**Week 1 – Introduction to Networks**

**Introduction**

**Hosts** / **End Systems**: The millions of connected devices that run **network applications**

**Communication Links**: Fiber, copper, radio, satellite etc. connected hosts / end systems.

**Packet Switches:** Routers / switches that that forward packets of data between devices.

**Transmission Rate**: Rate of data being transferred in **bits / second**.

**Internet Service Providers (ISP’s**): An organisation that provides everyone access to the internet.

**Protocols**: A standard method / set of rules that controls send/receive of information around the internet, e.g. **TCP / IP**.

**Internet Standards**: A specification of a technology method, applicable to the internet.

* **RFC’s (Request for comments)** is a publication by the **Internet Engineering Task Force (IETF)** that can become a standard / part of a standard. They are authored by engineers / computer scientists to describe methods, behaviours, research or innovations application to the internet and its connected systems.

**The Network Edge, Access Network and Network Core**

**The Network Edge**: The end devices / applications which connect to the network.

* Hosts such as clients and servers. PC’s, Macs, Smartphones, Web Servers, Email Servers etc.

**Access Network / Physical Media**: The network that physically connects a host to the first router on a path to another host.

* Home access: DSL, Cable, Fiber To The Home, Dial-up, Satellite
* Enterprise + Home access: Ethernet, WiFi (802.11) (Local Area Network)
* Wide-Area Wireless access: 5G, LTE

**Physical Media examples:** Guided media (Copper wire, Fiber Optics) + Unguided media (Satellite Radio)

**Network Core**: The mesh of interconnected routers / switches that links up the internet’s end systems.

**Two forms of network switches: Circuit Switching and Packet Switching**

|  |  |
| --- | --- |
| **Circuit Switching** | **Packet Switching** |
| Used in legacy systems, e.g. traditional telephone networks.   * Dedicated channel, guaranteeing full bandwidth and connection for the duration of the session. * If a connection is established, even if the client is idle, no one else can use the service (no sharing)   Two technologies:  **Frequency Division Multiplexing (FDM)**   * Users share divided frequency / use freq simultaneously     **Time Division Multiplexing (TDM)**   * Frequency given to one user who gets to use whole freq / round robin share use.     **Timing in Circuit Switching**   * Circuit establishment 🡪 Transfer Info 🡪 Tear-Down | Used in today’s internet.   * Data broken into **packets** (formatted **bits**) * Packets consist of a **header** + **payload** * **Payload** = data you want sent which is split into packets * **Header** = IP, Age (TTL), Checksum (error correction/integrity) * **Switches** help forward packets based on their header   **Timing in Packet Switching**   * Send header + packet 🡪 arrives @ router 🡪 finds next dest 🡪 sends header + packet to next     **^Time Series Diagram**  When packets arrive at a router, we need to:   * Find out next address * Modify TTL field (decrement-- for each router) * Other error checking * Processing time is negligible   Can the switch start transmitting as soon as it has processed the header?   * **Yes = Cut-Through Switching**   **Cut-Through Switching:** Forwarding the packet as soon as the dest address is processed.   * Reduces latency through switch and relies on dest for error handling.   **Store and Forward**: Store packet at each intermediate switch and verify integrity before forwarding to dest.   * Preferable when error-rates are high (i.e. wilderness) |

**Statistical Multiplexing**

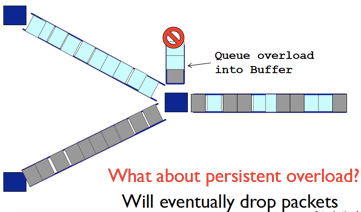
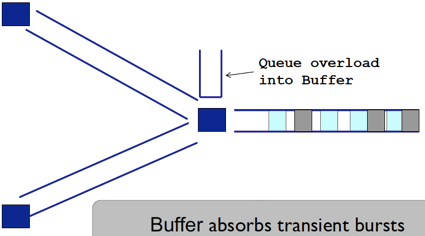
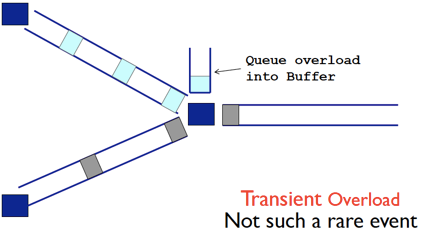
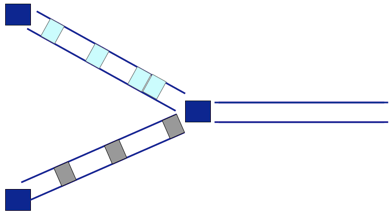
**Used in Packet Switching:** No communication link resources are reserved in advance and instead shared.

