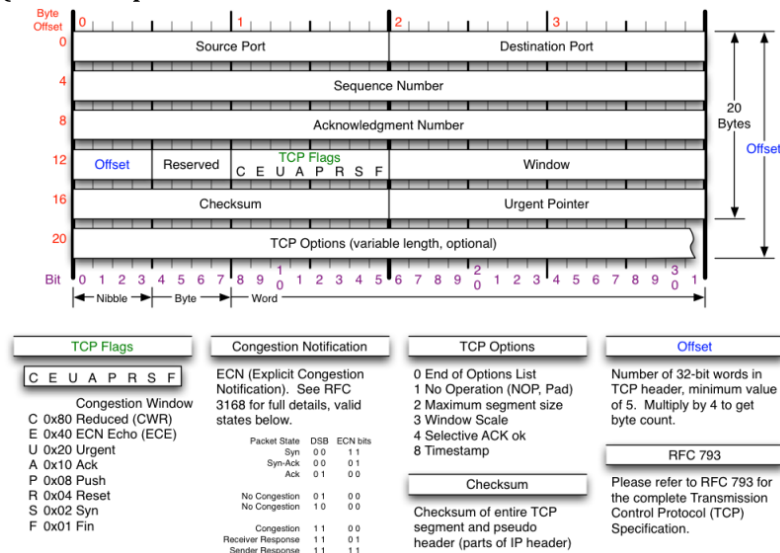


## 1> Protocol at the TCP level - Handshaking

TCP Packets: What is "SYN. SYN-ACK. ACK" ?

What is the sequence number and what is it used for? What is its initial value & why?

(I see the port number but where is the machine's IP address?)



## 2> Examples of Denial of Service

SYN flood

Distributed DOS

"Internet of Things DOS"

## 3> Which TCP client or server call will result in the first "SYN" packet?

## 4> TCP Handshaking and the speed of light

The moon is 1.3 light seconds distant. The TCP client is on the Earth and a lunar console runs a TCP server. Assume a new TCP connection is required each time.

3.1 Save the astronaut. How many seconds elapse between wanting to send a CLOSEAIRLOCK message and the server receiving the data?

```
fd=socket(...)
connect(fd,...,...)
write(fd,"CLOSEAIRLOCK!",13);
```

3.2 How many seconds elapse between requesting data from the server and receiving the result?

```
fd=socket(...)
connect(fd,...,...)
write(fd,"READTEMP!",9);
bytes=read(fd,buffer,256);
```

## 5> TCP and web performance

HTTP/1.0

If the client-server round trip time is 10 ms. What is the minimum time required to display a page with an image? Assume HTTP/1.0 (and that the image requires a separate request).

## 6> Better... Faster...

Performance improvements in HTTP/1.1

Improvements in HTTP/2.0

Why did Google create QUIC ?

## 7> Remote Procedure Calls

```
void updateScoreBoard(char*name, int score) {
    char* mesg; // todo: error checking!
    asprintf(&mesg, "newscore,%s,%d",name,score);

    write( fd, mesg, strlen(mesg+1));
    free(mesg);
    // Why did I also send the null byte?
}
// You could also send the message size
// My protocol! So I'll choose bigendian binary format
uint16_t mesglen = htons( strlen(mesg) );
write( fd, &mesglen, sizeof(mesglen) );
write( fd, mesg , strlen(mesg) );
```

## 8> Subverting protocols

Case study: Heartbleed April 2014

```
/* simplified */
sock_fd = accept(server_fd);
while(1) {
    secureread(sock_fd, &request, &reqsize);

    switch(request->request_type) {
        case HEARTBEAT:
            // echo the client message back
            securewrite(sock_fd,
                        request->content,
                        request->content_length
            );
            break;
        case (...):
            ...
            break;
    }
    free(mesg);
}
```