1 Summary

Computer systems designers often ponder where exactly to place functions in a distributed system, especially one to do with communication protocols. Saltzer et al. articulate the well-established idea behind where to place abstractions in a distributed system. Their principal of ënd to end arguments”classify functions placed at low levels of a distributed system as redundant/of little value compared to the cost of effort for providing them there. Using examples like ”careful file transfer”, Saltzer et al. succcessfully convey the strength of the end to end argument.

2 Strengths of the paper

The ”careful file transferïs a well-chosen example to highlight the strength of the end to end argument based on the threats of a transaction that can’t be totally negated even if we imagine a potentially perfect network. Because even then, there could be issues stemming from hardware faults at the source’s disk storage, software faults anywhere along the network, or even hardware errors along the path. Simply storing a checksum along with each file that is recalculated on the host machine to verify whether the transaction maintained file integrity works. Why bother spending all the extra time and effort in building such a reliable, perfect network when that’s not the solution for the application program’s reliability challenge. Sort of echoing Butler Lampson hints, it’s important for the designers to seriously consider where efforts need to be concentrated to achieve the intended outcome. The example of telephone communication, in that a live conversation between parties can rely on a UDP-like network whereas a voice mail should probably rely on a TCP-like network due to the inability of a sender to repeat themselves as the listener hears, reinforces the end-to-end argument since it’s not really plausible for a network provider to somehow provide both functions centrally. Instead, different applications require different functions at their endpoints.

3 Major weakness of the paper

It’s always tough for me to come up with some weakness in these seminal papers. For the sake of argument, I wish the end to end argument extended beyond systems-level examples. For example, consider an interpreted language like Python versus a pre-compiled one like C++. Is it worthwhile to use C++ when we’re not dealing with low-level hardware, for example, according to the end to end argument?

4 Future work opportunities

The authors discuss how this principal appears in the most popular networking protocols like TCP (reliable, end-to-end connection) and UDP (‘best-effort’ protocol with low latency). They also draw comparisons between RISC and CISC through the same lens. It’d be interesting to consider today’s paradigms (e.g., 4G vs 5G, Bluetooth and Zigbee) through this lens too.