HOWTO - Packet capturing (Tracing) in Ethereal/Wireshark Format

29.08.2007, Dieter Müller, Presales Consultant

With release 7.5 funkwerk devices support exporting trace information to a ethereal/wireshark readable format or directly to the ethereal program. This way a very detailed troubleshooting and packet analysis is possible also on links, which are difficult to trace with normal methods, e.g. a directly connected DSL line.

Requirements
The tracing in Ethereal / Wireshark format is implemented in all funkwerk R-Series (e.g. R232b / R3000), TR-Series (e.g. TR200) and W-Series (e.g. W1002) starting with software release 7.5. On the client side you can use either Windows or Linux platform for starting the trace.

Windows platform:
For tracing with Windows hosts you have to install the Brickware software package with minimum version 7.5

Linux platform:
For tracing with Linux hosts you have to download “bricktrace-linux” binary from download website or FTP-server

1.) Installation

1a.) Windows platform

Download and install the latest Brickware Tools from
http://www.funkwerk-ec.com/dl_bintec_brickware_en.html

You just have to install the DIME-Tools packets for the tracing.

Install Ethereal/Wireshark from www.ethereal.com or www.wireshark.org.

1b.) Linux platform

Download the binary “bricktrace-linux” from
ftp.funkwerk-ec.com
or
http://www.funkwerk-ec.com/dl_bintec_unix_tools_de.html

Install Ethereal/Wireshark for your Linux version from www.ethereal.com or www.wireshark.org,
or use the version provided within your linux distribution.

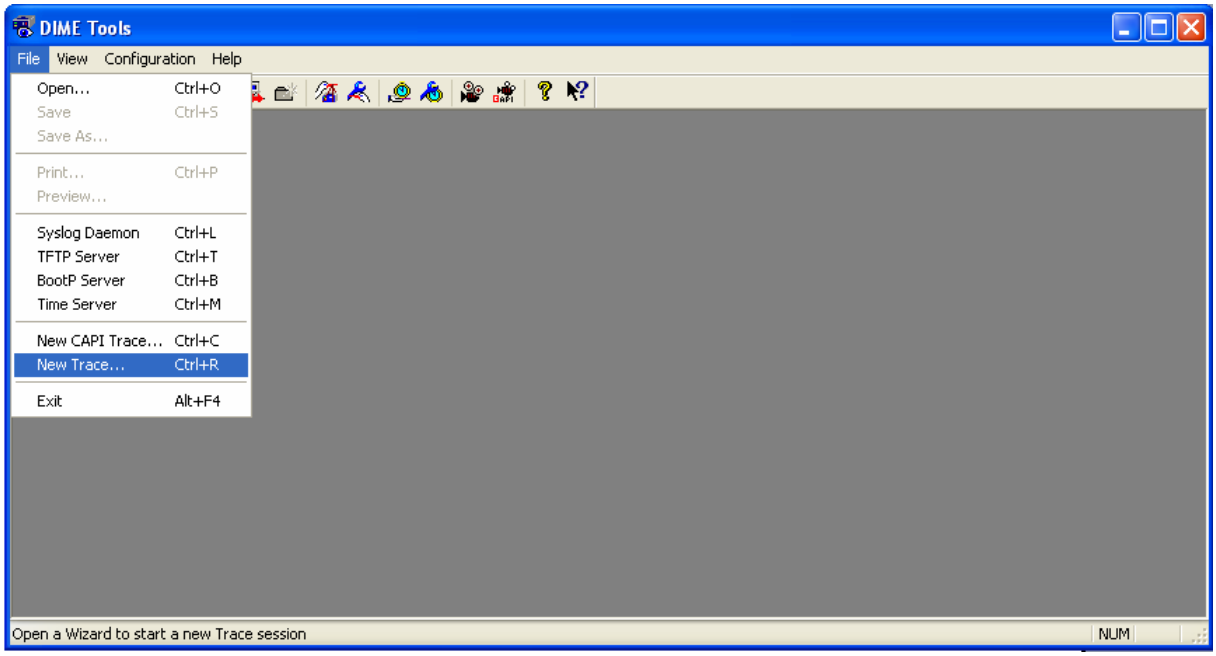
If necessary, update your funkwerk device with a software version 7.5 or higher.

