

**CSE/EE 461 – Lecture 10**  
**Virtual Circuits**  
**The End-to-End Argument**

---

Janet Davis  
[jlnd@cs.washington.edu](mailto:jlnd@cs.washington.edu)  
January 28, 2004  
Reading: Saltzer et al.

**Most important ideas from last time**

---

- Clarification of split horizon /poison reverse
- Link state routing
  - Dijkstra's algorithm example
  - LSP flooding complications
- Cost metrics
- Virtual circuits
  - Hard state vs. soft state
- "Routing is still cool"
- "I can see your notes"

L4.2

**This time...**

---

- Your questions:
  - OSPF & TTL = 0
  - Virtual circuits and hard vs. soft state
- Virtual circuits vs. datagrams
- The End-to-End Argument:  
How do we decide where in the system to place functions?

L4.3

**OSPF**

---

- OSPF =

L4.4

### TTL = 0 Clarification

---

- Problem: How to make sure other routers don't ignore my LSPs if I fail & restart?
  - Fishnet:
  
  - OSPF:

L4.5

### Virtual circuits: Hard state vs. Soft state

---

Hard state

Soft state

L4.6

### Network Service Models

---

Model	Datagram	Virtual circuit
Metaphor		
Other terms		
Packet forwarding		
Router state		

L4.7

### Datagrams or Virtual Circuits?

---

- Pros and Cons?
  - Simplicity/robustness versus stronger resource allocation
- We return to these tradeoffs later
  - Quality of Service (QOS)
  - At the heart of current Internet evolution
  - Intserv (connection oriented) vs Diffserv ("connectionless")

L4.8

### Motivation for the End-to-End Argument

---

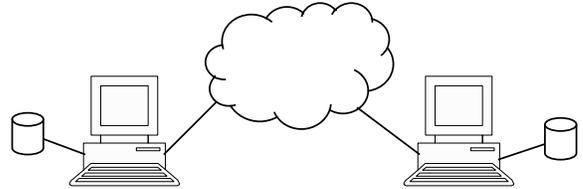
- How do we decide where in a system to place required functions?
- In particular, in the application or in the network?

L4.9

### Example: Careful File Transfer

---

- Goal: Copy file from A's disk to B's disk *reliably*

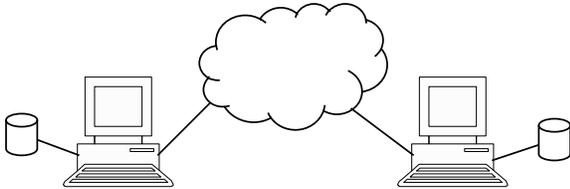


L4.10

### Threats to Careful File Transfer

---

- What could go wrong?



L4.11

### Approach 1: Point-to-Point

---

- Reduce the probability of each threat to an acceptably low level.

L4.12

## **Approach 2: End-to-End**

---

- Transfer the file, then check for errors & retry if needed.

L4.13

## **A common proposal**

---

- Suppose the communication system provides perfectly reliable data transmission.
- Does this help us?

L4.14

## **The End-to-End Argument**

---

- If the function in question can be implemented completely and correctly only with the help of the application,
- Then providing the function as a feature of the communication system is not possible.
- (The communication system can provide an incomplete version of the function if it helps performance.)

L4.15

## **The Essence of the End-to-End Argument**

---

Functions should be placed where they can be implemented correctly and completely.

L4.16

### **Example: Error detection**

---

- We need to do end-to-end error detection for correctness.
- Why also do it at the link layer?

L4.17

### **Performance Enhancements**

---

- Functionality at lower layer can enhance performance
  - Not required for correct operation
  - Can be required for reasonably efficient operation

L4.18

### **Enhancement tradeoffs**

---

- Applications have more semantic information about service needs
  - e.g.
- Network has more information about its true capabilities
  - e.g.
- This tension is the subject of countless papers...

L4.19

### **Identifying the ends**

---

Phone call

Music file transfer

L4.20

## Perspective

---

- E2E principle an influential idea since 1981
  - 293+ citations on Citeseer.org
- Helpful, but best taken with a grain of salt
- Pragmatics
  - Application designers often can't change the network!
  - Network designers often can't change the applications!

L4.21

## The last 3 weeks...

---

- Reliability: How can we build a network out of unreliable, distributed components?
- Putting data on the wire
  - Coding, framing, error detection
- Sending data reliably
  - Sliding window & connections
- Sending data over many hops
  - Bridging, routing, & virtual circuits
- Deciding where to put functions
  - The end-to-end argument

L4.22

## Next time...

---

- Heterogeneity: How do we support a range of media & applications?
- Layering and the design of the Internet
- Reading:
  - Peterson 3.1 & 4.1.1
  - Clark (focus on sections 1-7)

L4.23