN applications is driven by infrequent readings or notifications, rather than long-lived flows. Communication over multiple hops requires hop-by-hop compression and decompression and per-flow state at each intermediate node. Many LoWPAN routing protocols obtain receiver diversity via rerouting, which would require state migration and reduce compression effectiveness. In contrast, stateless and shared-context compression in 6LoWPAN doesn't require any per-flow state and lets routing protocols dynamically choose routes without affecting compression efficiency. Looking at 6LoWPAN's specifics, we can see how extensively it employs stateless compression.

Encapsulation Header Format

6LoWPAN uses header stacking to keep orthogonal concepts separate and enforce a well-defined method for expressing its capabilities. Analogous to IPv6 extension headers, 6LoWPAN expresses each capability in a self-contained subheader: *mesh addressing*, *fragmentation*, and *header compression*. Mesh addressing supports layer-two forwarding, and fragmentation supports the IPv6 minimum MTU requirement. 6LoWPAN identifies all header formats using a *header type* field placed at the beginning of each header. A 6LoWPAN Not-A-LoWPAN (NALP) type enables it to coexist with other protocols that operate directly on the link. The header stack is simple to parse and allows elision of headers when unneeded. The fragmentation header is elided for small datagrams, indicating that a single frame carries the entire payload. Similarly, the mesh header is elided when 6LoWPAN frames are delivered over a singe radio hop, so the path source and destination are identical to those in the link-layer header. Figure 2 shows typical header stacks.

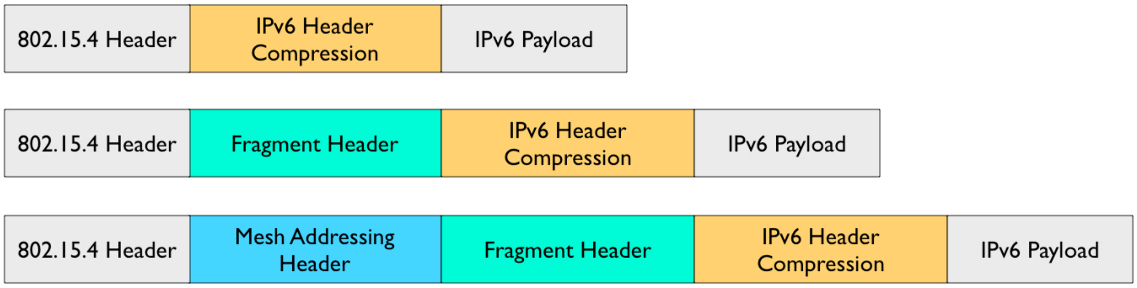

Figure 2. Typical 6LoWPAN Header Stacks.

Fragment Header

The fragment header is used when the payload is too large to fit in a single IEEE 802.15.4 frame. The Fragment header is analogous to the IEEE 1394 Fragment header and includes three fields: *Datagram Size*, *Datagram Tag*, and *Datagram Offset*. Datagram Size identifies the total size of the unfragmented payload and is included with every fragment to simplify buffer allocation at the receiver when fragments arrive out-of-order. Datagram Tag identifies the set of fragments that correspond to a given payload and is used to match up fragments of the same payload. Datagram Offset identifies the fragment's offset within the unfragmented payload and is in units of 8-byte chunks. To allow arbitrary byte offsets would require 11 bits to support the 1280 byte minimum MTU requirement, but requiring 8-byte alignment requires only 8 bits for the offset.

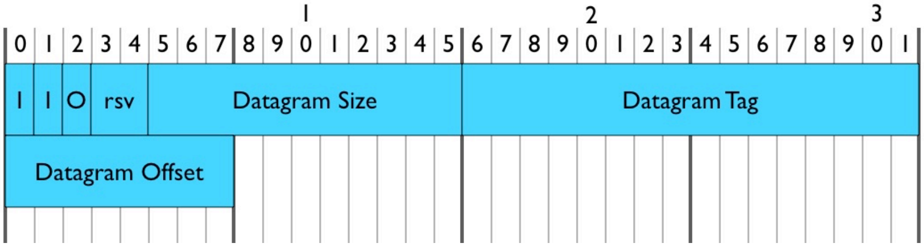

Figure 3. 6LoWPAN Fragment Header.

The Fragment header format is shown in Figure 3. The header type is only two bits. The third bit is used to compress the datagram offset on the first fragment as it is always zero. The fragment header is 4 bytes for the first fragment and 5 bytes for all subsequent fragments.

Mesh Addressing Header

The Mesh Addressing header is used to forward 6LoWPAN payloads over multiple radio hops and support layer-two forwarding. The mesh addressing header includes three fields: *Hop Limit*, *Source Address*, and *Destination Address*. The Hop Limit field is analogous to the IPv6 Hop Limit and limits the number of hops for forwarding. The Hop Limit field is decremented by each forwarding node, and if decremented to zero the frame is dropped. The source and destination addresses indicate the end-points of an IP hop. Both addresses are IEEE 802.15.4 link addresses and may carry either a short or extended address.

