# DOTS
# Interop test report

IETF 103 Hackathon

Kaname Nishizuka/NTT Communications
Jon Shallow/NCC Group
Liang Xia/Huawei

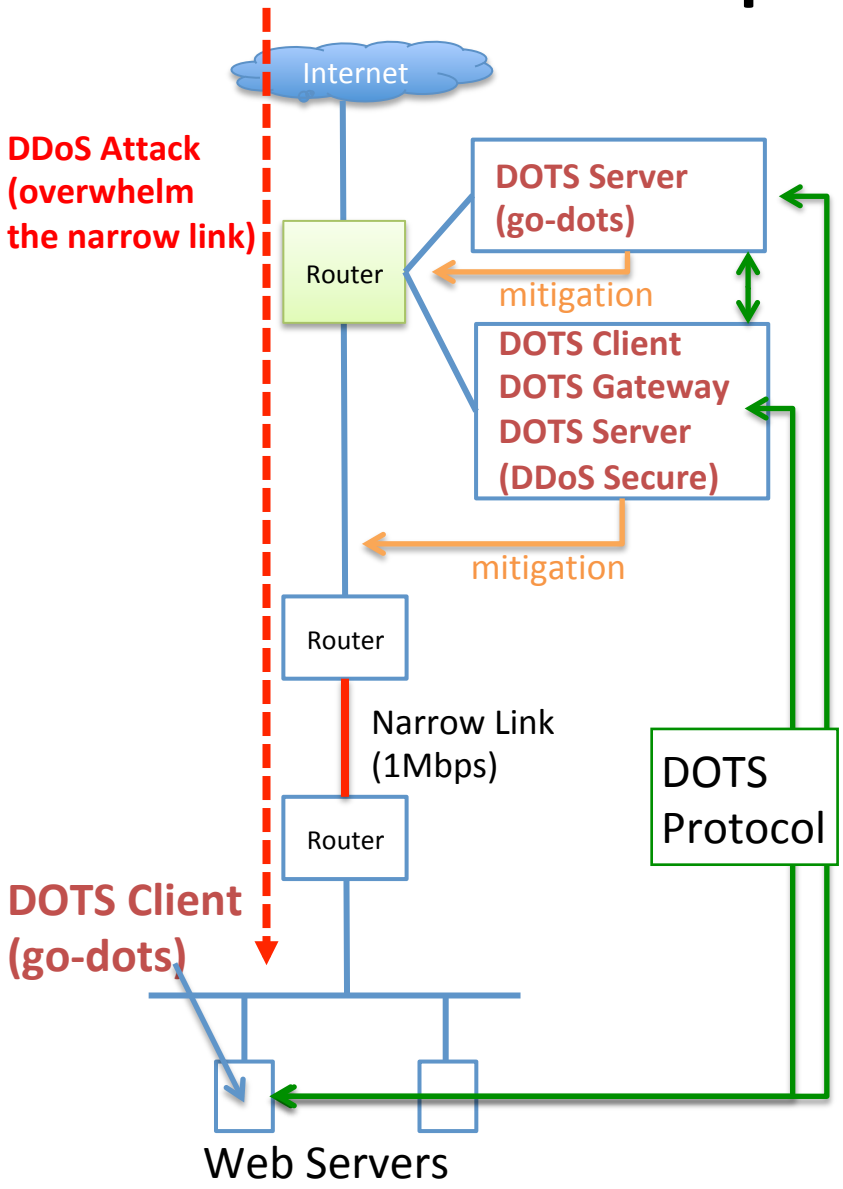
# Hackathon Plan

| 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<br>- The first data-channel interop | ✔ | ✔ | NTT, NCC Group, Huawei |
| IETF103 | 4th Interoperability Test<br>- with attack and protection demo | ✔ | ✔ | NTT, NCC Group, Huawei |

Objectives:

- Showing DOTS protocol's functional superiority for DDoS protection.
- Handling DDoS attack

# Interop test settings




Scenario:

- Setting up internet-facing web servers
  - A narrow link resides in the transit
- A DOTS client is in a web server and sets up DOTS sessions (in peace time) with a DOTS server in the upstream network (using narrow link)
- The server get DDoS-attacked
  - Make sure all of services in the attacked domain are not accessible
- The DOTS client requests for help with a mitigation request of the signal channel.

