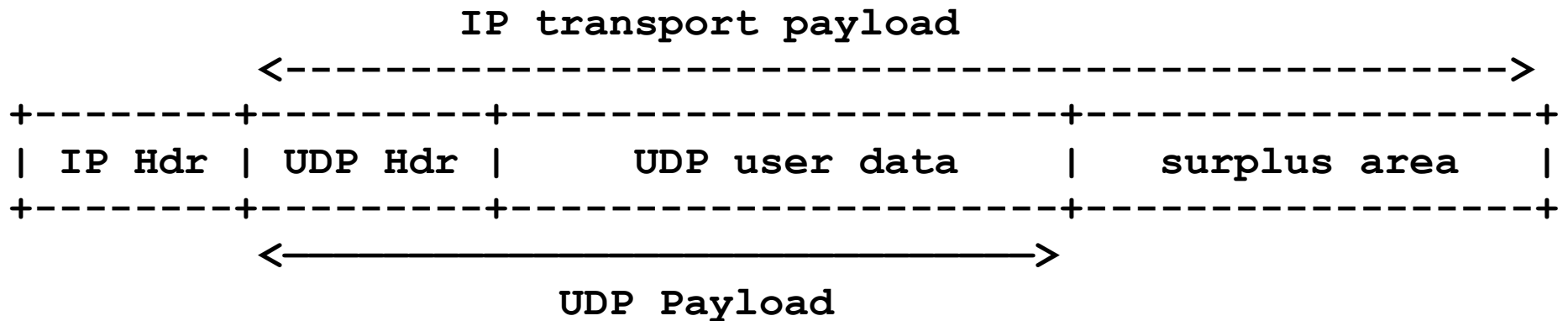


# Progress on UDP Options Implementation

Gorry Fairhurst, Tom Jones  
TSVWG

\*@erg.abdn.ac.uk

# UDP Option Area



**RFC793**

# Implementation in FreeBSD

0*	-	End of Options List (EOL)	- Implemented
1*	-	No operation (NOP)	- Implemented
2*	X 4	Option checksum (OCS)	- Implemented (CCO)
5*	4	Maximum segment size (MSS)	- Implemented
7	10	Timestamps (TIME)	- Implemented

