

Network Working Group
Internet-Draft
Intended status: Informational
Expires: July 7, 2016

J. Arkko
A. Eriksson
A. Keranen
Ericsson
January 4, 2016

Building Power-Efficient CoAP Devices for Cellular Networks
draft-ietf-lwig-cellular-06

Abstract

This memo discusses the use of the Constrained Application Protocol (CoAP) protocol in building sensors and other devices that employ cellular networks as a communications medium. Building communicating devices that employ these networks is obviously well known, but this memo focuses specifically on techniques necessary to minimize power consumption.

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 <http://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 July 7, 2016.

Copyright Notice

Copyright (c) 2016 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of

the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

| | |
|--|----|
| 1. Introduction | 2 |
| 2. Goals for Low-Power Operation | 3 |
| 3. Link-Layer Assumptions | 5 |
| 4. Scenarios | 7 |
| 5. Discovery and Registration | 8 |
| 6. Data Formats | 10 |
| 7. Real-Time Reachable Devices | 10 |
| 8. Sleepy Devices | 11 |
| 8.1. Implementation Considerations | 12 |
| 9. Security Considerations | 13 |
| 10. IANA Considerations | 13 |
| 11. References | 13 |
| 11.1. Normative References | 13 |
| 11.2. Informative References | 14 |
| Appendix A. Acknowledgments | 15 |
| Authors' Addresses | 15 |

1. Introduction

This memo discusses the use of the Constrained Application Protocol (CoAP) protocol [RFC7252] in building sensors and other devices that employ cellular networks as a communications medium. Building communicating devices that employ these networks is obviously well known, but this memo focuses specifically on techniques necessary to minimize power consumption. CoAP has many advantages, including being simple to implement; a thousand lines for the entire software above IP layer is plenty for a CoAP-based sensor, for instance. However, while many of these advantages are obvious and easily obtained, optimizing power consumption remains challenging and requires careful design [I-D.arkko-core-sleepy-sensors].

The memo targets primarily 3GPP cellular networks in their 2G, 3G, and LTE variants and their future enhancements, including possible power efficiency improvements at the radio and link layers. The exact standards or details of the link layer or radios are not relevant for our purposes, however. To be more precise, the material in this memo is suitable for any large-scale, public network that employs point-to-point communications model and radio technology for the devices in the network.

Our focus is devices that need to be optimized for power usage, and on devices that employ CoAP. As a general technology, CoAP is similar to HTTP. It can be used in various ways and network entities

may take on different roles. This freedom allows the technology to be used in efficient and less efficient ways. Some guidance is needed to understand what communication models over CoAP are recommended when low power usage is a critical goal.

The recommendations in this memo should be taken as complementary to device hardware optimization, microelectronics improvements, and further evolution of the underlying link and radio layers. Further gains in power efficiency can certainly be gained on several fronts; the approach that we take in this memo is to do what can be done at the IP, transport, and application layers to provide the best possible power efficiency. Application implementors generally have to use the current generation microelectronics, currently available radio networks and standards, and so on. This focus in our memo should by no means be taken as an indication that further evolution in these other areas is unnecessary. Such evolution is useful, is ongoing, and is generally complementary to the techniques presented in this memo. The evolution of underlying technologies may change what techniques described here are useful for a particular application, however.

The rest of this memo is structured as follows. Section 2 discusses the need and goals for low-power devices. Section 3 outlines our expectations for the low layer communications model. Section 4 describes the two scenarios that we address, and Section 5, Section 6, Section 7 and Section 8 give guidelines for use of CoAP in these scenarios.

2. Goals for Low-Power Operation

There are many situations where power usage optimization is unnecessary. Optimization may not be necessary on devices that can run on power feed over wired communications media, such as in Power-over-Ethernet (PoE) solutions. These devices may require a rudimentary level of power optimization techniques just to keep overall energy costs and aggregate power feed sizes at a reasonable level, but more extreme techniques necessary for battery powered devices are not required. The situation is similar with devices that can easily be connected to mains power. Other types of devices may get an occasional charge of power from energy harvesting techniques. For instance, some environmental sensors can run on solar cells. Typically, these devices still have to regulate their power usage in a strict manner, for instance to be able to use as small and inexpensive solar cells as possible.

In battery operated devices the power usage is even more important. For instance, one of the authors employs over a hundred different sensor devices in his home network. A majority of these devices are

wired and run on PoE, but in most environments this would be impractical because the necessary wires do not exist. The future is in wireless solutions that can cover buildings and other environments without assuming a pre-existing wired infrastructure. In addition, in many cases it is impractical to provide a mains power source. Often there are no power sockets easily available in the locations that the devices need to be in, and even if there were, setting up the wires and power adapters would be more complicated than installing a standalone device without any wires.

Yet, with a large number of devices the battery lifetimes become critical. Cost and practical limits dictate that devices can be largely just bought and left on their own. For instance, with hundred devices, even a ten-year battery lifetime results in a monthly battery change for one device within the network. This may be impractical in many environments. In addition, some devices may be physically difficult to reach for a battery change. Or, a large group of devices -- such as utility meters or environmental sensors -- cannot be economically serviced too often, even if in theory the batteries could be changed.

Many of these situations lead to a requirement for minimizing power usage and/or maximizing battery lifetimes. Using the power usage strategies described in [RFC7228], mains-powered sensor-type devices can use the Always-on strategy whereas battery or energy harvesting devices need to adjust behavior based on the communication interval. For intervals in the order of seconds, Low-power strategy is appropriate. For intervals ranging from minutes to hours either Low-power or Normally-off strategies are suitable. Finally, for intervals lasting days and longer, Normally-off is usually the best choice. Unfortunately, much of our current technology has been built with different objectives in mind. Networked devices that are "always on", gadgets that require humans to recharge them every couple of days, and protocols that have been optimized to maximize throughput rather than conserve resources.

Long battery lifetimes are required for many applications, however. In some cases these lifetimes should be in the order of years or even a decade or longer. Some communication devices already reach multi-year lifetimes, and continuous improvement in low-power electronics and advances in radio technology keep pushing these lifetimes longer. However, it is perhaps fair to say that battery lifetimes are generally too short at present time.

Power usage can not be evaluated solely based on lower layer communications. The entire system, including upper layer protocols and applications is responsible for the power consumption as a whole. The lower communication layers have already adopted many techniques

that can be used to reduce power usage, such as scheduling device wake-up times. Further reductions will likely need some co-operation from the upper layers so that unnecessary communications, denial-of-service attacks on power consumption, and other power drains are eliminated.

Of course, application requirements ultimately determine what kinds of communications are necessary. For instance, some applications require more data to be sent than others. The purpose of the guidelines in this memo is not to prefer one or the other application, but to provide guidance on how to minimize the amount of communications overhead that is not directly required by the application. While such optimization is generally useful, it is relatively speaking most noticeable in applications that transfer only a small amount of data, or operate only infrequently.

3. Link-Layer Assumptions

We assume that the underlying communications network can be any large-scale, public network that employs point-to-point communications model and radio technology. 2G, 3G, and LTE networks are examples of such networks, but not the only possible networks with these characteristics.

In the following we look at some of these characteristics and their implications. Note that in most cases these characteristics are not properties of the specific networks but rather inherent in the concept of public networks.

Public networks

Using a public network service implies that applications can be deployed without having to build a network to go with them. For economic reasons, only the largest users (such as utility companies) could afford to build their own network, and even they would not be able to provide a world-wide coverage. This means that applications where coverage is important can be built. For instance, most transport sector applications require national or even world-wide coverage to work.

But there are other implications, as well. By definition, the network is not tailored for this application and with some exceptions, the traffic passes through the Internet. One implication of this is that there are generally no application-specific network configurations or discovery support. For instance, the public network helps devices to get on the Internet, set up default routers, configure DNS servers, and so on, but does nothing for configuring possible higher-layer functions, such as

servers the device might need to contact to perform its application functions.

Public networks often provide web proxies and other functionality that can in some cases make a significant improvement for delays and cost of communication over the wireless link. For instance, resolving server DNS names in a proxy instead of the user's device may cut down on the general chattiness of the communications, therefore reducing overall delay in completing the entire transaction. Likewise, a CoAP proxy or pub/sub broker [I-D.koster-core-coap-pubsub] can assist a CoAP device in communication. However, unlike HTTP web proxies, CoAP proxies and brokers are not yet widely deployed in public networks.

Similarly, given the lack of available IPv4 addresses, the chances are that many devices are behind a network address translation (NAT) device. This means that they are not easily reachable as servers. Alternatively, the devices may be directly on the global Internet (either on IPv4 or IPv6) and easily reachable as servers. Unfortunately, this may mean that they also receive unwanted traffic, which may have implications for both power consumption and service costs.

Point-to-point link model

This is a common link model in cellular networks. One implication of this model is that there will be no other nodes on the same link, except maybe for the service provider's router. As a result, multicast discovery can not be reasonably used for any local discovery purposes. While the configuration of the service provider's router for specific users is theoretically possible, in practice this is difficult to achieve, at least for any small user that can not afford a network-wide contract for a private APN (Access Point Name). The public network access service has little per-user tailoring.

Radio technology

The use of radio technology means that power is needed to operate the radios. Transmission generally requires more power than reception. However, radio protocols have generally been designed so that a device checks periodically whether it has messages. In a situation where messages arrive seldom or not at all, this checking consumes energy. Research has shown that these periodic checks (such as LTE paging message reception) are often a far bigger contributor to energy consumption than message transmission.

Note that for situations where there are several applications on the same device wishing to communicate with the Internet in some manner, bundling those applications together so that they can communicate at the same time can be very useful. Some guidance for these techniques in the smartphone context can be found in [Android-Bundle].

Naturally, each device has a freedom to decide when it sends messages. In addition, we assume that there is some way for the devices to control when or how often it wants to receive messages. Specific methods for doing this depend on the specific network being used and also tend to change as improvements in the design of these networks are incorporated. The reception control methods generally come in two variants, fine grained mechanisms that deal with how often the device needs to wake-up for paging messages, and more crude mechanisms where the device simply disconnects from the network for a period of time. There are associated costs and benefits to each method, but those are not relevant for this memo, as long as some control method exists. Furthermore, devices could use Delay-Tolerant Networking (DTN) [RFC4838] mechanisms to relax the requirements for timeliness of connectivity and message delivery.

4. Scenarios

Not all applications or situations are equal. They may require different solutions or communication models. This memo focuses on two common scenarios at cellular networks:

Real-Time Reachable Devices

This scenario involves all communication that requires real-time or near real-time communications with a device. That is, a network entity must be able to reach the device with a small time lag at any time, and no pre-agreed wake-up schedule can be arranged. By "real-time" we mean any reasonable end-to-end communications latency, be it measured in milliseconds or seconds. However, unpredictable sleep states are not expected.

Examples of devices in this category include sensors that must be measurable from a remote source at any instant in time, such as process automation sensors and actuators that require immediate action, such as light bulbs or door locks.

Sleepy Devices

This scenario involves freedom to choose when device communicates. The device is often expected to be able to be in a sleep state for

much of its time. The device itself can choose when it communicates, or it lets the network assist in this task.

Examples of devices in this category include sensors that track slowly changing values, such as temperature sensors and actuators that control a relatively slow process, such as heating systems.

