Internet Engineering Task Force (IETF)
Request for Comments: 7547
Category: Informational
ISSN: 2070-1721

M. Ersue, Ed.
Nokia Networks
D. Romascanu
Avaya
J. Schoenwaelder
Jacobs University Bremen
U. Herberg

May 2015

Management of Networks with Constrained Devices: Problem Statement and Requirements

Abstract

This document provides a problem statement, deployment and management topology options, as well as requirements addressing the different use cases of the management of networks where constrained devices are involved.

Status of This Memo

This document is not an Internet Standards Track specification; it is published for informational purposes.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Not all documents approved by the IESG are a candidate for any level of Internet Standard; see Section 2 of RFC 5741.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at http://www.rfc-editor.org/info/rfc7547.

Table of Contents

1. Introduction

1.1. Overview

Constrained devices (also known as sensors, smart objects, or smart devices) with limited CPU, memory, and power resources can be connected to a network. It might be based on unreliable or lossy channels, it may use wireless technologies with limited bandwidth and a dynamic topology, or it may need the service of a gateway or proxy to connect to the Internet. In other scenarios, the constrained devices can be connected to a unconstrained network using off-the-shelf protocol stacks.

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. Constrained devices may also work under severe resource constraints such as limited battery and computing power, little memory and insufficient wireless bandwidth, and communication capabilities. A central entity, e.g., a base station or controlling server, might have more computational and communication resources and can act as a gateway between the constrained devices and the application logic in the core network.

Today, constrained devices of diverse size and with different resources and capabilities are being connected. Mobile personal gadgets, building-automation devices, cellular phones, machine-to-machine (M2M) devices, etc., benefit from interacting with other "things" in the near or somewhere in the Internet. With this the Internet of Things (IoT) becomes a reality, built up of uniquely identifiable objects (things). And over the next decade, this could grow to trillions of constrained devices and will greatly increase the Internet’s size and scope.

Network management is characterized by monitoring network status, detecting faults (and inferring their causes), setting network parameters, and carrying out actions to remove faults, maintain normal operation, and improve network efficiency and application performance. The traditional network monitoring application periodically collects information from a set of managed network elements, it processes the data, and it presents the results to the network management users. Constrained devices, however, often have limited power, have low transmission range, and might be unreliable. They might also need to work in hostile environments with advanced security requirements or need to be used in harsh environments for a long time without supervision. Due to such constraints, the

management of a network with constrained devices faces a different type of challenges compared to the management of a traditional IP network.

The IETF has already done substantial standardization work to enable communication in IP networks and to manage such networks as well as the manifold types of nodes in these networks [RFC6632]. However, the IETF so far has not developed any specific technologies for the management of constrained devices and the networks comprised by constrained devices. IP-based sensors or constrained devices in such an environment (i.e., devices with very limited memory, CPU, and energy resources) nowadays use application-layer protocols in an ad hoc manner to do simple resource management and monitoring.

This document provides a problem statement and lists requirements for the different use cases of management of a network with constrained devices. Sections 1.3 and 1.5 describe different topology options for the networking and management of constrained devices. Section 2 provides a problem statement on the issue of the management of networked constrained devices. Section 3 lists requirements on the management of applications and networks with constrained devices. Note that the requirements listed in Section 3 have been separated from the context in which they may appear. Depending on the concrete circumstances, an implementer may decide to address a certain relevant subset of the requirements.

The use cases in the context of networks with constrained devices can be found in [RFC7548]. This document provides a list of objectives for discussions and does not aim to be a strict requirements document for all use cases. In fact, there likely is not a single solution that works equally well for all the use cases.

1.2. Terminology

Concerning constrained devices and networks, this document generally builds on the terminology defined in [RFC7228], where the terms "constrained device", "constrained network", and others are defined.

Additionally, the following terms are used throughout:

AMI: (Advanced Metering Infrastructure) A system including hardware, software, and networking technologies that measures, collects, and analyzes energy use and that communicates with a hierarchically deployed network of metering devices, either on request or on a schedule.

C0: Class 0 constrained device as defined in Section 3 of [RFC7228].

C1: Class 1 constrained device as defined in Section 3 of [RFC7228].

C2: Class 2 constrained device as defined in Section 3 of [RFC7228].

Network of Constrained Devices: A network to which constrained devices are connected that may or may not be a constrained network (see [RFC7228] for the definition of the term constrained network).

M2M: (Machine to Machine) The automatic data transfer between devices of different kinds. In M2M scenarios, a device (such as a sensor or meter) captures an event, which is relayed through a network (wireless, wired, or hybrid) to an application.

MANET: (Mobile Ad Hoc Network [RFC2501]) A self-configuring and infrastructureless network of mobile devices connected by wireless technologies.

Smart Grid: An electrical grid that uses communication technologies to gather and act on information in an automated fashion to improve the efficiency, reliability, and sustainability of the production and distribution of electricity.

Smart Meter: An electrical meter in the context of a smart grid.

For a detailed discussion on the constrained networks as well as classes of constrained devices and their capabilities, please see [RFC7228].

1.3. Network Types and Characteristics in Focus

In this document, we differentiate the following types of networks concerning their transport and communication technologies:

(Note that a network in general can involve constrained and unconstrained devices.)

1. Wireline unconstrained networks, e.g., an Ethernet LAN with constrained and unconstrained devices involved.

2. A combination of wireline and wireless networks, possibly with a multi-hop connectivity between constrained devices, utilizing dynamic routing in both the wireless and wireline portions of the network. Such networks usually support highly distributed applications with many nodes (e.g., environmental monitoring) and

tend to deal with large-scale multipoint-to-point (MP2P) systems. Wireless Mesh Networks (WMNs), as a specific variant, use off-the-shelf radio technology such as Wi-Fi, WiMAX, and cellular 3G/4G. WMNs are reliable based on the redundancy they offer and have often a more planned deployment to provide dynamic and cost effective connectivity over a certain geographic area.

3. A combination of wireline and wireless networks with point-to-point (P2P) or point-to-multipoint (P2MP) communication generally with single-hop connectivity to constrained devices, utilizing static routing over the wireless network. Such networks support short-range, P2P, low-data-rate, source-to-sink types of applications, such as RFID systems, light switches, fire/smoke detectors, and home appliances. This type of network also supports confined short-range spaces such as a home, a factory, a building, or the human body. [IEEE802.15.1] (Bluetooth) and [IEEE802.15.4] are well-known examples of applicable standards for such networks. By using 6LoWPANs (IPv6 over Low-Power Wireless Personal Area Networks) [RFC4919] and RPL (Routing Protocol for Low-Power and Lossy Networks) [RFC6550] on top of IEEE 802.15.4, multi-hop connectivity and dynamic routing can be achieved. With RPL, the IETF has specified a proactive "route-over" architecture where routing and forwarding is implemented at the network layer. The protocol provides a mechanism whereby MP2P, P2MP, and P2P traffic are supported.

