Module One

CSG1105 Applied Communications

Introduction

The Chicken or the Egg? Which came first?

- In dealing with this material, we are often faced with this problem. One or more of the topics uses concepts from another topic we have not yet covered. It may seem that we are continually returning to topics already covered, but it may be necessary to deal with interactions
- Progression through the material is not necesarilly linear: you may need to revisit earlier topics multiple times

What is a Web Browser?

- We're all familiar with a Web browser, be it on a PC, Tablet or phone. We use it to access information in the way of text, images, audio, video or even applications.
- A Web browser is a **HTTP Client**
- It communicates with a **Web Server, AKA a HTTP Sever**
- We will look at HTTP a little later

Client/Server 1

- Browsers, (Firefox, Chrome, Edge etc.) are Web clients and that Web sites host Web servers.
- So, what are clients and servers?
- Firstly, a negative definition: **A server is not hardware**
- A server is a **process**, running at an **advertised location** providing a **service**.
- A process is a **running program**. A single computer will have many running processes, some may be servers, some clients
- A **client** is a process that consumes the product of a service
- Generally, there are many clients using the service provided by a server simultaneously. Generally, client and server are on separate computers, but not always.

Client/Server 2

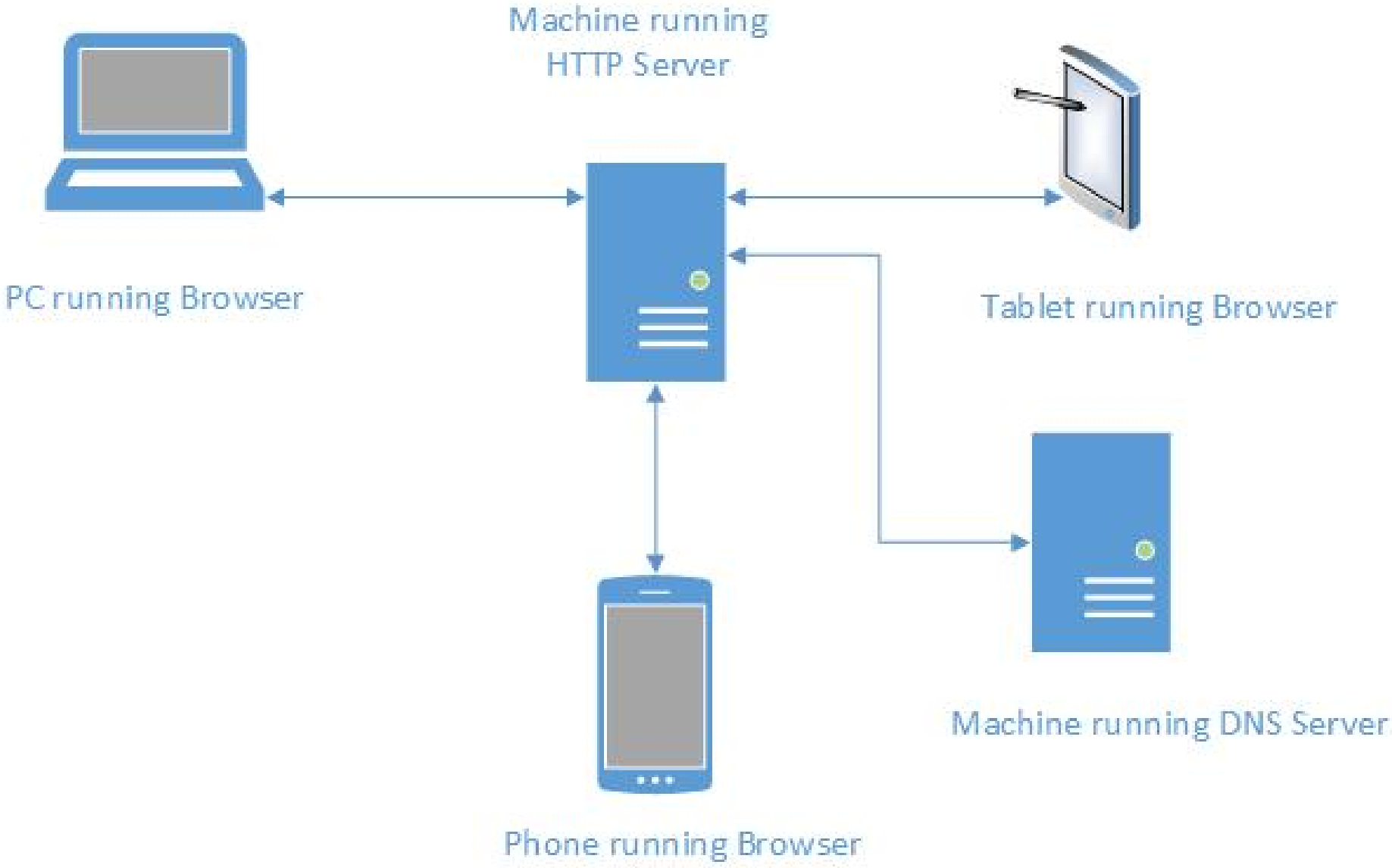


Figure 1: Client/Server

Client/Server 3

- There are thousands of types of servers
- We will be covering a number of commonly used servers in this unit
- Some examples are:
  - Web (http) server
  - Email (smtp) server
  - DNS server
  - Directory (ldap) server
  - File (smb) server

Protocols

- The clients and servers for any **service** communicate using an accepted **protocol**
- The previous slide identifies the protocols used (http,smtp,DNS,ldap,smb)
- A protocol is simply an accepted method of exchanging messages between the client and server
- **Question:** Who creates the protocols?

Standards Bodies

- There are **Open** and **Proprietary standards**
- Some standards are created by International groups, some by Industry bodies
- Many protocols used on the Internet are Open standards
- Three of the major international organizations are:
  - The IETF - Internet Engineering Task Force
  - The IEEE - Institute of Electrical and Electronic Engineers
  - The ISO - International Standards Organization

HTTP Protocol

- We will use some simple HTTP messages to illustrate a protocol in action
- If you don't specify a particular page on a Web server, it will return a default page, often **index.html**
- The browser will establish a connection to the web server (more on this later)

