Independent Submission
Internet-Draft
Intended status: Informational
Expires: January 14, 2021

H. Ayers
P. Levis
Stanford University

July 13, 2020

Design Considerations for Low Power Internet Protocols
draft-ayers-low-power-interop-01

Abstract

Low-power wireless networks provide IPv6 connectivity through 6LoWPAN, a set of standards to aggressively compress IPv6 packets over small maximum transfer unit (MTU) links such as 802.15.4.

The entire purpose of IP was to interconnect different networks, but we find that different 6LoWPAN implementations fail to reliably communicate with one another. These failures are due to stacks implementing different subsets of the standard out of concern for code size. We argue that this failure stems from 6LoWPAN's design, not implementation, and is due to applying traditional Internet protocol design principles to low-power networks.

We propose three design principles for Internet protocols on low-power networks, designed to prevent similar failures in the future. These principles are based around the importance of providing flexible tradeoffs between code size and energy efficiency. We apply these principles to 6LoWPAN and show that the modified protocol provides a wide range of implementation strategies while allowing implementations with different strategies to reliably communicate.

Table of Contents

1. Introduction

Interoperability has been fundamental to the Internet's success. The Internet Protocol (IP) allows devices with different software and link layers to communicate. IP provides a basic communication

substrate for many higher layer protocols and applications. In the decades of the Internet's evolution, we have accumulated and benefited from a great deal of wisdom and guidance in how to design, specify, and implement robust, interoperable protocols.

Over the past decade, the Internet has extended to tens of billions of low-power, embedded systems such as sensor networks and the Internet of Things. Hundreds of proprietary protocols have been replaced by 6LoWPAN, a standardized format for IP networking on low-power wireless link layers such as 802.15.4 [RFC6282] and Bluetooth Low Energy [RFC7668]. 6LoWPAN was created with the express purpose of bringing interoperable IP networking to low power devices, as stated in the 6LoWPAN WG charter. Many embedded operating systems have adopted 6LoWPAN, [tinyos] [riot] [mbedos] [contiki-ng] [contiki] [lite-os] and [zephyr], and every major protocol suite uses it (zigbee, openthread). In fact, devices today can communicate with the broader Internet.

However, in many cases 6LoWPAN implementations cannot communicate _with each other_. We find that no pairing of the major implementations fully interoperates. Despite 6LoWPAN's focus on interoperability, two key features of the protocol -- range extension via mesh networking of devices, and the convenience of different vendors being able to share a gateway -- are largely impossible 10 years later.

Each of the openly available 6LoWPAN stacks, most of which are used in production, implements a subset of the protocol _and_ includes compile-time flags to cut out additional compression/decompression options. As a result, two devices might both use 6LoWPAN, yet be unable to exchange certain IP packets because they use required features the other does not implement. This is especially problematic for gateways, which _need_ to be able to talk to multiple implementations to enable significant scaling in real world applications.

This draft argues that the failure of 6LoWPAN interoperability stems from applying traditional protocol design principles to low-power networks. Low-power protocols minimize energy consumption via compression. Squeezing every bit out of packet headers, however, requires many different options/operating modes. Principles such as Postel's Law - "Be liberal in what you accept, and conservative in what you send" [RFC1122] - state that an implementation must be able to receive every feature, even thought it only sends some of them. However, code space is tight on many systems. As a result, when an application does not fit, developers cut out portions of the networking stack and stop working with other devices. Put another way, 3kB of unused compression code seems tiny, but when removing it

allows an additional 3kB of useful features, developers cut out parts of 6LoWPAN and devices become part of a custom networked system rather than the Internet.

This draft presents three design principles which resolve this tension between interoperability and efficiency. Protocols following these principles get the best of both worlds: resource-limited devices can implement subsets of a protocol to save code space while remaining able to communicate with every other implementation.

Capability spectrum: a low-power protocol specifies a linear spectrum of capabilities. Simpler implementations have fewer capabilities and save less energy, while fuller implementations have strictly more capabilities and are able to save more energy. When two devices differ in capability levels, communication can always fall back to the lower one.

Capability discovery: a low-power protocol provides mechanisms to discover the capability of communicating devices. This discovery can be proactive (advertisements) or reactive (in response to errors).

Explicit, finite bounds: a low-power protocol specifies explicit, finite bounds on growth during decompression. Without explicit bounds, buffers must be sized too conservatively. In practice, implementations allocate smaller buffers and silently drop packets they cannot decompress.

This draft is not the first to observe poor 6LoWPAN interoperability [probe-it-plugtest], but it is the first to identify this root cause. It is the first to define design principles for protocols that allow implementations to reduce code size and still interoperate. This draft examines how these principles could be applied to a low power protocol and evaluates the overhead of doing so. It finds that applying these principles to 6LoWPAN promises interoperability across a wide range of device capabilities, while imposing a code size cost of less than 10%. In particular, capability discovery requires an order of magnitude less code than the code size difference at the extremes of our capability spectrum, with minimal runtime overhead.

2. Background

6LoWPAN is a set of standards on communicating IPv6 over low-power wireless link layers such as IEEE 802.15.4 and Bluetooth[RFC4944] [RFC6282] [RFC7668]. 6LoWPAN primarily specifies two things: aggressive compression of IPv6 headers and how to fragment/re-assemble packets on link layers whose MTU is smaller than the minimum 1280 bytes required by IPv6. 6LoWPAN also specifies optimized IPv6

Neighbor Discovery. 6LoWPAN is critical to ensuring that IPv6 communication does not exceed the energy budget of low power systems.

The interoperability IP provides is a goal in and of itself, and so many different network stacks, including ZigBee and Thread, have transitioned to supporting IP connectivity with 6LoWPAN. IP connectivity allows systems to easily incorporate new services and applications, and allows applications to build on all of the existing Internet systems, such as network management and monitoring.

Low Power Hardware and Operating Systems

IoT Platform	**Code (kB)**	**Year**
EMB-WMB	64	2012
Zolertia Z1	92	2013
TI CC2650	128	2015
NXP MKW40Z	160	2015
SAMR21 XPro	256	2014
Nordic NRF52840 DK	512	2018

Figure 1: Flash Size across IoT Platforms

Figure 1 shows a variety of older and more recent low-power platforms. Modern microcontrollers typically have 128-512 kB of code flash. Applications often struggle with these limits on code flash, and rarely leave code space unused. Embedded systems are application specific, and use their limited available resources toward different ends. Despite this, they still rely on a small number of reusable OSes for basic abstractions.

To support the highly constrained applications and devices for which embedded OSes are used, embedded OSes must be minimal. However, the manner in which they must be minimal varies - some applications require minimal use of radio energy, which can require code size consuming techniques like compression, while others require tiny/low cost MCUs without space for those mechanisms. To support this variety, OSes have compile-time flags to include or exclude parts of the system or networking stack (contiki,mbedos,contiki-ng). Some systems take a more extreme approach, dynamically generating the minimum code to compile from the application itself (tinyos). Part of this minimalist focus is that until application developers demand certain features, OSes are likely to leave them out entirely (a requirement often specified in contribution guides - contiki-ng). These techniques are critical to ensuring a given OS can support a wide range of embedded platforms, and influence the implementation of network protocols.

3. 6LoWPAN Today

The original 6LoWPAN working group charter stated "The Working Group will generate the necessary documents to ensure interoperable implementations of 6LoWPAN networks". However looking at implementations today, we find that each includes a different subset of the protocol. Source code shows this is due to concerns with code size. Experiences with a new implementation verify these concerns.

Feature Fail A 6LoWPAN receiver has much less flexibility than a sender: it must be able to process any valid compression it receives. Figure 2 shows the receiver features supported by 6 major open-source 6LoWPAN stacks. Some, such as TinyOS, are mostly developed and used in academia. Others, such as ARM Mbed and Nest's OpenThread, are developed and supported commercially. Contiki and Contiki-NG sit somewhere in the middle, having both significant academic and commercial use. Riot is an open-source OS with hundreds of contributors for industrial, academic, and hobbyist use. Two widely used open source stacks excluded from this analysis are LiteOS [lite-os] and Zephyr [zephyr] -- they are excluded because LiteOS uses the same 6LoWPAN library (LWIP) used in MBED-OS, and Zephyr simply imports OpenThread.

In almost all cases, each stack's support for features is symmetric for sending and receiving. There are significant mismatches in feature support between stacks. These mismatches lead to deterministic cases when IP communication fails. We verified these failures by modifying existing network applications and testing them on hardware, using Wireshark to verify packets were compressed and formatted as we expected when receivers failed to decode packets. Every implementation pair fails for some type of packet which can be organically generated by one of the stacks. This result may be surprising when compared to prior work which demonstrated successful interoperability, such as [RPL-interop]. However, this early success preceded the release of RFC 6282, which increased the complexity (and overhead) of 6LoWPAN.

Feature	Stack					
	Contiki	NG	OT	Riot	Arm	TinyOS
Uncompressed IPv6	o	o		o	o	o
6LoWPAN Fragmentation	o	o	o	o	o	o
1280 byte packets	o	o	o	o	o	o
Dispatch_IPHC header prefix	o	o	o	o	o	o
IPv6 Stateless Address Compression	o	o	o	o	o	o
Stateless multicast addr compression	o	o	o	o	o	o
802.15.4 16 bit short address support		o	o	o	o	o
IPv6 Address Autoconfiguration	o	o	o	o	o	o
Stateful Address Compression	o	o	o	o	o	o
Stateful multicast addr compression			o	o	o	
TC and Flow label compression	o	o	o	o	o	o
NH Compression: Ipv6 (tunneled)			o		o	o
IPv6 NH Compression: UDP	o	o	o	o	o	o
UDP port compression	o	o	o	o	o	o
UDP checksum elision						o
Compression + headers past first frag				o	o	
Compression of IPv6 Extension Headers		~	~		o	o
Mesh Header			o		o	~
Broadcast Header						o
Regular IPv6 ND	o	o		o	o	~
RFC 6775 6LoWPAN ND				o	o	
RFC 7400 Generic Header Compression						

~ = Partial Support

Figure 2: 6LoWPAN Interoperability Matrix

Stack	Code Size Measurements (kB)			
	6Lo-All	Compression	Frag	Mesh/Bcast Hdr
Contiki-NG	6.2	3.4	1.9	N/A
Contiki	11.3	6.0	3.3	N/A
OT	26.6	20.0	1.3	4.5
Riot	7.5	>4.7	1.5	N/A
Arm Mbed	22.1	18.0	3.1	1331
TinyOS	16.2	----	----	0.6

Figure 3: 6LoWPAN stack code size for 6 open source stacks. The code size varies by over a factor of 4, in part due to different feature sets. Compression dominated the code requirements of each stack. Individual components do not add up to total as some size cannot be clearly attributed to any subcomponent. TinyOS's whole-program optimization model precluded separating out subcomponents, and only includes incomplete mesh header support.

Why?

IP communication can consistently fail in low-power networks, despite the presence of succinct standards (RFC 6282 + RFC 4944 is 52 pages) designed for low-power devices. Worse, this failure is silent: the receiver will simply drop the packet. Examining the documentation and implementation of each stack, code size concerns motivated feature elision. Mbed, Riot, and Contiki even provide compile-time flags to remove additional features for particularly constrained use cases.

Figure 3 shows a break down of the code size of each stack. Compression dominates the code size of 6LoWPAN implementations, and in several cases 6LoWPAN's size is comparable to the whole rest of the IPv6 stack. The Contiki and Contiki-NG implementations are significantly smaller than the others in part because they elide significant and complex features. The ARM Mbed IPv6 stack uses 45kB of flash. This is nearly 1/3 of the available space on a CC2650, just for IPv6: it does not include storage, sensors, the OS kernel, cryptography, higher layer protocols, signal processing, or applications.

Can a careful developer implement a leaner, fully-featured stack? To answer this question, we implemented our own 6LoWPAN stack. Our open-source implementation is written in Rust, for Tock, a secure embedded OS [tock].

Our experiences support the comments and documentation of the other stacks. We surpassed the size of the Contiki-NG and Riot 6LoWPAN code before adding support for recursive compression of IPv6 or the mesh and broadcast headers. We noted several aspects of the protocol required surprisingly complex code to properly handle. For example, 6LoWPAN requires IPv6 tunneled inside a compressed packet to compress interior headers as well. This requires the decompression library to support recursive invocation, which increases minimum execution stack sizes and makes tracking buffer offsets during decompression more difficult. Refusing to support tunneled IPv6 packets (e.g., Contiki) greatly simplified the code. Another example: headers in the first 6LoWPAN sub-IP fragment must be compressed, while headers in subsequent fragments must not be compressed. Given that low-power link layers have variable length headers, correctly determining exactly where to fragment and what should be compressed requires complex interactions between layers of the stack. Finally, 6LoWPAN requires support for out-of-order reception of fragments, potentially from different packets. This forced our receiver to store and track state for a collection of received packets, preventing reliance on a single global receive buffer. The exercise of implementing a 6LoWPAN stack from the ground up affirmed that code size concerns encourage feature elision.

Why does it matter?

For any 6LoWPAN implementation, there exists a border router implementation that can connect it to the broader Internet. But this status-quo model of connectivity forces vertical integration and fails to meet the original design goals of 6LoWPAN, for two reasons.

First, a 6LoWPAN gateway can not know how to safely compress packets for different nodes, unless it communicates only with devices produced by the same vendor. As a result, for a coverage area containing devices produced by 5 different vendors, 5 gateways are required. If not for feature mismatches, a single gateway would suffice. Second, the current situation significantly limits the potential for range extension via mesh topologies. Most existing 6LoWPAN meshes rely on "route-over" mesh routing at the network layer, which requires that each node can at least partially decompress and recompress IP headers when forwarding. Mesh-under routing is no better, as implementation of the mesh header is not universal (see Figure 2). Poor interoperability worsens the usability, range, cost, and efficiency of these networks.

4. Traditional Principles: Not Low-Power

Over the past 45 years, the Internet community has coalesced on a small number of design principles. Connectivity through interoperability is a key premise of the Internet. Principles such as layering and encapsulation support composing protocols in new ways (e.g., tunneling), while the end-to-end principle [end-to-end] allows building a robust network out of an enormous and complex collection of unreliable parts. The robustness principle asserts that implementations should make no assumptions on packets they receive: bugs, transmission errors, and even memory corruption can cause a device to receive arbitrarily formatted packets. It also asserts that an implementation must be ready to receive any and all properly formatted packets. This aspect of the principle is often attributed to John Postel as Postel's Law, first written down in the initial specification of IPv4: "In general, an implementation should be conservative in its sending behavior, and liberal in its receiving behavior."

Protocols often have optional features ("MAY, SHOULD, and OPTIONAL" in RFC language). Implicitly, due to Postel's Law, a receiver needs to handle either case. This scenario creates an asymmetry, where sender code can have reduced complexity but receiver code must be large and complex.

We need to think about low-power protocols differently. They need new principles to help guide their design. These principles need to embrace that there is no "one size fits all" design, while defining how devices choosing different design points interoperate. Flexibility needs to exist not only for senders, but also receivers, without harming interoperability.

5. Three Principles

This section describes three protocol design principles which prevent failures in low-power protocols. These principles are absolutely necessary to ensure interoperable implementations in this space, and should be closely observed.

5.1. Principle 1: Capability Spectrum

A low power protocol should be deployable/installable on devices which are at the low end of code and RAM resources. Rather than require every device pay the potential energy costs of fewer optimizations, a protocol should support a linear spectrum of device capabilities.