Note that there may be hard real-time requirements, but they are expressed in terms of how fast the device can communicate, not in terms of how fast it can respond to a network stimuli. For instance, a fire detector can be classified as a sleepy device as long as it can internally quickly wake up on detecting fire and initiate the necessary communications without delay.

5. Discovery and Registration

In both scenarios the device will be attached to a public network. Without special arrangements, the device will also get a dynamically assigned IP address or an IPv6 prefix. At least one but typically several router hops separate the device from its communicating peers such as application servers. As a result, the address or even the existence of the device is typically not immediately obvious to the other nodes participating in the application. As discussed earlier, multicast discovery has limited value in public networks; network nodes cannot practically discover individual devices in a large public network. And the devices can not discover who they need to talk, as the public network offers just basic Internet connectivity.

Our recommendation is to initiate a discovery and registration process. This allows each device to inform its peers that it has connected to the network and that it is reachable at a given IP address. Registration also facilitates low-power operation since a device can delegate part of the discovery signaling and reachability requirements to another node.

The registration part is easy e.g., with a resource directory. The device should perform the necessary registration with these devices, for instance, as specified in [I-D.ietf-core-resource-directory]. In order to do this registration, the device needs to know its CoRE Link Format description, as specified in [RFC6690]. In essence, the registration process involves performing a GET on `.well-known/core/?rt=core-rd` at the address of the resource directory, and then doing a POST on the path of the discovered resource.

Other mechanisms enabling device discovery and delegation of functionality to a non-sleepy node include [I-D.vial-core-mirror-proxy] and [I-D.koster-core-coap-pubsub].

However, current CoAP specifications provide only limited support for discovering the resource directory or other registration services. Local multicast discovery only works in LAN-type networks, but not in these public cellular networks. Our recommended alternate methods for discovery are the following:

Manual Configuration

The DNS name of the resource directory is manually configured. This approach is suitable in situations where the owner of the devices has the resources and capabilities to do the configuration. For instance, a utility company can typically program its metering devices to point to the company servers.

Manufacturer Server

The DNS name of the directory or proxy is hardwired to the software by the manufacturer, and the directory or proxy is actually run by the manufacturer. This approach is suitable in many consumer usage scenarios, where it would be unreasonable to assume that the consumer runs any specific network services. The manufacturer's web interface and the directory/proxy servers can co-operate to provide the desired functionality to the end user. For instance, the end user can register a device identity in the manufacturer's web interface and ask specific actions to be taken when the device does something.

Delegating Manufacturer Server

The DNS name of the directory or proxy is hardwired to the software by the manufacturer, but this directory or proxy merely redirects the request to a directory or proxy run by the whoever bought the device. This approach is suitable in many enterprise environments, as it allows the enterprise to be in charge of actual data collection and device registries; only the initial bootstrap goes through the manufacturer. In many cases there are even legal requirements (such as EU privacy laws) that prevent providing unnecessary information to third parties.

Common Global Resolution Infrastructure

The delegating manufacturer server model could be generalized into a reverse-DNS -like discovery infrastructure that could answer the question "this is device with identity ID, where is my home registration server?". However, at present no such resolution system exists. (Note: The EPCglobal system for RFID resolution is reminiscent of this approach.)

Besides manual configuration, these alternate mechanisms are mostly suitable for large manufacturers and deployments. Good automated mechanism for discovery of devices that are manufactured and deployed in small quantities are still needed.

6. Data Formats

A variety of data formats exist for passing around data. These data formats include XML, JavaScript Object Notation (JSON) [RFC7159], Efficient XML Interchange (EXI) [W3C.REC-exi-20110310], and text formats. Message lengths can have a significant effect on the amount of energy required for the communications, and such it is highly desirable to keep message lengths minimal. At the same time, extreme optimization can affect flexibility and ease of programming. The authors recommend [I-D.jennings-core-senml] as a compact, yet easily processed and extendable textual format.

7. Real-Time Reachable Devices

These devices are often best modeled as CoAP servers. The device will have limited control on when it receives messages, and it will have to listen actively for messages, up to the limits of the underlying link layer. If the device acts also in client role in some phase of its operation, it can control how many transmissions it makes on its own behalf.

The packet reception checks should be tailored according to the requirements of the application. If sub-second response time is not needed, a more infrequent checking process may save some power.

For sensor-type devices, the CoAP Observe extension [RFC7641] may be supported. This allows the sensor to track changes to the sensed value, and make an immediate observation response upon a change. This may reduce the amount of polling needed to be done by the client. Unfortunately, it does not reduce the time that the device needs to be listening for requests. Subscription requests from other clients than the currently registered one may come at any time, the current client may change its request, and the device still needs to respond to normal queries as a server. As a result, the sensor can not rely having to communicate only on its own choice of observation interval.

In order to act as a server, the device needs to be placed in a public IPv4 address, be reachable over IPv6, or hosted in a private network. If the the device is hosted on a private network, then all other nodes need to access this device also need to reside in the same private network. There are multiple ways to provide private networks over public cellular networks. One approach is to dedicate

a special APN for the private network. Corporate access via cellular networks has often been arranged in this manner, for instance. Another approach is to use Virtual Private Networking (VPN) technology, for instance IPsec-based VPNs.

Power consumption from unwanted traffic is problematic in these devices, unless placed in a private network or protected by a operator-provided firewall service. Devices on an IPv6 network will have some protection through the nature of the 2^{64} address allocation for a single terminal in a 3GPP cellular network; the attackers will be unable to guess the full IP address of the device. However, this protects only the device from processing a packet, but since the network will still deliver the packet to any of the addresses within the assigned 64-bit prefix, packet reception costs are still incurred.

Note that the the VPN approach can not prevent unwanted traffic received at the tunnel endpoint address, and may require keep-alive traffic. Special APNs can solve this issue, but require explicit arrangement with the service provider.

8. Sleepy Devices

These devices are best modeled as devices that can delegate queries to some other node. For instance, as mirror proxy [I-D.vial-core-mirror-proxy] or CoAP Publish-Subscribe [I-D.koster-core-coap-pubsub] clients. When the device initializes itself, it makes a registration of itself in a proxy as described above in Section 5 and then continues to send periodic updates of sensor values.

As a result, the device acts only as a client, not a server, and can shut down all communication channels while it is during its sleeping period. The length of the sleeping period depends on power and application requirements. Some environmental sensors might use a day or a week as the period, while other devices may use a smaller values ranging from minutes to hours.

Other approaches for delegation include CoAP-options described in [I-D.castellani-core-alive] [I-D.fossati-core-publish-monitor-options]. In this memo we use mirror proxies as an example, because of their ability to work with both HTTP and CoAP implementations; but the concepts are similar and the IETF work is still in progress so the final protocol details are yet to be decided.

The ability to shut down communications and act as only a client has four impacts:

- o Radio transmission and reception can be turned off during the sleeping period, reducing power consumption significantly.
- o However, some power and time is consumed by having to re-attach to the network after the end of a sleep period.
- o The window of opportunity for unwanted traffic to arrive is much smaller, as the device is listening for traffic only part of the time. Note that networks may cache packets for some time though. On the other hand, stateful firewalls can effectively remove much of unwanted traffic for client type devices.
- o The device may exist behind a NAT or a firewall without being impacted. Note that "Simple Security" basic IPv6 firewall capability [RFC6092] blocks inbound UDP traffic by default, so just moving to IPv6 is not direct solution to this problem.

For sleepy devices that represent actuators, it is also possible to use the mirror proxy model. The device can make periodic polls to the proxy to determine if a variable has changed.

8.1. Implementation Considerations

There are several challenges in implementing sleepy devices. They need hardware that can be put to an appropriate sleep mode but yet awakened when it is time to do something again. This is not always easy in all hardware platforms. It is important to be able to shut down as much of the hardware as possible, preferably down to everything else except a clock circuit. The platform also needs to support re-awakening at suitable time scales, as otherwise the device needs to be powered up too frequently.

Most commercial cellular modem platforms do not allow applications to suspend the state of the communications stack. Hence, after a power-off period they need to re-establish communications, which takes some amount of time and extra energy.

Implementations should have a coordinated understanding of the state and sleeping schedule. For instance, it makes no sense to keep a CPU powered up, waiting for a message when the lower layer has been told that the next possible paging opportunity is some time away.

The cellular networks have a number of adjustable configuration parameters, such as the maximum used paging interval. Proper setting of these values has an impact on the power consumption of the device, but with the current business practices, such settings are rarely negotiated when the user's subscription is provisioned.

9. Security Considerations

There are no particular security aspects with what has been discussed in this memo, except for the ability to delegate queries for a resource to another node. Depending on how this is done, there are obvious security issues which have largely NOT yet been addressed in the relevant Internet Drafts [I-D.vial-core-mirror-proxy] [I-D.castellani-core-alive] [I-D.fossati-core-publish-monitor-options]. However, we point out that in general, security issues in delegation can be solved either through reliance on your local network support nodes (which may be quite reasonable in many environments) or explicit end-to-end security. Explicit end-to-end security through nodes that are awake at different times means in practice end-to-end data object security. We have implemented one such mechanism for sleepy nodes as described in [I-D.aks-lwig-crypto-sensors].

The security considerations relating to CoAP [RFC7252] and the relevant link layers should apply. Note that cellular networks universally employ per-device authentication, integrity protection, and for most of the world, encryption of all their communications. Additional protection of transport sessions is possible through mechanisms described in [RFC7252] or data objects.

10. IANA Considerations

There are no IANA impacts in this memo.

11. References

11.1. Normative References

- [RFC7159] Bray, T., Ed., "The JavaScript Object Notation (JSON) Data Interchange Format", RFC 7159, DOI 10.17487/RFC7159, March 2014, <<http://www.rfc-editor.org/info/rfc7159>>.
- [RFC6690] Shelby, Z., "Constrained RESTful Environments (CoRE) Link Format", RFC 6690, DOI 10.17487/RFC6690, August 2012, <<http://www.rfc-editor.org/info/rfc6690>>.
- [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>>.

- [RFC7641] Hartke, K., "Observing Resources in the Constrained Application Protocol (CoAP)", RFC 7641, DOI 10.17487/RFC7641, September 2015, <<http://www.rfc-editor.org/info/rfc7641>>.
- [I-D.ietf-core-resource-directory] Shelby, Z., Kostner, M., Bormann, C., and P. Stok, "CoRE Resource Directory", draft-ietf-core-resource-directory-05 (work in progress), October 2015.
- [W3C.REC-exi-20110310] Kamiya, T. and J. Schneider, "Efficient XML Interchange (EXI) Format 1.0", World Wide Web Consortium Recommendation REC-exi-20110310 <http://www.w3.org/TR/2011/REC-exi-20110310>, March 2011.
- [I-D.jennings-core-senml] Jennings, C., Shelby, Z., Arkko, J., and A. Keranen, "Media Types for Sensor Markup Language (SENML)", draft-jennings-core-senml-02 (work in progress), October 2015.
- [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>>.

