# Overcoming the Sorrows of the Young UDP Options

**Raffaele Zullo**, Tom Jones, Gorry Fairhurst

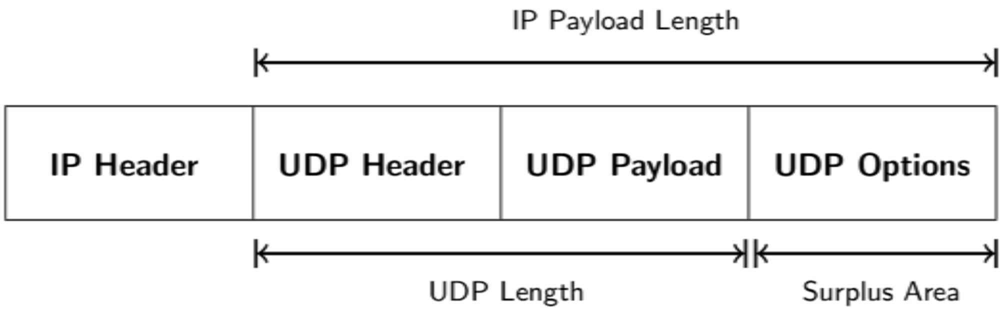
University of Aberdeen, UK

UNIVERSITY OF ABERDEEN

# Road map

- **UDP Options**
- **Path traversal pathologies**
  - UDP Checksum validation
  - UDP and IP length consistency check
- **Checksum Compensation Option**
- **Measurements**
  - Tracemore
- **Results**
  - Path characterisation
  - Path traversal using CCO
  - Path traversal using zero checksum
- **Genesis of UDP Options pathologies**
- **Conclusions**

# UDP Options (UDP-O)



- **Surplus area**
  - Added at the end of UDP Payload
  - Leverages the 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**
  - IPv4 Total Length (IPv6 Payload Length), Surplus area itself, UDP Length, UDP Checksum, IPv4 Checksum
- **Draft:** draft-touch-tsvwg-udpoptions [1]

# UDP Options

- **Usefulness of UDP Options**
  - Communicate remote parameters, e.g. the receiver maximum datagram size
  - Signal metadata about a stream to the network path, e.g. loss reports, RTT, ECN feedback
  - Enable higher level transport features
  - Transport partially covered payload, like in UDP-Lite
  - Enable Datagram Packetization Layer PMTU Discovery
  - Provide transport-layer fragmentation in order to avoid the fragility of IP fragmentation
    - Can benefit DNSSEC
- **Transport Layer Ossification**
  - **TCP:** TCP Fast Open amending the original handshake mechanism
    - Obstructed by devices built to enforce the original handshake requirements
  - **UDP:** Amendment of the original IPv6 specifications to allow endpoints to use a zero UDP checksum (for tunnel transports that carry an already checksum-protected packet)
    - Obstructed by devices that consider IPv6 UDP datagrams with zero checksum malformed