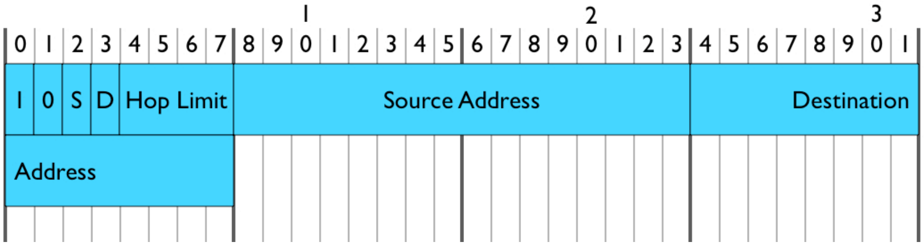

Figure 4. 6LoWPAN Mesh Addressing Header.

The Mesh Addressing format is shown in Figure 4. The header type is only two bits. The third and fourth bits indicate which addressing mode to use for the source and destination addresses. The following bits carry the hop limit and addressing fields. The Mesh Addressing header ranges between 5 and 17 bytes depending on the addressing modes in use.

IPv6 Header Compression in RFC 4944

RFC 4944 defines HC1, a stateless compression scheme optimized for link-local IPv6 communication [4]. HC1 is identified by an encoding byte following the Compressed IPv6 dispatch header, and it operates on fields in the upper-layer headers. 6LoWPAN elides some fields by assuming commonly used values. For example, it compresses the 64-bit network prefix for both source and destination addresses to a single bit each when they carry the well-known link-local prefix. 6LoWPAN compresses the Next Header field to two bits whenever the packet uses UDP, TCP, or ICMPv6. Furthermore, 6LoWPAN compresses Traffic Class and Flow Label to a single bit when their values are both zero. Each compressed form has reserved values that indicate that the fields are carried inline for use when they don't match the elided case. 6LoWPAN elides other fields by exploiting cross-layer redundancy. It can derive Payload Length – which is always elided – from the 802.15.4 frame or 6LoWPAN fragmentation header. The 64-bit interface identifier (IID) for both source and destination addresses are elided if the destination can derive them from the corresponding link-layer address in the 802.15.4 or mesh addressing header. Finally, 6LoWPAN always elides Version by communicating via IPv6.

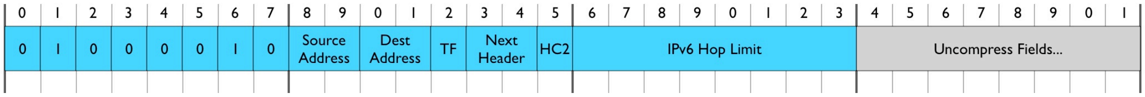

Figure 5. 6LoWPAN RFC 4944 IPv6 Header Compression.

The HC1 encoding is shown in Figure 5. The first byte is the dispatch byte and indicates the use of HC1. Following the dispatch byte are 8 bits that identify how the IPv6 fields are compressed. For each address, one bit is used to indicate if the IPv6 prefix is link-local and elided and one bit is used to indicate if the IID can be derived from the IEEE 802.15.4 link address. The TF bit indicates whether Traffic Class and Flow Label are both zero and elided. The two Next Header bits indicate if the IPv6 Next Header value is

UDP, TCP, or ICMP and compressed or carried inline. The HC2 bit indicates if the next header is compressed using HC2. Fully compressed, the HC1 encoding reduces the IPv6 header to three bytes, including the dispatch header. Hops Left is the only field always carried inline.

UDP Header Compression in RFC 4944

RFC 4944 uses stateless compression techniques to reduce the overhead of UDP headers. When the HC2 bit is set in the HC1 encoding, an additional 8-bits is included immediately following the HC1 encoding bits that specify how the UDP header is compressed. To effectively compress UDP ports, 6LoWPAN introduces a range of well-known ports (61616 – 61631). When ports fall in the well-known range, the upper 12 bits may be elided. If both ports fall within range, both Source and Destination ports are compressed down to a single byte. HC2 also allows elision of the UDP Length, as it can be derived from the IPv6 Payload Length field. The UDP encoding field is shown in Figure 6.

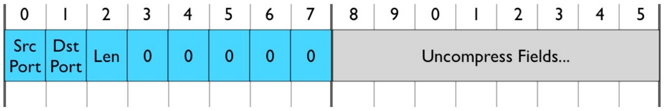

Figure 6. 6LoWPAN RFC 4944 UDP Header Compression Encoding.

