UDP Options: Overcoming the Sorrows of the Young Extension

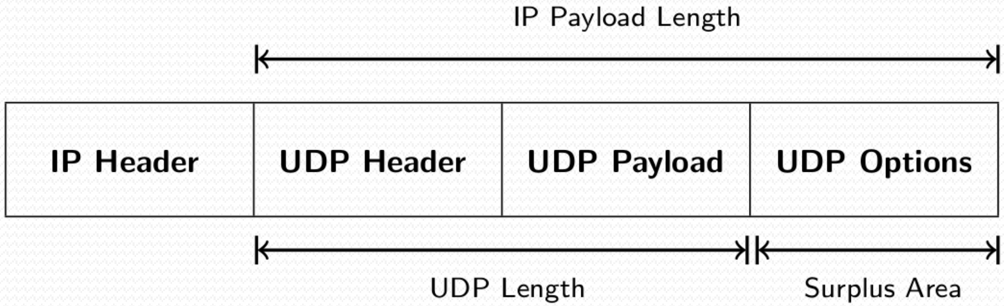
Raffaele Zullo, Tom Jones, Gorry Fairhurst

University of Aberdeen

Table of Contents

- **UDP Options**
- **Path traversal pathologies**
- **Checksum Compensation Option**
- **Measurements**
  - Methodology
- **Results**
  - Path traversal using CCO
  - Path traversal using zero checksum
- **Genesis of UDP Options pathologies**
- **Tools**
  - Tracemore
- **Conclusions**

UDP Options (UDP-O)



- **Surplus area**
  - **Redundancy between UDP Length and IP Payload Length**
    - IP Payload Length = IP Total Length – IP Header Length, for IPv4
    - IP Payload Length = IPv6 Payload Length – Length of IPv6 Extension Headers, for IPv6
  - Type-Length-Value Encoding
- **Fields affected**
  - Surplus area itself, IPv4 Total Length (IPv6 Payload Length), UDP Length, UDP Checksum, IPv4 Checksum

UDP Options

- **Usefulness of UDP Options**
  - Communicate **remote parameters**, e.g. the receiver maximum datagram size
  - Enable **higher level transport features**
  - Transport partially covered payload, like in **UDP-Lite**
  - Enable Datagram Packetization Layer PMTU Discovery (**DPLPMTUD**)
  - Provide transport-layer **fragmentation** in order to avoid the fragility of IP fragmentation (can benefit **DNSSEC**)
  - *Make transport parameters visible to on-path devices for encrypted transport protocols on top of UDP*
- **Transport Layer Ossification**
  - **TCP**, e.g. TFO Syn packets carrying payload
  - **UDP**, e.g. UDP zero checksum for IPv6 tunnel transports

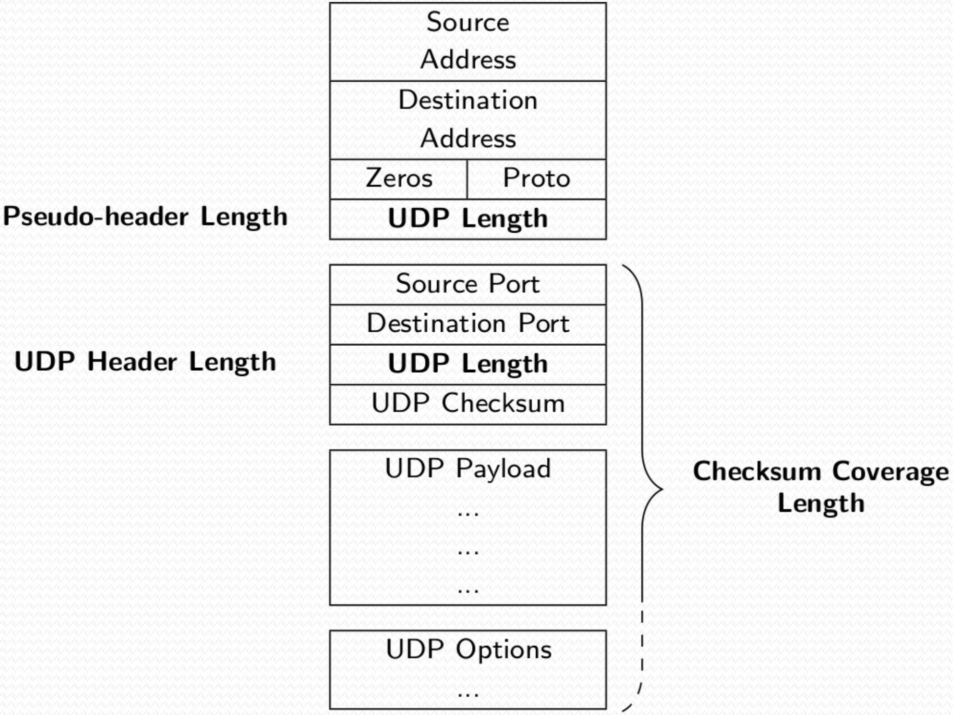
Can UDP Options be safely deployed?

UDP Options Pathologies

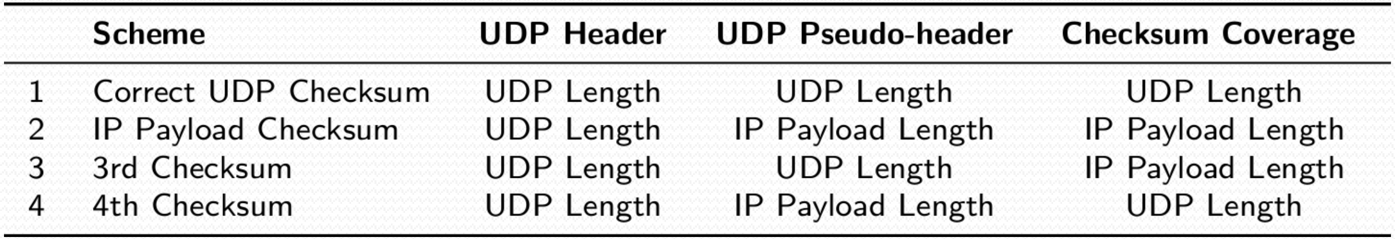
- **Pathologies**
  - **UDP Checksum validation**
    - Four checksum computation schemes observed in the wild (one benign)
  - **Length consistency check**
    - UDP Length = IP Payload Length
  - Pathologies tested but not detected
    - No deletion or alteration of the surplus area
    - No interference related IPv4 Checksum
      - It is computed on the IP header bytes only and involves the IP Total Length only
- **Devices affected**
  - **Middleboxes**
    - Home NATs, CGNs, Firewalls, IDS/IPS, etc
  - **End-hosts**
    - Due to checksum offloading to NIC

