DOTS
Internet-Draft
Intended status: Standards Track
Expires: December 31, 2021

M. Boucadair
Orange
J. Shallow

June 29, 2021

Distributed Denial-of-Service Open Threat Signaling (DOTS) Signal Channel Configuration Attributes for Robust Block Transmission draft-bosh-dots-quick-blocks-03

Abstract

This document specifies new DOTS signal channel configuration parameters that are negotiated between DOTS peers to enable the use of Q-Block1 and Q-Block2 Options. These options enable robust and faster transmission rates for large amounts of data with less packet interchanges as well as supporting faster recovery should any of the blocks get lost in transmission.

Status of This Memo

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

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

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

This Internet-Draft will expire on December 31, 2021.

Table of Contents

1. Introduction

The Constrained Application Protocol (CoAP) [RFC7252], although inspired by HTTP, was designed to use UDP instead of TCP. The message layer of CoAP over UDP includes support for reliable delivery, simple congestion control, and flow control. [RFC7959] introduced the CoAP Block1 and Block2 Options to handle data records that cannot fit in a single IP packet, so not having to rely on IP fragmentation and was further updated by [RFC8323] for use over TCP, TLS, and WebSockets.

The CoAP Block1 and Block2 Options work well in environments where there are no or minimal packet losses. These options operate synchronously where each individual block has to be requested and can only ask for (or send) the next block when the request for the previous block has completed. Packet, and hence block transmission rate, is controlled by Round Trip Times (RTTs).

There is a requirement for these blocks of data to be transmitted at higher rates under network conditions where there may be asymmetrical transient packet loss (i.e., responses may get dropped). An example is when a network is subject to a Distributed Denial of Service (DDoS) attack and there is a need for DDoS mitigation agents relying upon CoAP to communicate with each other (e.g., [I-D.ietf-dots-telemetry]). As a reminder, [RFC7959] recommends the use of Confirmable (CON) responses to handle potential packet loss.

However, such a recommendation does not work with a flooded pipe DDoS situation as the returning ACK packets may not get through.

The block-wise transfer specified in [RFC7959] covers the general case, but falls short in situations where packet loss is highly asymmetrical. The mechanism specified in [I-D.ietf-core-new-block] provides roughly similar features to the Block1/Block2 Options. It provides additional properties that are tailored towards the intended DOTS transmission. Concretely, [I-D.ietf-core-new-block] primarily targets applications such as DDoS Open Threat Signaling (DOTS) that can't use Confirmable (CON) responses to handle potential packet loss and that support application-specific mechanisms to assess whether the remote peer is able to handle the messages sent by a CoAP endpoint (e.g., DOTS heartbeats in Section 4.7 of [I-D.ietf-dots-rfc8782-bis]).

[I-D.ietf-core-new-block] includes guards to prevent a CoAP agent from overloading the network by adopting an aggressive sending rate. These guards are followed in addition to the existing CoAP congestion control as specified in Section 4.7 of [RFC7252]. Table 1 additional CoAP attributes that are used for the guards.

Parameter Name	Default Value
MAX_PAYLOADS	10
NON_MAX_RETRANSMIT	4
NON_TIMEOUT	2 s
NON_RECEIVE_TIMEOUT	4 s
NON_PROBING_WAIT	247 s
NON_PARTIAL_TIMEOUT	247 s

Table 1: Congestion Control Parameters

PROBING_RATE and other transmission parameters are negotiated between DOTS peers as discussed in Section 4.5.2 of [I-D.ietf-dots-rfc8782-bis]. Nevertheless, the attributes listed in Table 1 are not supported. This document defines new DOTS signal channel attributes that are meant to customize the configuration of robust block transmission in a DOTS context.

2. Terminology

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

14 [RFC2119][RFC8174] when, and only when, they appear in all capitals, as shown here.

Readers should be familiar with the terms and concepts defined in [RFC7252] and [RFC8612].

The terms "payload" and "body" are defined in [RFC7959]. The term "payload" is thus used for the content of a single CoAP message (i.e., a single block being transferred), while the term "body" is used for the entire resource representation that is being transferred in a block-wise fashion.

The meaning of the symbols in YANG tree diagrams are defined in [RFC8340] and [RFC8791].

