Prisoners Of Our Own Device

Micah Beck, University of Tennessee
mbeck@utk.edu
26 October 2023

I start with a thought experiment, in the form of an argument that might have been made in 1985 against the development of wide area packet forwarding in the Internet as an alternative to wide area virtual circuits as an evolution of the telephone system.

Imagine that, in 1985, a group of eminent leaders of the network research community, representing many of the leading US universities and corporations produced a white paper. This hypothetical paper would have supported the idea that new globally digital services should be delivered on top of virtual circuits[1], explaining their reasoning in this way:

> Services should be layered on top of virtual circuits. We already know how to deliver data over virtual circuits in the telephone system, and further improvements in speed and reliability, though desirable, will not transform digital services. Instead, we should focus on what digital services should be offered besides virtual circuits, and those services should be deployed on top of virtual circuits. This principle is merely a natural application of layering. Just as logical links (L2) are built on top of physical links (L1), packet delivery (L3) is built on top of L2, and global virtual circuits (L4) are built on top of L3, we should build all additional services on top of L4.

From our point of view, almost 40 years later, this would have seemed incredibly short sighted, clear evidence that these leaders simply "didn't get" the incredible potential of global Internet packet forwarding. The argument was made insistently by the proponents of the Internet that, even though data traffic was at the time a tiny fraction of voice traffic, it was growing exponentially. A further argument was made that ubiquitous digital applications would require the flexibility and low overhead of global packet delivery. While virtual circuits could ensure much stronger quality of service guarantees, the vast majority of mass market digital services did not require them or could adapt to a best effort environment. The Internet might not replace telephony and virtual circuits anytime soon (from the vantage point of 1985), but it could become an important alternative and evolve toward their quality of service (as it has).

End of thought experiment. I now turn to something that actually happened.

In 2021, a group of 18 eminent network researchers produced a white paper in support of the idea that new digital services should be delivered globally on top of packet delivery:

> Services should be layered on top of packet delivery. We already know how to deliver packets on the Internet, and further improvements in speed and reliability, though desirable, will not transform the Internet. Instead, we should focus on what other services the Internet should offer besides packet delivery, and those services should be deployed on top of packet delivery. This principle is merely a natural application of layering. Just as logical links (L2) are built on top of physical links (L1), and global best effort delivery (L3) is built on top of L2, we should build all additional services on top of L3. For convenience, we will say such services belong to the service layer or L3.5. Service implementations start at the service layer but may extend up through the application layer.

---

[1] Virtual circuits generalize telephone calls to implement high performance data transfer.

- Hari Balakrishnan, Sujata Banerjee, Israel Cidon, David Culler, Deborah Estrin, Ethan Katz-Bassett, Arvind Krishnamurthy, Murphy McCauley, Nick McKeown, Aurojit Panda, Sylvia Ratnasamy, Jennifer Rexford, Michael Schapira, Scott Shenker, Ion Stoica, David Tennenhouse, Amin Vahdat, and Ellen Zegura. 2021. [Revitalizing the public internet by making it extensible. SIGCOMM Comput. Commun. Rev. 51, 2 (April 2021), 18–24]

How should our evaluation of this modern-day statement compare to the hypothetical from 40 years ago? Does the argument that the existing common service layer should be preserved as the “thin waist of the protocol stack” hold up better at Layer 3 than it did at Layer 4? Let’s analyze the argument[2]:

1. “We already know how to deliver packets on the Internet, and further improvements in speed and reliability, though desirable, will not transform the Internet. Instead, we should focus on what other services the Internet should offer besides packet delivery…”
    - It would be more accurate to say that we know how to deliver loosely synchronous point-to-point packets on the Internet. Other network functionality, such as asynchronous and point-to-multipoint, are not supported.
2. “... and those services should be deployed on top of packet delivery.”
    - The conclusion that the common infrastructure need not support functionality other than loosely synchronous point-to-point packet delivery is presumably due to an end-to-end argument: Any other functionality required by applications can be implemented in layers above L3, starting with L3.5.
    - But some functionality cannot be effectively implemented without making reference to network topology, which is encapsulated within L3. Storage and processing are both extremely sensitive to resource placement within the network topology.
    - Services which are defined or optimized with reference to network topology in effect extend L3, which tends to reduce deployment scalability.
3. “This principle is merely a natural application of layering. Just as logical links (L2) are built on top of physical links (L1), and global best effort delivery (L3) is built on top of logical links (L2), we should build all additional services on top of L3.”
    - This would be possible if the existing common layer at L3 supported all required functionality, which it does not.

