Announcements

Outline

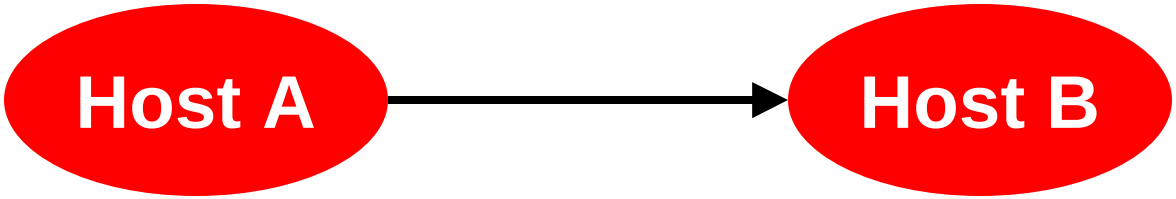
- End-to-End Arguments in System Design – J. H. Saltzer, D. P. Reed and D. D. Clark – MIT (1980)
  - Jerome Saltzer – Multics (a precursor to UNIX)
  - David Reed – Visicalc, Lotus 1-2-3, TCP/IP
  - David Clark – Multics, Internet
- KISS principle (Keep It Simple, Stupid)

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)

Example 1: Careful file transfer



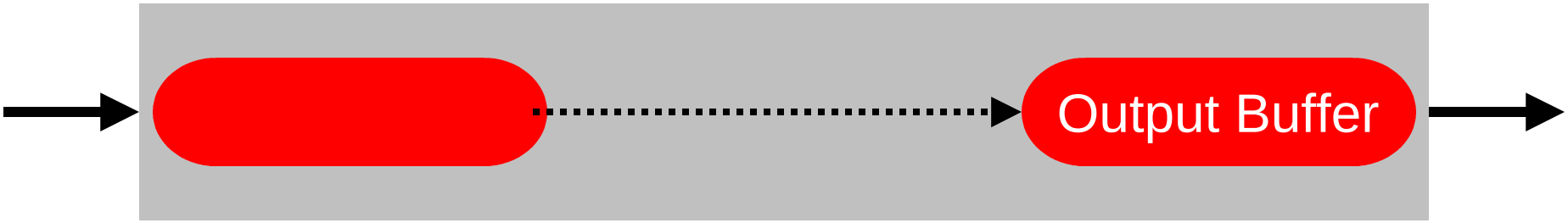
- At host A, the file system reads the file and gives to file transfer program
- File transfer program asks network to send file in packets
- Network moves packets from A to B
- Network takes packets from network and gives it to file receive program
- File receive program puts data inthe file system

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.
- What users really care is if a message was delivered to the user
  - E.g. Return-Receipt-To: email header will trigger mail servers to send a receipt when email is delivered. What you really want is when users read the email. After message is delivered to the system, the disk can crash, access can be turned off so that lower level acknowledgement is useless
- You can force the end system to guarantee delivery to application but simpler at the application level
- Two phase commit

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

How does it matter?

- Internet:
  - When ARPANET (->DARPANET->INTERNET) was being designed, there was a tension between adding intelligence in the network. Ultimately they chose IP data gram as the transport mechanism. 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) etc on top of IP.

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.

- Voice-over-IP
  - Megaco – Simple phones, complexity in the server
  - SIP – Complexity in the phones, simple network
    - What about wire-tapping?

- 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

Discussion