11.2. Informative References

- [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, <<http://www.rfc-editor.org/info/rfc4838>>.
- [RFC6092] Woodyatt, J., Ed., "Recommended Simple Security Capabilities in Customer Premises Equipment (CPE) for Providing Residential IPv6 Internet Service", RFC 6092, DOI 10.17487/RFC6092, January 2011, <<http://www.rfc-editor.org/info/rfc6092>>.
- [I-D.arkko-core-sleepy-sensors] Arkko, J., Rissanen, H., Loreto, S., Turanyi, Z., and O. Novo, "Implementing Tiny CoAP Sensors", draft-arkko-core-sleepy-sensors-01 (work in progress), July 2011.

[I-D.aks-lwig-crypto-sensors]

Sethi, M., Arkko, J., Keranen, A., and H. Back, "Practical Considerations and Implementation Experiences in Securing Smart Object Networks", draft-aks-lwig-crypto-sensors-00 (work in progress), October 2015.

[I-D.castellani-core-alive]

Castellani, A. and S. Loreto, "CoAP Alive Message", draft-castellani-core-alive-00 (work in progress), March 2012.

[I-D.fossati-core-publish-monitor-options]

Fossati, T., Giacomini, P., and S. Loreto, "Publish and Monitor Options for CoAP", draft-fossati-core-publish-monitor-options-01 (work in progress), March 2012.

[I-D.vial-core-mirror-proxy]

Vial, M., "CoRE Mirror Server", draft-vial-core-mirror-proxy-01 (work in progress), July 2012.

[I-D.koster-core-coap-pubsub]

Koster, M., Keranen, A., and J. Jimenez, "Publish-Subscribe Broker for the Constrained Application Protocol (CoAP)", draft-koster-core-coap-pubsub-04 (work in progress), November 2015.

[Android-Bundle]

"Optimizing Downloads for Efficient Network Access",
Android developer note
<http://developer.android.com/training/efficient-downloads/efficient-network-access.html>, February 2013.

Appendix A. Acknowledgments

The authors would like to thank Zach Shelby, Jan Holler, Salvatore Loreto, Matthew Vial, Thomas Fossati, Mohit Sethi, Jan Melen, Joachim Sachs, Heidi-Maria Rissanen, Sebastien Pierrel, Kumar Balachandran, Muhammad Waqas Mir, Cullen Jennings, Markus Isomaki, Hannes Tschofenig, and Anna Larmo for interesting discussions in this problem space.

Authors' Addresses

Jari Arkko
Ericsson
Jorvas 02420
Finland

Email: jari.arkko@piuha.net

Anders Eriksson
Ericsson
Stockholm 164 83
Sweden

Email: anders.e.eriksson@ericsson.com

Ari Keranen
Ericsson
Jorvas 02420
Finland

Email: ari.keranen@ericsson.com

Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: April 24, 2018

C. Gomez
Universitat Politecnica de Catalunya
M. Kovatsch
ETH Zurich
H. Tian
China Academy of Telecommunication Research
Z. Cao, Ed.
Huawei Technologies
October 21, 2017

Energy-Efficient Features of Internet of Things Protocols
draft-ietf-lwig-energy-efficient-08

Abstract

This document describes the challenges for energy-efficient protocol operation on constrained devices and the current practices used to overcome those challenges. It summarizes the main link-layer techniques used for energy-efficient networking, and it highlights the impact of such techniques on the upper layer protocols so that they can together achieve an energy efficient behavior. The document also provides an overview of energy-efficient mechanisms available at each layer of the IETF protocol suite specified 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 April 24, 2018.

Copyright Notice

Copyright (c) 2017 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

| | |
|--|----|
| 1. Introduction | 2 |
| 1.1. Terminology | 3 |
| 2. Overview | 3 |
| 3. Medium Access Control and Radio Duty Cycling | 5 |
| 3.1. Radio Duty Cycling techniques | 6 |
| 3.2. Latency and buffering | 7 |
| 3.3. Throughput | 7 |
| 3.4. Radio interface tuning | 8 |
| 3.5. Packet bundling | 8 |
| 3.6. Power save services available in example low-power radios | 8 |
| 3.6.1. Power Save Services Provided by IEEE 802.11 | 8 |
| 3.6.2. Power Save Services Provided by Bluetooth LE | 9 |
| 3.6.3. Power Save Services in IEEE 802.15.4 | 10 |
| 3.6.4. Power Save Services in DECT ULE | 12 |
| 4. IP Adaptation and Transport Layer | 14 |
| 5. Routing Protocols | 15 |
| 6. Application Layer | 16 |
| 6.1. Energy efficient features in CoAP | 16 |
| 6.2. Sleepy node support | 16 |
| 6.3. CoAP timers | 17 |
| 6.4. Data compression | 17 |
| 7. Summary and Conclusions | 18 |
| 8. Contributors | 18 |
| 9. Acknowledgments | 18 |
| 10. IANA Considerations | 19 |
| 11. Security Considerations | 19 |
| 12. References | 19 |
| 12.1. Normative References | 19 |
| 12.2. Informative References | 21 |
| Authors' Addresses | 23 |

1. Introduction

Network systems for physical world monitoring contain many battery-powered or energy-harvesting devices. For example, in an environmental monitoring system, or a temperature and humidity

monitoring system, there may not be always-on and sustained power supplies for the potentially large number of constrained devices. In such deployment scenarios, it is necessary to optimize the energy consumption of the constrained devices. In this document we describe techniques that are in common use at Layer 2 and at Layer 3, and we indicate the need for higher-layer awareness of lower-layer features.

Many research efforts have studied this "energy efficiency" problem. Most of this research has focused on how to optimize the system's power consumption in certain deployment scenarios, or how an existing network function such as routing or security could be more energy-efficient. Only few efforts have focused on energy-efficient designs for IETF protocols and standardized network stacks for such constrained devices [I-D.kovatsch-lwig-class1-coap].

The IETF has developed a suite of Internet protocols suitable for such constrained devices, including IPv6 over Low-Power Wireless Personal Area Networks (6LoWPAN) [RFC6282],[RFC6775],[RFC4944], the IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL) [RFC6550], and the Constrained Application Protocol (CoAP) [RFC7252]. This document tries to summarize the design considerations for making the IETF constrained protocol suite as energy-efficient as possible. While this document does not provide detailed and systematic solutions to the energy efficiency problem, it summarizes the design efforts and analyzes the design space of this problem. In particular, it provides an overview of the techniques used by the lower layers to save energy and how these may impact on the upper layers. Cross-layer interaction is therefore considered in this document from this specific point of view. Providing further design recommendations that go beyond the layered protocol architecture is out of the scope of this document.

After reviewing the energy-efficient designs of each layer, we summarize the document by presenting some overall conclusions. Though the lower layer communication optimization is the key part of energy efficient design, the protocol design at the upper layers is also important to make the device energy-efficient.

1.1. Terminology

Terms used in this document are defined in [RFC7228] [I-D.bormann-lwig-7228bis].

2. Overview

The IETF has developed protocols to enable end-to-end IP communication between constrained nodes and fully capable nodes. This work has expedited the evolution of the traditional Internet

protocol stack to a light-weight Internet protocol stack. As shown in Figure 1 below, the IETF has developed CoAP as the application layer and 6LoWPAN as the adaption layer to run IPv6 over IEEE 802.15.4 and Bluetooth Low-Energy, with the support of routing by RPL and efficient neighbor discovery by 6LoWPAN-ND. 6LoWPAN is currently being adapted by the 6lo working group to support IPv6 over various other technologies, such as ITU-T G.9959 [G9959], DECT ULE [TS102], MS/TP-BACnet [MSTP], and Near Field Communication (NFC) [NFC].

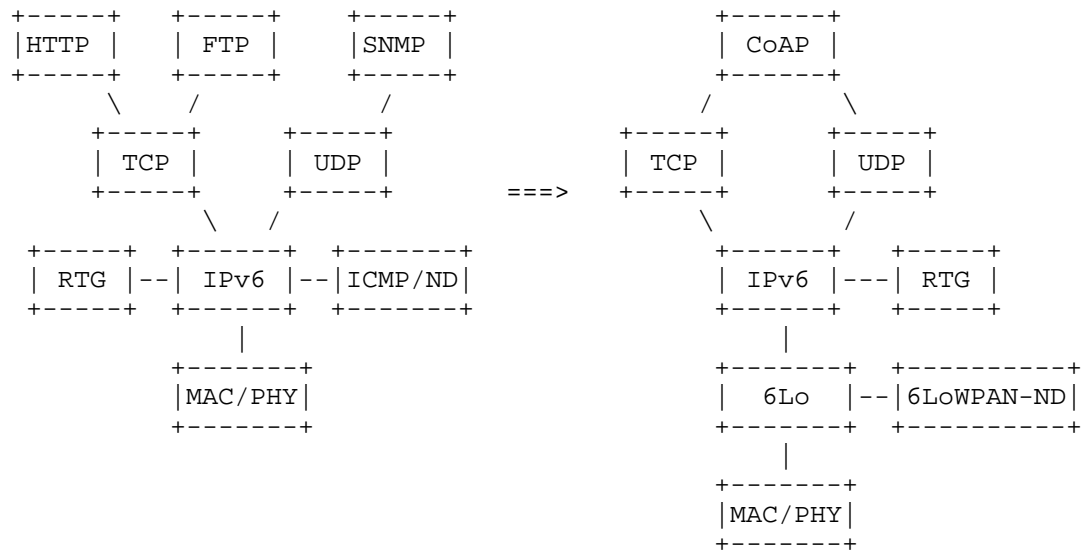


Figure 1: Traditional and Light-weight Internet Protocol Stack

There are numerous published studies reporting comprehensive measurements of wireless communication platforms [Powertrace]. As an example, below we list the energy consumption profile of the most common operations involved in communication on a prevalent sensor node platform. The measurement was based on the Tmote Sky with ContikiMAC [ContikiMAC] as the radio duty cycling algorithm. From this and many other measurement reports (e.g.[AN079]), we can see that the energy consumption of optimized transmission and reception are in the same order. For IEEE 802.15.4 and Ultra WideBand (UWB) links, transmitting may actually be even cheaper than receiving. It also shows that broadcast and non-synchronized communication transmissions are energy costly because they need to acquire the medium for a long time.

| Activity | Energy (uJ) |
|---------------------------------------|-------------|
| Broadcast reception | 178 |
| Unicast reception | 222 |
| Broadcast transmission | 1790 |
| Non-synchronized unicast transmission | 1090 |
| Synchronized unicast transmission | 120 |
| Unicast TX to awake receiver | 96 |
| Listening (for 1000 ms) | 63000 |

Figure 2: Power consumption of common operations involved in communication on the Tmote Sky with ContikiMAC

At the Physical layer, one approach that may allow reducing energy consumption of a device that uses a wireless interface is based on reducing the device transmit power level as long as the intended next hop(s) are still within range of the device. In some cases, if node A has to transmit a message to node B, a solution to reduce node A transmit power is to leverage an intermediate device, e.g. node C as a message forwarder. Let d be the distance between node A and node B. Assuming free-space propagation, where path loss is proportional to d^2 , if node C is placed right in the middle of the path between A and B (that is, at a distance $d/2$ from both node A and node B), the minimum transmit power to be used by node A (and by node C) is reduced by a factor of 4. However, this solution requires additional devices, it requires a routing solution, and it also increases transmission delay between A and B.

