

CN Problem Sheet #3

Problem 3.1: *IP over UDP tunnel (using the socket API)*

(10 points)

The Linux operating system supports Tun/Tap network interfaces that emulate a networking interface towards the protocol stack residing in the operating system kernel and that hand packets (or frames) over to a program running in user space. A Tun interface attaches to the IP network layer and exposes IP packets to a user space program while a Tap interface attaches to the data link layer and exposes Ethernet frames. The code available at

<http://cnds.eecs.jacobs-university.de/courses/cn-2017/src/tuntap/>

demonstrates how to create a Tun or a Tap interface and how to read packets or frames from it.

Write a program implementing an IP over UDP tunnel. Your program opens a Tun interface and sends the packets received from the Tun interface via a datagram socket (i.e., via UDP) to a remote instance of the same program. The remote instance puts IP packets received from the datagram socket back into the local networking stack via its local Tun interface. Similarly, packets received from the local Tun interface are forwarded to the remote destination. The code to send and receive datagram packets should be IP version agnostic (i.e., it should support IPv4 and IPv6 or any future IP version).

Your tunneling program must implement the following command line options:

```
-s <host>    forward UDP packets to <host> (defaults to localhost)
-p <port>    forward UDP packets to <port> (defaults to 4242)
-a <addr>    receive UDP packets on interface <addr> (defaults to any)
-l <port>    receive UDP packets over <port> (defaults to 4242)
-i <ifname>  name of the interface to be created
```

Additional hints:

- Make sure your source code is clearly structured.
- Your program has to react to either packets coming from the Tun interface or packets received on the UDP datagram socket. Hence, you need to implement an event loop. (Do not multiple threads, there is nothing CPU intensive where threads would help.)
- Take a look at the socket example code discussed in the Operating Systems course:

<http://cnds.eecs.jacobs-university.de/courses/os-2016/src/daytime/>
<http://cnds.eecs.jacobs-university.de/courses/os-2016/src/daytimed/>

Feel free to integrate the example code for handling UDP sockets and the example code for building an event loop into your solution.

- You can use Mininet to test your tunnel daemon over more complicated topologies.