UDP Checksum Pathologies

- UDP Checksum computation involves **three Length values**



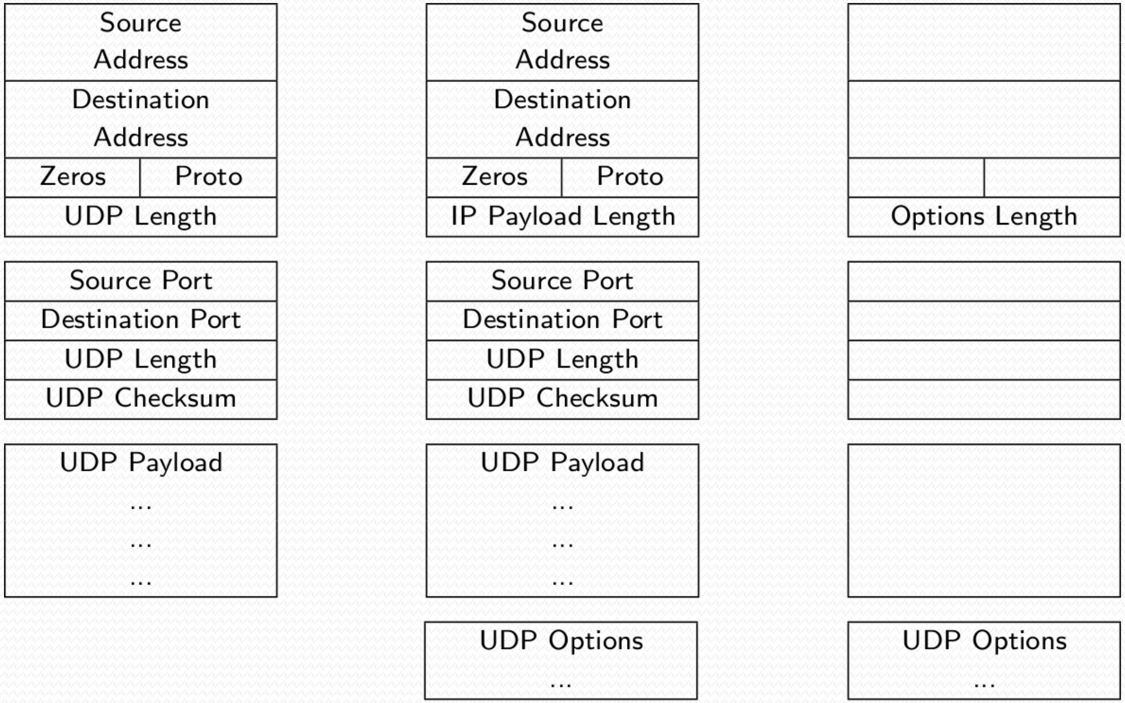
Four UDP Checksum Schemes



	Scheme	UDP Header	UDP Pseudo-header	Checksum Coverage
1	Correct UDP Checksum	UDP Length	UDP Length	UDP Length
2	IP Payload Checksum	UDP Length	IP Payload Length	IP Payload Length
3	3rd Checksum	UDP Length	UDP Length	IP Payload Length
4	4th Checksum	UDP Length	IP Payload Length	UDP Length

- Same value for UDP but four differing values for UDP-O
- Validation using 1st scheme is benign for UDP-O
- **Other 3 schemes discard UDP-O datagrams with the correct checksum**
- 2nd scheme (IP Payload Checksum) is the most prevalent

Correct CS vs IP Payload CS



Source Address
Destination Address
Zeros | Proto
UDP Length

Source Port
Destination Port
UDP Length
UDP Checksum

UDP Payload
…
…
…

Correct CS

Source Address
Destination Address
Zeros | Proto
IP Payload Length

Source Port
Destination Port
UDP Length
UDP Checksum

UDP Payload
…
…
…

UDP Options
…

IP Payload CS

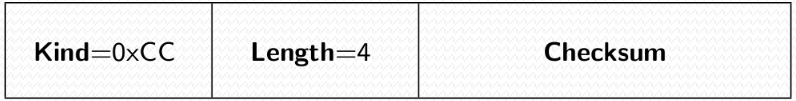
Options Length

UDP Options
…

Delta

Checksum Compensation Option

- **Format:**



Kind=0xCC	**Length**=4	**Checksum**

- **Definition:** CCO contains the 2-byte checksum of the Options area plus a 2-byte pseudo-header containing the length of the Options