3. Medium Access Control and Radio Duty Cycling

In networks, communication and power consumption are interdependent. The communication device is typically the most power-consuming component, but merely refraining from transmissions is not enough to achieve a low power consumption: the radio may consume as much power in listen mode as when actively transmitting. This illustrates the key problem known as idle listening, whereby the radio of a device may be in receive mode (ready to receive any message), even if no message is being transmitted to that device. Idle listening can consume a huge amount of energy unnecessarily. To reduce power consumption, the radio must be switched completely off -- duty-cycled

-- as much as possible. By applying duty-cycling, the lifetime of a device operating on a common button battery may be on the order of years, whereas otherwise the battery may be exhausted in a few days or even hours. Duty-cycling is a technique generally employed by devices that use the P1 strategy [RFC7228], which need to be able to communicate on a relatively frequent basis. Note that a more aggressive approach to save energy relies on the P0, Normally-off strategy, whereby devices sleep for very long periods and communicate infrequently, even though they spend energy in network reattachment procedures.

From the perspective of Medium Access Control (MAC) and Radio Duty Cycling (RDC), all upper layer protocols, such as routing, RESTful communication, adaptation, and management flows, are applications. Since the duty cycling algorithm is the key to energy-efficiency of the wireless medium, it synchronizes transmission and/or reception requests from the higher layers.

MAC and RDC are not in the scope of the IETF, yet lower layer designers and chipset manufacturers take great care to save energy. By knowing the behaviors of these lower layers, IETF engineers can design protocols that work well with them. The IETF protocols to be discussed in the following sections are the customers of the lower layers.

3.1. Radio Duty Cycling techniques

This subsection describes three main three RDC techniques. Note that more than one of these techniques may be available or can even be combined in a specific radio technology:

a) Channel sampling. In this solution, the radio interface of a device periodically monitors the channel for very short time intervals (i.e. with a low duty cycle) with the aim of detecting incoming transmissions. In order to make sure that a receiver can correctly receive a transmitted data unit, the sender may prepend a preamble of a duration at least the sampling period to the data unit to be sent. Another option for the sender is to repeatedly transmit the data unit, instead of sending a preamble before the data unit. Once a transmission is detected by a receiver, the receiver may stay awake until the complete reception of the data unit. Examples of radio technologies that use preamble sampling include ContikiMAC, the Coordinated Sampled Listening (CSL) mode of IEEE 802.15.4e, and the Frequently Listening (FL) mode of ITU-T G.9959 [G9959].

b) Scheduled transmissions. This approach allows a device to know the particular time at which it should be awake (during some time interval) in order to receive data. Otherwise, the device may remain

in sleep mode. The decision on the times at which communication is attempted relies on some form of negotiation between the involved devices. Such negotiation may be performed per transmission or per session/connection. Bluetooth Low Energy (Bluetooth LE) is an example of a radio technology based on this mechanism.

c) Listen after send. This technique allows a node to remain in sleep mode by default, wake up and poll a sender (which must be ready to receive a poll message) for pending transmissions. After sending the poll message, the node remains in receive mode, ready for a potential incoming transmission. After a certain time interval, the node may go back to sleep. For example, the Receiver Initiated Transmission (RIT) mode of 802.15.4e, and the transmission of data between a coordinator and a device in IEEE 802.15.4-2003 use this technique.

3.2. Latency and buffering

The latency of a data unit transmission to a duty-cycled device is equal to or greater than the latency of transmitting to an always-on device. Therefore, duty-cycling leads to a trade-off between energy consumption and latency. Note that in addition to a latency increase, RDC may introduce latency variance, since the latency increase is a random variable (which is uniformly distributed if duty-cycling follows a periodical behavior).

On the other hand, due to the latency increase of duty-cycling, a sender waiting for a transmission opportunity may need to store subsequent outgoing packets in a buffer, increasing memory requirements and potentially incurring queuing waiting time that contributes to the packet's overall delay and increases the probability of buffer overflow, leading to losses.

3.3. Throughput

Although throughput is not typically a key concern in constrained node network applications, it is indeed important in some services in such networks, such as over-the-air software updates or when off-line sensors accumulate measurements that have to be quickly transferred when there is an opportunity for connectivity.

Since RDC introduces inactive intervals in energy-constrained devices, it reduces the throughput that can be achieved when communicating with such devices. There exists a trade-off between the achievable throughput and energy consumption.

3.4. Radio interface tuning

The parameters controlling the radio duty cycle have to be carefully tuned to achieve the intended application and/or network requirements. On the other hand, upper layers should take into account the expected latency and/or throughput behavior due to RDC. The next subsection provides details on key parameters controlling RDC mechanisms, and thus fundamental trade-offs, for various examples of relevant low-power radio technologies.

3.5. Packet bundling

Another technique that may be useful to increase communication energy efficiency is packet bundling. This technique, which is available in several radio interfaces (e.g. LTE and some 802.11 variants), allows to aggregate several small packets into a single large packet. Header and communication overhead is therefore reduced.

3.6. Power save services available in example low-power radios

This subsection presents power save services and techniques used in a few relevant examples of wireless low-power radios: IEEE 802.11, Bluetooth LE and IEEE 802.15.4. For a more detailed overview of each technology, the reader may refer to the literature or to the corresponding specifications.

3.6.1. Power Save Services Provided by IEEE 802.11

IEEE 802.11 defines the Power Save Mode (PSM) whereby a station may indicate to an Access Point (AP) that it will enter a sleep mode state. While the station is sleeping, the AP buffers any frames that should be sent to the sleeping station. The station wakes up every Listen Interval (which can be a multiple of the Beacon Interval) in order to receive beacons. The AP signals in the beacon whether there is data pending for the station or not. If there are not frames to be sent to the station, the latter may get back to sleep mode. Otherwise, the station may send a message requesting the transmission of the buffered data and stay awake in receive mode.

IEEE 802.11v [IEEE80211v] further defines mechanisms and services for power save of stations/nodes that include flexible multicast service (FMS), proxy ARP advertisement, extended sleep modes, and traffic filtering. Upper layer protocols knowledge of such capabilities provided by the lower layer enables better interworking.

These services include:

Proxy ARP: The Proxy ARP capability enables an Access Point (AP) to indicate that the non-AP station (STA) will not receive ARP frames. The Proxy ARP capability enables the non-AP STA to remain in power-save for longer periods of time.

Basic Service Set (BSS) Max Idle Period management enables an AP to indicate a time period during which the AP does not disassociate a STA due to non-receipt of frames from the STA. This supports improved STA power saving and AP resource management.

FMS: A service in which a non-access point (non-AP) STA can request a multicast delivery interval longer than the delivery traffic indication message (DTIM) interval for the purposes of lengthening the period of time a STA may be in a power save state.

Traffic Filtering Service (TFS): A service provided by an access point (AP) to a non-AP STA that can reduce the number of frames sent to the STA by dropping individually addressed frames that do not match traffic filters specified by the STA.

Using the above services provided by the lower layer, the constrained nodes can achieve either client initiated power save (via TFS) or network assisted power save (Proxy-ARP, BSS Max Idle Period and FMS).

Upper layer protocols should synchronize with the parameters such as FMS interval and BSS MAX Idle Period, so that the wireless transmissions are not triggered periodically.

3.6.2. Power Save Services Provided by Bluetooth LE

Bluetooth LE is a wireless low-power communications technology that is the hallmark component of the Bluetooth 4.0, 4.1, and 4.2 specifications [Bluetooth42]. BT-LE has been designed for the goal of ultra-low-power consumption. IPv6 can be run over Bluetooth LE networks by using a 6LoWPAN variant adapted to BT-LE [RFC7668].

Bluetooth LE networks comprise a master and one or more slaves which are connected to the master. The Bluetooth LE master is assumed to be a relatively powerful device, whereas a slave is typically a constrained device (e.g. a class 1 device).

Medium access in Bluetooth LE is based on a Time Division Multiple Access (TDMA) scheme which is coordinated by the master. This device determines the start of connection events, in which communication between the master and a slave takes place. At the beginning of a connection event, the master sends a poll message, which may encapsulate data, to the slave. The latter must send a response, which may also contain data. The master and the slave may continue

exchanging data until the end of the connection event. The next opportunity for communication between the master and the slave will be in the next connection event scheduled for the slave.

The time between consecutive connection events is defined by the `connInterval` parameter, which may range between 7.5 ms and 4 s. The slave may remain in sleep mode since the end of its last connection event until the beginning of its next connection event. Therefore, Bluetooth LE is duty-cycled by design. Furthermore, after having replied to the master, a slave is not required to listen to the master (and thus may keep the radio in sleep mode) for `connSlaveLatency` consecutive connection events. `connSlaveLatency` is an integer parameter between 0 and 499 which should not cause link inactivity for more than `connSupervisionTimeout` time. The `connSupervisionTimeout` parameter is in the range between 100 ms and 32 s.

Upper layer protocols should take into account the medium access and duty-cycling behavior of Bluetooth LE. In particular, `connInterval`, `connSlaveLatency` and `connSupervisionTimeout` determine the time between two consecutive connection events for a given slave. The upper layer packet generation pattern and rate should be consistent with the settings of the aforementioned parameters (and vice versa). For example, assume `connInterval`=4 seconds, `connSlaveLatency`=7 and `connSupervisionTimeout`=32 seconds. With these settings, communication opportunities between a master and a slave will occur during a given interval every 32 seconds. Duration of the interval will depend on several factors, including number of connected slaves, amount of data to be transmitted, etc. In the worst case, only one data unit can be sent from master to slave and vice versa every 32 seconds.

3.6.3. Power Save Services in IEEE 802.15.4

IEEE 802.15.4 is a family of standard radio interfaces for low-rate, low-power wireless networking [fifteendotfour]. Since the publication of its first version in 2003, IEEE 802.15.4 has become the de-facto choice for a wide range of constrained node network application domains and has been a primary target technology of various IETF working groups such as 6LoWPAN [RFC6282], [RFC6775], [RFC4944] and 6TiSCH [I-D.ietf-6tisch-architecture]. IEEE 802.15.4 specifies a variety of related PHY and MAC layer functionalities.

IEEE 802.15.4 defines three roles called device, coordinator and Personal Area Network (PAN) coordinator. The device role is adequate for nodes that do not implement the complete IEEE 802.15.4 functionality, and is mainly targeted for constrained nodes with a limited energy source. The coordinator role includes synchronization

capabilities and is suitable for nodes that do not suffer severe constraints (e.g. a mains-powered node). The PAN coordinator is a special type of coordinator that acts as a principal controller in an IEEE 802.15.4 network.

IEEE 802.15.4 defines two main types of networks depending on their configuration: beacon-enabled and nonbeacon-enabled networks. In the first network type, coordinators periodically transmit beacons. The time between beacons is divided in three main parts: the Contention Access Period (CAP), the Contention Free Period (CFP) and an inactive period. In the first period, nodes use slotted Carrier Sense Multiple Access / Collision Avoidance (CSMA/CA) for data communication. In the second one, a TDMA scheme controls medium access. During the idle period, communication does not take place, thus the inactive period is a good opportunity for nodes to turn the radio off and save energy. The coordinator announces in each beacon the list of nodes for which data will be sent in the subsequent period. Therefore, devices may remain in sleep mode by default and wake up periodically to listen to the beacons sent by their coordinator. If a device wants to transmit data, or learns from a beacon that it is an intended destination, then it will exchange messages with the coordinator (and thus consume energy). An underlying assumption is that when a message is sent to a coordinator, the radio of the coordinator will be ready to receive the message.

