Internet RFC

- Request for comment (RFC) (started in 1969)
  - In informal way to publish new ideas/protocols
  - First publish Internet Drafts before publishing as RFC
  - RFC can be obsoleted by newer RFCs, not modified
  - Some of them become standards
- Check out www.rfc-editor.org
  - RFC 0133 **File Transfer and Recovery** R.L. Sundberg [Apr 1971]
  - RFC 1889 **RTP: A Transport Protocol for Real-Time Applications** Audio-Video Transport Working Group, H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson [January 1996]
  - RFC 1945 **Hypertext Transfer Protocol -- HTTP/1.0** T. Berners-Lee, R. Fielding, H. Frystyk [May 1996]

Outline

- End-to-end argument
- Status of home work project #1?
- Chapter 2: Direct Link Networks
  - Encoding
  - Framing
  - Error Detection
  - Sliding Window Algorithm

End-to-end arguments

- End-to-End Arguments in System Design – J. H. Saltzer, D. P. Reed and D. D. Clark – MIT (1980)
  - Jerome Saltzer – Multics (a precursor to UNIX)
  - David Reed – Visicalc, Lotus 1-2-3, TCP/IP
  - David Clark – Multics, Internet
- Recent followup article:

http://web.mit.edu/Saltzer/www/publications/endtoend/ANe2ecomment.html

- KISS principle (Keep It Simple, Stupid)

Modular vs End-to-End

- Modular:
  - Design, well designed modules, each of which performs a task completely
  - E.g. networking 7 layer OSI model
- End-to-end:
  - Some tasks a better left to higher layers
  - Keep the lower layers simple, perform error checking and other operations at lower layers if it is simple. Higher layers will duplicate this functionality anyway
  - E.g. Internet (only supports IP datagrams)

How does it matter?

- Internet:
  - When ARPANET (->DARPANET->INTERNET) was being designed, there was a tension between adding intelligence in the network. Earlier networks provided “reliable service”.
- Ultimately they chose IP data gram as the transport mechanism. IP provides a best effort service.
- That has enabled the technology to be useful for such a long time. People built TCP (reliable), UDP (unreliable), RTSP (streaming), IP-SEC (secure IP) , VOIP (voice over IP) etc on top of IP.
- Same argument for RISC/CISC (architecture)

Example 1: Careful file transfer

- At host A, the file system reads the file and gives to file transfer program
- File transfer program asks network to send file in packets
- Network moves packets from A to B
- Network takes packets from network and gives it to file receive program
- File receive program puts data inthe file system

Modular Approach

- Make sure that each of these steps mask failures
  - Read from disk, perform check sums to verify data integrity
  - Make sure the communication network delivers data reliably etc
- End-to-end approach
  - Transfer the file, perform a checksum of the complete file, if it matches then data was correct
  - Optimization: Perform checksums after parts of the file are received

Example 2: Faulty router

- With a modular design, proper check sums verified before putting packets in input buffer. Packets got corrupted inside the router before it copies data to the output buffer. Went undetected

Performance aspects

- Modular implementations do help similar applications
- Frees end user from complexities
- The tradeoff is that the lower levels should provide some functionality, which is easy to implement and leave complicated schemes to higher level mechanisms

Example 3: Delivery guarantees

- Lower level delivery guarantee mechanism can tell when a message was delivered to a computer.
- However, what users really care is if a message was understood by the recipient
  - E.g. Return-Receipt-To: email header will trigger mail servers to send a receipt when email is delivered.
  - Return-View-To: will prompt the email program to send a response as soon as someone opens the email.
  - What you really want is when users read the email and understand it. After message is delivered to the system, the disk can crash, access can be turned off so that lower level acknowledgement is useless. If you user understands the message, they will respond back
- You can force the end system to guarantee delivery to application but simpler at the application level

Example 4: Secure transmission of data

- Data transmission system performs encryption and decryption
  - Trust lower levels with key management
  - Data will be in clear text in the operating system
  - Authenticity of message must still be checked
- Leave end-to-end authorization to higher layers.
- Lower level can perform basic encryption
  - E.g. data between wireless access point and the wireless card is encrypted. Can be broken but it stops casual hackers and is not too hard to implement

Example 5: Duplicate message suppression

- Applications sometime know if a message is a duplicate and drop the second update

Threats to end-to-end paradigm

- Quality of Service (QoS) – Routers treat packets differently based on the source, $$$ etc. Internet2 provides heavy support for QoS.
- Firewalls, Caches, NAT
  - Prevents end-to-end interaction

Failures of E2E model

- Multicast
  - Can perform at application layers, but the lower levels can implement it easier

Performance Metrics

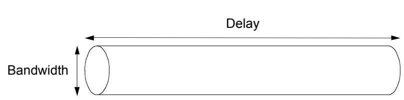
- Bandwidth (throughput)
  - data transmitted per time unit
  - link versus end-to-end
  - Notation - Mbps = 10^6 bits per second
- Latency (delay)
  - time to send message from point A to point B
  - one-way versus round-trip time (RTT)
  - components
    - Latency = Propagation + Transmit + Queue
    - Propagation = Distance / c; c - speed of light
    - Transmit = Size / Bandwidth

