

6LoWAPP	G. Moritz, Ed.	
Internet-Draft	University of Rostock	
Intended status: Informational	June 16, 2010	
Expires: December 18, 2010		

[TOC](#)

DPWS for 6LoWPAN **draft-moritz-6lowapp-dpws-enhancements-01**

Abstract

This draft describes adaptations and enhancements for deploying the Devices Profile for Web Service (DPWS) in 6LoWPAN networks.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79. This document may not be modified, and derivative works of it may not be created, and it may not be published except as an Internet-Draft.

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 December 18, 2010.

Copyright Notice

Copyright (c) 2010 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
 - [1.1.](#) Requirements Language
 - [1.2.](#) Terminology
- [2.](#) Discovery optimizations
 - [2.1.](#) General adaptations
 - [2.2.](#) Discovery addressing
 - [2.3.](#) Discovery proxy
- [3.](#) Message compression
 - [3.1.](#) HTTP compression
 - [3.2.](#) SOAP compression
 - [3.3.](#) Compression integration
 - [3.3.1.](#) Payload compression
 - [3.3.2.](#) TCP vs. UDP
- [4.](#) IANA Considerations
- [5.](#) Security Considerations
- [6.](#) References
 - [6.1.](#) Normative References
 - [6.2.](#) Informative References
- [S](#) Author's Address

1. Introduction

[TOC](#)

In August 2008 a technical committee (TC) at OASIS was formed for the Web Services Discovery and Web Services Devices Profile ([WS-DD](#)) ([OASIS Open, "OASIS Web Services Discovery and Web Services Devices Profile \(WS-DD\) TC," 2009.](#)) [WS-DD]. WS-DD standardizes a lightweight subset of the Web services protocol suite that makes it easy to find, share, and control devices on a network.

The work of this TC is based on the former DPWS, WS-Discovery, and SOAP-over-UDP specifications. DPWS makes use of existing Web services protocols, but also adds several extensions to enable Web services based communication on embedded devices also. Thereby, DPWS includes features like (1) discovery of devices and metadata exchange with services even in dynamic changing environments (2) eventing about state changes by WS-Eventing (3) security and integrity for discovery, metadata exchange and service usages. Because DPWS bases on existing Web services standards, it is fully capable of being integrated in the Web services framework.

This draft describes several adaptations and enhancements to expand DPWS deployments to 6LoWPAN networks, but is far away from a comprehensive specification. It only presents a basis for further discussions. Main scope is the definition of a profile, to describe: message compression and bidirectional message reduction, while staying fully compliant with

existing WS-DD specifications. The deployment of this profile is fully transparent for existing DPWS implementations and describes extension to be considered by 6LoWPAN networks only.

Readers of this draft should have a basic knowledge about the specifications [DPWS \(Driscoll and Mensch, "OASIS Devices Profile for Web Services \(DPWS\) Version 1.1," 2009.\)](#) [DPWS], [WS-Discovery \(Modi and Kemp, "OASIS Web Services Dynamic Discovery \(WS-Discovery\) Version 1.1," 2009.\)](#) [WS-Discovery], [SOAP-over-UDP \(Jeyaraman, "OASIS SOAP-over-UDP Version 1.1," 2009.\)](#) [SOAP-over-UDP] and related standards like SOAP, HTTP, XML and XML Schema.

1.1. Requirements Language

[TOC](#)

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 [RFC 2119 \(Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels," March 1997.\)](#) [RFC2119].

1.2. Terminology

[TOC](#)

DPWS In the remainder of this draft, DPWS is used as general term for the WS-DD specifications [DPWS \(Driscoll and Mensch, "OASIS Devices Profile for Web Services \(DPWS\) Version 1.1," 2009.\)](#) [DPWS], [WS-Discovery \(Modi and Kemp, "OASIS Web Services Dynamic Discovery \(WS-Discovery\) Version 1.1," 2009.\)](#) [WS-Discovery], and [SOAP-over-UDP \(Jeyaraman, "OASIS SOAP-over-UDP Version 1.1," 2009.\)](#) [SOAP-over-UDP].

DPWS specification Points to the [DPWS \(Driscoll and Mensch, "OASIS Devices Profile for Web Services \(DPWS\) Version 1.1," 2009.\)](#) [DPWS] specification directly.

Device A device is an endpoint implementing DPWS device functionalities as specified by [WS-DD \(OASIS Open, "OASIS Web Services Discovery and Web Services Devices Profile \(WS-DD\) TC," 2009.\)](#) [WS-DD].

Client A client is an endpoint implementing DPWS client functionalities as specified by [WS-DD \(OASIS Open, "OASIS Web](#)

[Services Discovery and Web Services Devices Profile \(WS-DD\) TC," 2009.\)](#) [WS-DD].