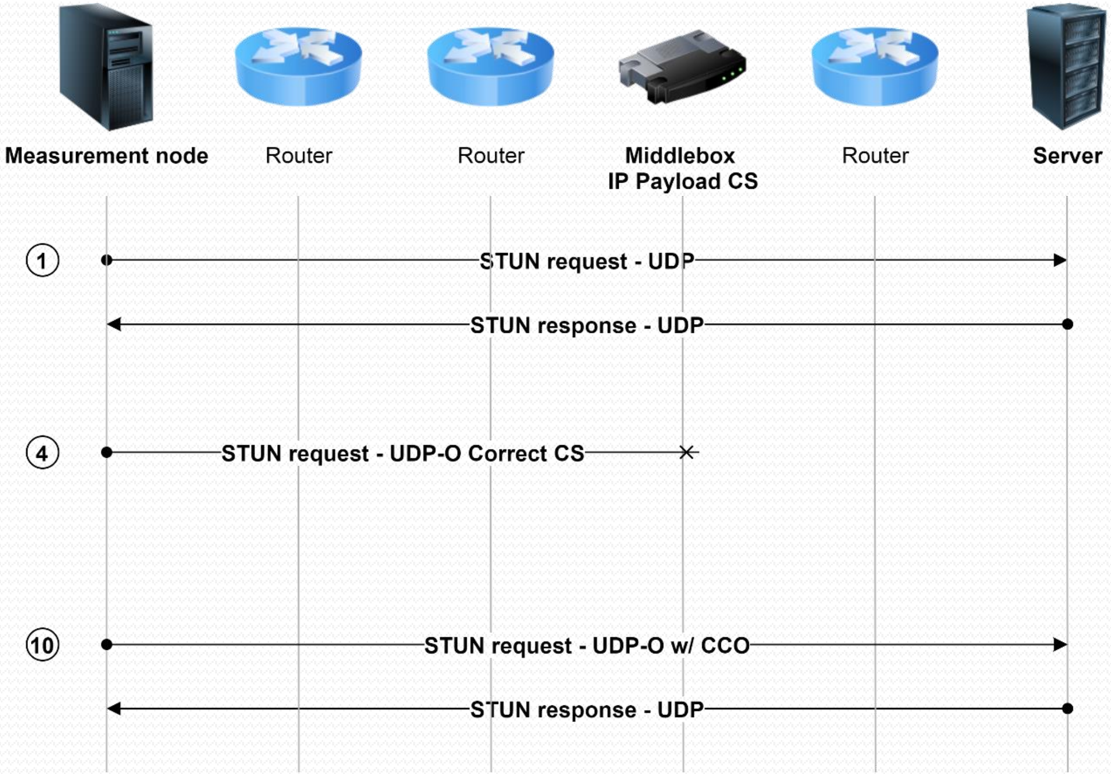
- **Purpose:** **CCO compensates the delta between the correct UDP checksum** (1st scheme) **and the IP Payload checksum** (2nd scheme)

- CCO checks the integrity of the Options area: it can replace UDP Option Checksum (OCS)

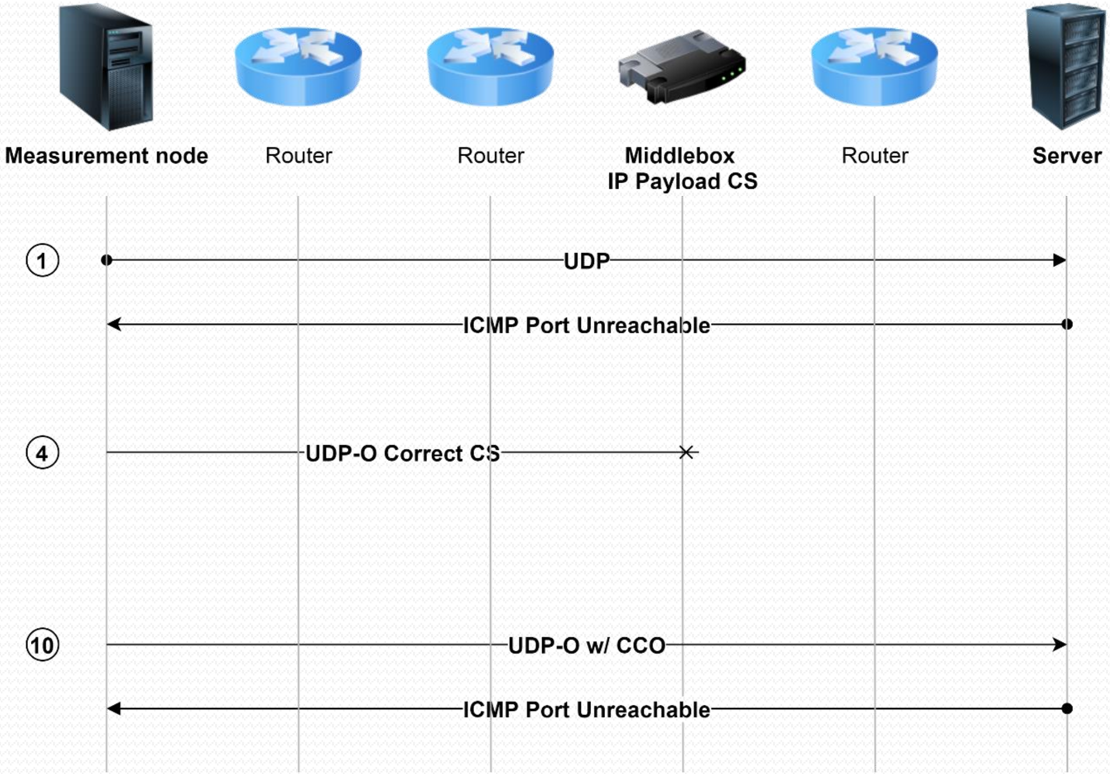
Measurements

- **Test suite**
  - **3 UDP** datagrams
  - **7 UDP-O** datagrams, one with **CCO**
- **Dataset**
  - IPv4 STUN servers
  - IPv4 and IPv6 Authoritative DNS servers and HTTP servers from Alexa Top-1m
  - **> 400K paths to servers tested**
- **Paths to UDP servers**
  - Using application packets, such as **DNS Query** or **STUN Bind Request**, encapsulated in UDP and UDP-O datagrams
- **Paths to HTTP servers**
  - HTTP servers are not expected to reply to UDP packets received on port 80
  - Some of them reply with **ICMP (or ICMPv6) Port Unreachable** messages
  - We can leverage the subset of ICMP messages received to infer which packets have reached the destination
  - **Caveats**:
    - Presence of a firewall before the HTTP server that replies with ICMP
    - ICMP rate limiting and other ICMP interference on the return path
    - Not all HTTP servers reply with ICMP