* Dynamically allocates bandwidth to each channel on an **as-need basis**

Each communication channel divided into streams 🡪 Each data stream divided into packets

* Packets are delivered in a **first-come first-serve** basis.
* Alternatively, packets may be delivered according to a scheduling discipline such as **Fair Queuing**(FIFO queue for all packets, achieving fairness)

**A pipeline view of Statistical Multiplexing**



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

*(We could have an infinite buffer, but no point because it would get filled up and there will be an infinite delay instead of p.loss)*

A lost packet may be re-transmitted by a previous node, the source node or not at all.

**Packet Switching vs. Circuit Switching**

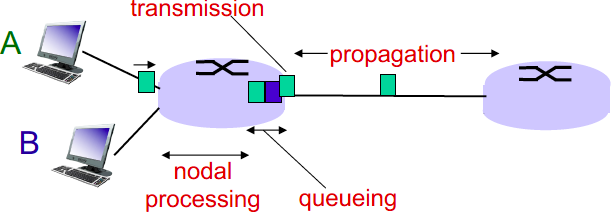
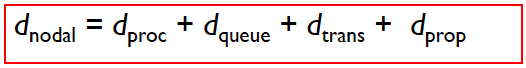
Packet Switching allows for more users to use the network simultaneously.

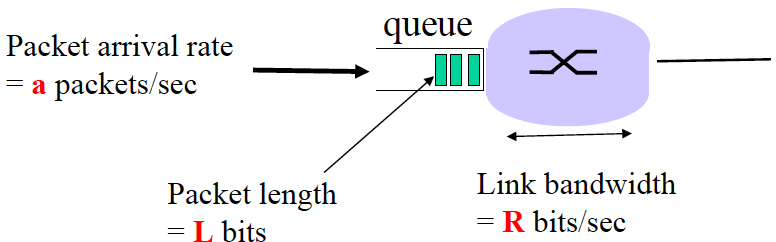
Packet Switching is **great for Bursty Data** **because of Statistical Multiplexing.**

**Excessive Congestion** can occur, resulting to Persistent Overload / packet loss.

**Four sources of packet delay**

|  |  |  |
| --- | --- | --- |
| **Nodal processing dPROC** | **< 1 ms** | Checking bit errors, determine output. Done very quickly. |
| **Queuing delay dQUEUE** | **Time in buffer** | If there is an overload, the packet will be put into a buffer.  Time waiting at the output link for transmission. |
| **Transmission delay dTRANS** | **dTRANS = L / R** | **L:** packet length (bits) / **R:** link bandwidth (bits/s) |
| **Propagation delay dPROP** | **dPROP = d / s** | **d:** length of physical link / **s:** propagation speed in medium (~2 x 108 m/sec = speed of light / fiber) |



**Queueing Delay**

Every sec: **La bits** arrive at queue

Every sec: **R bits** leave the queue

**Traffic Intensity = La / R** (La > R = overload, packets will drop)

**Arrival Rate (a)** = **1 / (L/R) = R/L** (packets per second)

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.

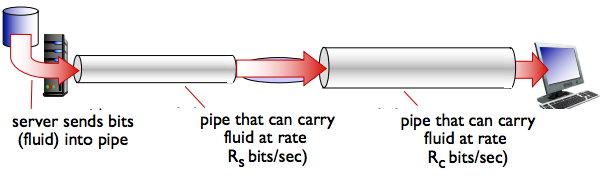
**La/R approaches -> 1**: delays become large

**La/R > 1**: more “work” than can be serviced, average delay is infinite (when **a (arrival rate) is random**) = collapse of system.

