Terminology for Constrained-Node Networks
draft-bormann-lwig-7228bis-05

Abstract

   The Internet Protocol Suite is increasingly used on small devices
   with severe constraints on power, memory, and processing resources,
   creating constrained-node networks. This document provides a number
   of basic terms that have been useful in the standardization work for
   constrained-node networks.

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 March 5, 2020.

Copyright Notice

Table of Contents

1. Introduction

   Small devices with limited CPU, memory, and power resources, so-
   called "constrained devices" (often used as sensors/actuators, smart
   objects, or smart devices) can form a network, becoming "constrained
   nodes" in that network. Such a network may itself exhibit
   constraints, e.g., with unreliable or lossy channels, limited and
   unpredictable bandwidth, and a highly dynamic topology.

   Constrained devices might be in charge of gathering information in
   diverse settings, including natural ecosystems, buildings, and
   factories, and sending the information to one or more server
   stations. They might also act on information, by performing some

physical action, including displaying it. Constrained devices may
work under severe resource constraints such as limited battery and
computing power, little memory, and insufficient wireless bandwidth
and ability to communicate; these constraints often exacerbate each
other. Other entities on the network, e.g., a base station or
controlling server, might have more computational and communication
resources and could support the interaction between the constrained
devices and applications in more traditional networks.

Today, diverse sizes of constrained devices with different resources
and capabilities are becoming connected. Mobile personal gadgets,
building-automation devices, cellular phones, machine-to-machine
(M2M) devices, and other devices benefit from interacting with other
"things" nearby or somewhere in the Internet. With this, the
Internet of Things (IoT) becomes a reality, built up out of uniquely
identifiable and addressable objects (things). Over the next decade,
this could grow to large numbers [FIFTY-BILLION] of Internet-
connected constrained devices, greatly increasing the Internet's size
and scope.

The present document provides a number of basic terms that have been
useful in the standardization work for constrained environments. The
intention is not to exhaustively cover the field but to make sure a
few core terms are used consistently between different groups
cooperating in this space.

The present document is an update of [RFC7228].

In this document, the term "byte" is used in its now customary sense
as a synonym for "octet". Where sizes of semiconductor memory are
given, the prefix "kibi" (1024) is combined with "byte" to
"kibibyte", abbreviated "KiB", for 1024 bytes [ISQ-13].

In computing, the term "power" is often used for the concept of
"computing power" or "processing power", as in CPU performance. In
this document, the term stands for electrical power unless explicitly
stated otherwise. "Mains-powered" is used as a shorthand for being
permanently connected to a stable electrical power grid.

2. Core Terminology

There are two important aspects to _scaling_ within the Internet of
Things:

o scaling up Internet technologies to a large number [FIFTY-BILLION]
   of inexpensive nodes, while

o scaling down the characteristics of each of these nodes and of the
   networks being built out of them, to make this scaling up
   economically and physically viable.

The need for scaling down the characteristics of nodes leads to
"constrained nodes".

2.1. Constrained Nodes

The term "constrained node" is best defined by contrasting the
characteristics of a constrained node with certain widely held
expectations on more familiar Internet nodes:

Constrained Node: A node where some of the characteristics that are
   otherwise pretty much taken for granted for Internet nodes at the
   time of writing are not attainable, often due to cost constraints
   and/or physical constraints on characteristics such as size,
   weight, and available power and energy. The tight limits on
   power, memory, and processing resources lead to hard upper bounds
   on state, code space, and processing cycles, making optimization
   of energy and network bandwidth usage a dominating consideration
   in all design requirements. Also, some layer-2 services such as
   full connectivity and broadcast/multicast may be lacking.

While this is not a rigorous definition, it is grounded in the state
of the art and clearly sets apart constrained nodes from server
systems, desktop or laptop computers, powerful mobile devices such as
smartphones, etc. There may be many design considerations that lead
to these constraints, including cost, size, weight, and other scaling
factors.

(An alternative term, when the properties as a network node are not
in focus, is "constrained device".)

There are multiple facets to the constraints on nodes, often applying
in combination, for example:

o constraints on the maximum code complexity (ROM/Flash),

o constraints on the size of state and buffers (RAM),

o constraints on the amount of computation feasible in a period of
   time ("processing power"),

o constraints on the available power, and

o constraints on user interface and accessibility in deployment
   (ability to set keys, update software, etc.).

Section 3 defines a small number of interesting classes ("class-N"
for N = 0, 1, 2) of constrained nodes focusing on relevant
combinations of the first two constraints. With respect to available
power, [RFC6606] distinguishes "power-affluent" nodes (mains-powered
or regularly recharged) from "power-constrained nodes" that draw
their power from primary batteries or by using energy harvesting;
more detailed power terminology is given in Section 4.

The use of constrained nodes in networks often also leads to
constraints on the networks themselves. However, there may also be
constraints on networks that are largely independent from those of
the nodes. We therefore distinguish "constrained networks" from
"constrained-node networks".