4. Self-configuring infrastructureless networks of mobile devices (e.g., MANET) are a particular type of network connected by wireless technologies. Infrastructureless networks are mostly based on P2P communications of devices moving independently in any direction and changing the links to other devices frequently. Such devices do act as a router to forward traffic unrelated to their own use.

Wireline unconstrained networks with constrained and unconstrained devices are mainly used for specific applications like Building Automation or Infrastructure Monitoring. Wireline and wireless networks with multi-hop or P2MP connectivity are used, e.g., for environmental monitoring as well as transport and mobile applications.

Furthermore, different network characteristics are determined by multiple dimensions: dynamicity of the topology, bandwidth, and loss rate. In the following, each dimension is explained, and networks in scope for this document are outlined:

Network Topology:

The topology of a network can be represented as a graph, with edges (i.e., links) and vertices (routers and hosts). Examples of different topologies include "star" topologies (with one central node and multiple nodes in one-hop distance), tree structures (with each node having exactly one parent), directed acyclic graphs (with each node having one or more parents), clustered topologies (where one or more "cluster heads" are responsible for a certain area of the network), mesh topologies (fully distributed), etc.

Management protocols may take advantage of specific network topologies, for example, by distributing large-scale management tasks amongst multiple distributed network management stations (e.g., in case of a mesh topology), or by using a hierarchical management approach (e.g., in case of a tree or clustered topology). These different management topology options are described in Section 1.6.

Note that in certain network deployments, such as community ad hoc networks (see the use case "Community Network Applications" in [RFC7548]), the topology is not preplanned; thus, it may be unknown for management purposes. In other use cases, such as industrial applications (see the use case "Industrial Applications" in [RFC7548]), the topology may be designed in advance and therefore taken advantage of when managing the network.

Dynamicity of the network topology:

The dynamicity of the network topology determines the rate of change of the graph as a function of time. Such changes can occur due to different factors, such as mobility of nodes (e.g., in MANETs or cellular networks), duty cycles (for low-power devices enabling their network interface only periodically to transmit or receive packets), or unstable links (in particular wireless links with strongly fluctuating link quality).

Examples of different levels of dynamicity of the topology are Ethernets (with typically a very static topology) on the one side, and Low-power and Lossy Networks (LLNs) on the other side. LLNs nodes are often duty-cycled and operate on unreliable wireless links and are potentially mobile (e.g., for sensor networks).

The more dynamic the topology is, the more have routing, transport and application-layer protocols to cope with interrupted connectivity and/or longer delays. For example, management protocols (with a given underlying transport protocol) that expect continuous session flows without changes of routes during a communication flow, may fail to operate.

Networks with a very low dynamicity (e.g., Ethernet) with no or infrequent topology changes (e.g., less than once every 30 minutes), are in the scope of this document if they are used with constrained devices (see, e.g., the use case "Building Automation" in [RFC7548]).

Traffic flows:

The traffic flow in a network determines from which sources data traffic is sent to which destinations in the network. Several different traffic flows are defined in [RFC7102], including P2P, MP2P, and P2MP flows as:

- P2P: Point-to-point refers to traffic exchanged between two nodes (regardless of the number of hops between the two nodes).

- P2MP: Point-to-multipoint traffic refers to traffic between one node and a set of nodes. This is similar to the P2MP concept in Multicast or MPLS Traffic Engineering.

- MP2P: Multipoint-to-point is used to describe a particular traffic pattern (e.g., MP2P flows collecting information from many nodes flowing inwards towards a collecting sink).

If one of these traffic patterns is predominant in a network, protocols (routing, transport, application) may be optimized for the specific traffic flow. For example, in a network with a tree topology and MP2P traffic, collection tree protocols are efficient to send data from the leaves of the tree to the root of the tree, via each node’s parent.

Bandwidth:

The bandwidth of the network is the amount of data that can be sent per unit of time between two communication endpoints. It is usually determined by the link with the minimum bandwidth on the path from the source to the destination of data packets. The bandwidth in networks can range from a few kilobytes per second (such as on some IEEE 802.15.4 link layers) to many gigabytes per second (e.g., on fiber optics).

For management purposes, the management protocol typically requires the sending of information between the network management station and the clients, for monitoring or control purposes. If the available bandwidth is insufficient for the management protocol, packets will be buffered and eventually dropped; thus, management is not possible with such a protocol.

Networks without bandwidth limitation (e.g., Ethernet) are in the scope of this document if they are used with constrained devices (see the use case "Building Automation" in [RFC7548]).

Loss rate:

The loss rate (or bit error rate) is the number of bit errors divided by the total number of bits transmitted. For wired networks, loss rates are typically extremely low, e.g., around 10^-12 or 10^-13 for the latest 10 Gbit Ethernet. For wireless networks, such as IEEE 802.15.4, the bit error rate can be as high as 10^-1 to 1 in case of interferences. Even when using a reliable transport protocol, management operations can fail if the loss rate is too high, unless they are specifically designed to cope with these situations.

1.4. Constrained Device Deployment Options

We differentiate the following deployment options for the constrained devices:

- A network of constrained devices that communicate with each other,
- Constrained devices that are connected directly to an IP network,
- A network of constrained devices that communicate with a gateway or proxy with more communication capabilities possibly acting as a representative of the device to entities in the unconstrained network,
- Constrained devices that are connected to the Internet or an IP network via a gateway/proxy,
- A hierarchy of constrained devices, e.g., a network of C0 devices connected to one or more C1 devices -- connected to one or more C2 devices -- connected to one or more gateways -- connected to some application servers or NMS, and
- The possibility of device grouping (possibly in a dynamic manner) such as that the grouped devices can act as one logical device at the edge of the network and one device in this group can act as the managing entity.

1.5. Management Topology Options

We differentiate the following options for the management of networks of constrained devices:

- A network of constrained devices managed by one central manager. A logically centralized management might be implemented in a hierarchical fashion for scalability and robustness reasons. The manager and the management application logic might have a gateway/proxy in between or might be on different nodes in different networks, e.g., management application running on a cloud server.

- Distributed management, where a network of constrained devices is managed by more than one manager. Each manager controls a subnetwork and may communicate directly with other manager stations in a cooperative fashion. The distributed management may be weakly distributed, where functions are broken down and assigned to many managers dynamically, or strongly distributed, where almost all managed things have embedded management functionality and explicit management disappears, which usually comes with the price that the strongly distributed management logic now needs to be managed.

