

31338 Network Servers

32520 Systems Administration

Week 10

Web Servers

(Connect to Lab 10)

Professor Jian Zhang

School of Electrical & Data Engineering

University of Technology, Sydney

Q & A

- What happens when you type a URL in the browser and press enter? (Eg. <https://www.example.com/index.html>)

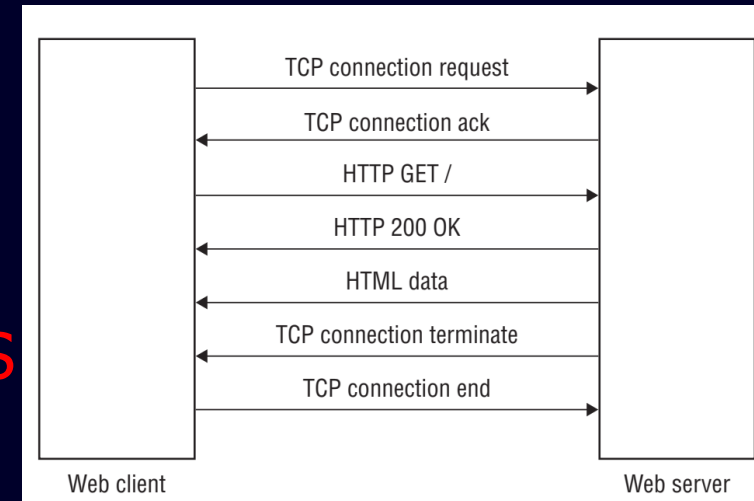
1. URL Parsing

2. DNS Resolution

- Check cache: Browser, OS, router, ISP
- DNS query (recursive query)

