Pathfinder

The pathfinder program finds the path taken by a packet to reach any URL, and also computes the longest common path when given a list of URLs. The implementation is similar to `traceroute`, with a max hop limit of `30`. The program has an element of parellelism - it uses one thread per hop value to find the path.

Design

The input file with all the URLs is read and a Vector of URLInfo structures is initialized. The URLInfo structure contains all relevant information for finding the path for one URL - and the fields have been documented in `pathfinder.c`.

Then a DNS lookup is performed to get the addresses from the URLs (in parallel using the GLIBC function `getaddrinfo_a`). A socket is created for each URL from which a UDP datagram is sent - with TTL value modified using `setsockopt`. Another socket is created using the `SOCK_RAW` flag for receiving the ICMP replies.

Then, multiple threads are spawned - max hop limit (30) of them. In each thread, the raw socket is added to an `epoll` instance. If the raw socket is not ready to be read, then a UDP datagram is sent. Else the ICMP reply is processed. In order to find out the actual destination of the packet from the ICMP hop limit exceeded reply, the source and destination port numbers have been chosen carefully such that there is a 1-1 mapping between these port numbers and a URL. A hashtable is used to resolve this mapping.

After finding the path for all URLs, a naive brute-force algorithm is used to compare all the paths and find the longest common path.