2.) Capturing

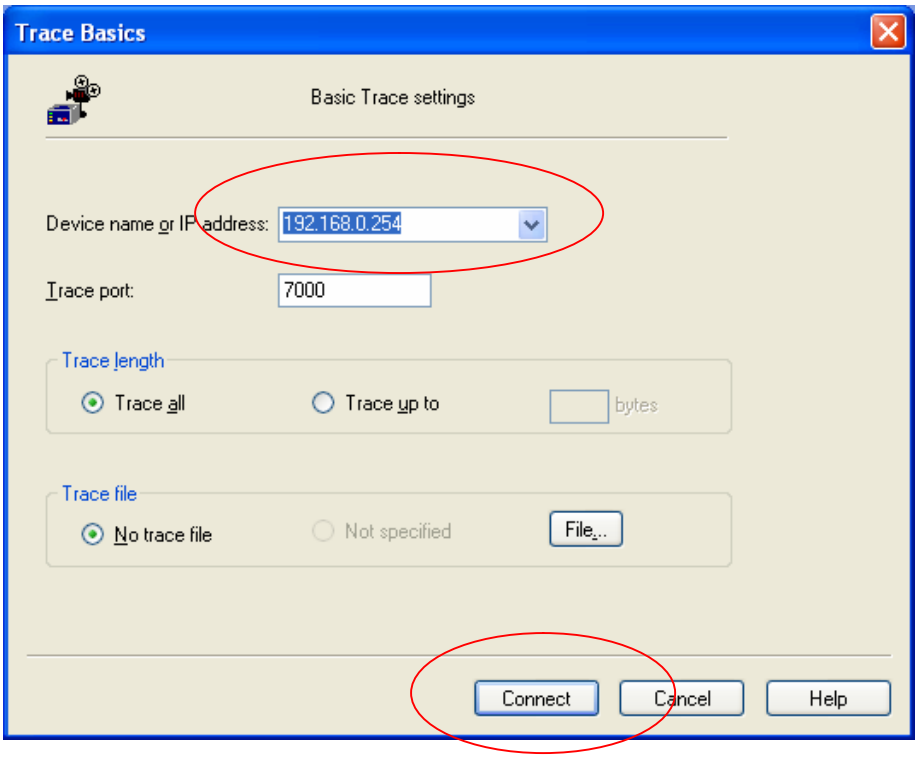
Take care that you have a IP connectivity from your host to the funkwerk device, e.g. you can do a ping from your host to the funkwerk device over a LAN / WAN or VPN link.

2a.) Windows platform

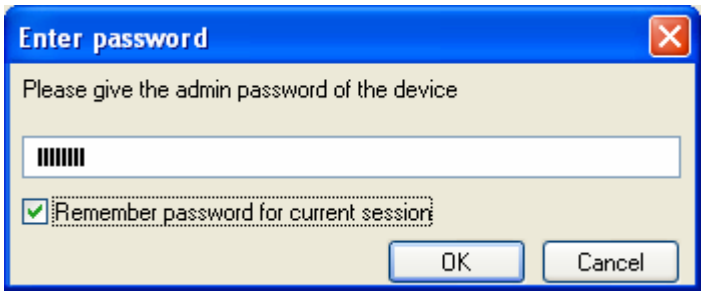
- Start the DIME Tools
- Start “New Trace …”



- Enter the IP address of the device and “Connect”

