CPS 514
Day 1: End to End Argument
Agenda

- Administra-tivia
  - Projects.
  - Next week

- Background: Layering, what are the ends?

- End-to-End

- Next week: Fate-Sharing & Tutorial!
Internet layering = “Protocol stack”

- Separation of concerns
  - Break problem into separate parts
  - Solve each one independently
  - Tie together through common interfaces: abstraction
  - Encapsulate data from the layer above inside data from the layer below
  - Allow independent evolution
Layers, Services, Protocols

Application
- Service: user-facing application.
- Functions: Application specific

Transport
- Service: multiplexing applications
- Functions: Connection establishment/termination, error control, flow control

Network
- Service: move packets to any other node in the network
- Functions: Routing, addressing

Link
- Service: move frames to other node across link
- Functions: Framing, media access control, error checking

Physical
- Service: move bits to other node across link
- Functions: Convert bits to signal
- Functionality needed for communication;
  - Security, reliability
  - Naming, location, QoS,
  - Congestion control, flow control

- Where should they go?
What are the ends?

• What are the ends in a Skype Application?

• What are the ends in a file transfer
How are the Ends and Middle different?

End
- Application
- Transport
- Network
- Link
- Physical

Some Core
- Application
- Transport
- Network
- Link
- Physical

Traditional Core
- Network
- Link
- Physical
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. Therefore, providing that questioned function as a feature of the communication system itself is not possible. (Sometimes an incomplete version of the function provided by the communication system may be useful as a performance enhancement.)
The Argument

- Philosophical argument about Modularity
- Keep the core simple and the edge smart
- Limits network by moving functionality out
- Ends justify the means!!!
Why put Functions in the network?

- Allows for global optimizations
- Gives network operators control
- Enables in-network enforcement
- Enables re-use of function
Why Keep Functions out of the network?

- Simple networks —> cheaper networks
- Diagnosis and debugging is easy
- Hard to predict necessarily functions for future apps
- Minimal: only those who want will implement
- Optimized functions: app can make app specific function
Exception(s) to the End to End Argument

- Re-implement functionality in N/W for performance optimization
  - Not for correctness
  - No need to focus on completeness

- Must consider overheads
  - All apps need to use this function
  - E.g. Retransmits in the wireless layer
Violations of the End to End Principle

- Network Address Translator
  - End: my browser or app
- Virtual Private Network
  - End: my browser or app
- TCP-Proxies
  - End: my browser or app
- SPAM
  - End: the user.
On-Going Debate

Routing

Traditionally done in the core … but hard for end-host to avoid countries that violate privacy
On-Going Debate

• Routing
  • Traditionally done in the core … but hard for end-host to avoid countries that violate privacy

Avoiding Internet Surveillance: The Complete Guide

Written by Dann Albright
January 25, 2015
What is a Secure Network?
Net Neutrality: End-to-End Strikes

- Add special functions to the network
- Classify traffic
- Change data based on user plans

Policies need information from the ends.
- User info, app info, data info

Send Us More $, or YOU’LL Never See Your Favorite Website via high-Speed Internet Access Again!
Net Neutrality: End-to-End Strikes
Back

- Add special functions to the network
  - Classify traffic
  - Change data based on user plans
- Policies need information from the ends.
  - User info, app info, data info
Summary

- End-to-End: keep the network simple
  - Pros v. Cons
- Discussed exceptions
- Discussed violations