

# Internet Protocols

June 5, 2024

## Contents

<b>1</b>	<b>HTTP</b>	<b>1</b>
1.1	URLs . . . . .	1
1.2	Hypertext Transfer Protocol . . . . .	2
1.3	HTTP message structure . . . . .	2
<b>2</b>	<b>TCP/IP</b>	<b>2</b>
2.1	TCP sequence numbers . . . . .	3
<b>3</b>	<b>Throughput calculation</b>	<b>6</b>
3.1	Transmission delay . . . . .	6
3.2	Propagation delay . . . . .	6
3.3	Traffic intensity . . . . .	6
3.4	Throughput . . . . .	7

## 1 HTTP

This section is dedicated to the HTTP protocol and related topics.

### 1.1 URLs

URLs (Uniform Resource Locators) is a way to reference a resource usually a web address. The resource can also be things such as websites, video files, pictures, files, or IOT devices. URLs are build up in the following way `{http://www.site.com/index.php?page=17#article}` where `{http}` is the protocol, `{www.site.com}` is the domain, `{index.php}` is the path, `{page=17}` is the query string, and `{article}` is the fragment. URLs can only be sent as ascii which means that some characters and symbols must be converted into an ascii representation. example: `{Æsel? -> %C3%86sel}`.

## 1.2 Hypertext Transfer Protocol

HTTP is an application layer protocol designed for web-traffic, it uses a client/server model. The client, typically a browser, will usually take the following steps:

1. Send request (HTTP GET or similar), about a website
2. Await response
3. Parse response
4. Run steps 1-3 for any other required resource (js document, image, and/or style sheets)

the server will usually run the following steps:

1. Await request
2. Receive HTTP request from client
3. Handle request, and read requested file.
4. Send HTTP response containing requested file.

Generally a client will send a http request which is then handled by the server, then the server sends a response which is handled by the client. The server is usually identified with an ip address or a DNS name, along with a port number (443 for standard https implementations)

## 1.3 HTTP message structure

A HTTP message is made of two parts a header and a body, the header contains information about the message such as protocol flags. The body contains the actual data that needs to be transmitted.

# 2 TCP/IP

The TCP protocol stack is build up of 5 layers which are the following:

- Application layer
  - Protocols that support network applications such as FTP, SMTP, HTTP, and DNS.

- Transport layer
  - Transfers data from a process on end-system A to end-system B through TCP or UDP
- Network layer
  - Routing and transfer of packages from source to destination
- Link layer
  - Data transfer of direct network neighbors connected via a single link (ethernet, WIFI, PPP, or SONET)
- Physical layer
  - The actual wire used to transfer the electrical signal that translates to bits.

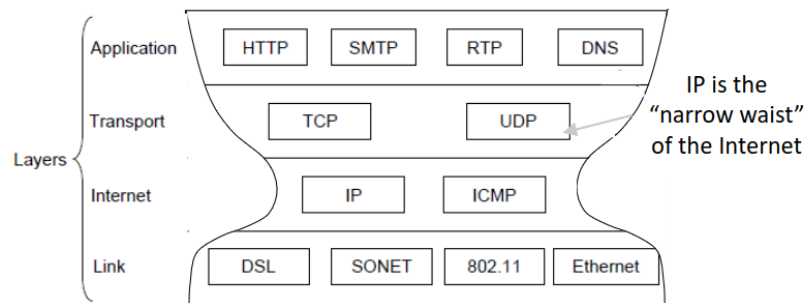


Figure 1: reference for TCP/IP protocol stack

The TCP header is built up of the following parts:

## 2.1 TCP sequence numbers

The ACK number can be found by taking the sum of the sequence number and the package length, the following sequence number will then be the ACK number, this cycle continues until the TCP connection is closed.