## Added:

9	6	Request (REQ)	- Implemented
10	6	Response (RES)	- Implemented

## To do:

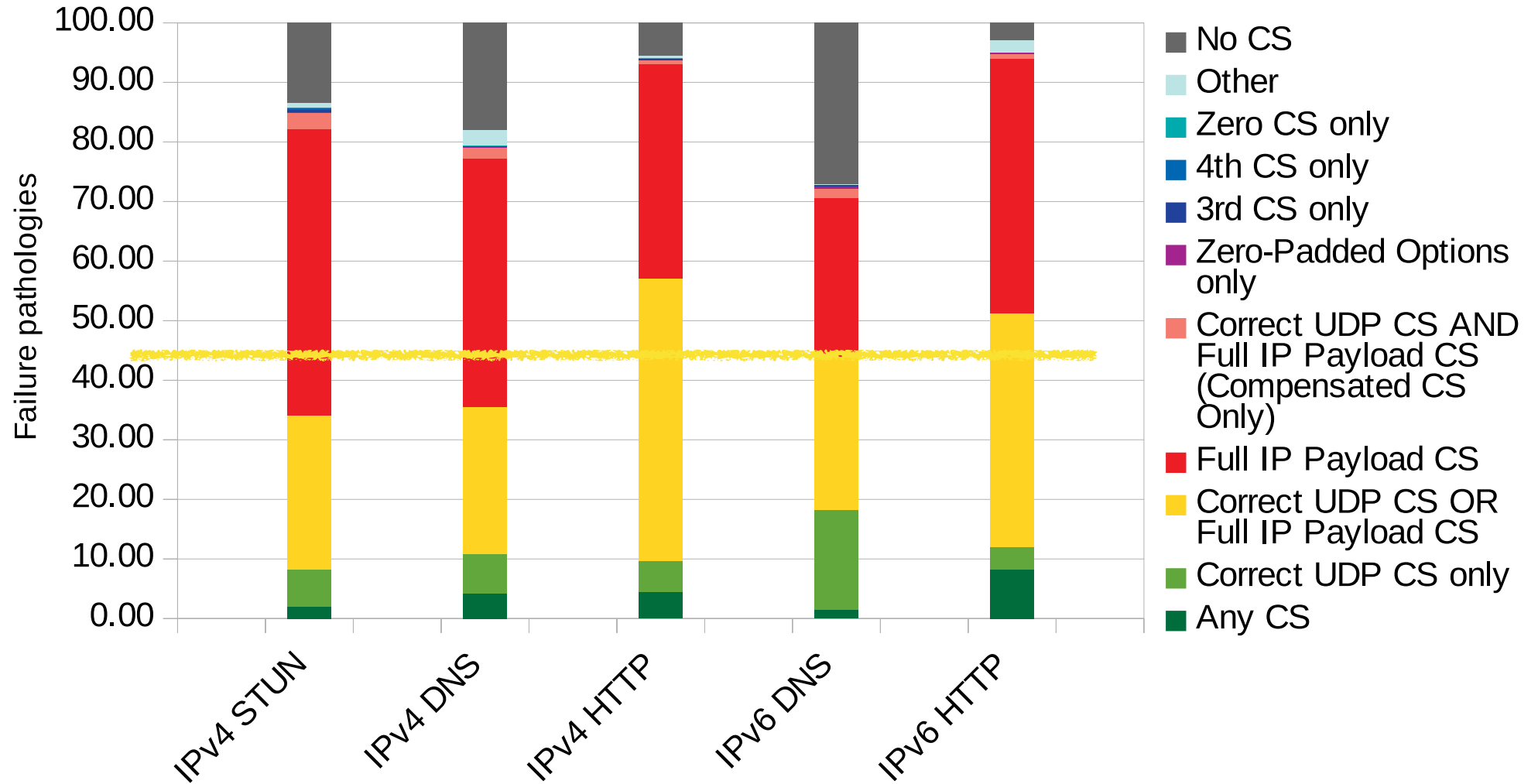
3*	4	Alternate checksum (ACS)	- Not yet implemented
----	---	--------------------------	-----------------------

Receiver has to know to use this.

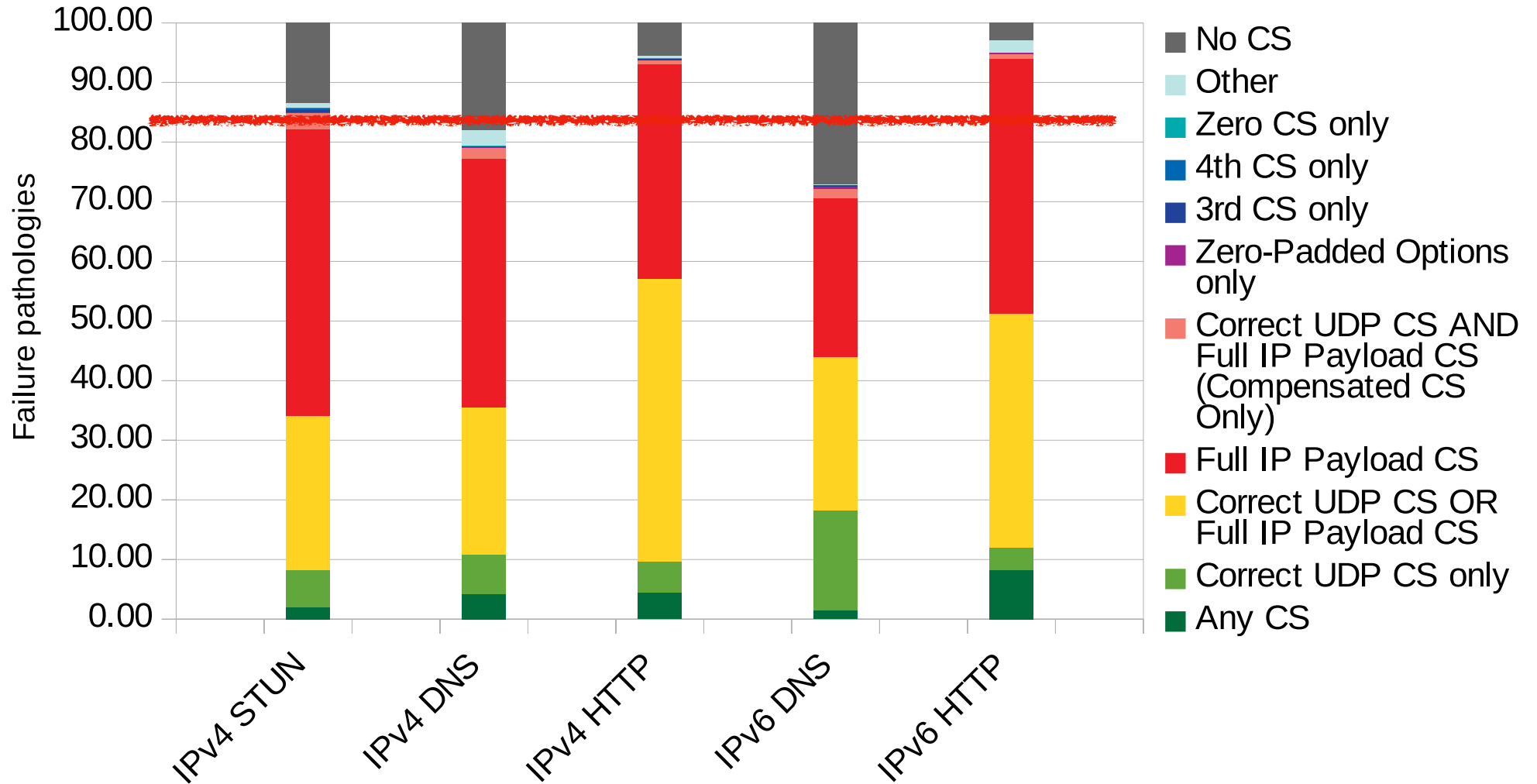
We need to agree on CRC Algorithm (Should we be using the CRC32c?).

