DOTS J. Francois
INTERNET-DRAFT Inria
Intended Status: Standard Track A. Lahmadi
Expires: May 4, 2017 University of Lorraine - LORIA
Marco Davids
SIDN Labs
Giovane C. M. Moura
SIDN Labs
October 31, 2016

# IPv6 DOTS Signal Option
## draft-francois-dots-ipv6-signal-option-01

Abstract

This document specifies an optional fall-back opportunistic method that employs the IPv6 Hop-by-Hop options extension header type. It allows a DOTS client to send a signaling message over a congested network due to a DDoS attack by ''tagging'' bypassing outgoing IPv6 packets to reach a DOTS server or gateway.

Table of Contents

# 1  Introduction

## 1.1 Overview

A distributed denial-of-service (DDoS) attack aims at rendering machines or network resources unavailable. These attacks have grown in frequency, intensity and target diversity [I-D.draft-ietf-dots-requirements]. Moreover, several protocols have been utilized to amplify the intensity of the attacks [kuhrer2014exit],  peaking at several hundred gigabits per second.

The DOTS aims at defining a common and open protocol to signal DDoS attacks to facilitate a coordinated response to these attacks. This document specifies a signalling mechanism that instead of designing a new application-layer protocol, it utilizes the IPv6 Hop-by-Hop header [RFC2460]. This header has the advantage to be fully inspected by all network devices and it is the first header in IPv6 extension headers [RFC7045].

The new option containing the attributes of the signalling message is included in an opportunistic way in available IPv6 packets leaving a network element until the message reaches a DOTS server. It thus constitutes an additional signalling channel but MUST NOT replace the original signalling channel used between DOTS client and servers as the one defined in [I-D.draft-reddy-dots-signal-channel]. The DOTS client will thus embed the signalling attributes into outgoing IPv6 packets not necessarily going to the DOTS server. Intermediate routers receiving such a packet will examine it and embed the same information into other IPv6 packets. domain in this opportunistic way to increase the probability that such a packet will be finally forwarded to a DOTS gateway or Server, but also in controlled way to avoid that the mechanism is exploited for a malicious purposes.

Only the Hop-by-Hop options header allows such behavior and using Destination options header is not enough to make the DOTS signal going through the network in an opportunistic way. Each network element recognizing this new option will select the best fitted IPv6 packets to deliver the signal to the DOTS server or gateway. For this reason the Hop-by-Hop header option is essential to make such behavior compared to other existing IPv6 extension headers [RFC6564].

## 1.2 Motivation

The traffic generated by a DDoS can be characterized according to various parameters, such as the layer (IP/ICMP or application), maximum and instant throughput, among others. Regardless its nature, we assume that for most cases, a DOTS client will be able to signal back one or few messages, during the attack, to the DOTS phase.

We have the same behavior in other DDoS attacks. For instance, on November 30th and December 1st, 2015, the Root DNS system was hit by an application layer (DNS) attack [rootops-ddos]. Each one of the 13 root server letters (A--M) was hit by attacks peaking at 5 million queries per second. By utilizing the RIPE Atlas DNSMON infrastructure, we can see that during the DDoS attacks, most of the root server letters remained reachable and able to respond to the DNS request sent by the probes employed by the DNSMON [ripe-dnsmon-ddos]. Few letters, however, had a packet loss rate of more than 99%. The DNSMON probes, however, experience mostly delays in their DNS requests instead.

Our signalling mechanism operates in an opportunistic way it is designed for DDoS as the ones on the Root DNS system.

### 1.3 Terminology

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 [RFC2119].

The terms DOTS client, DOTS server, DOTS gateway, DOTS agents, signal channel, DOTS signal and DOTS signal refers to the terminology introduced in [I-D.draft-ietf-dots-requirements].

The following terms are introduced:

Opportunistic DOTS signal:
an IPv6 packet containing the signalling attributes of an attack within the Hop-by-Hop extension header. The purpose is the same as the DOTS signal. It is used to request help for mitigating the attack.

DOTS opportunistic-capable router:
a router with the capacity to decode the opportunistic DOTS signal and re-embed such an information in other IPv6 packets.

All DOTS opportunistic-capable agents are defined as the DOTS agents supporting the opportunistic DOTS signal processing.

