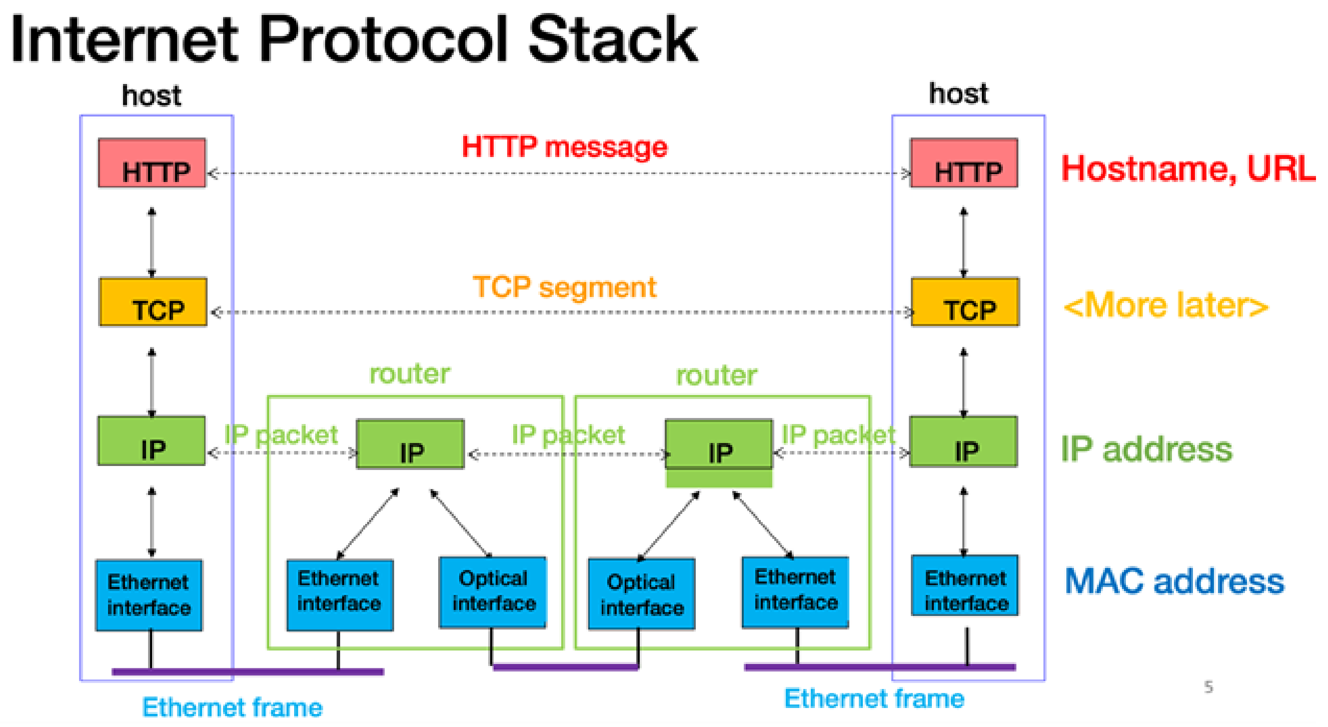
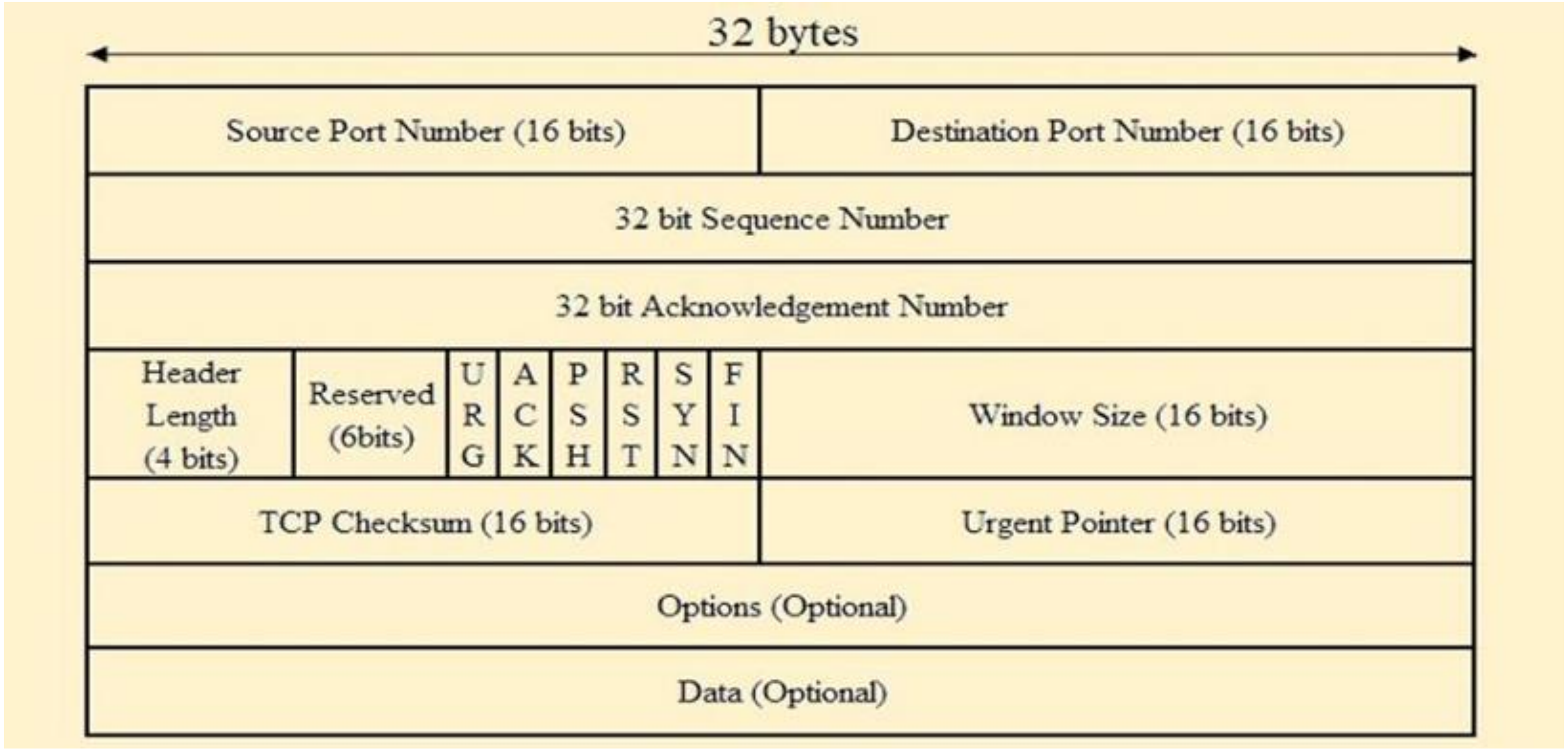
COS 316 Precept #5