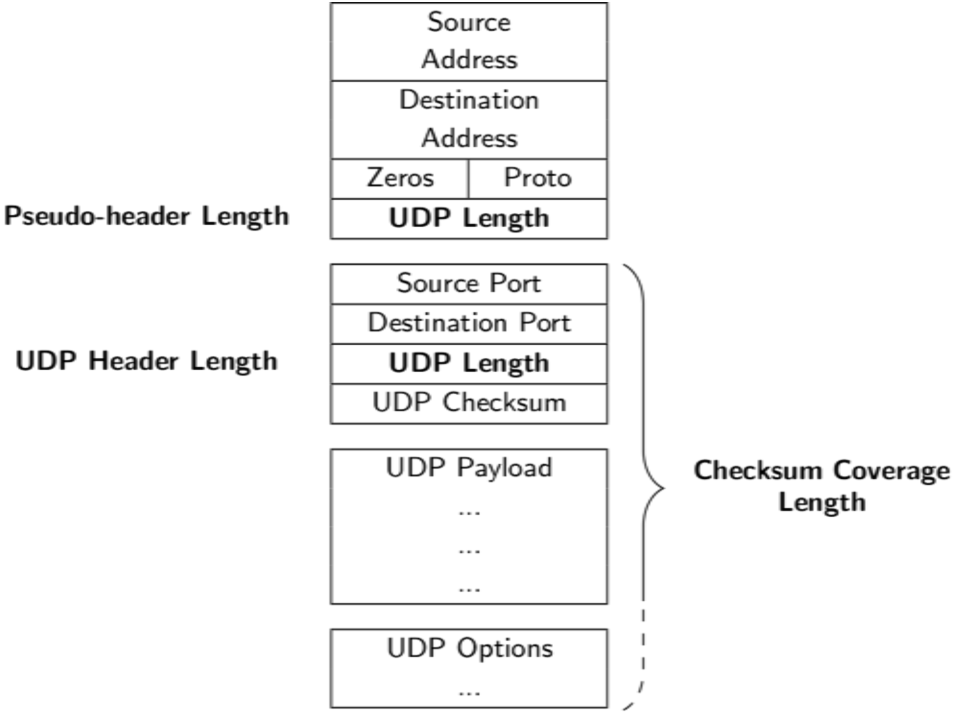
**<u>Can UDP Options be deployed?</u>**

# UDP Options Pathologies

- **Pathologies**
  - **UDP Checksum validation**
    - Four checksum computation schemes implemented 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**
    - Checksum offloading to NIC

# UDP Checksum Pathologies

- UDP Checksum computation involves three Length values



Source Address

Destination Address

Zeros | Proto

**UDP Length**

**Pseudo-header Length**

Source Port

Destination Port

**UDP Length**

UDP Checksum

**UDP Header Length**

UDP Payload

...

...

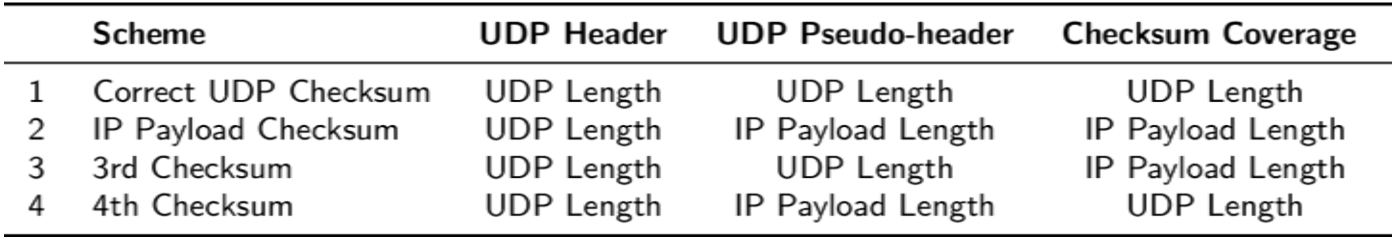
...

UDP Options

...