3. Establish a Connection: HTTP/HTTPS

- TCP/IP: 3-way handshake:
 - Sync (SYN), SYN-ACK, and ACK



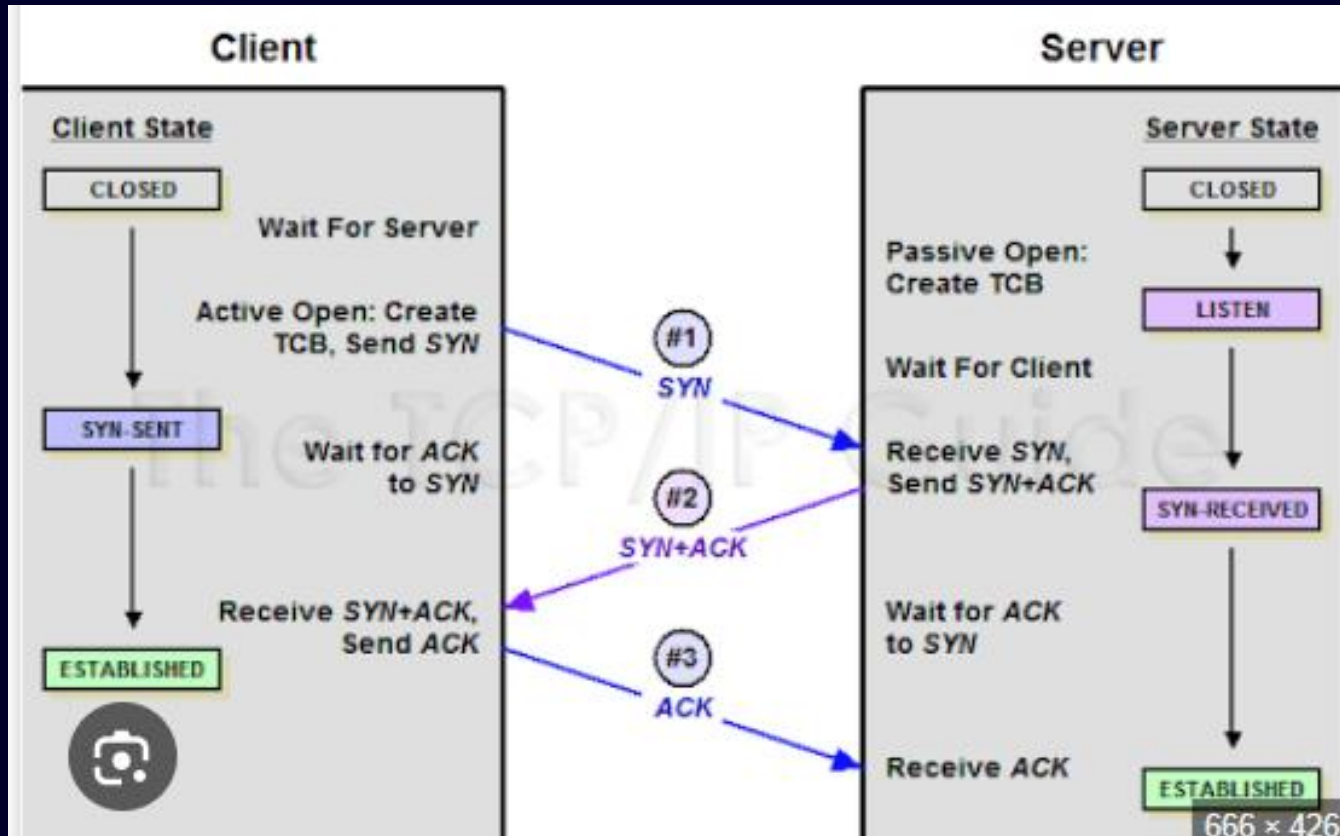
Basic HTTP session

4. Hypertext Transfer Protocol (HTTP)

- Browser:** sends an HTTP request to the web server GET/POST request
- Server:** Apache (open source), IIS (MS Webserver), PHP, JPS...
 - Receive the request and pass to handler (e.g., ASP.net, PHP, JSP...)
 - Handler (software package) process and return HTML
 - Send response code (e.g., 404 Not Found) and data (e.g., web page)
- Browser:** display HTML content

TCP/IP: 3-way handshake

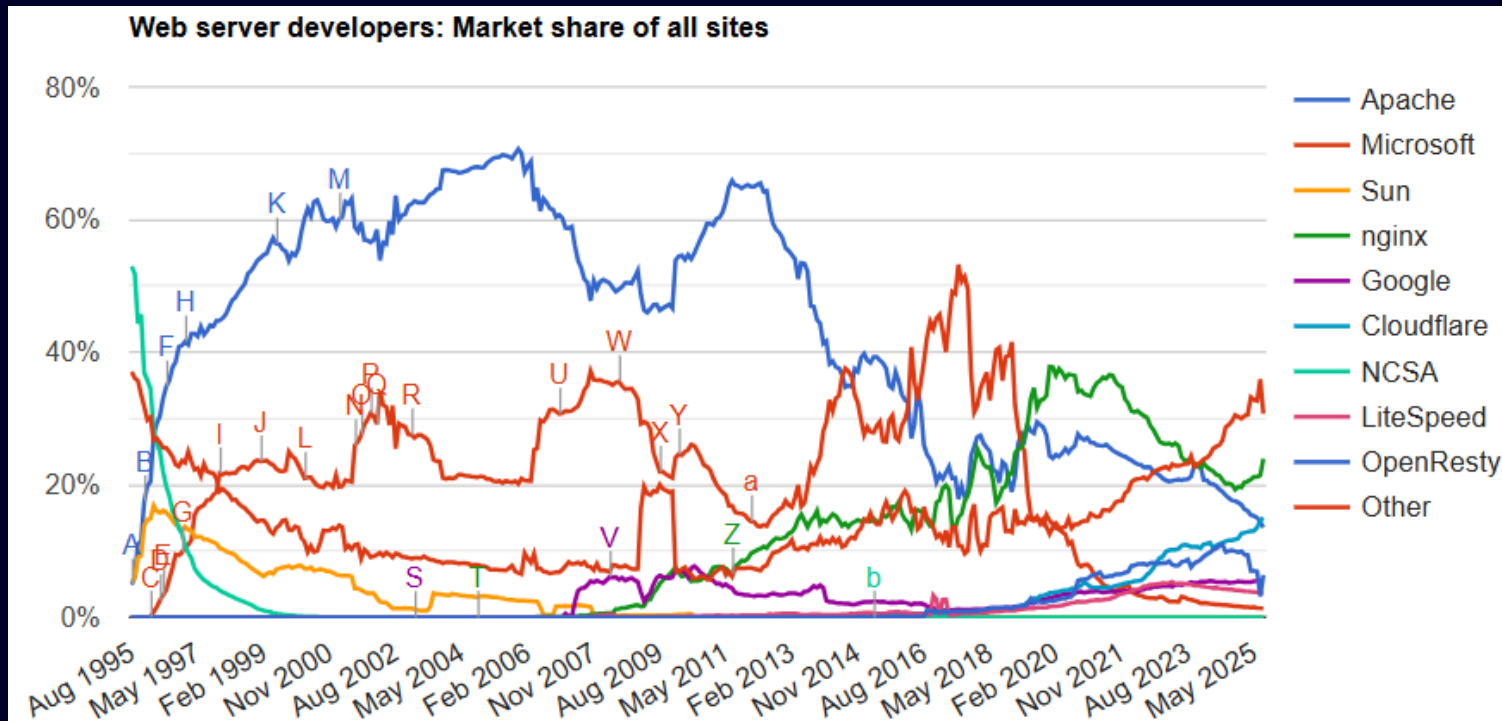
- TCP/IP: 3-way handshake
 - Sync (SYN), SYN-ACK, and ACK



