Discussion:
Is the End-to-End argument in system design still needed?

LIXIA ZHANG, UCLA

DINRG MEETING @IETF116

MARCH 30, 2023

Trigger of this talk (to trigger more discussions)

6.2. The End-to-End Principle Redux

The end-to-end principle is the idea that reliability and trustworthiness reside at the end nodes of networks rather than in the network itself. In other words, the idea was that the network itself was dumb and intelligence was at the edge or end. However, Internet architecture is evolving in such a way that this principle is changing.

.....

Rapidly, the end-to-end principle is becoming the edge-to-edge principle.

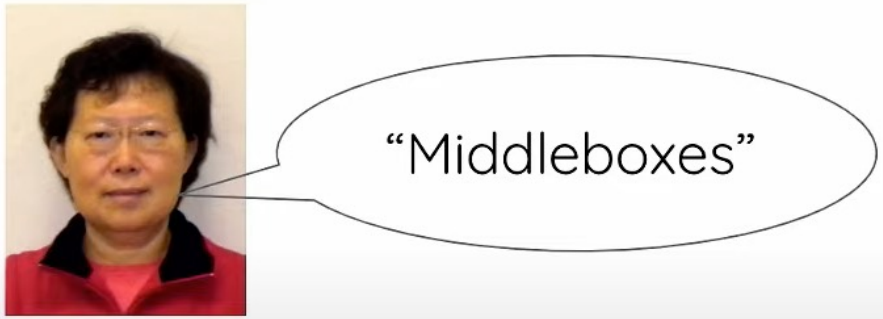
https://www.youtube.com/watch?v=hCm0-MUs5nk

38

Eroding End-to-End

Network devices started to read/modify end-to-end information

- NATs: transport port number, checksum
- Others: most transport header fields, state machine



“The death of an end-to-end internet”

- Over the past two decades, the Internet’s runaway success has caused its end-to-end architecture to be eroded by an organic proliferation of interposers or middleboxes, such as NATs, firewalls, IDS devices, and performance enhancing proxies, which now pose serious roadblocks to the Internet’s evolution.
- Through studies and anecdotes, we will discuss the architectural (and political) implications of middleboxes in our beloved end-to-end architecture, and strategies for protocol development and deployment in this brave new world.

END-TO-END ARGUMENTS IN SYSTEM DESIGN

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

M.I.T. Laboratory for Computer Science

This research was supported in part by the Advanced Research Projects Agency of the U.S. Department of Defense and monitored by the Office of Naval Research under contract number N00014-75-C-0661.

Revised version of a paper from the Second International Conference on Distributed Computing Systems, Paris, France, April 8-10, 1981, pp. 509-512.: Copyright 1981 by The Institute of Electrical and Electronics Engineers, Inc. Reprinted with permission.

Published in ACM Transactions in Computer Systems 2, 4, November, 1984, pages 277-288.

One can read an even earlier version of the paper at
http://web.mit.edu/Saltzer/www/publications/rfc/csr-rfc-185.pdf (April 1980, 9 pages

Internet started without middle boxes

- The network function at the time was simple: datagram delivery to destination address based on routing
  - No address shortage (yet)
    - No NAT
  - Few users
    - No need for scalable content distribution
    - No replicate servers → no load balancer
  - No big security concern (security issues always in people's mind; no immediate danger, no immediate action
    - no firewalls
    - no IDS devices
- At the time: lots attempts to enhance the middle for "better delivery"
  - none succeeded that well

Things changed quickly starting mid/late 90s

- NAT, firewalls, AKAMAI boxes: packet manipulators between source-destination host → "middleboxes"
- Tons of middle boxes today; the Internet would not work without them
  - Have to have NAT
  - Have to use proxies(cope w/ mobility, performance)
  - Have to better utilize network bandwidths
  - Have to deal with load scalability (CDNs)
  - Have to do something to protect end hosts/users
    - In particular, MaaS (mitigation as a service)

Patching up TCP/IP's deficits

- What's more: some of the above (especially the last two) increasingly provided by dominant players

END-TO-END ARGUMENTS IN SYSTEM DESIGN

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

M.I.T. Laboratory for Computer Science

In a system that includes communications, one usually draws a modular boundary around the communication subsystem and defines a firm interface between it and the rest of the system. When doing so, it becomes apparent that there is a list of functions each of which might be implemented in any of several ways: by the communication subsystem, by its client, as a joint venture, or perhaps redundantly, each doing its own version. In reasoning about this choice, the requirements of the application provide the basis for a class of arguments, which go as follows:

> 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.

What, if any, would be the function(s) that cannot be done in the middle?

- Lets open for discussion: look at popular middleboxes again:
  - Tons of middle boxes today; the Internet would not work without them
    - Have to have NAT
    - Have to use proxies(cope w/ mobility, performance)
    - Have to better utilize network bandwidths
    - Have to deal with load scalability (CDNs)
    - Have to do something to protect end hosts/users (MaaS)
- Some of the above intercept end-to-end TLS (or equivalent)
  - Besides, TLS (or equivalent): tools to support security
    - Not security itself

One thought: the need for true end-to-end security

- Distributed apps can help nudge the future away from centralization
  - Why don't we have such things today?
    - Can your phone dare to connect to mine directly?
- Can we have smart homes that do not depend on cloud services to operate
- We love cloud services (benefitting from economy of scale); we do not want to be controlled by cloud services
- How do we get from here (being controlled) to there (using the services, not being controlled)