## 2. Opportunistic DOTS signal option

The goal is to provide an efficient mechanism where nodes in a IPv6 network facing a DDoS attack can deliver a DOTS signal message sent by a DOTS client to the DOTS server. The specified mechanism does not generate transport packets to carry the DOST signal message but it only relies on existing IPv6 packets in the

network to include inside them a hop-by-hop extension header which contains an encoded DOTS signal message. The solution defines a new IPv6 Hop-by-Hop header option with the semantic that the network node SHOULD include the option content within one or multiple outgoing IPv6 packets available in that network node.

### 2.1 Hop-by-Hop option encoding

According to [RFC2460], options encoded into the IPv6 Hop-by-Hop header are formatted as Type-Length-Values (TLVs). The option for opportunistic DOTS signal is thus defined as follows:

```
0               7               15              22              31
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Option type  |Option Data Len|    DOTS Signal Attribute[1]   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| DOTS Signal Attribute[2] |  ...  | DOTS Signal Attribute[n]   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The first byte defines the Hop-by-Hop Option type number allocated to the DOTS opportunistic signalling. This number is not yet fixed but the first three bits MUST be set to 0. The first two zero bits indicate that routers which cannot handle the DOTS signal option will continue to process other options. The third 0 bit means that the option processing will not change the packet's final destination [RFC2460].

The second byte contains the length of the option content. The content of the DOTS Signal option is a variable-length field that contains one or more type-length-values (TLV) encoded DOTS signal attributes, and has the following format:

```
0               7               15
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Attr Type   | Attr Data Len | Attr Data ...    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The Attr Type is 8-bit identifier of a DOTS signal attribute.

The Attr Data Len is 8-bit unsigned integer which is the length of Attr Data in bytes.

The Attr Data is variable-length field that contains the data of the attribute.

Since using TLVs in Hop-by-Hop options is known to be a factor of attacks [I-D.draft-krishnan-ipv6-hopbyhop], DOTS attributes are encoded with fixed length when possible. Another issue is the size of IPv6 datagram is constrained by the path MTU (Maximum Transmission Unit) to avoid fragmentation. There are several options to overcome this issue. It is RECOMMENDED that DOTS opportunistic signal NOT embedded is such a requirement is not satisfied.

## 2.2 DOTS signal Option attributes

The first attribute embedded into the opportunistic DOTS signal is a TTL (Time-to-Live) field which indicates the maximum number of retransmission of the signal into another IPv6 packets until it MUST be discarded. Remaining attributes are similar to the header fields described in [I-D.draft-reddy-dots-signal-channel] (section 5.1.1) used to convey a DOTS signal through a HTTP POST.

The sequence of attributes to be inserted within the header MUST start with fixed-length attributes which are defined in the following order:

TTL: Time-to-Live. This is a mandatory attribute encoded in one byte.

Flags: one byte is reserved for flags.
The first bit indicates the type of the IP address of the host: 0 for IPv4, 1 for IPv6. The second bit indicate if the protocol to use is TCP (1) or UDP (0). The third bit indicates if the message is signed The remaining bit are not used yet.

host: the IP address of the DOTS server where the signal option SHOULD be delivered. Depending on the flags, this field is encoded in 4 or 16 bytes.

port: the listening port of the DOTS server.
It is encoded in 2 bytes.

The remaining attributes MUST be TLV encoded, and they are defined in the following order:

policy-id: defined in [I-D.draft-reddy-dots-signal-channel].

target-ip: defined in [I-D.draft-reddy-dots-signal-channel].
However, each address or prefix is encoded in its own TLV element. The distinction between IPv4 and IPv6 is done over the length of the value.

target-port: defined in [I-D.draft-reddy-dots-signal-channel].

However, each target port is encoded in its own TLV element.

target-protocol: defined in [I-D.draft-reddy-dots-signal-channel]. However each target protocol is encoded in its own TLV element.

lifetime (lt): lifetime of the mitigation request defined in [I-D.draft-reddy-dots-signal-channel].

The encoded attributes MUST be included in the option header in the order defined above.

