Position Paper for the
Applications and Services in the Year 2021

Submitted by
John Day, Lou Chitkushev, Ibrahim Matta
https://en.wikipedia.org/wiki/John_Day_(computer_scientist)
http://www.bu.edu/met/academic-community/faculty/full-time/lou-chitkushev/
http://www.cs.bu.edu/~matta/
Boston University

To most people, the Internet *is* its Applications, rather than all that supports the applications. Or more precisely it is the Web.[1] That perception contains more than a grain of truth because support in the Internet architecture for applications is basically non-existent. Even though, the basics were known during the ARPANET, nothing much has been done since USING (Users' Interest Group) was disbanded in 1974 and the stopgap kludges of well-known ports and DNS[2] were created, which basically limits the architecture to a rudimentary client-server model. We are already seeing this vacuum with patches all done from the perspective of "just solve my problem."

Our work on network architecture discovered that networks consisted of a recursive structure of layers (which seemed to contradict Dijkstra's rule that functions do not repeat), we have been forced to reconcile the 'contradiction.' Because we characterized a layer as "a distributed application that does IPC (Interprocess Communication)," which of course begged the question, "what is a distributed application?" These points might have been mere curiosities, if it were not that implications of the model[3] have required us to explore the nature of applications and services beyond the simple support of IPC.

The apparent contradiction was resolved decades ago with the recognition that the primary importance of layers is that they are a locus of distributed shared state, not merely local modularity. From which it follows that functions don't repeat in the same scope.[4] This led to the realization that all processes contain the three fundamental components of an operating system (scheduling, storage management, and IPC management). Successive recursions of these processes operate over different scopes, sometimes referred to as kernel, supervisor, and user (think of threads and heap management as merely the OS of a process at user level). The traditional OS is a heterogeneous distributed management system. Hence, operating systems are recursive as well.[5] Given that, naturally the Processes that make up a distributed application have tasks that perform these same 3 infrastructure components to manage the internal operation of the distributed application. This yields a single unifying model for distributed applications, operating systems, and networks, where for example a layer for a network is a distributed application, with routing and other resource allocation functions (scheduling), creating and maintaining the partially replicated data shared by the members of the layer (storage management), managing the use of the lower layer (IPC Management) and so we find that flow allocation, multiplexing, relaying, error and flow control are the tasks (the specific application) of the IPC Distributed Application. This model has proven to be very powerful and provide many new insights.

Of course, the biggest problem for the Internet, as always, is the lack of a full naming and addressing architecture and almost a complete lack of understanding of the fundamentals of addressing, as indicated by the red-herrings created by loc/id split and ICN[6] that have so distracted the field. While some major breakthroughs have been made in the nature of applications and application naming (really addressing), such as the distinction between the application process and the application entity, and encouraging recent insights on the nature of locality in data based on 30 year-old work, there remains much to explore. We believe that given the successes our model has yielded so far that it will continue to provide valuable insight and direction as we continue our explorations, especially with the joint EC project, ARCFIRE, for experiments over FIRE+ and GENI.

If accepted we intend to apply for travel support.

[1] Since we have found that there is only one application protocol, it is not surprising that HTTP has assumed this role indicating that considerable simplification is possible.

[2] Based on the misconception, still pervasive, that naming the host has anything to do with the problem. It doesn't.

[3] For example, that a global address space is unnecessary.

[4] Dijkstra's THE operating system [CACM 11:5, 1968] was so hardware constrained that he could only see one scope.

[5] Contrary to first thought virtualization is not recursion, although it is a hack approach to it.

[6] ICN is analogous to IPv4 pre-CIDR. There addresses were inexplicably assigned in order of requests, ignoring that addresses should exhibit "nearness." With no locality in the address, this required large and expanding router tables. This was found to be untenable for 32 bits with IPv4; it will be impossible with data, even with hashing.