Review of Internet Indirection Infrastructure

Swaminathan Sundararaman

March 26, 2008

This paper explores the possibility of routing packets on top of overlay networks. The main problem with the internet architecture is that it cannot efficiently support multicast, anycast, and host mobility where the sending host no longer knows the identity of the receiving host. Most importantly both the sending and receiving host need not be fixed. They achieve this through indirection (i.e., they route packets on top of chord where the sender sends the packet to an i3 server that has a trigger for a particular id and these triggers are registered by the client who are interested in receiving the packets that has a particular id).

Pros:

- Clean way of hiding location of both sender and the receiver and both of them can be mobile.
- Simpler way to implement multicast, anycast or host mobility.
- Heterogeneous multicast is a cool idea and would be useful in today's situation where receivers are almost always heterogeneous.

Cons:

- End-to-end guarantees provide by the internet is no longer valid as we have split the path from source to destination into two. Source to i3 server hosting the trigger and i3 server to the destination.
- It is not clear how TCP would work in i3. As mentioned above the path from source to destination is split into two and RTT computation, congestion avoidance metric, etc are no longer accurate.
- Routing on to of an overlay is very inefficient. It does not take locality into consideration at all. Sha-1 on IP address to get node ids is root cause for the triangle problem and routing inefficiencies.
- Even though stacking seems to be a nice idea, the idea of forwarding packets from each server to other servers who have triggers for the ids in the identifier stack is not good as its very inefficient and adds a large delay to the packet delivery (especially intermediate node does transcoding/processing before forwarding it to other nodes).
- Server selection algorithm in i3 is sub-optimal.
- Backup trigger for robustness doesn't seem like a clean idea. Duplicating all packets is not efficient as stacking automatically forwards the packets to other ids.
- The idea of sender informing the receiver to reestablish the trigger when i3 server hosting the trigger fails using alternate triggers increases the state stored in each s3 server.
- Their solution to avoid hot-spots is not clearly explained.
- Very weak evaluation section. It would have been good to see an exhaustive evaluation section to clearly see the impact of the routing on top of overlays.
- Cost model for i3 is not clear.