2.2. Constrained Networks

We define "constrained network" in a similar way:

Constrained Network: A network where some of the characteristics
   pretty much taken for granted with link layers in common use in
   the Internet at the time of writing are not attainable.

Constraints may include:

o low achievable bitrate/throughput (including limits on duty
   cycle),

o high packet loss and high variability of packet loss (delivery
   rate),

o highly asymmetric link characteristics,

o severe penalties for using larger packets (e.g., high packet loss
   due to link-layer fragmentation),

o limits on reachability over time (a substantial number of devices
   may power off at any point in time but periodically "wake up" and
   can communicate for brief periods of time), and

o lack of (or severe constraints on) advanced services such as IP
   multicast.

More generally, we speak of constrained networks whenever at least
some of the nodes involved in the network exhibit these
characteristics.

Again, there may be several reasons for this:

o cost constraints on the network,

o constraints posed by the nodes (for constrained-node networks),

o physical constraints (e.g., power constraints, environmental
   constraints, media constraints such as underwater operation,
   limited spectrum for very high density, electromagnetic
   compatibility),

o regulatory constraints, such as very limited spectrum availability
   (including limits on effective radiated power and duty cycle) or
   explosion safety, and

o technology constraints, such as older and lower-speed technologies
   that are still operational and may need to stay in use for some
   more time.

2.2.1. Challenged Networks

A constrained network is not necessarily a "challenged network"
[FALL]:

Challenged Network: A network that has serious trouble maintaining
   what an application would today expect of the end-to-end IP model,
   e.g., by:

   * not being able to offer end-to-end IP connectivity at all,

   * exhibiting serious interruptions in end-to-end IP connectivity,
      or

   * exhibiting delay well beyond the Maximum Segment Lifetime (MSL)
      defined by TCP [RFC0793].

All challenged networks are constrained networks in some sense, but
not all constrained networks are challenged networks. There is no
well-defined boundary between the two, though. Delay-Tolerant
Networking (DTN) has been designed to cope with challenged networks
[RFC4838].

2.3. Constrained-Node Networks

Constrained-Node Network: A network whose characteristics are
   influenced by being composed of a significant portion of
   constrained nodes.

A constrained-node network always is a constrained network because of
the network constraints stemming from the node constraints, but it

may also have other constraints that already make it a constrained
network.

The rest of this subsection introduces two additional terms that are
in active use in the area of constrained-node networks, without an
intent to define them: LLN and (6)LoWPAN.

2.3.1. LLN

A related term that has been used to describe the focus of the IETF
ROLL working group is "Low-Power and Lossy Network (LLN)". The ROLL
(Routing Over Low-Power and Lossy) terminology document [RFC7102]
defines LLNs as follows:

   LLN: Low-Power and Lossy Network. Typically composed of many
   embedded devices with limited power, memory, and processing
   resources interconnected by a variety of links, such as IEEE
   802.15.4 or low-power Wi-Fi. There is a wide scope of application
   areas for LLNs, including industrial monitoring, building
   automation (heating, ventilation, and air conditioning (HVAC),
   lighting, access control, fire), connected home, health care,
   environmental monitoring, urban sensor networks, energy
   management, assets tracking, and refrigeration.

Beyond that, LLNs often exhibit considerable loss at the physical
layer, with significant variability of the delivery rate, and some
short-term unreliability, coupled with some medium-term stability
that makes it worthwhile to both construct directed acyclic graphs
that are medium-term stable for routing and do measurements on the
edges such as Expected Transmission Count (ETX) [RFC6551]. Not all
LLNs comprise low-power nodes [I-D.hui-vasseur-roll-rpl-deployment].

LLNs typically are composed of constrained nodes; this leads to the
design of operation modes such as the "non-storing mode" defined by
RPL (the IPv6 Routing Protocol for Low-Power and Lossy Networks
[RFC6550]). So, in the terminology of the present document, an LLN
is a constrained-node network with certain network characteristics,
which include constraints on the network as well.

2.3.2. LoWPAN, 6LoWPAN

One interesting class of a constrained network often used as a
constrained-node network is "LoWPAN" [RFC4919], a term inspired from
the name of an IEEE 802.15.4 working group (low-rate wireless
personal area networks (LR-WPANs)). The expansion of the LoWPAN
acronym, "Low-Power Wireless Personal Area Network", contains a hard-
to-justify "Personal" that is due to the history of task group naming
in IEEE 802 more than due to an orientation of LoWPANs around a

single person. Actually, LoWPANs have been suggested for urban
monitoring, control of large buildings, and industrial control
applications, so the "Personal" can only be considered a vestige.
Occasionally, the term is read as "Low-Power Wireless Area Networks"
[WEI]. Originally focused on IEEE 802.15.4, "LoWPAN" (or when used
for IPv6, "6LoWPAN") also refers to networks built from similarly
constrained link-layer technologies [RFC7668] [RFC8105] [RFC7428].