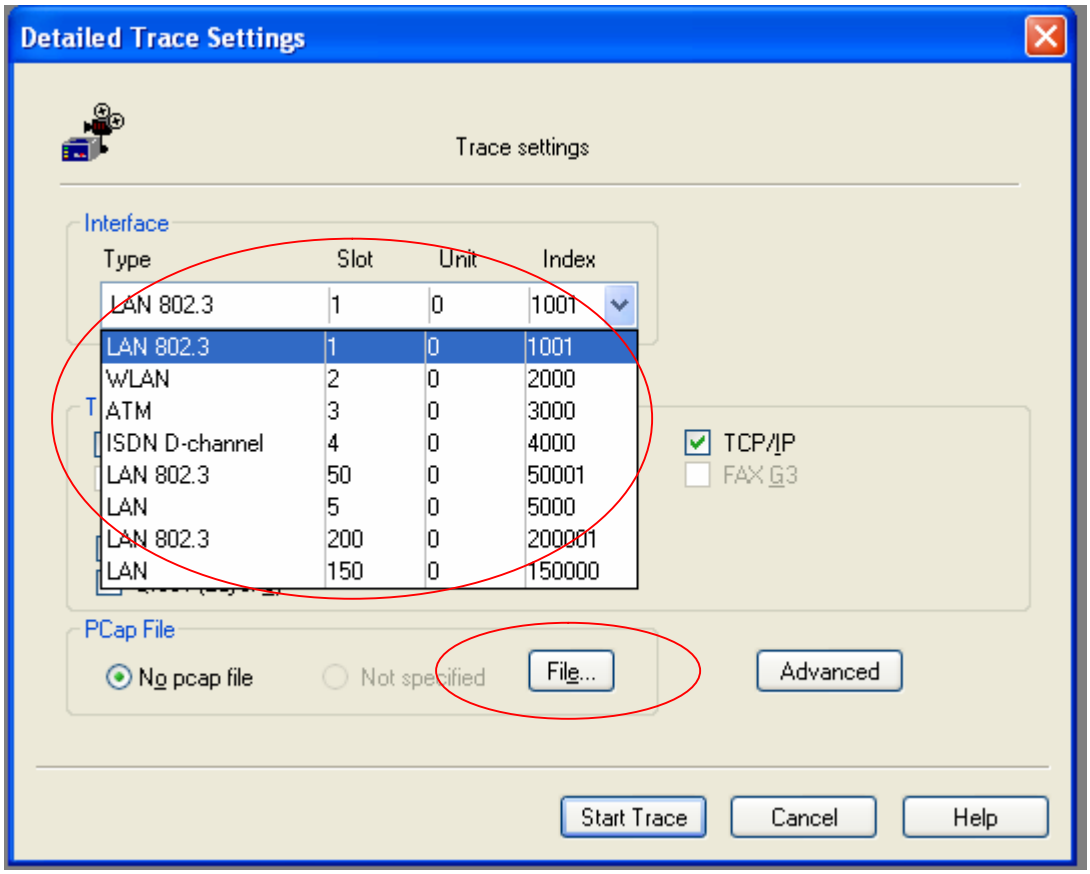


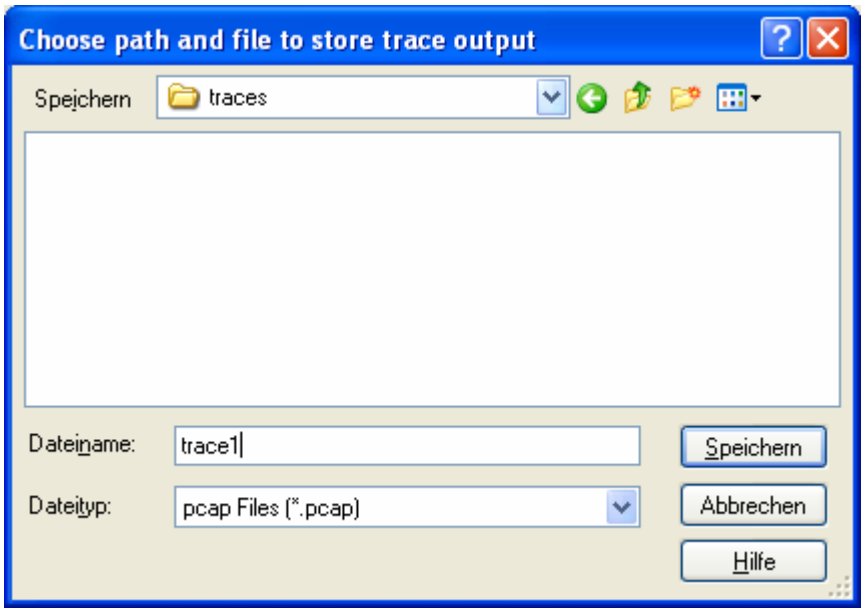
- Enter the admin password of the device (default: funkwerk or bintec (only R3400/R3800)



