

DOTS

Interop Report

IETF 102 Hackathon

Kaname Nishizuka/NTT Communications

Jon Shallow/NCC Group

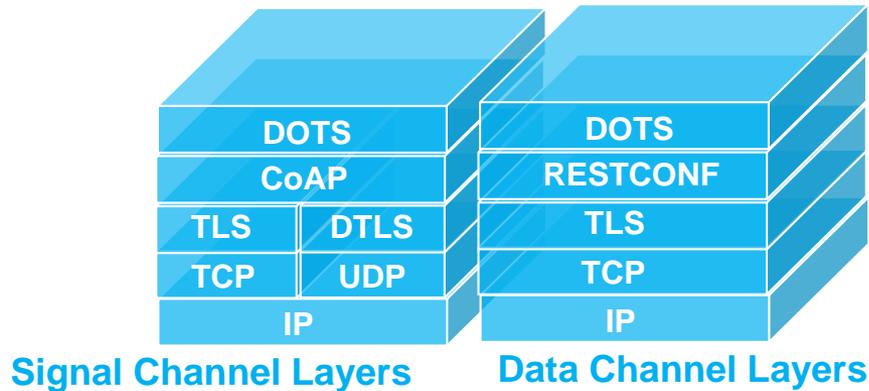
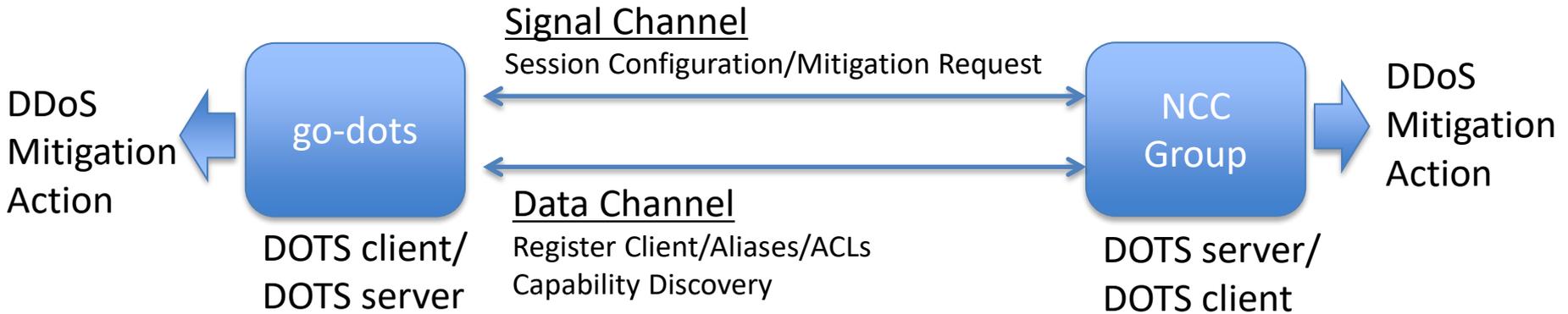
Liang Xia/Huawei

Hackathon Plan

Hackathon	What we did	Signal Channel	Data Channel	Participants
IETF99	Implementation of OSS (go-dots)	✓		NTT
IETF100	1st Interoperability Test	✓		NTT, NCC Group, Huawei
IETF101	2nd Interoperability Test	✓		NTT, NCC Group, Huawei
IETF102	3rd Interoperability Test(*) - The first data-channel interop	✓	✓	NTT, NCC Group, Huawei

(*) NTT & NCC Group did two internal interop tests before the IETF102 hackathon in preparation.

Interop test settings



Interop Results

draft version: draft-ietf-dots-signal-channel-20(latest)							✓ IETF101		
draft version: draft-ietf-dots-data-channel-16(latest)							✓ IETF102		
# Interoperability Testing Results									
# DOTS Signal Channel Features implementation status				client: nttdots	client: ncc				
#	feature	ncc*	nttdots*	arbor	server: ncc	server: nttdots	issue to WG	memo	
1	Session Configuration	✓	✓		✓ *1	✓		*1 unspecified value treatment->fixed	
2	Mitigation Request	✓	✓		✓	▲ *2	#1	*2 has bug on invalid prefix procession (nttdots)-> fixed	
3	CoAP Ping	✓	✓		✓	✓			
4-1	Observe (Mitigation)	✓	✓		✓	✓			
4-2	Observe (Session Config)	✓	✓		▲ *3	▲ *5		*3 no notifications->fixed *5 - get back multiple status = 6 when times o	
5	Efficacy Update	✓	✓		▲ *4	✓		*4 no lifetime refresh->fixed	
6	Request Conflicion Handling	✓							
7	Conflicion Notify								
8	Deadman's Trigger								
9	Gateway Function	✓							
10	Redirection	✓							
11	Happy Eyeballs	✓							
* supporting both PKI and PSK		Huawei's implementation is based on nttdots							
# DOTS Data Channel Features implementation status				client: nttdots	client: ncc				
#	feature	ncc	nttdots	arbor	server: ncc	server: nttdots	issue to WG		
1	Register DOTS clients	✓	✓		✓	✓			
2	Register Alias	✓	✓		✓	✓	#1,#3,#4		
3	Register Filtering Rules	✓	✓		✓	✓	#2,#4		
4	Capabilities	✓	✓		✓	✓			
5	Gateway Function	✓							

Data-channel - Evaluated Features

- Register DOTS clients
- Register Alias
- Register Filtering Rules
- Capabilities