RFC 4944 UDP/IPv6 Header Compression Examples

Typical header configurations using HC1 and HC2 are shown in Figure 7. The best-case compression efficiency occurs with link-local unicast communication, where HC1 and HC2 can compress a UDP/IPv6 header down to 7 bytes. The Version, Traffic Class, Flow Label, Payload Length, Next Header, and link-local prefixes for the IPv6 Source and Destination addresses are all elided. The suffix for both IPv6 source and destination addresses are derived from the IEEE 802.15.4 header.

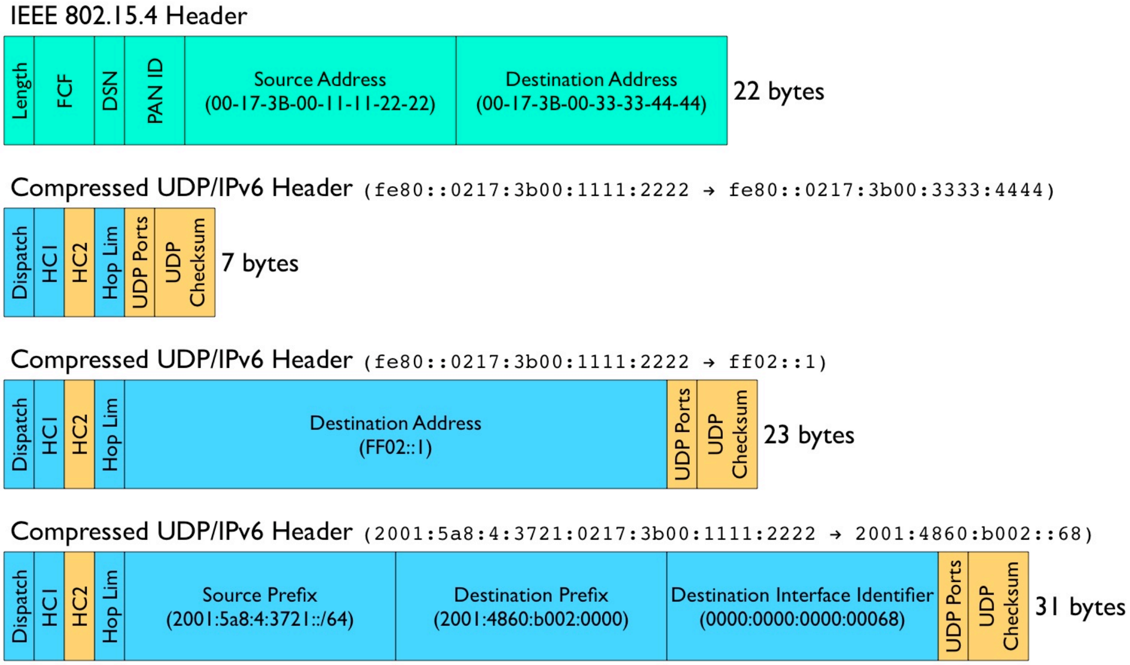

Figure 7. 6LoWPAN RFC 4944 Header Compression Examples.

However, RFC 4944 does not efficiently compress headers when communicating outside of link-local scope or when using multicast. Any prefix other than the link-local prefix must be carried inline. Any suffix must be at least 64 bits when carried inline even if derived from a short 802.15.4 address. As shown in Figure 7, HC1/HC2 can compress a link-local multicast UDP/IPv6 header down to 23 bytes in the best case. When communicating with nodes outside the LoWPAN, the IPv6 Source Address prefix and full IPv6 Destination Address must be carried inline.

Improved UDP/IPv6 Header Compression

To provide better compression over a broader range of scenarios, the 6LoWPAN working group is standardizing an improved header compression encoding format, called HC. The format defines a new encoding for compressing IPv6 header, called IPHC. The new format allows Traffic Class and Flow Label to be individually compressed, Hop Limit compression when common values (e.g. 1 or 255) are used, makes use of shared-context to elide the prefix from IPv6 addresses, and supports multicast addresses most often used for IPv6 ND and SLAAC.

Contexts act as shared state for all nodes within the LoWPAN. A single context holds a single prefix. IPHC identifies the context using a 4-bit index, allowing IPHC to support up to 16 contexts simultaneously within the LoWPAN. When an IPv6 address matches a context's stored prefix, IPHC compresses the prefix to the context's 4-bit identifier. Note that contexts are not limited to prefixes assigned to the LoWPAN but can contain any arbitrary prefix. As a result, share contexts can be configured such that LoWPAN nodes can compress the prefix in both Source and Destination addresses even when communicating with nodes outside the LoWPAN.