**Checksum Coverage Length**

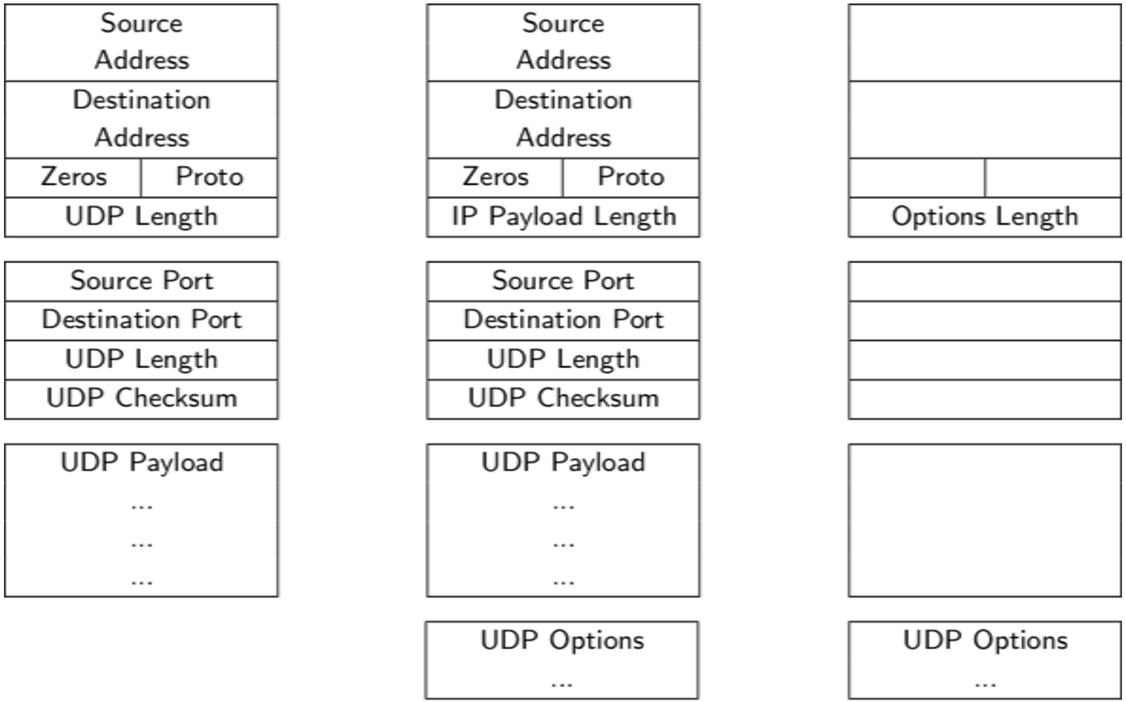
# Four UDP Checksum Schemes



| | 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
- 1st scheme is benign
- **UDP-O with the correct  checksum are discarded by devices implementing the other schemes**
- After a preliminary analysis the 2nd scheme (IP Payload checksum) turned out to be the most widespread
- A first sign of this issue was found in a bug detected in FreeBSD,  where the checksum length was based on the IP length, that was corrected during UDP-O Implementation [3]

# 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** |

- **Draft:** draft-ietf-fairhurst-udp-options-cco [2]

- **Definition:** CCO contains the 2-byte checksum of the Options area plus a 2-byte pseudo-header conceptually prefixed to the options and containing the length of the surplus area itself.

  CCO provides also an integrity check on the Options area: it can replace UDP OCS (Option Checksum)

- **Purpose:** **CCO compensates the difference between the correct UDP checksum and the IP Payload checksum**

- **Notes:** Padding and other measures have to taken into account in its calculation to also compensate the misalignment between the UDP header and the first byte of the options or the CCO itself

# Tracemore

- **Derived from *Mobile Tracebox* code base**
- **Written in C**
- All IP and UDP fields can be customised
  - **Can forge UDP and UDP-O packets**
  - **Can generate a checksum compliant with each of the 4 schemes or even not compliant with any scheme**
- Payload can be customised using crafted application packets, e.g. a DNS query
- Can perform traceroute / tracebox
- **Code available at:**
  **http://www.middleboxes.org/tracemore**

# Tracemore

- **Edge netwok** (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]
```

- **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]
```

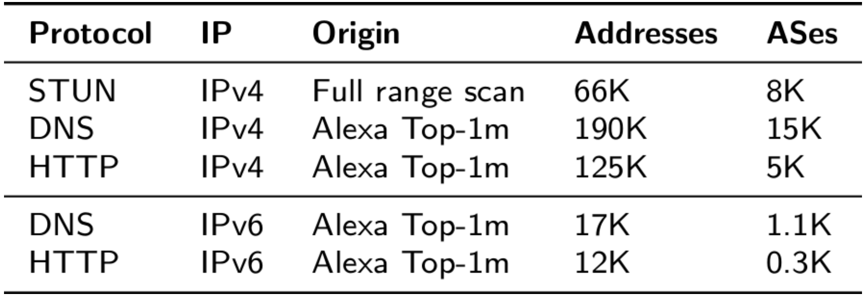
# Methodology

- **Paths to UDP servers**
  - Application packets, such as **DNS Query** or **STUN Bind Request**, are encapsulated in UDP and UDP-O datagrams and sent to UDP servers
  - Based on the subset of replies received we can infer which packets have reached the destination and therefore which pathology or pathologies affect the path

- **Paths to HTTP servers**
  - To increase the number of paths and ASes tested
  - 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
  - Further considerations:
    - 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

# Dataset

- **Paths tested**

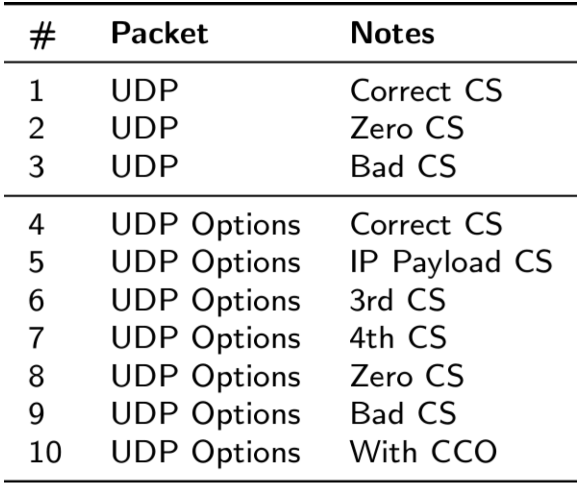


| Protocol | IP | Origin | Addresses | ASes |
|---|---|---|---|---|
| STUN | IPv4 | Full range scan | 66K | 8K |
| DNS | IPv4 | Alexa Top-1m | 190K | 15K |
| HTTP | IPv4 | Alexa Top-1m | 125K | 5K |
| DNS | IPv6 | Alexa Top-1m | 17K | 1.1K |
| HTTP | IPv6 | Alexa Top-1m | 12K | 0.3K |

