OP2: A richer network abstraction?

This document argues that it is possible to implement the "choose path" feature without violating layering. The proposed solution relies on the introduction of a new *path* layer in the network stack, operating between the transport and network layer. The new layer exposes the following abstraction to Internet end-systems: finding multiple functional paths between a pair of Internet nodes in different autonomous systems (ASs) and providing performance metrics about these paths.

The paths presented to the source process each take the form of an ordered sequence of autonomous system numbers (ASNs) along with a *performance vector* characterizing how well the path meets pre-defined performance criteria (e.g., latency, throughput). The sequence of ASNs indicates to the source process the exact set of ASs that a message would go through if it selected this path. This effectively allows messages to avoid geographical areas.

The path layer can be queried for a set of paths to a specific remote process identified by its IP address. A source process is free to pick any of the returned path and informs the path layer of its choice by writing the full sequence of ASNs in the path layer's header. The latter also contains IP addresses of the source and remote process.

The network layer's purpose changes significantly: the abstraction it provides to Internet end-systems is now limited to finding a functional path between two nodes *in the same AS*. While IP addresses still need to be global, the ones used by network-layer switches for forwarding packets must now reside in the same AS (except during AS crossings). The other layers remain unchanged in the interface they provide to adjacent layers.

Implementing the solution requires that routers at domain boundaries become path-layer switches. On receiving a packet from another AS, a path-layer switch examines its header to find the next AS on the path chosen by the source process. It then passes back the packet to its network layer, setting its own IP address as source and the IP address of a path-layer switch that can reach the next AS in one hop as destination.

Each pair of path-layer switches in the same domain can run a protocol to evaluate various performance metrics between them. Together with performance summaries received from adjacent ASs, this information can be used to establish performance vectors during path probing. This protocol doesn't need to scale well because the number of path-layer switches in a single AS remains small.

This solution achieves the desired goal and requires minimal changes to the existing network architecture to be implemented. Secondly, it doesn't impose a significant performance overhead on processes that do not care about the "choose path" feature. An empty sequence of ASNs in the link layer's header can let the network know it is free to choose any path it finds suitable. However, the path probing operation is non-trivial and might not scale well with the number of ASs on the path, making it a potential performance bottleneck in this new architecture.