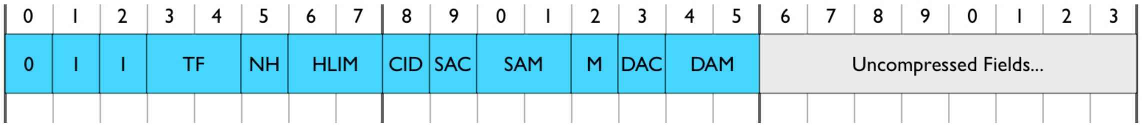

Figure 8. 6LoWPAN Improved IPv6 Header Compression

The improved header compression encoding is shown in Figure 8. The first three bits (011) form the header type and indicate the use of IPHC. The TF bits indicate whether the Traffic Class and/or Flow Label fields are compressed. The HLIM bits indicate whether the Hop Limit takes the value 1 or 255 and compressed, or carried inline.

Bits 8-15 of the IPHC encoding indicate the compression methods used for the IPv6 Source and Destination Addresses. When the Context Identifier (CID) bit is zero, the default context may be used to compress Source and/or Destination Addresses. This mode is typically when both Source and Destination Addresses are assigned to nodes in the same LoWPAN. When the CID bit is one, two additional 4-bit fields follow the IPHC encoding to indicate which one of 16 contexts is in use for the source and destination addresses.

The Source Address Compression (SAC) indicates whether stateless compression is used (typically for link-local communication) or stateful context-based compression is used (typically for global communication). The Source Address Mode (SAM) indicates whether the full Source Address is carried inline, upper 16 or 64-bits are elided, or the full Source Address is elided. When SAC is set and the Source Addresses' prefix is elided, the identified context is used to restore those bits.

The Multicast (M) field indicates whether the Destination Address is a unicast or multicast address. When the Destination Address is a unicast address, the DAC and DAM bits are analogous to the SAC and SAM bits. When the Destination Address is a multicast address, the DAM bits indicate different forms of multicast compression. For additional details, please refer to the current Internet Draft [2].

HC also defines a new framework for compressing arbitrary next headers, called NHC. HC2 in RFC 4944 is only capable of compressing UDP, TCP, and ICMPv6 headers, the latter two are not yet defined. Instead, the NHC header defines a new variable length Next Header identifier, allowing for future definition of arbitrary next header compression encodings.

HC initially defines a compression encoding for UDP headers, similar to that defined in RFC 4944. Like RFC 4944, HC utilizes the same well-known port range (61616-61631) to effectively compress UDP ports down to 4-bits each in the best case. However, HC no longer provides an option to carry the Payload Length in line, as it can always be derived from the IPv6 header. Finally, HC allows elision of the UDP Checksum whenever an upper layer message integrity check covers the same information and has at least the same strength. Such a scenario is typical when transport- or application-layer security is used. As a result, the UDP header can be compressed down to two bytes in the best case.

Improved UDP/IPv6 Header Compression Examples

Typical header configurations using IPHC and NHC are shown in Figure 9. As with RFC 4944, the best-case compression efficiency occurs with link-local unicast communication – IPHC and NHC can compress a UDP/IPv6 header down to 6 bytes. The Version, Traffic Class, Flow Label, Payload Length, Next Header, Hop Limit, and link-local prefixes for the IPv6 Source and Destination addresses are all elided. The suffix for both IPv6 source and destination addresses are derived from the IEEE 802.15.4 header.

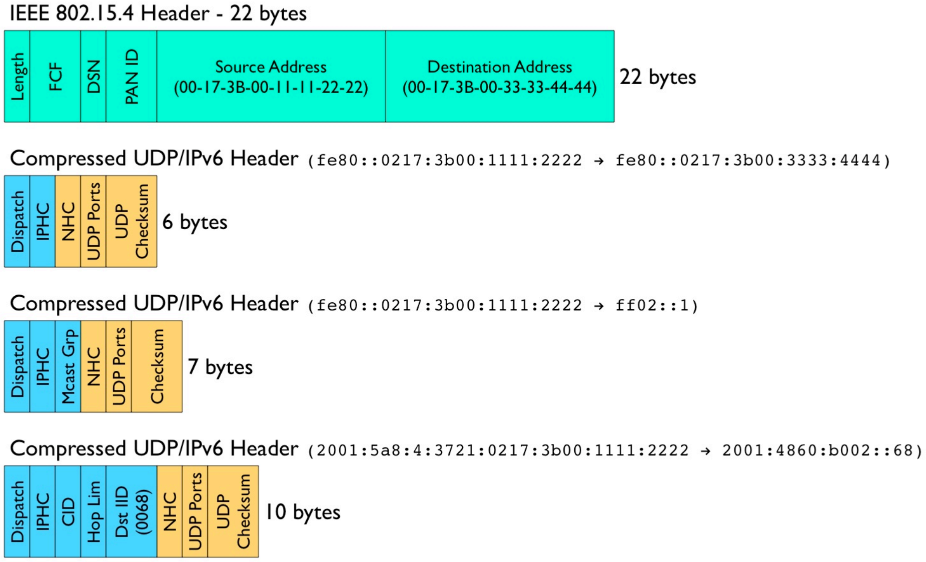
IEEE 802.15.4 Header - 22 bytes
Length
FCF
DSN
PAN ID
Source Address (00-17-3B-00-11-11-22-22)
Destination Address (00-17-3B-00-33-33-44-44)
22 bytes
Compressed UDP/IPv6 Header (fe80::0217:3b00:1111:2222 → fe80::0217:3b00:3333:4444)
Dispatch
IPHC
NHC
UDP Ports
UDP Checksum
6 bytes
Compressed UDP/IPv6 Header (fe80::0217:3b00:1111:2222 → ff02::1)
Dispatch
IPHC
Mcast Grp
NHC
UDP Ports
Checksum
7 bytes
Compressed UDP/IPv6 Header (2001:5a8:4:3721:0217:3b00:1111:2222 → 2001:4860:b002::68)
Dispatch
IPHC
CID
Hop Lim
Dst IID (0068)
NHC
UDP Ports
UDP Checksum
10 bytes