- **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

# Test Suite

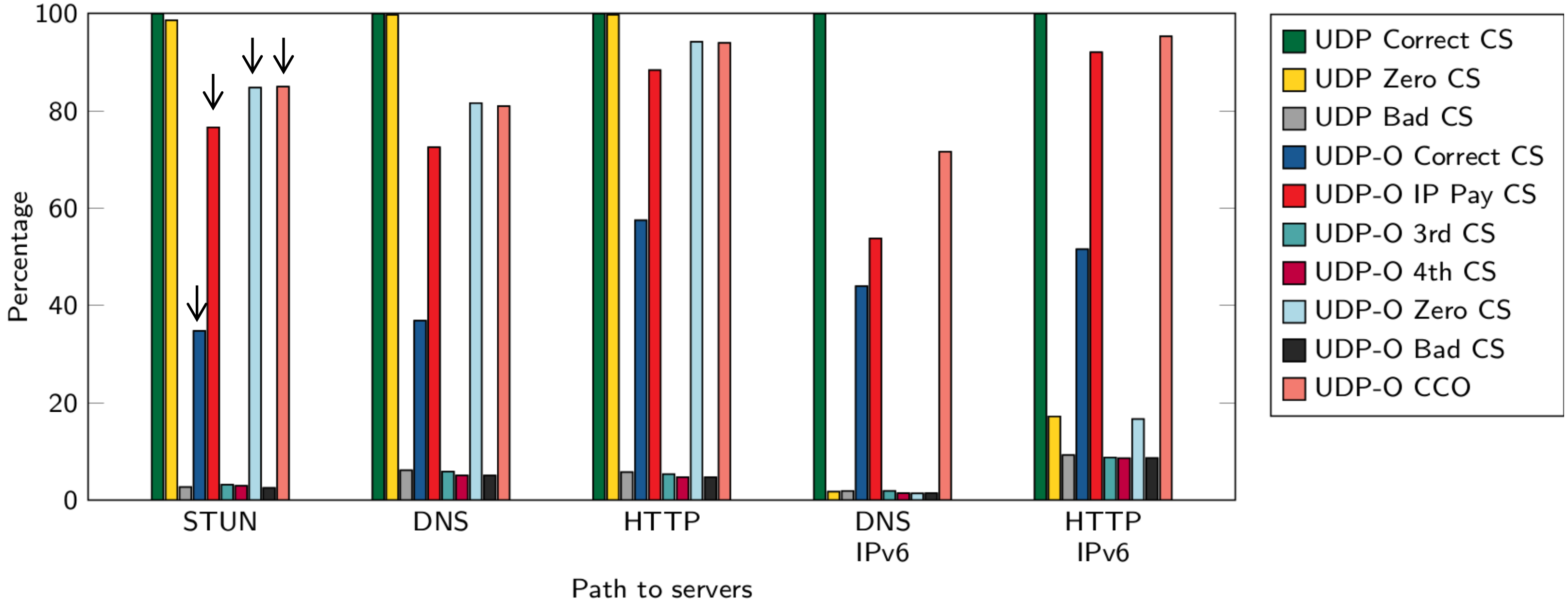
- **3 UDP datagrams**
  - To characterise the path in absence of UDP Options
- **7 UDP-O datagrams,** one with CCO
  - To detect interference with UDP Options