3. Classes of Constrained Devices

Despite the overwhelming variety of Internet-connected devices that
can be envisioned, it may be worthwhile to have some succinct
terminology for different classes of constrained devices.

Before we get to that, let's first distinguish two big rough groups
of devices based on their CPU capabilities:

o Microcontroller-class devices (ARM term: "M-class" [need ref]).
   These often (but not always) include RAM and code storage on chip
   and limit their support for general-purpose operating systems,
   e.g., they do not have an MMU (memory management unit). They use
   most of their pins for interfaces to application hardware such as
   digital in/out (the latter often PWM-controllable), ADC/DACs, etc.
   Where this hardware is specialized for an application, we may talk
   about "Systems on a Chip" (SOC). These devices often implement
   elaborate sleep modes to achieve microwatt- or at least milliwatt-
   level sustained power usage (Ps, see below).

o General-purpose-class devices (ARM term: "A-class"). These
   usually have RAM and Flash storage on separate chips (not always
   separate packages), and offer support for general-purpose
   operating systems such as Linux, e.g. an MMU. Many of the pins on
   the CPU chip are dedidated to interfacing with RAM and other
   memory. Some general-purpose-class devices integrate some
   application hardware such as video controllers, these are often
   called "Systems on a Chip" (SOC). While these chips also include
   sleep modes, they are usually more on the watt side of sustained
   power usage (Ps).

If the distinction between these groups needs to be made in this
document, we distinguish group "M" (microcontroller) from group "J"
(general purpose).

In this document, the class designations in Table 1 may be used as
rough indications of device capabilities. Note that the classes from
10 upwards are not really constrained devices in the sense of the
previous section; they may still be useful to discuss constraints in
larger devices:

Group	Name	data size (e.g., RAM)	code size (e.g., Flash)	Examples
M	Class 0, C0	<< 10 KiB	<< 100 KiB	ATtiny
M	Class 1, C1	~ 10 KiB	~ 100 KiB	STM32F103CB
M	Class 2, C2	~ 50 KiB	~ 250 KiB	STM32F103RC
M	Class 3, C3	~ 100 KiB	~ 500..1000 KiB	STM32F103RG
M	Class 4, C4	~ 300..500..1000 KiB	~ 1000...2000 KiB	"Luxury"
J	Class 10, C10	4-8 MiB	(?)	OpenWRT routers
J		fill in useful...	...J-group classes	
J	Class 15, C13	0.5..1 GiB	(lots)	Raspberry PI
J	Class 16, C15	1..4 GiB	(lots)	Smartphones
J	Class 17, C16	4..32 GiB	(lots)	Laptops
J	Class 19, C19	(lots)	(lots)	Servers

Table 1: Classes of Constrained Devices (KiB = 1024 bytes)

As of the writing of this document, these characteristics correspond to distinguishable clusters of commercially available chips and design cores for constrained devices. While it is expected that the boundaries of these classes will move over time, Moore's law tends to be less effective in the embedded space than in personal computing devices: gains made available by increases in transistor count and density are more likely to be invested in reductions of cost and power requirements than into continual increases in computing power.

Class 0 devices are very constrained sensor-like motes. They are so
severely constrained in memory and processing capabilities that most
likely they will not have the resources required to communicate
directly with the Internet in a secure manner (rare heroic, narrowly
targeted implementation efforts notwithstanding). Class 0 devices
will participate in Internet communications with the help of larger
devices acting as proxies, gateways, or servers. Class 0 devices
generally cannot be secured or managed comprehensively in the
traditional sense. They will most likely be preconfigured (and will
be reconfigured rarely, if at all) with a very small data set. For
management purposes, they could answer keepalive signals and send on/
off or basic health indications.

Class 1 devices are quite constrained in code space and processing
capabilities, such that they cannot easily talk to other Internet
nodes employing a full protocol stack such as using HTTP, Transport
Layer Security (TLS), and related security protocols and XML-based
data representations. However, they are capable enough to use a
protocol stack specifically designed for constrained nodes (such as
the Constrained Application Protocol (CoAP) over UDP [RFC7252]) and
participate in meaningful conversations without the help of a gateway
node. In particular, they can provide support for the security
functions required on a large network. Therefore, they can be
integrated as fully developed peers into an IP network, but they need
to be parsimonious with state memory, code space, and often power
expenditure for protocol and application usage.

Class 2 devices are less constrained and fundamentally capable of
supporting most of the same protocol stacks as used on notebooks or
servers. However, even these devices can benefit from lightweight
and energy-efficient protocols and from consuming less bandwidth.
Furthermore, using fewer resources for networking leaves more
resources available to applications. Thus, using the protocol stacks
defined for more constrained devices on Class 2 devices might reduce
development costs and increase the interoperability.