(This does not conflict with the CCO).

# Results (see MAP-RG)



# CCO helps



# Middlebox Problems

- The magic is in draft-ietf-fairhurst-udp-options-cco
- Most (not all) checksum issues can be fixed by CCO
- Not fixed:
  - Boxes that only allow 0s as options space
  - Only passes IP payload length == UDP Length,
  - Also some more bizarre pseudo header errors.

# The CCO method

CCO could be a direct replacement for the OCS checksum

Should it be an *option* or *always required*?

... Options have an upgrade path to when more of the Internet supports UDP

... Should be required in the header, more efficient.

In our case, OCS will be the default anyway.

# Implementation in FreeBSD

0*	-	End of Options List (EOL)	- Implemented
1*	-	No operation (NOP)	- Implemented
2*	X 4	Option checksum (OCS)	- Implemented (CCO)
5*	4	Maximum segment size (MSS)	- Implemented
7	10	Timestamps (TIME)	- Implemented

## Added:

9	6	Request (REQ)	- Implemented
10	6	Response (RES)	- Implemented

## To do:

3*	4	Alternate checksum (ACS)	- Not yet implemented
----	---	--------------------------	-----------------------

Receiver has to know to use this.

We need to agree on CRC Algorithm (Should we be using the CRC32c?).

(This does not conflict with the CCO).



# Implementation Topics

4\*        4            Lite (LITE)

LITE - Specification for LITE is complicated, but possible.

6\*        8/10        Fragmentation (FRAG)

FRAG - Support for fragments in transport and network protocols are difficult to handle, partly because of need to consider attack vectors and partly because of need to manage reassembly buffers. It isn't something an endpoint would enable as default.

The current spec puts data in the UDP payload, which does not seem correct.

8        (varies)    Authentication and Encryption (AE)

AE - Underspecified.

*Does anyone wish to implement these? (please say)*

# What next?

Looking forward to next revision of the draft!

Implementation work to do:

2*	2	Option checksum (OCS)	- Implement CCO
3*	4	Alternate checksum (ACS)	- CRC32c

**This work is partially supported by the European Commission under Horizon 2020 grant agreement no. 688421 Measurement and Architecture for a Middleboxed Internet (MAMI).**