

# COMS3200 Study Notes

Brae

Semester 1, 2018

## 1 Internet?

- Collection of billions of connected devices.
- Connected via **communication links** such as fiber, copper, radio and satellites.
- Controlled by **packet switches** such as routers and switches.
- Standardized by **protocols** such as TCP, IP, HTTP, Skype, 802.11
- Standards are made by organizations such as **RFC: Request for comments** and **IETF: Internet Engineering Task Force**

Actually a network of networks (ISPs connected together)
--

## 2 Protocol?

**Protocols** define a guide for messages (packets) sent and received between network entities by defining the:

- **format** of messages
- **order** of messages
- **actions** taken when messages are transmitted or received

## 3 Network Edge/Core

**Network Edges** are host devices i.e. client machines or servers.

**Network Cores** are interconnected routers.

<b>Frequency division multiplexing:</b> different channels transmitted in different frequency bands
---

## 4 Application Layer

The Application Layer provides the interface between the end-user and network communication.

Implementation aspects of network protocols

- transport-layer service models
- client-server paradigm

## 5 Network Applications

Network applications run on **different end systems** (network edges) and **communicate over the network**.

Network applications **do not** run on network cores.

Network applications allow for **rapid app development and propagation**.

## 6 Network Architectures

- Client-server
- Peer-to-peer (P2P)

**Client-server Architecture** is the classical architecture consisting of communication between **multiple clients** and a **singular server**.

The server is **always-on** with a **fixed address** that **can be scaled** to multiple devices. Clients communicate with directly with the server and **do not need to be always on or have a fixed address**. Clients **do not communicate with each other**.

**Peer-to-peer Architecture** is a form of network communication where clients (now peers) do not connect to an always-on server and instead **communicate directly with each other**.

Peers request service from other peer and provide service in return to other peers. Think torrents.

Peers are **intermittently connected and can change addresses**.

## 7 Processes

A **Process** is a program running within a host.

Inter-process communication is two processes communicating on the same host.

Messages are exchanged by processes communicating on different hosts.

**Client process:** initiates communication

**Server process:** waits for communication from clients

**P2P Applications have both client and server processes**

## 8 Sockets

Processes send and receive messages to and from sockets.

**Sockets** are connections between host devices.

## 9 Addressing Processes

Processes require **identifiers** so that messages can be sent back to the correct process.

Each **host** has a **32-bit IP address**.

A host can have **multiple processes** so IP addresses are combined with **port numbers** as **identifiers**.

## 10 App-Layer Protocol

App-Layer Protocol defines:

- **type of message** e.g. request, response

- **message syntax:** message fields and encoding
- **message semantics:** meaning of the fields
- **rules:** how processes should send/receive messages

Open protocols:

- defined in **RFCs**
- allows for **interoperability**

Proprietary protocols:

- normally implemented for a specific proprietary application

## 11 Transport Service Considerations

**Data Integrity** Reliability of data to reach the destination. Some applications require all data to reach the destination.

**Timing** Speed transportation takes. Some applications require fast transportation to work well.

**Throughput** Amount of data in a transfer. Some applications require large throughput while others require minimal throughput.

## 12 TCP & UDP

### TCP

- **reliable** transport protocol
- **flow control** prevent overwhelming receiver
- **congestion control** prevent overwhelming network
- **no** timing, minimum throughput guarantee, security
- **setup required** connections need to be established

### UDP

- **unreliable** transport protocol
- **no** flow control, congestion control, timing, throughput guarantee, security, or connection setup

## 13 Secure TCP

TCP & UDP connections have **no encryption**.

**SSL** connections are encrypted TCP connections.

SSL connections increase **data integrity** and offer **end-point authentication**.

SSL is an application layer protocol. Applications use SSL libraries.