The beacon interval and the duration of the beacon interval active portion (i.e. the CAP and the CFP), and thus the duty cycle, can be configured. The parameters that control these times are called `macBeaconOrder` and `macSuperframeOrder`, respectively. As an example, when IEEE 802.15.4 operates in the 2.4 GHz PHY, both times can be (independently) set to values in the range between 15.36 ms and 251.6 seconds.

In the beaconless mode, nodes use unslotted CSMA/CA for data transmission. The device may be in sleep mode by default and may activate its radio to either i) request to the coordinator whether there is pending data for the device, or ii) to transmit data to the coordinator. The wake-up pattern of the device, if any, is out of the scope of IEEE 802.15.4.

Communication between the two ends of an IEEE 802.15.4 link may also take place in a peer-to-peer configuration, whereby both link ends assume the same role. In this case, data transmission can happen at any moment. Nodes must have their radio in receive mode, and be ready to listen to the medium by default (which for battery-enabled nodes may lead to a quick battery depletion), or apply

synchronization techniques. The latter are out of the scope of IEEE 802.15.4.

The main MAC layer IEEE 802.15.4 amendment to date is IEEE 802.15.4e. This amendment includes various new MAC layer modes, some of which include mechanisms for low energy consumption. Among these, the Time-Slotted Channel Hopping (TSCH) is an outstanding mode which offers robust features for industrial environments, among others. In order to provide the functionality needed to enable IPv6 over TSCH, the 6TiSCH working group was created. TSCH is based on a TDMA schedule whereby a set of time slots are used for frame transmission and reception, and other time slots are unscheduled. The latter time slots may be used by a dynamic scheduling mechanism, otherwise nodes may keep the radio off during the unscheduled time slots, thus saving energy. The minimal schedule configuration specified in [I-D.ietf-6tisch-minimal] comprises 101 time slots; 95 of these time slots are unscheduled and the time slot duration is 15 ms.

The previously mentioned CSL and RIT are also 802.15.4e modes designed for low energy.

3.6.4. Power Save Services in DECT ULE

DECT Ultra Low Energy (DECT ULE) is a wireless technology building on the key fundamentals of traditional DECT / CAT-iq [EN300] but with specific changes to significantly reduce the power consumption at the expense of data throughput [TS102]. DECT ULE devices typically operate on special power optimized silicon, but can connect to a DECT Gateway supporting traditional DECT / CAT-iq for cordless telephony and data as well as the DECT ULE extensions. IPv6 can be run over DECT ULE by using a 6LoWPAN variant [I-D.ietf-6lo-dect-ule].

DECT defines two major roles: the Portable Part (PP) is the power constrained device, while the Fixed Part (FP) is the Gateway or base station in a star topology. DECT operates in license free and reserved frequency bands based on TDMA/FDMA and TDD using dynamic channel allocation for interference avoidance. It provides good indoor (~50 m) and outdoor (~300 m) coverage. It uses a frame length of 10 ms divided into 24 timeslots, and it supports connection oriented, packet data and connection-less services.

The FP usually transmits a so-called dummy bearer (beacon) that is used to broadcast synchronization, system and paging information. The slot/carrier position of this dummy bearer can automatically be reallocated in order to avoid mutual interference with other DECT signals.

At the MAC level DECT ULE communications between FP and PP are initiated by the PP. A FP can initiate communication indirectly by sending paging signal to a PP. The PP determines the timeslot and frequency on which the communication between FP and PP takes place. The PP verifies the radio timeslot/frequency position is unoccupied before it initiates its transmitter. An access-request message, which usually carries data, is sent to the FP. The FP sends a confirm message, which also may carry data. More data can be sent in subsequent frames. A MAC level automatic retransmission scheme significantly improves data transfer reliability. A segmentation and reassembly scheme supports transfer of larger higher layer SDUs and provides data integrity check. The DECT ULE packet data service ensures data integrity, proper sequencing, duplicate protection, but not guaranteed delivery. Higher layers protocols have to take this into consideration.

The FP may send paging information to PPs to trigger connection setup and indicate the required service type. The interval between paging information to a specific PP can be defined in range 10 ms to 327 seconds. The PP may enter sleep mode to save power. The listening interval is defined by the PP application. For short sleep intervals (below ~10 seconds) the PP may be able to retain synchronization to the FP dummy bearer and only turn on the receiver during the expected timeslot. For longer sleep intervals the PP can't keep synchronization and has to search for and resynchronize to the FP dummybearer. Hence, longer sleep interval reduces the average energy consumption, but adds a energy consumption penalty for acquiring synchronization to the FP dummy bearer. The PP can obtain all information to determine paging and acquire synchronization information in a single reception of one full timeslot.

Packet data latency is normally 30 ms for short packets (below or equal to 32 octets), however if retry and back-off scenarios occur, the latency is increased. The latency can actually be reduced to about 10 ms by doing energy consuming RSSI scanning in advance. In the direction from FP to PP the latency is usually increased by the used paging interval and the sleep interval. The MAC layer can piggyback commands to improve efficiency (reduce latency) of higher layer protocols. Such commands can instruct the PP to initiate a new packet transfer in N frames without the need for resynchronization and listening to paging or instruct the PP to stay in a higher duty cycle paging detection mode.

The DECT ULE technology allows per PP configuration of paging interval, MTU size, reassembly window size and higher layer service negotiation and protocol.

4. IP Adaptation and Transport Layer

6LoWPAN provides an adaptation layer designed to support IPv6 over IEEE 802.15.4. 6LoWPAN affects the energy-efficiency problem in three aspects, as follows.

First, 6LoWPAN provides one fragmentation and reassembly mechanism which is aimed at solving the packet size issue in IPv6 and could also affect energy-efficiency. IPv6 requires that every link in the internet have an MTU of 1280 octets or greater. On any link that cannot convey a 1280-octet packet in one piece, link-specific fragmentation and reassembly must be provided at a layer below IPv6 [RFC2460]. 6LoWPAN provides fragmentation and reassembly below the IP layer to solve the problem. One of the benefits from placing fragmentation at a lower layer such as the 6LoWPAN layer is that it can avoid the presence of more IP headers, because fragmentation at the IP layer will produce more IP packets, each one carrying its own IP header. However, performance can be severely affected if, after IP layer fragmentation, then 6LoWPAN fragmentation happens as well (e.g. when the upper layer is not aware of the existence of the fragmentation at the 6LoWPAN layer). One solution is to require higher layers awareness of lower layer features and generate small enough packets to avoid fragmentation. In this regard, the Block option in CoAP can be useful when CoAP is used at the application layer [RFC 7959].

Secondly, 6LoWPAN swaps computing with communication. 6LoWPAN applies compression of the IPv6 header. Subject to the packet size limit of IEEE 802.15.4, 40 octets long IPv6 header and 8 octets or 20 octets long UDP and TCP header will consume even more packet space than the data itself. 6LoWPAN provides IPv6 and UDP header compression at the adaptation layer. Therefore, a lower amount of data will be handled by the lower layers, whereas both the sender and receiver will spend more computing power on the compression and decompression of the packets over the air. Compression can also be performed at higher layers (see Section 6.4).

Finally, the 6LoWPAN working group developed the energy-efficient Neighbor Discovery called 6LoWPAN-ND, which is an energy efficient replacement of the IPv6 ND in constrained environments. IPv6 Neighbor Discovery was not designed for non-transitive wireless links, as its heavy use of multicast makes it inefficient and sometimes impractical in a low-power and lossy network. 6LoWPAN-ND describes simple optimizations to IPv6 Neighbor Discovery, its addressing mechanisms, and duplicate address detection for Low-power Wireless Personal Area Networks and similar networks. However, 6LoWPAN ND does not modify Neighbor Unreachability Detection (NUD) timeouts, which are very short (by default three transmissions spaced

one second apart). NUD timeout settings should be tuned taking into account the latency that may be introduced by duty-cycled mechanisms at the link layer, or alternative, less impatient NUD algorithms should be considered [I-D.ietf-6man-impatient-nud].

IPv6 underlies the higher layer protocols, including both TCP/UDP transport and applications. By design, the higher-layer protocols do not typically have specific information about the lower layers, and thus cannot solve the energy-efficiency problem.

The network stack can be designed to save computing power. For example the Contiki implementation has multiple cross layer optimizations for buffers and energy management, e.g., the computing and validation of UDP/TCP checksums without the need of reading IP headers from a different layer. These optimizations are software implementation techniques, and out of the scope of IETF and the LWIG working group.

5. Routing Protocols

RPL [RFC6550] is a routing protocol designed by the IETF for constrained environments. RPL exchanges messages periodically and keeps routing states for each destination. RPL is optimized for the many-to-one communication pattern, where network nodes primarily send data towards the border router, but has provisions for any-to-any routing as well.

The authors of the Powertrace tool [Powertrace] studied the power profile of RPL. Their analysis divides the routing protocol into control and data traffic. The control plane carries ICMP messages to establish and maintain the routing states. The data plane carries any application that uses RPL for routing packets. The study has shown that the power consumption of the control traffic goes down over time in a relatively stable network. The study also reflects that the routing protocol should keep the control traffic as low as possible to make it energy-friendly. The amount of RPL control traffic can be tuned by setting the Trickle [RFC6206] algorithm parameters (i.e. I_{min} , I_{max} and k) to appropriate values. However, there exists a trade-off between energy consumption and other performance parameters such as network convergence time and robustness.

RFC 6551 [RFC6551] defines routing metrics and constraints to be used by RPL in route computation. Among others, RFC 6551 specifies a Node Energy object that allows to provide information related to node energy, such as the energy source type or the estimated percentage of remaining energy. Appropriate use of energy-based routing metrics

may help to balance energy consumption of network nodes, minimize network partitioning and increase network lifetime.

6. Application Layer

6.1. Energy efficient features in CoAP

CoAP [RFC7252] is designed as a RESTful application protocol, connecting the services of smart devices to the World Wide Web. CoAP is not a chatty protocol. It provides basic communication services such as service discovery and GET/POST/PUT/DELETE methods with a binary header.

Energy efficiency is part of the CoAP protocol design. CoAP uses a fixed-length binary header of only four bytes that may be followed by binary options. To reduce regular and frequent queries of the resources, CoAP provides an observe mode, in which the requester registers its interest of a certain resource and the responder will report the value whenever it was updated. This reduces the request response round trips while keeping information exchange a ubiquitous service; an energy-constrained server can remain in sleep mode during the period between observe notification transmissions.

Furthermore, [RFC7252] defines CoAP proxies which can cache resource representations previously provided by sleepy CoAP servers. The proxies themselves may respond to client requests if the corresponding server is sleeping and the resource representation is recent enough. Otherwise, a proxy may attempt to obtain the resource from the sleepy server.

CoAP proxy and cache functionality may also be used to perform data aggregation. This technique allows a node to receive data messages (e.g. carrying sensor readings) from other nodes in the network, perform an operation based on the content in those messages, and transmit the result of the operation. Such operation may simply be intended to use one packet to carry the readings transported in several packets (which reduces header and transmission overhead), or it may be a more sophisticated operation, possibly based on mathematical, logical or filtering principles (which reduces the payload size to be transmitted).

