# TrustNode:
# Beginners Tutorial

**March 19, 2018**

**Marian Ulbricht**

InnoRoute GmbH,
Marsstrasse 14a
80335 Munich, Germany

Contents

This document gives an introduction into the first steps with the TrustNode. All TrustNode related commands needed, are described in the Document *TrustNodeUserManual.pdf* which can be found here. The following scenarios will be explained step by step:

- Connecting devices, using the hardware accelerated FlowCache

For all Tutorials four additional devices with the following capabilities will be needed:

- 1G Ethernet port[1]
- full administration rights
- a Debian based operating system is recommended

Figure 1 gives an overview of the tutorial setup. Three PCs are connected vie Ethernet cables to the physical ports 0 to 2 on the front side of the TrustNode. An additional PC is used for the terminal connection, via microUSB cable to the backside of the TrustNode.

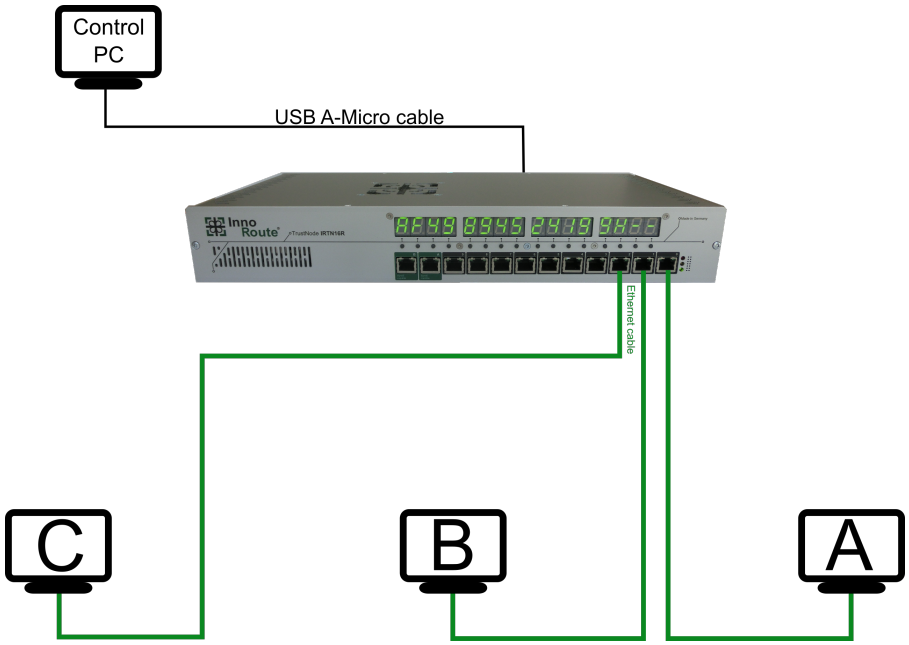


Figure 1: Tutorial setup

## 1.1 Setting up the Terminal connection

To have access to the TrustNode root console, start your favourite terminal application[2] on the ControlPC. Select the USB serial connection which is installed if you plug[3] in the USB-cable into the

[1] other speed classes are not supported
[2] e.g. putty: `http://www.putty.org/`
[3] Plug-in careful and use a strain-relief for free hanging cables.

TrustNode. (normally /dev/ttyUSB0) Apply the connection parameters 115200 Baud, 8 data-bits, no parity-bit, 1 stop-bit. After pressing ↵, the prompt *Trustnode login* should be visible. Login with the username *root* and the password *innoroot*. The welcome screen should be visible, as shown in Listing 5.

```
 _____                    _____              _
|_   _|                  |  __ \            | |
  | |  _ __  _ __   ___ | |__) |___  _   _| |_ ___
  | | | '_ \| '_ \ / _ \|  _  // _ \| | | | __/ _ \
 _| |_| | | | | | | (_) | | \ \ (_) | |_| | ||  __/
|_____|_| |_|_| |_|\___/|_|  \_\___/ \__,_|\__\___|
www.trustno.de
Trustnode v0.9.6
```

