Design Philosophy of DARPA Internet Protocols

Baris Kasikci

Checkpoint

- By tonight
  - *Email me with your group information (names, topics you are interested in)*
  - *Fill in https://goo.gl/forms/3QvYHQjIvwTWczFs1*
    - I will finalize the presentation assignments after 8 PM today
- 9 People missed reviews
  - *You can miss 4 without penalty*
- Watch the conference talk example

Fundamental Goal

> *"The top level goal for the DARPA Internet Architecture was to develop an effective technique for* ***multiplexed*** *utilization of existing* ***interconnected*** *networks."*

- Multiplexing (single channel used by many communicating parties)
  - *Circuit switching versus packet switching*
- Connecting multiple heterogeneous networks
  - *Gateways are key*

Secondary Goals

1. Internet communication **must continue despite loss of networks or gateways.**
2. The Internet must support **multiple types of communications** service.
3. The Internet architecture must accommodate a **variety of networks**.
4. The Internet architecture must permit **distributed management** of its resources.
5. The Internet architecture must be **cost effective**.
6. The Internet architecture must permit **host attachment** with a low level of effort.
7. The resources used in the internet architecture must be **accountable**.

Datagrams

- Connectionless service
  - *No state established ahead of time*
- Key building block for switching
- UDP is an app-level interface to the datagram service of the Internet
- Building block for other protocols (TCP)

TCP vs. UDP

- TCP: Reliability (ssh, email)
- UDP: Loss is OK (video, voice)
- No QoS guarantees in the lower levels

Supporting Varieties of Networks

- How do you support a heterogeneous set of networks?
- How do you shield applications from the heterogeneity?

Fate Sharing

- Move state to the end point
  - *Survivability*
- OK to lose the state information of an entity if the entity is also lost
  - *TCP state versus router state*

Weaknesses, Strengths, Open Questions?

- Strengths
  - *Simple idea of datagrams*
  - *Scalability*
  - *Immense success of the Internet*
- Weaknesses
  - *Narrow IP interface hurts innovation at the IP level*
  - *Hiding secrets may hurt efficiency*
    - Can be overcome by not hiding power (e.g, DPDK)

End-to-End Arguments In System Design

Baris Kasikci

End to End Principle

- Placing functions at lower levels of a system may not be beneficial
  - *Functions can be redundant at low levels*
  - *Low-level function placement may be costly*
- A correct communication system can only be built with the help of the endpoints
  - *Examples: Detecting crashes, delivering messages, sequencing messages, etc.*

Low-Level Functionality

- The paper argues that low-level functionality is mainly a performance optimization
- When does it make sense to stick to the end-to-end argument?

Weaknesses, Strengths, Open Questions?

Discussion Point: Congestion Control

- Is congestion control amenable to an end-to-end implementation?

- Criticism of the end-to-end principle
    - *Some functions are better suited for lower-levels*