CS 455: INTRODUCTION TO DISTRIBUTED SYSTEMS

[NETWORKING]

Lugging a torrent of bits
From here to there
And through thin air

With fidelity ... for an error
begets a retransmission and then another

What's done to a bit, is done to the next
Be it a blockchain or a simple text

Shrideep Pallickara
Computer Science
Colorado State University

Topics covered in this lecture

- Encoding
- Bandwidth and Latency
- Multiplexing
- Network Architecture
- Encapsulation
Communications & Networking:
Topics that we will cover

- Data transmission
- Switched Networks
- Bandwidth and Latency
- Multiplexing
- Internet Architecture
- IP routing
- The TCP and UDP protocols
For our purposes, let’s ignore details of modulation

- Assume we are working with two signals
  - High and low

- In practice:
  - Different voltages on a copper-based link
  - Different power-levels on an optical link

Let’s do the obvious thing

- Map 1 to a high signal
- Map 0 to a low signal
Non-return to zero (NRZ)

Problems with NRZ because of consecutive 1’s and 0’s: **BASELINE WANDER**

- Receiver keeps *average* of the signal seen so far
- Average is used to *distinguish* between low and high
- Lots of consecutive 1/0’s will make it difficult to detect a significant change
Problems with NRZ because of consecutive 1’s and 0’s: Clock Recovery

- Every clock cycle, sender transmits and the receiver receives
- Sender and receiver’s clocks must be perfectly synchronized
  - Otherwise, it is not possible to decode the signal

Manchester encoding

- 0 is a low-to-high transition
- 1 is a high-to-low transition
Manchester encoding and NRZ

NRZ

Manchester Encoding

Some more about Manchester encoding

- Doubles the rate at which signal transitions are made on the link
  - Receiver has $\frac{1}{2}$ the time to detect each pulse

- Rate of signal changes: baud rate

- Bit rate is $\frac{1}{2}$ the baud rate
  - Encoding is considered 50% efficient
NRZI (Non return to zero inverted)

- Make a transition from current signal to encode a 1
  - Stay at current signal to encode a 0
- Solves the problem of consecutive 1’s
  - But does nothing for consecutive 0’s

4B/5B encoding

- Attempts to address inefficiencies in Manchester encoding
  - Without suffering from problems due to extended high/low signals
- The crux here is to insert extra bits into bitstream
  - Breakup long sequences of 1s or 0s
  - 4 bits of actual data encoded in a 5-bit code
  - 5-bit codes are carefully selected
    - No more than 1 leading 0 & no more than 2 trailing 0s
### 4B/5B: Rules for the conversion of 4-bit codes to 5-bit codes

- **Objective** is to ensure that in each translation there is:
  - No more than one leading \(0\)
  - No more than two trailing \(0\)’s
  - When sent back-to-back
    - No pair of 5-bit codes results in more than 3 consecutive \(0\)’s being transmitted

- **5-bit codes are transmitted using NRZI**
  - This is why they are so concerned with consecutive \(0\)’s

<table>
<thead>
<tr>
<th>4B</th>
<th>5B</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>11110</td>
</tr>
<tr>
<td>0001</td>
<td>01001</td>
</tr>
<tr>
<td>0010</td>
<td>10100</td>
</tr>
<tr>
<td>0011</td>
<td>10101</td>
</tr>
<tr>
<td>0100</td>
<td>01010</td>
</tr>
<tr>
<td>0110</td>
<td>01110</td>
</tr>
<tr>
<td>0111</td>
<td>01111</td>
</tr>
<tr>
<td>1000</td>
<td>10010</td>
</tr>
<tr>
<td>1001</td>
<td>10011</td>
</tr>
<tr>
<td>1010</td>
<td>10110</td>
</tr>
<tr>
<td>1011</td>
<td>10111</td>
</tr>
<tr>
<td>1100</td>
<td>11010</td>
</tr>
<tr>
<td>1101</td>
<td>11011</td>
</tr>
<tr>
<td>1110</td>
<td>11100</td>
</tr>
<tr>
<td>1111</td>
<td>11101</td>
</tr>
</tbody>
</table>
NETWORKING: EXPECTATIONS AND LINKS