Listing 1: Welcome screen

## 1.2 Client preparation

The 3 client PCs A,B,C should be configured with static IPv4 addresses, also the MAC address of each PC should be noted. The MAC addresses will be named as *<MAC(A/B/C)>* in the following descriptions.

```
sudo service network-manager stop #stop the network manager we using the terminal
sudo ifconfig eth0 192.168.0.1 #asuming eth0 is your 1G Ethernet interface
ifconfig eth0| grep eth0 #note the hardware address
```

Listing 2: preparation of PC A

```
sudo service network-manager stop #stop the network manager we using the terminal
sudo ifconfig eth0 192.168.0.2 #asuming eth0 is your 1G Ethernet interface
ifconfig eth0| grep eth0 #note the hardware address
```

Listing 3: preparation of PC B

```
sudo service network-manager stop #stop the network manager we using the terminal
sudo ifconfig eth0 192.168.0.3 #asuming eth0 is your 1G Ethernet interface
ifconfig eth0| grep eth0 #note the hardware address
```

Listing 4: preparation of PC C

## Using the FlowCache

This section shows the configuration of the FlowCache for forwarding packets between the client PCs.

### 2.1 TrustNode configuration

```
TNchangemod 6 #load bitstream which includes Flowcache
TNflowtable add -T192.168.0.1 -O0 #send all packets for PC A to port 0
TNflowtable add -T192.168.0.2 -O1 #send all packets for PC B to port 1
TNflowtable add -T192.168.0.3 -O2 #send all packets for PC C to port 2
```

Listing 5: FlowCache configuration on controlPC

### 2.2 Connection test

After the configuration is done, check the connection between the client PCs using the ping command.

```
ping 192.168.0.1 #ping PC A
ping 192.168.0.2 #ping PC B
```

Listing 6: ping on PC C to PC A

## Flowcache and Ethernet switch

The previous Section 2 uses the FlowCache to apply forwarding rules. Regarding to the bitstream structure described in the TrsutNodeUserManual.pdf the FlowCache overrides the from the hardware Ethernet switch assigned action. This tutorial shows the interaction between Ethernet switch and FlowCache. Please launch the commands from Section 2 first. And keep the ping between the clients running.

### 3.1 Deleting FlowCache rules

To delete the rules of the previous section run the following commands:

```
TNflowtable del -T192.168.0.1 -O0 #send all packets for PC A to port 0
TNflowtable del -T192.168.0.2 -O1 #send all packets for PC B to port 1
TNflowtable del -T192.168.0.3 -O2 #send all packets for PC C to port 2
```

Listing 7: FlowCache configuration on controlPC

After removing the flows, packets are not any-more classified from FlowCache. During setting up the previous tutorial the hardware Ethernet switch learned the OSI layer 2 (L2) addresses of all attached PCs and forward the packets also without FlowCache, the ping between the PCs will be still alive. To overrule the Ethernet switch decision for all unclassified flows, a specific rule in the rule table of

the FlowCache can be used. The flow ID 0x155 is applied automatically to all unclassified packets, defining an action for this flow ID in the cation table will redefine the default action for unmatched packets:

```
TNflowtable ActT_add -i0x155 -b1
```

Listing 8: FlowCache configuration on controlPC

The command in Listing 8 will set the bad-flag in the Network-on-Chip (NoC) header of every unmatched packet, which will advise the action unit to drop the packet. A ping between the PCs is now not possible any more, except a new FlowCache rule for packet forwarding is defined.

### Disable hardware acceleration

To forward every packet from the external front connectors directly to the CPU, apply the following commands to the FlowCache:

```
TNflowtable add -I0 -O16
TNflowtable add -I1 -O17
TNflowtable add -I2 -O18
TNflowtable add -I3 -O19
TNflowtable add -I4 -O20
TNflowtable add -I5 -O21
TNflowtable add -I6 -O22
TNflowtable add -I7 -O23
TNflowtable add -I8 -O24
TNflowtable add -I9 -O25
TNflowtable add -I10 -O26
TNflowtable add -I11 -O27
```

Listing 9: FlowCache configuration on controlPC

This section will describe the setup of the automatic Flowcache configuration by Open vSwitch (OVS) rules. Therefore the TrustNode default configuration needs to be changed to achive the following goals:

- setup OVS
- replace the linux-bridge with OVS
- automatically transfer rules form OVS to the FlowCache

This is a new tutorial, please reload the bitstream #7 and reset the FlowCache memory.