The following table provides the value of types that are used by the TLV encoded attributes.

| Attribute type | value |
|---|---|
| policy-id | 0 |
| target-ip | 1 |
| target-port | 2 |
| target-protocol | 3 |
| lifetime | 4 |

## 2.3 Example

Following is an example of an encoded Hop-by-Hop Option header to signal that a web service is under attack.

```
0               7              15              22              31
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Next header   | Hdr Ext Len=6 |    TTL=128     | Flags=IPv4,TCP|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         host=192.0.2.1                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            port=443           | A. type=policy| Att Data Len=2|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              143              |  Attr. type=ip| Att Data Len=4|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           192.0.2.20                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Attr. type=ip |Att Data Len=16|                                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                                 +
|                                                                 |
+                                                                 +
```

```
   |                                                               |
   +                                                               +
   |                        2001:db8:6401::1                       |
   +                               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                               |Attr. type=port| Att Data Len=2|
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |              8080             |Attr. type=port| Att Data Len=2|
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |              443              |Attr.type=proto| Att Data Len=2|
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |              TCP              | Attr. type=lt | Att Data Len=2|
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |              600              |       1       | Opt Data Len=0|
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

In the previous example, the message is not signed and terminates with padding. If it is the case, then the signature MUST BE added at the end such that the integrity and authenticity can be checked by the DOTS server or gateway. The TTL attributes MUST be excluded from the signature calculation (see section 6).

## 3 Option Processing

### 3.2 Opportunistic DOTS signal initialization by a DOTS client

When a DOTS client needs to inform the DOTS server that it is under attack, it firstly makes a connection attempt and applies the mechanisms described in [I-D.draft-reddy-dots-signal-channel].

In addition, it MAY activates an opportunistic mechanism to include the Hop-by-Hop header option specified in this document in one or multiple available IPv6 packets leaving the node. Because the DOTS client location is independent of the signalling, it can be positioned in a part of the network where there is no passing-by traffic which can serve for opportunistic signalling. DOTS client MAY also create and emit IPv6 datagrams without payload but with the signal encoded in the Hop-by-Hop option header.

Otherwise, the selection of packets has to be configured  a priori. The configuration is composed of a sequence of rules defined in a hierarchical order such that they are triggered in a sequential manner.

The selection of packets has to be configured  a priori. The configuration is composed of a sequence of rules defined in a hierarchical order such that they are triggered in a sequential

manner.

Each rule is defined by:

- o a set of filters over the IPv6 packet headers. Only packets matching those filters are selected for opportunistic signalling. For instance, only packets heading to a given subnetwork or to specific address close to a DOTS server can be selected to increase the chance to reach the latter.

- o a ratio to select only a proportion of packets matching the filters in order to limit the induced overhead of the opportunistic signalling.

- o a timeout until the rule is active and selected IPv6 packets embed the DOTS opportunistic signal.

The objective is to apply each ordered rule after another according to their timeouts. The first rule is triggered immediately after the opportunistic signalling is activated.

In all cases (embedding information into an existing packet or creating an new packet with no payload), the client MUST avoid fragmentation.

Although the definition of rules MUST be configured by the user. It is RECOMMENDED to order them inversely related to the number of packets that would be selected. This can be approximated regarding the definition of filters. The core idea is to benefit from the first instants of the attack before losing connectivity by using a maximum number of outgoing packets to include the DOTS signalling option. It is thus RECOMMENDED to define the first as matching all IPv6 packets with a ratio equals one to rapidly disseminate the information but with a short timeout to limit the implied overhead.