| # | 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 |

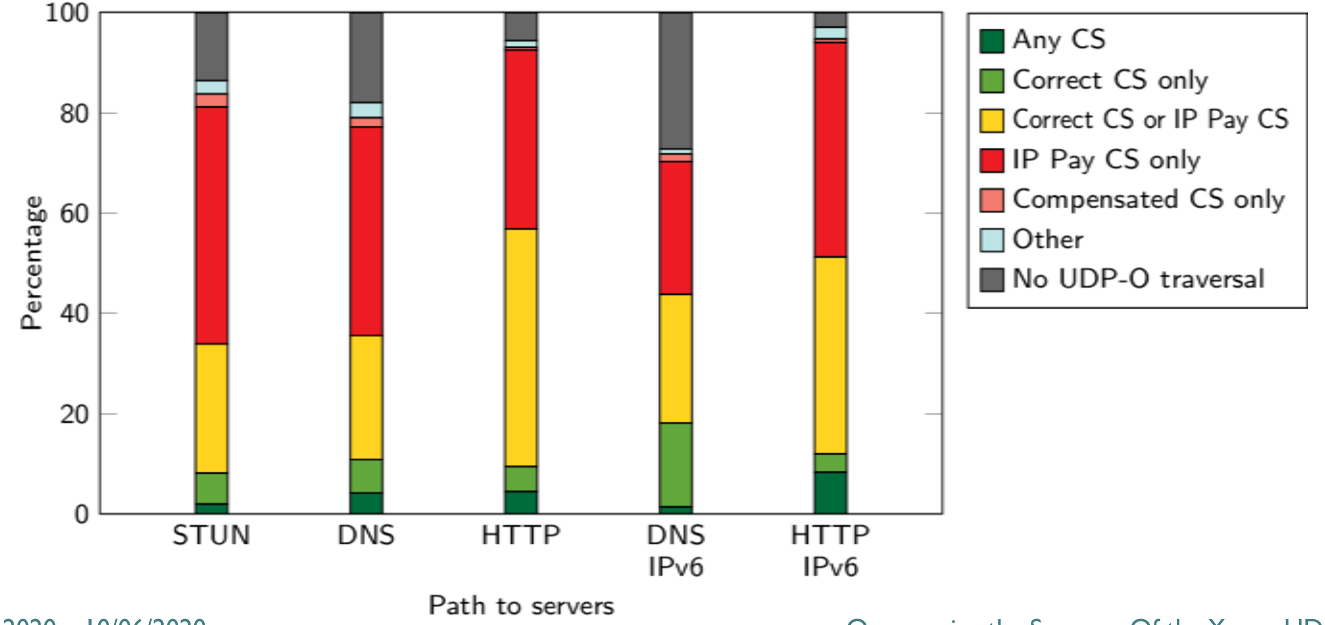
# Overall Traversal Results

- Limited traversal rate for UDP-O datagrams compliant with the original specification (**UDP-O Correct CS**)
- Better performances for UDP-O packets with **IP Payload CS**, **zero CS** or using **CCO**



