Review of

End-to-End Arguments in System Design

JH Saltzer, DP Reed and DD Clark

What is the problem?

Providing a function at a lower level of the system has minimal value when compared to the effort needed to support that functionality at that level. Making design decisions that take into account the different layers of the systems while assigning functions to each of them is a crucial task. These decisions can affect both the functionality and performance of the system.

Summary

There has not been any formal recognition given for a method called the “End-to-End argument” that has been in use for many years albeit loosely. This paper tries to formulate different arguments using examples for why the “End-to-End” model would work especially since the emergence of data communication networks, where there is a need and a justification for such a model. The End-to-End argument can be understood as avoiding all unnecessary functionality from slowing down the underlying systems and delegating these functionalities to the endpoints.

Key Insights

- Having a functionality built into a lower layer in the system can become a costly mistake because it serves as a common layer to a variety of application and this may include many that do not need it or may need to make adjustments in order to counter it.
- There are a myriad of performance and functional advantages to isolating the functions of each layer and ensuring each layer only delivers what is necessary. The generalization of the underlying system also means newer applications can be added with minimal impact or change needed.

Strengths

- The paper attempts to explain each facet of the End-to-End argument using multiple examples and this enables readers to relate the argument to different modern systems as well.

- The End-to-End argument shows that incorporating the functionality at the endpoints significantly improves performance.

Limitations/Weaknesses

- The paper explains the need for End-to-End effectively and gives examples of the cases where it works but does not guide the readers on how to effectively implement it or even to identify where the endpoints could be. The paper aimed to provide guidance on this front but it is not clear that they achieve that.
- End-to-End model rests on the belief that the endpoints are to be trusted, it is not clear that this method accounts for any cases where this trust is not implicit.

Summary of Key Results

- End-to-End methods have been used in many popular applications in all system areas to deliver a variety of different functionalities, and hence empirically proving the success of employing such a model.
- The End-to-End argument is a kind of "Occam's Razor" because it is simplest to make the endpoints shoulder the responsibility to ensure that the system that lies under them are robust. This means that ensuring each layer only provides that which is absolutely necessary from that layer.

Open Questions

- How can we define a good heuristic for identifying the endpoints to which these arguments must be applied?
- The scope of the argument needs to be extended to cases where the implicit trust in the endpoints does not exist. How does one ensure that the endpoint has done its job?