Constrained devices with capabilities significantly beyond Class 2
devices exist. They are less demanding from a standards development
point of view as they can largely use existing protocols unchanged.
The previous version of the present document therefore did not make
any attempt to define constrained classes beyond Class 2. These
devices, and to a certain extent even J-group devices, can still be
constrained by a limited energy supply. Class 3 and 4 devices are
less clearly defined than the lower classes; they are even less
constrained. In particular Class 4 devices are powerful enough to
quite comfortably run, e.g., JavaScript interpreters, together with
elaborate network stacks. Additional classes may need to be defined

based on protection capabilities, e.g., an MPU (memory protection
unit; true MMUs are typically only found in J-group devices).

With respect to examining the capabilities of constrained nodes,
particularly for Class 1 devices, it is important to understand what
type of applications they are able to run and which protocol
mechanisms would be most suitable. Because of memory and other
limitations, each specific Class 1 device might be able to support
only a few selected functions needed for its intended operation. In
other words, the set of functions that can actually be supported is
not static per device type: devices with similar constraints might
choose to support different functions. Even though Class 2 devices
have some more functionality available and may be able to provide a
more complete set of functions, they still need to be assessed for
the type of applications they will be running and the protocol
functions they would need. To be able to derive any requirements,
the use cases and the involvement of the devices in the application
and the operational scenario need to be analyzed. Use cases may
combine constrained devices of multiple classes as well as more
traditional Internet nodes.

3.1. Firmware/Software upgradeability

Platforms may differ in their firmware or software upgradeability.
The below is a first attempt at classifying this.

```
+------+-------------------------------------------------------------+
| Name | Firmware/Software upgradeability                            |
+------+-------------------------------------------------------------+
| F0   | no (discard for upgrade)                                    |
|      |                                                             |
| F1   | replaceable, out of service during replacement, reboot      |
|      |                                                             |
| F2   | patchable during operation, reboot required                 |
|      |                                                             |
| F3   | patchable during operation, restart not visible externally  |
|      |                                                             |
| F9   | app-level upgradeability, no reboot required ("hitless")    |
+------+-------------------------------------------------------------+
```

3.2. Isolation functionality

TBD. This section could discuss the ability of the platform to
isolate different components. The categories below are not mutually
exclusive; we need to build relevant clusters.

```
+------+-------------------------------------------------------------+
| Name | Isolation functionality                                     |
+------+-------------------------------------------------------------+
| Is0  | no isolation                                                |
|      |                                                             |
| Is2  | MPU (memory protection unit), at least boundary registers |
|      |                                                             |
| Is5  | MMU with Linux-style kernel/user                            |
|      |                                                             |
| Is7  | Virtualization-style isolation                             |
|      |                                                             |
| Is8  | Secure enclave isolation                                    |
+------+-------------------------------------------------------------+
```

3.3. Shielded secrets

Some platforms can keep shielded secrets (usually in conjuction with
secure enclave functionality).

4. Power Terminology

Devices not only differ in their computing capabilities but also in
available power and/or energy. While it is harder to find
recognizable clusters in this space, it is still useful to introduce
some common terminology.

4.1. Scaling Properties

The power and/or energy available to a device may vastly differ, from
kilowatts to microwatts, from essentially unlimited to hundreds of
microjoules.

Instead of defining classes or clusters, we simply state, using the
International System of Units (SI units), an approximate value for
one or both of the quantities listed in Table 2:

```
+------+-----------------------------------------------+---------+
| Name | Definition                                    | SI Unit |
+------+-----------------------------------------------+---------+
| Ps   | Sustainable average power available for the   | W       |
|      | device over the time it is functioning        | (Watt)  |
|      |                                               |         |
| Et   | Total electrical energy available before the  | J       |
|      | energy source is exhausted                    | (Joule) |
+------+-----------------------------------------------+---------+
```

Table 2: Quantities Relevant to Power and Energy

The value of Et may need to be interpreted in conjunction with an indication over which period of time the value is given; see Section 4.2.

Some devices enter a "low-power" mode before the energy available in a period is exhausted or even have multiple such steps on the way to exhaustion. For these devices, Ps would need to be given for each of the modes/steps.

4.2. Classes of Energy Limitation

As discussed above, some devices are limited in available energy as opposed to (or in addition to) being limited in available power. Where no relevant limitations exist with respect to energy, the device is classified as E9. The energy limitation may be in total energy available in the usable lifetime of the device (e.g., a device that is discarded when its non-replaceable primary battery is exhausted), classified as E2. Where the relevant limitation is for a specific period, the device is classified as E1, e.g., a solar-powered device with a limited amount of energy available for the night, a device that is manually connected to a charger and has a period of time between recharges, or a device with a periodic (primary) battery replacement interval. Finally, there may be a limited amount of energy available for a specific event, e.g., for a button press in an energy-harvesting light switch; such devices are classified as E0. Note that, in a sense, many E1 devices are also E2, as the rechargeable battery has a limited number of useful recharging cycles.