This may seem familiar -- the IP [RFC0791] and TCP [RFC0793] specifications provide optional fields which can be used by endpoints at their leisure; many non-standard HTTP headers will be ignored unless both client and server support them; TLS ciphersuite support is often asymmetrical. But this principle is different. For those examples, no linear spectrum exists -- support for any particular capability is generally unrelated to support for any other. Checking for support of any feature requires explicit enumeration of each, making it impossible to effectively compress such options. A non-linear spectrum requires storing feature support for every neighbor in RAM, or re-discovering capabilities on every exchange.

Low power protocols require simpler capability management. A low power protocol should define a capability spectrum with a clear ordering in which especially resource constrained devices can reduce code size or RAM use by eliding features. Such a spectrum makes a protocol usable by extremely low resource devices without forcing more resourceful devices to communicate inefficiently.

This capability spectrum should be a linear scale. For a device to support capability level N, it must also support all lower capability levels. More complex configuration approaches (e.g., a set of independent options) might allow for a particular implementation to be more efficient, picking the features that give the most benefit for the least added complexity. However, this sort of optimization makes interoperability more difficult, as two devices must negotiate each specific feature to use.

5.2. Principle 2: Capability Discovery

The second principle follows from the first: if different capability levels exist, there should be a mechanism for two devices to determine what level to communicate with.

The capability negotiation we propose here differs from capability discovery mechanisms built for traditional systems, such as IP Path MTU discovery or the Link Layer Discovery Protocol (LLDP). IP Path MTU discovery relies on continual probing until an acceptable value is discovered. LLDP requires regular, detailed capability advertisements at a fixed interval. The energy overhead of network probing or advertising is unacceptable in most low power environments. Capability discovery in low power networks should require no more than one failure between any two neighbors, even if this slightly increases the overhead per error. Proactive capability discovery should be built into baseline communication required for tasks like neighbor discovery or route maintenance. Further, assumptions for traditional systems that prohibit storing per-endpoint state do not apply, as nodes store information about link-

layer neighbors, _not IP endpoints_. This is needed because low-power networks with route over topologies frequently involve decompression and re-compression at each hop to enable forwarding. Low power nodes have few neighbors, so storing a few bits of state for each is feasible and can significantly reduce the amount of radio energy needed for communication. The code size cost of storing state is small compared to the cost of complex compression mechanisms.

In a low power network with capability discovery, if two devices wish to communicate, they default to the lower of their supported capability levels. E.g. a level 2 and a level 4 device should communicate at level 2. One offshoot of this principle is that it requires implementations have symmetric capabilities for send and receive - no benefits can be realized from an asymmetric implementation.

5.3. Principle 3: Explicit and Finite Bounds

Protocols must specify explicit and reasonable bounds on recursive or variable features so implementations can bound RAM use. This allow implementations to safely limit their RAM use without silent interoperability failures. This also ensures that capability discovery is sufficient for interoperability.

The idea of imposing bounds is, on its own, not unique to this space. TCP enforces a finite limit of 40 bytes for TCP options which may be appended to the TCP header, as does IPv4. DHCP allows for the communication of maximum DHCP message sizes. In the space of low power Internet protocols, however, this idea _must be pervasive._ Notably, the original designers of a specification may not know exactly what these values should be. This is not a new problem: TCP congestion control, for example, had to specify initial congestion window values. In this space, bounds should initially be very conservative. Over time, if increasing resources or knowledge suggests they should grow, then future devices will have the onus of using fewer resources to interoperate with earlier ones. The capability spectrum defined in the previous two principles can be helpful in this regard.

6. A Principled 6LoWPAN

This section proposes how to apply the three principles in the previous section to 6LoWPAN through specific modifications to the protocol. These modifications ensure that two 6LoWPAN devices can communicate even if they choose different code size/energy efficiency tradeoffs. We refer to this modified protocol as Principled 6LoWPAN (P6LoWPAN).

This application of our principles is not intended as a suggestion that these changes be made immediately to 6LoWPAN, as modifying an established protocol is a complex task very different from constructing new protocols. Instead, this application is a tool for evaluating these principles, and an example for how they should be applied.

6.1. Principle 1: Capability Spectrum

We propose replacing the large collection of "MUST" requirements -- the features in Figure 2 --into 6 levels of functionality. These "Capability Levels" are depicted in Figure 4.

Capability	**Basic Description / Added Features**
Level 0	Uncompressed IPv6 + ability to send ICMP errors
	- Uncompressed IPv6 - 6LoWPAN Fragmentation (Fragment Header) - 1280 Byte Packets - Stateless decompression of source addresses
Level 1	IPv6 Compression Basics + Stateless Addr Compression
	- Support for the Dispatch_IPHC Header Prefix - Correctly handle elision of IPv6 length and version - Stateless compression of all unicast addresses - Stateless compression of multicast addresses - Compression + 16 bit link-layer addresses

	- IPv6 address autoconfiguration
Level 2	Stateful IPv6 Address Compression
	- Stateful compression of unicast addresses - Stateful compression of multicast addresses
Level 3	IPv6 Traffic Class and Flow Label Compression
	- Traffic Class compression - Flow Label Compression - Hop Limit Compression
Level 4	Next Header Compression + UDP Port Compression
	- Handle Tunneled IPv6 correctly - Handle the compression of the UDP Next Header - Correctly handle elision of the UDP length field - Correctly handle the compression of UDP ports - Handle headers past the first fragment, when first fragment compressed.
Level 5 (all routers)	Entire Specification
	- Support the broadcast header and the mesh header - Support compression of all IPv6 Extension headers

Figure 4: Capability Spectrum

These levels prioritize features which provide the greatest energy savings per byte of added code size, based off our code size measurements and the number of bits saved by each additional compression mechanism. They allow for a wide range of code size/ efficiency tradeoffs.

For example, addresses dominate an uncompressed IPv6 header. Level 0 devices only support compressed source addresses, while level 1 devices support all stateless address compression. In one early design of this spectrum, Level 0 supported only uncompressed packets. However, this raises a problem with ICMP error generation. If a node cannot decompress the source address of a received packet, it cannot send ICMP errors. ICMP errors are required for capability discovery. Stateful compression depends on an out-of-band signal to set up state, such that nodes only send statefully compressed packets to nodes who also support it. Therefore decompressing stateless source addresses is a minimum requirement.

The classes in this scale do not precisely reflect the current feature support of the implementations described in Section 3. For example, Contiki supports UDP port compression (level 4) but does not support 802.15.4 short addresses (level 2) or stateful multicast compression (level 3): following this formulation, Contiki only provides level 1 support. If Contiki supported 16-bit addresses, it would provide level 2 support. A concrete spectrum such as the one above gives stack designers a structure and set of guidelines on the order in which to implement features. Based on our experiences developing a 6LoWPAN stack, we believe that if this scale existed as part of the initial specification, implementations would have made an effort to adhere to it.

One additional advantage of this spectrum is that it allows for some future additions to the P6LoWPAN specification without breaking interoperability between new and old implementations. For example, our scale does not include support for Generic Header Compression [RFC7400] because none of the open-source stacks we analyzed implement it. Despite this, support for this RFC could easily be added as a new class on this linear scale (as Class 6), and devices supporting it would know to not use it when communicating with lower class implementations.

This spectrum requires that a node store 3 bits of state for each neighbor. Given that low-power nodes often store ten or more bytes for each entry in their link table (link quality estimates, addresses, etc.), this cost is small. 6LoWPAN already assumes that

routers are more resourceful devices, P6LoWPAN routers are required to be level 5.

6.2. Principle 2: Capability Discovery

We propose two mechanisms by which P6LoWPAN performs capability discovery: neighbor discovery (ND) and ICMP\@. Neighbor discovery [RFC4861] is analogous to ARP in IPv4: it allows IPv6 devices to discover the link layer addresses of neighboring addresses as well as local gateways. Devices use neighbor discovery to proactively discover capability levels and ICMP to detect when incompatible features are used. Of the two, only ICMP is required. Neighbor discovery simply allows a pair of differing nodes to avoid an initial ICMP error, and allows for optimization of host-router communication during neighbor discovery.

ICMP: We propose adding a new ICMPv6 message type--P6LoWPAN Class Unsupported--which a device sends in response to receiving 6LoWPAN features it does not understand. This error encodes the device's capability level. A node receiving such an error updates its link table entry with the capability level. In the future, any packets sent to that address use at most the supported level.

Neighbor discovery: We propose adding an IPv6 ND option that allows a device to communicate its capability class during network association. This option would be included in Router Solicitations and Neighbor Advertisements, and would allow all devices that obtain link-layer addresses via ND to also know how to send packets which that neighbor can receive. When a node uses ND to resolve an IP address to a link layer address, it learns the supported capability level as well as the link layer address. This option minimizes the energy cost of communicating capabilities. It is worth noting that RFC 7400 already employs a similar method for communicating whether devices implement General Header Compression: adding such an option is clearly viable [RFC7400].

6.3. Principle 3: Provide Reasonable Bounds

Section 3 discussed two missing bounds which affect 6LoWPAN interoperability: limits on header decompression and bounds on recursion when decompressing tunneled IPv6.

For P6LoWPAN, we propose that header decompression be bounded to 51 bytes. This bound allows for significant RAM savings in implementations that decompress first fragments into the same buffer in which the fragment was originally held. 51 bytes is a good tradeoff between RAM savings and how frequently we expect such a bound would force packets to be sent uncompressed. A 51 byte limit

allows for transmission of a packet containing a maximally compressed IP header (+38 bytes), a maximally compressed UDP header (+6 bytes), and one maximally compressed IPv6 extension header (+7 bytes). This allows saving hundreds of bytes of RAM, without jeopardizing interoperability. Packets requiring more decompression than this are extremely rare, and could be sent uncompressed. How rare? It is only possible to surpass this limit if tunneled IPv6 is used or multiple IPv6 extension headers are present. As of 2014, a real-world study of IPv6 extension header use found that 99% of packets with _multiple_ extension headers were dropped in the real Internet, as published at IETF 90.

Second, we propose that headers for tunneled IPv6 should not be compressed. The primary motivation for this feature was from the RPL protocol [RFC6550], as discussed in Section 3. However, the fact that RPL must tunnel IPv6 in this way is generally agreed to be a problem and a wart in its design that should be avoided when possible. This change allows implementations to avoid recursive functions to decompress these headers, and instead use simple if/else statements.

7. Evaluation

This section evaluates the costs of applying our principles to 6LoWPAN. The principles are written such that interoperability comes by construction, and thus interoperability of the modified protocol cannot be directly evaluated without observing implementations written by different stakeholders. But indirect evaluation is possible. Can a reasonable set of capability levels provide a good range of implementation complexity from which a developer can choose? Is the overhead of the proposed mechanisms low enough to make them viable? Are the savings afforded by a linear capability spectrum worth the associated limitations? We find the incremental costs of capability discovery mechanisms is small, adding 172–388 bytes of code in the worst case. We find that the capability spectrum allows meaningful savings in code size and memory usage. Finally, we find capability discovery has a low run-time performance cost when a linear spectrum is used.

7.1. Implementations

First, we implemented the proposed P6LoWPAN on the Contiki-NG 6LoWPAN stack, modifying it such that a compile-time option determines which features of 6LoWPAN are compiled. We selected Contiki-NG because it has the smallest 6LoWPAN stack of those tested, so any overheads the mechanisms introduce would be most pronounced. Our changes required modifying 500 lines of code relative to the head of the 4.2 release of Contiki-NG. We did not add additional 6LoWPAN features which were

absent from the original Contiki-NG 6LoWPAN stack. Our code size numbers therefore represent a conservative lower bound of the total possible savings. All code sizes provided in this section are compiled with the Texas Instruments CC2650 as the target.

We also added ICMP and ND support for capability discovery. The updated stack responds to incompatible 6LoWPAN messages with an ICMP error, and communicates its capability level in Router Solicitation messages using the 6CIO prefix originally defined in [RFC7400]. It stores the capability class of each node in its link table, and compresses IPv6 packets by the maximum amount supported by the destination.

Finally, we implemented a second modified 6LoWPAN stack in Contiki-NG, which does not follow the recommendation of using a linear capability spectrum. In this modified implementation, each node can select any of the 6LoWPAN features it chooses. We refer to this implementation as FLEX-6LoWPAN. For this alternative policy, we isolated 26 features of 6LoWPAN as single bit flags in a 32 bit bitfield. Thus, FLEX-6LoWPAN stores and communicates capabilities using 4 byte objects. FLEX-6LOWPAN also supports the added granularity required to maximally compress outgoing messages intended for a device supporting any specific combination of features. We did not add back in any 6LoWPAN features which the Contiki-NG stack did not originally support. This second implementation required modifying about 300 additional lines of code from the P6LoWPAN implementation.

7.2. Compile-Time Costs

Figure 5 shows the size of the original Contiki-NG 6LoWPAN stack compiled at each possible capability level. Each capability level adds between 0.25 and 1.05 kB of code, and the spectrum enables implementations to cut the size of the 6LoWPAN stack by up to 45%. The code size cost of adding capability discovery, using the P6LoWPAN implementation with the linear capability spectrum, is shown in Figure 6. Capability discovery adds 178-388 bytes, a fraction of the size which implementations can save by supporting lower capability levels. The code added for communication varies across capability levels because the number of code paths for ICMP error generation and compression changes.

Capability	Code Size (kB)	Increase (kB)
Level 0	3.2	\-
Level 1	4.2	1.0
Level 2	4.8	0.6
Level 3	5.1	0.3
Level 4	5.6	0.5
Level 5	6.2	0.6

Figure 5: 6LoWPAN code size of different capabilities levels in Contiki-NG. The spectrum spans a nearly 100% increase in code size.

Capability	Base	w/Discovery	Increase
Level 0	3.2	3.4	188 bytes
Level 1	4.2	4.4	260 bytes
Level 2	4.8	5.2	388 bytes
Level 3	5.1	5.4	340 bytes
Level 4	5.6	5.9	296 bytes
Level 5	6.2	6.3	172 bytes

Figure 6: The cost of implementing capability discovery in Contiki-NG is on average less than 5% of the total 6LoWPAN size; the maximum size reduction from choosing a lower capability level is 10x the discovery cost.

Figure 7 presents the compile-time costs of using an arbitrary bitfield instead of a linear capability spectrum by comparing our P6LoWPAN implementation with our FLEX-6LoPWAN implementation. The bitfield approach requires 32 bits per neighbor to store capabilities, instead of 3 bits. More importantly, it complicates determining the allowable compression between two nodes, as demonstrated by the code size increase. The important takeaway here is that opting for a less restrictive set of feature combinations mitigates much of the savings provided by implementing capabilities. For example, a FLEX-6LoWPAN device with the equivalent of level 4 capabilities requires more code space than a level 5 P6LoWPAN device - the linear capability spectrum makes a difference. The code size addition for FLEX-6LoWPAN is a conservative lower bound, as we did not need to add checks for handling 6LoWPAN compression features that Contiki-NG does not support.

7.3. Run-time Performance

ND Cost 6LoWPAN ND communication [RFC6775] is host-initiated and flows through routers (which must be level 5), and nodes store neighbor capability levels alongside link layer addresses: thus there is no possibility of communication failures due to capability mismatches. Therefore the cost of capability discovery in networks that use IPv6 ND is exclusively that certain ND messages become longer (router solicitations and neighbor advertisements are sent with an added capability option). To put this added cost in perspective, The equation below shows the total link-layer payload bytes sent/received for ND by a node in its initial wake-up period. This equation assumes the configuration described in RFC 6775 as the "Basic Router Solicitation Exchange" - route over topology, 1 6LoWPAN context, 1 on-link prefix, and the host requires address registration. All variables not affected by the use of capability discovery are assigned the minimum possible value for the scenario discussed, so that the overhead of capability discovery represents a worst case.