# Protection Capability

2 independent implementations:

- go-dots (https://github.com/nttdots/go-dots)
  - Tested as a client/server
  - Insertion of protecting ACLs on routers
  - Traffic redirection or RTBH by BGP route injection
- DDoS Secure (NCC Group)
  - Tested as a client/server/gateway
  - Inline protection as a DDoS Mitigation System (DMS)

Tested functions in both peace-time and attack-time:

- signal channel:
  - session configuration, mitigation request,  CoAP ping, observe
- data-channel:
  - registration of client/alias/filtering rules

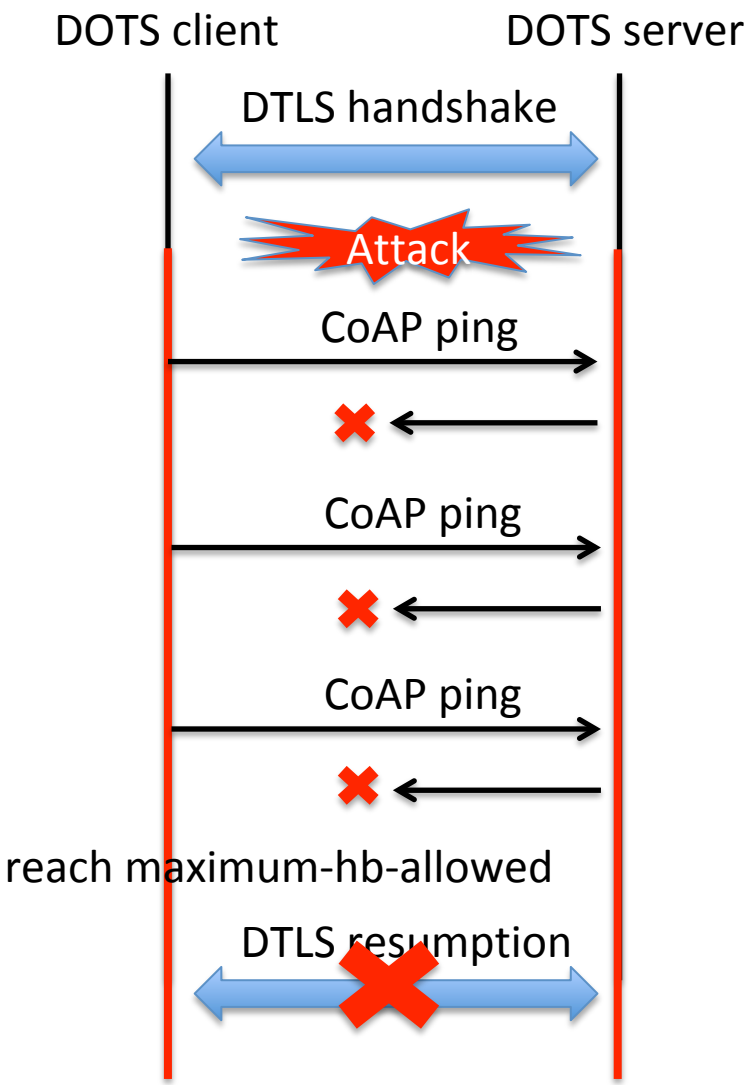
# Result Summary

High level summary of communications in attack-time

| | |
|---|---|
| Mitigation request | OK |
| CoAP ping | NG -> retry happens |
| DTLS handshake | NG |
| All DataChannel | NG |

- Attack-time (overwhelmed link by SYN flood):
  - over 90% packet loss for downstream
  - no packet loss for upstream
- Mitigation request works even in attack-time by design
- Failing functions in attack-time:
  - heartbeat mechanism
  - reconnection of DTLS
  - All of the data-channel communications

