Recitation 11 — E2E

What is the end-to-end (E2E) argument?

- An argument that there are functions that can be completed correctly/completely *only* by applications. If lower-layers support these functions, it should be only as performance enhancements.
- The E2E argument does *not* prevent the network from implementing features in the middle of the network

What are some reasons for this argument?

(You may disagree with these, or find other arguments more compelling!)

1. In almost every situation, the endpoints will have to perform the same sort of functionality anyway (in the file example, the endpoints have to do some sort of checksumming, because errors can happen on the local machine).
2. Other applications can suffer huge performance hits when the lower layers make decisions for them. For example, voice applications care about latency, not bit errors; correcting for bit errors will lead to unacceptably high overhead.

Example: Careful File Transfer

- The steps
  - Host A reads file from disks, passes to file transfer program
  - File transfer program asks the communication system to transmit the data
  - Communications system moves data from Host A to Host B
  - And then the reverse at Host B
- What can go wrong?
  - Hardware failure at A (bad read)
  - File system or file transfer program or communication system might make a mistake in copying/moving
  - Local memory could have failure
  - Underlying communications system may drop or change data
  - Hosts could crash
- How do we fix this?
  - E2E check and retry

Discussion: What is the value of the E2E principle?

- Discipline in thinking
- Minimizing functionality and dependence on support machines
- By minimizing what is required of the interior of the network, edges can innovate without requiring cooperation from the network