(D)TLS is used for statements that apply to both Transport Layer Security (TLS) [RFC8446] and Datagram Transport Layer Security (DTLS) [RFC6347]. Specific terms are used for any statement that applies to either protocol alone.

3. DOTS Attributes for Robust Block Transmission

Section 6.2 of [I-D.ietf-core-new-block] defines the following attributes that are used for congestion control purposes:

MAX_PAYLOADS: is the maximum number of payloads that can be transmitted at any one time.

NON_MAX_RETRANSMIT: is the maximum number of times a request for the retransmission of missings payloads can occur without a response from the remote peer. By default, NON_MAX_RETRANSMIT has the same value as MAX_RETRANSMIT (Section 4.8 of [RFC7252]).

NON_TIMEOUT: is the maximum period of delay between sending sets of MAX_PAYLOADS payloads for the same body. NON_TIMEOUT has the same value as ACK_TIMEOUT (Section 4.8 of [RFC7252]).

NON_RECEIVE_TIMEOUT: is the maximum time to wait for a missing payload before requesting retransmission. By default, NON_RECEIVE_TIMEOUT has a value of twice NON_TIMEOUT.

NON_PROBING_WAIT: is used to limit the potential wait needed calculated when using PROBING_WAIT. By default, NON_PROBING_WAIT has the same value as EXCHANGE_LIFETIME (Section 4.8.2 of [RFC7252]).

NON_PARTIAL_TIMEOUT: is used for expiring partially received bodies. By default, NON_PARTIAL_TIMEOUT has the same value as EXCHANGE_LIFETIME (Section 4.8.2 of [RFC7252]).

These attributes are used together with PROBING_RATE parameter which in CoAP indicates the average data rate that must not be exceeded by a CoAP endpoint in sending to a peer endpoint that does not respond. The single body of blocks will be subjected to PROBING_RATE (Section 4.7 of [RFC7252]), not the individual packets. If the wait time between sending bodies that are not being responded to calculated using on PROBING_RATE exceeds NON_PROBING_WAIT, then the gap time is limited to NON_PROBING_WAIT.

This document augments the "ietf-dots-signal-channel" (dots-signal) DOTS signal YANG module defined in [I-D.ietf-dots-rfc8782-bis] with these additional attributes that can be negotiated between DOTS peers to enable robust and faster transmission:

max-payloads: This attribute echoes the MAX_PAYLOADS parameter in [I-D.ietf-core-new-block].

This is an optional attribute.

For the sake of more flexible configuration, this document defines also the following attributes:

non-max-retransmit: This attribute echoes the NON_MAX_RETRANSMIT parameter in [I-D.ietf-core-new-block]. The default value of this attribute is 'max-retransmit'. Note that DOTS uses a default value of '3' instead of '4' used for the generic CoAP use (Section 4.5.2 of [I-D.ietf-dots-rfc8782-bis]) for max-transmit.

This is an optional attribute.

non-timeout: This attribute echoes the NON_TIMEOUT parameter in [I-D.ietf-core-new-block]. The default value of this attribute is 'ack-timeout'.

This is an optional attribute.

non-probing-wait: This attribute echoes the NON_PROBING_WAIT parameter in [I-D.ietf-core-new-block]. The default value of this attribute is 247s.

This is an optional attribute.

non-partial-timeout: This attribute echoes the NON_PARTIAL_TIMEOUT parameter in [I-D.ietf-core-new-block]. The default value of this attribute is 274s.

This is an optional attribute.