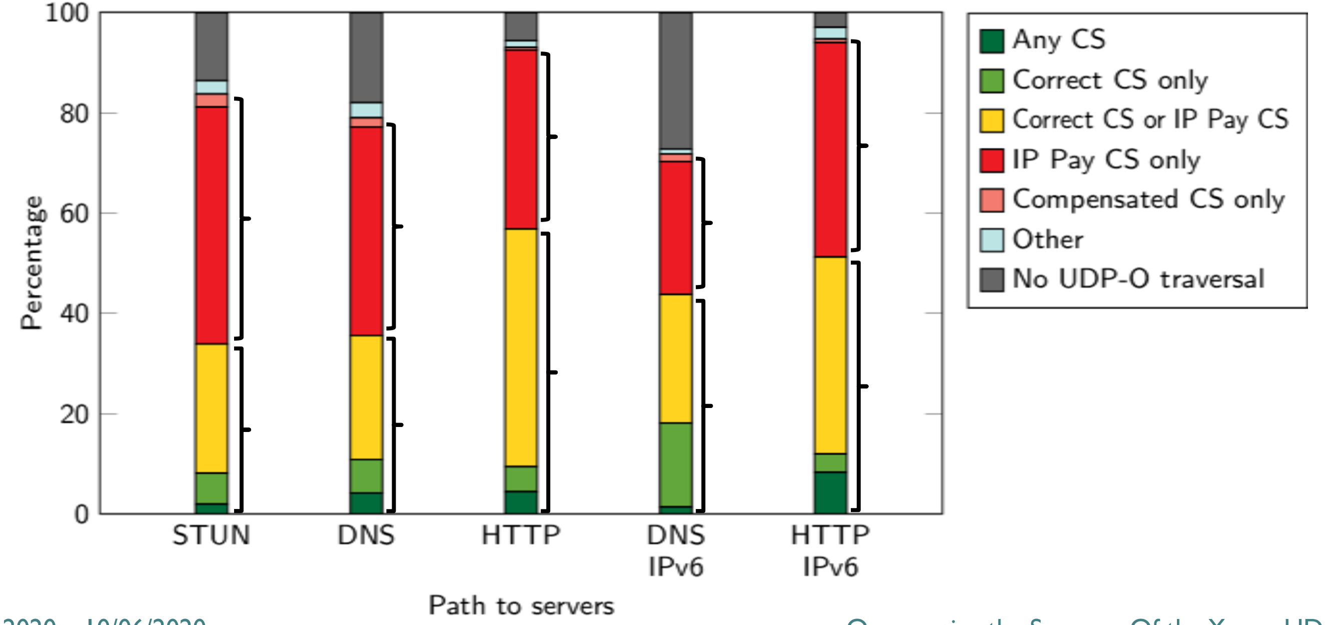
# Path Characterisation

- **IP Payload checksum is the most widespread pathology** (**80%** of paths traversed by UDP-O)
  - Can be present in combination with the benign pathology on the same path
  - 3rd and 4th checksum pathology very rare
- **No UDP-O traversal on about 16% of the paths** (length consistency check)



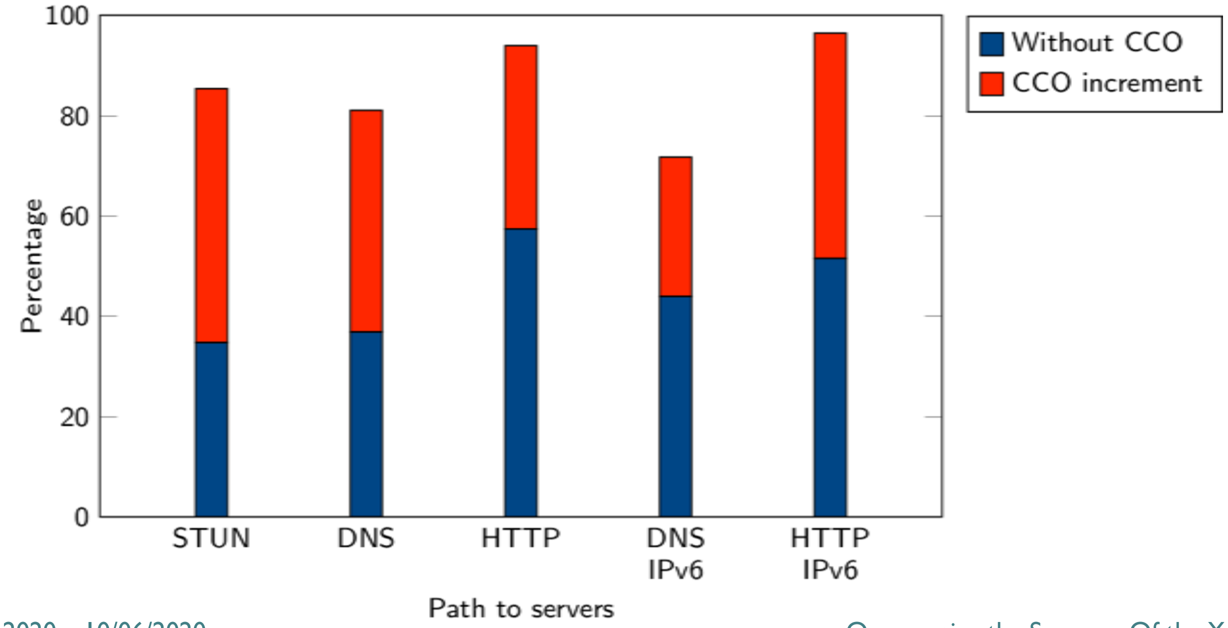
# Path Characterisation

- Any CS, **Correct CS only, Correct CS or IP payload CS categories**
  - Can be traversed by UDP-O according to original specifications
- **IP payload CS and Compensated CS only categories**
  - Can be traversed only using CCO



