



# End to End Arguments in System Design

J.H. Saltzer, D.P. Reed and D.D. Clark  
M.I.T. Laboratory for Computer Science

Presented By:  
Ankur Aggarwal

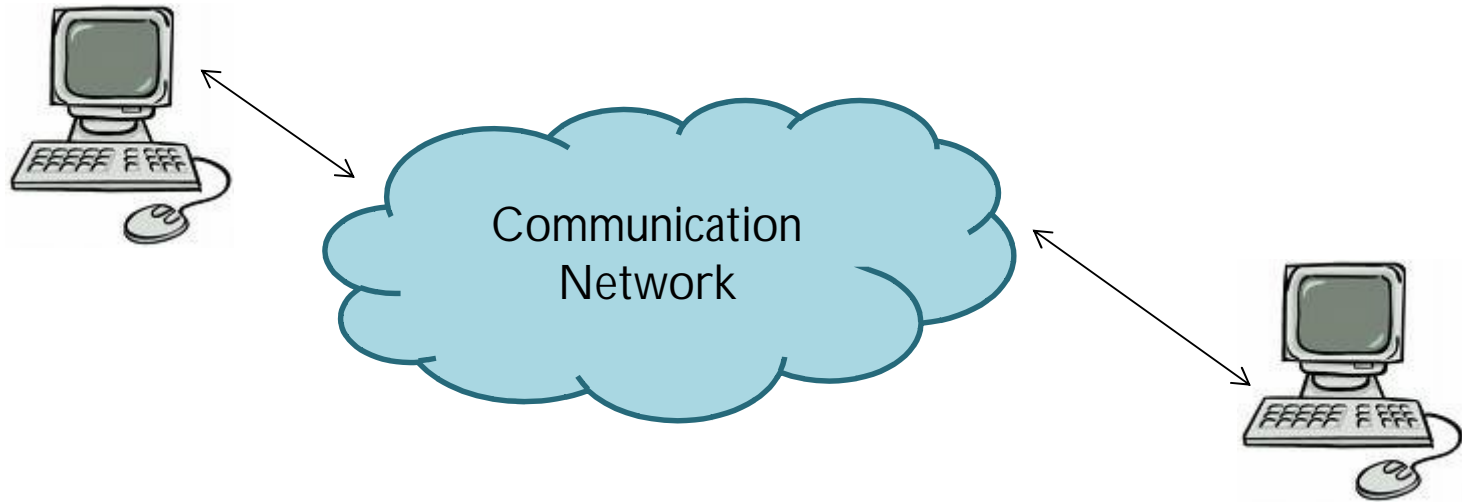
09/11/08



# Highlights

- Choosing proper boundaries between functions
- Rationale for moving function upwards

# How and Where?



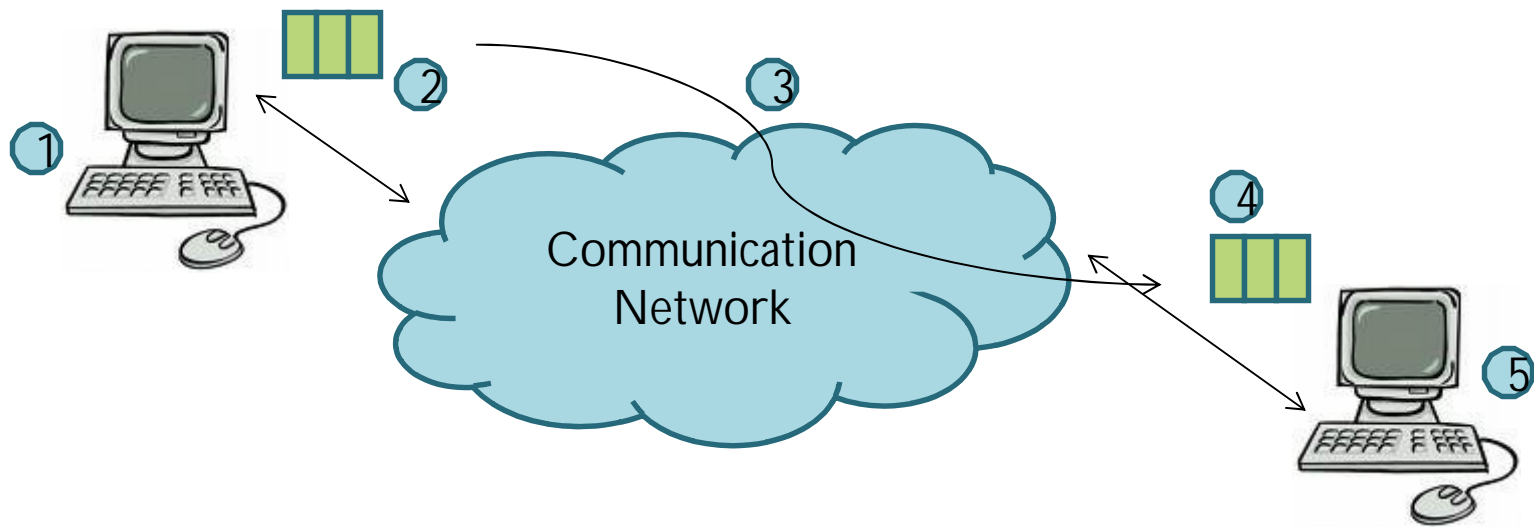
- Communication Network
- Client
- Joint venture
- Each doing its own version?



