

End-to-End Arguments in System Design

J.H. Saltzer, D.P. Reed and D.D. Clark

Presented by Ankit Singla
CS 598 Sep. 3, 2009

The Question

In modular systems, given multiple possible points of function implementation, where do we place our functions?

The Answer

Implementing functions at lower levels might be redundant or of little value.

The End-to-End Argument

*"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." **

* Performance is a factor

Thank You!

Questions?

Careful File Transfer

File transfer from host A to host B

- File system at A: read from disk, return blocks to application
- Communication system at A: break into packets, transmit
- Network: move packets from A to B
- Communication system at B: recover data from packets, return to file transfer application
- File system at B: write to disk at B

So much can go wrong ...

Performance considerations

- If 1 in 100 packets were dropped, transfer of large files will suffer under the check-and-retry method.
- Perfection in the communication system is expensive, hard to achieve, unnecessary, and even redundant.
- Treat the problem as a trade-off.
- To implement functions efficiently, making use of all the information helps.
- Efficiency for some apps is a burden for others.

Delivery Guarantees

- ARPANET delivered a RFNM after successful delivery of a message
- Is this useful to applications? Applications need to know if the message was acted upon.
- Alternative strategy: Receiver is sophisticated and guarantees action upon accepting a message
- What about two-phase commits? What about receiver's failure / refusal to execute the action?

Secure Transmission of Data

- Problem of application level authentication
- Data transmission system needs to be trusted with encryption keys
- Point of vulnerability: transfer of unencrypted data from the target host to the specific application
- Communication system level encryption has other purposes though

Duplicate Message Suppression

- Time-out based retries often cause duplicate messages
- Should the network attempt to tackle this?
- Duplicates created by the application are non-identical messages to the network!

Other Examples

- Guaranteeing FIFO Message Delivery
- Transaction Management

Identifying the end-points

- Live conversation over a communication network
 - Even the parties involved in conversation can take care of reliability to some extent.
- Voice-mail like function
 - Need for better accuracy implies some reliability is required at lower levels.
 - Packet ordering and duplicate suppression still need handling by the receiver.

Discussion

- Ease of deployment and change?
- Scalability, decentralization, transparency?
- Congestion control? Security?
- Complexity of applications?
- Other end-to-end arguments? Variants of THE end-to-end argument?

Thank You!

[This time's for real]