IPv6 DOTS Signal Option
draft-francois-dots-ipv6-signal-option-00

Jérôme François, Inria, jerome.francois@inria.fr
Abdelkader Lahmadi, Université de Lorraine,
abdelkader.lahmadi@loria.fr
Marco Davids, SIDN Labs, marco.davids@sidn.nl
Giovane Moura, SIDN Labs, giovane.moura@sidn.nl

IETF 96 Berlin
Key idea

- Regular paths for delivering DOTS signals might be also affected by the DDoS → Adding an auxiliary mechanism for signaling (does not substitute)
  - Embed the information into traffic being able to be routed
  - without impact its original routing or content being processed but end-hosts
- Use IPv6 Hop-by-Hop Option Header [RFC2460]
  - signaling information is embedded into outgoing IPv6 packets
  - in an opportunistic manner (not all packets, not only those outgoing to the DOTS server... but some well chosen)
  - the DOTS client initiate this process, intermediate capable routers can store the information and embed it into other packets
  - once such a packet reaches the server of the gateway
- Intended for the intra-domain use case
Option encoding

- TLV-encoded in the IPv6 header

```
+-------------------------------+-------------------+-------------------------+
| Option type | Option Data Len | DOTS Signal Attribute[1] |
+-------------------------------+-------------------+-------------------------+
| DOTS Signal Attribute[2] | ... | DOTS Signal Attribute[n] |
+-------------------------------+-------------------+-------------------------+
```

- DOTS attributes
  - from draft-reddy-dots-transport
  - + a specific TTL value to avoid embedding the information into new packets indefinitely
  - + address and port of the DOTS server to reach (+ flags)
  - a mix between TLV and fixed-length field

```
| Attribute type | value |
+---------------+-------|
| policy-id     | 0     |
| target-ip     | 1     |
| target-port   | 2     |
| target-protocol | 3   |
| lifetime      | 4     |
+---------------+-------|
```
Option processing (Example)

- The client tries to initialize the regular signaling.
- The client initializes the Hop-by-hop based signaling.
- Outgoing IPv6 are selected for marking.
- Non-capable routers ignore the option and forward the packets.
- The client continues the marking.
- When arriving at capable agents (gateways, routers), embedded information is stored.
- The gateway tries to initialize the regular signaling.
- The capable router having saved the information embeds it again in other IPv6 packets.

Diagram:
- C: Client
- S: Server
- G: Gateway
- R: Capable Router
- ∅: Non-capable Router

⇒ Forwarded IPv6 packet
Option processing (Example)

- The client tries to initialize the regular signaling
- The client initializes the Hop-by-hop based signaling → outgoing IPv6 are selected for *marking*

- Non-capable routers ignore the option and forward the packets
- The client continues the marking
- When arriving at capable agents (gateways, routers), embedded information is stored
- The gateway tries to initialize the regular signaling
- The capable router having saved the information embeds it again in other IPv6 packets

![Diagram](image-url)

- **C** Client
- **S** Server
- **G** Gateway
- **R** Capable Router
- **R** Non-capable Router
  - Forwarded IPv6 packet

*Hop-by-hop signaling*

*Regular signaling*
Option processing (Example)

- The client tries to initialize the regular signaling
- The client initializes the Hop-by-hop based signaling → outgoing IPv6 are selected for *marking*
- Non-capable routers ignore the option and forward the packets
- The client continues the *marking*

![Diagram showing client, server, gateway, and routers with arrows indicating forwarding of IPv6 packets.]

- **C**: Client
- **S**: Server
- **G**: Gateway
- **R**: Capable Router
- ****: Non-capable Router
- ****: Forwarded IPv6 packet
Option processing (Example)

- The client tries to initialize the regular signaling
- The client initializes the Hop-by-hop based signaling → outgoing IPv6 are selected for *marking*
- Non-capable routers ignore the option and forward the packets
- The client continues the *marking*
- When arriving at capable agents (gateways, routers), embedded information is stored

![Diagram](image)

- Client (C)
- Server (S)
- Gateway (G)
- Capable Router (R)
- Non-capable Router (grey R)
- Forwarded IPv6 packet (red arrows)
Option processing (Example)

- The client tries to initialize the regular signaling
- The client initializes the Hop-by-hop based signaling → outgoing IPv6 are selected for *marking*
- Non-capable routers ignore the option and forward the packets
- The client continues the *marking*
- When arriving at capable agents (gateways, routers), embedded information is stored
- The gateway tries to initialize the regular signaling
- The capable router having saved the information embeds it again in other IPv6 packets

![Diagram of option processing](image)
Option processing

- Selection of packets is rule-based to only consider a subset
- A sequence of rules where each is defined by
  - 1st level: a filter on IPv6 header to be matched
  - 2nd level: a ratio of previously matched packets
  - + a timeout
- When a rule expires (timeout) the next one is applied
- Rules are manually configured
- Recommendation: first rules should select more packets
  (taking benefit of the first instant before losing connectivity)

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
Deployment considerations

- IPv6 extension headers are often rate-limited or dropped entirely
  - One reason is the overhead of processing
  - Our proposed option is only used under a DDoS attack and performance might be so already degraded
  - Practical for an operator to allow such an option within its own network but more difficult in the inter-domain cases with non-cooperative networks in between

- Modification to IP layers implementations
  - capable routers: need to extract store and embed signaling information
  - clients: need to create the specific option header to be embedded then
  - servers and gateways: all DOTS signaling information contained in IPv6 headers has to transmitted to the application layer
Security considerations

- Forged option headers from non legitimate sources to entail additional processing on routers
  - Source-based filtering to discard those since we know which sources can emit such IPv6 packets
  - The option can be signed by the clients and verified by the servers and gateways (intermediate capable routers do not for efficiency reason → exclude TTL from the signature calculation)
- Replay attack from a compromised router to inject more packets
  - Thanks to the id and TTL, other agents will not consider the header
Next steps

- Receive comments
- Improve the document