Here is the an example of rules:

1: all outgoing IPv6 packets with a 10 second timeout
2: all outgoing IPv6 packets with a ratio of 10% and a 1 minute timeout
3: all outgoing multicast IPv6 packets with a ratio of 10% and a 1 minute timeout
4: all outgoing anycast IPv6 packets with a ratio of 10% and a 5 minute timeout
5: all outgoing IPv6 packets heading to the DOTS server with a ratio of 100% and a one hour timeout

### 3.2 Processing by a non DOTS opportunistic-capable router

When receiving an opportunistic DOTS signal encoded in a IPv6 packet, a non DOT opportunistic capable router simply skips the Hop-by-Hop option and continue the normal processing of the IPv6 packet because the option type MUST start with three zero bits.

### 3.3 Processing by a DOTS opportunistic-capable router

A DOTS opportunistic-capable router MUST store DOTS signalling information whose it is aware of. If a router processes an IPv6 DOTS opportunistic signal and supports this option, it first checks if it has already stored the associated information. In that case, the router simply skips the option and continues the normal processing otherwise it stores the encoded information in order to embed it again in other IPv6 packets similarly to the DOTS client. Hence, a set of rules are also defined in advance and are triggered upon the reception of a new opportunistic DOTS signal. Once all rule have been applied, signalling information MUST be discarded by the router. When embedding the information into other IPv6 packets, the router MUST decrease the TTL by one since opportunistic signalling does not prevent loops in the dissemination of signalling.

### 3.4 Processing by a DOTS opportunistic-capable gateway

If a DOTS gateway has DOTS capabilities, it will apply the same strategy as a DOTS client by making attempts of direct connections to the DOST server and in addition it inserts the Hop-by-Hop header DOTS signalling option in leaving IPv6 packets using the strategy specified above.

### 3.5 Processing by a DOTS opportunistic-capable server

When the IP layer of the host where the DOTS server is running receives an IPv6 packet carrying a Hop-by-Hop DOTS signal option header it MUST extracts the content of the option and provides the attributes data to the server program.

## 4 Deployment considerations

This mechanism will be potentially used by networks with IPv6 capable elements and requires that of IPv6 traffic exist in the network during the attack. The existing IPv6 traffic to be used could be of any type from management or user levels. It is also important to emphasize that while our mechanism utilizes an IPv6 header field, it can also be used to signal IPv4 attacks as well - given that the network devices are dual stacked.

IPv6 extension headers are often rate-limited or dropped entirely [HBH-HEADER]. To be able to use the mechanism specified in this document, network operators need to avoid discarding packets or ignoring the processing of the hop-by-hop option on their deployed network elements. However, instead of dropping or ignoring packets with hop-by-hop option carrying DOTS signal, they need to assign these packets to slow forwarding path, and be processed by the router's CPU. This behavior will not affect the performance of the network devices since the network is already facing a DDoS attack and fast forwarding paths are saturated by the attacker traffic.

If the DOTS server, gateway and the client are located in the same administrative domain, marking the IPv6 packets with the proposed hop-by-hop header option could be done in a straight forward way, while considering that an agreement exists inside the domain to avoid dropping or rate limiting of IPv6 extension headers as described above. The proposed mechanism becomes less practical and difficult to deploy when the DOST server is running on the Internet. In such scenario, the mechanism could be used in the intra-domain part to deliver the hop-by-hop option carrying the DOTS signal until it reaches a DOTS gateway located in the same domain as the client, then the gateway will apply mechanisms provided by the DOTS transport protocol [I-D.draft-reddy-dots-signal-channel] to inform the server running on Internet about the attack. This deployment scenario requires that at least one DOTS gateway is deployed in the same domain than the DOTS client.

## 5 Impact on existing IP layer implementations

For this option to be applicable within an IP system, it requires modifications to existing IP layer implementation. At DOTS capable nodes (client, gateway and server), it requires a service interface used by upper-layer protocols and application programs to ask the IP layer to insert and listen to the Hop-by-Hop header option in IPv6 packets with the content and strategies described in Section 3. A DOTS client invokes the service interface to insert the option, A DOTS gateway invokes the service interface for listening and inserting the option, and finally a DOTS server only invokes the service interface to listen to the DOTS signalling option.