Figure 9. 6LoWPAN Improved Header Compression Examples

The improvements made in IPHC become obvious with multicast communication and global communication. As shown in Figure 9, IPHC can compress communication to well-known multicast addresses down to 7 bytes (vs. 23 bytes with HC1). Well-known multicast addresses have limited their group IDs to only the bottom few bytes, and IPHC takes advantage of this property. When communicating with global addresses, IPHC can compress a UDP/IPv6 header down to 9 or 10 bytes (vs. 31 bytes with HC1). Using context-based compression, the prefix of both addresses can be compressed. The Source Addresses' IID may be compressed if it is derivable from the IEEE 802.15.4 header. Finally, unlike RFC 4944, IPHC need only carry 2 bytes of the IID inline if the upper 6 bytes are all zeros.

IPv6/6LoWPAN Architecture

The 6LoWPAN format specification defines how fragmentation, compression, and layer-two forwarding are represented in an 802.15.4 frame. However, the implementation of those capabilities is out of that document's scope. 6LoWPAN's dependencies on the specific operations defined in the 802.15.4 MAC are minimal, supporting essentially any MAC protocol that provides the 802.15.4 frame format. Similarly, the 6LoWPAN format doesn't specify how IPv6 capabilities, such as ND and SAA, are orchestrated to configure the LoWPAN to be consistent with the adaptation layer. Next, we outline IPv6 over 6LoWPAN's key architectural issues.

IEEE 802.15.4 in Practice

802.15.4 presents several pragmatic issues that have significant architectural impact beyond the 6LoWPAN adaptation layer. Whereas in conventional WPAN settings, the user typically adjusts device and host placement so that the link between them is adequate, in typical LoWPAN settings a network of many devices is embedded in a physical environment at particular, meaningful locations. Network protocols must deal with the many exigencies that arise. Multi-hop routing extends range and helps avoid obstacles. Thus, a LoWPAN network isn't typically a single broadcast domain. Moreover, the link quality between any node pair is often complex and time-varying due to environmental factors. Hop-by-hop retransmission schemes help make lossy 802.15.4 links viable for multihop communication, but alone they aren't sufficient. Links that are reasonably good on average – say, with 90 percent packet-reception reliability – will often experience bursts of loss due to changes in the noise floor and spurious interference. Routing can overcome such bursts when forwarding datagrams by selecting an alternate path. In effect, routing can exploit receiver diversity by dynamically selecting from multiple next-hop candidates. To deal with these link challenges, the network layer requires extra visibility into detailed link behavior to build and maintain effective routing structures.

Many LoWPAN applications have significant device mobility within the LoWPAN, giving rise to time-varying connectivity relationships, in addition to variations induced by changing environmental factors. For example, package tracking might involve numerous devices moving among a set of stationary ones. This isn't IP mobility in the traditional sense because nodes might remain in close physical proximity and be connected within the LoWPAN. However, such variations require that the routing topology adapt to connectivity changes.

The 802.15.4 specification defines only a limited set of power-management mechanisms for edge devices and no power management for forwarding devices. Consequently, most commercial implementations and industrial standards built on 802.15.4 forego the defined power-management mechanisms when defining routing protocols. To conserve energy, nodes must duty cycle the radio, but doing so requires both transmitter and receiver to coordinate when and how to communicate. Common mechanisms for this involve sampled listening techniques, in which the receiver periodically listens for lengthened transmissions, or scheduling techniques, which involve time synchronization between nodes. 6LoWPAN, so far, avoids requiring particular MAC features. When adapting IPv6 components to operate over 802.15.4 links, we should exercise similar care regarding dependencies on the specific underlying MAC protocol.