Table 3 provides a summary of the classifications described above.

```
+------+-----------------------------+----------------------------+
| Name | Type of energy limitation   | Example Power Source       |
+------+-----------------------------+----------------------------+
| E0   | Event energy-limited        | Event-based harvesting     |
|      |                             |                            |
| E1   | Period energy-limited       | Battery that is            |
|      |                             | periodically recharged or  |
|      |                             | replaced                   |
|      |                             |                            |
| E2   | Lifetime energy-limited     | Non-replaceable primary    |
|      |                             | battery                    |
|      |                             |                            |
| E9   | No direct quantitative      | Mains-powered              |
|      | limitations to available    |                            |
|      | energy                      |                            |
+------+-----------------------------+----------------------------+
```

Table 3: Classes of Energy Limitation

4.3. Strategies for Using Power for Communication

Especially when wireless transmission is used, the radio often
consumes a big portion of the total energy consumed by the device.
Design parameters, such as the available spectrum, the desired range,
and the bitrate aimed for, influence the power consumed during
transmission and reception; the duration of transmission and
reception (including potential reception) influence the total energy
consumption.

Different strategies for power usage and network attachment may be
used, based on the type of the energy source (e.g., battery or mains-
powered) and the frequency with which a device needs to communicate.

The general strategies for power usage can be described as follows:

Always-on: This strategy is most applicable if there is no reason
   for extreme measures for power saving. The device can stay on in
   the usual manner all the time. It may be useful to employ power-
   friendly hardware or limit the number of wireless transmissions,
   CPU speeds, and other aspects for general power-saving and cooling
   needs, but the device can be connected to the network all the
   time.

Normally-off: Under this strategy, the device sleeps such long
   periods at a time that once it wakes up, it makes sense for it to
   not pretend that it has been connected to the network during
   sleep: the device reattaches to the network as it is woken up.
   The main optimization goal is to minimize the effort during the

reattachment process and any resulting application communications.
If the device sleeps for long periods of time and needs to
communicate infrequently, the relative increase in energy
expenditure during reattachment may be acceptable.

Low-power: This strategy is most applicable to devices that need to
operate on a very small amount of power but still need to be able
to communicate on a relatively frequent basis. This implies that
extremely low-power solutions need to be used for the hardware,
chosen link-layer mechanisms, and so on. Typically, given the
small amount of time between transmissions, despite their sleep
state, these devices retain some form of attachment to the
network. Techniques used for minimizing power usage for the
network communications include minimizing any work from re-
establishing communications after waking up and tuning the
frequency of communications (including "duty cycling", where
components are switched on and off in a regular cycle) and other
parameters appropriately.

Table 4 provides a summary of the strategies described above.

```
+------+--------------+---------------------------------------------+
| Name | Strategy     | Ability to communicate                      |
+------+--------------+---------------------------------------------+
| P0   | Normally-off | Reattach when required                      |
|      |              |                                             |
| P1   | Low-power    | Appears connected, perhaps with high        |
|      |              | latency                                     |
|      |              |                                             |
| P9   | Always-on    | Always connected                            |
+------+--------------+---------------------------------------------+
```

Table 4: Strategies of Using Power for Communication

Note that the discussion above is at the device level; similar
considerations can apply at the communications-interface level. This
document does not define terminology for the latter.

A term often used to describe power-saving approaches is "duty-
cycling". This describes all forms of periodically switching off
some function, leaving it on only for a certain percentage of time
(the "duty cycle").

[RFC7102] only distinguishes two levels, defining a Non-Sleepy Node
as a node that always remains in a fully powered-on state (always
awake) where it has the capability to perform communication (P9) and
a Sleepy Node as a node that may sometimes go into a sleep mode (a

low-power state to conserve power) and temporarily suspend protocol
communication (P0); there is no explicit mention of P1.

4.4. Strategies of Keeping Time over Power Events

[This subsection is very drafty.]

Many applications for a device require it to keep some concept of
time.

Time-keeping can be relative to a previous event (last packet
received), absolute on a device-specific scale (e.g., last reboot),
or absolute on a world-wide scale ("wall-clock time").

Some devices lose the concept of time when going to sleep: after
wakeup, they don't know how long they slept. Some others do keep
some concept of time during sleep, but not precise enough to use as a
basis for keeping absolute time. Some devices have a continuously
running source of a reasonably accurate time (often a 32,768 Hz watch
crystal). Finally, some devices can keep their concept of time even
during a battery change, e.g., by using a backup battery or a
supercapacitor to power the real-time clock (RTC).

The actual accuracy of time may vary, with errors ranging from tens
of percent from on-chip RC oscillators (not useful for keeping
absolute time, but still useful for, e.g., timing out some state) to
approximately 1e-4 to 1e-5 ("watch crystal") of error. More precise
timing is available with temperature compensated crystal oscillators
(TCXO). Further improvement requires significantly higher power
usage, bulk, fragility, and device cost, e.g. oven-controlled crystal
oscillators (OCXO) can reach 1e-8, and Rubidium frequency sources can
reach 1e-11 over the short term and 1e-9 over the long term.

