End-to-End Arguments and Design Philosophy

ECE/CS598HPN

Instructor: Radhika Mittal

Acknowledgement: Presentation contents derived from Ion Stoica and Scott Shenker at UC Berkeley

End-to-End Arguments in System Design

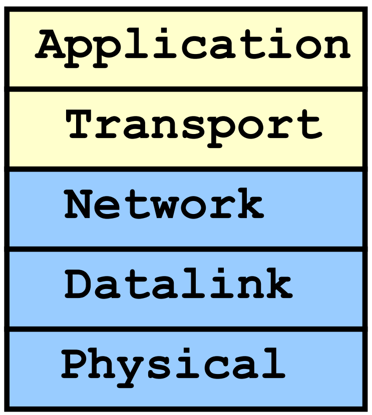
J.H. Saltzer, D.P. Reed and D.D. Clark

IEEE Trans. On Communication, 1984

Where should functionality be placed?

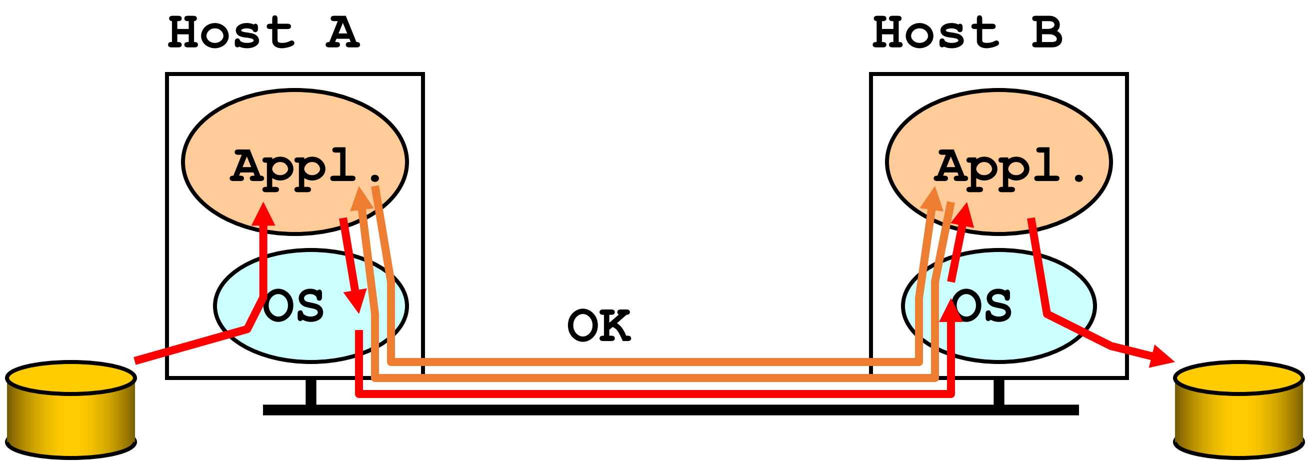
- The most influential paper about placing functionality.
- The "Sacred Text" of the Internet
  - endless disputes about what it means
  - everyone cites it as supporting their position

Where should functionality be placed?



- More about which layer is responsible for the functionality.
- Less about where the functionality is implemented (end-host or switch).
  - *Still has some implications.*

Example: Reliable File Transfer



- Solution 1: make each step reliable, and then concatenate them
- Solution 2: end-to-end check and retry

Example (cont'd)

- Solution 1 not complete
  - What happens if any element misbehaves?
  - The receiver has to do the check anyway!
- Solution 2 is complete
  - Full functionality can be entirely implemented at application layer with no need for reliability from lower layers
- Is there any need to implement reliability at lower layers?

Conservative Interpretation

- "Don't implement a function at the lower levels of the system unless it can be completely implemented at this level" (Peterson and Davie)

- Unless you can relieve the burden from hosts, then don't bother

Radical Interpretations

- Don't implement anything in the network that <u>can</u> be implemented correctly by the hosts
  - Makes network layer absolutely minimal
  - Ignores performance issues

Moderate Interpretation

- Think twice before implementing functionality in the network
- If hosts can implement functionality correctly, implement it a lower layer only as a performance enhancement
- But do so only if it does not impose burden on applications that do not require that functionality

Challenge

- Install functions in network that aid application performance….

- …without limiting the application flexibility of the network

Extended Version of E2E Argument

- Don't put application semantics in network
  - Leads to loss of flexibility
  - Cannot change old applications easily
  - Cannot introduce new applications easily
- Normal E2E argument: performance issue
  - introducing more functionality imposes more overhead
  - subtle issue, many tough calls (e.g., multicast)
- Extended version:
  - short-term performance vs long-term flexibility

Questions

- Do these belong to “network” layer?
  - Multicast?
  - Quality of Service (QoS)?
  - Web caches?

- How is end-to-end principle violated in today’s networks?