Mesh Under vs. Route Over

Two important architectural issues for IPv6 over LoWPAN are how link-level factors inform routing and at what layer datagram forwarding occurs within the LoWPAN. Traditionally, IP routing occurs at the network layer in a manner largely independent from the underlying links that implement the individual hops. 6LoWPAN, in its role as an adaptation between the link (layer two) and the network (layer three), can support routing at either layer. Figure 10 show the difference in packet-processing between the two approaches.

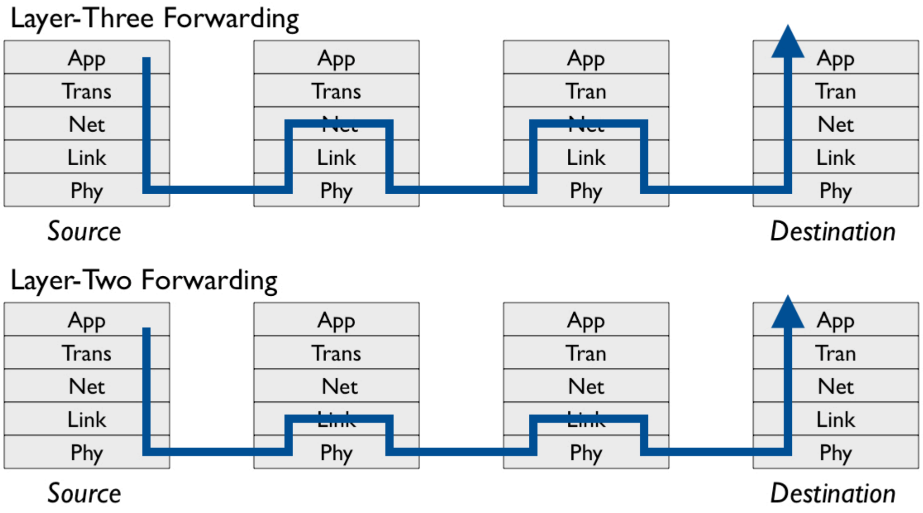

Figure 10. Mesh-Under (Layer-Two) vs. Route-Over (Layer-Three) Forwarding.

In a mesh under organization, the network stack performs no IP routing within the LoWPAN; instead, the adaptation layer seeks to mask the lack of a full broadcast at the physical level by transparently routing and forwarding packets within the LoWPAN. By emulating a full broadcast link, it potentially provides compatibility with IPv6 protocols that expect such communication behavior. The challenge is that logical link emulation is significantly more complex in LoWPANs than in traditional infrastructure-based 802.11 topologies. Mesh topologies require forwarding over multiple radio hops, and link-local multicast must deliver packets to all nodes in the entire LoWPAN. Many mechanisms that exist to form, maintain, and diagnose IP routing must also be recreated at the link layer for meshing to operate reliably.

Alternatively, route over performs routing at the IP layer, with each node serving as an IP router. We can view it as a collection of over-lapping link-local scopes, with each link-local domain defined by the inherent radio connectivity. Unlike mesh under, route over supports layer three forwarding mechanisms within the LoWPAN that can utilize network-layer capabilities defined by IP, such as IPv6 routing and ICMPv6 for configuration and management. Route over also lets IP routing protocols span different link technologies, enabling better integration into more capable networks. It also lets IP-based protocols constrain IP communication to local radio coverage, rather than an entire LoWPAN. Issues of link- versus network-layer routing aren't unique to 6LoWPAN — they arise in Frame Relay, Asynchronous Transfer Mode (ATM), switched Ethernet, and

802.11 meshing. For example, we've experienced similar challenges with IP over ATM, in which independent link-level routing makes it difficult to optimize IP routes end-to-end with two non communicating routing protocols operating a different layers independently of each other. Additionally, two independent routing layers can have unintended interactions, especially when reacting to changes in link state (mutli-layer recovery). It is also worth mentioning that a multi-layer routing architecture is challenging in term of route optimality and multi-layer recovery involving complex bottom-up token based approach).

Addressing and Autoconfiguration

Using Source Address Autoconfiguration (SAA), each host generates a link-local IPv6 unicast address from its IEEE EUI-64 address, 16-bit address, or both. In mesh under, the link-local scope covers an entire LoWPAN, possibly over multiple hops, and a link-local address is sufficient for communication within the LoWPAN, whereas a routable address is required to communicate outside. In route over, a link-local address is sufficient to communicate with nodes in direct radio communication, but a routable address is required to communicate with devices that are multiple radio hops away.