- Hierarchical management, where a hierarchy of networks with constrained devices are managed by the managers at their corresponding hierarchy level. That is, each manager is responsible for managing the nodes in its subnetwork. It passes information from its subnetwork to its higher-level manager and disseminates management functions received from the higher-level manager to its subnetwork. Hierarchical management is essentially a scalability mechanism, logically the decision-making may be still centralized.

1.6. Managing the Constrainedness of a Device or Network

The capabilities of a constrained device or network and the constrainedness thereof influence and have an impact on the requirements for the management of such a network or devices.

Note that the list below gives examples and does not claim completeness.

A constrained device:

- might only support an unreliable (e.g., lossy) radio link, i.e., the client and server of a management protocol need to gracefully handle incomplete command exchanges or missing commands.

- might only be able to go online from time to time, where it is reachable, i.e., a command might be necessary to repeat after a longer timeout or the timeout value with which one endpoint waits on a response needs to be sufficiently high.

- might only be able to support a limited operating time (e.g., based on the available battery) or may behave as 'sleepy endpoints', setting their network links to a disconnected state during long periods of time, i.e., the devices need to economize their energy usage with suitable mechanisms and the managing entity needs to monitor and control the energy status of the constrained devices it manages.

- might only be able to support one simple communication protocol, i.e., the management protocol needs to be possible to downscale from constrained (C2) to very constrained (C0) devices with modular implementation and a very basic version with just a few simple commands.

- might only be able to support a communication protocol, which is not IP based.

- might only be able to support limited or no user and/or transport security, i.e., the management system needs to support a less-costly and simple but sufficiently secure authentication mechanism.

- might not be able to support compression and decompression of exchanged data based on limited CPU power, i.e., an intermediary entity which is capable of data compression should be able to communicate with both, devices that support data compression (e.g., C2) and devices that do not support data compression (e.g., C1 and C0).

- might only be able to support a simple encryption, i.e., it would be beneficial if the devices use cryptographic algorithms that are supported in hardware and the encryption used is efficient in terms of memory and CPU usage.

- might only be able to communicate with one single managing entity and cannot support the parallel access of many managing entities.

- might depend on a self-configuration feature, i.e., the managing entity might not know all devices in a network and the device needs to be able to initiate connection setup for the device configuration.

- might depend on self- or neighbor-monitoring features, i.e., the managing entity might not be able to monitor all devices in a network continuously.

- might only be able to communicate with its neighbors, i.e., the device should be able to get its configuration from a neighbor.

- might only be able to support parsing of data models with limited size, i.e., the device data models need to be compact containing the most necessary data and if possible parsable as a stream.

- might only be able to support a limited or no-failure detection, i.e., the managing entity needs to handle the situation, where a failure does not get detected or gets detected late gracefully, e.g., with asking repeatedly.

- might only be able to support the reporting of just one or a limited set failure types.

- might only be able to support a limited set of notifications, possible only an "I am alive." message.

- might only be able to support a soft-reset from failure recovery.

- might possibly generate a large amount of redundant reporting data, i.e., the intermediary management entity (see [RFC7252]) should be able to filter and aggregate redundant data.

A network of constrained devices:

- might only support an unreliable (e.g., lossy) radio link, i.e., the client and server of a management protocol need to repeat commands as necessary or gracefully ignore incomplete commands.

- might be necessary to manage based on multicast communication, i.e., the managing entity needs to be prepared to configure many devices at once based on the same data model.

- might have a very large topology supporting 10,000 or more nodes for some applications and as such node naming is a specific issue for constrained networks.

- needs to support self-organization, i.e., given the large number of nodes and their potential placement in hostile locations and frequently changing topology, manual configuration of nodes is typically not feasible. As such, the network would benefit from the ability to reconfigure itself so that it can continue to operate properly and support reliable connectivity.

- might need a management solution that is energy efficient, using as little wireless bandwidth as possible since communication is highly energy demanding.

- needs to support localization schemes to determine the location of devices since the devices might be moving and location information is important for some applications.

- needs a management solution that is scalable as the network may consist of thousands of nodes and may need to be extended continuously.

- needs to provide fault tolerance. Faults in network operation including hardware and software errors or failures detected by the transport protocol should be handled smoothly. In such a case, it should be possible to run the protocol at a reduced level but avoid failing completely. For example, self-monitoring mechanisms or graceful degradation of features can be used to provide fault tolerance.

- might require new management capabilities, for example, network coverage information and a constrained device power distribution map.

- might require a new management function for data management, since the type and amount of data collected in constrained networks is different from those of the traditional networks.

- might also need energy-efficient key management.

1.7. Configuration and Monitoring Functionality Levels

Devices often differ significantly on the level of configuration management support they provide. This document classifies the configuration management functionality as follows:

CL0: Devices are preconfigured and allow no runtime configuration changes. Configuration parameters are often hard coded and compiled directly into the firmware image.

CL1: Devices have explicit configuration objects. However, changes require a restart of the device to take effect.

CL2: Devices allow management systems to replace the entire configuration (or predetermined subsets) in bulk. Configuration changes take effect by soft-restarts of the system (or subsystems).

CL3: Devices allow management systems to modify configuration objects without bulk replacements and changes take effect immediately.

CL4: Devices support multiple configuration datastores and they might distinguish between the currently running and the next startup configuration.

CL5: Devices support configuration datastore locking and device-local configuration change transactions, i.e., either all configuration changes are applied or none of them are.

CL6: Devices support configuration change transactions across devices.

This document defines a classification of devices with regard to different levels of monitoring support. In general, a device may be in several of the levels listed below:

ML0: Devices push predefined monitoring data.

ML1: Devices allow management systems to pull predefined monitoring data.

ML2: Devices allow management systems to pull user-defined filtered subsets of monitoring data.

ML3: Devices are able to locally process monitoring data in order to detect threshold crossings or to aggregate data.

At the time of this writing, constrained devices often implement a combination of one of CL0-CL2 with one of ML0-ML1.

2. Problem Statement

The terminology for the "Internet of Things" is still nascent, and depending on the network type or layer in focus, diverse technologies and terms are in use. Common to all these considerations is the "Things" or "Objects" are supposed to have physical or virtual identities using interfaces to communicate. In this context, we need

to differentiate between the constrained and smart devices identified by an IP address compared to virtual entities such as Smart Objects, which can be identified as a resource or a virtual object by using a unique identifier. Furthermore, the smart devices usually have limited memory and CPU power as well as aim to be self-configuring and easy to deploy.

However, the constraints of the network nodes require a rethinking of the protocol characteristics concerning power consumption, performance, bandwidth consumption, memory, and CPU usage. As such, there is a demand for protocol simplification, energy-efficient communication, less CPU usage, and a smaller memory footprint.