Request

- A request is sent:

```
GET / HTTP/1.1
```

- The request consists of:
  - the **Method** GET
  - the **path** "/"
  - the **Protocol Version** 1.1
  - This line will be followed by a list of headers detailing the client

Response

- If the page exists, it is returned with a protocol version from the server and a status code and message.
- This will be followed by headers giving details of the returned content.
- A new line then separates the headers from the content, in this case, a HTML page

```
HTTP/1.0 200 OK
Server: SimpleHTTP/0.6 Python/3.6.9
Date: Tue, 04 Feb 2020 06:11:46 GMT
Content-type: text/html
Content-Length: 91
Last-Modified: Tue, 04 Feb 2020 06:08:17e

<html>
```

Response for missing page

- If the page DOESN'T exist

```
GET /badpage.html HTTP/1.1
```

- Error message

```
HTTP/1.0 404 File not found
```

HTTP 2

- The HTTP protocol has many more features which may be found in RFC2616 found at https://tools.ietf.org/html/rfc2616
- What we have looked at is an example of a request/response style protocol
- Many of the protocols used on the Internet follow this mode of operation

Uniform Resource Locators (URLs)

- The HTTP protocol allows the browser to make requests and receive responses over an existing connection
- Earlier, we defined a server as a process running at an advertised location
- The method of specifying a location is a **Uniform Resource Locator (URL)**
- The formal specification of URLs is documented in RFC 3986 available at https://tools.ietf.org/html/rfc3986

URL Syntax

- A simple URL consists of three elements
  - The Scheme
  - The Authority
  - The Path
- Using the address from the previous slide as an example:

https://tools.ietf.org/html/rfc3986

Scheme

- The scheme is the **Protoco** used for the connection
- Some examples of protocols used in URLs
  - HTTP
  - FTP (file transfer protocol)
  - mailto (send to email address)
  - file (a file on the client computer)
- There are many other protocols, some of which are now obsolete e.g. gopher

Authority

- The Authority provides the location of the resource
- The syntax is userinfo@host:port
- For the moment, we are going to focus on the host and consider the others later The host portion of the Authority is a **Fully-Qualified Domain Name (FQDN)** which we will look at later

Path

- The Path locates where the resource is stored on the target machine
- The Web server has a tree structured set of folders to contain content
- The path specifies where in relation to a specified root folder
- As a machine may implement more than one web server, this is usually a virtual location

FQDN

- FQDN - **Fully Qualified Domain Name**
- An FQDN is the complete domain name computer on the Internet
- It consists of two parts:
  - The host name and
  - The domain name An example: www.curtin.edu.au

