**Top-down approach**: Going from Application Layer down to Physical Wires etc.

* **Application Layer:** Applications used to build the services.
  + i.e. Email client to enable email exchange between two people.
* **Transport Layer**, **Network Layer, Physical Layer** (mostly self-study)

To learn the fundamentals of computer networks:

1. What **hard problems** must they solve?
2. What **design strategies** have been proven valuable?
3. How do we evaluate **network performance**?

+ What are **TCP/IP**, **DNS**, **HTTP, NAT, VPN’s, 802.11** how does browsing the Web work?

What is the internet?

* **Hosts or End Systems:** The millions of connected devices that run **network apps**
* **Communication Links:** fiber / copper/ radio/ satellite etc.
* **Bandwidth:** thetransmission rate / bits per second
* **Packet Switches: Routers / Switches** that forward packets / chunks of data between devices
* **Internet**: Interconnected ISP’s (Internet Service Providers = org that provides access to int services)
* **Protocols**: Control sending and receiving (transmission) of msgs. (TCP, IP, HTTP, Skype, 802.11)
* **Internet Standards**: RFC (Request for comments), IETF (Internet Engineering Task Force)

A service view of the internet

* The internet is an infrastructure that provides services to applications
  + Web / VoiP, email, games, ecommerce, social nets etc.
* Provides programming interfaces to apps
  + Hooks that allow sending / receiving app programs to connect to the internet
  + Provides service options, analogous to a postal service.

What is a protocol?

* They are basically a set of rules.
* **Protocols** **define the format, order of msgs sent and received among network entities and actions taken on msg transmission and receipt.**
* All communication activity on the internet is governed by protocols
* **TCP (Transmission Control Protocol)** is one of the most popular protocols

The network structure

* **Network Edge**: The end devices which connect to the network
  + Hosts = clients and servers. Servers are often in data centres.
* **Access Network** **/ Physical Media**: What allows you to to connect to the internet.
  + Wired, wireless communication links etc.
* **Network Core**: Interconnected routers / network of networks

Access networks

* How do we connect the End Systems to the Edge Router?
  + Residential access nets, institutional access networks (school, company), mobile access networks
* Things to keep in mind about access networks:
  + Bandwidth (bits/sec) of access network? Shared or dedicated bandwidth?
  + DSL = dedicated bandwidth. Cable = shared bandwidth.
  + Congestion if too many people are using shared network.
* Access network example #1: Home network
  + Wireless Access Point (54 Mbps), Router, Firewall, Wireless Devices, cable or DSL modem, wired ethernet etc.
* Access network example #2: Enterprise network
  + Ethernet switch, institutional mail / web servers, institutional router, institutional link to ISP.
  + 10mbps, 100mbps, 1gbps, 10gbps transmission rates.
* Wireless Access Networks
  + Shared wireless access network connects end systems to the router
  + **Wireless LANs** = within building, 802.11 (WiFi), 11 / 54 / 450 mbps transmission rate
  + **Wide-area wireless access** = provided by telco (cellular operation, 10’s kms), 1 to 10mbps, 3G / 4G: LTE

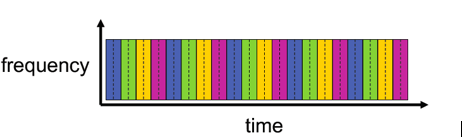
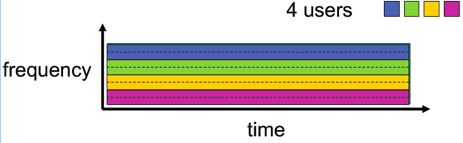
Physical Media

* **Bits**: Propagates between transmitter / receiver
* **Physical Links:** what lies between transmitter / receiver
* **Guided Media:** Copper / Fiber / **Unguided Media:** signals propagate freely i.e. radio
* **Twisted Pair:** two insulated copper wires.
* **Fiber Optic Cable**: glass fiber carrying light pulses, each pulse = 1 bit
  + Low error rate / high-speed operation 10’s-100’s gbps
* **Coaxial Cable:** two copper conductors / broadband
* **Radio**: Signal carried in electromagnetic spectrum (no physical wire)
  + Propagation environment effects: Reflection / Obstruction by objects / Interference
  + Radio link types: **Microwave (45mbps), LAN (WiFi) (1/45/450mbps), Wide-Area (~10mbps), Satellite (kbps to 45mbps)**