6.2. Sleepy node support

Beyond these features of CoAP, there have been a number of proposals to further support sleepy nodes at the application layer by leveraging CoAP mechanisms. A good summary of such proposals can be found in [I-D.rahman-core-sleepy-nodes-do-we-need], while an example application (in the context of illustrating several security

mechanisms) in a scenario with sleepy devices has been described [I-D.ietf-lwig-crypto-sensors]. Approaches to support sleepy nodes include exploiting the use of proxies, leveraging the Resource Directory [I-D.ietf-core-resource-directory] or signaling when a node is awake to the interested nodes. Recent work defines publish-subscribe and message queuing extensions to CoAP and the Resource Directory in order to support devices that spend most of their time in asleep [I-D.ietf-core-coap-pubsub]. Notably, this work has been adopted by the CoRE Working Group.

In addition to the work within the scope of CoAP to support sleepy nodes, other specifications define application layer functionality for the same purpose. The Lightweight Machine-to-Machine (LWM2M) specification from the Open Mobile Alliance (OMA) defines a Queue Mode whereby an LWM2M Server queues requests to an LWM2M Client until the latter (which may often stay in sleep mode) is online. LWM2M functionality operates on top of CoAP.

oneM2M defines a CoAP binding with an application layer mechanism for sleepy nodes [oneM2M].

6.3. CoAP timers

CoAP offers mechanisms for reliable communication between two CoAP endpoints. A CoAP message may be signaled as a confirmable (CON) message, and an acknowledgment (ACK) is issued by the receiver if the CON message is correctly received. The sender starts a Retransmission TimeOut (RTO) for every CON message sent. The initial RTO value is chosen randomly between 2 and 3 s. If an RTO expires, the new RTO value is doubled (unless a limit on the number of retransmissions has been reached). Since duty-cycling at the link layer may lead to long latency (i.e. even greater than the initial RTO value), CoAP RTO parameters should be tuned accordingly in order to avoid spurious RTOs which would unnecessarily waste node energy and other resources. On the other hand, note that CoAP can also run on top of TCP [I-D.ietf-core-coap-tcp-tls]. In that case, similar guidance applies to TCP timers, albeit with greater motivation to carefully configure TCP RTO parameters, since [RFC6298] reduced the default initial TCP RTO to 1 second, which may interact more negatively with duty-cycled links than default CoAP RTO values.

6.4. Data compression

Another method intended to reduce the size of the data units to be communicated in constrained-node networks is data compression, which allows to encode data using less bits than the original data representation. Data compression is more efficient at higher layers, particularly before encryption is used. In fact, encryption

mechanisms may generate an output that does not contain redundancy, making it almost impossible to reduce the data representation size. In CoAP, messages may be encrypted by using DTLS (or TLS when CoAP over TCP is used), which is the default mechanism for securing CoAP exchanges.

7. Summary and Conclusions

We summarize the key takeaways in this document:

- a. Internet protocols designed by IETF can be considered as the customer of the lower layers (PHY, MAC, and Duty-cycling). To reduce power consumption, it is recommended that Layer 3 designs should operate based on awareness of lower-level parameters rather than treating the lower layer as a black box (Sections 4, 5 and 6).
- b. It is always useful to compress the protocol headers in order to reduce the transmission/reception power. This design principle has been employed by many protocols in 6Lo and CoRE working group (Sections 4 and 6).
- c. Broadcast and non-synchronized transmissions consume more than other TX/RX operations. If protocols must use these ways to collect information, reduction of their usage by aggregating similar messages together will be helpful in saving power (Sections 2 and 6.1).
- d. Saving power by sleeping as much as possible is used widely (Section 3).

8. Contributors

Jens T. Petersen, RTX, contributed the section on power save services in DECT ULE.

9. Acknowledgments

Carles Gomez has been supported by the Spanish Government, FEDER and the ERDF through projects TEC2012-32531 and TEC2016-79988-P.

Authors would like to thank the review and feedback from a number of experts in this area: Carsten Bormann, Ari Keranen, Hannes Tschofenig, Dominique Barthel, Bernie Volz and Charlie Perkins.

The text of this document was improved based on IESG Document Editing session during IETF87. Thanks to Ted Lemon and Joel Jaeggli for initiating and facilitating this editing session.

10. IANA Considerations

This document has no IANA requests.

11. Security Considerations

This document discusses the energy efficient protocol design, and does not incur any changes or challenges on security issues besides what the protocol specifications have analyzed.

12. References

12.1. Normative References

[Bluetooth42]

Bluetooth Special Interest Group, "Bluetooth Core Specification Version 4.2", December 2014, <<https://www.bluetooth.org/en-us/specification/adopted-specifications>>.

[EN300]

ETSI, "Digital Enhanced Cordless Telecommunications (DECT); Common Interface (CI)", March 2015, <https://www.etsi.org/deliver/etsi_en/300100_300199/30017501/02.06.01_60/en_30017501v020601p.pdf>.

[fifteendotfour]

IEEE Computer Society, "IEEE Std. 802.15.4-2015 IEEE Standard for Local and metropolitan area networks--Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs)", 2015, <<https://standards.ieee.org/findstds/standard/802.15.4-2015.html>>.

[G9959]

International Telecommunication Union, "Short range narrow-band digital radiocommunication transceivers - PHY and MAC layer specifications, ITU-T Recommendation G.9959", January 2015, <<http://www.itu.int/rec/T-REC-G.9959>>.

[IEEE80211v]

IEEE, "Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications, Amendment 8: IEEE 802.11 Wireless Network Management.", February 2012.

- [MSTP] ANSI/ASHRAE, "Addenda: BACnet -- A Data Communication Protocol for Building Automation and Control Networks, ANSI/ASHRAE Addenda an, at, au, av, aw, ax, and az to ANSI/ASHRAE Standard 135-2012", July 2014, <https://www.ashrae.org/File%20Library/docLib/StdAddenda/07-31-2014_135_2012_an_at_au_av_aw_ax_az_Final.pdf>.
- [NFC] NFC Forum, "NFC Logical Link Control Protocol version 1.3, NFC Forum Technical Specification", March 2016.
- [oneM2M] oneM2M, "oneM2M specifications", <<http://www.onem2m.org/technical/published-documents>>.
- [RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, DOI 10.17487/RFC2460, December 1998, <<https://www.rfc-editor.org/info/rfc2460>>.
- [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>>.
- [RFC6206] Levis, P., Clausen, T., Hui, J., Gnawali, O., and J. Ko, "The Trickle Algorithm", RFC 6206, DOI 10.17487/RFC6206, March 2011, <<https://www.rfc-editor.org/info/rfc6206>>.
- [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>>.
- [RFC6298] Paxson, V., Allman, M., Chu, J., and M. Sargent, "Computing TCP's Retransmission Timer", RFC 6298, DOI 10.17487/RFC6298, June 2011, <<https://www.rfc-editor.org/info/rfc6298>>.
- [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>>.

- [RFC6775] Shelby, Z., Ed., Chakrabarti, S., Nordmark, E., and C. Bormann, "Neighbor Discovery Optimization for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs)", RFC 6775, DOI 10.17487/RFC6775, November 2012, <<https://www.rfc-editor.org/info/rfc6775>>.
- [RFC7228] Bormann, C., Ersue, M., and A. Keranen, "Terminology for Constrained-Node Networks", RFC 7228, DOI 10.17487/RFC7228, May 2014, <<https://www.rfc-editor.org/info/rfc7228>>.
- [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>>.
- [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>>.
- [TS102] ETSI, "Digital Enhanced Cordless Telecommunications (DECT); Ultra Low Energy (ULE); Machine to Machine Communications; Part 2: Home Automation Network (phase 2", March 2015, <https://www.etsi.org/deliver/etsi_ts/102900_102999/10293902/01.01.01_60/ts_10293902v010101p.pdf>.

12.2. Informative References

- [AN079] Kim, C., "Measuring Power Consumption of CC2530 With Z-Stack", September 2012, <<http://www.ti.com/lit/an/swra292/swra292.pdf>>.
- [ContikiMAC] Dunkels, A., "The ContikiMAC Radio Duty Cycling Protocol, SICS Technical Report T2011:13", December 2011, <<https://www.mysciencework.com/publication/download/2f406d3c4ccleda32a234f7alad2cc3b/7eb199e4f8b00857e21af2b7d2b31c0d>>.
- [I-D.bormann-lwig-7228bis] Bormann, C., Ersue, M., Keranen, A., and C. Gomez, "Terminology for Constrained-Node Networks", draft-bormann-lwig-7228bis-01 (work in progress), May 2017.

- [I-D.ietf-6lo-dect-ule]
Mariager, P., Petersen, J., Shelby, Z., Logt, M., and D. Barthel, "Transmission of IPv6 Packets over DECT Ultra Low Energy", draft-ietf-6lo-dect-ule-09 (work in progress), December 2016.
- [I-D.ietf-6man-impatient-nud]
Nordmark, E. and I. Gashinsky, "Neighbor Unreachability Detection is too impatient", draft-ietf-6man-impatient-nud-07 (work in progress), October 2013.
- [I-D.ietf-6tisch-architecture]
Thubert, P., "An Architecture for IPv6 over the TSCH mode of IEEE 802.15.4", draft-ietf-6tisch-architecture-12 (work in progress), August 2017.
- [I-D.ietf-6tisch-minimal]
Vilajosana, X., Pister, K., and T. Watteyne, "Minimal 6TiSCH Configuration", draft-ietf-6tisch-minimal-21 (work in progress), February 2017.
- [I-D.ietf-core-coap-pubsub]
Koster, M., Keranen, A., and J. Jimenez, "Publish-Subscribe Broker for the Constrained Application Protocol (CoAP)", draft-ietf-core-coap-pubsub-02 (work in progress), July 2017.
- [I-D.ietf-core-coap-tcp-tls]
Bormann, C., Lemay, S., Tschofenig, H., Hartke, K., Silverajan, B., and B. Raymor, "CoAP (Constrained Application Protocol) over TCP, TLS, and WebSockets", draft-ietf-core-coap-tcp-tls-09 (work in progress), May 2017.
- [I-D.ietf-core-resource-directory]
Shelby, Z., Koster, M., Bormann, C., Stok, P., and C. Amsuess, "CoRE Resource Directory", draft-ietf-core-resource-directory-11 (work in progress), July 2017.
- [I-D.ietf-lwig-crypto-sensors]
Sethi, M., Arkko, J., Keranen, A., and H. Back, "Practical Considerations and Implementation Experiences in Securing Smart Object Networks", draft-ietf-lwig-crypto-sensors-04 (work in progress), August 2017.

[I-D.kovatsch-lwig-class1-coap]

Kovatsch, M., "Implementing CoAP for Class 1 Devices",
draft-kovatsch-lwig-class1-coap-00 (work in progress),
October 2012.

[I-D.rahman-core-sleepy-nodes-do-we-need]

Rahman, A., "Sleepy Devices: Do we need to Support them in
CORE?", draft-rahman-core-sleepy-nodes-do-we-need-01 (work
in progress), February 2014.

[Powertrace]

Dunkels, Eriksson, Finne, and Tsiftes, "Powertrace:
Network-level Power Profiling for Low-power Wireless
Networks", March 2011, <[https://core.ac.uk/download/
pdf/11435067.pdf?repositoryId=362](https://core.ac.uk/download/pdf/11435067.pdf?repositoryId=362)>.