On the application layer, the IETF is already developing protocols like the Constrained Application Protocol (CoAP) [RFC7252] enabling the communication of constrained devices and networks, e.g., for smart energy applications or home automation environments. In fact, the deployment of such an environment involves many, in some scenarios up to million, constrained devices (e.g., smart meters), which produce a large amount of data. This data needs to be collected, filtered, and preprocessed for further use in diverse services.

Considering the high number of nodes to deploy, one has to think about the manageability aspects of the smart devices and plan for easy deployment, configuration, and management of the networks of constrained devices as well as the devices themselves. Consequently, seamless monitoring and self-configuration of such network nodes becomes more and more imperative. Self-configuration and self-management are already a reality in the standards of some organizations such as 3GPP. To introduce self-configuration of smart devices successfully, a device-initiated connection establishment is often required.

A simple and efficient application-layer protocol, such as CoAP, is essential to address the issue of efficient object-to-object communication and information exchange. Such an information exchange should be done based on interoperable data models to enable the exchange and interpretation of diverse application- and management-related data.

In an ideal world, we would have only one network management protocol for monitoring, configuration, and exchanging management data, independently of the type of the network (e.g., smart grid, wireless access, or core network). Furthermore, it would be desirable to derive the basic data models for constrained devices from the core models used today to enable reuse of functionality and end-to-end information exchange. However, the current management protocols seem

to be too heavyweight compared to the capabilities the constrained devices have and are not applicable directly for use in a network of constrained devices. Furthermore, the data models addressing the requirements of such smart devices need yet to be designed.

So far, the IETF has not developed any specific technologies for the management of constrained devices and the networks comprised by constrained devices. IP-based sensors or constrained devices in such an environment, i.e., today, devices with very limited memory and CPU resources use, e.g., application-layer protocols to do simple resource management and monitoring. This might be sufficient for some basic cases; however, there is a need to reconsider the network management mechanisms based on the new, changed, and reduced requirements coming from smart devices and the network of such constrained devices. Although it is questionable whether we can take the same comprehensive approach we use in an IP network and use it for the management of constrained devices. Hence, the management of a network with constrained devices is necessarily designed in a simplified and less complex manner.

As Section 1.6 highlights, there are diverse characteristics of constrained devices or networks, which stem from their constrainedness and therefore have an impact on the requirements for the management of such a network with constrained devices. The use cases discussed in [RFC7548] show that the requirements on constrained networks are manifold and need to be analyzed from different angles, e.g., concerning the design of the management architecture, the selection of the appropriate protocol features, as well as the specific issues that are new in the context of constrained devices. Examples of such issues are careful management of scarce energy resources, the necessity for self-organization and self-management of such devices but also the implementation considerations to enable the use of common communication technologies on a constrained hardware in an efficient manner. For an exhaustive list of issues and requirements that need to be addressed for the management of a network with constrained devices, please see Sections 1.6 and 3.

3. Requirements on the Management of Networks with Constrained Devices

This section describes the requirements categorized by management areas listed in subsections.

Note that the requirements listed in this section have been separated from the context in which they may appear. In general, this document does not recommend the realization of any subset of the described requirements. As such, this document avoids selecting any of the requirements as mandatory to implement. A device might be able to

provide only a particular selected set of requirements and might not be capable to provide all requirements in this document. On the other hand, a device vendor might select a specific relevant subset of the requirements to implement.

The following template is used for the definition of the requirements.

Req-ID: An ID composed of two numbers: a section number indicating the topic area and a unique three-digit number per section.

Title: The title of the requirement.

Description: The rationale and description of the requirement.

Source: The origin of the requirement and the matching use case or application. For the discussion of referred use cases for constrained management, please see [RFC7548].

Requirement Type: Functional Requirement, Non-functional Requirement. A functional requirement is related to a function or component. As such, functional requirements may be technical details or specific functionality that define what a system is supposed to accomplish. Non-functional requirements (also known as design constraints or quality requirements) impose implementation-related considerations such as performance requirements, security, or reliability.

Device type: The device types by which this requirement can be supported: C0, C1, and/or C2.

Priority: The priority of the requirement showing its importance for a particular type of device: High, Medium, and Low. The priority of a requirement can be High, e.g., for a C2 device, but Low for a C1 or C0 device, as the realization of complex features in a C1 device is in many cases not possible.

3.1. Management Architecture/System

Req-ID: 1.001

Title: Support multiple device classes within a single network

Description: Larger networks usually consist of devices belonging to different device classes (e.g., constrained mesh endpoints and less constrained routers) communicating with each other. Hence, the management architecture must be applicable to networks that have a mix of different device classes. See Section 3 of [RFC7228] for the definition of Constrained Device Classes.

Source: All use cases

Requirement Type: Non-functional Requirement

Device type: C1 and/or C2

Priority: High

Req-ID: 1.002

Title: Management scalability

Description: The management architecture must be able to scale with the number of devices involved and operate efficiently in any network size and topology. This implies that, e.g., the managing entity is able to handle large amounts of device monitoring data and the management protocol is not sensitive to the decrease of the time between two client requests. To achieve good scalability, caching techniques, in-network data aggregation techniques, and hierarchical management models may be used.

Source: General requirement for all use cases to enable large-scale networks

Requirement Type: Non-functional Requirement

Device type: C0, C1, and C2

Priority: High

Req-ID: 1.003

Title: Hierarchical management

Description: Provide a means of hierarchical management, i.e., provide intermediary management entities on different levels, which can take over the responsibility for the management of a subhierarchy of the network of constraint devices. The intermediary management entity can, e.g., support management data aggregation to handle, e.g., high-frequent monitoring data or provide a caching mechanism for the uplink and downlink communication. Hierarchical management contributes to management scalability.

Source: Use cases where a large amount of devices are deployed with a hierarchical topology

Requirement Type: Non-functional Requirement

Device type: Managing and intermediary entities

Priority: Medium

Req-ID: 1.004

Title: Minimize state maintained on constrained devices

Description: The amount of state that needs to be maintained on constrained devices should be minimized. This is important in order to save memory (especially relevant for C0 and C1 devices) and in order to allow devices to restart, for example, to apply configuration changes or to recover from extended periods of inactivity.

Note: One way to achieve this is to adopt a RESTful architecture that minimizes the amount of state maintained by managed constrained devices and that makes resources of a device addressable via URIs.

Source: Basic requirement that concerns all use cases

Requirement Type: Functional Requirement

Device type: C0, C1, and C2

Priority: High

Req-ID: 1.005

Title: Automatic resynchronization with eventual consistency

Description: To support large scale networks, where some constrained devices may be offline at any point in time, it is necessary to distribute configuration parameters in a way that allows temporary inconsistencies but eventually converges, after a sufficiently long period of time without further changes, towards global consistency.