<https://www.youtube.com/watch?v=xMtP5ZB3wSk&t=48>

Web servers

- 86.0% of all the websites is used with Unix
- Top 3 in Aug. 2024: **1.** Nginx (21.40%), **2.** Cloudflare (14.62%), **3.** Apache (14.37%), > 50% of the world's Internet web servers are UNIX



<https://www.netcraft.com/blog/august-2025-web-server-survey/>

Apache vs Nginx

- Share
 - Both are used by large Fortune 500 companies
 - Nginx market share has been steadily growing for years.
- Basic Architecture
 - Process driven, create a new thread for each request
 - Event driven, handle multiple requests within one thread
- Performance: both support static content
 - Support dynamic content within the server
 - Needs to pass requests to an external process that provides content.
- OS Support
 - All Unix-like, fully support Windows
 - Almost all Unix-like, Windows partially

Web Service Stack for Developers

- A very common installation for web is the “**LAMP**” stack or “**LEMP**”/ “**LNMP**”/ stack
 - **L**inux – operating system ☺
 - **A**pache – web server, **N**ginx, **E**ngine-X
 - **M**ySQL – database server
 - **P**HP – for applications and dynamic web pages

Often pre-bundled together.

Also available on **Windows** (“**WAMP**”) and other platforms

- LNMP/LNMP/LTMP: More options
 - Nginx/Tengine/OpenResty
 - MySQL/MariaDB/Percona
 - PHP/Perl/Python

Apache configuration (Lab 10a)

- Config directory is `/etc/httpd/conf`
 - Edit main Apache config file: `/etc/httpd/conf/httpd.conf`
 - consists of Apache directives and blocks `<Block> </Block>`
 - **Modify `ServerName`** and note down the Web documents in **`DocumentRoot`** directory located in `/var/www/html`
- Apache is a modular web server
 - modules provide **additional functionalities**, e.g. authentication, web programming languages, SSL, etc.
 - module configuration files: `ls -l /etc/httpd/conf.d/*.conf`
 - in main `httpd.conf` file, it says: **`Include conf.d/*.conf`**
- Log files (**recording**) in `/var/log/httpd/*`
 - **`access_log`** for web accesses logs, **`error_log`** for error messages

```
[root@site ~]# cat /etc/httpd/conf/httpd.conf
#
# This is the main Apache HTTP server configuration file. It contains the
# configuration directives that give the server its instructions.
# See <URL:http://httpd.apache.org/docs/2.4/> for detailed information.
# In particular, see
# <URL:http://httpd.apache.org/docs/2.4/mod/directives.html>
# for a discussion of each configuration directive.
```

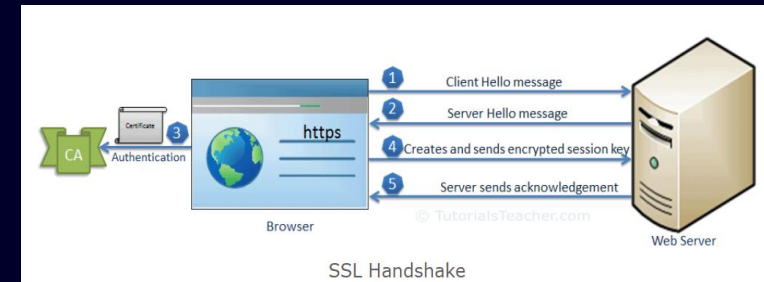
Web server security (lab 10b)

- Web servers frequently attacked
- Most vulnerabilities relate to server misconfiguration
 - **turn off** features you don't need
 - **don't run web server as root and don't run applications as the web server user if possible**
 - provided "**minimum**" access necessary to files/directories
 - consider **chroot environment** for public web servers
 - consider **masking** details of your server
 - monitor logs for **unusual** requests (did they succeed?)
- Vulnerabilities also often introduced in poor web application programming (e.g., injection attacks)

Conclusion: to develop a HTTPS=secure for the URL by using an SSL certificate

Secure Sockets Layer (Lab 10b)

- HTTPS appears when a website is secured by an SSL certificate
- Configuring Secure Sockets Layer (SSL*)
 - Certificate exchange (mutual authentication)
 - Encrypted network traffic
- Basic idea (handshake) – 5 steps:
 1. The client sends a "client hello" message.
 2. The server responds with a "server hello" message
 3. The client verifies the server's SSL certificate from CA (Certificate Authority) and authenticates the server.
 4. The client creates a session key, encrypts it with the server's public key and sends it to the server
 5. The server decrypts the session key with its private key and sends the Ack to the client encrypted with the session key.
- In the lab:
 1. Remove the existing SSL server key (private), its certificate and generate new ones. 1) `../private/localhost.key` and 2) `localhost.crt`
 2. Specify a new DocumentRoot for the SSL web server from a diff. location
 3. Edit `ssl.conf` file in Dir: `/etc/httpd/conf.d`. Uncomment DocumentRoot directive in `ssl.conf`, change it to point to a new directory `/var/www/secure`. You need to create it and add `index.html` file inside.