Bandwidth versus Latency

- For interactive performance, is it better to login through a 1 Gbps satellite link or 56 kbps dialup link?
- Relative importance
  - 1-byte: 1ms vs 100ms dominates 1Mbps vs 100Mbps
  - 25MB: 1Mbps vs 100Mbps dominates 1ms vs 100ms
- Infinite bandwidth
  - RTT dominates
    - Throughput = TransferSize / TransferTime
    - TransferTime = RTT + 1/Bandwidth x TransferSize
  - 1-MB file to 1-Gbps link as 1-KB packet to 1-Mbps link

Delay x Bandwidth Product

- Amount of data “in flight” or “in the pipe”
- Example: 100ms x 45Mbps = 560KB



Direct Link Networks

Direct Link Networks

- Hosts are directly connected by some medium
  - Twisted pair: telephone cable, Ethernet (Category 5: Cat5)
  - Coaxial pair: TV
  - Optical Fiber
  - Wireless: Infrared, Radio, Microwave
- Common bandwidth designators:
  - DS1 (or T1): 1.544 Mbps
  - DS3 (or T3): 44.736 Mbps (for example, Charter Athens has 2 DS3 links now)
  - STS-1 (OC1): 51.840 Mbps
  - STS-12: 622.080 Mbps …

Last Mile

- Plain Old Telephone Service (POTS):
  - 28.8 Kbps to 56 Kbps
- ISDN
- xDSL 1.544 Mbps to 8.448 Mbps
- Cable (40 Mbps down, 20 Mbps up) – Shared
  - wish we can get that much huh?

Relevant technologies

- Encoding - (how are we talking - language)
- Framing - (how do know the end)
- Error
- Reliable transmission (over unreliable links)

Encoding

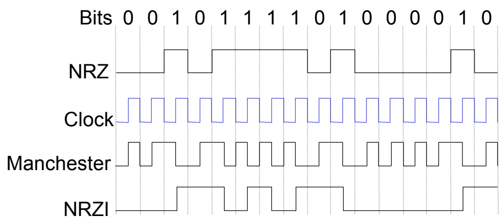
Encoding

- Signals propagate over a physical medium
  - modulate electromagnetic waves
  - e.g., vary voltage
- Encode binary data onto signals
  - e.g., 0 as low signal and 1 as high signal
  - known as Non-Return to zero (NRZ)
- Problem: Consecutive 1s or 0s
  - Low signal (0) may be interpreted as no signal
  - High signal (1) leads to baseline wander
  - Unable to recover clock

Alternative Encodings

- Non-return to Zero Inverted (NRZI)
  - make a transition from current signal to encode a one; stay at current signal to encode a zero
  - solves the problem of consecutive ones
- Manchester
  - transmit XOR of the NRZ encoded data and the clock
  - only 50% efficient
- 4B/5B
  - every 4 bits of data encoded in a 5-bit code
  - 5-bit codes selected to have no more than one leading 0 and no more than two trailing 0s (thus, never get more than three consecutive 0s)
  - resulting 5-bit codes are transmitted using NRZI
  - achieves 80% efficiency

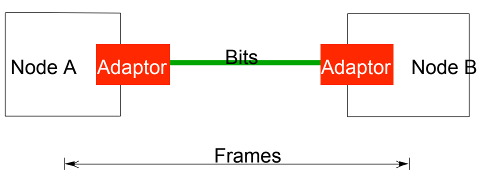
Encodings (cont)



Framing

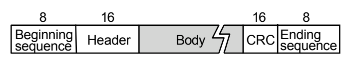
Framing

- Break sequence of bits into a frame
- Typically implemented by network adaptor



Approaches

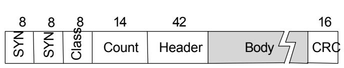
- Sentinel-based
  - delineate frame with special pattern: 01111110
  - e.g., HDLC (ISO), SDLC (IBM), PPP (dialup)



  - problem: what if the special pattern appears in the payload itself?
  - solution: bit stuffing
    - sender: insert 0 after five consecutive 1s
    - receiver: delete 0 that follows five consecutive 1s

Approaches (cont)

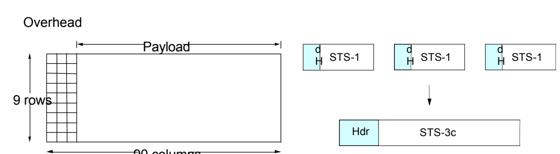
- Counter-based
  - include payload length in header
  - e.g., DDCMP (DECNET)



  - problem: count field itself corrupted
  - solution: catch when CRC fails

Approaches (cont)

- Clock-based
  - each frame is 125us long
  - e.g., SONET: Synchronous Optical Network
  - STS-n (STS-1 = 51.84 Mbps)



Three STS-1 frames multiplexed onto one STS-3c