Hello The Hello message of a device as defined in [WS-Discovery \(Modi and Kemp, "OASIS Web Services Dynamic Discovery \(WS-Discovery\) Version 1.1," 2009.\)](#) [WS-Discovery].

Bye The Bye message of a device as defined in [WS-Discovery \(Modi and Kemp, "OASIS Web Services Dynamic Discovery \(WS-Discovery\) Version 1.1," 2009.\)](#) [WS-Discovery].

Probe The Probe message of a client as defined in [WS-Discovery \(Modi and Kemp, "OASIS Web Services Dynamic Discovery \(WS-Discovery\) Version 1.1," 2009.\)](#) [WS-Discovery].

Probe Match The Probe Match message of a device as defined in [WS-Discovery \(Modi and Kemp, "OASIS Web Services Dynamic Discovery \(WS-Discovery\) Version 1.1," 2009.\)](#) [WS-Discovery].

Resolve The Resolve message of a client as defined in [WS-Discovery \(Modi and Kemp, "OASIS Web Services Dynamic Discovery \(WS-Discovery\) Version 1.1," 2009.\)](#) [WS-Discovery].

Resolve Match The Resolve Match message of a device as defined in [WS-Discovery \(Modi and Kemp, "OASIS Web Services Dynamic Discovery \(WS-Discovery\) Version 1.1," 2009.\)](#) [WS-Discovery].

WSDL Acronym for the document describing the services in [Web Services Description Language \(Christensen, Curbera, Meredith, and Weerawarana, "W3C Web Services Description Language \(WSDL\) 1.1," March 2001.\)](#) [WSDL] format.

Edge Router Edge Routers are the routers that connect LWPANs to an IPv6 infrastructure via backhaul or backbone links when such an infrastructure is available. (Defined in [6LoWPAN Neighbor Discovery \(Shelby, Z., Thubert, P., Hui, J., Chakrabarti, S., Bormann, C., and E. Nordmark, "6LoWPAN Neighbor Discovery," October 2009.\)](#) [I-D.ietf-6lowpan-nd])

2. Discovery optimizations

[TOC](#)

DPWS describes two different modes for discovery of devices: ad-hoc mode and managed mode. In managed mode, a registry called Discovery Proxy is applied to suppress multicast messages. This section describes adaptations for both of these modes.

2.1. General adaptions

[TOC](#)

A DEVICE MUST include the transport specific addresses in its Hello and Probe Match messages.

In accordance to DPWS, embedding of a transport specific address in Hello and Probe messages is not mandatory. This behavior is counterproductive for WSN, with the constraints for energy consumption and limited bandwidth. Thus, the optional fields for the transport specific addresses are now mandatory to avoid Resolve messages.

A DEVICE SHOULD include all device types in Hello and Probe Match messages.

In the current version of DPWS it is not mandatory to include the type field in the Hello and Probe Match messages. A client MAY infer to services provided by the device with the help of the device type. Inclusion of the device type can avoid further discovery and metadata exchange messages.

A SERVICE SHOULD NOT provide the WSDL file for CLIENTS at run time. Providing the WSDL during the discovery phase is optional in DPWS. WSDL are used in general at development time only for code generation. These WSDL files have a size of several kB in most analyzed scenarios. The expensive and memory consuming storage of these WSDL files on the device and on the client node is not applicable for WSN. Furthermore, the exchange itself consumes a high bandwidth.

2.2. Discovery addressing

[TOC](#)

*All SOAP-over-UDP messages inside the 6LoWPAN network MUST use the port 61616 as target port. (Exact port to be defined)

*Devices inside the 6LoWPAN network MUST listen to the IPv6 multicast address: FF02::C0. (Exact address to be defined)

*Clients inside the 6LoWPAN network MUST listen to the IPv6 multicast address: FF02::C1. (Exact address to be defined)

In RFC4944, a UDP port compression is described which works most efficient when using ports in a specific range. Thus, the used ports should be changed to fit in this range. The ports have to be mapped on the edge router of the 6LoWPAN network for incoming and outgoing SOAP-over-UDP messages.

DPWS defines one IPv4 and one IPv6 multicast addresses to be used for discovery message exchange. But DPWS differentiates between device and client roles. Hence there are messages to be processed by clients (Hello) and by devices (Probe, Resolve) exclusively. Usage of one and the same address for all these messages implies overhead in sending to

and receiving of these messages independent of the nodes role. Inside 6LoWPAN networks, different addresses have to be used. The mapping into compliant addresses is done by the edge router of the 6LoWPAN network. The multicast addresses for discovery have a link local scope. For a transparent integration, in ad-hoc mode, edge routers of 6LoWPAN networks SHOULD forward incoming and outgoing link-local scope multicast discovery messages. Another solution might be to use a different scope for the multicast discovery messaging.