UDP Servers Methodology

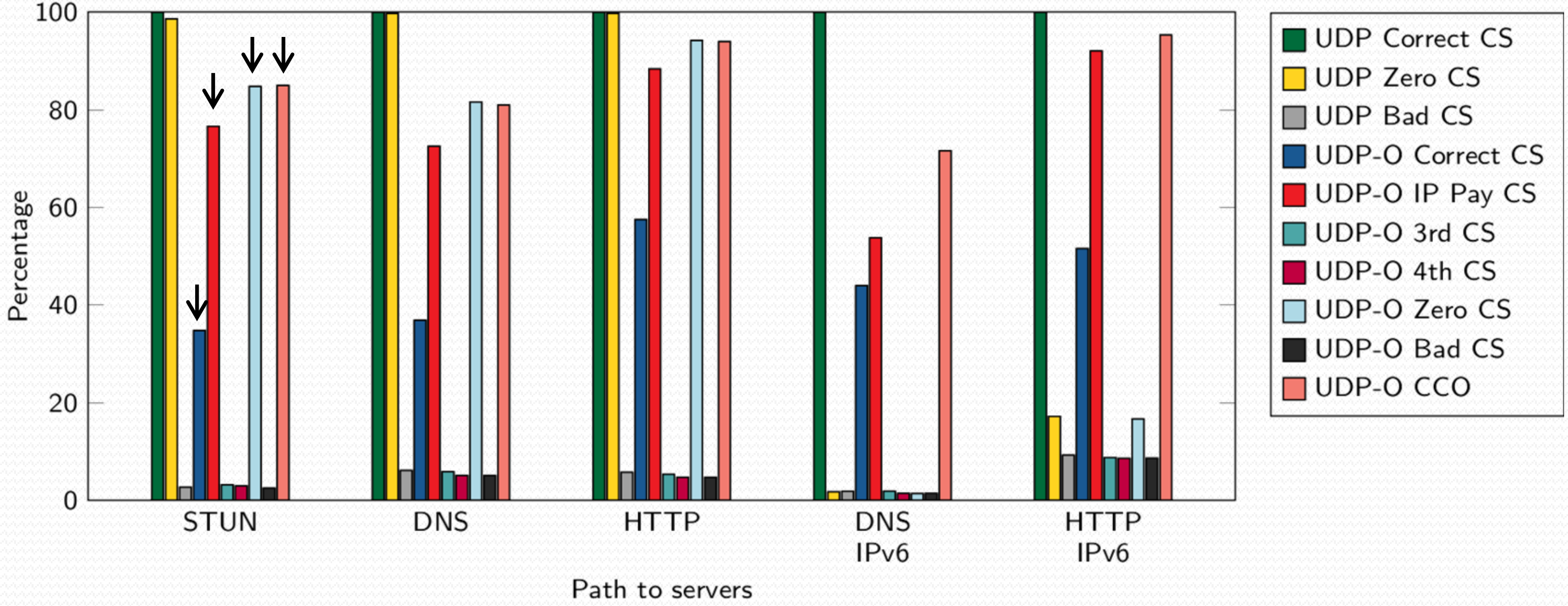


HTTP Servers Methodology



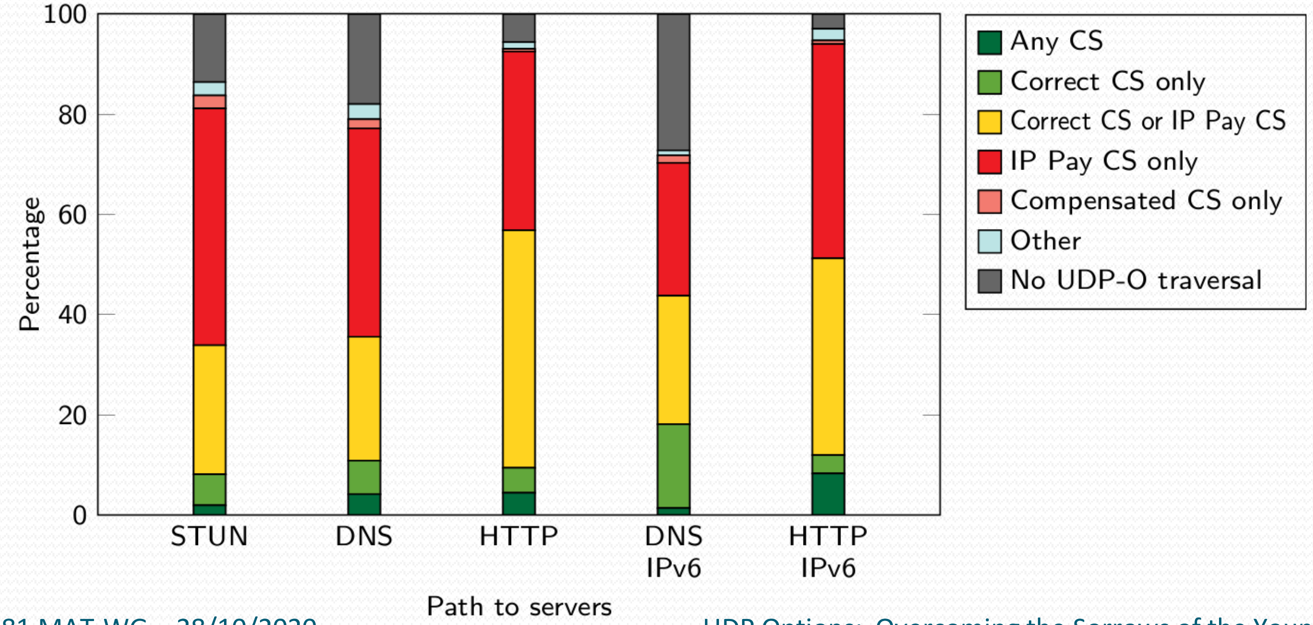
Overall Traversal Results

- Limited traversal rate for **UDP-O Correct CS** (original specifications)
- Better performances for **IP Payload CS**, **Zero CS** and **CCO**