**Throughput**

**Throughput**: The rate (bits / time unit) at which bits are transferred between sender / receiver.

* **Instantaneous:** Rate at given point in time (not useful, not accurate) | **Average:** Rate over long period of time (useful)



If RS < RC , what is the average throughput?

* RS is the average throughput

**The link on the end-end path that is the slowest will be the bottleneck. You can’t get faster than bottleneck.**

**Week 2 – Application Layer: Principles & Web**

Each layer:

* Depends on the layer above
* Supports the layer above
* Independent of the other layers

Benefits of layering

* Common Abstraction for various network technologies
* API-like use

Cons of layering

* Layer N may duplicate functionality (error recovery)
* Headers can get really big
* Layer violations when network doesn’t trust endpoints (e.g. firewalls)

**The Internet Protocol Stack / Layer**

|  |  |
| --- | --- |
| **Layer** | **Tasks** |
| Application | Use an application / use data (Skype etc.) |
| Transport | Ensure the reliable arrival of data + error checking |
| Network | Deliver packets across the network |
| Data Link | Data transfer between neighbouring network nodes |
| Physical | Packets on the wire |
| Physical | Bits on the wire |

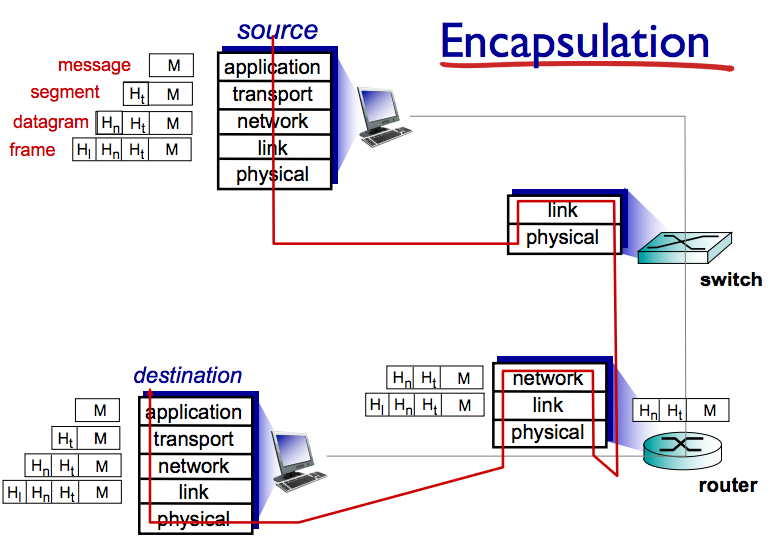
If layering didn’t exist, each new application would have to be re-implemented for every network technology = **inefficient**

We need to implement layers across machines: Hosts / Routers / Switches

* **Host implements ALL layers**: Bits arrive on wire and must make it up to the application
* **Router implements Physical Layer** 🡪 **Datalink Layer 🡪 Network Layer**

Bits arrive on wire 🡪 Packets must be delivered to next-hop 🡪 Routers participate in global delivery

* **Router does NOT implement Transport Layer** (doesn’t support reliable delivery) **and Application Layer**

**Logical Communication**: Layers interact with it’s peer’s corresponding layers.

**Physical Communication**:

* Data goes down to the physical network
* Layers get processed / transmitted to another link
* Goes upwards to other host’s application layer.

**M** = bits / file / message that you want to send

1. Transport layear appends metadata **HT**

2. Network layer appends metadata **HN = datagram**

3. Link layer appends metadata **HL = frame**

4. Finally sent to physical layer.

5. Switch runs Link + Physical Layer.

6. Router runs Network + Link + Physical

**The metadata is only treated at their relevant layer.**

When it reaches dest, as it moves up the stack, each header will be stripped off one-at-a-time and passed up the layers.