TCP TIME WAIT

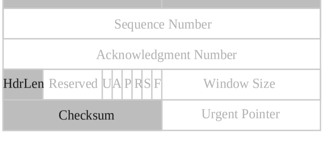
```
Active Internet connections (including servers)
Proto Local Address       Foreign Address    State
tcp   172.30.0.77.65291   172.30.0.161.9     TIME_WAIT
```

- "Connection" on client remains in TIME WAIT for twice the maximum segment lifetime
- Guards against old packets in the network

Graceful termination

- How to acknowledge the final packet ?
- Send an acknowledge ?
- But then that is the final packet ...
  - How is that acknowledged?
- Eventually, simply stop!

TCP (concluded)



Source Port | Destination Port

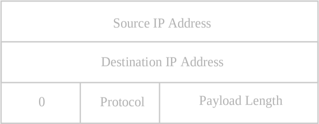
Sequence Number

Acknowledgment Number

HdrLen | Reserved | U | A | P | R | S | F | Window Size

Checksum | Urgent Pointer

- Source Port
- Destination Port
  - Transport address
- Checksum
  - As for UDP
    - includes pseudo-header



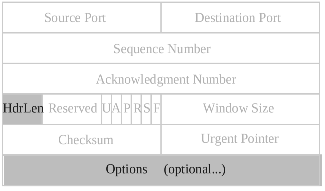
Source IP Address

Destination IP Address

0 | Protocol | Payload Length

  - But not optional.

TCP Options



- ◇ Header Length
  - ○ Allows size of options to be calculated
    - ▹ Counts 32 bit words in header
    - ▹ Minimum value 5 (20 bytes)
    - ▹ Maximum value 15 (60 bytes)
      - ▫ Max 40 bytes of options

- ◇ Options
  - ○ Data not needed in every packet
  - ○ Extensions to TCP

TCP Options

```
172.30.0.77.65486 > 172.30.0.161.9: S
  334775908:334775908(0) win 16384
  <mss 1460,nop,wscale 0,nop,nop,
                       timestamp 11188 0>
```

- ◇ 4 different options used

- ◇ NOP
  - ○ padding (keeps alignment of other options)

- ◇ MSS
- ◇ WSCALE
- ◇ TIMESTAMP

TCP Specification

Options: variable

Options may occupy space at the end of the TCP header
and are a multiple of 8 bits in length.

All options are included in the checksum.

An option may begin on any octet boundary.

There are two cases for the format of an option:

Case 1: A single octet of option-kind.

Case 2: An octet of option-kind, an octet of option-length,
and the actual option-data octets.

The option-length counts the two octets of option-kind and option-length
as well as the option-data octets.

Note that the list of options may be shorter than the data offset field
might imply.

The content of the header beyond the End-of-Option option must be
header padding (i.e., zero).

TCP Specification

Currently defined options include:

Kind	Length	Meaning
0	-	End of option list.
1	-	No-Operation.
2	4	Maximum Segment Size.

- ◇ End of Option
  - ◦ 1 byte 0
    - ▹ What follows is padding
  - ◦ Padding required to be 0
- ◇ No Operation
  - ◦ 1 byte 1

TCP Max Segment Size Opt

- ◇ Allows one TCP to tell the other
  the maximum segment size to send
  - ◦ Segment ~= packet
  - ◦ but one segment may be fragmented into several packets
- ◇ Maximum Segment Size
  - ▹ 4 bytes
  - ◦ `Kind` 2
  - ◦ `Length` 4
  - ◦ `Value (2 bytes)` MSS

TCP MSS Practice

- ◇ TCP will attempt to avoid fragmentation
  - ◦ By advising peer of the maximum segment size
  - ◦ Segment size (within window) not usually important
- ◇ Works correctly only when low MTU links
- ◇ are directly connected to hosts
- ◇ Largely overtaken by PMTUD now
  - ▫ Or should be
- ◇ Option only permitted when SYN is set
  - ◦ Data not required in every packet

TCP Window Size Limitation

- Maximum window size is 64K bytes (-1)
- Once "window" size bytes of data have been sent,
  TCP must stall and wait for ACK of
  (at least) first segment before sending more
- At 56K bits per second
      - (original arpanet)
  - sending 64K bytes takes more than 9 seconds
- As long as RTT is less than 9 seconds,
      - (Round Trip Time)
  - TCP need never stall

TCP Window Size Limitation

- At 100 M bits/second
      - (current network speeds)
  - sending 64K bytes
    - takes about 5.5 milliseconds
- So RTT must be less than 5.5 milliseconds
  - to avoid stall
- OK for LAN
- WAN might easily have RTT of 500ms
  - TCP can only use about 1% of available bandwidth
- Want to extend TCP
  - to allow higher throughput
  - Add an option

TCP Window Scale Option

- Allows TCP systems to
  - discover that both implement the option
  - specify that all window values
    - should be multiplied by a power of two
- Limited to 2^{14}
  - So, max window is $(64K - 1) * 2^{14}$ ~= 1GB
  - Thus max of $^1/_4$ of the entire window space
- If both TCPs send wscale option
  - window scaling protocol is used
- Each TCP indicates
  - how much its window should be scaled
- If either does not send option
  - No window scaling
  - In either direction

TCP Option Usage

```
172.30.0.77.65486 > 172.30.0.161.9: S
  334775908:334775908(0) win 16384
  <mss 1460,nop,wscale 0,nop,nop,
                       timestamp 11188 0>
```

- WScale == 0
  - Scale window size by 2^0
    - $2^0 == 1$
    - So, no scaling
- Why is option included?
  - So other host knows
    - this TCP supports the window scale option