

Integrating packet scheduler to WireGuard

Objective

WireGuard provides VPN tunnels through network interface peers. Unlike IPSec, the network interfaces are responsible for encrypting outgoing packets. Wireguard uses UDP between peers. Currently WireGuard has no way to combat *buffer-bloat* problem. As WireGuard interfaces are responsible for encrypting outgoing data in *softirq*, they should also implement the AQM techniques to avoid *buffer-bloat* problem (AQM in forwarding layer is not sufficient). I would like to integrate fq_codel and PIE qdiscs to the sending pipeline of WireGuard. I also like to optimize the qdisc implementation by parallelizing it, offloading the computation to NIC hardware and auditing locks. Finally, I would like to implement a RCP-like explicit feedback mechanism by processing L3/L4 headers (ECN, Ipv6 FlowInfo and TCP option field) of incoming data.

Implementation

Timeline

Simulated a call center where calls are generated from different probabilistic distributions and assigned to agents based on their scheduled data. Different skill matrices were being calculated.

Implemented an interval tree to make the call assignment efficient instead of searching all the agents.

About Me

I am a PhD student of computer science department of Kent State University. I am doing research on *Network-centric TCP for interactive video streaming*. I am acquainted with TCP and Qdiscs in Linux kernel. I also worked on Mininet and GStreamer. Currently I am getting acquainted with WireGuard, Netfilter, nftable and Ethernet driver. I had my Bachelor degree from Bangladesh University of Engineering and Technology (BUET). I worked as a software engineer for five years. I am proficient in C, C++, Java, Shell Script and Python. I deeply care about Linux and free software in general. I wish to remain engaged in Linux kernel development in the future.

My Github: <https://github.com/tamimcse>

My resume: <https://web.cs.kent.edu/~mislam4/resume.pdf>