

# End-to-End Arguments in System Design

Saltzer, Reed and Clark  
Presented by Pekka Nikander

# Presentation roadmap

- Why this paper?
- What is the point?
  - Some finer details
  - Cost versus benefit
  - Why not lower layers?
- But this all is trivial, isn't it?
- And what is wrong in the paper?
- Summary

# Why this paper?

- Seminal paper, often referenced
  - citeseer index: 370 citations
- End-to-end principle is often discussed in connection with Internet protocols
  - see how it was originally argued

# What is the point?

- If a function can be “implemented only with the **knowledge** and help of the application”, any (partial) implementation of it is useful only “as a performance enhancement”. [p 2]
- A major part of the discussion concerns reliability and performance (next slides)
- Note that the argument includes the application, making the results **application specific**

# Some finer details

- Many “careful” applications require application-specific end-to-end reliability functions, at least to protect against crashes
  - Especially “automated” applications, for example, e-mail delivery
- **Partial** reliability functions are still usable within the network
  - They do not provide “perfect” reliability
  - E.g., SMTP runs on TCP (and not UDP)

# Cost versus benefit

- “amount of effort to put into reliability” is “an engineering tradeoff based on performance”
- Reliability measures have a performance cost
  - Reduced bandwidth available to apps
  - Added delay
  - E.g. 2G link layer is “too” reliable for TCP

# Why not on lower layers

- Lower layers are shared
- Reliability imposes a “tax” that is paid **also** by those applications that don’t need them
- Lower layers may not have as much information as higher layers, leading to less efficient implementation

# This is all trivial, isn't it?

- The basic argument is indeed trivial
  - But only after you “grog” it
- It is the subtleties that make it hard
  - What are the **application** requirements?
    - Are these **typical** or not?
  - What are the **end-points** in question?
- What is the most **economic** way?

# What is wrong?

- Security arguments hold only partially
  - Need to consider traffic analysis

# Summary

- Don't make lower layers too perfect
- Only one guideline
  - Must consider also other guidelines
- Functionality in the network imposes a **cost**
  - Worth implementing if many apps use it
  - Should be possible to **avoid** the cost, i.e., apps should have access to raw service