Source: Use cases with large-scale networks with many devices

Requirement Type: Functional Requirement

Device type: C0, C1, and C2

Priority: High

Req-ID: 1.006

Title: Support for lossy links and unreachable devices

Description: Some constrained devices will only be able to support lossy and unreliable links characterized by a limited data rate, a high latency, and a high transmission error rate. Furthermore, constrained devices often duty cycle their radio or the whole device in order to save energy. Some classes of devices labeled as 'sleepy endpoints' set their network links to a disconnected state during long periods of time. In all cases, the management system must not assume that constrained devices are always reachable.

Source: Basic requirement for networks of constrained devices with unreliable links and constrained devices that sleep to save energy

Requirement Type: Non-functional Requirement

Device type: C0, C1, and C2

Priority: High

Req-ID: 1.007

Title: Network-wide configuration

Description: Provide means by which the behavior of the network can be specified at a level of abstraction (network-wide configuration) higher than a set of configuration information specific to individual devices. It is useful to derive the device-specific configuration from the network-wide configuration. Such a repository can be used to configure predefined device or protocol parameters for the whole network. Furthermore, such a network-wide view can be used to monitor and manage a group of routers or a whole network. For example, monitoring the performance of a network requires information additional to what can be acquired from a single router using a management protocol.

Note: The identification of the relevant subset of the policies to be provisioned is according to the capabilities of each device and can be obtained from a preconfigured data-repository.

Source: In general, all use cases of network and device configuration based on a network view in a top-down manner

Requirement Type: Non-functional Requirement

Device type: C0, C1, and C2

Priority: Medium

Req-ID: 1.008

Title: Distributed management

Description: Provide a means of simple distributed management, where a network of constrained devices can be managed or monitored by more than one manager. Since the connectivity to a server cannot be guaranteed at all times, a distributed approach may provide higher reliability, at the cost of increased complexity. This requirement implies the handling of data consistency in case of concurrent read and write access to the device datastore. It might also happen that no management (configuration) server is accessible and the only reachable node is a peer device. In this case, the device should be able to obtain its configuration from peer devices.

Source: Use cases where the count of devices to manage is high

Requirement Type: Non-functional Requirement

Device type: C1 and C2

Priority: Medium

3.2. Management Protocols and Data Models

Req-ID: 2.001

Title: Modular implementation of management protocols

Description: Management protocols should be specified to allow for modular implementations, i.e., it should be possible to implement only a basic set of protocol primitives on highly constrained devices, while devices with additional resources may provide more support for additional protocol primitives. See Section 1.7 for a discussion on the level of configuration management and monitoring support constrained devices may provide.

Source: Basic requirement interesting for all use cases

Requirement Type: Non-functional Requirement

Device type: C0, C1, and C2

Priority: High

Req-ID: 2.002

Title: Compact encoding of management data

Description: The encoding of management data should be compact and space efficient, enabling small message sizes.

Source: General requirement to save memory for the receiver buffer and on-air bandwidth

Requirement Type: Functional Requirement

Device type: C0, C1, and C2

Priority: High

Req-ID: 2.003

Title: Compression of management data or complete messages

Description: Management data exchanges can be further optimized by applying data compression techniques or delta encoding techniques. Compression typically requires additional code size and some additional buffers and/or the maintenance of some additional state information. For C0 devices, compression may not be feasible.

Source: Use cases where it is beneficial to reduce transmission time and bandwidth, e.g., mobile applications that require saving on-air bandwidth

Requirement Type: Functional Requirement

Device type: C1 and C2

Priority: Medium

Req-ID: 2.004

Title: Mapping of management protocol interactions

Description: It is desirable to have a lossless automated mapping between the management protocol used to manage constrained devices and the management protocols used to manage regular devices. In the ideal case, the same core management protocol can be used with certain restrictions taking into account the resource limitations of constrained devices. However, for very resource-constrained devices, this goal might not be achievable.

Source: Use cases where high-frequency interaction with the management system of a unconstrained network is required

Requirement Type: Functional Requirement

Device type: C1 and C2

Priority: Medium

Req-ID: 2.005

Title: Consistency of data models with the underlying information model

Description: The data models used by the management protocol must be consistent with the information model used to define data models for unconstrained networks. This is essential to facilitate the integration of the management of constrained networks with the management of unconstrained networks. Using an underlying information model for future data model design enables further top-down model design and model reuse as well as data interoperability (i.e., exchange of management information between the constrained and unconstrained networks). This is a strong requirement, despite the fact that the underlying information models are often not explicitly documented in the IETF.

Source: General requirement to support data interoperability, consistency, and model reuse

Requirement Type: Non-functional Requirement

Device type: C0, C1, and C2

Priority: High

Req-ID: 2.006

Title: Lossless mapping of management data models

Description: It is desirable to have a lossless automated mapping between the management data models used to manage regular devices and the management data models used for managing constrained devices. In the ideal case, the same core data models can be used with certain restrictions taking into account the resource limitations of constrained devices. However, for very resource-constrained devices, this goal might not be achievable.

Source: Use cases where consistent data exchange with the management system of a unconstrained network is required

Requirement Type: Functional Requirement

Device type: C2

Priority: Medium

Req-ID: 2.007

Title: Protocol extensibility

Description: Provide means of extensibility for the management protocol, i.e., by adding new protocol messages or mechanisms that can deal with changing requirements on a supported message and data types effectively, without causing interoperability problems or having to replace/update large amount of deployed devices.

Source: Basic requirement useful for all use cases

Requirement Type: Functional Requirement

Device type: C0, C1, and C2

Priority: High

3.3. Configuration Management

Req-ID: 3.001

Title: Self-configuration capability

Description: Automatic configuration and reconfiguration of devices without manual intervention. Compared to the traditional management of devices where the management application is the central entity configuring the devices, in the autoconfiguration scenario the device is the active part and initiates the configuration process. Self-configuration can be initiated during the initial configuration or for subsequent configurations, where the configuration data needs to be refreshed. Self-configuration should be also supported during the initialization phase or in the event of failures, where prior knowledge of the network topology is not available or the topology of the network is uncertain.

Source: In general, all use cases requiring easy deployment and plug&play behavior as well as easy maintenance of many constrained devices

Requirement Type: Functional Requirement

Device type: C0, C1, and C2

Priority: High for device categories C0 and C1; Medium for C2

Req-ID: 3.002

Title: Capability discovery

Description: Enable the discovery of supported optional management capabilities of a device and their exposure via at least one protocol and/or data model.

Source: Use cases where the device interaction with other devices or applications is a function of the level of support for its capabilities

Requirement Type: Functional Requirement

Device type: C1 and C2

Priority: Medium

Req-ID: 3.003

Title: Asynchronous transaction support

Description: Provide configuration management with asynchronous (event-driven) transaction support. Configuration operations must support a transactional model, with asynchronous indications that the transaction was completed.