DNS - Domain Name System

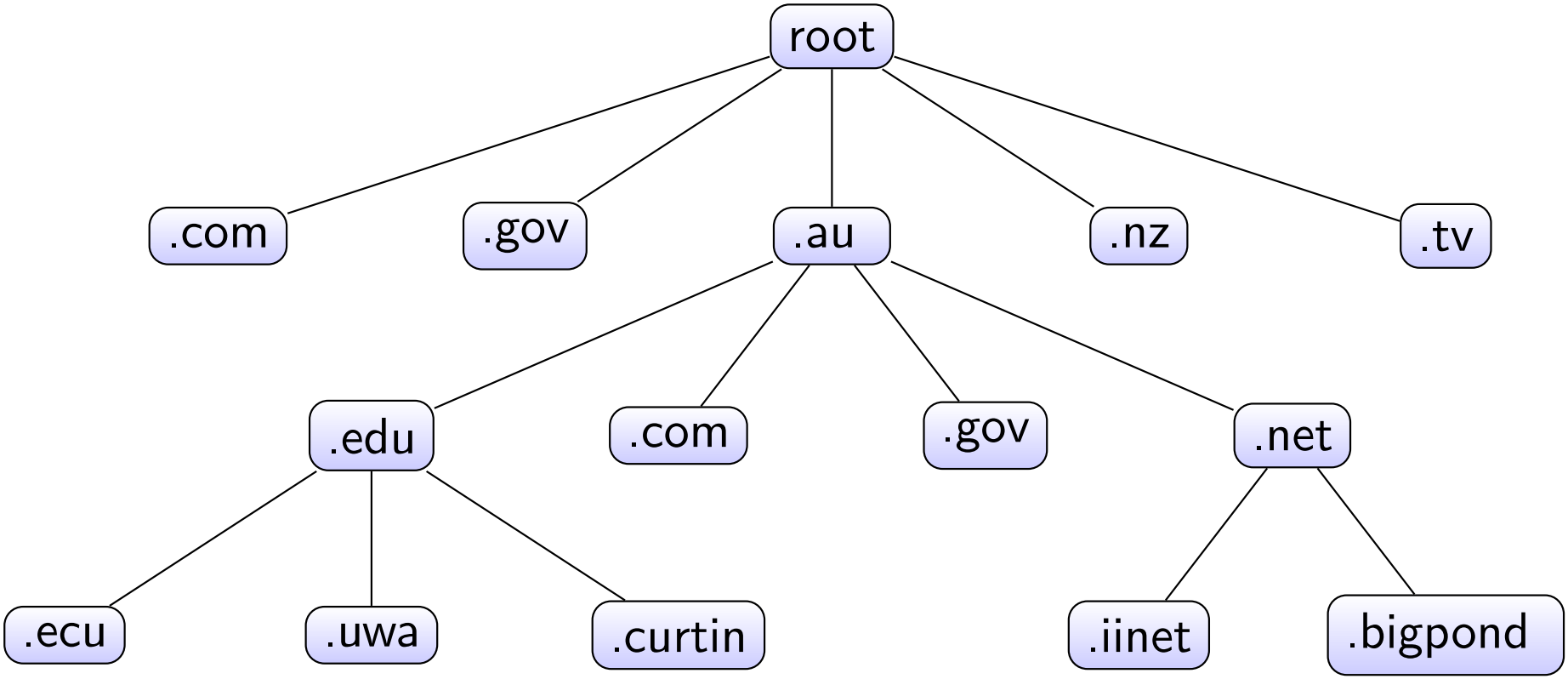
- Your phone has a Contacts app that don't have to remember them

Stan	0412 345 678
Lee	+64 0422 987 655

Stan	0412 345 678
Lee	+64 0422 987 65

- DNS is the "Contacts app of the Internet" A world-wide distributed directory service used to index computers, services and other resources
- DNS is a protocol with cleints and servers
- It is a tiered system, where each level has delegated control over a portion of the namespace
- One purpose is to translate human readable FQDN strings into a numerical address (IP address)