A device may need to fire up a more accurate frequency source during
wireless communication, this may also allow it to keep more precise
time during the period.

The various time sources available on the device can be assisted by
external time input, e.g. via the network using the NTP protocol
[RFC5905]. Information from measuring the deviation between external
input and local time source can be used to increase the accuracy of
maintaining time even during periods of no network use.

Errors of the frequency source can be compensated if known
(calibrated against a known better source, or even predicted, e.g.,
in a software TCXO). Even with errors partially compensated, an
uncertainty remains, which is the more fundamental characteristic to
discuss.

   Battery solutions may allow the device to keep a wall-clock time
   during its entire life, or the wall-clock time may need to be reset
   after a battery change. Even devices that have a battery lasting for
   their lifetime may not be set to wall-clock time at manufacture time,
   possibly because the battery is only activated at installation time
   where time sources may be questionable or because setting the clock
   during manufacture is deemed too much effort.

   Devices that keep a good approximation of wall-clock time during
   their life may be in a better position to securely validate external
   time inputs than devices that need to be reset episodically, which
   can possibly be tricked by their environment into accepting a long-
   past time, for instance with the intent of exploiting expired
   security assertions such as certificates.

   From a practical point of view, devices can be divided at least on
   the two dimensions proposed in Table 5 and Table 6. Corrections to
   the local time of a device performed over the network can be used to
   improve the uncertainty exhibited by these basic device classes.

```
   +------+-----------------------------+---------------------------+
   | Name | Type                        | Uncertainty (roughly)     |
   +------+-----------------------------+---------------------------+
   | T0   | relative time while awake   | (usually high)            |
   |      |                             |                           |
   | T1   | relative time               | (usually high during      |
   |      |                             | sleep)                    |
   |      |                             |                           |
   | T2   | relative time               | 1e-4 or better            |
   |      |                             |                           |
   | T5   | absolute time (e.g., since  | 1e-4 or better            |
   |      | boot)                       |                           |
   |      |                             |                           |
   | T7   | wall-clock time             | 1e-4 or better            |
   |      |                             |                           |
   | T8   | wall-clock time             | 1e-5 or better            |
   |      |                             |                           |
   | T9   | wall-clock time             | 1e-6 or better (TCXO)     |
   |      |                             |                           |
   | T10  | wall-clock time             | 1e-7 or better (OCXO or   |
   |      |                             | Rb)                       |
   +------+-----------------------------+---------------------------+
```

            Table 5: Strategies of Keeping Time over Power Events

```
+------+-------------------------------+-----------------------+
| Name | Permanency (from type T5      | Uncertainty           |
|      | upwards):                     |                       |
+------+-------------------------------+-----------------------+
| TP0  | time needs to be reset on certain |                   |
|      | occasions                     |                       |
|      |                               |                       |
| TP1  | time needs to be set during   | (possibly reduced...  |
|      | installation                  |                       |
|      |                               |                       |
| TP9  | reliable time is maintained   | ...by using external  |
|      | during lifetime               | input)                |
+------+-------------------------------+-----------------------+
```

Table 6: Permanency of Keeping Time

5. Classes of Networks

5.1. Classes of link layer MTU size

Link layer technologies used by constrained devices can be
categorized on the basis of link layer MTU size. Depending on this
parameter, the fragmentation techniques needed (if any) to support
the IPv6 MTU requirement may vary.

We define the following classes of link layer MTU size:

```
+------+--------------------+----------------------------------+
| Name | L2 MTU size (bytes) | 6LoWPAN Fragmentation applicable*? |
+------+--------------------+----------------------------------+
| S0   | 3 - 12             | need new kind of fragmentation   |
|      |                    |                                  |
| S1   | 13 - 127           | yes                              |
|      |                    |                                  |
| S2   | 128 - 1279         | yes                              |
|      |                    |                                  |
| S2   | >= 1280            | no fragmentation needed          |
+------+--------------------+----------------------------------+
```

*if no link layer fragmentation is available
(note: 'Sx' stands for 'Size x')

S0 technologies require fragmentation to support the IPv6 MTU
requirement. If no link layer fragmentation is available,
fragmentation is needed at the adaptation layer below IPv6. However,
6LoWPAN fragmentation [RFC4944] cannot be used for these
technologies, given the extremely reduced link layer MTU. In this

case, lightweight fragmentation formats must be used (e.g.
[I-D.ietf-lpwan-ipv6-static-context-hc]).

S1 and S2 technologies require fragmentation at the subnetwork level
to support the IPv6 MTU requirement. If link layer fragmentation is
unavailable or insufficient, fragmentation is needed at the
adaptation layer below IPv6. 6LoWPAN fragmentation [RFC4944] can be
used to carry 1280-byte IPv6 packets over these technologies.