Expectations that we have of a network

- Application **programmer**
  - Error-free and timely delivery of messages

- Network **designer**
  - Cost effective design
  - Effective and fair allocation of resources

- Network **provider**
  - Easy to administer and manage
  - Isolate faults and account for usage
A network must provide connectivity among a set of computers

Physical medium: Link

Nodes

Multiple-access

Multiple access links are limited in size

- Geographical distances that can be covered
- Number of nodes that can be connected
Connectivity between nodes need not imply a direct physical connection. Otherwise …

- Networks would be very **limited** in the number of nodes they could connect

- Number of wires out the back of a node
  - Unmanageable
  - Very expensive

Switched networks: Indirect connectivity among cooperating nodes
Switched networks: Indirect connectivity among cooperating nodes

- Nodes with at least two links
  - Run software that forwards data on one link out on another

- Types
  - Circuit switched
  - Packet switched

Switched networks: Circuit switched networks

- Establish a dedicated circuit
  - Across a set of links
  - No one else can use this till termination

- Allows source to send a stream of bits
  - Across circuit to the destination node

- Employed by the telephone system
  - Also known as POTS (Plain Old Telephone System)
Switched networks: Packet switched networks

- Nodes in the network send discrete data blocks to each other
- Use store-and-forward
  1. Receive complete packet over some link
  2. Store packet in internal memory
  3. Forward complete packet to another node
- Used by the overwhelming majority of computer networks

Interconnection of networks

Router/Gateway forwards messages between networks
Addressing: A node must be able to say which nodes it wishes to communicate with

- Assign an address (byte string) to each node
  - Distinguish node from other nodes in the network
- Source specifies address of the destination node
- Switches and routers use address to forward messages \textit{towards} the destination node
  - \textbf{Routing}

\textbf{COST EFFECTIVE RESOURCE SHARING}
How do all hosts that want to communicate share the network …

- At the same time?
- How about **sharing** links?
  - Hosts want to use it at the same time

- **Multiplexing** …
  - **ANALOGY**: Sharing CPU among multiple processes

Data sent by multiple users can be multiplexed over the physical links

Switch 1   

Switch 2
Multiplexing data onto a physical link

- Synchronous time division multiplexing (STDM)
  - Divide time into quanta
  - Assign quanta in round-robin fashion

- Frequency division multiplexing (FDM)
  - Transit data *flows* at different frequencies

Problems with STDM and FDM

- {Problem-1} **Limited** to specific situations
  - Max number of flows is *fixed*
  - Known *ahead* of time

- {Problem-2} If one of the flows does not have data?
  - Its share of the physical link remains *idle*

- In computer communications:
  1. Amount of time a link is idle can be very large
  2. Data flows are fluid
Statistical multiplexing

- Physical link is shared over time
- Data is transmitted from each flow **on demand**
  - Not a predetermined slot
  - When there is only one flow?
    - *No need to wait* for quantum to come around

Limiting transmissions so that other flows can have a turn

- Upper bound on **size** of data block that each flow is allowed to transmit
  - Packet
- Larger application messages
  - Fragmented into several packets
  - Receiver reassembles these
- Each flow sends packets over the link
  - Decision made on a **packet-by-packet basis**
Multiplexing packets from multiple sources onto a shared link

Switch 1

Deciding which packet to send over a shared link

- In some cases, decision is made by switches
- Service packets using
  - FIFO
  - Round robin
    - Ensure flows receive a certain share of the bandwidth
    - Maximum threshold for delays for certain packets
- Networks that allow special treatment of flows
  - Quality of Service
Network performance is measured in two fundamental ways

- **Bandwidth**
  - Number of bits transmitted over the network in a given time (e.g., 10 million bits per second, 10 Mbps)
  - Also called *throughput*

- **Latency**
  - How long it takes for a message to go from one end of the network to another?
Components of latency

- Speed-of-light propagation delay
  - $3 \times 10^8$ m/sec in vacuum
  - $2.3 \times 10^8$ m/sec in cable
- Amount of time to transmit a unit of data
- Queuing delays