If C = total link layer payload sent/recieved for ND, and N = endpoints requiring address resolution:

C = Router Soliciation {RS} + Min. IP Hdr {2} + Router Advertisement {104} + Min. IP Hdr {2} + (Neighbour Solicitation {24} + Min. IP Hdr {2})*N + (Neighbour Advertisement {24} + Min. IP Hdr {2})*N + Address Registration Options in first NS {24} + Address Registration Options in first NA {16}.

Figure 8 shows the values of RS and NA for each 6LoWPAN implementation, and the resulting total ND cost. Notably, use of an arbitrary bitfield increases the size of the capability option by 4 bytes, making use of existing ND options like the 6CIO option impossible. In both cases the additional bytes added for capability discovery are small compared to the total cost of ND(<= 8% linear spectrum / <= 16% arbitrary bitfield).

ICMP Cost In networks that do not use IPv6 ND the cost of capability discovery is the energy/latency required for one ICMP packet per failure between any two nodes. For P6LoWPAN capability based failures can only happen in one direction, so the size of this link-layer payload is:

C_icmp = Compressed IP Header Size + 4

For FLEX-6LoWPAN C_icmp = 48, because the recipient does not know the capabilities of the sender, and thus must send an uncompressed packet to ensure successful reception of its own capabilities. This example

reveals why use of an arbitrary bitfield is so undesirable - the ability to compress headers in ICMP errors can reduce overhead by a factor of 4 or more (in the common case of 8 byte compressed headers).

--	Linear Spectrum	Arbitrary Bitfield
6LoWPAN Code Size	5.9 kB	6.5 kB
RAM per neighbor	19 Bytes	22 Bytes

Figure 7: Resource requirements for a 6LoWPAN stack in Contiki-NG using a linear capability spectrum vs. using an arbitrary capability bitfield.

--	RS	NA	C (Total ND Cost)
6LoWPAN	20	24	168 + 52*N
P6LoWPAN	24	28	172+56*N
FLEX-6LoWPAN	28	32	176+60*N

Figure 8: Total ND cost for each implementation

8. Discussion and Conclusions

A new generation of low-power devices face a connectivity dilemma: Internet protocols are not designed for energy efficiency, but compression and other energy saving adaptations takes up precious code space. Device deployments specialized for single-vendor local networks make trade-offs specific to their application requirements. As a result, IP communication between IP enabled devices fails. This problem is not specific to 6LoWPAN -- Iova et. al. recently noted similar issues in the RPL protocol: "RPL has too large of a footprint for resource-constrained devices, and requires all devices in a network to run the same mode of operation, limiting heterogeneity" [iova].

Part of the challenge is that some traditional protocol design principles do not apply well to the low-power setting. We present three design principles for low-power protocols that attempt to remedy this. These principles explicitly acknowledge the unique code space/energy tradeoffs of low-power devices.

Looking forward, considering this tension is critical for protocol designers in this ecosystem of diverse hardware capabilities and

application tradeoffs. 6LoWPAN is not the only low power Internet protocol -- the low power space uses its own routing protocols, address discovery protocols, and application layer protocols [RFC6550] [RFC7252]. Additional protocols will follow as the space matures. Many of these protocols will be initially developed outside the IETF -- Jonathan Hui was a graduate student when he presented the first complete IPv6-based network architecture for sensor nets [hui], as was Adam Dunkels when he created Contiki. We present a roadmap for how these principles can reframe the discussion of how to connect the next hundred billion devices to the Internet.

9. Definitions

9.1. Terminology

In this document, the key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" are to be interpreted as described in BCP 14, RFC 2119 [RFC2119].

10. Security Considerations

This informational document does have some implications for security if followed.

First, capability advertisements of the type recommended in this document are liable to leak some information regarding the type of device sending those advertisements. In any situation for which this information is privileged, such advertisements must be suppressed.

Second, implementations should be careful not to take for granted that the suggestions in this document will be implemented by all other transmitting devices. Accordingly, though this document recommends reasonable bounds, receivers still must be careful to prevent buffer overflows in the event these bounds are not followed.

11. IANA Considerations

This document has no actions for IANA.

12. References

12.1. Normative References

[RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, DOI 10.17487/RFC4861, September 2007, <https://www.rfc-editor.org/info/rfc4861>.

[RFC4944] Montenegro, G., Kushalnagar, N., Hui, J., and D. Culler, "Transmission of IPv6 Packets over IEEE 802.15.4 Networks", RFC 4944, DOI 10.17487/RFC4944, September 2007, <https://www.rfc-editor.org/info/rfc4944>.

[RFC6282] Hui, J., Ed. and P. Thubert, "Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks", RFC 6282, DOI 10.17487/RFC6282, September 2011, <https://www.rfc-editor.org/info/rfc6282>.

[RFC6775] Shelby, Z., Ed., Chakrabarti, S., Nordmark, E., and C. Bormann, "Neighbor Discovery Optimization for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs)", RFC 6775, DOI 10.17487/RFC6775, November 2012, <https://www.rfc-editor.org/info/rfc6775>.

12.2. Informative References

[contiki] Dunkels, A., "Contiki OS", n.d., <http://www.contiki-os.org/>.

[contiki-ng] Duquennoy, S., "Contiki-NG", n.d., <https://github.com/contiki-ng/contiki-ng>.

[end-to-end] Clark, D., "End-to-end Arguments in System Design", n.d..

[hui] Culler, D., "IP is Dead, Long Live IP for Wireless Sensor Networks", n.d..

[iova] Kiraly, C., "RPL The Routing Standard for the Internet of Things...Or Is It?", n.d..

[lite-os] Huawei, ., "LiteOS", n.d., <https://liteos.github.io/>.

[mbedos] ARM, ., "ARM MbedOS", n.d., <https://os.mbed.com/>.

[probe-it-plugtest] Huang, X., "Tehcnical Interoperability 6LoWPAN-CoAP Report from Interop Event", n.d..

[RFC0791] Postel, J., "Internet Protocol", STD 5, RFC 791, DOI 10.17487/RFC0791, September 1981, <https://www.rfc-editor.org/info/rfc791>.

[RFC0793] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, DOI 10.17487/RFC0793, September 1981, <https://www.rfc-editor.org/info/rfc793>.

[RFC1122] Braden, R., Ed., "Requirements for Internet Hosts - Communication Layers", STD 3, RFC 1122, DOI 10.17487/RFC1122, October 1989, <https://www.rfc-editor.org/info/rfc1122>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC6550] Winter, T., Ed., Thubert, P., Ed., Brandt, A., Hui, J., Kelsey, R., Levis, P., Pister, K., Struik, R., Vasseur, JP., and R. Alexander, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks", RFC 6550, DOI 10.17487/RFC6550, March 2012, <https://www.rfc-editor.org/info/rfc6550>.

[RFC7252] Shelby, Z., Hartke, K., and C. Bormann, "The Constrained Application Protocol (CoAP)", RFC 7252, DOI 10.17487/RFC7252, June 2014, <https://www.rfc-editor.org/info/rfc7252>.

[RFC7400] Bormann, C., "6LoWPAN-GHC: Generic Header Compression for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs)", RFC 7400, DOI 10.17487/RFC7400, November 2014, <https://www.rfc-editor.org/info/rfc7400>.

[RFC7668] Nieminen, J., Savolainen, T., Isomaki, M., Patil, B., Shelby, Z., and C. Gomez, "IPv6 over BLUETOOTH(R) Low Energy", RFC 7668, DOI 10.17487/RFC7668, October 2015, <https://www.rfc-editor.org/info/rfc7668>.

[riot] Berlin, F., "riot OS", n.d., <https://www.riot-os.org/>.

[RPL-interop] Ko, J., "Contikierpl and tinyrpl - Happy Together", n.d..

[tinyos] Levis, P., "TinyOS An Operating System for Sensor Networks", n.d., <https://link.springer.com/chapter/10.1007/3-540-27139-2_7>.

[tock] Levy, A., "Multiprogramming a 64kB Computer Safely and Efficiently", n.d..

[zephyr] Foundation, T., "zephyrOS", n.d., <https://www.zephyrproject.org/>.

Authors' Addresses

Hudson Ayers
Stanford University
350 Serra Mall
Stanford, CA
United States

Email: hayers@stanford.edu

Philip Levis
Stanford University
350 Serra Mall
Stanford, CA
United States

Email: pal@cs.stanford.edu

6lo Working Group
Internet-Draft
Intended status: Experimental
Expires: January 14, 2021

C. Gomez
UPC

July 13, 2020

IPv6 over Low-Power Wireless Personal Area Network (6LoWPAN) Dispatch Type for SCHC
draft-gomez-6lo-schc-dispatch-00

Abstract

A new framework called Static Context Header Compression (SCHC) has been designed to support IPv6 over Low Power Wide Area Network (LPWAN) technologies [RFC8724]. One of the SCHC components is a header compression mechanism. If used properly, SCHC header compression allows a greater compression ratio than that achievable with traditional 6LoWPAN header compression [RFC6282]. For this reason, it may make sense to use SCHC header compression in some 6LoWPAN environments. In its current form, this document proposes a number of 6LoWPAN Dispatch type approaches to signal when a packet header has been compressed by using SCHC header compression.

Table of Contents

1. Introduction

RFC 6282 is the main specification for IPv6 over Low power Wireless Personal Area Network (6LoWPAN) IPv6 header compression [RFC6282]. This RFC was designed assuming IEEE 802.15.4 as the layer below the 6LoWPAN adaptation layer, and it has also been reused (with proper adaptations) for IPv6 header compression over many other technologies relatively similar to IEEE 802.15.4 in terms of characteristics such as physical layer bit rate, layer 2 maximum payload size, etc. Examples of such technologies comprise BLE, DECT-ULE, ITU G.9959, MS/TP, NFC, and PLC. RFC 6282 provides additional functionality, such as a mechanism for UDP header compression.

In the best case, RFC 6282 allows to compress a 40-byte IPv6 header down to a 2-byte compressed header (in link-local interactions) or a 3-byte compressed header (when global IPv6 addresses are used). On the other hand, an RFC 6282 compressed UDP header has a typical size of 4 bytes. Therefore, in advantageous conditions, a 48-byte uncompressed IPv6/UDP header may be compressed down to a 7-byte format by using RFC 6282.

Recently, a new framework called Static Context Header Compression (SCHC) has been designed to support IPv6 over Low Power Wide Area Network (LPWAN) technologies [RFC8724]. SCHC comprises header

compression and fragmentation functionality tailored to the extraordinary constraints of LPWAN technologies, which are more severe than those exhibited by IEEE 802.15.4 or other relatively similar technologies.

SCHC header compression allows a greater compression ratio than that of RFC 6282. If used properly, SCHC allows to compress an IPv6/UDP header down to e.g. a single byte. Therefore, it may make sense to use SCHC header compression in some 6LoWPAN environments [I-D.toutain-6lo-6lo-and-schc], considering its greater efficiency.

If SCHC header compression is added to the panoply of header compression mechanisms used in 6LoWPAN environments, then there is a need to signal when a packet header has been compressed by using SCHC. To this end, in its current form, the present document proposes a number of 6LoWPAN Dispatch type approaches for SCHC header compression, based on exploiting RFC 4944 and/or RFC 8025 Dispatch type space [RFC4944][RFC8025].

2. Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL","SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

3. Frame Format

Figure 1 illustrates the content of an encapsulated, SCHC compressed, IPv6 datagram:

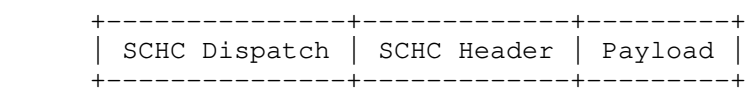

Figure 1: Encapsulated, SCHC compressed IPv6 datagram

The SCHC Header corresponds to a packet header that has been compressed by using SCHC. As defined in [RFC8724], the SCHC Header comprises a Rule ID, and a compression residue. (Note: more details, including a discussion on padding, to be added.)

4. SCHC Dispatch Type Approaches

This section presents 3 different approaches for the SCHC Dispatch type to be discussed.

4.1. Approach 1

A first approach for the SCHC Dispatch Pattern is using Not a LoWPAN (NALP) Dispatch type space [RFC4944]. The first two bits in a NALP Dispatch type are 00. Approach 1 defines that a Dispatch starting by "001" indicates that a SCHC-compressed packet comes next.

SCHC Dispatch Pattern: 001XXXXX

The last 5 bits of the Dispatch (indicated as 'X' above) may be used to define 32 different Rule IDs.

This approach has pros and cons. A single byte is used for the Dispatch plus the Rule ID. However, is 32 a relatively low number of possible Rule ID values? On the other hand, there may be backwards compatibility issues with existing implementations, where SCHC-compressed packets might be misunderstood as other types of packets.

4.2. Approach 2

A second approach, that also uses NALP Dispatch type space, is:

SCHC Dispatch Pattern: 001YYYYY YYYYYYYY

The last 13 bits of the Dispatch (indicated as 'Y' above) may be used to define 8192 different Rule IDs.

With this approach, two bytes are used for the SCHC Dispatch plus the Rule ID, but 8192 possible Rule IDs can be used. The same backwards compatibility issues in Approach 1 may exist for Approach 2 as well.

4.3. Approach 3

A third approach, which is not based on using NALP space, is using the RFC 8025 concept of "pages", which would allocate one page for SCHC-compressed headers:

SCHC Dispatch Pattern: 1111ZZZZ (ZZZZ to be determined)

With this approach, and with the aim to minimize header overhead, a whole page is allocated for the SCHC Dispatch type. A 1-byte Rule ID follows the SCHC Dispatch Pattern.

In this case, two bytes are used for the SCHC Dispatch plus the Rule ID. 256 possible Rule IDs can be used. There are no backwards compatibility issues.

5. IANA Considerations

TBD

6. Security Considerations

TBD

7. Acknowledgments

Ana Minaburo and Laurent Toutain suggested for the first time the use of SCHC in environments where 6LoWPAN has traditionally been used.

Carles Gomez has been funded in part by the Spanish Government through project PID2019-106808RA-I00, and by Secretaria d'Universitats i Recerca del Departament d'Empresa i Coneixement de la Generalitat de Catalunya 2017 through grant SGR 376.

8. References

8.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC4944] Montenegro, G., Kushalnagar, N., Hui, J., and D. Culler, "Transmission of IPv6 Packets over IEEE 802.15.4 Networks", RFC 4944, DOI 10.17487/RFC4944, September 2007, <https://www.rfc-editor.org/info/rfc4944>.

[RFC6282] Hui, J., Ed. and P. Thubert, "Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks", RFC 6282, DOI 10.17487/RFC6282, September 2011, <https://www.rfc-editor.org/info/rfc6282>.

[RFC8025] Thubert, P., Ed. and R. Cragie, "IPv6 over Low-Power Wireless Personal Area Network (6LoWPAN) Paging Dispatch", RFC 8025, DOI 10.17487/RFC8025, November 2016, <https://www.rfc-editor.org/info/rfc8025>.

[RFC8724] Minaburo, A., Toutain, L., Gomez, C., Barthel, D., and JC. Zuniga, "SCHC: Generic Framework for Static Context Header Compression and Fragmentation", RFC 8724, DOI 10.17487/RFC8724, April 2020, <https://www.rfc-editor.org/info/rfc8724>.

