Contribution ID: **300** Type: **not specified**

MPTCP: Extending kernel functionality with eBPF and Netlink

Tuesday, September 13, 2022 5:30 PM (30 minutes)

Multipath TCP (MPTCP) was initially supported in v5.6 of the Linux kernel. In subsequent releases, the MPTCP development community has steadily expanded from the initial baseline feature set to now support a broad range of MPTCP features on the wire and through the socket and generic Netlink APIs.

With core MPTCP functionality established, our next goal is to make MPTCP more extensible and customizable at runtime. The two most common tools in the kernel's networking subsystem for these purposes are generic Netlink and BPF. Each has tradeoffs that make them better suited for different scenarios. Our choices for extending MPTCP show some of those tradeoffs, and also leave our community with some open questions about how to best use these interfaces and frameworks.

This talk will take MPTCP as a use-case to illustrate questions any network subsystems could have when looking at extending kernel functionality and controls from the userspace. Two main examples will be presented: one where BPF seems more appropriate and one where a privileged generic Netlink API can be used.

As one example, we are extending the MPTCP packet scheduler using BPF. When there are multiple active TCP subflows in a MPTCP connection, the MPTCP stack must decide which of those subflows to use to transmit each data packet. This requires low latency and low overhead, and direct access to low-level TCP connection information. Customizable schedulers can optimize for latency, redundancy, cost, carrier policy, or other factors. In the past such customization would have been implemented as a kernel module, with more compatibility challenges for system administrators. We have patches implementing a proof-of-concept BPF packet scheduler, and hope to discuss with the netdev/BPF maintainers and audience how we might best structure the BPF/kernel API – similar to what would be done for a kernel module API – to balance long-term API stability, future evolution of MPTCP scheduler features, and usability for scheduler authors.

The next customization feature is the userspace path manager added in v5.19. MPTCP path managers advertise addresses available for multipath connections, and establish or close additional TCP subflows using the available interfaces. There are a limited number of interactions with a path manager during the life of a MPTCP connection. Operations are not very sensitive to latency, and may need access to a restricted amount of data from userspace. This led us to expand the MPTCP generic Netlink API and update the Multipath TCP Daemon (mptcpd) to support the new commands. Generic Netlink has been a good fit for path manager commands and events, the concepts are familiar and the message format makes it possible to maintain forward and backward compatibility between different kernel versions and userspace binaries. However the overhead of userspace communication does have tradeoffs, especially for busy servers.

MPTCP development for the Linux kernel and mptcpd are public and open. You can find us at mptcp@lists.linux.dev, https://github.com/multipath-tcp/mptcp_net-next/wiki (soon via https://mptcp.dev), and https://github.com/intel/mptcpd

I agree to abide by the anti-harassment policy

Yes

Primary authors: MARTINEAU, Mat (Intel); BAERTS, Matthieu (Tessares)

Presenter: BAERTS, Matthieu (Tessares)

Session Classification: eBPF & Networking

Track Classification: eBPF & Networking Track