Packet Switching vs Circuit switching

The fundamental question in networking;

How is data transferred through the net?

circuit switching:

- End to end transfer
- all resources used
- Dedicated allocation
- reserved resources

packet switching:

- chunks sent over the line at a time
- resources used in chunks
- If the number of packets arriving exceeds the number that can be serviced, they are dropped
- on-demand allocation

Great for: bursty data $-->$ resource sharing, simpler, no call setup
Bad for: Applications with hard resource requirements
Excessive congestion: packet delay and loss
Need protocols for reliable data transfer, and congestion control

How can we provide circuit-like behavior?
Bandwidth guarantees needed for audio/video apps
Common solution: over-povision

Packet Switching: Statistical Multiplexing

We can use statistics to pick the best option for scheduling of packets. This will lower the amount of time that is wasted while elevating the speed things are transmitted

Packet Switching allows more users to use the network by exploiting the randomness of the users.

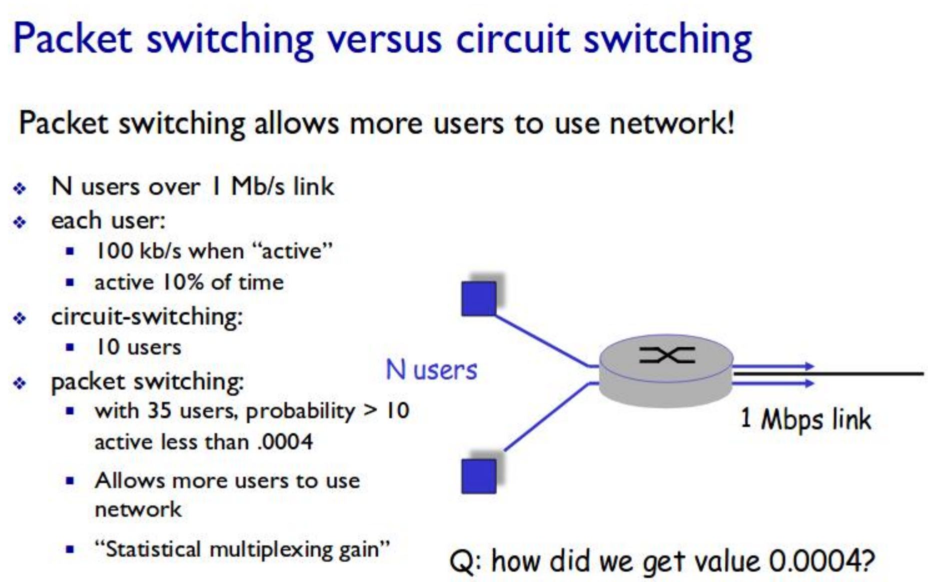

Figure 1: all the math

1 Mbps / Each user needs 100 kbps = 10^6 bps / $100x10^3$ bps

Probability that a user is active: 10%

4 Types of delay

processing $--->$ determine output link, check bit errors

queueing $-->$ biggest portion of delay, it is the slowest
$-->$ time waiting at output link for transmission

transmission
R$-->$ link bandwidth in bits per second
L$-->$ packet length in bits
$D_t = L/R$

propagation
d = length of physical link
s = propagation speed in medium
$D_{prop} = d/s$

EX; Propagation Delay:
Packet of length 1024 bytes
$--> 1024 * 8$ bits
$-->$ 8192 bits

Over 27 kbps dial up modem access limit
$--> \frac{8192}{27*10^3}$ sec
$-->$ 0.303 sec
Over 10 Gbps Ethernet
$--> \frac{8192\ bits}{10*10^9\ bits/sec}$
$-->$ 8192 nano sec
$--> 8192 * 10^{-7}$

Nodal Delay $= D_N = D_p + D_q + D_t + D_{prop}$

EX; Queueing Delay:
traffic intensity $= \frac{La}{R}$