For all unicast addresses, regardless of their scope, it's cost effective to derive them from the 802.15.4 link address. 6LoWPAN's binding between link, adaptation, and IP headers enables 6LoWPAN to elide IP addresses derived from link addresses and removes the need for address resolution. Similarly, autoconfiguration should configure interface addressing using a common prefix so that 6LoWPAN header compression can elide the prefix. 6LoWPAN can use the short link address to derive IP addresses – allowing reduced header overhead when using the mesh addressing header or HC – and maintain privacy by using a token with local scope.

Neighbor Discovery

IPv6 Neighbor Discovery (ND) lets a node discover neighbors, maintain reachability information, configure default routes, and propagate configuration parameters [3]. Because ND is only intended for interaction between link neighbors, it should come as no surprise that ND is only defined for operation over a single IP link. ND performs address resolution and Neighbor Unreachability Detection (NUD) by sending unicast queries to neighboring nodes. Hosts listen for Router Advertisements from routers to receive important network information (e.g. prefixes, default hop limit, etc.) and configure default routes. Nodes also use ND to perform Duplicate Address Detection (DAD) when assigning an IPv6 address to an interface. All ND communication occurs within link-local scope.

However, there are significant challenges to using the current ND specification within LoWPANs. Specifically, ND makes extensive use of link-local multicast for sending address resolution solicitations, router advertisements, and DAD solicitations. In a mesh under configuration that emulates a single IP link over the entire LoWPAN, ND can potentially be used unmodified. However, supporting multicast to all nodes within the LoWPAN must be implemented using expensive floods. In the best case, every node in the LoWPAN must receive a message destined to the link-local all-nodes multicast address with reasonably high reliability – surely an expensive operation, especially in large networks with multiple radio hops.

Interestingly, the existing ND protocol might actually be better suited to a route over configuration. Because the link-local scope is defined by radio communication range, a link-local multicast reduces to a link-layer broadcast. Performing address resolution, NUD, and sending router advertisements no longer require expensive multicast delivery over multiple hops at the link layer. However, DAD presents a different challenge in a route over configuration. DAD assumes that the link provides transitive reachability (e.g. if node a can reach b and node b can reach c, then node a can reach c). By assuming transitive reachability, nodes can determine that an address is unique within a link simply be sending a link-local multicast. A route over configuration, however, does not support transitive reachability. In fact, a route over configuration is best characterized as a set of overlapping link-local scopes, each one defined by a node and its direct radio neighbors. As a result, a link-local multicast is no longer sufficient, as a node must check the uniqueness of an address with its two-hop neighbors, at the very least. But due to the dynamic nature of radio connectivity and node mobility common to many applications, even verifying address uniqueness among two-hop neighbors is insufficient.

In general, nodes must maintain uniqueness of assigned IPv6 addresses (including link-local addresses) within the entire LoWPAN, for both mesh under and route over configurations. The naïve solution would fall back to sending a multicast to all nodes in the LoWPAN – no different that what is already specified in ND. A more efficient solution would utilize border routers to maintain a list of nodes within the LoWPAN. Utilizing extra capabilities at the border router, LoWPAN nodes can then communicate via unicast with a border router to verify the uniqueness of an address. The IETF 6LoWPAN working group is currently working on such ND optimizations to support efficient operation over LoWPANs.

Routing

Limited memory and communication capabilities constrain the routing state at each node as well as the routing information that might be communicated. These restrictions preclude using protocols that rely on complete link-state information. Traditional distance vector mobile ad-hoc networks (manet) protocols are also ill-suited because they assume a high rate of mobility for all nodes in the network, whereas LoWPAN nodes are better characterized by more structured mobility within a set of stationary nodes. Consequently, manet protocols use frequent floods to discover and maintain routes. Caches used to optimize communication only trade memory for communication. In addition, most of these protocols exchange route maintenance information at rates that far outpace typical LoWPAN communication and react to link fading with expensive route-repair actions. Instead, LoWPAN routing protocols must operate using incomplete information and tolerate some inconsistency. Interestingly, we're returning to scalability issues similar to those encountered with the early Internet, but this time in a wireless setting. The new Routing over Low Power and Lossy Links (ROLL) working group within the IETF routing directorate will soon address these challenges [8].

Security Considerations

6LoWPAN takes advantage of the strong AES-128 link-layer security mechanisms provided by IEEE 802.15.4. Transport layer mechanisms have also been shown to be feasible on 6LoWPAN networks. However, while network-layer security mechanisms such as IPsec and Secure Neighbor Discovery are becoming mature, their feasibility on LoWPANs is still being questioned.