Source: Use cases that require transaction-oriented processing because of reliability or distributed architecture functional requirements

Requirement Type: Functional Requirement

Device type: C1 and C2

Priority: Medium

Req-ID: 3.004

Title: Network reconfiguration

Description: Provide a means of iterative network reconfiguration in order to recover the network from node and communication failures. The network reconfiguration can be failure-driven and self-initiated (automatic reconfiguration). The network reconfiguration can be also performed on the whole hierarchical structure of a network (network topology).

Source: Practically all use cases, as network connectivity is a basic requirement

Requirement Type: Functional Requirement

Device type: C0, C1, and C2

Priority: Medium

3.4. Monitoring Functionality

Req-ID: 4.001

Title: Device status monitoring

Description: Provide a monitoring function to collect and expose information about device status and expose it via at least one management interface. The device monitoring might make use of the hierarchical management through the intermediary entities and the caching mechanism. The device monitoring might also make use of neighbor-monitoring (fault detection in the local network) to support fast fault detection and recovery, e.g., in a scenario where a managing entity is unreachable and a neighbor can take over the monitoring responsibility.

Source: All use cases

Requirement Type: Functional Requirement

Device type: C0, C1, and C2

Priority: High; Medium for neighbor-monitoring

Req-ID: 4.002

Title: Energy status monitoring

Description: Provide a monitoring function to collect and expose information about device energy parameters and usage (e.g., battery level and average power consumption).

Source: Use case "Energy Management"

Requirement Type: Functional Requirement

Device type: C0, C1, and C2

Priority: High for energy reporting devices; Low for others

Req-ID: 4.003

Title: Monitoring of current and estimated device availability

Description: Provide a monitoring function to collect and expose information about current device availability (energy, memory, computing power, forwarding-plane utilization, queue buffers, etc.) and estimation of remaining available resources.

Source: All use cases. Note that monitoring energy resources (like battery status) may be required on all kinds of devices.

Requirement Type: Functional Requirement

Device type: C0, C1, and C2

Priority: Medium

Req-ID: 4.004

Title: Network status monitoring

Description: Provide a monitoring function to collect, analyze, and expose information related to the status of a network or network segments connected to the interface of the device.

Source: All use cases

Requirement Type: Functional Requirement

Device type: C1 and C2

Priority: Low, based on the realization complexity

Req-ID: 4.005

Title: Self-monitoring

Description: Provide self-monitoring (local fault detection) feature for fast fault detection and recovery.

Source: Use cases where the devices cannot be monitored centrally in an appropriate manner, e.g., self-healing is required

Requirement Type: Functional Requirement

Device type: C1 and C2

Priority: High for C2; Medium for C1

Req-ID: 4.006

Title: Performance monitoring

Description: The device will provide a monitoring function to collect and expose information about the basic performance parameter of the device. The performance management functionality might make use of the hierarchical management through the intermediary devices.

Source: Use cases "Building Automation" and "Transport Applications"

Requirement Type: Functional Requirement

Device type: C1 and C2

Priority: Low

Req-ID: 4.007

Title: Fault detection monitoring

Description: The device will provide fault detection monitoring. The system collects information about network states in order to identify whether faults have occurred. In some cases, the detection of the faults might be based on the processing and analysis of the parameters retrieved from the network or other devices. In case of C0 devices, the monitoring might be limited to the check whether or not the device is alive.

Source: Use cases "Environmental Monitoring", "Building Automation", "Energy Management", "Infrastructure Monitoring"

Requirement Type: Functional Requirement

Device type: C0, C1 and C2

Priority: Medium

Req-ID: 4.008

Title: Passive and reactive monitoring

Description: The device will provide passive and reactive monitoring capabilities. The system or manager collects information about device components and network states (passive monitoring) and may perform postmortem analysis of collected data. In case events of interest have occurred, the system or the manager can adaptively react (reactive monitoring), e.g., reconfigure the network. Typically, actions (reactions) will be executed or sent as commands by the management applications.

Source: Diverse use cases relevant for device status and network state monitoring

Requirement Type: Functional Requirement

Device type: C2

Priority: Medium

Req-ID: 4.009

Title: Recovery

Description: Provide local, central and hierarchical recovery mechanisms (recovery is in some cases achieved by recovering the whole network of constrained devices).

Source: Use cases "Industrial Applications", "Home Automation", and "Building Automation", as well as mobile applications that involve different forms of clustering or area managers

Requirement Type: Functional Requirement

Device type: C2

Priority: Medium

Req-ID: 4.010

Title: Network topology discovery

Description: Provide a network topology discovery capability (e.g., use of topology extraction algorithms to retrieve the network state) and a monitoring function to collect and expose information about the network topology.

Source: Use cases "Community Network Applications" and mobile applications

Requirement Type: Functional Requirement

Device type: C1 and C2

Priority: Low, based on the realization complexity

Req-ID: 4.011

Title: Notifications

Description: The device will provide the capability of sending notifications on critical events and faults.

Source: All use cases

Requirement Type: Functional Requirement

Device type: C0, C1, and C2

Priority: Medium for C2; Low for C0 and C1

Req-ID: 4.012

Title: Logging

Description: The device will provide the capability of building, keeping, and allowing retrieval of logs of events (including but not limited to critical faults and alarms).

Source: Use cases "Industrial Applications", "Building Automation", and "Infrastructure Monitoring"

Requirement Type: Functional Requirement

Device type: C2

Priority: High for some medical or industrial applications; Medium otherwise

3.5. Self-Management

Req-ID: 5.001

Title: Self-management -- Self-healing

Description: Enable event-driven and/or periodic self-management functionality in a device. The device should be able to react in case of a failure, e.g., by initiating a fully or partly reset and initiate a self-configuration or management data update as necessary. A device might be further able to check for failures

cyclically or on a schedule in order to trigger self-management as necessary. It is a matter of device design and subject for discussion how much self-management a C1 device can support.

Failure detection and self-management logic are assumed to be generally useful for the self-healing of a device.

Source: The requirement generally relates to all use cases in this document.

Requirement Type: Functional Requirement

Device type: C1 and C2

Priority: High for C2; Medium for C1

3.6. Security and Access Control

Req-ID: 6.001

Title: Authentication of management system and devices

Description: Systems having a management role must be properly authenticated to the device such that the device can exercise proper access control and in particular distinguish rightful management systems from rogue systems. On the other hand, managed devices must authenticate themselves to systems having a management role such that management systems can protect themselves from rogue devices. In certain application scenarios, it is possible that a large number of devices need to be (re-)started at about the same time. Protocols and authentication systems should be designed such that a large number of devices (re-)starting simultaneously does not negatively impact the device authentication process.