S3 technologies do not require fragmentation to support the IPv6 MTU
requirement.

5.2. Class of Internet Integration

The term "Internet of Things" is sometimes confusingly used for
connected devices that are not actually employing Internet
technology. Some devices do use Internet technology, but only use it
to exchange packets with a fixed communication partner ("device-to-
cloud" scenarios, [RFC7452]). More general devices are prepared to
communicate with other nodes in the Internet as well.

We define the following classes of Internet technology level:

```
        +------+--------------------------------------+
        | Name | Internet technology                  |
        +------+--------------------------------------+
        | I0   | none (local interconnect only)       |
        |      |                                      |
        | I1   | device-to-cloud only                 |
        |      |                                      |
        | I9   | full Internet connectivity supported |
        +------+--------------------------------------+
```

5.3. Classes of physical layer bit rate

[This section is a trial balloon. We could also talk about burst
rate, sustained rate; bits/s, messages/s, ...]

Physical layer technologies used by constrained devices can be
categorized on the basis of physical layer (PHY) bit rate. The PHY
bit rate class of a technology has important implications with regard
to compatibility with existing protocols and mechanisms on the
Internet, responsiveness to frame transmissions and need for header
compression techniques.

We define the following classes of PHY bit rate:

```
+------+-----------+-------------------------------------------------+
| Name | PHY bit   | Comment                                         |
|      | rate      |                                                 |
|      | (bit/s)   |                                                 |
+------+-----------+-------------------------------------------------+
| B0   | < 10      | Tx time of 150-byte frame > MSL                 |
|      |           |                                                 |
| B1   | 10 - 10^3 | Unresponsiveness if human expects reaction to   |
|      |           | sent frame (frame size > 62.5 byte)             |
|      |           |                                                 |
| B2   | 10^3 -    | Responsiveness if human expects reaction to     |
|      | 10^6      | sent frame, but header compression still        |
|      |           | needed                                          |
|      |           |                                                 |
| B3   | > 10^6    | Header compression yields relatively low        |
|      |           | performance benefits                            |
+------+-----------+-------------------------------------------------+
```

(note: 'Bx' stands for 'Bit rate x')

B0 technologies lead to very high transmission times, which may be
close to or even greater than the Maximum Segment Lifetime (MSL)
assumed on the Internet [RFC0793]. Many Internet protocols and
mechanisms will fail when transmit times are greater than the MSL.
B0 technologies lead to a frame transmission time greater than the
MSL for a frame size greater than 150 bytes.

B1 technologies offer transmission times which are lower than the MSL
(for a frame size greater than 150 bytes). However, transmission
times for B1 technologies are still significant if a human expects a
reaction to the transmission of a frame. With B1 technologies, the
transmission time of a frame greater than 62.5 bytes exceeds 0.5
seconds, i.e. a threshold time beyond which any response or reaction
to a frame transmission will appear not to be immediate [RFC5826].

B2 technologies do not incur responsiveness problems, but still
benefit from using header compression techniques (e.g. [RFC6282]) to
achieve performance improvements.

Over B3 technologies, the relative performance benefits of header
compression are low. For example, in a duty-cycled technology
offering B3 PHY bit rates, energy consumption decrease due to header
compression may be comparable with the energy consumed while in a
sleep interval. On the other hand, for B3 PHY bit rates, a human
user will not be able to perceive whether header compression has been
used or not in a frame transmission.

6. IANA Considerations

   This document makes no requests of IANA.

7. Security Considerations

   This document introduces common terminology that does not raise any
   new security issues. Security considerations arising from the
   constraints discussed in this document need to be discussed in the
   context of specific protocols. For instance, Section 11.6 of
   [RFC7252], "Constrained node considerations", discusses implications
   of specific constraints on the security mechanisms employed.
   [RFC7416] provides a security threat analysis for the RPL routing
   protocol. Implementation considerations for security protocols on
   constrained nodes are discussed in [RFC7815] and
   [I-D.ietf-lwig-tls-minimal]. A wider view of security in
   constrained-node networks is provided in [RFC8576].

8. Informative References

   [FALL]   Fall, K., "A Delay-Tolerant Network Architecture for
              Challenged Internets", SIGCOMM 2003,
              DOI 10.1145/863955.863960, 2003.

   [FIFTY-BILLION]
              Ericsson, "More Than 50 Billion Connected Devices",
              Ericsson White Paper 284 23-3149 Uen, February 2011,
              <http://www.ericsson.com/res/docs/whitepapers/
              wp-50-billions.pdf>.

   [I-D.hui-vasseur-roll-rpl-deployment]
              Vasseur, J., Hui, J., Dasgupta, S., and G. Yoon, "RPL
              deployment experience in large scale networks", draft-hui-
              vasseur-roll-rpl-deployment-01 (work in progress), July
              2012.