Critics of layering new digital services on top of virtual circuits noted that in 1985 the existing L4 did not support the functionality required by the Internet application environment. If today there are required network functions that cannot be supported by the current L3, the same criticism applies.

Critics of implementing new digital services on to of virtual circuits in 1985 knew that this could not be done without compromising the ability of the existing L4 to deliver the additional functions

[2] The brief discussion here is presented at greater length in a recent paper. [Beck & Moore, How We Ruined the Internet, arXiv:2209.03482306.01101, June 2023]

required at scale. This was due to the tradeoff between performance and generality. A similar tradeoff rules out substantial extension of the current L3, and to the addition of topology-sensitive services at L3.5.

The conclusion in 1985 was that the common service layer should be lowered from L4 to L3, enabling heterogeneity at L4 and above but at a cost in the ability to provide quality of service guarantees. Is it appropriate to consider lowering the common service layer from L3 to L2? What would this mean?

The idea of placing the common service layer at L2 is not new. Some past examples were:

- In 1997, David Clark of MIT considered the possibility of using ATM as the thin waist of the network stack, giving it the shape of a funnel placed on its end rather than an hourglass. Clark commented that “The virtue of this approach is that a single technology solution might be able to offer a sophisticated range of services and thus support an even wider range of applications than, for example, the Internet approach. So the funnel, while narrow at the bottom, might be wide at the top.” [Clark, “Interoperation, Open Interfaces and Protocol Architectures,” in The Unpredictable Certainty: White Papers, 1997].
- In the early 2000s, the Global Environment for Network Innovation (GENI) project contemplated creating a common service layer at L2. In particular, Jon Turner’s high performance programmable router project at Washington University in Saint Louis achieved interoperability by using Ethernet at L2 and defining a standard hardware interface for modules that would implement heterogeneous services at L3.

Without getting into a detailed analysis of previous proposals and implementation efforts, what has not been investigated is the provisioning of large scale storage and processing resources at L2 (see Figure 1). A different approach would be to create an interoperable environment that could support L3 services very different from those that are currently deployed at L3. This would be equivalent to the definition of a common local OS that could manage the storage and processing resources of a node (or closely coupled set of nodes) and communication that need not be global.

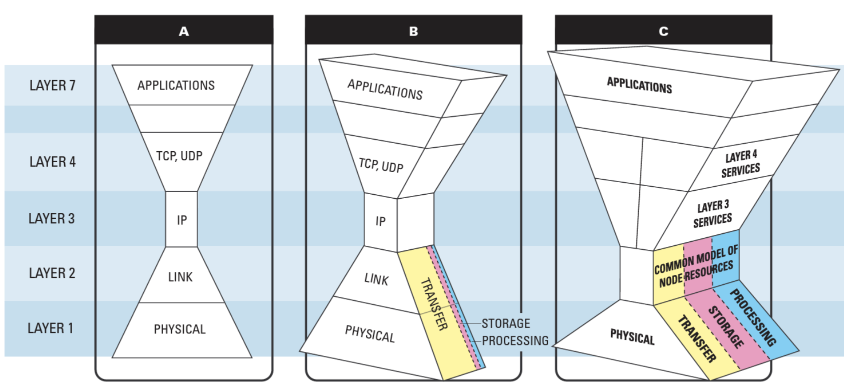

Figure 1. Lowering the “thin waist” to a generalized Layer 2

Such a generalized L2 would be capable of implementing the current style of packet delivery services at L3. But it could also manage resources sufficient to implement additional services that might be implemented at L3 or layered on top of one or more L3 services. Such services must today be implemented using private infrastructure in the form of middleboxes or edge servers. Making storage and processing available at L2 would enable a variety of higher layer services to use them without deploying private resources.