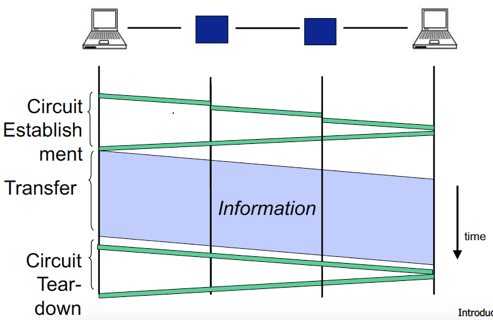
The Network Core is the mesh of interconnected routers / switches. Two forms of switched networks:

**1. Circuit Switching**: Used in legacy systems / traditional telephone networks

* End-to-end resources allocated / reserved for “call” between source and destination. Dedicated resources: no sharing
* If a connection is established with customer, even if customer is idle, no one else can use it (no sharing)
* Two technologies for Circuit Switching:
  + 1. **FDM (Frequency Division Multiplexing)**: Users share divided frequency / use frequency simultaneously
    2. **TDM (Time Division Multiplexing)**: Frequency is given to one user who gets to use the whole frequency / round robin to share use



* Timing in Circuit Switching: **Circuit Establishment 🡪 Transfer of Information 🡪 Circuit Tear-Down**



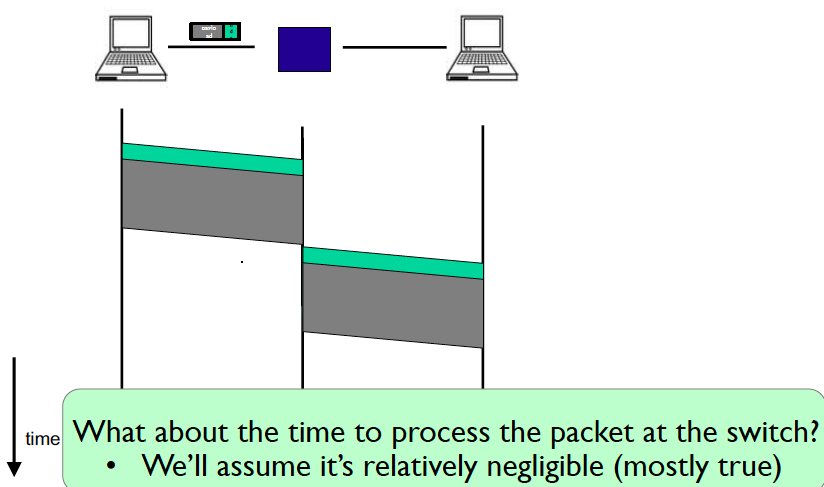
* Pros and cons of circuit switching?
  + PRO: Uninterrupted connection, reserved and dedicated for you.
  + PRO: Potentially faster depending on how much reserved / generally super performance
  + CON: Reserved channel even for idle connections, can’t be used by anyone. I.e. Waste of resources.

**2. Packet Switching**: Used in the internet

* Data is chopped into small chunks of formatted **bits** (These chunks are **Packets**)
* Packets consist of a **Header** and **Payload**
* **Payload**: The data you want to send which is split up into different packets / chunks
* **Header**: Header holds the instructions to the network for how to handle the packet
  + I.e. Who does it go to and how is it routed, what is the quality that the network should provide for this payload

1. **Internet Address**
2. **Age (TTL: Time To Live)**: To avoid looping, every time it goes through a router it decrements ctr. Packet will be dropped when counter goes to zero.
3. **Checksum to protect header**: For error correction. If bits are flipped, you know how to fix it or you know something is wrong with the data.

* **Switches** help “forward” packets based on their headers.
* Timing in Packet Switching: **Sending header + packet** 🡪 **Arrives @ Router** 🡪 **Finds next dest** 🡪 **Sends H+P to next**

****

When a packet comes, we need to:

* Find out next address
* Modify the TTL field (decrement for each router)
* Other error checking etc.

The processing time is pretty much negligible.