DNS


root
.com
.gov
.au
.nz
.tv
.edu
.com
.gov
.net
.ecu
.uwa
.curtin
.iinet
.bigpond

Tracing a DNS Query

1. Our browser sends a query to ECU's DNS server
2. ECU's web server queries the **.edu.au** root server for Curtin's DNS server address
3. The **.edu.au** server returns the address of Curtin's DNS server
4. The ECU DNS server sends a query to the Curtin DNS Server
5. The Curtin DNS server returns the address of `www.curtin.edu.au`
6. ECU's DNS server returns the address of `www.curtin.edu.au` to the client PC
7. The client PC can use the address to contact `www.curtin.edu.au`

DNS Query

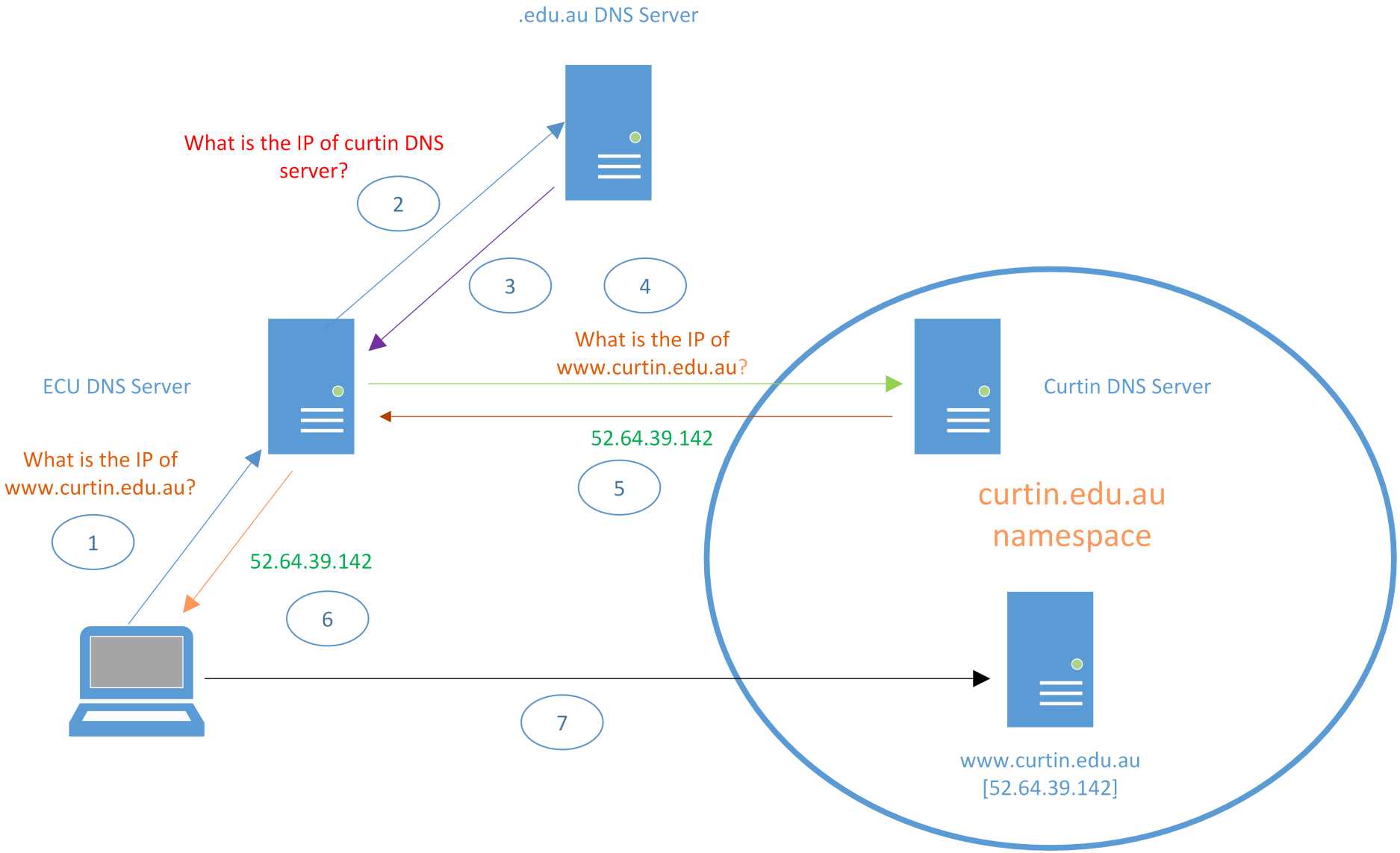


Figure 2: DNS Query