   [I-D.ietf-lpwan-ipv6-static-context-hc]
              Minaburo, A., Toutain, L., Gomez, C., Barthel, D., and J.
              Zuniga, "Static Context Header Compression (SCHC) and
              fragmentation for LPWAN, application to UDP/IPv6", draft-
              ietf-lpwan-ipv6-static-context-hc-21 (work in progress),
              July 2019.

   [I-D.ietf-lwig-tls-minimal]
              Kumar, S., Keoh, S., and H. Tschofenig, "A Hitchhiker's
              Guide to the (Datagram) Transport Layer Security Protocol
              for Smart Objects and Constrained Node Networks", draft-
              ietf-lwig-tls-minimal-01 (work in progress), March 2014.

   [ISQ-13]   International Electrotechnical Commission, "International
              Standard -- Quantities and units -- Part 13: Information
              science and technology", IEC 80000-13, March 2008.

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

   [RFC4838] Cerf, V., Burleigh, S., Hooke, A., Torgerson, L., Durst,
              R., Scott, K., Fall, K., and H. Weiss, "Delay-Tolerant
              Networking Architecture", RFC 4838, DOI 10.17487/RFC4838,
              April 2007, <https://www.rfc-editor.org/info/rfc4838>.

   [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>.

   [RFC5905] Mills, D., Martin, J., Ed., Burbank, J., and W. Kasch,
              "Network Time Protocol Version 4: Protocol and Algorithms
              Specification", RFC 5905, DOI 10.17487/RFC5905, June 2010,
              <https://www.rfc-editor.org/info/rfc5905>.

   [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>.

   [RFC6551] Vasseur, JP., Ed., Kim, M., Ed., Pister, K., Dejean, N.,
              and D. Barthel, "Routing Metrics Used for Path Calculation
              in Low-Power and Lossy Networks", RFC 6551,
              DOI 10.17487/RFC6551, March 2012,
              <https://www.rfc-editor.org/info/rfc6551>.

   [RFC6606] Kim, E., Kaspar, D., Gomez, C., and C. Bormann, "Problem
              Statement and Requirements for IPv6 over Low-Power
              Wireless Personal Area Network (6LoWPAN) Routing",
              RFC 6606, DOI 10.17487/RFC6606, May 2012,
              <https://www.rfc-editor.org/info/rfc6606>.

   [RFC7102] Vasseur, JP., "Terms Used in Routing for Low-Power and
              Lossy Networks", RFC 7102, DOI 10.17487/RFC7102, January
              2014, <https://www.rfc-editor.org/info/rfc7102>.

   [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>.

   [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>.

   [RFC7416] Tsao, T., Alexander, R., Dohler, M., Daza, V., Lozano, A.,
              and M. Richardson, Ed., "A Security Threat Analysis for
              the Routing Protocol for Low-Power and Lossy Networks
              (RPLs)", RFC 7416, DOI 10.17487/RFC7416, January 2015,
              <https://www.rfc-editor.org/info/rfc7416>.

   [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>.

   [RFC7452] Tschofenig, H., Arkko, J., Thaler, D., and D. McPherson,
              "Architectural Considerations in Smart Object Networking",
              RFC 7452, DOI 10.17487/RFC7452, March 2015,
              <https://www.rfc-editor.org/info/rfc7452>.

   [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>.

   [RFC7815] Kivinen, T., "Minimal Internet Key Exchange Version 2
              (IKEv2) Initiator Implementation", RFC 7815,
              DOI 10.17487/RFC7815, March 2016,
              <https://www.rfc-editor.org/info/rfc7815>.

   [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>.

   [RFC8576] Garcia-Morchon, O., Kumar, S., and M. Sethi, "Internet of
              Things (IoT) Security: State of the Art and Challenges",
              RFC 8576, DOI 10.17487/RFC8576, April 2019,
              <https://www.rfc-editor.org/info/rfc8576>.

   [WEI]   Shelby, Z. and C. Bormann, "6LoWPAN: the Wireless Embedded
              Internet", Wiley-Blackwell monograph,
              DOI 10.1002/9780470686218, ISBN 9780470747995, 2009.

Authors' Addresses

   Carsten Bormann
   Universitaet Bremen TZI
   Postfach 330440
   Bremen D-28359
   Germany

   Phone: +49-421-218-63921
   EMail: cabo@tzi.org

   Mehmet Ersue
   Nokia Solutions and Networks
   St.-Martinstrasse 76
   Munich 81541
   Germany

   Phone: +49 172 8432301
   EMail: mehmet.ersue@nsn.com

   Ari Keranen
   Ericsson
   Hirsalantie 11
   Jorvas 02420
   Finland

   EMail: ari.keranen@ericsson.com

   Carles Gomez
   UPC/i2CAT
   Escola d'Enginyeria de Telecomunicacio i Aeroespacial
   de Castelldefels
   C/Esteve Terradas, 7
   Castelldefels 08860
   Spain

   Phone: +34-93-413-7206
   EMail: carlesgo@entel.upc.edu