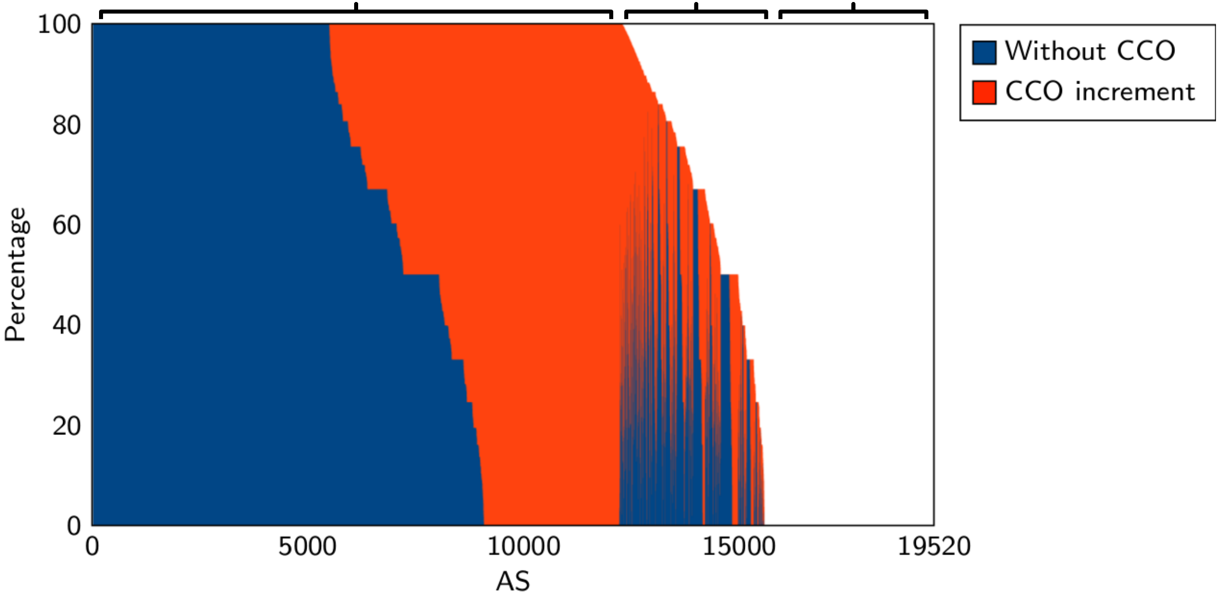
# Path Traversal Using CCO

- **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



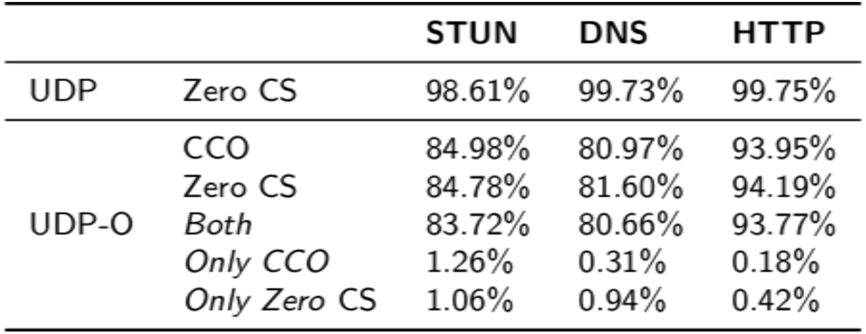
# Path Traversal Using CCO per AS

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**
  - A small percentage of paths can be traversed only using 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

# Conclusions and Future Work

- **First analysis of UDP-O path pathologies**
  - Implications on UDP-O design and deployment
- 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

- Validate our results on a larger dataset
  - Scans over other UDP protocols (on IPv4 full range and IPv6 target lists)
- Focus our measurements on edge networks
  - Release a tool to measure UDP-O without root privileges

# Thank you

**Tracemore**

http://www.middleboxes.org/tracemore

*Questions, comments, etc*

**Raffaele Zullo**
<raffaele.zullo@gmail.com>
<raffaele@erg.abdn.ac.uk>

# References

[1] J. Touch, "Transport options for UDP", 2019, IETF Internet draft draft-touch-tsvwg-udpoptions