Related Work in Low-Power WPANs

RFC 4944 gives a complete description of 6LoWPAN, the packet format standardized by the IETF to enable IPv6 communication over low-power, wireless personal area networks (LoWPANs). Support for IP in resource-constrained environments has a long history, including over telephone modems that gave rise to Point-to-Point Protocol, Dynamic Host Configuration Protocol for autoconfiguration, and header compression. 6LoWPAN differs in how it exploits shared context, frequently occurring simple cases, and cross-layer redundancy to vastly reduce header overhead when communicating over a dynamic, multihop topology. 6LoWPAN builds on prior work with stateless IP header compression. Many efforts have addressed links in which multihop forwarding is required, including frame relay and Asychronous Transfer Mode. 6LoWPAN is unique in that it also addresses severe resource constraints.

IEEE 802.15.1 (Bluetooth) is another wireless link technology that falls under the WPAN classification. Intended to serve as a cable-replacement technology, Bluetooth supports relatively high throughput for a limited number of nodes within a small range. IEEE 802.15.3 pushes WPAN capabilities further, with greater throughput and support for more nodes. Although both are intended for battery operation, they only target lifetimes of several days to several weeks. In contrast, 802.15.4 is intended for low data-rate applications in which numerous nodes must be low-cost and have multiyear lifetimes on modest batteries. The 802.15.4 standard supports up to 64,000 nodes within a PAN compared to a small handful with other WPAN links. 802.15.4 has also reduced complexity, intended to function with eight-bit microcontrollers providing 8 Kbytes of RAM or less. Although IP over Bluetooth using the Bluetooth Network Encapsulation Protocol has been around for several years, it’s typically used to provide a point-to-point connection over a single radio hop.

Researchers have developed numerous mesh network layers over 802.15.4, as open source projects (such as TinyOS, industrial forums (ZigBee and WirelessHART), or proprietary offerings (Dust Networks, Sensicast, and Millenial Net). Each has defined its own set of incompatible packet formats tied to particular MAC features, routing algorithms, and addressing. Many address only the individual 802.15.4 subnet, leaving all further communication protocols to be defined via ad hoc gateways. 6LoWPAN potentially lets us unify this disparate activity and enable embedded 802.15.4 devices to be incorporated into Ethernet, Wi-Fi, General Packet Radio Service, and other environments within a uniform IP framework. Many embedded TCP/IP stacks provide IP host functionality and are widely used in wired and powered settings. However, few embedded IP stacks directly address the issues related to supporting IP over low-power mesh topologies in LoWPANs.

Within the IETF, the mobile ad hocs networks (manet) working group and related research activities' tremendous effort has been devoted to reactive and proactive routing protocols for mobile devices. This work has assumed capable, high-bandwidth links and powerful hosts with high, random mobility. As such, it used conventional IP datagrams and frame formats and hasn't attend to the impact of resource constraints. Work in the IETF Ad-Hoc Network Autoconfiguration (AUTOCONF) working group is devoted to developing solutions for stateless address autoconfiguration and Neighbor Discovery in settings in which IP connectivity is naturally viewed as a collection of overlapping partial broadcast domains. The Routing Over Low power and Lossy links (ROLL) working group was recently chartered to address routing in LoWPANs (independently of the link layer).

Conclusion

Until recently, extending IP out to wireless industrial networks was thought to be impractical, if not impossible. Vendors embraced proprietary protocols because they presumed that IP, which is memory- and bandwidth-intensive, couldn't be scaled down to operate on the microcontrollers and low-power wireless links used in these environments. Recent efforts within the IETF make IP over low power communication links now feasible, including IEEE 802.15.4. These developments make IP attractive for use in low-power devices, everything from handhelds to instruments.

References

1. A. Dunkels and JP. Vasseur. IP for Smart Objects. IPSO Alliance White Paper, September 2008.
2. J. Hui and P. Thubert. Compression Format for IPv6 Datagrams in 6LoWPAN Networks. Internet Draft (Work in Progress), December 2008.
3. IPv6 over Low power WPAN (6lowpan) Working Group. Internet Engineering Task Force (IETF). http://www.ietf.org/html.charters/6lowpan-charter.html.
4. G. Montenegro, N. Kushalnagar, J. Hui, and D. Culler. Transmission of IPv6 Packets over IEEE 802.15.4 Networks. RFC 4944 (Proposed Standard), September 2007.
5. T. Narten, E. Nordmark, W. Simpson and H. Soliman. Neighbor Discovery for IP version 6 (IPv6). RFC 2461 (Proposed Standard), September 2007.
6. E. Nordmark. Stateless IP/ICMP Translation Algorithm (SIIT). RFC 2765 (Proposed Standard), February 2000.
7. E. Nordmark and R. Gilligan. Basic Transition Mechanisms for IPv6 Hosts and Routers. RFC 4213 (Proposed Standard), October 2005.
8. Routing Over Low power and Lossy networks (ROLL) Working Group. Internet Engineering Task Force (IETF). http://www.ietf.org/html.charters/roll-charter.html.