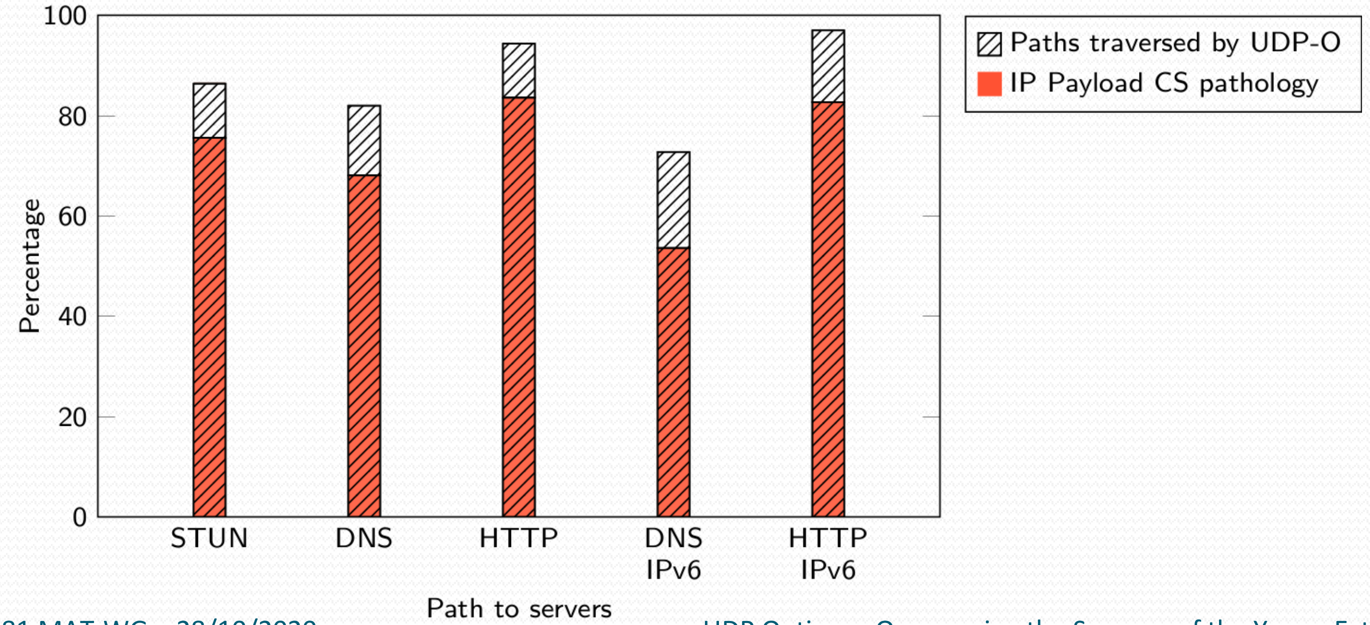
Path Characterisation

- **IP Payload checksum is the most widespread pathology**
  - Can be present in combination with the benign pathology
- **No UDP-O traversal on about 16% of the paths** (due to a length consistency check)



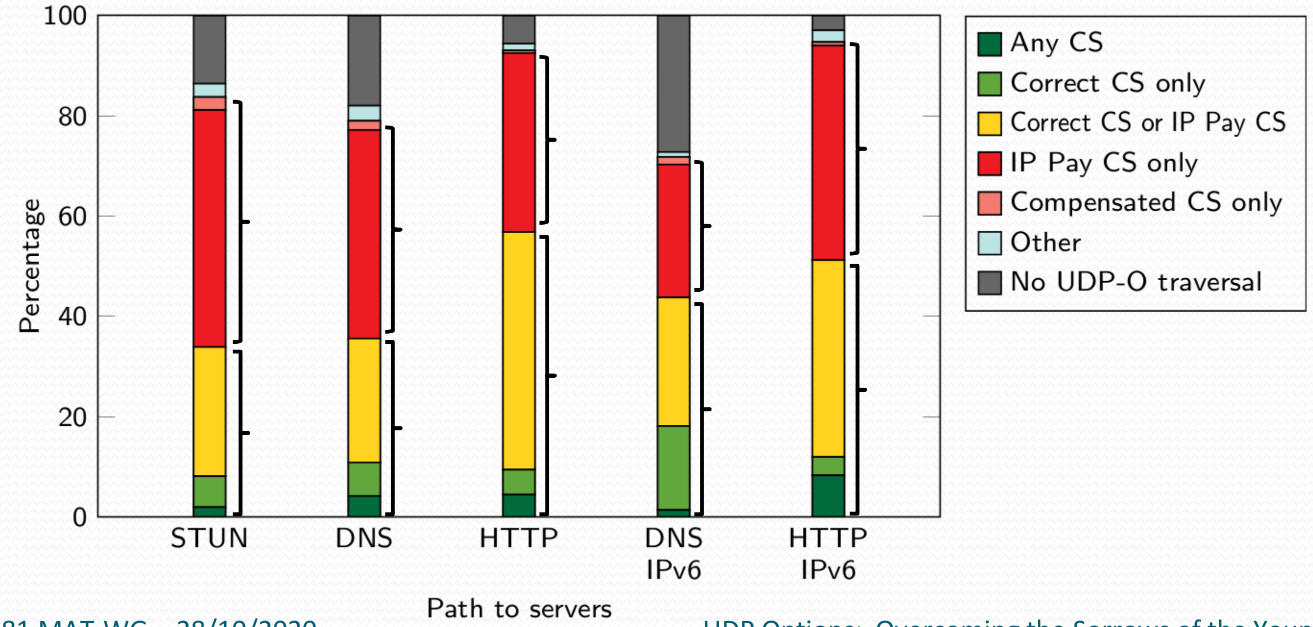
IP Payload Checksum Pathology

- **About 80% of the paths traversed by at least one UDP-O datagram are affected by the IP Payload Checksum pathology**
  - Alone or in conjunction with the benign pathology