An example of PUT message to convey the configuration parameters for the DOTS signal channel is depicted in Figure 1. In this example, the 'max-payloads' is set to '15' when no mitigation is active, while it is set to '10' when a mitigation is active. The same value is used for both 'non-max-retransmit' and 'non-timeout' in idle and mitigation times.

```
Header: PUT (Code=0.03)
Uri-Path: ".well-known"
Uri-Path: "dots"
Uri-Path: "config"
Uri-Path: "sid=123"
Content-Format: "application/dots+cbor"

{
  "ietf-dots-signal-channel:signal-config": {
    "mitigating-config": {
      "heartbeat-interval": {
        "current-value": 30
      },
      "missing-hb-allowed": {
        "current-value": 15
      },
      "probing-rate": {
        "current-value": 15
      },
      "max-retransmit": {
        "current-value": 3
      },
      "ack-timeout": {
        "current-value-decimal": "2.00"
      },
      "ack-random-factor": {
        "current-value-decimal": "1.50"
      },
      "ietf-dots-robust-trans:max-payloads": {
        "current-value": 10
      },
      "ietf-dots-robust-trans:non-max-retransmit": {
        "current-value": 3
      },
      "ietf-dots-robust-trans:non-timeout": {
```

```
            "current-value-decimal": "2.00"
          },
          "ietf-dots-robust-trans:non-probing-wait": {
            "current-value-decimal": "247.00"
          },
          "ietf-dots-robust-trans:non-partial-wait": {
            "current-value-decimal": "247.00"
          }
        },
        "idle-config": {
          "heartbeat-interval": {
            "current-value": 0
          },
          "max-retransmit": {
            "current-value": 3
          },
          "ack-timeout": {
            "current-value-decimal": "2.00"
          },
          "ack-random-factor": {
            "current-value-decimal": "1.50"
          },
          "ietf-dots-robust-trans:max-payloads": {
            "current-value": 15
          },
          "ietf-dots-robust-trans:non-max-retransmit": {
            "current-value": 3
          },
          "ietf-dots-robust-trans:non-timeout": {
            "current-value-decimal": "2.00"
          },
          "ietf-dots-robust-trans:non-probing-wait": {
            "current-value-decimal": "247.00"
          },
          "ietf-dots-robust-trans:non-partial-wait": {
            "current-value-decimal": "247.00"
          }
        }
      }
    }
```


Figure 1: Example of PUT to Convey the Configuration Parameters

4. DOTS Fast Block Transmission YANG Module

4.1. Tree Structure

This document defines the YANG module "ietf-dots-robust-trans" (Section 4), which has the following tree structure:

```
module: ietf-dots-robust-trans

  augment-structure /dots-signal:dots-signal/dots-signal:message-type
                    /dots-signal:signal-config
                    /dots-signal:mitigating-config:
    +-- max-payloads
    |  +-- (direction)?
    |  |  +--:(server-to-client-only)
    |  |     +-- max-value?   uint16
    |  |     +-- min-value?   uint16
    |  +-- current-value?     uint16
    +-- non-max-retransmit
    |  +-- (direction)?
    |  |  +--:(server-to-client-only)
    |  |     +-- max-value?   uint16
    |  |     +-- min-value?   uint16
    |  +-- current-value?     uint16
    +-- non-timeout
    |  +-- (direction)?
    |  |  +--:(server-to-client-only)
    |  |     +-- max-value-decimal?   decimal64
    |  |     +-- min-value-decimal?   decimal64
    |  +-- current-value-decimal?     decimal64
    +-- non-probing-wait
    |  +-- (direction)?
    |  |  +--:(server-to-client-only)
    |  |     +-- max-value-decimal?   decimal64
    |  |     +-- min-value-decimal?   decimal64
    |  +-- current-value-decimal?     decimal64
    +-- non-partial-wait:
       +-- (direction)?
       |  +--:(server-to-client-only)
       |     +-- max-value-decimal?   decimal64
       |     +-- min-value-decimal?   decimal64
       +-- current-value-decimal?     decimal64

  augment-structure /dots-signal:dots-signal/dots-signal:message-type
                    /dots-signal:signal-config/dots-signal:idle-config:
    +-- max-payloads
    |  +-- (direction)?
    |  |  +--:(server-to-client-only)
    |  |     +-- max-value?   uint16
    |  |     +-- min-value?   uint16
```

```
   |  +-- current-value?     uint16
   +-- non-max-retransmit
   |  +-- (direction)?
   |  |  +--:(server-to-client-only)
   |  |     +-- max-value?   uint16
   |  |     +-- min-value?   uint16
   |  +-- current-value?     uint16
   +-- non-timeout
   |  +-- (direction)?
   |  |  +--:(server-to-client-only)
   |  |     +-- max-value-decimal?   decimal64
   |  |     +-- min-value-decimal?   decimal64
   |  +-- current-value-decimal?     decimal64
   +-- non-probing-wait
   |  +-- (direction)?
   |  |  +--:(server-to-client-only)
   |  |     +-- max-value-decimal?   decimal64
   |  |     +-- min-value-decimal?   decimal64
   |  +-- current-value-decimal?     decimal64
   +-- non-partial-wait:
      +-- (direction)?
      |  +--:(server-to-client-only)
      |     +-- max-value-decimal?   decimal64
      |     +-- min-value-decimal?   decimal64
      +-- current-value-decimal?     decimal64
```

4.2. YANG/JSON Mapping Parameters to CBOR

The YANG/JSON mapping parameters to CBOR are listed in Table 2.