2.3. Discovery proxy

[TOC](#)

In managed mode, a device and service registry is applied. It is possible, to use more than one discovery proxy in a network. In managed mode, one discovery proxy SHOULD be deployed at the edge router to hide expensive external multicast messages from the 6LoWPAN network and omit multicast flooding.

3. Message compression

[TOC](#)

Because DPWS bases on SOAP and thus on XML for data representation, XML compression techniques and/or encoding concepts have to be used to reduce message sizes.

3.1. HTTP compression

[TOC](#)

DPWS for 6LoWPAN requires HTTP header compression. While [CHOPAN \(Frank, B., "Chopan - Compressed HTTP Over PANs," September 2009.\)](#) [I-D.frank-6lowapp-chopan] describes a general and generic HTTP compression, this draft focuses on a more DPWS specific compression scheme as described here.

DPWS uses SOAP-over-HTTP binding for unicast messaging. All messages are using the POST method of HTTP in version 1.1. The transport specific addresses (target host) can be elided and derived from transport layer. In accordance to HTTP 1.1, all connections marked not explicit as close are keep-alive connections. But keep-alive TCP connections are not applicable for WSN. The SOAPAction field is mandatory when using the SOAP-over-HTTP binding, but can be empty. Because DPWS uses WS-Addressing, the SOAPAction field is redundant. The content-type of SOAP-over-HTTP is always application/soap+xml; charset=utf-8. To sum up, only few fields are left, which are analyzed by devices and clients and which provide useful information. A specific

compression scheme is required to omit unnecessary HTTP header fields and allow a compression (optimized binary format) of the remaining required fields. The fields which have to be left are:

Content-length: This is required to allow resource constrained sensor nodes to know about length of data to be analyzed and thus e.g. required memory allocations.

HTTP Status Codes: The status code may be analyzed in error and fault cases. Status code 200/OK can be implied if this field is missing, to use it only in error/fault cases.

3.2. SOAP compression

[TOC](#)

Different XML specific and XML non-specific compression schemes are already known. The following table presents an overview about existing schemes and compressors, including the emerging EXI format of the W3C. The table shows resulting sizes of different compressors applied to DPWS messages. The values present the sizes of the SOAP envelopes (excluding HTTP headers) after compression and in the last line pure XML messages for comparison. The messages were recorded in a realistic scenario, implemented by using compliant DPWS toolkits. An overall number of 18 different message types are included in the evaluations and the table shows the averages over all these message types. Most of the compressors suffer from the simple XML structures. Sensor nodes will not provide complex services and thus simple message have to be assumed. These measurements might provide a basis for further discussions on message size optimizations.

For the measurements of EXI schema-informed format, separated results are presented: optimized and default. The default format used XML schema files as defined in DPWS without optimizations. This includes an inconsistency of different namespaces and versions used by DPWS and among Web services specifications (especially WS-Addressing and WS-Eventing). For the optimized format, some improvements are added. Most values of the XML tags and attributes in DPWS are well-known URIs. If these values are included in the XML schema files and with corrected dependencies, the average message size is reduced significantly as presented in the table.

Using the table below as starting point, further investigations were necessary with special focus on EXI. The influence of the Deflate compression and Bit/Byte aligned modes of EXI is minor compared to the optimization possibilities of the schema files. With carefully optimized schema files, the message sizes could be further reduced to ~96 Byte on an average in Bit aligned compressed mode (~121 Bytes in Byte aligned mode) with a minimum of 10 Bytes for the smallest message (21 Byte in Byte aligned mode) and a maximum of 391 Bytes for the biggest message (464 Byte in Byte aligned mode).

It is important to notice that the encoded messages used for the evaluations are still compliant SOAP envelopes that can be re-encoded in compliant XML utf-8 if the corresponding XML schema documents are used.

Compressor	Average in Byte	Average in %	Minimum in Byte	Maximum in Byte
EXI schema-informed (optimized)	153,72	19,97	66,00	349,00
EXI schema-informed (default)	206,61	26,49	122,00	415,00
Fast Infoset schema-informed/with compression	218,39	27,98	103,00	455,00
Fast Infoset schema-informed/without compression	242,00	30,09	97,00	563,00
EXI schema-less/with compression	315,67	40,31	192,00	633,00
gzip (C=9)	419,83	54,54	271,00	749,00
XMLPPM	427,44	55,61	274,00	755,00
gzip (C=1)	437,83	56,96	297,00	799,00
Xmill (C=9)	457,89	59,46	300,00	824,00
Xmill (C=1)	463,56	60,14	303,00	852,00
EXI schema-less/without compression	467,77	59,64	234,00	1118,00
bzip2 (C=1)	472,94	61,41	304,00	852,00
bzip2 (C=9)	474,78	61,82	315,00	852,00
Fast Infoset (schema-less)	561,89	69,70	315,00	1301,00
XML	814,89	100,00	418,00	2089,00

Table 1: DPWS SOAP envelope compression comparison

3.3. Compression integration

[TOC](#)

This section describes general points, which might be discussed more in general in 6LoWAPP.