Diagram to the left is a **Time Series Diagram.**

Can the switch start transmitting as soon as it has processed the header? **Yes, it is called Cut-Through switching**. **Fast Switches.**

* We will always assume that we are not using Cut-Through switching, i.e. a switch processes / forwards a packet after it has received it entirely. This is called **Store and Forward** switching.
* Packet switching leverages **Statistical Multiplexing**: No link resources are reserved in advance, unlike Circuit Switching.
  + Dynamically allocates bandwidth to each channel on an as-need basis.
  + A communication channel is divided into an arbitrary number of variable bitrate digital channels/data streams.
  + Each stream is divided into packets that normally are delivered asynchronously in **first-come first-serve basis**. Alternatively, the packets may be delivered according to a **scheduling discipline** or **fair queuing**.
  + **It usually implies** **“on-demand” service rather than one that pre-allocates resources for each data stream.**

In data networks, communication flow is typically **Bursty**

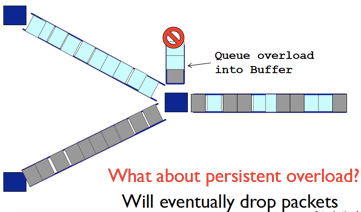
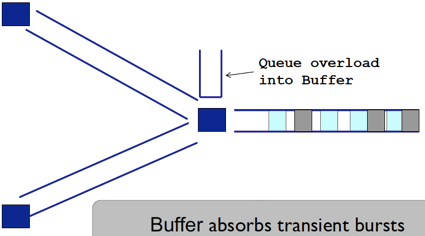
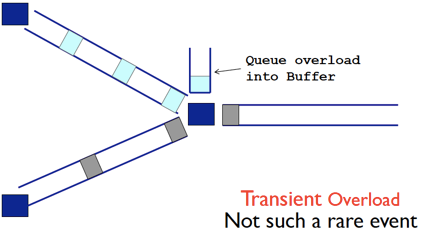
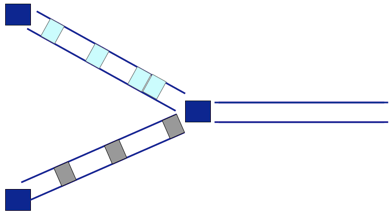
* There is a period of time where you are transmitting a lot of data and then a period where you are sending nothing.
* E.g. Voice communications: Periods of speaking vs. pause in speaking. It is not a constant stream of data.
* Video too (if compression is present), where periods of no movement on the video = less bits being sent through. However, uncompressed video will be constant.
* Without Statistical Multiplexing, everytime you’re exceeding bandwidth, you’ll have overload.
* All these multimedia and application characteristics impact network design.

What happens when communication flow shares the total capacity.

* If flow is shared, over a total period of time, you can allow other active sources to share the bandwidth where some nodes are not that active = **NO OVERLOADING**
* **Statistical multiplexing relies on the assumption that not all flows burst at the same time**.
* Overflow can still occur, but not frequently if there are bursts which all occur at the same time.

Statistical Multiplexing: A pipeline view.

* The packet size varies for different technologies.
* A router switch is simply another computer that forwards packets.



**Packets get forwarded 🡪 Back-to-back packets (Transient Overload), placed in a queue 🡪 Persistent Overload = packet loss**

* A buffer is only finite. It will fill eventually and you’ll have to drop packets.
* Why can’t we have a very large buffer? You could, but the buffer would just keep getting filled up and there would nevertheless be a huge amount of delay. E.g. Talking to someone on Skype and having huge lag / silence in between.

Pros and Cons of packet switching

* PRO: Requires less infrastructure. With circuit switching, you have dedicated linking so will require more infrastructure.
* CON: Packet loss.

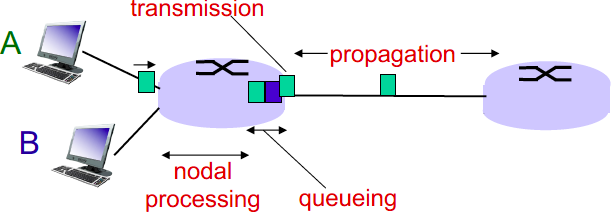
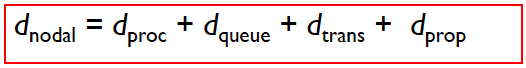
Packet Switching VS. Circuit Switching

* Packet switching allows more users to use the network
  + E.g. 1mbps link 🡪 Each user has 100kb/s when active, are active 10% of the time.
* Circuit switching: 10 users for example.
* Packet switching: 35 users, probability > 10 active at the same time is less than 0.0004%
* Is packet switching a “Slam-dunk winner”?
  + **Great for Bursty Data because of statistical multiplexing**: Resources sharing + simpler / no call setup
  + **Excessive congestion is possible:** Packet delay and loss
    - Protocols are needed for reliable data transfer and congestion control.
  + How to provide circuit-like behaviour?
    - Bandwidth guarantees need for audio/video apps.
    - Still an unsolved problem for internet-wide domain, but possible for within enterprise-level domains.