Testing & Benchmarking

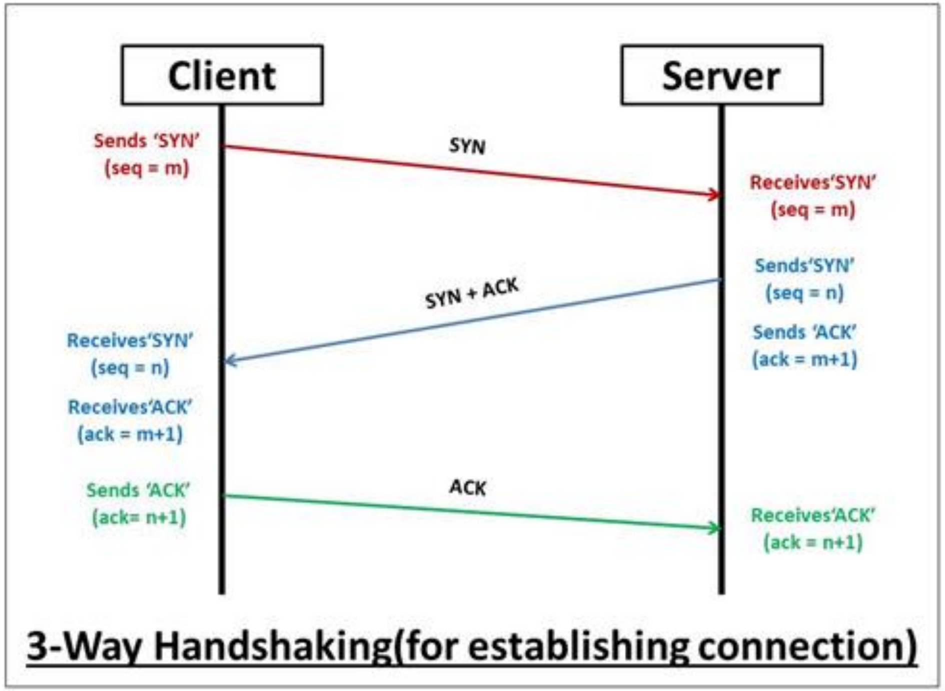
TCP is about streams



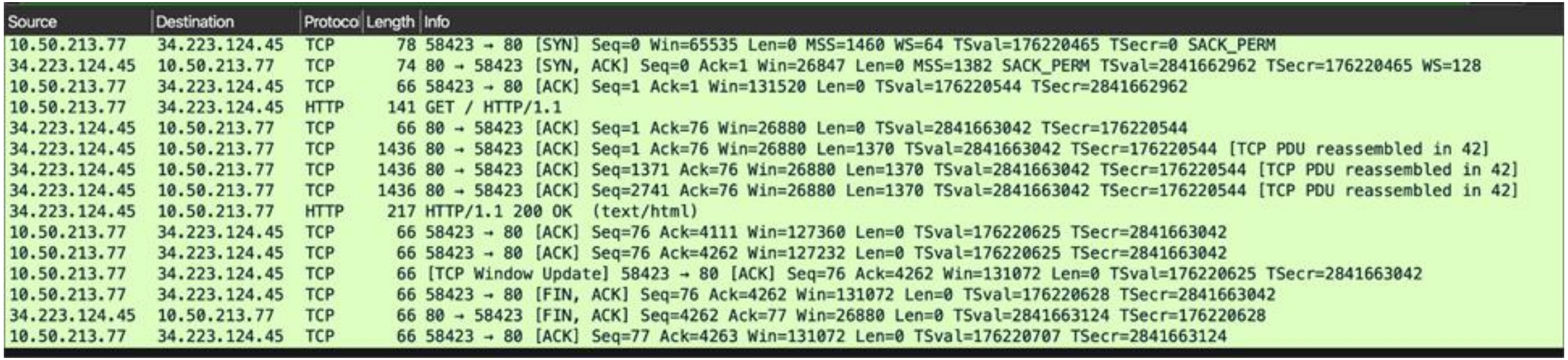
The TCP header



3-way handshake



Example



```
Source          Destination     Protoco Length Info
10.50.213.77    34.223.124.45   TCP       78 58423 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TSval=176220465 TSecr=0 SACK_PERM
34.223.124.45   10.50.213.77    TCP       74 80 → 58423 [SYN, ACK] Seq=0 Ack=1 Win=26847 Len=0 MSS=1382 SACK_PERM TSval=2841662962 TSecr=176220465 WS=128
10.50.213.77    34.223.124.45   TCP       66 58423 → 80 [ACK] Seq=1 Ack=1 Win=131520 Len=0 TSval=176220544 TSecr=2841662962
10.50.213.77    34.223.124.45   HTTP     141 GET / HTTP/1.1
34.223.124.45   10.50.213.77    TCP       66 80 → 58423 [ACK] Seq=1 Ack=76 Win=26880 Len=0 TSval=2841663042 TSecr=176220544
34.223.124.45   10.50.213.77    TCP     1436 80 → 58423 [ACK] Seq=1 Ack=76 Win=26880 Len=1370 TSval=2841663042 TSecr=176220544 [TCP PDU reassembled in 42]
34.223.124.45   10.50.213.77    TCP     1436 80 → 58423 [ACK] Seq=1371 Ack=76 Win=26880 Len=1370 TSval=2841663042 TSecr=176220544 [TCP PDU reassembled in 42]
34.223.124.45   10.50.213.77    TCP     1436 80 → 58423 [ACK] Seq=2741 Ack=76 Win=26880 Len=1370 TSval=2841663042 TSecr=176220544 [TCP PDU reassembled in 42]
34.223.124.45   10.50.213.77    HTTP     217 HTTP/1.1 200 OK  (text/html)
10.50.213.77    34.223.124.45   TCP       66 58423 → 80 [ACK] Seq=76 Ack=4111 Win=127360 Len=0 TSval=176220625 TSecr=2841663042
10.50.213.77    34.223.124.45   TCP       66 58423 → 80 [ACK] Seq=76 Ack=4262 Win=127232 Len=0 TSval=176220625 TSecr=2841663042