8.2. Informative References

[I-D.toutain-6lo-6lo-and-schc]
Minaburo, A. and L. Toutain, "Comparison of 6lo and SCHC", draft-toutain-6lo-6lo-and-schc-00 (work in progress), November 2019.

Author's Address

Carles Gomez
UPC
C/Esteve Terradas, 7
Castelldefels 08860
Spain

Email: carlesgo@entel.upc.edu

6Lo Working Group
Internet-Draft
Intended status: Standards Track
Expires: December 5, 2020

J. Hou
B. Liu
Huawei Technologies
Y-G. Hong
ETRI
X. Tang
SGEPRI
C. Perkins

June 3, 2020

Transmission of IPv6 Packets over PLC Networks
draft-ietf-6lo-plc-04

Abstract

Power Line Communication (PLC), namely using the electric-power lines for indoor and outdoor communications, has been widely applied to support Advanced Metering Infrastructure (AMI), especially smart meters for electricity. The inherent advantage of existing electricity infrastructure facilitates the expansion of PLC deployments, and moreover, a wide variety of accessible devices raises the potential demand of IPv6 for future applications. This document describes how IPv6 packets are transported over constrained PLC networks, such as ITU-T G.9903, IEEE 1901.1 and IEEE 1901.2.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on December 5, 2020.

Table of Contents

1. Introduction

The idea of using power lines for both electricity supply and communication can be traced back to the beginning of the last century. With the advantage of existing power grid, Power Line Communication (PLC) is a good candidate for supporting various service scenarios such as in houses and offices, in trains and vehicles, in smart grid and advanced metering infrastructure (AMI). The data acquisition devices in these scenarios share common features

such as fixed position, large quantity, low data rate and low power consumption.

Although PLC technology has evolved over several decades, it has not been fully adapted for IPv6 based constrained networks. The 6lo related scenarios lie in the low voltage PLC networks with most applications in the area of Advanced Metering Infrastructure (AMI), Vehicle-to-Grid communications, in-home energy management and smart street lighting. IPv6 is important for PLC networks, due to its large address space and efficent address auto-configuration. A comparison among various existing PLC standards is provided to facilitate the selection of the most applicable standard in particular scenarios.

This specification provides a brief overview of PLC technologies. Some of them have LLN characteristics, i.e. limited power consumption, memory and processing resources. This specification is focused on the transmission of IPv6 packets over those "constrained" PLC networks. The general approach is to adapt elements of the 6LoWPAN specifications [RFC4944], [RFC6282], and [RFC6775] to constrained PLC networks. There was work previously proposed as [I-D.popa-6lo-6loplc-ipv6-over-ieee19012-networks], which did not reach consensus. This document provides a more structured specification than the previous work, expanding to a larger variety of PLC networks.

2. Requirements Notation and Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

This document often uses the following acronyms and terminologies:

6LoWPAN: IPv6 over Low-Power Wireless Personal Area Network

AMI: Advanced Metering Infrastructure

BBPLC: Broadband Power Line Communication

CID: Context ID

Coordinator: A device capable of relaying messages.

DAD: Duplicate Address Detection

EV: Electric Vehicle

IID: IPv6 Interface Identifier

IPHC: IP Header Compression

LAN: Local Area Network

MSDU: MAC Service Data Unit

MTU: Maximum Transmission Unit

NBPLC: Narrowband Power Line Communication

OFDM: Orthogonal Frequency Division Multiplexing

PANC: PAN Coordinator, a coordinator which also acts as the primary controller of a PAN.

PLC: Power Line Communication

PLC device: An entity follows the PLC standards and implements the protocol stack described in this draft.

PSDU: PHY Service Data Unit

RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks

RA: Router Advertisement

WAN: Wide Area Network

The terminology used in this draft is aligned with IEEE 1901.2

IEEE 1901.2	IEEE 1901.1	ITU-T G.9903	This document
PAN Coordinator	Central Coordinator	PAN Coordinator	PAN Coordinator
Coordinator	Proxy Coordinator	Full-function device	Coordinator
Device	Station	PAN Device	PLC Device

Table 1: Terminology Mapping between PLC standards

3. Overview of PLC

PLC technology enables convenient two-way communications for home users and utility companies to monitor and control electric plugged devices such as electricity meters and street lights. Due to the large range of communication frequencies, PLC is generally classified into two categories: Narrowband PLC (NBPLC) for automation of sensors (which have low frequency band and low power cost), and Broadband PLC (BBPLC) for home and industry networking applications.

Various standards have been addressed on the MAC and PHY layers for this communication technology, e.g., BBPLC (1.8-250 MHz) including IEEE 1901 and ITU-T G.hn, and NBPLC (3-500 kHz) including ITU-T G.9902 (G.hnem), ITU-T G.9903 (G3-PLC) [ITU-T_G.9903], ITU-T G.9904 (PRIME), IEEE 1901.2 [IEEE_1901.2] (combination of G3-PLC and PRIME PLC) and IEEE 1901.2a [IEEE_1901.2a] (an amendment to IEEE 1901.2).

Moreover, recently a new PLC standard IEEE 1901.1 [IEEE_1901.1], which aims at the medium frequency band less than 12 MHz, has been published by the IEEE standard for Smart Grid Powerline Communication Working Group (SGPLC WG). IEEE 1901.1 balances the needs for bandwidth versus communication range, and is thus a promising option for 6lo applications.

This specification is focused on IEEE 1901.1, IEEE 1901.2 and ITU-T G.9903.

3.1. Protocol Stack

The protocol stack for IPv6 over PLC is illustrated in Figure 1. The PLC MAC/PHY layer corresponds to IEEE 1901.1, IEEE 1901.2 or ITU-T G.9903. The 6lo adaptation layer for PLC is illustrated in Section 4. For multihop tree and mesh topologies, a routing protocol is likely to be necessary. The routes can be built in mesh-under mode at layer 2 or in route-over mode at layer 3, as explained in Section 3.4.

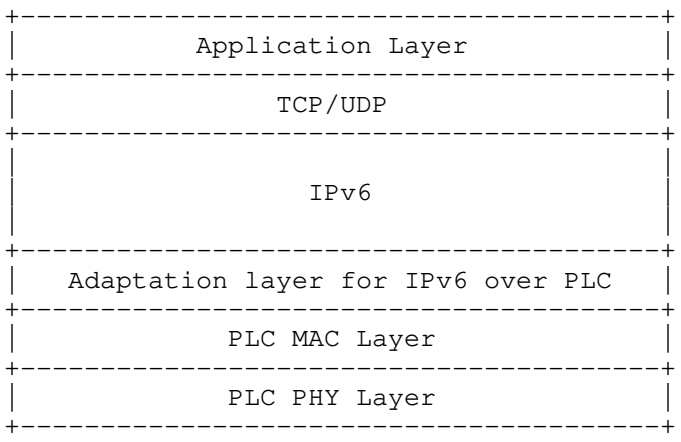

Figure 1: PLC Protocol Stack

3.2. Addressing Modes

Each PLC device has a globally unique long address of 48-bit ([IEEE_1901.1]) or 64-bit ([IEEE_1901.2], [ITU-T_G.9903]) and a short address of 12-bit ([IEEE_1901.1]) or 16-bit ([IEEE_1901.2], [ITU-T_G.9903]). The long address is set by the manufacturer according to the IEEE EUI-48 MAC address or the IEEE EUI-64 address. Each PLC device joins the network by using the long address and communicates with other devices by using the short address after joining the network. Short addresses can be assigned during the onboarding process, by the PANC or the JRC in CoJP [I-D.ietf-6tisch-minimal-security].

3.3. Maximum Transmission Unit

The Maximum Transmission Unit (MTU) of the MAC layer determines whether fragmentation and reassembly are needed at the adaptation layer of IPv6 over PLC. IPv6 requires an MTU of 1280 octets or greater; thus for a MAC layer with MTU lower than this limit, fragmentation and reassembly at the adaptation layer are required.

The IEEE 1901.1 MAC supports upper layer packets up to 2031 octets. The IEEE 1901.2 MAC layer supports the MTU of 1576 octets (the original value of 1280 bytes was updated in 2015 [IEEE_1901.2a]). Though these two technologies can support IPv6 natively without fragmentation and reassembly, it is possible to configure a smaller MTU in high-noise communication environment. Thus the 6lo functions, including header compression, fragmentation and reassembly, are still applicable and useful.

The MTU for ITU-T G.9903 is 400 octets, insufficient for supporting IPv6's MTU. For this reason, fragmentation and reassembly as per [RFC4944] MUST be enabled for G.9903-based networks.

3.4. Routing Protocol

Routing protocols suitable for use in PLC networks include:

- RPL (Routing Protocol for Low-Power and Lossy Networks) [RFC6550] is a layer 3 routing protocol. AODV-RPL [I-D.ietf-roll-aodv-rpl] updates RPL to include reactive, point-to-point, and asymmetric routing. IEEE 1901.2 specifies Information Elements (IEs) with MAC layer metrics, which can be provided to L3 routing protocol for parent selection. For IPv6-addressable PLC networks, a layer-3 routing protocol such as RPL and/or AODV-RPL SHOULD be supported in the standard.

- IEEE 1901.1 supports L2 routing. Each PLC node maintains a L2 routing table, in which each route entry comprises the short addresses of the destination and the related next hop. The route entries are built during the network establishment via a pair of association request/confirmation messages. The route entries can be changed via a pair of proxy change request/confirmation messages. These association and proxy change messages MUST be approved by the central coordinator (PANC in this document).

- LOADng is a reactive protocol operating at layer 2 or layer 3. Currently, LOADng is supported in ITU-T G.9903 [ITU-T_G.9903], and the IEEE 1901.2 standard refers to ITU-T G.9903 for LOAD-based networks.

4. IPv6 over PLC

6LoWPAN standards [RFC4944], [RFC6775], and [RFC6282] provides useful functionality including link-local IPv6 addresses, stateless address auto-configuration, neighbor discovery and header compression. However, due to the different characteristics of the PLC media, the 6LoWPAN adaptation layer cannot perfectly fulfill the requirements. These considerations suggest the need for a dedicated adaptation layer for PLC, which is detailed in the following subsections.

4.1. Stateless Address Autoconfiguration

To obtain an IPv6 Interface Identifier (IID), a PLC device performs stateless address autoconfiguration [RFC4944]. The autoconfiguration can be based on either a long or short link-layer address.

The IID can be based on the device's 48-bit MAC address or its EUI-64 identifier [EUI-64]. A 48-bit MAC address MUST first be extended to a 64-bit Interface ID by inserting 0xFFFE at the fourth and fifth octets as specified in [RFC2464]. The IPv6 IID is derived from the 64-bit Interface ID by inverting the U/L bit [RFC4291].

For IEEE 1901.2 and ITU-T G.9903, a 48-bit "pseudo-address" is formed by the 16-bit PAN ID, 16 zero bits and the 16-bit short address. Then, the 64-bit Interface ID MUST be derived by inserting 16-bit 0xFFFE into as follows:

16_bit_PAN:00FF:FE00:16_bit_short_address

For the 12-bit short addresses used by IEEE 1901.1, the 48-bit pseudo-address is formed by 24-bit NID (Network IDentifier, YYYYYY), 12 zero bits and a 12-bit TEI (Terminal Equipment Identifier, XXX). The 64-bit Interface ID MUST be derived by inserting 16-bit 0xFFFE into this 48-bit pseudo-address as follows:

YYYY:YYFF:FE00:0XXX

Since the derived Interface ID is not global, the "Universal/Local" (U/L) bit (7th bit) and the Individual/Group bit (8th bit) MUST both be set to zero. In order to avoid any ambiguity in the derived Interface ID, these two bits MUST NOT be used to generate the PANID (for IEEE 1901.2 and ITU-T G.9903) or NID (for IEEE 1901.1). In other words, the PANID or NID MUST always be chosen so that these bits are zeros.

For privacy reasons, the IID derived by the MAC address SHOULD only be used for link-local address configuration. A PLC host SHOULD use the IID derived by the link-layer short address to configure the IPv6 address used for communication with the public network; otherwise, the host's MAC address is exposed. Implementations should look at [RFC8064] as well, in order to generate a stable IPv6 address using an opaque IID.

4.2. IPv6 Link Local Address

The IPv6 link-local address [RFC4291] for a PLC interface is formed by appending the IID, as defined above, to the prefix FE80::/64 (see Figure 2).

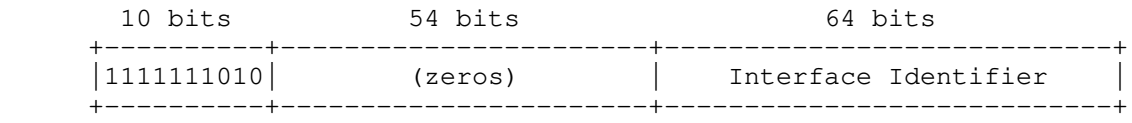

Figure 2: IPv6 Link Local Address for a PLC interface

4.3. Unicast Address Mapping

The address resolution procedure for mapping IPv6 unicast addresses into PLC link-layer addresses follows the general description in section 7.2 of [RFC4861]. [RFC6775] improves this procedure by eliminating usage of multicast NS. The resolution is realized by the NCEs (neighbor cache entry) created during the address registration at the routers. [RFC8505] further improves the registration procedure by enabling multiple LLNs to form an IPv6 subnet, and by inserting a link-local address registration to better serve proxy registration of new devices.

4.3.1. Unicast Address Mapping for IEEE 1901.1

The Source/Target Link-layer Address options for IEEE_1901.1 used in the Neighbor Solicitation and Neighbor Advertisement have the following form.

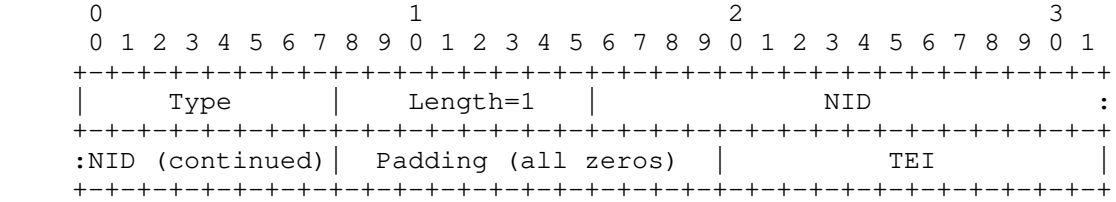

Figure 3: Unicast Address Mapping for IEEE 1901.1

Option fields:

Type: 1 for Source Link-layer Address and 2 for Target Link-layer Address.

Length: The length of this option (including type and length fields) in units of 8 octets. The value of this field is 1 for the 12-bit IEEE 1901.1 PLC short addresses.

NID: 24-bit Network IDentifier

Padding: 12 zero bits

TEI: 12-bit Terminal Equipment Identifier

In order to avoid the possibility of duplicated IPv6 addresses, the value of the NID MUST be chosen so that the 7th and 8th bits of the first byte of the NID are both zero.

4.3.2. Unicast Address Mapping for IEEE 1901.2 and ITU-T G.9903

The Source/Target Link-layer Address options for IEEE_1901.2 and ITU-T G.9903 used in the Neighbor Solicitation and Neighbor Advertisement have the following form.