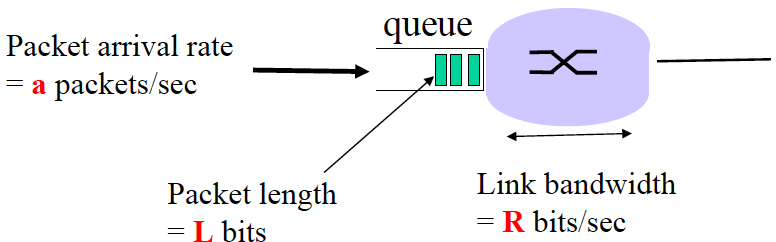
Internet structure: The internet is a Network of networks

* End systems connect to internet via. **Access ISP’s**: Residential, company and university ISP’s
* Access ISP’s in turn must be interconnected: Any two hosts can send packets to each other.
* Resulting network of networks is very complex.
* Given millions of access ISP’s, how do we connect them together? (E.g. Optus, Telstra, TPG etc.)
  + **OPTION #1: Interconnect all ISP’s.**
    - Can be bad, because connecting each access ISP doesn’t scale = O(N2) connections
  + **OPTION #2: Connect each ISP to a global transit ISP. Customer / Provider ISP’s have an economic agreement**
    - Can be bad, because single point of failure. Attacking the global transit ISP = whole internet breaks
  + **OPTION #3: Multiple global ISP’s as different businesses (each are competitors, e.g. Optus / Telstra / TPG)**
    - ISP’s can have a **Peering Arrangement** with each other to interconnect.
    - **Internet Exchange Points (ISX)** is a building full of high-speed switches / routers which help connect the global ISP’s.
    - **Regional Networks** may arise to connect access nets to ISP’s. Smaller companies using bigger ISP’s.
    - **Content Provider Networks** (E.g. Google, Microsoft etc.) may run their own network to bring services, content close to end users. Faster access to services for customers, as their customers don’t need to go through multiple hops through the network to reach their services.

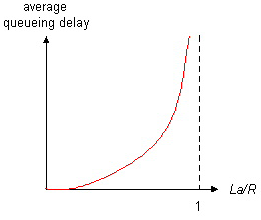
Four sources of packet delay



* **Nodal processing dPROC**
  + Checking bit errors, determine the output.
  + Typically this process is done very quickly ( < millisecond), can mostly be ignored in calculations
* **Queuing delay dQUEUE**
  + If there is an overload, the packet will be put into a buffer = time sitting in the buffer
  + Time waiting at the output link for transmission
  + Queuing delay depends on the congestion level of the router.
* **Transmission delay dTRANS**
  + The amount of time taken to transmit the **whole packet** **of size L**.
  + **L:** packet length (bits)
  + **R:** link bandwidth (bits/s)
  + **dTRANS = L / R**
* **Propagation delay dPROP**
  + The amount of time taken to transmit **1bit** through the physical link E.g. Fiber optic from SYD 🡪 New York
  + **d:** length of physical link
  + **s:** propagation speed in medium (~2 x 108 m/sec = speed of light / fiber optic cables)
  + **dPROP = d / s**

More on Queuing Delay

* Every second: **aL** bits arrive to queue
* Every second: **R** bits leave the router
* If aL > R 🡪 Queue will fill up, packets drop
  + **Traffic Intensity = aL / R**
* 1 packet arrives every L/R seconds
  + **Arrival Rate (a)** = **1 / (L/R) = R/L** (packets per sec)
  + If arrival rate is R/L and Bandwidth is R, traffic intensity = (R/L) \* (L/R) = 1
* What is the Average queuing delay (queue is empty at time 0)?
  + **{0 + L/R + 2L/R + . . . + (N-1)L/R} / N = L / (R\*N) {1 + 2 + . . . + (N-1)} = L (N – 1) / (2R)**
  + 1st packet has no delay, 2nd packet has to wait for 1 packet, 3rd has to wait for 2 and so on.
  + General rule of thumb: **Traffic Intensity = 1 is BAD** / **at 0.8 you should be fixing or expanding your network**
* **La/R ~ 0**: average traffic intensity. Queuing delay is small.
* **La/R approaches -> 1**: delays become large
* **La/R > 1**: more “work” than can be serviced, average delay is infinite (when **a (arrival rate) is random**) = collapse of sys



“Real” Internet delays and routes

* What do real internet delay and loss look like?
* **Traceroute** program provides delay measurement from source to router along end-to-end Internet path towards dest.

**For all *i* routers:**

**Send 3 packets towards destination router 🡪 router returns packets 🡪 interval between transmission / reply is timed**