As part of the task to event tracing with static tracepoints, I decided to investigate on how to trace the events for a simple packet exchange between a tcp server & client.

As part of this task, I wrote simple TCP server(https://github.com/sbcd90/tcp-hello-world-kernel-trace/blob/master/src/HelloServer.cpp) & a simple TCP client(https://github.com/sbcd90/tcp-hello-world-kernel-trace/blob/master/src/HelloClient.cpp).

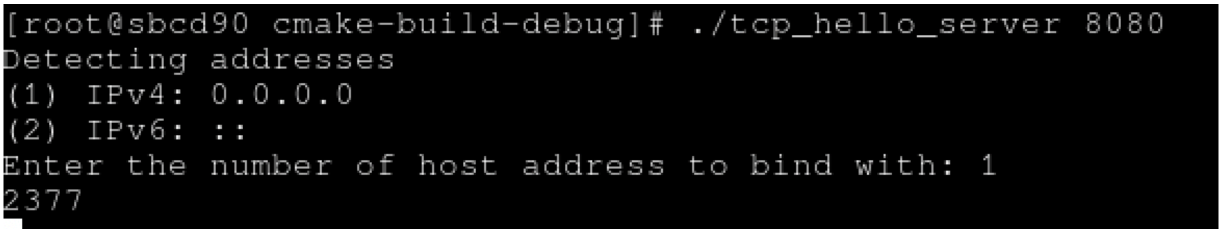
The TCP server listens for open connections on a particular port & TCP client connects to it & a simple “Hello World” message is exchanged.

Now, I tried to figure out which could be the relevant tracepoints to trace this communication.
Two events were identified.

- events/net/netif_receive_skb
-events/tcp

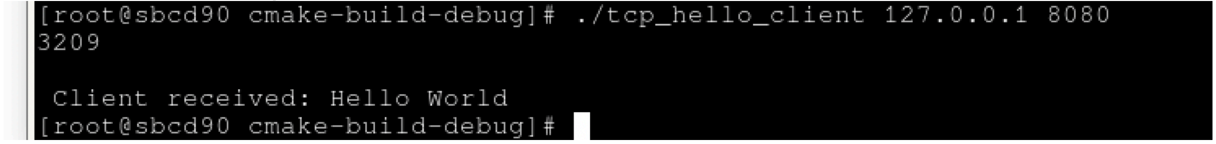
Captured the traces along with running the tcp server & client. Followed the steps described in this link.
https://github.com/sbcd90/tcp-hello-world-kernel-trace#execute

The TCP server running on port 8080 has process id: 2377 as shown in screenshot below.



```
[root@sbcd90 cmake-build-debug]# ./tcp_hello_server 8080
Detecting addresses
(1) IPv4: 0.0.0.0
(2) IPv6: ::
Enter the number of host address to bind with: 1
2377
```

The TCP client has process id: 3209



```
[root@sbcd90 cmake-build-debug]# ./tcp_hello_client 127.0.0.1 8080
3209

 Client received: Hello World
[root@sbcd90 cmake-build-debug]#
```

On checking the event tracepoint ***events/net/netif_receive_skb (https://github.com/sbcd90/tcp-hello-world-kernel-trace/blob/master/trace/net_receive_skb_trace.txt)*** It correctly captures the events of the communication.

Similar trace is also captured for ***events/tcp*** static tracepoints (https://github.com/sbcd90/tcp-hello-world-kernel-trace/blob/master/trace/tcp_trace.txt)

Also, captured the function trace by setting ***echo function > current_tracer*** to further learn about tcp communication.

Was reading an interesting article on network packet tracing. https://blog.yadutaf.fr/2017/07/28/tracing-a-packet-journey-using-linux-tracepoints-perf-ebpf/ Will work further on it in my leisure time.