Review of End-to-End Arguments in System Design

Gwendolyn Stockman

February 2, 2007

1 Summary

This paper presents a design principle that emphasizes making underlying systems more flexible for the applications that run on them. "End-to-end" design focuses on where to implement functions such as those to provide bit-reliability, delivery guarantees, encryption/decryption, and in-order delivery of packets over networks. Saltzer, Reed, and Clark make the point that there is no crisp and clear correct answer as to where to implement such functions. The general idea of "end-to-end" design is that functions implemented at lower levels should be thought of as building blocks, small and simple. While the more complex functions required by various applications should often be left to the applications themselves.

2 Pros and Cons:

When application-specific functionality is implemented at the underlying system layer all applications, regardless of whether the functionality is required or not will be slowed down. A major benefit of end-to-end design is that this functionality is instead implemented at the application layer that prevents applications without need of the functionality from paying for it.

However, some functionality may be implemented numerous times as a result of each application having its own version even if the requirements are the same. This could be easily fixed by having an extendable library.

3 Clean-slate:

While "end-to-end" design requires a layered model, this idea is not constrained by the current technologies and instead follows the clean-slate idea as much as is possible. The ideas in this paper are very versatile and can be applied to a variety of different technologies and need to be taken into account in the future, specifically when attempting to think and design on a "clean-slate."