# End to End Argument

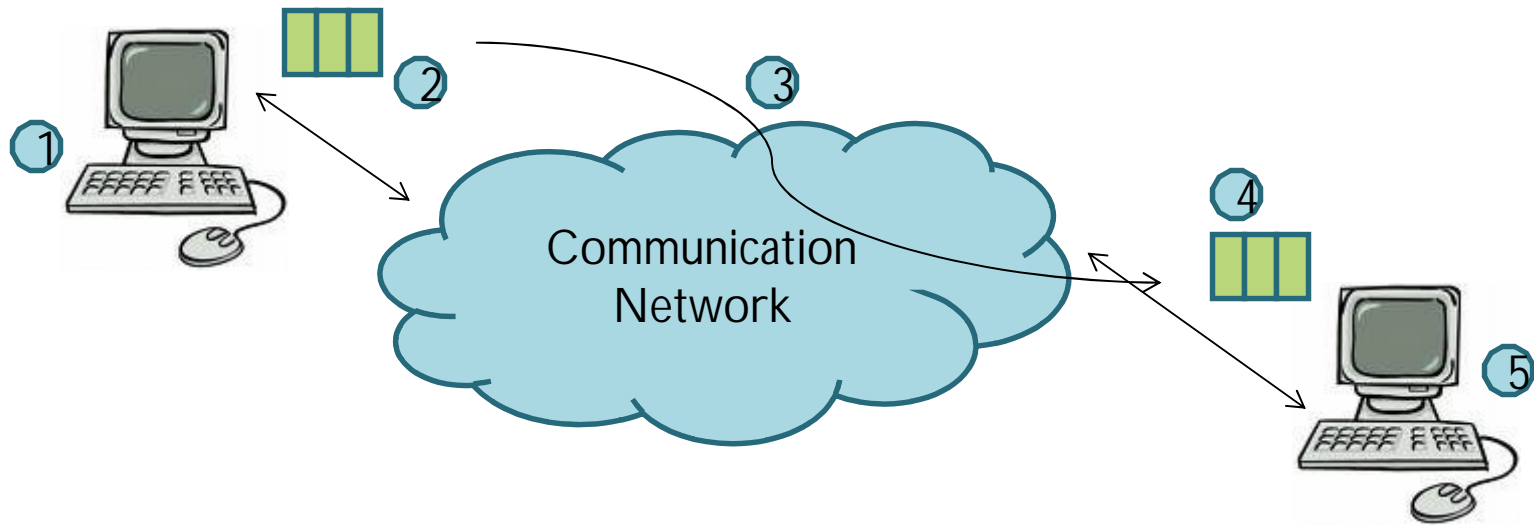
- “In a system that includes communications, one usually draws a modular boundary around the communication subsystem and defines a firm interface between it and the rest of the system.”
- “The function in question can completely and correctly be implemented only with the knowledge and help of the application standing at the end points of the communication system.”
  - Providing it as a feature of communication system is not possible.
  - Sometimes may help\*

