A Critique of

"An Architecture for Active Networking"

for UCSB CS 290I Winter 1999

Mihai Christodorescu

A mathematical model of an Active Network is presented in this paper. Each packet contains a triplet of data specifying a function ID, a state ID, and a list of arguments for the function. When a packet arrives at the active node, the node runs the function given by the ID, with the right arguments, in an environment retrieved from a cache. The resulting environment is cached for later retrieval, and the (possibly modified) packet is routed to the next hop.

One potential problem is the presumed relationship between the node–cached state, and the flow of packets. If the flow of packets is defined as the packets parts of a message (e.g. all the packets part of an MPEG movie), then the standard IP routing does not guarantee the same path for all the packets in the flow. That would mean that the state might not be the same for all the packets part of the same message. The paper does not consider this option. Their test setup involves a linear topology sender–router–router–...–router–receiver, which might explain why they have not consider multiple paths in a flow.

An Active Networking infrastructure is not clearly defined, and

I tend to believe they favor separate implementations for each AN function. Since their main focus is efficiency, and performance gains due to AN nodes, the AN functions / filters might be seen as independent from each other. The absence of a clear infrastructure will reduce the amount of code sharing, and it will make functionality aggregation complicated if not impossible.

They implement a function called "Active Processing", that is basically congestion control using frame dropping, where the drop function can be specified by the application-level protocol (which knows how to organize the data in logical units, and how network frames are related inside this units). Also, the feedback mechanism is also application-level defined, as, in their configuration, the feedback needed for congestion control is provided at the application-level. Thus a lot of functionality is moved in the upper layers, where more information is available.

Unfortunately the paper does not have any implementation details which would make it more interesting. The jump from theoretical approach to statistics is quite disconcerning.