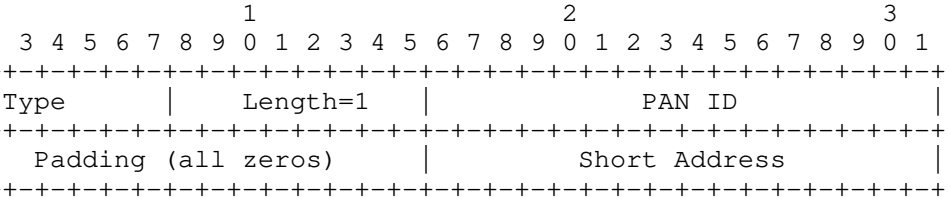

Figure 4: Unicast Address Mapping for IEEE 1901.2

Option fields:

Type: 1 for Source Link-layer Address and 2 for Target Link-layer Address.

Length: The length of this option (including type and length fields) in units of 8 octets. The value of this field is 1 for the 16-bit IEEE 1901.2 PLC short addresses.

PAN ID: 16-bit PAN IDentifier

Padding: 16 zero bits

Short Address: 16-bit short address

In order to avoid the possibility of duplicated IPv6 addresses, the value of the PAN ID MUST be chosen so that the 7th and 8th bits of the first byte of the PAN ID are both zero.

4.4. Neighbor Discovery

Neighbor discovery procedures for 6LoWPAN networks are described in Neighbor Discovery Optimization for 6LoWPANs [RFC6775] and [RFC8505]. These optimizations support the registration of sleeping hosts. Although PLC devices are electrically powered, sleeping mode SHOULD still be used for power saving.

For IPv6 address prefix dissemination, Router Solicitations (RS) and Router Advertisements (RA) MAY be used as per [RFC6775]. If the PLC network uses route-over mesh, the IPv6 prefix MAY be disseminated by the layer 3 routing protocol, such as RPL which includes the prefix in the DIO message. As per [I-D.ietf-roll-unaware-leaves], it is possible to have PLC devices configured as RPL-unaware-leaves, which don't not participate to RPL at all, along with RPL-aware PLC devices. In this case, the prefix dissemination SHOULD use the RS/RA messages.

For context information dissemination, Router Advertisements (RA) MUST be used as per [RFC6775]. The 6LoWPAN context option (6CO) MUST be included in the RA to disseminate the Context IDs used for prefix compression.

For address registration in route-over mode, a PLC device MUST register its addresses by sending unicast link-local Neighbor Solicitation to the 6LR. If the registered address is link-local, the 6LR SHOULD NOT further register it to the registrar (6LBR, 6BBR). Otherwise, the address MUST be registered via an ARO or EARO included in the DAR ([RFC6775]) or EDAR ([RFC8505]) messages. For RFC8505 compliant PLC devices, the 'R' flag in the EARO MUST be set when sending Neighbor Solicitaitons in order to extract the status information in the replied Neighbor Advertisements from the 6LR. If DHCPv6 is used to assign addresses or the IPv6 address is derived by unique long or short link layer address, Duplicate Address Detection (DAD) MUST NOT be utilized. Otherwise, the DAD MUST be performed at the 6LBR (as per [RFC6775]) or proxied by the routing registrar (as per [RFC8505]). The registration status is feedbacked via the DAC or EDAC message from the 6LBR and the Neighbor Advertisement (NA) from the 6LR.

For address registration in mesh-under mode, since all the PLC devices are the link-local neighbors to the 6LBR, DAR/DAC or EDAR/ EDAC messages are not required. A PLC device MUST register its addresses by sending the unicast NS message with an ARO or EARO. The registration status is feedbacked via the NA message from the 6LBR.

4.5. Header Compression

The compression of IPv6 datagrams within PLC MAC frames refers to [RFC6282], which updates [RFC4944]. Header compression as defined in [RFC6282] which specifies the compression format for IPv6 datagrams on top of IEEE 802.15.4, is included in this document as the basis for IPv6 header compression in PLC. For situations when PLC MAC MTU cannot support the 1280-octet IPv6 packet, headers MUST be compressed according to [RFC6282] encoding formats.

4.6. Fragmentation and Reassembly

PLC differs from other wired technologies in that the communication medium is not shielded; thus, to successfully transmit data through power lines, PLC Data Link layer provides the function of segmentation and reassembly. A Segment Control Field is defined in the MAC frame header regardless of whether segmentation is required. The number of data octets of the PHY payload can change dynamically based on channel conditions, thus the MAC payload segmentation in the MAC sublayer is enabled and guarantees a reliable one-hop data transmission. Fragmentation and reassembly is still required at the adaptation layer, if the MAC layer cannot support the minimum MTU demanded by IPv6, which is 1280 octets.

In IEEE 1901.1 and IEEE 1901.2, the MAC layer supports payloads as big as 2031 octets and 1576 octets respectively. However when the channel condition is noisy, it is possible to configure smaller MTU at the MAC layer. If the configured MTU is smaller than 1280 octects, the fragmentation and reassembly defined in [RFC4944] MUST be used.

In ITU-T G.9903, the maximum MAC payload size is fixed to 400 octets, so to cope with the required MTU of 1280 octets by IPv6, fragmentation and reassembly at 6lo adaptation layer MUST be provided referring to [RFC4944].

5. Internet Connectivity Scenarios and Topologies

The network model can be simplified to two kinds of network devices: PAN Coordinator (PANC) and PAN Device. The PANC is the primary coordinator of the PLC subnet and can be seen as a master node; PAN Devices are typically PLC meters and sensors. The PANC also serves as the Routing Registrar for proxy registration and DAD procedures, making use of the updated registration procedures in [RFC8505]. IPv6 over PLC networks are built as tree, mesh or star according to the use cases. Every network requires at least one PANC to communicate with each PAN Device. Note that the PLC topologies in this section are based on logical connectivity, not physical links.

The star topology is common in current PLC scenarios. In single-hop star topologies, communication at the link layer only takes place between a PAN Device and a PANC. The PANC typically collects data (e.g., a meter reading) from the PAN devices, and then concentrates and uploads the data through Ethernet or LPWAN (see Figure 5). The collected data is transmitted by the smart meters through PLC, aggregated by a concentrator, sent to the utility and then to a Meter Data Management System for data storage, analysis and billing. This

topology has been widely applied in the deployment of smart meters, especially in apartment buildings.

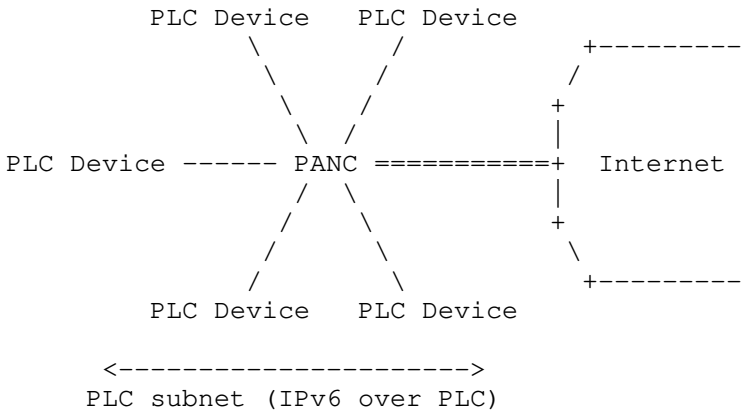

Figure 5: PLC Star Network connected to the Internet

A tree topology is useful when the distance between a device A and PANC is beyond the PLC allowed limit and there is another device B in between able to communicate with both sides. Device B in this case acts both as a PAN Device and a Coordinator. For this scenario, the link layer communications take place between device A and device B, and between device B and PANC. An example of PLC tree network is depicted in Figure 6. This topology can be applied in the smart street lighting, where the lights adjust the brightness to reduce energy consumption while sensors are deployed on the street lights to provide information such as light intensity, temperature, humidity. Data transmission distance in the street lighting scenario is normally above several kilometers thus the PLC tree network is required. A more sophisticated AMI network may also be constructed into the tree topology which is depicted in [RFC8036]. A tree topology is suitable for AMI scenarios that require large coverage but low density, e.g., the deployment of smart meters in rural areas. RPL is suitable for maintenance of a tree topology in which there is no need for communication directly between PAN devices.

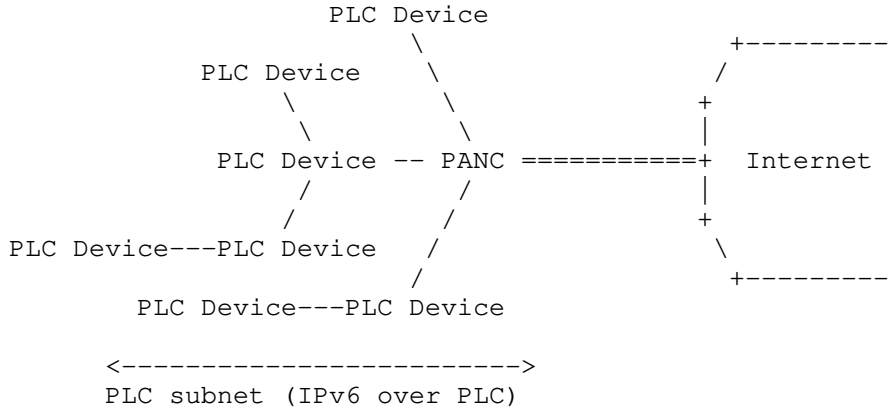

Figure 6: PLC Tree Network connected to the Internet

Mesh networking in PLC is of great potential applications and has been studied for several years. By connecting all nodes with their neighbors in communication range (see Figure 7), mesh topology dramatically enhances the communication efficiency and thus expands the size of PLC networks. A simple use case is the smart home scenario where the ON/OFF state of air conditioning is controlled by the state of home lights (ON/OFF) and doors (OPEN/CLOSE). AODV-RPL enables direct PAN device to PAN device communication, without being obliged to transmit frames through the PANC, which is a requirement often cited for AMI infrastructure.

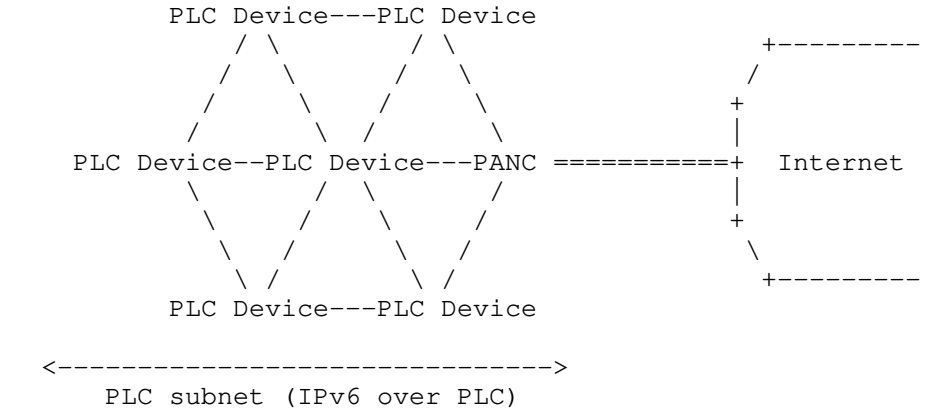

Figure 7: PLC Mesh Network connected to the Internet

6. IANA Considerations

There are no IANA considerations related to this document.

7. Security Consideration

Due to the high accessibility of power grid, PLC might be susceptible to eavesdropping within its communication coverage, e.g., one apartment tenant may have the chance to monitor the other smart meters in the same apartment building. For security consideration, link layer security is guaranteed in every PLC technology.

Malicious PLC devices could paralyze the whole network via DOS attacks, e.g., keep joining and leaving the network frequently, or multicast routing messages containing fake metrics. A device may also join a wrong or even malicious network, exposing its data to illegal users. Mutual authentication of network and new device can be conducted during the onboarding process of the new device. Methods include protocols such as [RFC7925] (exchanging pre-installed certificates over DTLS) , [I-D.ietf-6tisch-minimal-security] (which uses pre-shared keys), and [I-D.ietf-6tisch-dtsecurity-zerotouch-join] (which uses IDevID and MASA service). It is also possible to use EAP methods such as [I-D.ietf-emu-eap-noob] via transports like PANA [RFC5191]. No specific mechanism is specified by this document as an appropriate mechanism will depend upon deployment circumstances. The network encryption key appropriate for the layer-2 can also be acquired during the onboarding process.

IP addresses may be used to track devices on the Internet; such devices can in turn be linked to individuals and their activities. Depending on the application and the actual use pattern, this may be undesirable. To impede tracking, globally unique and non-changing characteristics of IP addresses should be avoided, e.g., by frequently changing the global prefix and avoiding unique link-layer derived IIDs in addresses. [RFC3315], [RFC3972], [RFC4941], [RFC5535], [RFC7217], and [RFC8065] provide valuable information for IID formation with improved privacy, and are RECOMMENDED for IPv6 networks.

8. Acknowledgements

We gratefully acknowledge suggestions from the members of the IETF 6lo working group. Great thanks to Samita Chakrabarti and Gabriel Montenegro for their feedback and support in connecting the IEEE and ITU-T sides. Authors thank Scott Mansfield, Ralph Droms, Pat Kinney for their guidance in the liaison process. Authors wish to thank

Stefano Galli, Thierry Lys, Yizhou Li, Yuefeng Wu and Michael Richardson for their valuable comments and contributions.

9. References

9.1. Normative References

[IEEE_1901.1]
IEEE-SA Standards Board, "Standard for Medium Frequency (less than 15 MHz) Power Line Communications for Smart Grid Applications", IEEE 1901.1, May 2018, <https://ieeexplore.ieee.org/document/8360785>.

[IEEE_1901.2]
IEEE-SA Standards Board, "IEEE Standard for Low-Frequency (less than 500 kHz) Narrowband Power Line Communications for Smart Grid Applications", IEEE 1901.2, October 2013, <https://standards.ieee.org/findstds/standard/1901.2-2013.html>.

[ITU-T_G.9903]
International Telecommunication Union, "Narrowband orthogonal frequency division multiplexing power line communication transceivers for G3-PLC networks", ITU-T G.9903, February 2014, <https://www.itu.int/rec/T-REC-G.9903>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC2464] Crawford, M., "Transmission of IPv6 Packets over Ethernet Networks", RFC 2464, DOI 10.17487/RFC2464, December 1998, <https://www.rfc-editor.org/info/rfc2464>.

[RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, DOI 10.17487/RFC4861, September 2007, <https://www.rfc-editor.org/info/rfc4861>.

[RFC4944] Montenegro, G., Kushalnagar, N., Hui, J., and D. Culler, "Transmission of IPv6 Packets over IEEE 802.15.4 Networks", RFC 4944, DOI 10.17487/RFC4944, September 2007, <https://www.rfc-editor.org/info/rfc4944>.

[RFC6282] Hui, J., Ed. and P. Thubert, "Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks", RFC 6282, DOI 10.17487/RFC6282, September 2011, <https://www.rfc-editor.org/info/rfc6282>.

[RFC6550] Winter, T., Ed., Thubert, P., Ed., Brandt, A., Hui, J., Kelsey, R., Levis, P., Pister, K., Struik, R., Vasseur, JP., and R. Alexander, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks", RFC 6550, DOI 10.17487/RFC6550, March 2012, <https://www.rfc-editor.org/info/rfc6550>.

[RFC6775] Shelby, Z., Ed., Chakrabarti, S., Nordmark, E., and C. Bormann, "Neighbor Discovery Optimization for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs)", RFC 6775, DOI 10.17487/RFC6775, November 2012, <https://www.rfc-editor.org/info/rfc6775>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8505] Thubert, P., Ed., Nordmark, E., Chakrabarti, S., and C. Perkins, "Registration Extensions for IPv6 over Low-Power Wireless Personal Area Network (6LoWPAN) Neighbor Discovery", RFC 8505, DOI 10.17487/RFC8505, November 2018, <https://www.rfc-editor.org/info/rfc8505>.