3.3.1. Payload compression

[TOC](#)

Many protocols (like DPWS) already provide concepts for discovery of devices (ad-hoc networking), data dissemination, eventing, etc. 6LoWPAN protocols allow a seamless connectivity of IP networks with wireless sensor networks, without the need for application layer gateways. These gateways must be aware of application layer data and need an understanding of semantics of payload. Instead, the communication of 6LoWPAN nodes with existing networking devices or other existing implementations must be transparent and invisible for these external hosts. Application layer data compression and encoding should only affect the 6LoWPAN network and communication inside the network, like 6LoWPAN does with IPv6 headers. But payload on application layer is not self-contained in one packet like IP, TCP and UDP headers. Defining new extension to be implemented by the external nodes is not a proper solution and violates the core concept of 6LoWPAN protocols. New possibilities for application layer data encoding must be found, to allow efficient in-transit data encoding for traffic inside the 6LoWPAN network without affecting external communication. A new Encoding Router (ER) role might be defined. This ER is located at the edge router and not only translates compressed network and transport layer protocols, but also standardized application layer encoding and compression (e.g. EXI format in compliant XML/SOAP envelopes). This requires no understanding of semantics of the payload, but allows a seamless connectivity. The deployment of an ER might violate the layered model, because the ER must receive external message as representative to the 6LoWPAN nodes, encodes the messages and forwards them (outgoing traffic vice versa).

3.3.2. TCP vs. UDP

[TOC](#)

TCP makes use of mechanisms like sliding window and flow control, to optimize throughput. These mechanisms are questionable in wireless sensor networks. To reduce TCP overhead, also UDP might be used inside 6LoWPAN networks instead. But most application layer protocols base on TCP because of the required end-to-end reliability. The usage of the lightweight UDP for internal communication instead of TCP would require additional mechanisms to assure end-to-end reliability between endpoints. Also definition of an extension for UDP to provide this functionality is possible, but reinventing TCP must be omitted.

[TOC](#)

4. IANA Considerations

The new defined discovery addresses have to be registered at IANA.

5. Security Considerations

[TOC](#)

No security issues have been identified in this draft.

6. References

[TOC](#)

6.1. Normative References

[TOC](#)

[RFC2119]	Bradner, S. , " Key words for use in RFCs to Indicate Requirement Levels ," BCP 14, RFC 2119, March 1997 (TXT , HTML , XML).
-----------	--

6.2. Informative References

[TOC](#)

[DPWS]	Driscoll and Mensch, " OASIS Devices Profile for Web Services (DPWS) Version 1.1 ," 2009.
[EXI-format]	Scheider and Kamiya, " W3C Efficient XML Interchange (EXI) Format 1.0 Candidate Recommendation ," December 2009.
[I-D.frank-6lowapp-chopan]	Frank, B., " Chopan - Compressed HTTP Over PANs ," draft-frank-6lowapp-chopan-00 (work in progress), September 2009 (TXT).
[I-D.ietf-6lowpan-nd]	Shelby, Z., Thubert, P., Hui, J., Chakrabarti, S., Bormann, C., and E. Nordmark, " 6LoWPAN Neighbor Discovery ," draft-ietf-6lowpan-nd-07 (work in progress), October 2009 (TXT).
[SOAP-over-UDP]	Jeyaraman, " OASIS SOAP-over-UDP Version 1.1 ," 2009.
[WS-DD]	OASIS Open, " OASIS Web Services Discovery and Web Services Devices Profile (WS-DD) TC ," 2009.
[WS-Discovery]	Modi and Kemp, " OASIS Web Services Dynamic Discovery (WS-Discovery) Version 1.1 ," 2009.
[WSDL]	Christensen, Curbera, Meredith, and Weerawarana, " W3C Web Services Description Language (WSDL) 1.1 ," March 2001.

Author's Address[TOC](#)

	Guido Moritz (editor)
	University of Rostock
	18051 Rostock,
	Germany
Phone:	+49 381 498 7269
Email:	guido.moritz@uni-rostock.de
URI:	http://www.imd.uni-rostock.de/