

Integration of the packetdrill Testing Tool in INET

Irene Rüngeler (i.ruengeler@fh-muenster.de)

Michael Tüxen (tuexen@fh-muenster.de)

Outline

- Testing in INET
- packetdrill
- packetdrill in INET
- Summary and Outlook

Testing

- INET supports smoke testing, fingerprint testing, functional testing, statistical testing
- Testing of transport protocols can make use of this, but also needs flexible ways to
 - Control the interaction with its upper layer
 - Control the interaction with its peer

packetdrill

- Developed by Google and released under GPL in 2013
- Script based testing tool for transport protocols
- Allows testing TCP and UDP stacks using local mode and remote mode
- Runs on Linux, FreeBSD, ...

Address Configuration



Address blocks used (RFC 5735):

192.0.2.0/24 Test-Net-1

192.168.0.0/16 Private Use Networks

UDP Example

```
0.000 socket(..., SOCK_DGRAM, IPPROTO_UDP) = 3
+1.000 sendto(3, ..., 1000, 0, ..., ...) = 1000
+0.000 v udp(1000)
+1.000 ^ udp(1000)
+0.000 recvfrom(3, ..., 1000, 0, ..., ...) = 1000
+1.000 close(3) = 0
```

TCP Example

```
0.000 socket(..., SOCK_STREAM, IPPROTO_TCP) = 3
+0.000 setsockopt(3, SOL_SOCKET, SO_REUSEADDR, [1], 4) = 0
+0.000 bind(3, ..., ...) = 0
+0.000 listen(3, 1) = 0
+0.100 ^ S 0:0(0) win 32792 <mss 1460, sackOK, nop, nop, nop, wscale 7>
+0.000 v S. 0:0(0) ack 1 <...>
+0.100 ^ . 1:1(0) ack 1 win 257
+0.000 accept(3, ..., ...) = 4
+0.100 ^ F. 1:1(0) ack 1 win 260
+0.000 v . 1:1(0) ack 2
+0.020 close(4) = 0
+0.000 v F. 1:1(0) ack 2
+0.000 ^ . 2:2(0) ack 2 win 32792
```

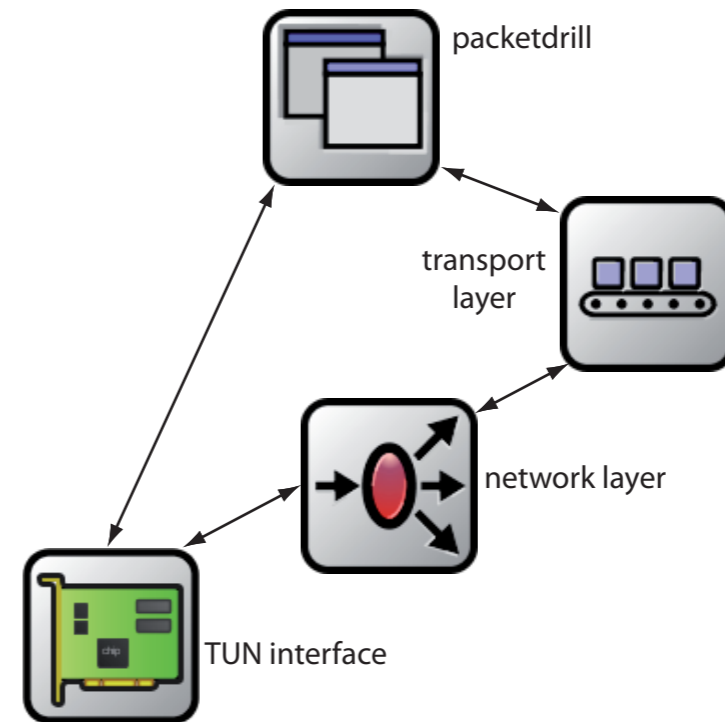
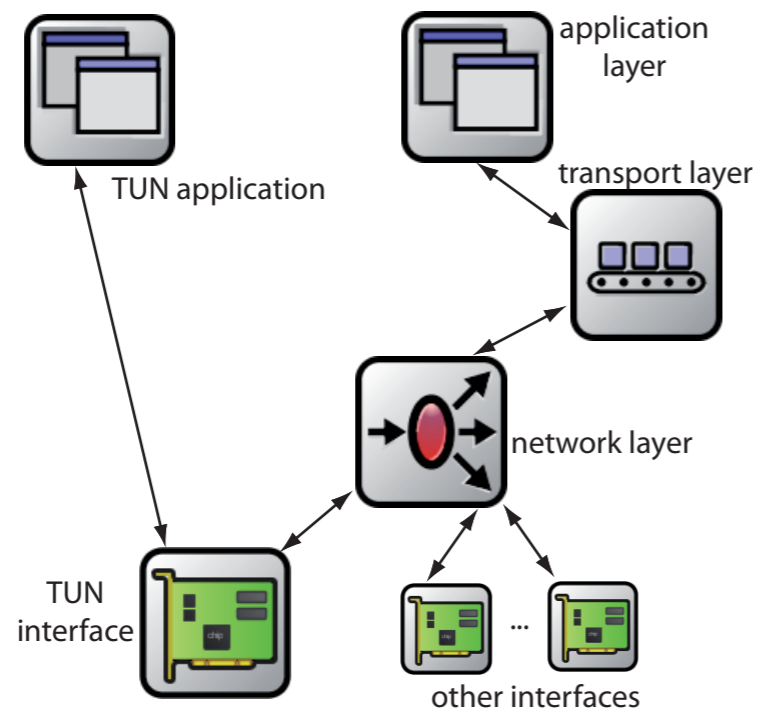
Processing of Scripts