- Note: Implementers must check that the mapping output provided by their YANG-to-CBOR encoding schemes is aligned with the content of Table 2.

Parameter Name	YANG Type	CBOR Key	CBOR Major Type & Information	JSON Type
ietf-dots-robust-trans:max-payloads	container	TBA1	5 map	Object
ietf-dots-robust-trans:non-max-retransmit	container	TBA2	5 map	Object
ietf-dots-robust-trans:non-timeout	container	TBA3	5 map	Object
ietf-dots-robust-trans:non-probing-wait	container	TBA4	5 map	Object
ietf-dots-robust-trans:non-partial-wait	container	TBA5	5 map	Object

Table 2: YANG/JSON Mapping Parameters to CBOR

4.3. YANG Module

This module uses the data structure extension defined in [RFC8791].

```
<CODE BEGINS> file "ietf-dots-robust-trans@2020-05-04.yang"
module ietf-dots-robust-trans {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-dots-robust-trans";
  prefix dots-robust;

  import ietf-dots-signal-channel {
    prefix dots-signal;
    reference
      "RFC YYYY: Distributed Denial-of-Service Open Threat
                 Signaling (DOTS) Signal Channel Specification";
  }
  import ietf-yang-structure-ext {
    prefix sx;
    reference
      "RFC 8791: YANG Data Structure Extensions";
  }

  organization
    "IETF DDoS Open Threat Signaling (DOTS) Working Group";
```

```
     contact
       "WG Web:   <https://datatracker.ietf.org/wg/dots/>
        WG List:  <mailto:dots@ietf.org>

        Author:  Mohamed Boucadair
                 <mailto:mohamed.boucadair@orange.com>;

        Author:  Jon Shallow
                 <mailto:ietf-supjps@jpshallow.com>";
     description
       "This module contains YANG definitions for the configuration
        of parameters that can be negotiated between a DOTS client
        and a DOTS server for robust block transmission.

        Copyright (c) 2021 IETF Trust and the persons identified as
        authors of the code.  All rights reserved.

        Redistribution and use in source and binary forms, with or
        without modification, is permitted pursuant to, and subject
        to the license terms contained in, the Simplified BSD License
        set forth in Section 4.c of the IETF Trust's Legal Provisions
        Relating to IETF Documents
        (http://trustee.ietf.org/license-info).

        This version of this YANG module is part of RFC XXXX; see
        the RFC itself for full legal notices.";

     revision 2021-05-04 {
       description
         "Initial revision.";
       reference
         "RFC XXXX: Distributed Denial-of-Service Open Threat
                    Signaling (DOTS) Configuration Attributes
                    for Robust Block Transmission";
     }

     grouping robust-transmission-attributes {
       description
         "A set of DOTS signal channel session configuration
          that are negotiated between DOTS agents when
          making use of Q-Block1 and Q-Block2 Options.";
       container max-payloads {
         description
           "Indicates the maximum number of payloads that
            can be transmitted at any one time.";
         choice direction {
           description
             "Indicates the communication direction in which the
```

```
            data nodes can be included.";
          case server-to-client-only {
            description
              "These data nodes appear only in a mitigation message
               sent from the server to the client.";
            leaf max-value {
              type uint16;
              description
                "Maximum acceptable max-payloads value.";
            }
            leaf min-value {
              type uint16;
              description
                "Minimum acceptable max-payloads value.";
            }
          }
        }
        leaf current-value {
          type uint16;
          default "10";
          description
            "Current max-payloads value.";
        }
      }
      container non-max-retransmit {
        description
          "Indicates the the maximum number of times a
           request for the retransmission of missings payloads
           can occur without a response from the remote peer.";
        leaf max-value {
          type uint16;
          config false;
          description
            "Maximum acceptable non-max-retransmit value.";
        }
        leaf min-value {
          type uint16;
          config false;
          description
            "Minimum acceptable non-max-retransmit value.";
        }
        leaf current-value {
          type uint16;
          default "3";
          description
            "Current non-max-retransmit value.";
        }
      }
```

```
      container non-timeout {
        description
          "Indicates the maximum period of delay between
           sending sets of MAX_PAYLOADS payloads for the same