Although it was not described in these terms, the L2 approach is similar to the strategy taken by PlanetLab in the early 2000s. PlanetLab created a uniform mechanism for the allocation of “slivers” of network intermediate nodes which together formed a “slice” of its virtual networking infrastructure. Use of these resources, along with access to Internet connectivity, were made available through a common Linux-based node OS. Although communication was implemented in overlay on the Internet, the ability to manage node storage and processing resources and access them through a general programmable interface is analogous to the creation of a common local OS service layer at L2.

PlanetLab and GENI reached deployment as testbeds, not global infrastructure. Today, the FABRIC testbed is similar in its purpose as a high performance data intensive testbed which incorporates modern programmable routers that are collocated with overlay servers. It remains an open question whether an L2 service could be designed which included storage and processing but was also capable of being deployed in a scalable manner to support services worldwide and in a wide variety of environments.

The common services which have gained the widest deployment, shaping the modern Information and and Communication Technology (ICT) environment are Unix and the Internet. They revolutionized the fields of operating systems and networking respectively, crowding out competitors (those that remain have evolved to be similar to these in architecture while differing in details). If we could identify a common characteristic of these two most successful examples, perhaps we could apply it to a common generalized L2 service of a wide area infrastructure. This is the claimed role of the Hourglass Theorem, and the claim that a design approach of “minimal sufficiency” has been key to the deployment scalability of both Unix and the Internet [Micah Beck. 2019. On the hourglass model. Commun. ACM 62, 7 (July 2019), 48–57.].

The Hourglass Theorem states that, given a set of necessary higher layer services and applications that must be supported, a common service layer which is “logically weakest” will have the greatest variety of underlying services at lower layers than can support it. The minimal sufficiency design principle favors choice of the weakest possible common service which can support a class of goal applications. Note that logical weakness does not rule out providing strong services, it just argues against applying a universal requirement (throughout the infrastructure) of services beyond those that are minimally required.

A current effort to create such a minimally sufficient service appropriate to being adopted universally at L2 is Exposed Buffer Architecture (EBA). [Micah Beck, Deployment scalability in exposed buffer processing, 17th IEEE International Conf on Mobile Ad-Hoc and Smart Systems, Delhi, India, December 10-13, 2020] EBA has been implemented in overlay (in a form called Logistical Networking) on a national scale (using PlanetLab and REDDnet, an NSF-funded Major Research Infrastructure project led by Paul Sheldon of Vanderbilt University). The EBA architectural approach may be adopted in a major European data intensive high performance computing project. However there has to date been no native EBA implementation in the US.

There was a time when the End-to-End arguments were held up as the best guide available to the definition of a common service layer which can be deployed at global scale. It was an attempt to generalize a number of specific architectural advances prior to 1985 including Multics, Unix, the Internet and RISC. A recent retrospective analysis [Beck & Moore, “How We Ruined The Internet”, arXiv:2209.03482306.01101, June 2023] details how limitations of the Internet architecture has fallen short in the support of important Web and multimedia applications and at large scale, contributing to many of the problems of the modern ICT environment.

The End-to-End arguments are not couched in sufficiently precise terms to be characterized as valid or invalid. In some cases, the predictions which have been attributed to End-to-End arguments have not been born out in practice. The Hourglass Theorem is a formal principle which in some cases agrees with the common interpretation of End-to-End arguments, but which diverges from them in some cases where they have fallen short. However, there may today be a widespread disillusionment with the application of logical principles to system design. This may have led to reason taking a back seat to intuition and expedience.

I do not claim to have argued conclusively for the efficacy of moving the common service layer to L2. I do claim that there are compelling reasons give serious consideration to such an approach. The history of the networking community’s failure to date in creating a means of extending the Internet which achieves widespread acceptance and longevity supports this reasoning.