Intermediate nodes (routers or middle boxes) IP layer needs to be extended to perform processing of the new Hop-by-Hop header option as described in Section 3. They mainly parse the first host attribute of the option and make a selection of a leaving IPv6 packet where the option will be inserted.

Every node inserting the new proposed Hop-by-Hop option SHOULD only select IPv6 packets with enough left space to avoid fragmentation.

## 6 Security Considerations

Any IPv6 header option could be used by an attacker to create an attack on the routers and intermediate boxes that process packets containing the option. The proposed IPv6 option in this document MAY be abused by an attacker to create a covert channel at the IP layer where data is hidden inside the content of the option [RFC6564]. However, this attack is not specific to the proposed option and it is a known issue of IPv6 header extensions and options. The option MAY also be used by an attacker to forge or modify opportunistic DOTS signal leading to trigger additional processing on intermediate nodes and DOTS servers.

However the proposed option should be only initiated by a DOTS client and information embedded in new IPv6 messages by opportunistic DOTS capable routers. Defining proper policies to filter all messages with this option set and originated from other nodes would limit security issues since these DOTS opportunistic-capable agents SHOULD be trustworthy.

In addition, the message MAY be signed using techniques to enforce authenticity and integrity over the opportunistic DOTS signal channel. The signalling message specification includes a flag to indicate if the message is signed by the choice of the signature algorithm is let to the users. This signature has to be computed by the DOTS opportunistic-capable client and checked by the DOTS opportunistic-capable gateway or router. Hence, intermediate routers MUST NOT modify the message and its signature except the TTL, which so has not be considered during the signature computation.

Assuming a compromised router, the attacker could nevertheless replay the message or increase the TTL but thanks to the unique policy-id all intermediate-DOTS capable router will drop such messages and thus limiting their forwarding in the network.

Besides, an attacker can also listen  opportunistic DOTS signals to monitor the impact of its own attack. These considerations are not specific to the proposed option and supposes that the attacker is able to compromise intermediate routers.

## 7 IANA Considerations

This draft defines a new IPv6 [RFC2460] hop-by-hop option. This requires an IANA RFC3692-style update of: http://www.iana.org/assignments/ipv6-parameters/ipv6-parameters.xhtml and ultimately the assignment of a new hop-by-hop option according to the guidelines described in [RFC5237].

# 7 References

## 7.1 Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, DOI 10.17487/RFC2460, December 1998, <http://www.rfc-editor.org/info/rfc2460>.

[RFC6564] Krishnan, S., Woodyatt, J., Kline, E., Hoagland, J., and M. Bhatia, "A Uniform Format for IPv6 Extension Headers", RFC 6564, DOI 10.17487/RFC6564, April 2012, <http://www.rfc-editor.org/info/rfc6564>.

[RFC7045] Carpenter, B. and S. Jiang, "Transmission and Processing of IPv6 Extension Headers", RFC 7045, DOI 10.17487/RFC7045, December 2013, <http://www.rfc-editor.org/info/rfc7045>.

## 7.2 Informative References

[I-D.draft-ietf-dots-requirements]
A. Mortensen., R. Moskowitz., and T. Reddy., "DDoS Open Threat Signaling Requirements", draft-ietf-dots-requirements-00 (work in progress), October 2015.

[kuhrer2014exit]
Kuhrer, Marc and Hupperich, Thomas and Rossow, Christian and Holz, Thorsten. Exit from Hell? Reducing the Impact of Amplification DDoS Attacks. In: 23rd USENIX Security Symposium (USENIX Security 14).

[I-D.draft-reddy-dots-transport]
T. Reddy., D. Wing., P. Patil., M. Geller., M. Boucadair.,and R. Moskowitz., "Co-operative DDoS Mitigation", draft-reddy-dots-transport-03 (work in progress), March 2016.

[rootops-ddos]
rootops.: Events of 2015-11-30. Online: http://root-servers.org/news/events-of-20151130.txt

[ripe-dnsmon-ddos]
RIPE NCC DNS Monitoring Service (DNSMON). Online: https://atlas.ripe.net/dnsmon/

[HBH-HEADER]
Baker, F., "IPv6 Hop-by-Hop Options Extension Header",Work in Progress, draft-ietf-6man-hbh-header-handling-03,March 2016.

Acknowledgements

This work is partly funded by FLAMINGO, a Network of Excellence project (ICT-318488) supported by the European Commission under its Seventh Framework Programme.

## Authors' Addresses

Jerome Francois
Inria
615 rue du Jardin Botanique
54600 Villers-les-Nancy
France

Phone: +33 3 83 59 30 66
EMail: jerome.francois@inria.fr

Abdelkader Lahmadi
University of Lorraine - LORIA
615 rue du Jardin Botanique
54600 Villers-les-Nancy
France

Phone: +33 3 83 59 30 00
Email: Abdelkader.Lahmadi@loria.fr

Marco Davids
SIDN Labs
Meander 501
6825 MD Arnhem
The Netherlands

Email: marco.davids@sidn.nl

Giovane C. M. Moura
SIDN Labs
Meander 501
6825 MD Arnhem
The Netherlands

Email: giovane.moura@sidn.nl