           body. By default, this parameter has the same value
           as ACK_TIMEOUT.";
        choice direction {
          description
            "Indicates the communication direction in which the
             data nodes can be included.";
          case server-to-client-only {
            description
              "These data nodes appear only in a mitigation message
               sent from the server to the client.";
            leaf max-value-decimal {
              type decimal64 {
                fraction-digits 2;
              }
              units "seconds";
              description
                "Maximum ack-timeout value.";
            }
            leaf min-value-decimal {
              type decimal64 {
                fraction-digits 2;
              }
              units "seconds";
              description
                "Minimum ack-timeout value.";
            }
          }
        }
        leaf current-value-decimal {
          type decimal64 {
            fraction-digits 2;
          }
          units "seconds";
          default "2";
          description
            "Current ack-timeout value.";
        }
      }
      container non-probing-wait {
        description
          "Is used to limit the potential wait needed calculated
           when using probing-rate.";
        choice direction {
          description
```

```
            "Indicates the communication direction in which the
             data nodes can be included.";
          case server-to-client-only {
            description
              "These data nodes appear only in a mitigation message
               sent from the server to the client.";
            leaf max-value-decimal {
              type decimal64 {
                fraction-digits 2;
              }
              units "seconds";
              description
                "Maximum non-probing-wait value.";
            }
            leaf min-value-decimal {
              type decimal64 {
                fraction-digits 2;
              }
              units "seconds";
              description
                "Minimum non-probing-wait value.";
            }
          }
        }
        leaf current-value-decimal {
          type decimal64 {
            fraction-digits 2;
          }
          units "seconds";
          default "247";
          description
            "Current non-probing-wait value.";
        }
      }
      container non-partial-wait {
        description
          "Is used for expiring partially received bodies.";
        choice direction {
          description
            "Indicates the communication direction in which the
             data nodes can be included.";
          case server-to-client-only {
            description
              "These data nodes appear only in a mitigation message
               sent from the server to the client.";
            leaf max-value-decimal {
              type decimal64 {
                fraction-digits 2;
```

```
              }
              units "seconds";
              description
                "Maximum non-partial-wait value.";
            }
            leaf min-value-decimal {
              type decimal64 {
                fraction-digits 2;
              }
              units "seconds";
              description
                "Minimum non-partial-wait value.";
            }
          }
        }
        leaf current-value-decimal {
          type decimal64 {
            fraction-digits 2;
          }
          units "seconds";
          default "247";
          description
            "Current non-partial-wait value.";
        }
      }
    }

    sx:augment-structure "/dots-signal:dots-signal"
                       + "/dots-signal:message-type"
                       + "/dots-signal:signal-config"
                       + "/dots-signal:mitigating-config" {
      description
        "Indicates DOTS configuration parameters to use for
         robust transmission when a mitigation is active.";
      uses robust-transmission-attributes;
    }
    sx:augment-structure "/dots-signal:dots-signal"
                       + "/dots-signal:message-type"
                       + "/dots-signal:signal-config"
                       + "/dots-signal:idle-config" {
      description
        "Indicates DOTS configuration parameters to use for
         robust transmission when no mitigation is active.";
      uses robust-transmission-attributes;
    }
  }
  <CODE ENDS>
```

5. IANA Considerations

5.1. DOTS Signal Channel CBOR Mappings Registry

This specification registers the following parameters in the IANA "DOTS Signal Channel CBOR Key Values" registry [Key-Map].

- Note to the RFC Editor: Please replace TBA1/TBA2/TBA3 with the CBOR keys that are assigned from the 128-255 range. Please update Table 2 accordingly.