The Design Philosophy of the DARPA Internet Protocols

David D. Clark

SIGCOMM'88

Goals

0. **Connect existing networks**
   - Initially ARPANET and ARPA packet radio network
1. Survivability
   - Ensure communication service even in the presence of network and router failures
2. Support multiple types of services
3. Must accommodate a variety of networks
4. Allow distributed management
5. Must be cost effective
6. Allow host attachment with a low level of effort
7. Allow resource accountability

Connect Existing Networks

- Existing networks: ARPANET and ARPA packet radio
- Decision: packet switching
  - Existing networks already were using this technology
  - Met the needs of target applications.
- Store and forward router architecture

- Internet: a packet switched communication network consisting of different networks connected by store-and-forward gateways (routers).

Survivability

1. As long as the network is not partitioned, two endpoints should be able to communicate
2. Failures (excepting network partition) should not interfere with endpoint semantics.

- Stateless network. Maintain state only at end-points
  - Eliminates network state restoration.
  - **Fate-sharing**

Types of Services

- Use of the term "communication services" already implied that they wanted application-neutral network.
- Realized TCP wasn't needed (or wanted) by some applications.
- Separated TCP from IP, and introduced UDP.

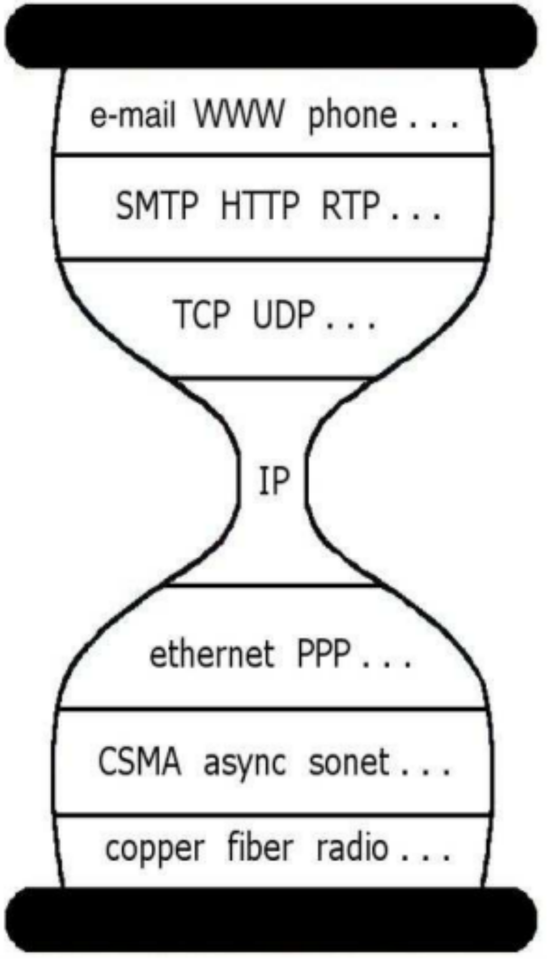
Variety of Networks

- Incredibly successful!
  - Minimal requirements on networks
  - No need for reliability, in-order, fixed size packets, etc.

- IP over everything
  - Then: ARPANET, X.25, DARPA satellite network..
  - Now: Ethernet, wifi, cellular,…

Key feature: Datagrams

- No connection state needed
- Good building block for variety of services
- Minimal network assumptions



Distributed Management of Resources

- Different gateways in the Internet operated by different administrators that do not trust one another.
    - Different AS or domains.

- Routing across different domains governed by certain policies.
    - Requires manually setting tables.

- BGP for inter-domain routing developed in 1989 (after the paper was written).

Other goals

- Cost-effectiveness:
  - 40 bytes of header.
  - Cost of retransmissions.
- Cost of attaching a host:
  - Dumb (stateless) network and smarter hosts increases host attachment effort.
- Accountability:
  - Not quite provided by the Internet.

Questions

- What priority order would a commercial design have?

0 **Connect existing networks**
1. Survivability
2. Support multiple types of services
3. Must accommodate a variety of networks
4. Allow distributed management
5. Must be cost effective
6. Allow host attachment with a low level of effort
7. Allow resource accountability

Questions

- What goals are missing from this list?

0 **Connect existing networks**

1. Survivability
2. Support multiple types of services
3. Must accommodate a variety of networks
4. Allow distributed management
5. Must be cost effective
6. Allow host attachment with a low level of effort
7. Allow resource accountability

Questions

- Which goals led to the success of the Internet?

0 **Connect existing networks**

1. Survivability
2. Support multiple types of services
3. Must accommodate a variety of networks
4. Allow distributed management
5. Must be cost effective
6. Allow host attachment with a low level of effort
7. Allow resource accountability

New Terms and Concepts

- Fate-sharing
- Flow
  - Sequence of packets from a source to a destination.
  - New building block?
- Soft-state
  - Routers maintain “non-critical” per-flow state that can be recovered upon crash or failures.
  - Desired type of service still enforced by end points.

Key Advantages

- The service can be implemented by a large variety of network technologies
- Does not require routers to maintain any fine grained state about traffic. Thus, network architecture is
  - Robust
  - Scalable