Source: Basic security requirement for all use cases

Requirement Type: Functional Requirement

Device type: C0, C1, and C2

Priority: High; Medium for the (re-)start of a large number of devices

---

Req-ID: 6.002

Title: Support suitable security bootstrapping mechanisms

Description: Mechanisms should be supported that simplify the bootstrapping of device that is the discovery of newly deployed devices in order to provide them with appropriate access control permissions.

Source: Basic security requirement for all use cases

Requirement Type: Functional Requirement

Device type: C0, C1, and C2

Priority: High

Req-ID: 6.003

Title: Access control on management system and devices

Description: Systems acting in a management role must provide an access control mechanism that allows the security administrator to restrict which devices can access the managing system (e.g., using an access control white list of known devices). On the other hand, managed constrained devices must provide an access control mechanism that allows the security administrator to restrict how systems in a management role can access the device (e.g., no-access, read-only access, and read-write access).

Source: Basic security requirement for use cases where access control is essential

Requirement Type: Functional Requirement

Device type: C0, C1, and C2

Priority: High

Req-ID: 6.004

Title: Select cryptographic algorithms that are efficient in both code space and execution time

Description: Cryptographic algorithms have a major impact in terms of both code size and overall execution time. Therefore, it is necessary to select mandatory to implement cryptographic algorithms that are reasonable to implement with the available code space and that have a small impact at runtime. Furthermore, some wireless technologies (e.g., IEEE 802.15.4) require the support of certain cryptographic algorithms. It might be useful to choose algorithms that are likely to be supported in wireless chipsets for certain wireless technologies.

Source: Generic requirement to reduce the footprint and CPU usage of a constrained device

Requirement Type: Non-functional Requirement

Device type: C0, C1, and C2

Priority: High; Medium for hardware-supported algorithms

3.7. Energy Management

Req-ID: 7.001

Title: Management of energy resources

Description: Enable managing power resources in the network, e.g., reduce the sampling rate of nodes with critical battery and reduce node transmission power, put nodes to sleep, put single interfaces to sleep, reject a management job based on available energy or criteria predefined by the management application (such as importance levels forcing execution even if the energy level is low), etc. The device may further implement standard data models for energy management and expose it through a management protocol interface, e.g., EMAN MIB modules [RFC7460] and [RFC7461] as well as other EMAN extensions. It might be necessary to use a subset of EMAN MIBs for C1 and C2 devices.

Source: Use case "Energy Management"

Requirement Type: Functional Requirement

Device type: C0, C1, and C2

Priority: Medium for the use case "Energy Management"; Low otherwise

---

Req-ID: 7.002

Title: Support of energy-optimized communication protocols

Description: Use an optimized communication protocol to minimize energy usage for the device (radio) receiver/transmitter, on-air bandwidth usage (i.e., maximize protocol efficiency), and the amount of data communication between nodes. Minimizing data communication implies data aggregation and filtering but also a compact format for the transferred data.

Source: Use cases "Energy Management" and mobile applications

Requirement Type: Non-functional Requirement

Device type: C2

Priority: Medium

Req-ID: 7.003

Title: Support for Layer 2 (L2) energy-aware protocols

Description: The device will support L2 energy-management protocols (e.g., energy-efficient Ethernet [IEEE802.3az]) and be able to report on these.

Source: Use case "Energy Management"

Requirement Type: Non-functional Requirement

Device type: C0, C1, and C2

Priority: Medium

Req-ID: 7.004

Title: Dying gasp

Description: When energy resources draw below the red-line level, the device will send a "dying gasp" notification and perform, if still possible, a graceful shutdown including conservation of critical device configuration and status information.

Source: Use case "Energy Management"

Requirement Type: Functional Requirement

Device type: C0, C1, and C2

Priority: Medium

3.8. Software Distribution

Req-ID: 8.001

Title: Group-based provisioning

Description: Support group-based provisioning, i.e., firmware update and configuration management of a large set of constrained devices with eventual consistency and coordinated reload times. The device should accept group-based configuration management based on bulk commands, which aim similar configurations of a large set of constrained devices of the same type in a given group and which may share a common data model. Activation of configuration may be based on preloaded sets of default values.

Source: All use cases

Requirement Type: Non-functional Requirement

Device type: C0, C1, and C2

Priority: Medium

3.9. Traffic Management

Req-ID: 9.001

Title: Congestion avoidance

Description: Support congestion control principles as defined in [RFC2914], e.g., the ability to avoid congestion by modifying the device’s reporting rate for periodical data (which is usually redundant) based on the importance and reliability level of the management data. This functionality is usually controlled by the managing entity, where the managing entity marks the data as important or relevant for reliability. However, reducing a device’s reporting rate can also be initiated by a device if it is able to detect congestion or has insufficient buffer memory.

Source: Use cases with high reporting rate and traffic, e.g., AMI or M2M

Requirement Type: Non-functional Requirement

Device type: C1 and C2

Priority: Medium

Req-ID: 9.002

Title: Reroute traffic

Description: Provide the ability for network nodes to redirect traffic from overloaded intermediary nodes in a network to another path in order to prevent congestion on a central server and in the primary network.

Source: Use cases with high reporting rate and traffic, e.g., AMI or M2M

Requirement Type: Non-functional Requirement

Device type: Intermediary entity in the network

Priority: Medium

Req-ID: 9.003

Title: Traffic Shaping

Description: Provide the ability to apply traffic-shaping policies to incoming and outgoing links on an overloaded intermediary node (as necessary) in order to reduce the amount of traffic in the network.

Source: Use cases with high reporting rate and traffic, e.g., AMI or M2M

Requirement Type: Non-functional Requirement

Device type: Intermediary entity in the network

Priority: Medium

3.10. Transport Layer

Req-ID: 10.001

Title: Scalable transport layer

Description: Enable the use of a scalable transport layer, i.e., not sensitive to a high rate of incoming client requests, which is useful for applications requiring frequent access to device data.

Source: Applications with frequent access to the device data

Requirement Type: Non-functional Requirement

Device type: C0, C1 and C2

Priority: Medium

Req-ID: 10.002

Title: Reliable unicast transport of messages

Description: Diverse applications need a reliable transport of messages. The reliability might be achieved based on a transport protocol such as TCP or can be supported based on message repetition if an acknowledgment is missing.

Source: Generally, applications benefit from the reliability of the message transport

Requirement Type: Functional Requirement

Device type: C0, C1, and C2

Priority: High

Req-ID: 10.003

Title: Best-effort multicast

Description: Provide best-effort multicast of messages, which is generally useful when devices need to discover a service provided by a server or many devices need to be configured by a managing entity at once based on the same data model.