Authors' Addresses

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

Email: carlesgo@entel.upc.edu

Matthias Kovatsch
ETH Zurich
Universitaetstrasse 6
Zurich, CH-8092
Switzerland

Email: kovatsch@inf.ethz.ch

Hui Tian
China Academy of Telecommunication Research
Huayuanbeilu No.52
Beijing, Haidian District 100191
China

Email: tianhui@ritt.cn

Zhen Cao (editor)
Huawei Technologies
China

Email: zhencao.ietf@gmail.com

Light-Weight Implementation Guidance (lwig)
Internet-Draft
Intended status: Informational
Expires: April 24, 2019

D. Migault
Ericsson
T. Guggemos
LMU Munich
October 21, 2018

Minimal ESP
draft-mglt-lwig-minimal-esp-07

Abstract

This document describes a minimal implementation of the IP Encapsulation Security Payload (ESP) defined in RFC 4303. Its purpose is to enable implementation of ESP with a minimal set of options to remain compatible with ESP as described in RFC 4303. A minimal version of ESP is not intended to become a replacement of the RFC 4303 ESP, but instead to enable a limited implementation to interoperate with implementations of RFC 4303 ESP.

This document describes what is required from RFC 4303 ESP as well as various ways to optimize compliance with RFC 4303 ESP.

This document does not update or modify RFC 4303, but provides a compact description of how to implement the minimal version of the protocol. If this document and RFC 4303 conflicts then RFC 4303 is the authoritative description.

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 April 24, 2019.

Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

1. Requirements notation

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

2. Introduction

ESP [RFC4303] is part of the IPsec suite protocol [RFC4301]. IPsec is used to provide confidentiality, data origin authentication, connectionless integrity, an anti-replay service (a form of partial sequence integrity) and limited traffic flow confidentiality.

Figure 1 describes an ESP Packet. Currently ESP is implemented in the kernel of major multi purpose Operating Systems (OS). The ESP and IPsec suite is usually implemented in a complete way to fit multiple purpose usage of these OS. However, completeness of the IPsec suite as well as multi purpose scope of these OS is often performed at the expense of resources, or a lack of performance. As a result, constraint devices are likely to have their own implementation of ESP optimized and adapted to their specificities. With the adoption of IPsec by IoT devices with minimal IKEv2 [RFC7815] and ESP Header Compression (EHC) with [I-D.mglt-ipsecme-diet-esp] or [I-D.mglt-ipsecme-ikev2-diet-esp-extension], it becomes crucial that ESP implementation designed for constraint devices remain interoperable with the standard ESP implementation to avoid a fragmented usage of ESP. This document describes the the minimal properties and ESP implementation needs to meet.

For each field of the ESP packet represented in Figure 1 this document provides recommendations and guidance for minimal implementations. The primary purpose of Minimal ESP is to remain

interoperable with other nodes implementing RFC 4303 ESP, while limiting the standard complexity of the implementation.

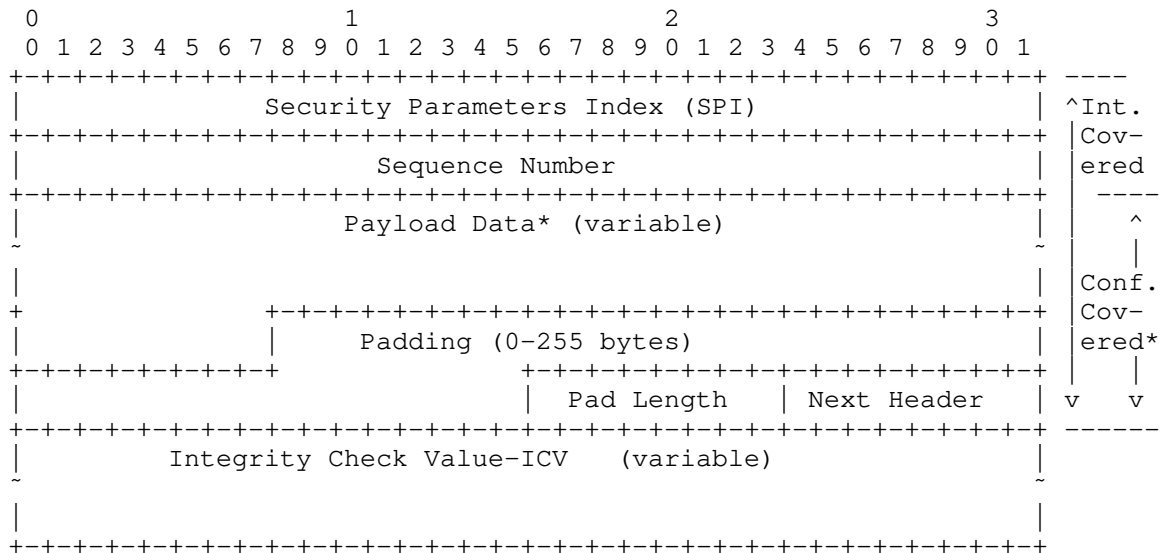


Figure 1: ESP Packet Description

3. Security Parameter Index (SPI) (32 bit)

According to the [RFC4303], the SPI is a mandatory 32 bits field and is not allowed to be removed.

The SPI has a local significance to index the Security Association (SA). From [RFC4301] section 4.1, nodes supporting only unicast communications can index their SA only using the SPI. On the other hand, nodes supporting multicast communications must also use the IP addresses and thus SA lookup needs to be performed using the longest match.

For nodes supporting only unicast communications, it is RECOMMENDED to index SA with the SPI only. Some other local constraints on the node may require a combination of the SPI as well as other parameters to index the SA.

It is RECOMMENDED to randomly generate the SPI indexing each inbound session. A random generation provides a stateless way to generate the SPIs, while keeping the probability of collision between SPIs relatively low. In case of collision, the SPI is simply re-generated.

However, for some constraint nodes, generating a random SPI may consume too much resource, in which case SPI can be generated using predictable functions or even a fix value. In fact, the SPI does not need to be random. Generating non random SPI MAY lead to privacy and security concerns. As a result, this alternative should be considered for devices that would be strongly impacted by the generation of a random SPI and after understanding the privacy and security impact of generating non random SPI.

When a constraint node uses fix value for SPIs, it imposes some limitations on the number of inbound SA. This limitation can be alleviated by how the SA lookup is performed. When fix SPI are used, it is RECOMMENDED the constraint node has as many SPI values as ESP session per host IP address, and that SA lookup includes the IP addresses.

Note that SPI value is used only for inbound traffic, as such the SPI negotiated with IKEv2 [RFC7296] or [RFC7815] by a peer, is the value used by the remote peer when it sends traffic. As SPI are only used for inbound traffic by the peer, this allows each peer to manage the set of SPIs used for its inbound traffic.

The use of fix SPI MUST NOT be considered as a way to avoid strong random generators. Such generator will be required in order to provide strong cryptographic protection and follow the randomness requirements for security described in [RFC4086]. Instead, the use of a fix SPI should only be considered as a way to overcome the resource limitations of the node, when this is feasible.

The use of a limited number of fix SPI or non random SPIs come with security or privacy drawbacks. Typically, a passive attacker may derive information such as the number of constraint devices connecting the remote peer, and in conjunction with data rate, the attacker may eventually determine the application the constraint device is associated to. If the SPI is fixed by a manufacturer or by some software application, the SPI may leak in an obvious way the type of sensor, the application involved or the model of the constraint device. When identification of the application or the hardware is associated to privacy, the SPI MUST be randomly generated. However, one needs to realize that in this case this is likely to be sufficient and a thorough privacy analysis is required. More specifically, traffic pattern MAY leak sufficient information in itself. In other words, privacy leakage is a complex and the use of random SPI is unlikely to be sufficient.

As the general recommendation is to randomly generate the SPI, constraint devices that will use a limited number of fix SPI are expected to be very constraint devices with very limited

capabilities, where the use of randomly generated SPI may prevent them to implement IPsec. In this case the ability to provision non random SPI enables these devices to secure their communications. These devices, due to there limitations, are expected to provide limited information and how the use of non random SPI impacts privacy requires further analysis. Typically temperature sensors, wind sensors, used outdoor do not leak privacy sensitive information. When used indoor, the privacy information is stored in the encrypted data and as such does not leak privacy.

As far as security is concerned, revealing the type of application or model of the constraint device could be used to identify the vulnerabilities the constraint device is subject to. This is especially sensitive for constraint devices where patches or software updates will be challenging to operate. As a result, these devices may remain vulnerable for relatively long period. In addition, predictable SPI enable an attacker to forge packets with a valid SPI. Such packet will not be rejected due to an SPI mismatch, but instead after the signature check which requires more resource and thus make DoS more efficient, especially for devices powered by batteries.

Values 0-255 SHOULD NOT be used. Values 1-255 are reserved and 0 is only allowed to be used internal and it MUST NOT be send on the wire.

[RFC4303] mentions :

"The SPI is an arbitrary 32-bit value that is used by a receiver to identify the SA to which an incoming packet is bound. The SPI field is mandatory. [...]"

"For a unicast SA, the SPI can be used by itself to specify an SA, or it may be used in conjunction with the IPsec protocol type (in this case ESP). Because the SPI value is generated by the receiver for a unicast SA, whether the value is sufficient to identify an SA by itself or whether it must be used in conjunction with the IPsec protocol value is a local matter. This mechanism for mapping inbound traffic to unicast SAs MUST be supported by all ESP implementations."

4. Sequence Number(SN) (32 bit)

According to [RFC4303], the Sequence Number (SN) is a mandatory 32 bits field in the packet.

The SN is set by the sender so the receiver can implement anti-replay protection. The SN is derived from any strictly increasing function that guarantees: if packet B is sent after packet A, then SN of packet B is strictly greater then the SN of packet A.

Some constraint devices may establish communication with specific devices, like a specific gateway, or nodes similar to them. As a result, the sender may know whereas the receiver implements anti-replay protection or not. Even though the sender may know the receiver does not implement anti replay protection, the sender **MUST** implement a always increasing function to generate the SN.

Usually, SN is generated by incrementing a counter for each packet sent. A constraint device may avoid maintaining this context and use another source that is known to always increase. Typically, constraint nodes using 802.15.4 Time Slotted Channel Hopping (TSCH), whose communication is heavily dependent on time, can take advantage of their clock to generate the SN. This would guarantee a strictly increasing function, and avoid storing any additional values or context related to the SN. When the use of a clock is considered, one should take care that packets associated to a given SA are not sent with the same time value.

For inbound traffic, it is **RECOMMENDED** to provide a anti-replay protection, and the size of the window depends on the ability of the network to deliver packet out of order. As a result, in environment where out of order packets is not possible the window size can be set to one. However, while **RECOMMENDED**, there is no requirements to implement an anti replay protection mechanism implemented by IPsec. A node **MAY** drop anti-replay protection provided by IPsec, and instead implement its own internal mechanism.

[RFC4303] mentions :

"This unsigned 32-bit field contains a counter value that increases by one for each packet sent, i.e., a per-SA packet sequence number. For a unicast SA or a single-sender multicast SA, the sender **MUST** increment this field for every transmitted packet. Sharing an SA among multiple senders is permitted, though generally not recommended. [...] The field is mandatory and **MUST** always be present even if the receiver does not elect to enable the anti-replay service for a specific SA."