9.2. Informative References

[I-D.ietf-6tisch-dtsecurity-zerotouch-join] Richardson, M., "6tisch Zero-Touch Secure Join protocol", draft-ietf-6tisch-dtsecurity-zerotouch-join-04 (work in progress), July 2019.

[I-D.ietf-6tisch-minimal-security] Vucinic, M., Simon, J., Pister, K., and M. Richardson, "Constrained Join Protocol (CoJP) for 6TiSCH", draft-ietf-6tisch-minimal-security-15 (work in progress), December 2019.

[I-D.ietf-emu-eap-noob] Aura, T. and M. Sethi, "Nimble out-of-band authentication for EAP (EAP-NOOB)", draft-ietf-emu-eap-noob-01 (work in progress), June 2020.

[I-D.ietf-roll-aodv-rpl]
Anamalamudi, S., Zhang, M., Perkins, C., Anand, S., and B. Liu, "AODV based RPL Extensions for Supporting Asymmetric P2P Links in Low-Power and Lossy Networks", draft-ietf-roll-aodv-rpl-08 (work in progress), May 2020.

[I-D.ietf-roll-unaware-leaves]
Thubert, P. and M. Richardson, "Routing for RPL Leaves", draft-ietf-roll-unaware-leaves-15 (work in progress), April 2020.

[I-D.popa-6lo-6loplc-ipv6-over-ieee19012-networks]
Popa, D. and J. Hui, "6LoPLC: Transmission of IPv6 Packets over IEEE 1901.2 Narrowband Powerline Communication Networks", draft-popa-6lo-6loplc-ipv6-over-ieee19012-networks-00 (work in progress), March 2014.

[IEEE_1901.2a]
IEEE-SA Standards Board, "IEEE Standard for Low-Frequency (less than 500 kHz) Narrowband Power Line Communications for Smart Grid Applications - Amendment 1", IEEE 1901.2a, September 2015, <https://standards.ieee.org/findstds/standard/1901.2a-2015.html>.

[RFC3315] Droms, R., Ed., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, DOI 10.17487/RFC3315, July 2003, <https://www.rfc-editor.org/info/rfc3315>.

[RFC3972] Aura, T., "Cryptographically Generated Addresses (CGA)", RFC 3972, DOI 10.17487/RFC3972, March 2005, <https://www.rfc-editor.org/info/rfc3972>.

[RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, DOI 10.17487/RFC4291, February 2006, <https://www.rfc-editor.org/info/rfc4291>.

[RFC4941] Narten, T., Draves, R., and S. Krishnan, "Privacy Extensions for Stateless Address Autoconfiguration in IPv6", RFC 4941, DOI 10.17487/RFC4941, September 2007, <https://www.rfc-editor.org/info/rfc4941>.

[RFC5191] Forsberg, D., Ohba, Y., Ed., Patil, B., Tschofenig, H., and A. Yegin, "Protocol for Carrying Authentication for Network Access (PANA)", RFC 5191, DOI 10.17487/RFC5191, May 2008, <https://www.rfc-editor.org/info/rfc5191>.

[RFC5535] Bagnulo, M., "Hash-Based Addresses (HBA)", RFC 5535, DOI 10.17487/RFC5535, June 2009, <https://www.rfc-editor.org/info/rfc5535>.

[RFC7217] Gont, F., "A Method for Generating Semantically Opaque Interface Identifiers with IPv6 Stateless Address Autoconfiguration (SLAAC)", RFC 7217, DOI 10.17487/RFC7217, April 2014, <https://www.rfc-editor.org/info/rfc7217>.

[RFC7925] Tschofenig, H., Ed. and T. Fossati, "Transport Layer Security (TLS) / Datagram Transport Layer Security (DTLS) Profiles for the Internet of Things", RFC 7925, DOI 10.17487/RFC7925, July 2016, <https://www.rfc-editor.org/info/rfc7925>.

[RFC8036] Cam-Winget, N., Ed., Hui, J., and D. Popa, "Applicability Statement for the Routing Protocol for Low-Power and Lossy Networks (RPL) in Advanced Metering Infrastructure (AMI) Networks", RFC 8036, DOI 10.17487/RFC8036, January 2017, <https://www.rfc-editor.org/info/rfc8036>.

[RFC8064] Gont, F., Cooper, A., Thaler, D., and W. Liu, "Recommendation on Stable IPv6 Interface Identifiers", RFC 8064, DOI 10.17487/RFC8064, February 2017, <https://www.rfc-editor.org/info/rfc8064>.

[RFC8065] Thaler, D., "Privacy Considerations for IPv6 Adaptation-Layer Mechanisms", RFC 8065, DOI 10.17487/RFC8065, February 2017, <https://www.rfc-editor.org/info/rfc8065>.

Authors' Addresses

Jianqiang Hou
Huawei Technologies
101 Software Avenue,
Nanjing 210012
China

Email: houjianqiang@huawei.com

Bing Liu
Huawei Technologies
No. 156 Beiqing Rd. Haidian District,
Beijing 100095
China

Email: remy.liubing@huawei.com

Yong-Geun Hong
Electronics and Telecommunications Research Institute
161 Gajeong-Dong Yuseung-Gu
Daejeon 305-700
Korea

Email: yghong@etri.re.kr

Xiaojun Tang
State Grid Electric Power Research Institute
19 Chengxin Avenue
Nanjing 211106
China

Email: itc@sgepri.sgcc.com.cn

Charles E. Perkins

Email: charliep@computer.org

6Lo Working Group
Internet-Draft
Intended status: Informational
Expires: January 14, 2021

Y-G. Hong
ETRI
C. Gomez
UPC
Y-H. Choi
ETRI
AR. Sangi
Huaiyin Institute of Technology
T. Aanstoot
Modio AB
S. Chakrabarti
July 13, 2020

IPv6 over Constrained Node Networks (6lo) Applicability & Use cases
draft-ietf-6lo-use-cases-09

Abstract

This document describes the applicability of IPv6 over constrained node networks (6lo) and provides practical deployment examples. In addition to IEEE 802.15.4, various link layer technologies such as ITU-T G.9959 (Z-Wave), BLE, DECT-ULE, MS/TP, NFC, and PLC (IEEE 1901.2) are used as examples. The document targets an audience who like to understand and evaluate running end-to-end IPv6 over the constrained node networks connecting devices to each other or to other devices on the Internet (e.g. cloud infrastructure).

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 14, 2021.

Table of Contents

1. Introduction

Running IPv6 on constrained node networks has different features from general node networks due to the characteristics of constrained node networks such as small packet size, short link-layer address, low bandwidth, network topology, low power, low cost, and large number of devices [RFC4919][RFC7228]. For example, some IEEE 802.15.4 link layers[IEEE802154] have a frame size of 127 octets and IPv6 requires the layer below to support an MTU of 1280 bytes, therefore an appropriate fragmentation and reassembly adaptation layer must be provided at the layer below IPv6. Also, the limited size of IEEE 802.15.4 frame and low energy consumption requirements make the need for header compression. The IETF 6LoPWAN (IPv6 over Low powerWPAN) working group published an adaptation layer for sending IPv6 packets over IEEE 802.15.4 [RFC4944], which includes a compression format for IPv6 datagrams over IEEE 802.15.4-based networks [RFC6282], and Neighbor Discovery Optimization for 6LoPWAN [RFC6775].

As IoT (Internet of Things) services become more popular, IPv6 over various link layer technologies such as Bluetooth Low Energy (Bluetooth LE), ITU-T G.9959 (Z-Wave), Digital Enhanced Cordless Telecommunications - Ultra Low Energy (DECT-ULE), Master-Slave/Token Passing (MS/TP), Near Field Communication (NFC), and Power Line Communication (PLC) have been defined at IETF 6lo working group[IETF_6lo]. IPv6 stacks for constrained node networks use a variation of the 6LoWPAN stack applied to each particular link layer technology.

In the 6LoPWAN working group, the [RFC6568], "Design and Application Spaces for 6LoWPANs" was published and it describes potential application scenarios and use cases for low-power wireless personal area networks. Hence, this 6lo applicability document aims to provide guidance to an audience who are new to IPv6-over-low-power networks concept and want to assess if variance of 6LoWPAN stack (6lo) can be applied to the constrained layer two (L2) network of their interest. This 6lo applicability document puts together various design space dimensions such as deployment, network size, power source, connectivity, multi-hop communication, traffic pattern, security level, mobility, and QoS requirements etc. In addition, it describes a few set of 6LoPWAN application scenarios and practical deployment as examples.

This document provides the applicability and use cases of 6lo, considering the following aspects:

- 6lo applicability and use cases are uniquely different from those of 6LoWPAN defined for IEEE 802.15.4.

- It covers various IoT related wire/wireless link layer technologies providing practical information of such technologies.
- A general guideline on how the 6LoWPAN stack can be modified for a given L2 technology is described.
- Various 6lo use cases and practical deployment examples are described.

2. Conventions and Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

3. 6lo Link layer technologies

3.1. ITU-T G.9959

The ITU-T G.9959 Recommendation [G.9959] targets low-power Personal Area Networks (PANs), and defines physical layer and link layer functionality. Physical layers of 9.6 kbit/s, 40 kbit/s and 100 kbit/s are supported. G.9959 defines how a unique 32-bit HomeID network identifier is assigned by a network controller and how an 8-bit NodeID host identifier is allocated to each node. NodeIDs are unique within the network identified by the HomeID. The G.9959 HomeID represents an IPv6 subnet that is identified by one or more IPv6 prefixes [RFC7428]. The ITU-T G.9959 can be used for smart home applications.

3.2. Bluetooth LE

Bluetooth LE was introduced in Bluetooth 4.0, enhanced in Bluetooth 4.1, and developed even further in successive versions. Bluetooth SIG has also published Internet Protocol Support Profile (IPSP). The IPSP enables discovery of IP-enabled devices and establishment of link-layer connection for transporting IPv6 packets. IPv6 over Bluetooth LE is dependent on both Bluetooth 4.1 and IPSP 1.0 or newer.

Many Devices such as mobile phones, notebooks, tablets and other handheld computing devices which support Bluetooth 4.0 or subsequent chipsets also support the low-energy variant of Bluetooth. Bluetooth LE is also being included in many different types of accessories that collaborate with mobile devices such as phones, tablets and notebook computers. An example of a use case for a Bluetooth LE accessory is a heart rate monitor that sends data via the mobile phone to a server on the Internet [RFC7668]. A typical usage of Bluetooth LE is

smartphone-based interaction with constrained devices. Bluetooth LE was originally designed to enable star topology networks. However, recent Bluetooth versions support the formation of extended topologies, and IPv6 support for mesh networks of Bluetooth LE devices is being developed [I-D.ietf-6lo-blemesh]

3.3. DECT-ULE

DECT ULE is a low power air interface technology that is designed to support both circuit switched services, such as voice communication, and packet mode data services at modest data rate.

The DECT ULE protocol stack consists of the PHY layer operating at frequencies in the 1880 - 1920 MHz frequency band depending on the region and uses a symbol rate of 1.152 Mbps. Radio bearers are allocated by use of FDMA/TDMA/TDD techniques.

In its generic network topology, DECT is defined as a cellular network technology. However, the most common configuration is a star network with a single Fixed Part (FP) defining the network with a number of Portable Parts (PP) attached. The MAC layer supports traditional DECT as this is used for services like discovery, pairing, security features etc. All these features have been reused from DECT.

The DECT ULE device can switch to the ULE mode of operation, utilizing the new ULE MAC layer features. The DECT ULE Data Link Control (DLC) provides multiplexing as well as segmentation and re-assembly for larger packets from layers above. The DECT ULE layer also implements per-message authentication and encryption. The DLC layer ensures packet integrity and preserves packet order, but delivery is based on best effort.

The current DECT ULE MAC layer standard supports low bandwidth data broadcast. However the usage of this broadcast service has not yet been standardized for higher layers [RFC8105]. DECT-ULE can be used for smart metering in a home.

3.4. MS/TP

Master-Slave/Token-Passing (MS/TP) is a Medium Access Control (MAC) protocol for the RS-485 [TIA-485-A] physical layer and is used primarily in building automation networks.

An MS/TP device is typically based on a low-cost microcontroller with limited processing power and memory. These constraints, together with low data rates and a small MAC address space, are similar to those faced in 6LoWPAN networks. MS/TP differs significantly from

6LoWPAN in at least three respects: a) MS/TP devices are typically mains powered, b) all MS/TP devices on a segment can communicate directly so there are no hidden node or mesh routing issues, and c) the latest MS/TP specification provides support for large payloads, eliminating the need for fragmentation and reassembly below IPv6.

MS/TP is designed to enable multidrop networks over shielded twisted pair wiring. It can support network segments up to 1000 meters in length at a data rate of 115.2 kbit/s or segments up to 1200 meters in length at lower bit rates. An MS/TP interface requires only a UART, an RS-485 [TIA-485-A] transceiver with a driver that can be disabled, and a 5 ms resolution timer. The MS/TP MAC is typically implemented in software.

Because of its superior "range" (~1 km) compared to many low power wireless data links, MS/TP may be suitable to connect remote devices (such as district heating controllers) to the nearest building control infrastructure over a single link [RFC8163]. MS/TP can be used for building automation networks.

3.5. NFC

NFC technology enables simple and safe two-way interactions between electronic devices, allowing consumers to perform contactless transactions, access digital content, and connect electronic devices with a single touch. NFC complements many popular consumer level wireless technologies, by utilizing the key elements in existing standards for contactless card technology (ISO/IEC 14443 A&B and JIS-X 6319-4). NFC can be compatible with existing contactless card infrastructure and it enables a consumer to utilize one device across different systems.

Extending the capability of contactless card technology, NFC also enables devices to share information at a distance that is less than 10 cm with a maximum communication speed of 424 kbps. Users can share business cards, make transactions, access information from a smart poster or provide credentials for access control systems with a simple touch.

NFC’s bidirectional communication ability is ideal for establishing connections with other technologies by the simplicity of touch. In addition to the easy connection and quick transactions, simple data sharing is also available [I-D.ietf-6lo-nfc]. NFC can be used for secure transfer in healthcare services.

3.6. PLC

PLC is a data transmission technique that utilizes power conductors as medium. Unlike other dedicated communication infrastructure, power conductors are widely available indoors and outdoors. Moreover, wired technologies cause less interference to the radio medium than wireless technologies and are more reliable than their wireless counterparts. PLC is a data transmission technique that utilizes power conductors as medium[I-D.ietf-6lo-plc].

The below table shows some available open standards defining PLC.

PLC Systems	Frequency Range	Type	Data Rate	Distance
IEEE1901	<100MHz	Broadband	200Mbps	1000m
IEEE1901.1	<15MHz	PLC-IoT	10Mbps	2000m
IEEE1901.2	<500kHz	Narrowband	200Kbps	3000m

Table 1: Some Available Open Standards in PLC

[IEEE1901] defines a broadband variant of PLC but is effective within short range. This standard addresses the requirements of applications with high data rate such as: Internet, HDTV, Audio, Gaming etc. Broadband operates on OFDM (Orthogonal Frequency Division Multiplexing) modulation.

[IEEE1901.2] defines a narrowband variant of PLC with less data rate but significantly higher transmission range that could be used in an indoor or even an outdoor environment. It is applicable to typical IoT applications such as: Building Automation, Renewable Energy, Advanced Metering, Street Lighting, Electric Vehicle, Smart Grid etc. Moreover, IEEE 1901.2 standard is based on the 802.15.4 MAC sub-layer and fully endorses the security scheme defined in 802.15.4 [RFC8036]. A typical use case of PLC is smart grid.

