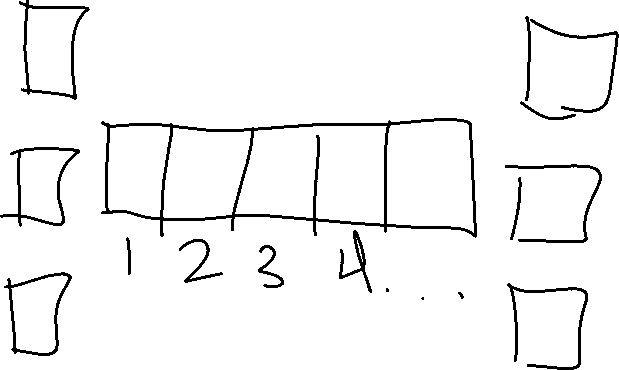
Notes 1-18

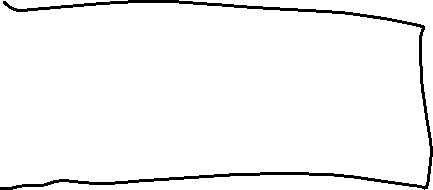
Multiplexing

* Any node needs to be able to speak to any other node at any other point
* We also want the ability for 2 senders to be on the network at the same time



STDN (Synchronous Time Division Multiplexing)

Better Way: Computers will have its own frequency



* This is using bandwidth
* FDM (Frequency Division Multiplexing)
  + This is how TV cable works
* Both of these ways require you to know how many devices are on the net work
* To fix this, we use a dynamic approach.
  + We set an upper bound that is known in advance that is then broken into fixed-size messages
  + Each sender can send a sequence of packages and sets an upper bound on how many you can send until the next sender can work

1 goal: connectivity

2 goal: cost effectivity

3 goal: Usefulness? Commonality?

* Put everything you need in your program OR the network provides those communication services to the program
* We want to hide the complexity, but not hide enough to constrain any services or application designers
  + Find a ‘sweet-spot’
* Think of pipes or channels.
  + What is it we want the channel to do for us?
    - Home message is always delivered?
    - Delivered in the same order?
    - Message is private?
    - Message is delivered without errors?
    - Could be any of these
* We can make network smart and end nodes simple and dumb or end nodes smart and network simple and dumb
* People who designed old telephone network made the end nodes dumb and the network smart
* We want our network to have some sense of reliability:
* 3 kinds of errors:
  + Bit errors: not common, can correct
  + Packet errors: a complete packet (some message we want to send, we break it up into portions and send those portions) is lost by the network. It might have been dropped due to errors. At each network hop, there is a chance that the forwarding device might ‘drop the packet on the floor’. It could also be the forwarding software has a bug / defect and the packet could have been altered. It could have also been corrupted or ‘mangled’
    - Sometimes correct
  + Link Level / Node Errors: can’t really correct
* b = bit. B = byte
* kilo, mega, and giga are not powers of 2, so…
  + k = 2^10, m = 2^20, g = 2^30 when talking about computers
  + Other way is 10^3, 10^6, and 10^9
  + Use #1 (or the 2^n) when we mean a quantity of data
  + We use #2 (or the 10^n) when we mean a distance or a rate (transmission speed)
* Bandwidth:
  + Can be 1 of 2 things:
    - A data rate
    - A frequency range in cycles/sec (Hertz).