5. Padding

The purpose of padding is to respect the 32 bit alignment of ESP. ESP **MUST** have at least one padding byte Pad Length that indicates the padding length. ESP padding bytes are generated by a succession of unsigned bytes starting with 1, 2, 3 with the last byte set to Pad Length, where Pad Length designates the length of the padding bytes.

Checking the padding structure is not mandatory, so the constraint device may not proceed to such checks, however, in order to

interoperate with existing ESP implementations, it MUST build the padding bytes as recommended by ESP.

In some situation the padding bytes may take a fix value. This would typically be the case when the Data Payload is of fix size.

[RFC4303] mentions :

"If Padding bytes are needed but the encryption algorithm does not specify the padding contents, then the following default processing MUST be used. The Padding bytes are initialized with a series of (unsigned, 1-byte) integer values. The first padding byte appended to the plaintext is numbered 1, with subsequent padding bytes making up a monotonically increasing sequence: 1, 2, 3, When this padding scheme is employed, the receiver SHOULD inspect the Padding field. (This scheme was selected because of its relative simplicity, ease of implementation in hardware, and because it offers limited protection against certain forms of "cut and paste" attacks in the absence of other integrity measures, if the receiver checks the padding values upon decryption.)"

ESP [RFC4303] also provides Traffic Flow Confidentiality (TFC) as a way to perform padding to hide traffic characteristics, which differs from respecting a 32 bit alignment. TFC is not mandatory and MUST be negotiated with the SA management protocol. TFC has not yet being widely adopted for standard ESP traffic. One possible reason is that it requires to shape the traffic according to one traffic pattern that needs to be maintained. This is likely to require extra processing as well as providing a "well recognized" traffic shape which could end up being counterproductive. As such TFC is not expected to be supported by a minimal ESP implementation.

As a result, TFC cannot not be enabled with minimal, and communication protection that were relying on TFC will be more sensitive to traffic shaping. This could expose the application as well as the devices used to a passive monitoring attacker. Such information could be used by the attacker in case a vulnerability is disclosed on the specific device. In addition, some application use - such as health applications - may also reveal important privacy oriented informations.

Some constraint nodes that have limited battery life time may also prefer avoiding sending extra padding bytes. However the same nodes may also be very specific to an application and device. As a result, they are also likely to be the main target for traffic shaping. In most cases, the payload carried by these nodes is quite small, and the standard padding mechanism may also be used as an alternative to TFC, with a sufficient trade off between the require energy to send

additional payload and the exposure to traffic shaping attacks. In addition, the information leaked by the traffic shaping may also be addressed by the application level. For example, it is preferred to have a sensor sending some information at regular time interval, rather when an specific event is happening. Typically a sensor monitoring the temperature, or a door is expected to send regularly the information - i.e. the temperature of the room or whether the door is closed or open) instead of only sending the information when the temperature has raised or when the door is being opened.

6. Next Header (8 bit)

According to [RFC4303], the Next Header is a mandatory 8 bits field in the packet. Next header is intended to specify the data contained in the payload as well as dummy packet. In addition, the Next Header may also carry an indication on how to process the packet [I-D.nikander-esp-beet-mode].

The ability to generate and receive dummy packet is required by [RFC4303]. For interoperability, it is RECOMMENDED a minimal ESP implementation discards dummy packets. Note that such recommendation only applies for nodes receiving packets, and that nodes designed to only send data may not implement this capability.

As the generation of dummy packets is subject to local management and based on a per-SA basis, a minimal ESP implementation may not generate such dummy packet. More especially, in constraint environment sending dummy packets may have too much impact on the device life time, and so may be avoided. On the other hand, constraint nodes may be dedicated to specific applications, in which case, traffic pattern may expose the application or the type of node. For these nodes, not sending dummy packet may have some privacy implication that needs to be measured. However, for the same reasons exposed in Section 5 traffic shaping at the IPsec layer may also introduce some traffic pattern, and on constraint devices the application is probably the most appropriated layer to limit the risk of leaking information by traffic shaping.

In some cases, devices are dedicated to a single application or a single transport protocol, in which case, the Next Header has a fix value.

Specific processing indications have not been standardized yet [I-D.nikander-esp-beet-mode] and is expected to result from an agreement between the peers. As a result, it is not expected to be part of a minimal implementation of ESP.

[RFC4303] mentions :

"The Next Header is a mandatory, 8-bit field that identifies the type of data contained in the Payload Data field, e.g., an IPv4 or IPv6 packet, or a next layer header and data. [...] the protocol value 59 (which means "no next header") MUST be used to designate a "dummy" packet. A transmitter MUST be capable of generating dummy packets marked with this value in the next protocol field, and a receiver MUST be prepared to discard such packets, without indicating an error."

7. ICV

The ICV depends on the crypto-suite used. Currently recommended [RFC8221] only recommend crypto-suites with an ICV which makes the ICV a mandatory field.

As detailed in Section 8 we recommend to use authentication, the ICV field is expected to be present that is to say with a size different from zero. This makes it a mandatory field which size is defined by the security recommendations only.

[RFC4303] mentions :

"The Integrity Check Value is a variable-length field computed over the ESP header, Payload, and ESP trailer fields. Implicit ESP trailer fields (integrity padding and high-order ESN bits, if applicable) are included in the ICV computation. The ICV field is optional. It is present only if the integrity service is selected and is provided by either a separate integrity algorithm or a combined mode algorithm that uses an ICV. The length of the field is specified by the integrity algorithm selected and associated with the SA. The integrity algorithm specification MUST specify the length of the ICV and the comparison rules and processing steps for validation."

8. Cryptographic Suites

The cryptographic suites implemented are an important component of ESP. The recommended suites to use are expected to evolve over time and implementer SHOULD follow the recommendations provided by [RFC8221] and updates. Recommendations are provided for standard nodes as well as constraint nodes.

This section lists some of the criteria that may be considered. The list is not expected to be exhaustive and may also evolve overtime. As a result, the list is provided as indicative:

1. Security: Security is the criteria that should be considered first for the selection of cipher suites. The security of cipher

suites is expected to evolve over time, and it is of primary importance to follow up-to-date security guidances and recommendations. The chosen cipher suites MUST NOT be known vulnerable or weak (see [RFC8221] for outdated ciphers). ESP can be used to authenticate only or to encrypt the communication. In the later case, authenticated encryption must always be considered [RFC8221].

2. **Interoperability:** Interoperability considers the cipher suites shared with the other nodes. Note that it is not because a cipher suite is widely deployed that is secured. As a result, security SHOULD NOT be weakened for interoperability. [RFC8221] and successors consider the life cycle of cipher suites sufficiently long to provide interoperability. Constraint devices may have limited interoperability requirements which makes possible to reduce the number of cipher suites to implement.
3. **Power Consumption and Cipher Suite Complexity:** Complexity of the cipher suite or the energy associated to it are especially considered when devices have limited resources or are using some batteries, in which case the battery determines the life of the device. The choice of a cryptographic function may consider re-using specific libraries or to take advantage of hardware acceleration provided by the device. For example if the device benefits from AES hardware modules and uses AES-CTR, it may prefer AUTH_AES-XCBC for its authentication. In addition, some devices may also embed radio modules with hardware acceleration for AES-CCM, in which case, this mode may be preferred.
4. **Power Consumption and Bandwidth Consumption:** Similarly to the cipher suite complexity, reducing the payload sent, may significantly reduce the energy consumption of the device. As a result, cipher suites with low overhead may be considered. To reduce the overall payload size one may for example:
 1. Use of counter-based ciphers without fixed block length (e.g. AES-CTR, or ChaCha20-Poly1305).
 2. Use of ciphers with capability of using implicit IVs [I-D.ietf-ipsecme-implicit-iv].
 3. Use of ciphers recommended for IoT [RFC8221].
 4. Avoid Padding by sending payload data which are aligned to the cipher block length - 2 for the ESP trailer.

9. IANA Considerations

There are no IANA consideration for this document.

10. Security Considerations

Security considerations are those of [RFC4303]. In addition, this document provided security recommendations and guidances over the implementation choices for each fields.

11. Acknowledgment

The authors would like to thank Daniel Palomares, Scott Fluhrer, Tero Kivinen, Valery Smyslov, Yoav Nir, Michael Richardson for their valuable comments.

12. References

12.1. Normative References

- [I-D.ietf-ipsecme-implicit-iv]
Migault, D., Guggemos, T., and Y. Nir, "Implicit IV for Counter-based Ciphers in Encapsulating Security Payload (ESP)", draft-ietf-ipsecme-implicit-iv-05 (work in progress), June 2018.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC4086] Eastlake 3rd, D., Schiller, J., and S. Crocker, "Randomness Requirements for Security", BCP 106, RFC 4086, DOI 10.17487/RFC4086, June 2005, <<https://www.rfc-editor.org/info/rfc4086>>.
- [RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, DOI 10.17487/RFC4301, December 2005, <<https://www.rfc-editor.org/info/rfc4301>>.
- [RFC4303] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, DOI 10.17487/RFC4303, December 2005, <<https://www.rfc-editor.org/info/rfc4303>>.
- [RFC7296] Kaufman, C., Hoffman, P., Nir, Y., Eronen, P., and T. Kivinen, "Internet Key Exchange Protocol Version 2 (IKEv2)", STD 79, RFC 7296, DOI 10.17487/RFC7296, October 2014, <<https://www.rfc-editor.org/info/rfc7296>>.

- [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>>.
- [RFC8221] Wouters, P., Migault, D., Mattsson, J., Nir, Y., and T. Kivinen, "Cryptographic Algorithm Implementation Requirements and Usage Guidance for Encapsulating Security Payload (ESP) and Authentication Header (AH)", RFC 8221, DOI 10.17487/RFC8221, October 2017, <<https://www.rfc-editor.org/info/rfc8221>>.

12.2. Informative References

- [I-D.mglt-ipsecme-diet-esp]
Migault, D., Guggemos, T., Bormann, C., and D. Schinazi, "ESP Header Compression and Diet-ESP", draft-mglt-ipsecme-diet-esp-06 (work in progress), May 2018.
- [I-D.mglt-ipsecme-ikev2-diet-esp-extension]
Migault, D., Guggemos, T., and D. Schinazi, "Internet Key Exchange version 2 (IKEv2) extension for the ESP Header Compression (EHC) Strategy", draft-mglt-ipsecme-ikev2-diet-esp-extension-01 (work in progress), June 2018.
- [I-D.nikander-esp-beet-mode]
Nikander, P. and J. Melen, "A Bound End-to-End Tunnel (BEET) mode for ESP", draft-nikander-esp-beet-mode-09 (work in progress), August 2008.

Appendix A. Document Change Log

[RFC Editor: This section is to be removed before publication]

-00: First version published.

-01: Clarified description

-02: Clarified description

Authors' Addresses

Daniel Migault
Ericsson
8400 boulevard Decarie
Montreal, QC H4P 2N2
Canada

Email: daniel.migault@ericsson.com

Tobias Guggemos
LMU Munich
MNM-Team
Oettingenstr. 67
80538 Munich, Bavaria
Germany

Email: guggemos@mn-m-team.org