https://datatracker.ietf.org/doc/draft-touch-tsvwg-udp-options/

[2] 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/

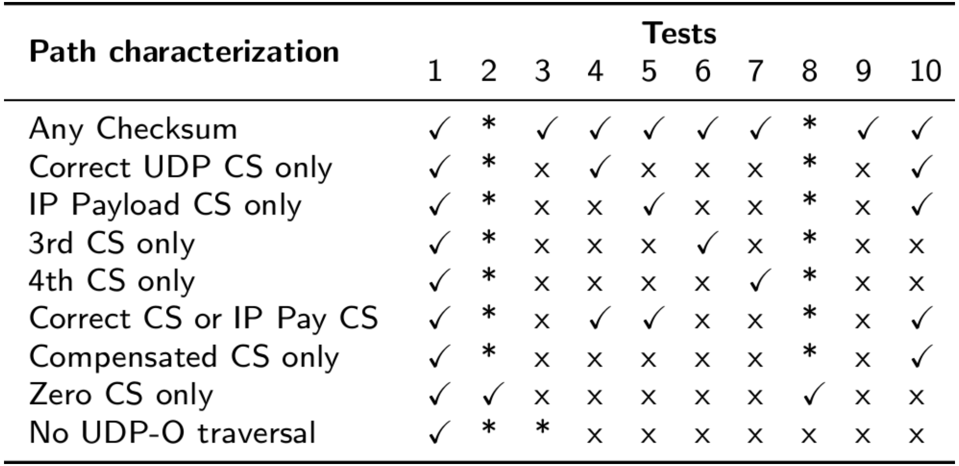
[3] "BSD Revision 334705", 2018

https://svnweb.freebsd.org/base?view=revision&revision=334705

[4] G. Fairhurst, T. Jones, and R. Zullo, "A Tale of Two Checksums" 2018, IETF

# Path Characterisation

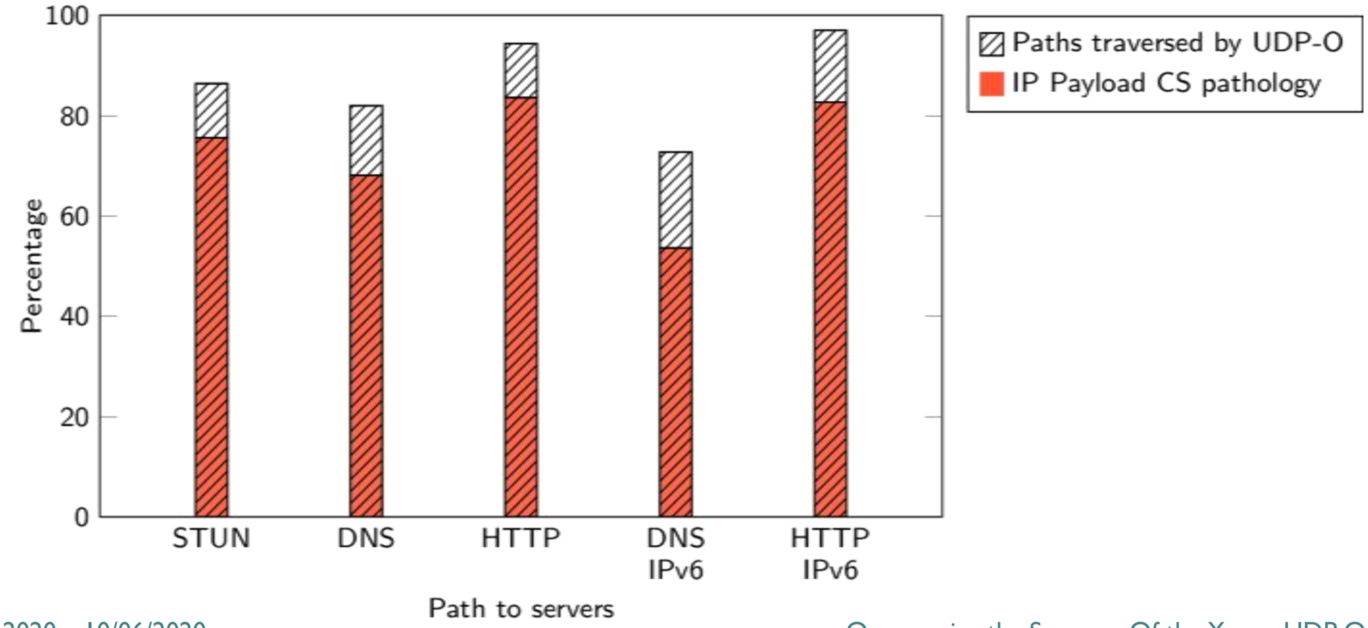
- Each packet in the test suite provide information about the path
- Only their combination can highlight the pathology or pathologies that affect the path



| **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 | ✓ | * | * | × | × | × | × | × | × | × |

# 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)


Percentage
STUN
DNS
HTTP
DNS IPv6
HTTP IPv6
Path to servers
▨ Paths traversed by UDP-O
🟥 IP Payload CS pathology