Parameter Name	CBOR Key Value	CBOR Major Type	Change Controller	Specification Document(s)
ietf-dots-robust-trans: max-payloads	TBA1	5	IESG	[RFCXXXX]
ietf-dots-robust-trans: non-max-retransmit	TBA2	5	IESG	[RFCXXXX]
ietf-dots-robust-trans: non-timeout	TBA3	5	IESG	[RFCXXXX]
ietf-dots-robust-trans: non-probing-wait	TBA4	5	IESG	[RFCXXXX]
ietf-dots-robust-trans: non-partial-wait	TBA5	5	IESG	[RFCXXXX]

5.2. DOTS Signal Filtering Control YANG Module

This document requests IANA to register the following URI in the "ns" subregistry within the "IETF XML Registry" [RFC3688]:

```
URI: urn:ietf:params:xml:ns:yang:ietf-dots-robust-trans
Registrant Contact: The IESG.
XML: N/A; the requested URI is an XML namespace.
```

This document requests IANA to register the following YANG module in the "YANG Module Names" subregistry [RFC6020] within the "YANG Parameters" registry.

```
Name: ietf-dots-robust-trans
Namespace: urn:ietf:params:xml:ns:yang:ietf-dots-robust-trans
Maintained by IANA: N
Prefix: dots-robust
Reference: RFC XXXX
```

6. Security Considerations

The security considerations for the DOTS signal channel protocol are discussed in Section 11 of [I-D.ietf-dots-rfc8782-bis].

CoAP-specific security considerations are discussed in Section 11 of [I-D.ietf-core-new-block].

This document defines YANG data structures that are meant to be used as an abstract representation in DOTS signal channel messages. As such, the "ietf-dots-robust-trans" module does not introduce any new vulnerabilities beyond those specified above.

7. Acknowledgements

TBC

8. References

8.1. Normative References

[I-D.ietf-core-new-block]
Boucadair, M. and J. Shallow, "Constrained Application Protocol (CoAP) Block-Wise Transfer Options for Faster Transmission", draft-ietf-core-new-block-11 (work in progress), April 2021.

[I-D.ietf-dots-rfc8782-bis]
Boucadair, M., Shallow, J., and T. Reddy.K, "Distributed Denial-of-Service Open Threat Signaling (DOTS) Signal Channel Specification", draft-ietf-dots-rfc8782-bis-06 (work in progress), March 2021.

[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>.

[RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <https://www.rfc-editor.org/info/rfc3688>.

[RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <https://www.rfc-editor.org/info/rfc6020>.

[RFC6347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security Version 1.2", RFC 6347, DOI 10.17487/RFC6347, January 2012, <https://www.rfc-editor.org/info/rfc6347>.

[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>.

[RFC7959] Bormann, C. and Z. Shelby, Ed., "Block-Wise Transfers in the Constrained Application Protocol (CoAP)", RFC 7959, DOI 10.17487/RFC7959, August 2016, <https://www.rfc-editor.org/info/rfc7959>.

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

[RFC8323] Bormann, C., Lemay, S., Tschofenig, H., Hartke, K., Silverajan, B., and B. Raymor, Ed., "CoAP (Constrained Application Protocol) over TCP, TLS, and WebSockets", RFC 8323, DOI 10.17487/RFC8323, February 2018, <https://www.rfc-editor.org/info/rfc8323>.

[RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <https://www.rfc-editor.org/info/rfc8446>.

[RFC8791] Bierman, A., Bjoerklund, M., and K. Watsen, "YANG Data Structure Extensions", RFC 8791, DOI 10.17487/RFC8791, June 2020, <https://www.rfc-editor.org/info/rfc8791>.

8.2. Informative References

[I-D.ietf-dots-telemetry] Boucadair, M., Reddy, T., Doron, E., Chen, M., and J. Shallow, "Distributed Denial-of-Service Open Threat Signaling (DOTS) Telemetry", draft-ietf-dots-telemetry-15 (work in progress), December 2020.

[Key-Map] IANA, "DOTS Signal Channel CBOR Key Values", <https://www.iana.org/assignments/dots/dots.xhtml#dots-signal-channel-cbor-key-values>.

[RFC8340] Bjorklund, M. and L. Berger, Ed., "YANG Tree Diagrams", BCP 215, RFC 8340, DOI 10.17487/RFC8340, March 2018, <https://www.rfc-editor.org/info/rfc8340>.

[RFC8612] Mortensen, A., Reddy, T., and R. Moskowitz, "DDoS Open Threat Signaling (DOTS) Requirements", RFC 8612, DOI 10.17487/RFC8612, May 2019, <https://www.rfc-editor.org/info/rfc8612>.

Authors' Addresses

Mohamed Boucadair
Orange
Rennes 35000
France

Email: mohamed.boucadair@orange.com

Jon Shallow
United Kingdom

Email: supjps-ietf@jpshallow.com