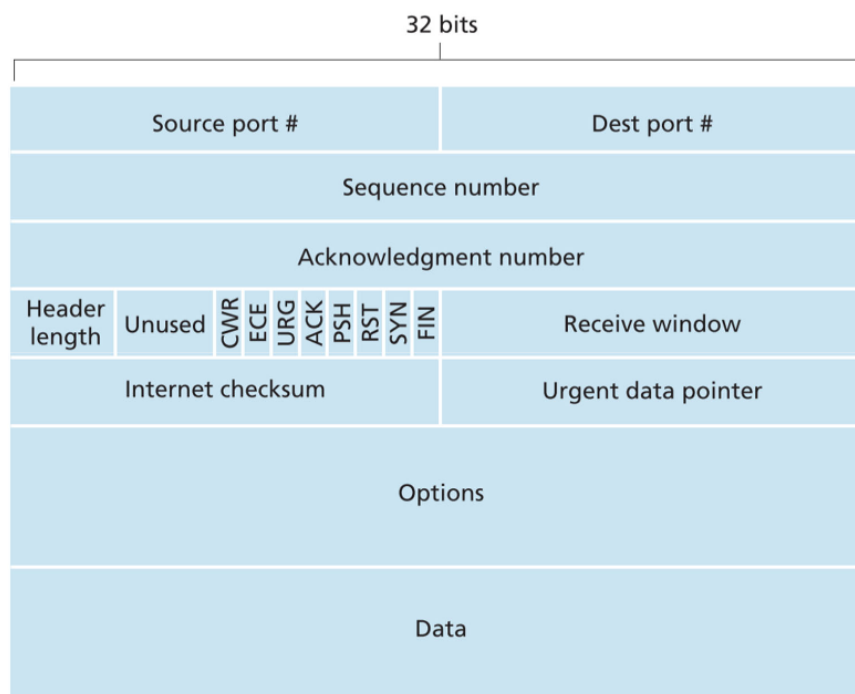


Figure 2: TCP header structure

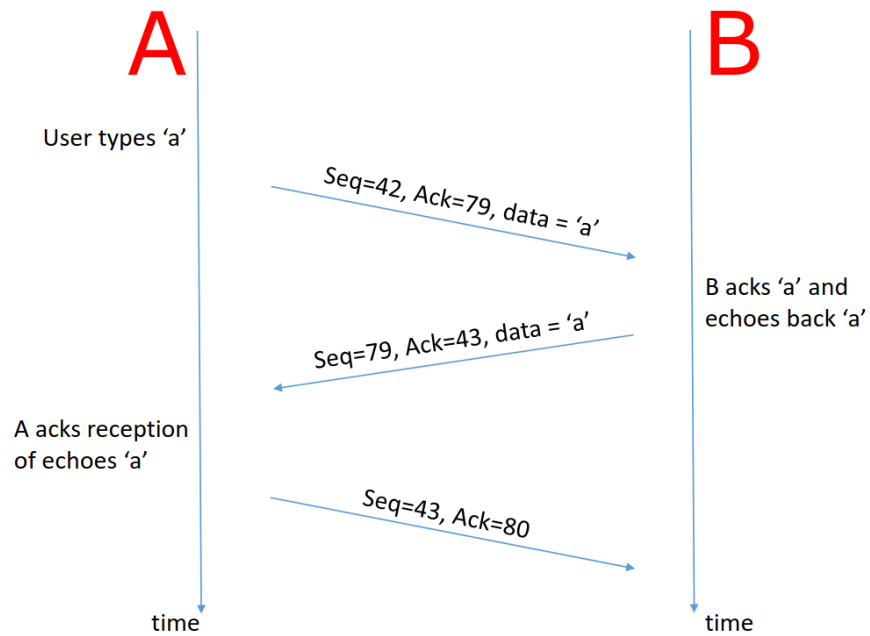


Figure 3: Diagram of TCP connection

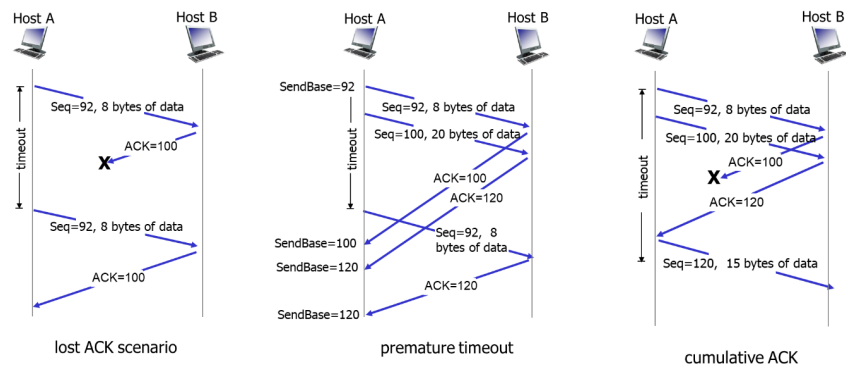


Figure 4: TCP transmission scenario examples

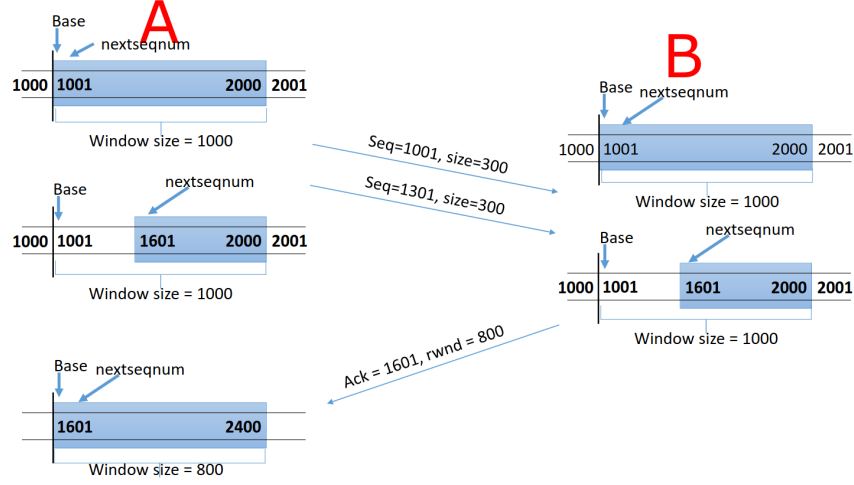


Figure 5: TCP transmission example with ACK and SEQ numbers

### 3 Throughput calculation

#### 3.1 Transmission delay

transmission delay is calculated with the following equation:

$$d_{trans} = \frac{L}{R} \quad (1)$$

Where L is the length of the packet in bits, and R is the transmission rate in bits.

#### 3.2 Propagation delay

Propagation delay is calculated with the following equation:

$$d = \frac{d}{s} \quad (2)$$

Where d is the distance of the physical link, and s being the speed of data transfer through the physical link (just barely the speed of light)

#### 3.3 Traffic intensity

traffic intensity is calculated with the following equations:

$$\frac{L \cdot a}{R} \quad (3)$$

where  $L$  is the length of the package in bits,  $a$  being average rate of packages being delivered, and  $R$  being the transmission rate in bits per second.

### 3.4 Throughput

Throughput is determined by the slowest link in a network it can be expressed with the following statement:

$$\min(R_c, R_s, \frac{R}{n}) \quad (4)$$

$n$  being the number of connected clients,  $R$  being the transmission rate of the shared link between server and client,  $R_c$  being the transmission rate of the clients,  $R_s$  being the transmission rate of the server link.