The Programmable Network Stack

Micah Beck, University of Tennessee

According to Wikipedia, a Protocol Stack is "…an implementation of a computer networking protocol suite." This definition is accompanied by an illustration of the OSI model with familiar layers: Physical, Link, Network, Transport, etc. The Wikipedia entry goes on to assert that "…strictly speaking, the suite is the definition of the communication protocols, and the stack is the software implementation of them. [Webopedia]"

Conflating a protocol and its software implementation is a fundamental confusion. The process that software describes cannot be executed using only a communication protocol – it *requires* storage (memory) and processing. If the execution of the Network Layer implementation is supported by the Link Layer, then where does this higher layer get these other necessary resources?

Storage and processing are implemented in the router services that in fact implement the protocol stack. Models that leave out the full resources and services of the router do not account for these necessary resources. If programmability is to appear in the stack (e.g., at the Network layer), an accurate model must represent these resources. Otherwise here is no architectural explanation for how programmability can be supported.

The physical resources of the router do include storage and processing, and these are used to implement Link Layer services. These physical resources are also used to implement the node operating system, administrative functions, routing and peering and other features of the local system. All of these, together with the implementation of Link Layer protocols, are available to the implementation of the Network Layer, along with some fundamental physical layer resources (such as instruction execution and access to memory and secondary storage). But these general local resources are not included in the conventional description of the Link Layer (see Figure 1A).

If we generalize our model of the Physical and Link Layers to include all of the local resources of the intermediate node, then we have a more complete picture of how the Network Layer is implemented [Future of Information and Communication Conference, Beck & Moore, 2019, "Interoperable Convergence of Storage,networking and Computation"]. In this more complete picture of the stack (see Figure 1B), all of the resources of the router that go into implementing the Network Layer are represented in the Local Layer (a renaming of a generalized Link Layer which models local storage and processing). They are available to use in expanding the services of the network Layer. However that would mean adding functionality to the Network Layer that models these resources. The Network Layer, as the common "spanning layer" of the current architecture, cannot be made stronger or more complex without jeopardizing its deployment scalability (tendency to be widely adopted voluntarily). So the inclusion of programmability cannot be accounted for by reference to the current stack.

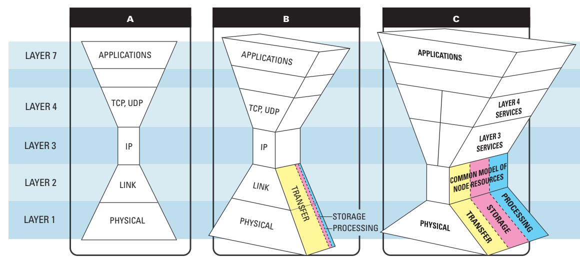

Fig. 1. A: Internet stack; B: Exposed view; C: The Programmable Stack

One option is to define a more general stack, with the spanning layer being the Local Layer (see Figure 1C). In such a stack the current Network Layer can be supported, but so can other services. The basis of interoperability in this wider community of implementation would be a virtual model of node resources (the node OS and communication in the local area). In this model all of the resources of the Local Layer would be accessible to the implementation of various Network Layer services.

Such a strategy would require the community of Network Layer service implementers to adopt a common model of the local node and network. The design space of possible abstractions is large, but there is a principle that can be used to guide the design of a service that might exhibit the necessary deployment scalability, namely the Deployment Scalability Tradeoff [Communications of the ACM, Beck, July 2019, "On the Hourglass Model"]. That principle states that there is a correlation between the deployment scalability of a spanning layer and the degree to which it is logically weak, simple, general. These are the same criteria used in the design of the two most successful infrastructures of the computer era, the Unix/POSIX kernel and the Internet.

In the case of both POSIX and the Internet, the design of the common interface was highly disciplined and constraints have been applied for decades by consortia consisting of users and implementers. These successful infrastructure efforts were motivated by a belief in the underlying design principles (even when they were not clearly stated but intuited by leading architects) and an understanding of the value of interoperability to both the implementers of supporting platforms and of applications. These efforts have shaped the computer revolution, and it is our role to continue and extend them to new and broader domains.