

1. Summary of the Paper

End-To-End Arguments in System Design [1] lays out a straightforward argument for placing responsibility on end system applications for ensuring their desired application-level protocol guarantees. End To End refers to two systems connected through the internet or other series of networks that have some sort of application running on them. The paper lays out 4 possible paths to meet ensuring system design properties, which I like to think of as 'guarantees':

1. Only client applications provide the guarantee
2. Only the underlying infrastructure provides the guarantee
3. Both client and infrastructure provide the guarantees, but independently of each other
4. The client and infrastructure work together to provide the capability guarantee

Saltzer et al argue that reliability and assurance guarantees must *at least* be enforced by the two end systems, or option 1. Because of that, it is more expensive and complex to also implement the same guarantees in lower layers unless there is some sort of extra consideration (Such as errors that are caught and retransmitted at lower layers may prevent higher level, larger, and more expensive retransmissions). The main claim of the paper is that is not feasible for many guarantees to be provided solely by the interconnecting infrastructure, especially when knowledge and help of the application layer is required.

2. Critical Review

(~1.5 pages)

a) What problem(s) did the paper address?

Saltzer et al argue that in many cases where end applications need operational guarantees such as careful file transfer, encryption, duplicate message suppression, and byte ordering, the end applications necessarily must implement at least some amount of checking. It is impossible to fully rely on the underlying infrastructure to provide these capabilities. This paper claims that this design pattern of end-to-end system design is not new, but it has not been talked about or studied in a direct fashion.

b) Why would anyone care about this particular problem? Who is the intended audience?

The audience of this paper are systems engineers, researchers, and designers of applications that are faced with design decisions while building reliability and security into their software. Because the internet is made up of many parts, is decentralized, and is generally assumed to be unreliable, Saltzer et al. make very direct observations to help these teams place their application functions at the correct layer.

c) What are the existing gaps that necessitate this work?

The end-to-end argument had not been directly described in a concise way prior to the publication of this paper, and many yet-to-be-designed applications and systems could benefit from their designers contemplating the fundamental argument of this paper.

d) What is the authors approach in solving this problem?

The authors define the set of 4 possibilities in which a function could be located, then describe many real-life use cases showing the end-to-end design principle is superior.

e) Does the evaluation support their hypothesis?

In general, yes. For point-to-point networking with unreliable low-level infrastructure between high-level applications, their observations and evaluation are spot on.

f) What are the possible inefficiencies in their approach?

The authors assume that end-to-end applications are working together towards the same purpose, while supporting infrastructure may or may not be. In cases where one endpoint is malicious, infrastructure such as firewalls may be the entities in the end-to-end path that have the knowledge or capability to implement a function. Two applications owned by different jurisdictions may also have different definitions of reliability, encryption, etc, and underlying infrastructure may be required to enforce standards.

g) In your opinion, how can we address those inefficiencies?

At the end of the day, there is a person or entity responsible for initiating a particular workflow. That person or entity should be able to apply a set of guarantees to their workflow that can be enforced by underlying applications and infrastructure. Often, users will choose certain applications (i.e. firefox over internet explorer) because of their built-in reliability or encryption, but I can imagine a world where users themselves may be able to directly specify some of the capability guarantees that they need.

h) What are the future research questions left unanswered (explicitly or implicitly)?

The authors propose that the future should lead to a stack of layers, many of which may not be defined yet. They are providing the end-to-end design principle as a tool for those designing those unresearched layers.

i) Judging from all of the above answers, do the authors justify the conclusions they reach? Why or why not?

Yes, they do a thorough job of laying out their argument, use many real world examples, and do not say that their argument is the best in every case. They leave the future a bit open-ended, so it is hard to say the conclusions are not justified.

3. References

Literatur

- [1] J. H. Saltzer, D. P. Reed und D. D. Clark. „End-to-End Arguments in System Design“. In: *ACM Trans. Comput. Syst.* 2.4 (Nov. 1984), S. 277–288. ISSN: 0734-2071. DOI: 10.1145/357401.357402. URL: <https://doi.org/10.1145/357401.357402>.