# Example – File Transfer



- 1 - Read from disk
- 2 - Write data to packets
- 3 - Move packets from A to B
- 4 - Read data from packets
- 5 - Write to disk

# Threats



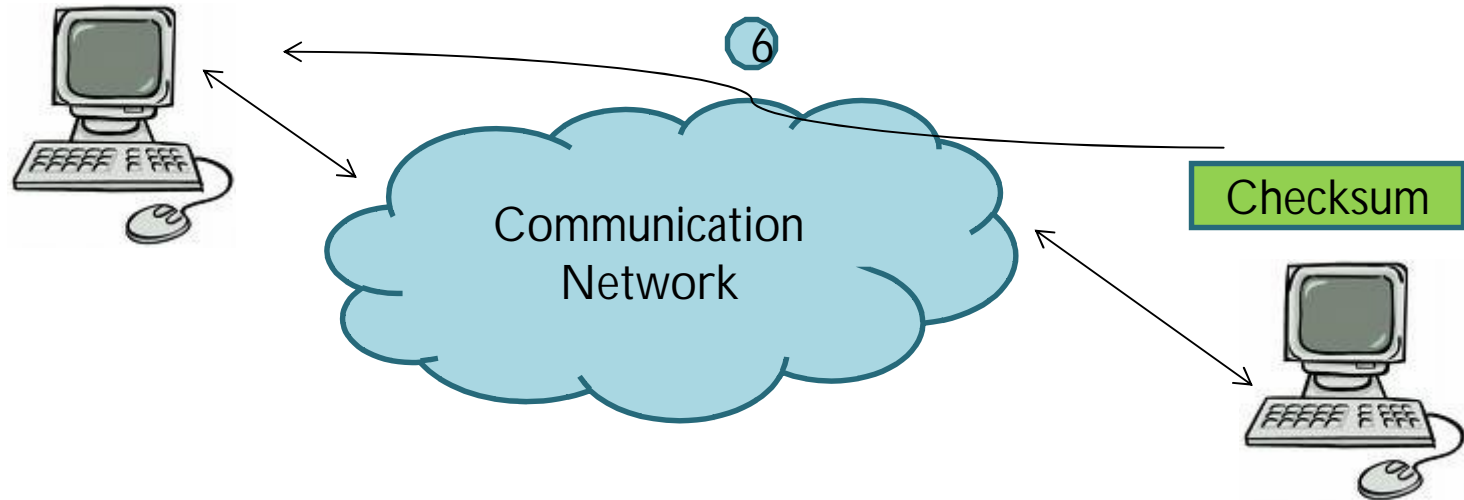
- Incorrect Read
  - Hardware Faults
- Mistake in buffering / copying
- Dropping a packet, Change of bits
- Hosts may crash



# Approach I

- Reinforce each step using duplicates, timeouts, retry
  - Reducing probability of threats to an acceptable small value
- Uneconomical
- \*Requires writing correct programs

# Approach II



- If checksum equal, file transfer complete; else resend
- Works fine if number of failures are small



# Comm. System can help

- What if we provide error checks within communication network?
  - Not all threats are addressed
  - Application still has to perform end to end guarantee
  - Might be useful but should not go out of the way



# Performance

- What if length of the file increases?
  - Transmission time increases
  - Probability of correctness decreases
  - If communication system unreliable
    - Frequent Retries
  - If too many reliability measures
    - Performance Costs

