End-to-end

Overarching theme

It’s often best to put functionality at the application layer, instead of in a lower layer

- Why?
  - Proper functionality may be impossible without help of application
  - Application may need to duplicate the functionality regardless
  - Not all applications will need the functionality

Examples of end-to-end

“careful file transfer”

- goal: move file from disk of computer A to disk of computer B without damage

- errors could happen in any piece of the system
  - even with a perfect communication system, the application will still need to check for errors (e.g. end-to-end check and retry)
  - therefore, making the communication system perfect doesn’t help the application that much (it will just reduce the frequency of retries)

Examples of end-to-end

delivery guarantees:

- Lets the sender know that a message was delivered
    - Not very useful when done by the communication system
        - Would like to know that the application at the other end properly handled the message, not just that it received it.

Examples of end-to-end

More examples:

- Data encryption
  - Application may need to do end-to-end encryption anyway
- Duplicate message suppression
  - Duplicated messages may be caused by parts of the system other than the communication system
- FIFO message delivery
  - A distributed application may send messages from different sites, so the messages won’t be received in FIFO order
- etc.

Other applications of end-to-end

- End-to-end arguments are not just for communication systems

- other examples in the paper: RISC architectures, operating systems