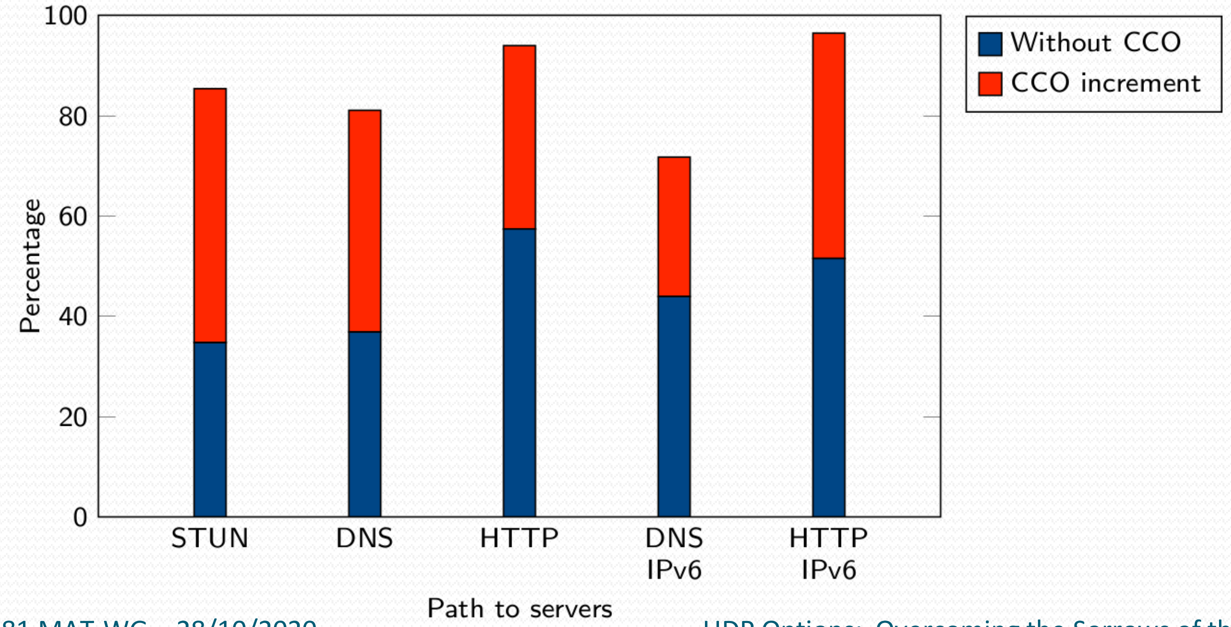
Path Characterisation

- **Traversed according to original UDP-O specifications:** Any CS, Correct CS only, Correct CS or IP payload CS
- **Traversed only using CCO:** IP payload CS and Compensated CS only



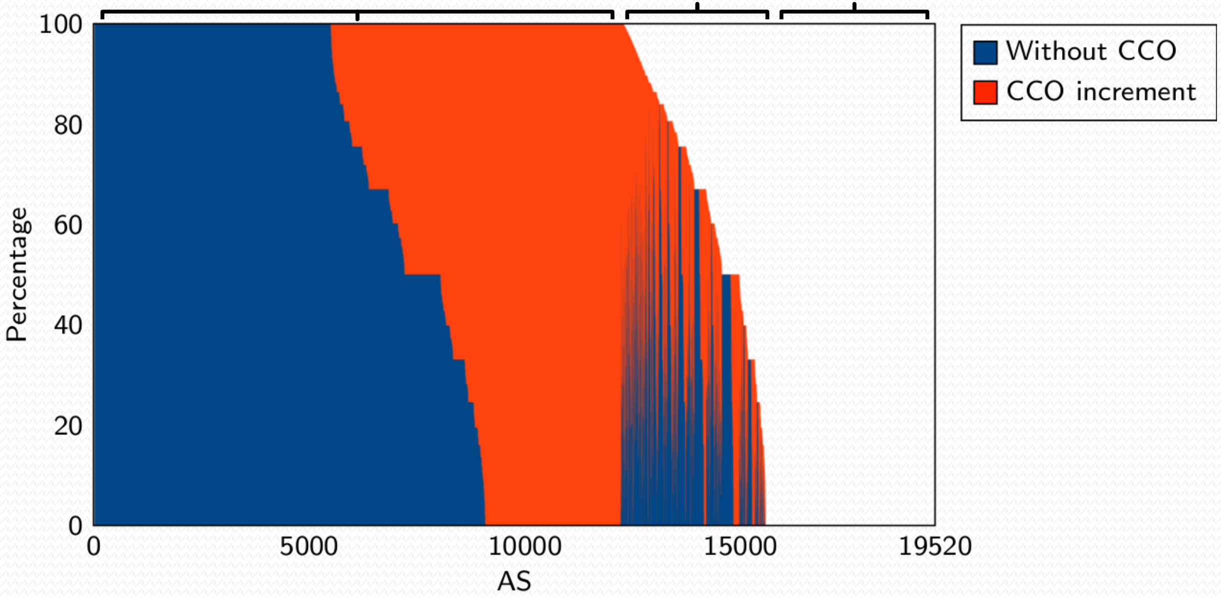
Path Traversal Using CCO

- **CCO significantly increases UDP-O traversal rate**
- For IPv4 paths to STUN and DNS servers, the increment from using the CCO is even greater than the number of paths originally traversed by UDP-O



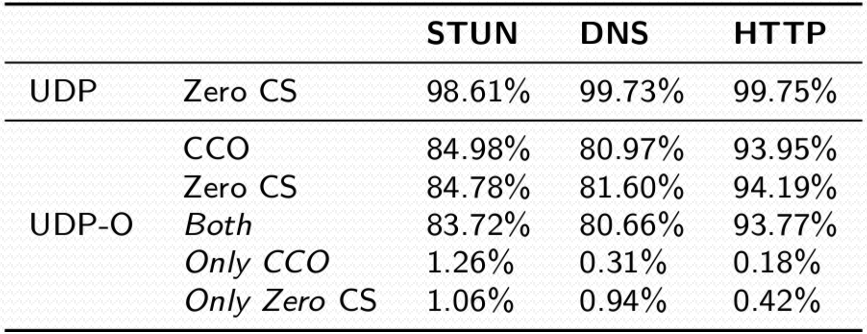
Path Traversal Using CCO per AS

1. ASes in which all paths can be traversed by UDP-O (63%)
2. ASes in which a subset of paths can be traversed by UDP-O (18%)
3. ASes in which no measured path could be traversed by UDP-O, without or with the CCO (19%)



Path traversal using zero CS

Comparison of UDP-O traversal with CCO and zero checksum



		STUN	DNS	HTTP
UDP	Zero CS	98.61%	99.73%	99.75%
UDP-O	CCO	84.98%	80.97%	93.95%
	Zero CS	84.78%	81.60%	94.19%
	Both	83.72%	80.66%	93.77%
	Only CCO	1.26%	0.31%	0.18%
	Only Zero CS	1.06%	0.94%	0.42%