Virtual Hosts (Lab 10c)

- A single web server responds to multiple URLs
 - Add `<VirtualHost>` directive/contrainer in `httpd.conf`
 - `<VirtualHost _default_:443>`
 - `DocumentRoot "/home/httpd/ona"`
 - `ServerName www.openna.com`
 - `ServerAdmin admin@openna.com`
 - `ErrorLog /var/log/httpd/error_log`
 -
 - `</VirtualHost>`
 - If **exists**, the default virtual host is located at the `/var/www/html` directory
- Name-based (**better!**)
 - Single IP address
 - Selects pages to serve based on URL in HTTP GET request
 - Needs HTTP/1.1 (or later) to work
- IP-based
 - Server has multiple IP addresses, one for each virtual host
 - Selects pages to serve based on IP address
 - Wasteful of IP addresses

Virtual Hosts Configuration (Lab 10c)

Virtual Hosts Configuration Example

1. Specify a different *DocumentRoot* for the <virtualHost>
2. Put a new index.html file inside it for the virtual host

```
<VirtualHost *:80>

    DocumentRoot "/var/www/a"

    ServerName www.it.netserv.edu.au

</VirtualHost>

<VirtualHost *:80>

    DocumentRoot "/var/www/b"

    ServerName www2.it.netserv.edu.au

</VirtualHost>
```

Windows web server (Lab 10d)

- Windows since Windows NT 3.5 have own web server: **Internet Information Server (IIS)**
- Early versions vulnerable to attacks, insecure and performance issues. Dramatically improved recently.
- IIS from v8 onwards is a completely re-written, modular system.
- Windows Server 2019 & 2022 support IIS version 10.0
 - Note: IIS is not just web server, it provide File transfer server as well.
- IIS provides:
 - WWW (HTTP) server
 - File Transfer Protocol (FTP) server
- In the lab:
 - **Add** the “Web Server (IIS)” role and it associate features. As listed a. to d.
 - **Use** Tools menu, check the overall setting (GUI), select “*default web site*” and *follow the lab-handout to create web pages. . . .*

IIS vs Apache (Lab 10d)

- Basically, offers same services as Apache
- Adds some “window-ISM^{*}s” such as windows authentication
- Nice admin GUI, including optional Web based Mgt tools
- Provides application backend (ASP.NET, Azure)

Features	IIS	Apache
Supported OS	Windows	Linux, Unix, Windows, macOS
User support & fixes	Corporate support	Community support
Cost	Free, but bundled with Windows	Completely free
Development	Closed, proprietary	Open source
Security	Excellent	Good
Performance	Good	Good

Assignment 3 -- Final Skills Test

Tasks:

1. Setup static networking on both virtual machines according to the requirement, this task is like the question in Assessment 1 Skills Test, refer to Lab2b System updates, Lab3a static networking.
2. Set up Linux or Window VM as DHCP server and another VM as DHCP client, this task is like the question in Assessment Skills Test, refer to Lab 4abc.
3. Add users and groups, refer to Lab 5a, and using file permission to support file sharing amount users, refer to Lab7a Managing filesystems, including mounting and unmounting.
4. Configure Window server as master DNS server for a domain and add DNS records, accordingly, refer to Lab 6ab.
5. Add a second hard disk and create partitions according to the requirement, refer to Lab 2d Disk partitioning, and Lab 7a Managing filesystems including mounting and unmounting.
6. Configure NFS file sharing, refer to Lab9a NFS
7. Configure SMB file sharing, refer to Lab9b Samba
8. Configure the Linux server as web server, add a default webpage and configure a virtual host with different default web page, refer to Lab10a and Lab10c

Final Skills Test (FST)

- Students can select to either use a Lab PC or their own laptop.
- Students are required to show UTS student ID card and sign the attendee sheet.
- Students are required to turn off mobile phones and only allowed to have one window open with Canvas FST. Students can only bring printed journals and can't open e-copy from a new window.
- Any attempt to open any other windows/browsers or access sites like Chat/GPT or similar may be considered as an academic misconduct -- [Academic misconduct | University of Technology Sydney \(uts.edu.au\)](https://www.uts.edu.au/academic-misconduct)
- A misconduct may impact on the whole score of the student who studies for the subject.

What you need to do at home!

- Before test:
 - Import new virtual machines before the Final Skills Test!
 - Install related packages.
- At home: do your practices several times from scratch.
 - Configure the VMs with static IP address and make sure the two VMs can ping each other.
 - Add second hard drive, 1G.
 - Setup firewall.

Have a good journal with your final skills test