## 5.1 Removing the linux bridge

If you are using TrustNode image version 1.0 or above, the OVS is configured as default option, you can jump to Section 5.2.2. Edit the file /etc/config/network and comment the bridge definition at the end of the file:

```
...
config interface 'TN14'
option 'ifname' 'TN14'
option 'proto' 'none'
option 'auto'   '1'

config interface 'TN15'
option 'ifname' 'TN15'
option 'proto' 'none'
option 'auto'   '1'

#comment this!
#config interface 'TN'
#option 'type' 'bridge'
#option 'proto' 'none'
#option 'ifname' 'TN0 TN1 TN2 TN3 TN4 TN5 TN6 TN7 TN8 TN9 TN10 TN11 TN12 TN13 TN14'
#option 'auto'   '1'
```

Listing 10: Network configuration of TrustNode from controlPC

To commit the changes, launch /etc/init.d/network reload or reboot.

## 5.2 Setup OpenvSwitch

The OVS is preconfigured, the flowcache Interaction works for the OVS-bride TNbr which is already created.[4] To add the TrustNode ports to the OVS-br type the following commands:

[4]To change the name of the bridge edit the file /usr/share/InnoRoute/TNflowdump.conf and restart /etc/init.d/TN_flowdump start|stop

```
ovs-vsctl add-port TNbr TN0 -- set Interface TN0 ofport_request=1
ovs-vsctl add-port TNbr TN1 -- set Interface TN1 ofport_request=2
ovs-vsctl add-port TNbr TN2 -- set Interface TN2 ofport_request=3
ovs-vsctl add-port TNbr TN3 -- set Interface TN3 ofport_request=4
ovs-vsctl add-port TNbr TN4 -- set Interface TN4 ofport_request=5
ovs-vsctl add-port TNbr TN5 -- set Interface TN5 ofport_request=6
ovs-vsctl add-port TNbr TN6 -- set Interface TN6 ofport_request=7
ovs-vsctl add-port TNbr TN7 -- set Interface TN7 ofport_request=8
ovs-vsctl add-port TNbr TN8 -- set Interface TN8 ofport_request=9
ovs-vsctl add-port TNbr TN9 -- set Interface TN9 ofport_request=10
ovs-vsctl add-port TNbr TN10 -- set Interface TN10 ofport_request=11
ovs-vsctl add-port TNbr TN11 -- set Interface TN11 ofport_request=12
ovs-vsctl add-port TNbr TN12 -- set Interface TN12 ofport_request=13
ovs-vsctl add-port TNbr TN13 -- set Interface TN13 ofport_request=14
ovs-vsctl add-port TNbr TN14 -- set Interface TN14 ofport_request=15
ovs-vsctl add-port TNbr TN15 -- set Interface TN15 ofport_request=16
```

Listing 11: Network-OVS configuration of TrustNode from controlPC

#### 5.2.1 Conection test

According to the previous tutorials we are going to test the connection now using the ping command on PC #C.

```
ping 192.168.0.1 #ping PC A
ping 192.168.0.2 #ping PC B
```

Listing 12: ping on PC C to PC A

The expected behaviour is a working ping, because the hardware Ethernet-switch still handles the packets.

#### 5.2.2 Disabling hardware switch

To forward all traffic to the OVS, the rules from Section 4 have to be applied to the FlowCache:

```
TNflowtable add -I0 -O16
TNflowtable add -I1 -O17
TNflowtable add -I2 -O18
```

Listing 13: FlowCache configuration on controlPC

The command ovs-ofctl dump-flows TNbr should now show an increasing packet count for the OVS-fallbackflow. After forwarding all traffic to the OVS, the behaviour can be tested by disabling the OVS-fail-save-mode.

```
ovs-vsctl set-fail-mode TNbr secure
```

Listing 14: Network-OVS configuration of TrustNode from controlPC

In the secure mode, the OVS-br don't handle packets matching to no rule, if no controller is connected. The ping between the connected PCs should not possible with this configuration. To enable the failsave-mode again after the tutorial use ovs-vsctl set-fail-mode TNbr standalone.