# DOTS Data Channel Features implementation status					client: nttdots	client: ncc	
#	feature	ncc	nttdots	arbor	server: ncc	server: nttdots	issue to WG
1	Register DOTS clients	✓	✓		✓	✓	
2	Register Alias	✓	✓		✓	✓	#1,#3,#4
3	Register Filtering Rules	✓	✓		✓	✓	#2,#4
4	Capabilities	✓	✓		✓	✓	
5	Gateway Function	✓					

Only minor issues are found (except for fragment bit in ACL)

- Message exchange over RESTCONF/TLS is stable
- We fixed several bugs about message format on each side
- Among them, 4 issues with the draft data channel were raised during the hackathon
 - <https://github.com/boucadair/draft-ietf-dots-data-channel/issues>

[Fixed] #1: Clarify ambiguity (about the text 6.1 Create Alias)

6.1. Create Aliases

In POST requests, at least one of the 'target-prefix', 'target-fqdn', or 'target-uri' attributes MUST be present. DOTS agents can safely ignore Vendor-Specific parameters they don't understand.

should read (as it is true for both cases)

In POST or PUT requests, at least one of the 'target-prefix', 'target-fqdn', or 'target-uri' attributes MUST be present. DOTS agents can safely ignore Vendor-Specific parameters they don't understand.

[Fixed] #3: /aliases?content=config
includes pending-lifetime in the response

- “pending-lifetime” is defined as “config false”
- so, it should not be included in the response if GET to retrieve an alias is with content=config
- The text was updated to use "content=all"

[Fixed] #4: What gets returned for ?content=nonconfig

- RFC8040 – RESTCONF B.3.1 “content” Parameter
 - To retrieve only the non-configuration child resources, the “content” parameter is set to “nonconfig”. Note that configuration ancestors (if any) and list key leafs (if any) are also returned.
- “name” of ACL/alias should be also returned because it is key leaf
- [implementation guideline] see RFC8040

#2: fragment bits representation

[Text below is the now corrected version in draft]

- ipv4

- <https://tools.ietf.org/html/rfc7951#section-6.5>
The "bits" Type

```
"ipv4": {  
  "flags": "more"  
}
```

- ipv6

- <https://tools.ietf.org/html/rfc7951#section-6.9>
The "empty" Type

```
"ipv6": {  
  "fragment": [null]  
}
```

#2: fragment bits representation

- The current draft says:

```
{
  "name": "drop-last-fragment",
  "matches": {
    "ipv4": {
      "flags": ""
    }
  },
  "actions": {
    "forwarding": "drop"
  }
}
```

- Can it be used for representation that only ‘more’ bit is 0?
 - undefined bits are checked that they are 0 (reserved and (don’t)fragment) (RFC7951 6.5. The "bits" Type‘)
- How best to represent a fragment filtering rule with netmod-acl or enhancement.
 - BGP flowspec(RFC5575) like representation for fragments?

Signal-channel - Evaluated Features

- Session Configuration
- Mitigation Request
- CoAP Ping
- Observe (Mitigation)
- Observe (Session Config)
- Efficacy Update
- Request Conflict Handling
- Conflict Notify
- Deadman's Trigger
- Gateway Function
- Redirection
- Happy Eyeballs



Core Specifications

Additional Specifications

Bugs with new features

no significant issue on the draft

- Several bugs are found on both sides but fixed during or after the hackathon
- 2 issues are raised but already fixed

# DOTS Signal Channel Features implementation status					# Interoperability Testing Results			
#	feature	ncc*	nttdots*	arbor	client: nttdots	client: ncc	issue to WG	memo
					server: ncc	server: nttdots		
1	Session Configuration	✓	✓		✓ *1	✓	#2	*1 unspecified value treatment->fixed
2	Mitigation Request	✓	✓		✓	▲ *2	#1	*2 has bug on invalid prefix procession (nttdots)-> fixed
3	CoAP Ping	✓	✓		✓	✓		
4-1	Observe (Mitigation)	✓	✓		✓	✓		
4-2	Observe (Session Config)	✓	✓		▲ *3	▲ *5		*3 no notifications->fixed *5 - get back multiple status = 6 when times
5	Efficacy Update	✓	✓		▲ *4	✓		*4 no lifetime refresh->fixed
6	Request Conflicion Handling	✓						
7	Conflicion Notify							
8	Deadman's Trigger							
9	Gateway Function	✓						
10	Redirection	✓						
11	Happy Eyeballs	✓						

▲ represents the buggy status at the hackathon.
but it is now fixed

[Fixed] #1:Missing mandatory lifetime in example (Fig 7,8)

- lifetime is mandatory attribute in mitigation request[PUT]
- fixed editorial nits

[Fixed] #2:Signal Channel 4.5.3

Configuration Freshness and Notifications

- The DOTS server needs to be able to update Signal Channel configuration parameters
- DOTS client was doing a PUT to refresh configuration – overwriting any DOTS server update
- Solution was for DOTS client to do a GET to refresh configuration within the Max-Age time
 - Draft signal channel now updated

Takeaway

- Successful interop tests are continuing
- Core specification of DOTS is now mature enough
 - both signal-channel and data-channel
 - several issues are found (not so significant) during the hackathon
 - they've been sorted out and fixed
- Implementations (more implementations are welcome)
 - One OSS
 - <https://github.com/nttdots/go-dots>
 - One proprietary
 - NCC Group

Questions
Or
Comments?

Thank You