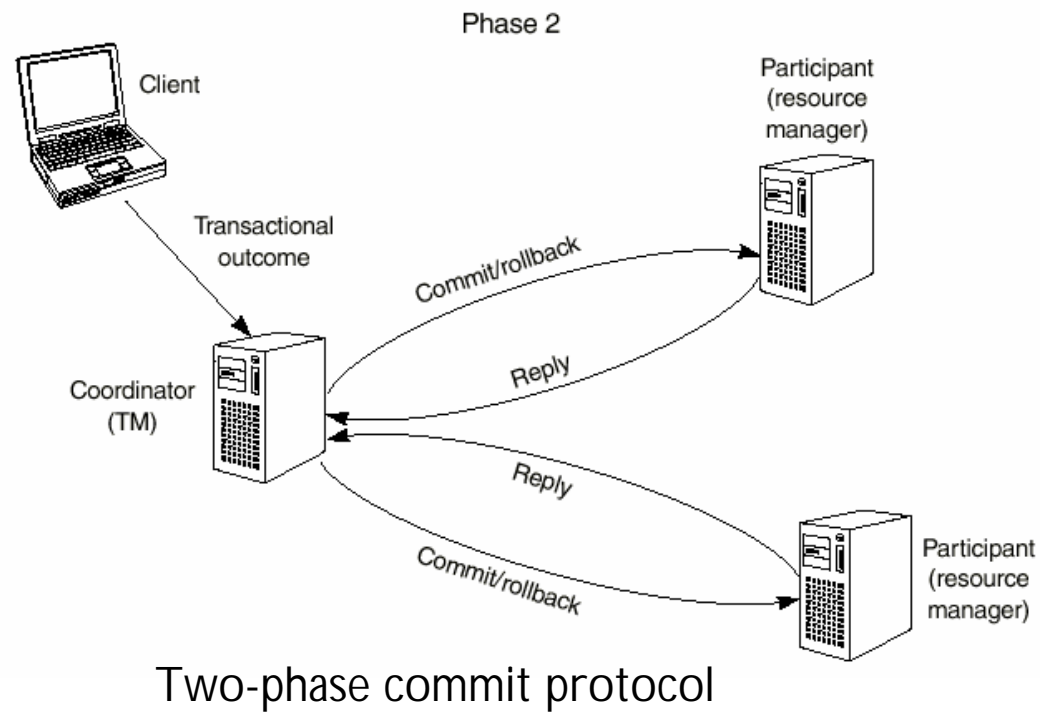


# Delivery Guarantees

- Lower level may be wasting its effort providing a function that must by nature be implemented at application level
- Acknowledgement of delivery
  - - useful as a form of congestion control
  - - does not tell whether the host acted on the message
- Make target host responsible to act just after delivery

# Delivery Guarantees

- What if action requested on target host be done only if some actions requested at other hosts are successful?





# Secure Transmission of Data

- What if communication system provides encryption and decryption?
  - - Must be trusted to manage the encryption keys
  - - Data will be in clear as it passes into the target node
  - - Authenticity still must be checked by the application
  - - Unsophisticated encryption – same key can be used by all hosts with frequent changes



# Duplicate Message Suppression

- What if communication system provides duplicate message suppression?
  - - Application may generate duplicates in its own failure retry procedure
  - - Looks like different message to the communication system
  - - Authenticity still must be checked by the application
  - - Application must know how to detect its own duplicates and suppress them.
  - - Retried request can only be known by application
    - Useful for detecting system crashes



# FIFO Message delivery

- Messages sent through different communication paths may arrive in different order than sent.
- Distributed application – cannot take advantage to ensure actions occur in the correct order.



# SWALLOW

- Distributed data storage system
  - Object ID, Version, Type of Access, Value (if write)
  - Underlying message system does not suppress messages
  - Duplicate read discarded by the originator, duplicate write identified by object id and version
  - Eliminated Ack' s from communication system
    - - Write ACK's provided by higher layers
- Simplified low-level message communication protocol with significant reduced effect on host load and network load



# Identifying the ends

- Have to take it case-by-case (application requirements)
- Two people in Real-time conversation
  - Low level reliability measures would induce delays
  - Better to slightly damaged packets
- Voice packets being stored in a file
  - Delays do not matter, accuracy is important



# Changing Tones

- A possible re-framing of E2E is “trust-to-trust”.
- Original: “The function in question can completely and correctly be implemented only with the knowledge and help of the application standing at the end points of the communication system.”
- Revised: “The function in question can completely and correctly be implemented only with the knowledge and help of the application standing at points where it can be trusted to perform its job properly.”