- **Zero checksum traversal is not always better than CCO**
  - Interference with zero checksum was also observed with regular UDP datagrams
  - Results are limited to IPv4
- **Zero checksum can be an alternative for UDP Options that, by design, should not be covered by a checksum**
  - e.g. LITE

Genesis of UDP-O Pathologies

- **Checksum pathologies**
  - **Ambiguity in the role of the two lengths**
  - **Analogy with TCP checksum computation**
    - Since TCP has no length field the length of a TCP segment is deduced from the IP header and the checksum is computed over all transport layer bytes
- **Length consistency check**
  - **Assumption that UDP Length and IP Payload length coincide**
    - Detection of malformed packets
  - **Prevention of covert channel communication**

Network Equipment Manufacturers

- **Manufacturer #1**
  - Explained that on it the default behavior for a stateful firewall was to discard all packets with incorrect checksums

    This is actually reasonable since, before applying rules that involve transport layer to the packet, transport layer integrity should be verified
- **Manufacturer #2**
  - Confirmed that their middleboxes performed a consistency check between IP and UDP length along with other integrity checks on datagrams and discarded them in the case of a length mismatch

The case of Correct CS OR IP Pay CS

- **Dual checksum validation**
  - Cannot be due to two distinct devices
    - Each device would discard the checksum compliant to the other
  - A possible explanation is that the two validations happen at different layers within a single device
- **Observed on Linux devices: workstations, servers, Android smartphones**
  - IP Payload checksum validation only observed when checksum offloading enabled
- **Linux kernel code**
  - If the checksum is validated by the NIC the datagram is directly accepted otherwise the checksum is verified again using the kernel routine
- **Less benign than expected**
  - Incoming UDP-O packets are not validated correctly by the NIC so they need to be validated at kernel level
  - For outgoing UDP-O packets offloading must be disabled
- **CCO can help**
  - Incoming UDP-O packets are validated directly by the NIC
  - The checksum on outgoing UDP-O packets can be offloaded, leaving only the checksum on the surplus area to be computed at kernel level

UDP-O Measurement Tools

- **Tracemore**
  - To reproduce our measurements
  - To test UDP-O in your network
  - Can pinpoint the interfering node
  - Requires root
  - Code available
  - Measurement script available
- Basic Scapy script
  - Single UDP Option and precomputed CCO
- **Mobile Tracebox**
  - To quickly test UDP-O from an Android device
  - Does not require root

Tracemore

- **Derived from *Mobile Tracebox* code base**
- **Written in C**
- All IP and UDP fields can be customised
  - **UDP and UDP-O packets**
  - **4 UDP-O checksum schemes**
- Payload can be customised using crafted application packets, e.g. a DNS query
- Embodies **traceroute** / **tracebox** methodology
- **Code:**

  https://github.com/raffaelezullo/tracemore

Tracemore

- **End-to-end:** DNS server

UDP

```
 0:  212.25.x.x    [UDP 33 bytes]
64:  87.240.x.x    [UDP 64 bytes]
```

UDP-O

```
 0:  212.25.x.x    [UDP 33 bytes]
64:  * * *
```

UDP-O w/CCO

```
 0:  212.25.x.x    [UDP 33 bytes]
64:  87.240.x.x    [UDP 64 bytes]
```

- **Traceroute:** edge and core network (*Three UK*)

UDP

```
 0:  10.190.x.x    [UDP 33 bytes]
 1:  * * *
 2:  172.23.x.x
 3:  172.23.x.x
 4:  172.23.x.x
 5:  * * *
 6:  188.31.x.x
 7:  188.31.x.x
 8:  188.31.x.x
 9:  188.31.x.x
10:  195.66.x.x
11:  1.1.x.x       [UDP 64 bytes]
```

UDP-O

```
 0:  10.190.x.x     [UDP 33 bytes]
 1:  * * *
 2:  * * *
 3:  * * *
```

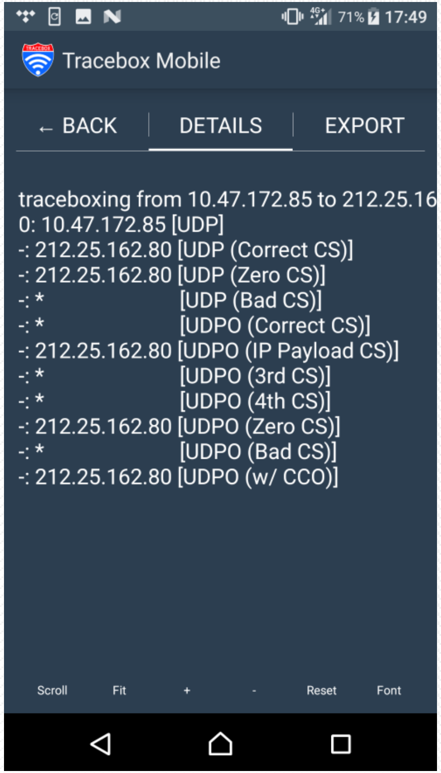
UDP-O w/CCO

```
 0:  10.190.x.x    [UDP 33 bytes]
 1:  * * *
 2:  172.23.x.x
 3:  172.23.x.x
 4:  172.23.x.x
 5:  * * *
 6:  188.31.x.x
 7:  188.31.x.x
 8:  188.31.x.x
 9:  188.31.x.x
10:  195.66.x.x
11:  1.1.x.x       [UDP 64 bytes]
```

Mobile Tracebox

- To quickly test UDP-O from an Android device
- Does not require root
- Settings: Server-based, UDP, UDP Options (Experimental)
- Example output (*Three UK):*

UDP-O packets with the correct checksum cannot be received unless CCO is used



```
traceboxing from 10.47.172.85 to 212.25.16
0: 10.47.172.85 [UDP]
-: 212.25.162.80 [UDP (Correct CS)]
-: 212.25.162.80 [UDP (Zero CS)]
-: *             [UDP (Bad CS)]
-: *             [UDPO (Correct CS)]
-: 212.25.162.80 [UDPO (IP Payload CS)]
-: *             [UDPO (3rd CS)]
-: *             [UDPO (4th CS)]
-: 212.25.162.80 [UDPO (Zero CS)]
-: *             [UDPO (Bad CS)]
-: 212.25.162.80 [UDPO (w/ CCO)]
```