### 5.3 Adding OVS flows

The script /usr/share/IoonRoute/scriptsTNdumpflow.py which is automatically launched at start-up listens to changes of the flow-table configuration of TNbr and applies changes automatically to the FlowCache. To check the behaviour printout the actual FlowCache configuration with TNflowtable print -c10:

```
Print MasterTable from 0 to 9.
ID:0  VALID_BIT:0x0
ID:1  VALID_BIT:0x1  INPORT:0x0 MASK_INPORT:0x0 OutPort_enable:0x1 OutPort:0x10
prio:1 EMA_RT:3 EMA_HT:3 ActT:1
ID:2  VALID_BIT:0x1  INPORT:0x1 MASK_INPORT:0x0 OutPort_enable:0x1 OutPort:0x11
prio:1 EMA_RT:4 EMA_HT:4 ActT:2
ID:3  VALID_BIT:0x1  INPORT:0x2 MASK_INPORT:0x0 OutPort_enable:0x1 OutPort:0x12
prio:1 EMA_RT:5 EMA_HT:5 ActT:3
ID:4  VALID_BIT:0x0
ID:5  VALID_BIT:0x0
ID:6  VALID_BIT:0x0
ID:7  VALID_BIT:0x0
ID:8  VALID_BIT:0x0
ID:9  VALID_BIT:0x0
```

Listing 15: FlowCache configuration in idle mode from controlPC

According to Listing 15 the FlowCache should contain the tree flows added in Section 5.2.1. Now launch the following commands for adding flows to the OVS-br which is similar to connecting a OpenFlow-controller:

```
ovs-ofctl add-flow TNbr in_port=2,hard_timeout=30,actions=output:3
ovs-ofctl add-flow TNbr in_port=3,hard_timeout=30,actions=output:2
```

Listing 16: OVS adding flows from controlPC

Now for the next 30 seconds the FlowCache should contain 2 additional flows with the priority 5 which are automatically created from the OVS flowtables. Note: OVS port counting begins from 1, TrustNode port counting begins from 0!

```
Print MasterTable from 0 to 9.
ID:0  VALID_BIT:0x0
ID:1  VALID_BIT:0x1  INPORT:0x0 MASK_INPORT:0x0 OutPort_enable:0x1 OutPort:0x10
prio:1 EMA_RT:3 EMA_HT:3 ActT:1
ID:2  VALID_BIT:0x1  INPORT:0x1 MASK_INPORT:0x0 OutPort_enable:0x1 OutPort:0x11
prio:1 EMA_RT:4 EMA_HT:4 ActT:2
ID:3  VALID_BIT:0x1  INPORT:0x2 MASK_INPORT:0x0 OutPort_enable:0x1 OutPort:0x12
```

```
prio:1 EMA_RT:5 EMA_HT:5 ActT:3
ID:4  VALID_BIT:0x1  INPORT:0x1 MASK_INPORT:0x0 OutPort_enable:0x1 OutPort:0x2
Cut_enable:0x1 CutValue:0x1 prio:5 EMA_RT:1 EMA_HT:1 ActT:4
ID:5  VALID_BIT:0x1  INPORT:0x2 MASK_INPORT:0x0 OutPort_enable:0x1 OutPort:0x1
Cut_enable:0x1 CutValue:0x1 prio:5 EMA_RT:2 EMA_HT:2 ActT:5
ID:6  VALID_BIT:0x0
ID:7  VALID_BIT:0x0
ID:8  VALID_BIT:0x0
ID:9  VALID_BIT:0x0
```

Listing 17: FlowCache configuration after adding OVS-flows from controlPC

### 5.4 Adding a controller

To add a OpenFlow controller to the TrustNode run the following command:

```
ovs-vsctl set-controller TNbr tcp:<IP_ADDR>
```

Listing 18: OVS adding flows from controlPC

### 5.5 Conclusion

With this setup the TrustNode can be configured via OpenFlow. Flows can be applied using the commandline interface or an OpenFlow controller. The OVS-flows are automatically transferred and update to the FlowCache hardware tables.

**L2** OSI layer 2

**NoC** Network-on-Chip

**OVS** Open vSwitch

**OSI** Open Systems Interconnection