Source: Use cases where a device needs to discover services as well as use cases with high amount of devices to manage, which are hierarchically deployed, e.g., AMI or M2M

Requirement Type: Functional Requirement

Device type: C0, C1, and C2

Priority: Medium

Req-ID: 10.004

Title: Secure message transport

Description: Enable secure message transport providing authentication, data integrity, and confidentiality by using existing transport-layer technologies with a small footprint such as TLS/DTLS.

Source: All use cases

Requirement Type: Non-functional Requirements

Device type: C1 and C2

Priority: High

3.11. Implementation Requirements

Req-ID: 11.001

Title: Avoid complex application-layer transactions requiring large application-layer messages

Description: Complex application-layer transactions tend to require large memory buffers that are typically not available on C0 or C1 devices and only by limiting functionality on C2 devices. Furthermore, the failure of a single large transaction requires repeating the whole transaction. On constrained devices, it is often more desirable to split a large transaction into a sequence of smaller transactions that require less resources and allow making progress using a sequence of smaller steps.

Source: Basic requirement that concerns all use cases with memory constrained devices

Requirement Type: Non-functional Requirement

Device type: C0, C1, and C2

Priority: High

Req-ID: 11.002

Title: Avoid reassembly of messages at multiple layers in the protocol stack

Description: Reassembly of messages at multiple layers in the protocol stack requires buffers at multiple layers, which leads to inefficient use of memory resources. This can be avoided by making sure the application layer, the security layer, the transport layer, the IPv6 layer, and any adaptation layers are aware of the limitations of each other such that unnecessary fragmentation and reassembly can be avoided. In addition, message size constraints must be announced to protocol peers such that they can adapt and avoid sending messages that can't be processed due to resource constraints on the receiving device.

Source: Basic requirement that concerns all use cases with memory constrained devices

Requirement Type: Non-functional Requirement

Device type: C0, C1, and C2

Priority: High

4. Security Considerations

This document discusses the problem statement and requirements on networks of constrained devices. Section 1.6 mentions a number of limitations that could prevent the implementation of strong cryptographic algorithms. Requirements for security and access control are listed in Section 3.6.

Often, constrained devices might be deployed in unsafe environments where attackers can gain physical access to the devices. As a consequence, it is crucial that devices are robust and tamper resistant, have no backdoors, do not provide services that are not essential for the primary function, and properly protect any security credentials that may be stored on the device (e.g., by using hardware protection mechanisms). Furthermore, it is important that any

credentials leaking from a single device do not simplify the attack on other (similar) devices. In particular, security credentials should never be shared.

Since constrained devices often have limited computational resources, care should be taken in choosing efficient but cryptographically strong cryptographic algorithms. Designers of constrained devices that have a long expected lifetime need to ensure that cryptographic algorithms can be updated once devices have been deployed. The ability to perform secure firmware and software updates is an important management requirement.

Constrained devices might also generate sensitive data or require the processing of sensitive data. Therefore, it is an important requirement to properly protect access to the data in order to protect the privacy of humans using Internet-enabled devices. For certain types of data, protection during the transmission over the network may not be sufficient, and methods should be investigated that provide protection of data while it is cached or stored (e.g., when using a store-and-forward transport mechanism).

5. Informative References

[RFC2914] Floyd, S., "Congestion Control Principles", BCP 41, RFC 2914, DOI 10.17487/RFC2914, September 2000, <http://www.rfc-editor.org/info/rfc2914>.

[RFC2501] Corson, S. and J. Macker, "Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations", RFC 2501, DOI 10.17487/RFC2501, January 1999, <http://www.rfc-editor.org/info/rfc2501>.

[RFC6632] Ersue, M., Ed. and B. Claise, "An Overview of the IETF Network Management Standards", RFC 6632, DOI 10.17487/RFC6632, June 2012, <http://www.rfc-editor.org/info/rfc6632>.

[RFC7102] Vasseur, JP., "Terms Used in Routing for Low-Power and Lossy Networks", RFC 7102, DOI 10.17487/RFC7102, January 2014, <http://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, <http://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, <http://www.rfc-editor.org/info/rfc7252>.

[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, <http://www.rfc-editor.org/info/rfc4919>.

[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, <http://www.rfc-editor.org/info/rfc6550>.

[RFC7460] Chandramouli, M., Claise, B., Schoening, B., Quittek, J., and T. Dietz, "Monitoring and Control MIB for Power and Energy", RFC 7460, DOI 10.17487/RFC7460, March 2015, <http://www.rfc-editor.org/info/rfc7460>.

[RFC7461] Parello, J., Claise, B., and M. Chandramouli, "Energy Object Context MIB", RFC 7461, DOI 10.17487/RFC7461, March 2015, <http://www.rfc-editor.org/info/rfc7461>.

[RFC7548] Ersue, M., Ed., Romascanu, D., Schoenwaelder, J., and A. Sehgal, "Management of Networks with Constrained Devices: Use Cases", RFC 7548, DOI 10.17487/RFC7548, May 2015, <http://www.rfc-editor.org/info/rfc7548>.

[IEEE802.15.4] IEEE, "Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs)", IEEE Standard 802.15.4, September 2011, <https://standards.ieee.org/about/get/802/802.15.html>.

[IEEE802.15.1] IEEE, "Part 15.1: Wireless medium access control (MAC) and physical layer (PHY) specifications for wireless personal area networks (WPANs)", IEEE Standard 802.15.1, June 2005, <https://standards.ieee.org/about/get/802/802.15.html>.

[IEEE802.3az] IEEE, "ETHERNET", IEEE Standard 802.3az, 2012-2014, <https://standards.ieee.org/about/get/802/802.3.html>.

Acknowledgments

The following reviewed and provided valuable comments during the creation of this document:

Dominique Barthel, Andy Bierman, Carsten Bormann, Zhen Cao, Benoit Claise, Hui Deng, Bert Greevenbosch, Joel M. Halpern, Ulrich Herberg, James Nguyen, Anuj Sehgal, Zach Shelby, Peter van der Stok, Thomas Watteyne, and Bert Wijnen.

The authors would like to thank the reviewers and the participants on the Coman and OPSAWG mailing lists for their valuable contributions and comments.

Juergen Schoenwaelder was partly funded by Flamingo, a Network of Excellence project (ICT-318488) supported by the European Commission under its Seventh Framework Programme.

Authors' Addresses

Mehmet Ersue (editor)
Nokia Networks

EMail: mehmet.ersue@nokia.com

Dan Romascanu
Avaya

EMail: dromasca@avaya.com

Juergen Schoenwaelder
Jacobs University Bremen

EMail: j.schoenwaelder@jacobs-university.de

Ulrich Herberg

EMail: ulrich@herberg.name