- Outgoing packets are verified
- Incoming packets are injected
- Systemcalls are executed and parameters are verified
- Timing is verified
- In case of an error, processing stops

Packetdrill Port to INET

- It should be possible to share test scripts between Google's packetdrill and the INET port
- Reuse code where appropriate
- Integrate into the C++ environment of OMNeT++/INET where appropriate
- A packetdrill test is similar to a module test and consists of a .pkt file, a .ned file, a routing file and a .test file

TUN Interfaces



Limitations of the INET Port

- Remote mode unsupported
- Python snippets, shell code, command line arguments unsupported
- Blocking system calls unsupported
- getsockopt unsupported
- IPv6 unsupported

Status

- TUN interface integrated in the official INET repository
- packetdrill with UDP support under review by Rudolf
- TCP support ready for submission based on Rudolf's feedback
- SCTP support added to Google's version and the INET port. Will be provided after the TCP support is integrated

Conclusion and Outlook

- packetdrill is a very useful tool for testing kernel transport protocol stacks
- The port to INET allows testing the transport protocols in the INET framework sharing test scripts with Google's packetdrill
- Support for multihoming needs to be added (for SCTP and MPTCP)
- Implementation of test suites for TCP and SCTP

Configuration file

```
%description:  
SUT is server. It accepts a connection. The peer closes the connection.  
%#-----  
%infile: omnetpp.ini  
[General]  
network = PacketDrillTcp  
debug-on-errors = true  
ned-path = ./../src;../lib  
**.scriptFile="./../lib/openPassive.pkt"  
**.pdhost.numTcpTunApps = 1  
**.hasTun = true  
**.startTime = 2s  
**.pdhost.routingFile = "../lib/pdhost.mrt"  
**.pdhost.numPcapRecorders=1  
**.pdhost.pcapRecorder[0].pcapFile = "openPassive.pcap"  
**.pdhost.pcapRecorder[0].moduleNamePatterns="tun"  
**.pdhost.pcapRecorder[0].sendingSignalNames="packetSentToUpper"  
**.pdhost.pcapRecorder[0].receivingSignalNames = "packetReceivedFromUpper"  
**.pdapp.dataTransferMode = "bytecount"  
**.tcp.mss = 1460  
**.tcp.sackSupport = true  
**.tcp.windowScalingSupport = true  
**.tcp.windowScalingFactor = 6  
**.tcp.advertisedWindow = 29200  
**.tcp.useDataNotification = true  
%#-----  
%not-contains: test.out  
Packetdrill error:  
%#-----
```