Conclusions and Future Work

- **First analysis of UDP-O path pathologies**
- Limited traversal success for UDP-O according to the original specification
- **Checksum Compensation Option**
- **CCO can significantly increase UDP-O traversal rate**

  ⇨ **Redesign OCS to achieve CCO function**
  - Zero checksum can be an alternative for specific UDP Options such as LITE
- Genesis of UDP-O pathologies
- **Measurement Tools**

- Validate our results on a **larger dataset**
  - Scans over other UDP protocols (on IPv4 full range and IPv6 target lists)
- Edge networks **crowdsourced measurement**
  - New tools (such as Mobile Tracebox) for UDP-O measurement

Thank you

R Zullo, T Jones, G Fairhurst - Overcoming the Sorrows of the Young UDP Options (TMA2020)

https://tma.ifip.org/2020/wp-content/uploads/sites/9/2020/06/tma2020-camera-paper70.pdf

Tracemore

https://github.com/raffaelezullo/tracemore

Questions, comments, etc

<raffaele.zullo@gmail.com>

<raffaele@erg.abdn.ac.uk>

References

[1] R. Zullo, T. Jones, and G. Fairhurst, “Overcoming the Sorrows of the Young UDP Options”, 2020 Network Traffic Measurement and Analysis Conference (TMA), IEEE, 2020, https://tma.ifip.org/2020/wp-content/uploads/sites/9/2020/06/tma2020-camera-paper70.pdf

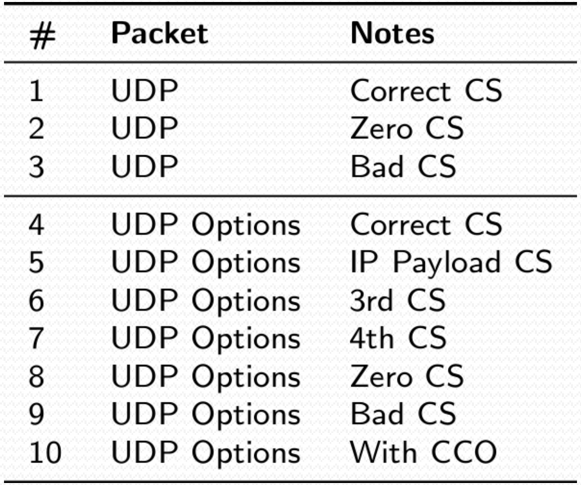
[2] J. Touch, “Transport options for UDP”, 2019, IETF Internet draft draft-touch-tsvwg-udpoptions, https://datatracker.ietf.org/doc/draft-touch-tsvwg-udp-options/

[3] G. Fairhurst, T. Jones, and R. Zullo, “Checksum Compensation Optionsfor UDP Options,” 2018, IETF Internet-Draft draft-fairhurst-udpoptions-cco, https://datatracker.ietf.org/doc/draft-fairhurst-udp-options-cco/

[4] G. Fairhurst, T. Jones, and R. Zullo, “A Tale of Two Checksums”, 2018, IETF, http://www.middleboxes.org/raffaelezullo/publications/ietf103-maprg-cco-slides.pdf

Test Suite

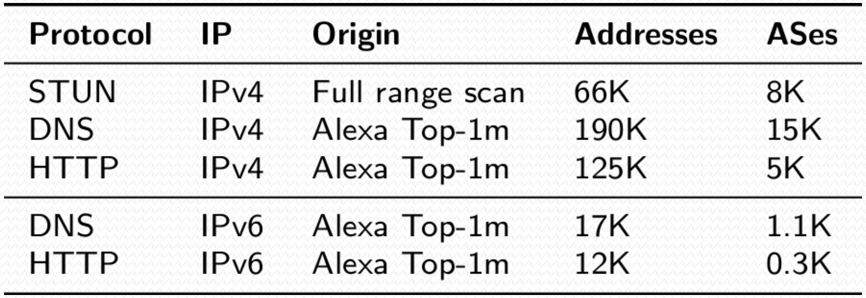
- **3 UDP datagrams**
  - To characterise the path in absence of UDP Options
- **7 UDP-O datagrams,** one with CCO
  - To detect interference with UDP Options



#	Packet	Notes
1	UDP	Correct CS
2	UDP	Zero CS
3	UDP	Bad CS
4	UDP Options	Correct CS
5	UDP Options	IP Payload CS
6	UDP Options	3rd CS
7	UDP Options	4th CS
8	UDP Options	Zero CS
9	UDP Options	Bad CS
10	UDP Options	With CCO

Dataset

- **Paths tested**



Protocol	IP	Origin	Addresses	ASes
STUN	IPv4	Full range scan	66K	8K
DNS	IPv4	Alexa Top-1m	190K	15K
HTTP	IPv4	Alexa Top-1m	125K	5K
DNS	IPv6	Alexa Top-1m	17K	1.1K
HTTP	IPv6	Alexa Top-1m	12K	0.3K

- **STUN** servers list obtained from a preliminary **IPv4 full range scan**
- Autoritative **DNS** servers and **HTTP** servers list obtained from **Alexa Top-1m**
  - About one quarterof the servers in the full HTTP list were eligible for our test

Characterising the Path

- Each packet in the test suite provide information about the path
- Only their combination can highlight the pathology or pathologies that affect the path

Path characterization	**Tests**									
	1	2	3	4	5	6	7	8	9	10
Any Checksum	✓	*	✓	✓	✓	✓	✓	*	✓	✓
Correct UDP CS only	✓	*	×	✓	×	×	×	*	×	✓
IP Payload CS only	✓	*	×	×	✓	×	×	*	×	✓
3rd CS only	✓	*	×	×	×	✓	×	*	×	×
4th CS only	✓	*	×	×	×	×	✓	*	×	×
Correct CS or IP Pay CS	✓	*	×	✓	✓	×	×	*	×	✓
Compensated CS only	✓	*	×	×	×	×	×	*	×	✓
Zero CS only	✓	✓	×	×	×	×	×	✓	×	×
No UDP-O traversal	✓	*	*	×	×	×	×	×	×	×