The Delay x Bandwidth product

Viewing the Network as a pipe
The Delay x Bandwidth product

- The product gives us information about how many bits fit in the pipe

- Transcontinental channel
  - 50 ms one-way latency
  - Bandwidth: 45 Mbps
  - Can hold: 50 \times 10^{-3} \text{ seconds} \times 45 \times 10^6 \text{ bits/second}
    - \(2.25 \times 10^6 \text{ bits} = 280 \text{ KB}\)

The Delay x Bandwidth product

- Corresponds to how many bits the sender must send
  - Before first bit arrives at the receiver

- Bits in the pipe are said to be in flight
Bandwidth and latency improvements are not in lockstep

- Over past 35-40 years approximately
  - Bandwidth improvements: 220-1200 times
  - Latency improvements: 4-20 times

- Ethernet 802.3 (1978)
  - 10 Mbps
  - Latency 3 millisecond

- Ethernet 802.3ae (2003)
  - 10,000 Mbps (1000 times)
  - Latency 0.19 millisecond (15 times)

What does not change as the bandwidth increases?

- Speed of light

- High-speed does not mean that latency improves at the same rate as bandwidth
  - Transcontinental delays of 100 ms for 1-Mbps/1-Gbps link
Sending 1 MB data over a cross country link. Delay 100 ms

- 1 Mbps link
  - Pipe: $100 \times 10^{-3} \times 10^6 = 100 \text{ Kb} = 0.1 \text{ Mb}$
  - So you need 80 pipes to transmit 1 MB
    - $8 \text{ Mb}/0.1 \text{ Mb} = 80$

- 1 Gbps
  - Pipe: $100 \times 10^{-3} \times 10^9 = 100 \text{ Mb}$
  - So you need $8 \text{ Mb}/100 \text{ Mb} = \text{approx } 1/12\text{th}$ of the pipe is utilized

-- Support for common services

 COLORADO STATE UNIVERSITY  
 Computer Science Department  
 Networking  
 L2.45
More accurate to think of network as allowing applications to communicate

- When 2 applications need to communicate
  - Lot of things need to happen
  - Beyond just sending messages between the hosts
- Build all functionality into each app?
- Identify and build right set of **common services**
  - Hide complexity without constraining functionality

Processes communicating over an abstract channel
Guarantees provisioned in the channel

- Guaranteed delivery?
- Ordered delivery?
- Thwart eavesdropping?

Not just which functionality, but where they will be provided

- View network as a **bit pipe**
  - High-level communication semantics provided by end hosts
  - Keeps switches in the middle *very simple*

- Alternative: Push functionality **onto** switches
  - End hosts are dumb devices
    - Telephones
Mask failures so that the network appears more reliable than it really is

- Bit errors
- Burst errors: Consecutive bits are corrupted
- Packet failures
  - Discarded because the switch buffer is full
    - Congested
  - Routing mistakes
- Node and link failures
  - Route around failed nodes and links
All communications in distributed systems based on sending/receiving messages

- **No shared memory**

- Sending message from A to B
  - Build message in A's address space
  - Send message over the network
  - Reconstruct message at B

But A and B must agree on the meaning of the bits

- Signaling 1's and 0's
- What is the last bit of the message?
- Detect if the message is lost or damaged
  - Respond to problems
- Representation of data types
Layering and Protocols

- Start with services provided by hardware
- Add a sequence of layers
  - Each providing higher level of service
- Services at higher layers implemented in terms of lower layers

Advantages of layering

- Decomposes problem into manageable components
- Provides modular design
  - Adding functionality may result only in minor modifications
Advantages of layering

- Each layer can be changed **independently** of the other
- Change as technology improves

Layered Architectures: Requests go down the hierarchy; results flow upward

Component at layer $L_i$ can call components at layer $L_{i-1}$
Example of a layered network system

- Application programs
- Process-to-process channels
- Host-to-Host connectivity
- Hardware

Layered system with alternative abstractions at a given layer

- Application programs
  - Request/Reply channels
  - Message stream channel
- Host-to-Host connectivity
- Hardware
Abstract objects that comprise layers of a network system are called **protocols**

- Provides a **service interface** to other objects on the same computer
  - Wishing to use its communication services
- Defines the **form** and **meaning of messages** exchanged by protocol peers
- Protocol also refers to modules that implement a specification

The contents of this slide-set are based on the following references