10.50.213.77    34.223.124.45   TCP       66 [TCP Window Update] 58423 → 80 [ACK] Seq=76 Ack=4262 Win=131072 Len=0 TSval=176220625 TSecr=2841663042
10.50.213.77    34.223.124.45   TCP       66 58423 → 80 [FIN, ACK] Seq=76 Ack=4262 Win=131072 Len=0 TSval=176220628 TSecr=2841663042
34.223.124.45   10.50.213.77    TCP       66 80 → 58423 [FIN, ACK] Seq=4262 Ack=77 Win=26880 Len=0 TSval=2841663124 TSecr=176220628
10.50.213.77    34.223.124.45   TCP       66 58423 → 80 [ACK] Seq=77 Ack=4263 Win=131072 Len=0 TSval=176220707 TSecr=2841663124
```

Overview

- What is *testing*?
  - evaluation of software against user requirements & systems specs
  - identify defects in software - show the presence of bugs, but not their absence

- What is *benchmarking*?
  - evaluation of system performance - time (CPU vs wall clock), memory, etc.

Testing - Basic Approach in Go

- Source files and associated test files are placed in the same package/folder

- The name of the test file for any given source file is **`_test.go`**
  - E.g., **`router.go`** and **`router_test.go`**

- Import “**`testing`**”

- Test functions need to have the “**`Test`**” prefix, and the next character in the function name should be capitalized

Benchmarking - Basic Approach in Go

- Benchmarks also reside in the **`_test.go`** files
- Import “**`testing`**”
- Benchmark functions need to have the “**`Benchmark`**” prefix, and the next character in the function name should be capitalized

Benchmark - Exercises

- How to eliminate certain code in benchmarks?
  - **`b.ResetTimer(), b.StartTimer(), b.StopTimer()`**
- How to benchmark specific functions:
  - **`go test --bench=Fib20`**
- How to show memory allocations?
  - **`go test --bench=. --benchmem`**
    or
  - **`b.ReportAllocs()`**