

IP/TCP Additions for Computer Networks

Joshua Hill

CS1680

Abstract:

In this project, I constructed a RFC-compliant version TCP on top of a Virtual IP Network with my partner. This project was implemented using almost exclusively goLang, and incorporates a link layer, IP forwarding, and routing in the IP Network. On top of this, the TCP uses the lower-level IP Network to support an API for sockets, equipped with a TCP state machine, transport layer, sliding window protocol, and REPL interface. The goal of the Capstone was to add additional features to the IP/TCP socket implementation. Specifically, I allowed support in the Virtual IP Network for a 'traceroute' REPL command. A traceroute request attempts to print out a successive path of network IPs in which a packet could take through the network to arrive at a destination IP. My implementation follows the Internet Control Message Protocol (ICMP) traceroute request, and was successful in providing accurate output. Additionally, I integrated a congestion control algorithm (Tahoe) in the TCP. Congestion control algorithms such as Tahoe aim to prevent packet loss in a congested network from the side of the sender. This algorithm works with the sliding window protocol to limit the maximum size of data a given node can send in the network. Both of these additions ultimately serve to add additional functionality to the IP/TCP project in Computer Networks.