# DTLS resumption



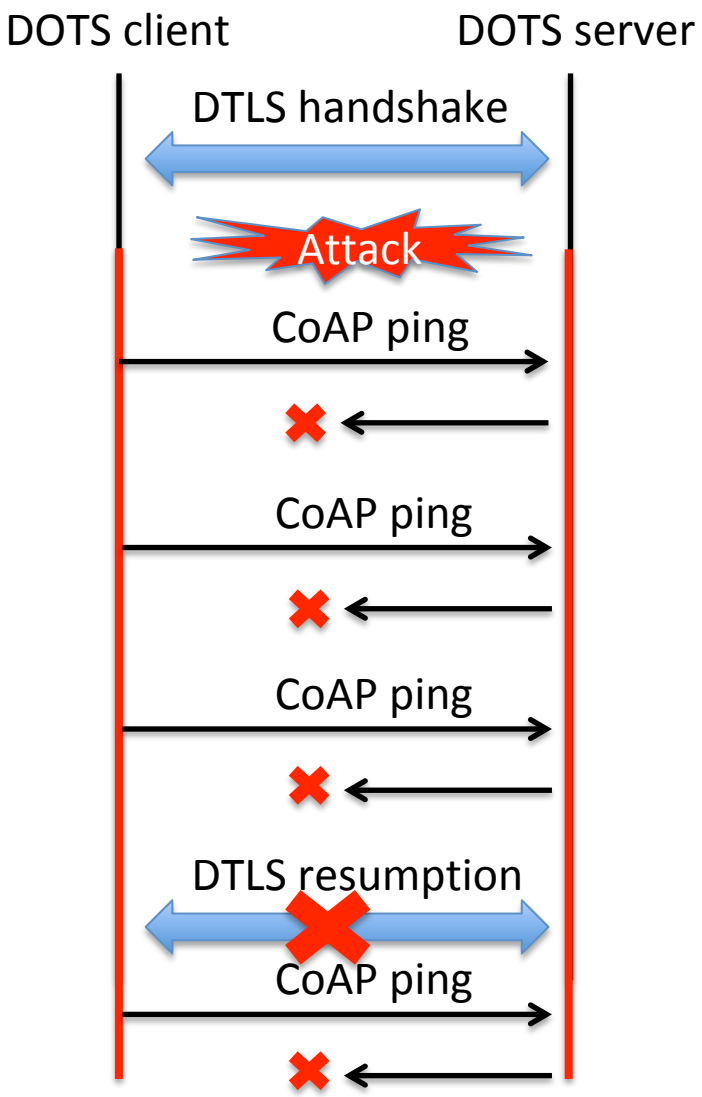

4.7. Heartbeat Mechanism

"After the maximum 'missing-hb-allowed' threshold is reached, the DOTS client SHOULD try to resume the (D)TLS session. The DOTS client SHOULD send mitigation requests over the current DOTS session, and in parallel, for example, try to resume the (D)TLS session or use 0-RTT mode in DTLS 1.3 to piggyback the mitigation request in the ClientHello message."

DTLS1.2 resumption(or re-negotiation) will fail in attack-time.

**Recommendation:** try to keep sending mitigation requests over the current DOTS session

# Issue: Trigger of Disconnection




4.7. Heartbeat Mechanism

"If the DOTS server does not receive any traffic from the peer DOTS client, then the DOTS server sends heartbeat requests to the DOTS client and after maximum 'missing-hb-allowed' threshold is reached, the DOTS server concludes the session is disconnected."

**Question:** Is there a better trigger for the server side heartbeat (it leads to conclusion of disconnection) for "deadman-trigger"?

# Implementation Implication

- It is important to make DOTS protocol robust to “incomplete communication”
  - must not rely on nor wait for returning packets in attack-time

# For more flexible protection

- It is assumed that data-channel filtering rules can not be installed/removed in attack-time
- What if a DOTS client need to change “active-when-mitigation” filters in attack-time?
- Options:
  1. should we add a control of data-channel filtering rules via signal-channel by name of aces?
  2. should we add an interface to get status of data-channel filtering rules via signal-channel?

# Takeaways

- Confirmed functional superiority of DOTS protocol for DDoS protection.
  - Mitigation requests can work even in attack-time by design
- Supportive functions do not work under attack
  - heartbeat mechanism
  - reconnection of DTLS
  - data-channel: install of filtering rules
- A lot of Implementation considerations in attack-time communication
- Core specification of DOTS is mature enough
  - No significant issue was found in real protection scenario

Questions

Or

Comments?

Thank You