3.7. Comparison between 6lo Link layer technologies

In above clauses, various 6lo link layer technologies are described. The following table shows dominant parameters of each use case corresponding to the 6lo link layer technology.

	Z-Wave	BLE	DECT-ULE	MS/TP	NFC	PLC
Usage	Home Auto-mation	Interact w/ Smart Phone	Meter Reading	Building Auto-mation	Health-care Service	Smart Grid
Topology & Subnet	L2-mesh or L3-mesh	Star & Mesh	Star No mesh	MS/TP No mesh	P2P L2-mesh	Star Tree Mesh
Mobility Requirement	No	Low	No	No	Moderate	No
Security Requirement	High + Privacy required	Partially	High + Privacy required	High + Authen. required	High	High + Encrypt. required
Buffering Requirement	Low	Low	Low	Low	Low	Low
Latency, QoS Requirement	High	Low	Low	High	High	Low
Data Rate	Infrequ-ent	Infrequ-ent	Infrequ-ent	Frequent	Small	Infrequ-ent
RFC # or Draft	RFC7428	RFC7668	RFC8105	RFC8163	draft-ietf-6lo-nfc	draft-ietf-6lo-plc

Table 2: Comparison between 6lo Link layer technologies

4. 6lo Deployment Scenarios

4.1. G3-PLC usage of 6lo in network layer

G3-PLC [G3-PLC] is a narrow-band PLC technology that is based on ITU-T G.9903 Recommendation [G.9903]. G3-PLC supports multi-hop mesh network, and facilitates highly-reliable, long-range communication. With the abilities to support IPv6 and to cross transformers, G3-PLC is regarded as one of the next-generation NB-PLC technologies.

G3-PLC has got massive deployments over several countries, e.g. Japan and France.

The main application domains targeted by G3-PLC are smart grid and smart cities. This includes, but is not limited to the following applications:

- Smart Metering
- Vehicle-to-Grid Communication
- Demand Response (DR)
- Distribution Automation
- Home/Building Energy Management Systems
- Smart Street Lighting
- Advanced Metering Infrastructure (AMI) backbone network
- Wind/Solar Farm Monitoring

In the G3-PLC specification, the 6lo adaption layer utilizes the 6LoWPAN functions (e.g. header compression, fragmentation and reassembly). However, due to the different characteristics of the PLC media, the 6LoWPAN adaptation layer cannot perfectly fulfill the requirements[I-D.ietf-6lo-plc]. The ESC dispatch type is used in the G3-PLC to provide native mesh routing and bootstrapping functionalities[RFC8066].

4.2. Netricity usage of 6lo in network layer

The Netricity program in HomePlug Powerline Alliance [NETRICITY] promotes the adoption of products built on the IEEE 1901.2 Low-Frequency Narrow-Band PLC standard, which provides for urban and long distance communications and propagation through transformers of the distribution network using frequencies below 500 kHz. The technology also addresses requirements that assure communication privacy and secure networks.

The main application domains targeted by Netricity are smart grid and smart cities. This includes, but is not limited to the following applications:

- Utility grid modernization
- Distribution automation

- Meter-to-Grid connectivity
- Micro-grids
- Grid sensor communications
- Load control
- Demand response
- Net metering
- Street Lighting control
- Photovoltaic panel monitoring

Netricity system architecture is based on the PHY and MAC layers of IEEE 1901.2 PLC standard. Regarding the 6lo adaptation layer and IPv6 network layer, Netricity utilizes IPv6 protocol suite including 6lo/6LoWPAN header compression, DHCPv6 for IP address management, RPL routing protocol, ICMPv6, and unicast/multicast forwarding. Note that the layer 3 routing in Netricity uses RPL in non-storing mode with the MRHOF objective function based on the own defined Estimated Transmission Time (ETT) metric.

5. Guidelines for adopting IPv6 stack (6lo/6LoWPAN)

The following guideline targets new candidate constrained L2 technologies that may be considered for running modified 6LoWPAN stack on top. The modification of 6LoWPAN stack SHOULD be based on the following:

- Addressing Model: Addressing model determines whether the device is capable of forming IPv6 Link-local and global addresses and what is the best way to derive the IPv6 addresses for the constrained L2 devices. Whether the device is capable of forming IPv6 Link-local and global addresses, L2-address-derived IPv6 addresses are specified in [RFC4944], but there exist implications for privacy. For global usage, a unique IPv6 address must be derived using an assigned prefix and a unique interface ID. [RFC8065] provides such guidelines. For MAC derived IPv6 address, please refer to [RFC8163] for IPv6 address mapping examples. Broadcast and multicast support are dependent on the L2 networks. Most low-power L2 implementations map multicast to broadcast networks. So care must be taken in the design when to use broadcast and try to stick to unicast messaging whenever possible.

- MTU Considerations: The deployment SHOULD consider their need for maximum transmission unit (MTU) of a packet over the link layer and SHOULD consider if fragmentation and reassembly of packets are needed at the 6LoWPAN layer. For example, if the link layer supports fragmentation and reassembly of packets, then 6LoWPAN layer may skip supporting fragmentation/reassembly. In fact, for most efficiency, choosing a low-power link layer that can carry unfragmented application packets would be optimum for packet transmission if the deployment can afford it. Please refer to 6lo RFCs [RFC7668], [RFC8163], [RFC8105] for example guidance.

- Mesh or L3-Routing: 6LoWPAN specifications do provide mechanisms to support for mesh routing at L2. [RFC6550] defines layer three (L3) routing for low power lossy networks using directed graphs. 6LoWPAN is routing protocol agnostic and other L2 or L3 routing protocols can be run using a 6LoWPAN stack.

- Address Assignment: 6LoWPAN developed a new version of IPv6 Neighbor Discovery[RFC4861][RFC4862] that relies on a proactive registration to avoid the use of multicast. 6LoWPAN Neighbor Discovery[RFC6775][RFC8505] inherits from IPv6 Neighbor Discovery for mechanisms such as Stateless Address Autoconfiguration(SLAAC) and Neighbor Unreachability Detection(NUD), but uses a unicast method for Duplicate Address Detection(DAD), and avoids multicast lookups from all nodes by using non-onlink prefixes. A 6LoWPAN Node is also expected to be an IPv6 host per[RFC8200] which means it should ignore consumed routing headers and Hop-by-Hop options; when operating in a RPL network[RFC6550], it is also beneficial to support IP-in-IP encapsulation [I-D.ietf-roll-useofrplinfo]. The 6LoWPWAN Node should also support [RFC8505] and use it as the default Neighbor Discovery method. It is the responsibility of the deployment to ensure unique global IPv6 addresses for the Internet connectivity. For local-only connectivity IPv6 ULA may be used. [RFC6775] specifies the 6LoWPAN border router(6LBR) which is responsible for prefix assignment to the 6lo/6LoWPAN network. 6LBR can be connected to the Internet or Enterprise network via its one of the interfaces. Please refer to [RFC7668] and [RFC8105] for examples of address assignment considerations. In addition, privacy considerations [RFC8065] must be consulted for applicability. In certain scenarios, the deployment may not support autoconfiguration of IPv6 addressing due to regulatory and business reasons and may choose to offer a separate address assignment service.

- Header Compression: IPv6 header compression [RFC6282] is a vital part of IPv6 over low power communication. Examples of header compression for different link-layers specifications are found in

[RFC7668], [RFC8163], [RFC8105]. A generic header compression technique is specified in [RFC7400].

- Security and Encryption: Though 6LoWPAN basic specifications do not address security at the network layer, the assumption is that L2 security must be present. In addition, application level security is highly desirable. The working groups [IETF_ace] and [IETF_core] should be consulted for application and transport level security. 6lo working group is working on address authentication [I-D.ietf-6lo-ap-nd] and secure bootstrapping is also being discussed at IETF. However, there may be different levels of security available in a deployment through other standards such as hardware level security or certificates for initial booting process. Encryption is important if the implementation can afford it.

- Additional processing: [RFC8066] defines guidelines for ESC dispatch octets use in the 6LoWPAN header. An implementation may take advantage of ESC header to offer a deployment specific processing of 6LoWPAN packets.

6. 6lo Use Case Examples

As IPv6 stacks for constrained node networks use a variation of the 6LoWPAN stack applied to each particular link layer technology, various 6lo use cases can be provided. In this clause, various 6lo use cases which are based on each particular link layer technology are described.

6.1. Use case of ITU-T G.9959: Smart Home

Z-Wave is one of the main technologies that may be used to enable smart home applications. Born as a proprietary technology, Z-Wave was specifically designed for this particular use case. Recently, the Z-Wave radio interface (physical and MAC layers) has been standardized as the ITU-T G.9959 specification.

Example: Use of ITU-T G.9959 for Home Automation

Variety of home devices (e.g. light dimmers/switches, plugs, thermostats, blinds/curtains and remote controls) are augmented with ITU-T G.9959 interfaces. A user may turn on/off or may control home appliances by pressing a wall switch or by pressing a button in a remote control. Scenes may be programmed, so that after a given event, the home devices adopt a specific configuration. Sensors may also periodically send measurements of several parameters (e.g. gas presence, light, temperature, humidity, etc.) which are collected at

a sink device, or may generate commands for actuators (e.g. a smoke sensor may send an alarm message to a safety system).

The devices involved in the described scenario are nodes of a network that follows the mesh topology, which is suitable for path diversity to face indoor multipath propagation issues. The multihop paradigm allows end-to-end connectivity when direct range communication is not possible. Security support is required, specially for safety-related communication. When a user interaction (e.g. a button press) triggers a message that encapsulates a command, if the message is lost, the user may have to perform further interactions to achieve the desired effect (e.g. a light is turned off). A reaction to a user interaction will be perceived by the user as immediate as long as the reaction takes place within 0.5 seconds [RFC5826].

6.2. Use case of Bluetooth LE: Smartphone-based Interaction

The key feature behind the current high Bluetooth LE momentum is its support in a large majority of smartphones in the market. Bluetooth LE can be used to allow the interaction between the smartphone and surrounding sensors or actuators. Furthermore, Bluetooth LE is also the main radio interface currently available in wearables. Since a smartphone typically has several radio interfaces that provide Internet access, such as Wi-Fi or 4G, the smartphone can act as a gateway for nearby devices such as sensors, actuators or wearables. Bluetooth LE may be used in several domains, including healthcare, sports/wellness and home automation.

Example: Use of Bluetooth LE-based Body Area Network for fitness

A person wears a smartwatch for fitness purposes. The smartwatch has several sensors (e.g. heart rate, accelerometer, gyrometer, GPS, temperature, etc.), a display, and a Bluetooth LE radio interface. The smartwatch can show fitness-related statistics on its display. However, when a paired smartphone is in the range of the smartwatch, the latter can report almost real-time measurements of its sensors to the smartphone, which can forward the data to a cloud service on the Internet. In addition, the smartwatch can receive notifications (e.g. alarm signals) from the cloud service via the smartphone. On the other hand, the smartphone may locally generate messages for the smartwatch, such as e-mail reception or calendar notifications.

The functionality supported by the smartwatch may be complemented by other devices such as other on-body sensors, wireless headsets or head-mounted displays. All such devices may connect to the smartphone creating a star topology network whereby the smartphone is the central component. Support for extended network topologies (e.g. mesh networks) is being developed as of the writing.

6.3. Use case of DECT-ULE: Smart Home

DECT is a technology widely used for wireless telephone communications in residential scenarios. Since DECT-ULE is a low-power variant of DECT, DECT-ULE can be used to connect constrained devices such as sensors and actuators to a Fixed Part, a device that typically acts as a base station for wireless telephones. Therefore, DECT-ULE is specially suitable for the connected home space in application areas such as home automation, smart metering, safety, healthcare, etc.

Example: Use of DECT-ULE for Smart Metering

The smart electricity meter of a home is equipped with a DECT-ULE transceiver. This device is in the coverage range of the Fixed Part of the home. The Fixed Part can act as a router connected to the Internet. This way, the smart meter can transmit electricity consumption readings through the DECT-ULE link with the Fixed Part, and the latter can forward such readings to the utility company using Wide Area Network (WAN) links. The meter can also receive queries from the utility company or from an advanced energy control system controlled by the user, which may also be connected to the Fixed Part via DECT-ULE.

6.4. Use case of MS/TP: Building Automation Networks

The primary use case for IPv6 over MS/TP (6LoBAC) is in building automation networks. [BACnet] is the open international standard protocol for building automation, and MS/TP is defined in [BACnet] Clause 9. MS/TP was designed to be a low cost multi-drop field bus to inter-connect the most numerous elements (sensors and actuators) of a building automation network to their controllers. A key aspect of 6LoBAC is that it is designed to co-exist with BACnet MS/TP on the same link, easing the ultimate transition of some BACnet networks to native end-to-end IPv6 transport protocols. New applications for 6LoBAC may be found in other domains where low cost, long distance, and low latency are required.

Example: Use of 6LoBAC in Building Automation Networks

The majority of installations for MS/TP are for "terminal" or "unitary" controllers, i.e. single zone or room controllers that may connect to HVAC or other controls such as lighting or blinds. The economics of daisy-chaining a single twisted-pair between multiple devices is often preferred over home-run Cat-5 style wiring.

A multi-zone controller might be implemented as an IP router between a traditional Ethernet link and several 6LoBAC links, fanning out to multiple terminal controllers.

The superior distance capabilities of MS/TP (~1 km) compared to other 6lo media may suggest its use in applications to connect remote devices to the nearest building infrastructure. for example, remote pumping or measuring stations with moderate bandwidth requirements can benefit from the low cost and robust capabilities of MS/TP over other wired technologies such as DSL, and without the line-of-site restrictions or hop-by-hop latency of many low cost wireless solutions.

6.5. Use case of NFC: Alternative Secure Transfer

According to applications, various secured data can be handled and transferred. Depending on security level of the data, methods for transfer can be alternatively selected.

Example: Use of NFC for Secure Transfer in Healthcare Services with Tele-Assistance

A senior citizen who lives alone wears one to several wearable 6lo devices to measure heartbeat, pulse rate, etc. The 6lo devices are densely installed at home for movement detection. An LoWPAN Border Router (LBR) at home will send the sensed information to a connected healthcare center. Portable base stations with LCDs may be used to check the data at home, as well. Data is gathered in both periodic and event-driven fashion. In this application, event-driven data can be very time-critical. In addition, privacy also becomes a serious issue in this case, as the sensed data is very personal.

While the senior citizen is provided audio and video healthcare services by a tele-assistance based on LTE connections, the senior citizen can alternatively use NFC connections to transfer the personal sensed data to the tele-assistance. At this moment, hidden hackers can overhear the data based on the LTE connection, but they cannot gather the personal data over the NFC connection.

6.6. Use case of PLC: Smart Grid

Smart grid concept is based on numerous operational and energy measuring sub-systems of an electric grid. It comprises of multiple administrative levels/segments to provide connectivity among these numerous components. Last mile connectivity is established over LV segment, whereas connectivity over electricity distribution takes place in HV segment.

Although other wired and wireless technologies are also used in Smart Grid (Advance Metering Infrastructure - AMI, Demand Response - DR, Home Energy Management System - HEMS, Wide Area Situational Awareness - WASA etc), PLC enjoys the advantage of existing (power conductor) medium and better reliable data communication. PLC is a promising wired communication technology in that the electrical power lines are already there and the deployment cost can be comparable to wireless technologies. The 6lo related scenarios lie in the low voltage PLC networks with most applications in the area of Advanced Metering Infrastructure, Vehicle-to-Grid communications, in-home energy management and smart street lighting.

Example: Use of PLC for Advanced Metering Infrastructure

Household electricity meters transmit time-based data of electric power consumption through PLC. Data concentrators receive all the meter data in their corresponding living districts and send them to the Meter Data Management System (MDMS) through WAN network (e.g. Medium-Voltage PLC, Ethernet or GPRS) for storage and analysis. Two-way communications are enabled which means smart meters can do actions like notification of electricity charges according to the commands from the utility company.

With the existing power line infrastructure as communication medium, cost on building up the PLC network is naturally saved, and more importantly, labor operational costs can be minimized from a long-term perspective. Furthermore, this AMI application speeds up electricity charge, reduces losses by restraining power theft and helps to manage the health of the grid based on line loss analysis.

Example: Use of PLC (IEEE1901.1) for WASA in Smart Grid

Many sub-systems of Smart Grid require low data rate and narrowband variant (IEEE1901.2) of PLC fulfils such requirements. Recently, more complex scenarios are emerging that require higher data rates.

WASA sub-system is an appropriate example that collects large amount of information about the current state of the grid over wide area from electric substations as well as power transmission lines. The collected feedback is used for monitoring, controlling and protecting all the sub-systems.

7. IANA Considerations

There are no IANA considerations related to this document.

8. Security Considerations

Security considerations are not directly applicable to this document. The use cases will use the security requirements described in the protocol specifications.

9. Acknowledgements

Carles Gomez has been funded in part by the Spanish Government through the Jose Castillejo CAS15/00336 grant, and through the TEC2016-79988-P grant. His contribution to this work has been carried out in part during his stay as a visiting scholar at the Computer Laboratory of the University of Cambridge.

Thomas Watteyne, Pascal Thubert, Xavier Vilajosana, Daniel Migault, and Jianqiang HOU have provided valuable feedback for this draft.

Das Subir and Michel Veillette have provided valuable information of jupiterMesh and Paul Duffy has provided valuable information of Wi-SUN for this draft. Also, Jianqiang Hou has provided valuable information of G3-PLC and Netricity for this draft. Kerry Lynn and Dave Robin have provided valuable information of MS/TP and practical use case of MS/TP for this draft.

Deoknyong Ko has provided relevant text of LTE-MTC and he shared his experience to deploy IPv6 and 6lo technologies over LTE MTC in SK Telecom.

10. References

10.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

10.2. Informative References

[BACnet] "ASHRAE, "BACnet-A Data Communication Protocol for Building Automation and Control Networks", ANSI/ASHRAE Standard 135-2016", January 2016, <http://www.techstreet.com/ashrae/standards/ashrae-135-2016?product_id=1918140#jumps>.

[G.9903] "International Telecommunication Union, "Narrowband orthogonal frequency division multiplexing power line communication transceivers for G3-PLC networks", ITU-T Recommendation", August 2017.

[G.9959] "International Telecommunication Union, "Short range narrow-band digital radiocommunication transceivers - PHY and MAC layer specifications", ITU-T Recommendation", January 2015.

[G3-PLC] "G3-PLC Alliance", <http://www.g3-plc.com/home/>.

[IEEE1901] "IEEE Standard, IEEE Std. 1901-2010 - IEEE Standard for Broadband over Power Line Networks: Medium Access Control and Physical Layer Specifications", 2010, <https://standards.ieee.org/findstds/standard/1901-2010.html>.

[IEEE1901.2] "IEEE Standard, IEEE Std. 1901.2-2013 - IEEE Standard for Low-Frequency (less than 500 kHz) Narrowband Power Line Communications for Smart Grid Applications", 2013, <https://standards.ieee.org/findstds/standard/1901.2-2013.html>.

[IEEE802154] IEEE standard for Information Technology, "IEEE Std. 802.15.4, Part. 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks".

[I-D.ietf-6lo-ap-nd] Thubert, P., Sarikaya, B., Sethi, M., and R. Struik, "Address Protected Neighbor Discovery for Low-power and Lossy Networks", draft-ietf-6lo-ap-nd-23 (work in progress), April 2020.

[I-D.ietf-6lo-blemesh] Gomez, C., Darroudi, S., Savolainen, T., and M. Spoerk, "IPv6 Mesh over BLUETOOTH(R) Low Energy using IPSP", draft-ietf-6lo-blemesh-07 (work in progress), December 2019.

[I-D.ietf-6lo-nfc]
Choi, Y., Hong, Y., Youn, J., Kim, D., and J. Choi, "Transmission of IPv6 Packets over Near Field Communication", draft-ietf-6lo-nfc-16 (work in progress), July 2020.

[I-D.ietf-6lo-plc]
Hou, J., Liu, B., Hong, Y., Tang, X., and C. Perkins, "Transmission of IPv6 Packets over PLC Networks", draft-ietf-6lo-plc-04 (work in progress), June 2020.

[I-D.ietf-roll-useofrplinfo]
Robles, I., Richardson, M., and P. Thubert, "Using RPI Option Type, Routing Header for Source Routes and IPv6-in-IPv6 encapsulation in the RPL Data Plane", draft-ietf-roll-useofrplinfo-40 (work in progress), June 2020.

[IETF_6lo]
"IETF IPv6 over Networks of Resource-constrained Nodes (6lo) working group", <https://datatracker.ietf.org/wg/6lo/charter/>.

[IETF_ace]
"IETF Authentication and Authorization for Constrained Environments (ace) working group", <https://datatracker.ietf.org/wg/ace/charter/>.

[IETF_core]
"IETF Constrained RESTful Environments (core) working group", <https://datatracker.ietf.org/wg/core/charter/>.

[NETRICITY]
"Netricity program in HomePlug Powerline Alliance", <http://groups.homeplug.org/tech/Netricity>.

[RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, DOI 10.17487/RFC4861, September 2007, <https://www.rfc-editor.org/info/rfc4861>.

[RFC4862] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", RFC 4862, DOI 10.17487/RFC4862, September 2007, <https://www.rfc-editor.org/info/rfc4862>.

[RFC4919] Kushalnagar, N., Montenegro, G., and C. Schumacher, "IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals", RFC 4919, DOI 10.17487/RFC4919, August 2007, <https://www.rfc-editor.org/info/rfc4919>.

[RFC4944] Montenegro, G., Kushalnagar, N., Hui, J., and D. Culler, "Transmission of IPv6 Packets over IEEE 802.15.4 Networks", RFC 4944, DOI 10.17487/RFC4944, September 2007, <https://www.rfc-editor.org/info/rfc4944>.

[RFC5826] Brandt, A., Buron, J., and G. Porcu, "Home Automation Routing Requirements in Low-Power and Lossy Networks", RFC 5826, DOI 10.17487/RFC5826, April 2010, <https://www.rfc-editor.org/info/rfc5826>.

[RFC6282] Hui, J., Ed. and P. Thubert, "Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks", RFC 6282, DOI 10.17487/RFC6282, September 2011, <https://www.rfc-editor.org/info/rfc6282>.

[RFC6550] Winter, T., Ed., Thubert, P., Ed., Brandt, A., Hui, J., Kelsey, R., Levis, P., Pister, K., Struik, R., Vasseur, JP., and R. Alexander, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks", RFC 6550, DOI 10.17487/RFC6550, March 2012, <https://www.rfc-editor.org/info/rfc6550>.

[RFC6568] Kim, E., Kaspar, D., and JP. Vasseur, "Design and Application Spaces for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs)", RFC 6568, DOI 10.17487/RFC6568, April 2012, <https://www.rfc-editor.org/info/rfc6568>.

[RFC6775] Shelby, Z., Ed., Chakrabarti, S., Nordmark, E., and C. Bormann, "Neighbor Discovery Optimization for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs)", RFC 6775, DOI 10.17487/RFC6775, November 2012, <https://www.rfc-editor.org/info/rfc6775>.

[RFC7228] Bormann, C., Ersue, M., and A. Keranen, "Terminology for Constrained-Node Networks", RFC 7228, DOI 10.17487/RFC7228, May 2014, <https://www.rfc-editor.org/info/rfc7228>.

[RFC7400] Bormann, C., "6LoWPAN-GHC: Generic Header Compression for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs)", RFC 7400, DOI 10.17487/RFC7400, November 2014, <https://www.rfc-editor.org/info/rfc7400>.

[RFC7428] Brandt, A. and J. Buron, "Transmission of IPv6 Packets over ITU-T G.9959 Networks", RFC 7428, DOI 10.17487/RFC7428, February 2015, <https://www.rfc-editor.org/info/rfc7428>.

[RFC7668] Nieminen, J., Savolainen, T., Isomaki, M., Patil, B., Shelby, Z., and C. Gomez, "IPv6 over BLUETOOTH(R) Low Energy", RFC 7668, DOI 10.17487/RFC7668, October 2015, <https://www.rfc-editor.org/info/rfc7668>.

[RFC8036] Cam-Winget, N., Ed., Hui, J., and D. Popa, "Applicability Statement for the Routing Protocol for Low-Power and Lossy Networks (RPL) in Advanced Metering Infrastructure (AMI) Networks", RFC 8036, DOI 10.17487/RFC8036, January 2017, <https://www.rfc-editor.org/info/rfc8036>.

[RFC8065] Thaler, D., "Privacy Considerations for IPv6 Adaptation-Layer Mechanisms", RFC 8065, DOI 10.17487/RFC8065, February 2017, <https://www.rfc-editor.org/info/rfc8065>.

[RFC8066] Chakrabarti, S., Montenegro, G., Droms, R., and J. Woodyatt, "IPv6 over Low-Power Wireless Personal Area Network (6LoWPAN) ESC Dispatch Code Points and Guidelines", RFC 8066, DOI 10.17487/RFC8066, February 2017, <https://www.rfc-editor.org/info/rfc8066>.

[RFC8105] Mariager, P., Petersen, J., Ed., Shelby, Z., Van de Logt, M., and D. Barthel, "Transmission of IPv6 Packets over Digital Enhanced Cordless Telecommunications (DECT) Ultra Low Energy (ULE)", RFC 8105, DOI 10.17487/RFC8105, May 2017, <https://www.rfc-editor.org/info/rfc8105>.

[RFC8163] Lynn, K., Ed., Martocci, J., Neilson, C., and S. Donaldson, "Transmission of IPv6 over Master-Slave/Token-Passing (MS/TP) Networks", RFC 8163, DOI 10.17487/RFC8163, May 2017, <https://www.rfc-editor.org/info/rfc8163>.

[RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <https://www.rfc-editor.org/info/rfc8200>.

[RFC8352] Gomez, C., Kovatsch, M., Tian, H., and Z. Cao, Ed., "Energy-Efficient Features of Internet of Things Protocols", RFC 8352, DOI 10.17487/RFC8352, April 2018, <https://www.rfc-editor.org/info/rfc8352>.

[RFC8505] Thubert, P., Ed., Nordmark, E., Chakrabarti, S., and C. Perkins, "Registration Extensions for IPv6 over Low-Power Wireless Personal Area Network (6LoWPAN) Neighbor Discovery", RFC 8505, DOI 10.17487/RFC8505, November 2018, <https://www.rfc-editor.org/info/rfc8505>.

[TIA-485-A] "TIA, "Electrical Characteristics of Generators and Receivers for Use in Balanced Digital Multipoint Systems", TIA-485-A (Revision of TIA-485)", March 2003, <https://global.ihs.com/doc_detail.cfm?item_s_key=00032964>.

Appendix A. Design Space Dimensions for 6lo Deployment

The [RFC6568] lists the dimensions used to describe the design space of wireless sensor networks in the context of the 6LoWPAN working group. The design space is already limited by the unique characteristics of a LoWPAN (e.g. low power, short range, low bit rate). In [RFC6568], the following design space dimensions are described: Deployment, Network size, Power source, Connectivity, Multi-hop communication, Traffic pattern, Mobility, Quality of Service (QoS). However, in this document, the following design space dimensions are considered:

- Deployment/Bootstrapping: 6lo nodes can be connected randomly, or in an organized manner. The bootstrapping has different characteristics for each link layer technology.

- Topology: Topology of 6lo networks may inherently follow the characteristics of each link layer technology. Point-to-point, star, tree or mesh topologies can be configured, depending on the link layer technology considered.

- L2-Mesh or L3-Mesh: L2-mesh and L3-mesh may inherently follow the characteristics of each link layer technology. Some link layer technologies may support L2-mesh and some may not support.

- Multi-link subnet, single subnet: The selection of multi-link subnet and single subnet depends on connectivity and the number of 6lo nodes.

- Data rate: Typically, the link layer technologies of 6lo have low rate of data transmission. But, by adjusting the MTU, it can deliver higher upper layer data rate.

- Buffering requirements: Some 6lo use case may require more data rate than the link layer technology support. In this case, a buffering mechanism to manage the data is required.

- Security and Privacy Requirements: Some 6lo use case can involve transferring some important and personal data between 6lo nodes. In this case, high-level security support is required.

- Mobility across 6lo networks and subnets: The movement of 6lo nodes depends on the 6lo use case. If the 6lo nodes can move or moved around, a mobility management mechanism is required.

- Time synchronization requirements: The requirement of time synchronization of the upper layer service is dependent on the 6lo use case. For some 6lo use case related to health service, the measured data must be recorded with exact time and must be transferred with time synchronization.

- Reliability and QoS: Some 6lo use case requires high reliability, for example real-time service or health-related services.

- Traffic patterns: 6lo use cases may involve various traffic patterns. For example, some 6lo use case may require short data length and random transmission. Some 6lo use case may require continuous data and periodic data transmission.

- Security Bootstrapping: Without the external operations, 6lo nodes must have the security bootstrapping mechanism.

- Power use strategy: to enable certain use cases, there may be requirements on the class of energy availability and the strategy followed for using power for communication [RFC7228]. Each link layer technology defines a particular power use strategy which may be tuned [RFC8352]. Readers are expected to be familiar with [RFC7228] terminology.

- Update firmware requirements: Most 6lo use cases will need a mechanism for updating firmware. In these cases support for over the air updates are required, probably in a broadcast mode when bandwith is low and the number of identical devices is high.

- Wired vs. Wireless: Plenty of 6lo link layer technologies are wireless, except MS/TP and PLC. The selection of wired or wireless link layer technology is mainly dependent on the

requirement of 6lo use cases and the characteristics of wired/ wireless technologies. For example, some 6lo use cases may require easy and quick deployment, whereas others may need a continuous source of power.

Authors' Addresses

Yong-Geun Hong
ETRI
218 Gajeongno, Yuseong
Daejeon 34129
Korea

Phone: +82 42 860 6557
Email: yghong@etri.re.kr

Carles Gomez
Universitat Politecnica de Catalunya/Fundacio i2cat
C/Esteve Terradas, 7
Castelldefels 08860
Spain

Email: carlesgo@entel.upc.edu

Younghwan Choi
ETRI
218 Gajeongno, Yuseong
Daejeon 34129
Korea

Phone: +82 42 860 1429
Email: yhc@etri.re.kr

Abdur Rashid Sangi
Huaiyin Institute of Technology
No.89 North Beijing Road, Qinghe District
Huaian 223001
P.R. China

Email: sangi_bahrian@yahoo.com

Take Aanstoot
Modio AB
S:t Larsgatan 15, 582 24
Linkoping
Sweden

Email: take@modio.se

Samita Chakrabarti
San Jose, CA
USA

Email: samitac.ietf@gmail.com