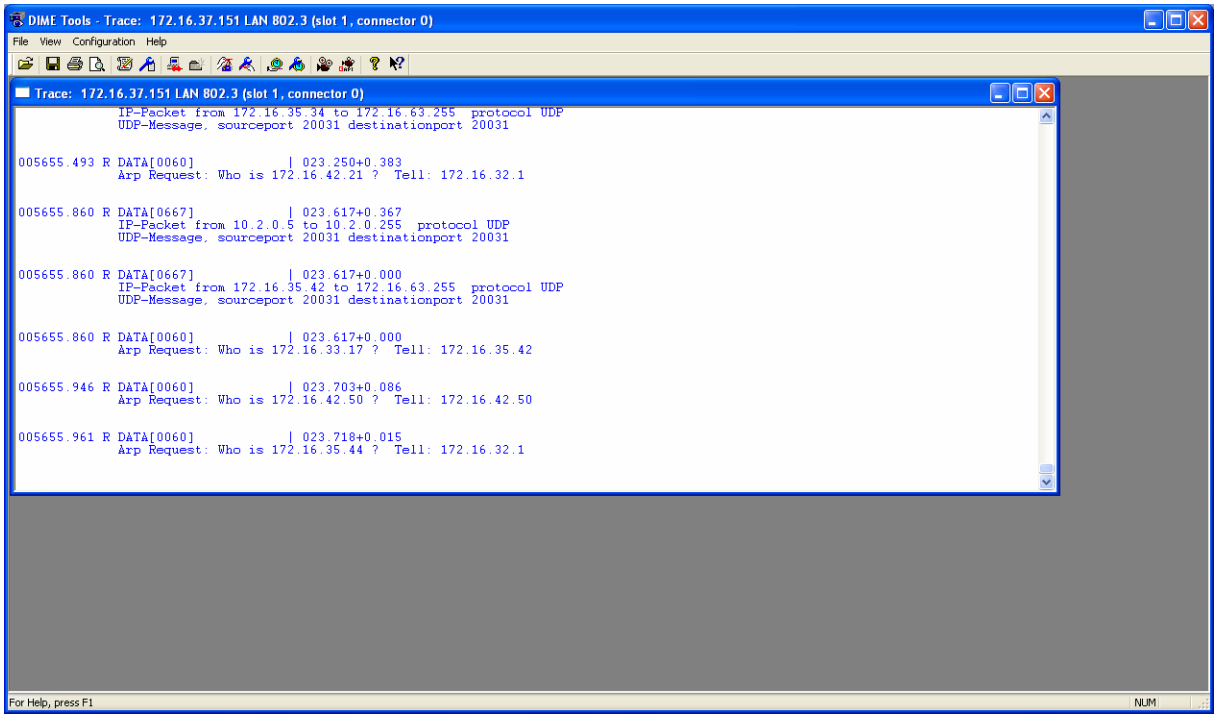
- Select the trace settings

  - Choose an interface (e.g. LAN Port 1001) or Ethernet-over-ATM (50001)
  - If you want to trace a isdn channel select the respective B- or D-channel
  - Select a Pcap File and filename

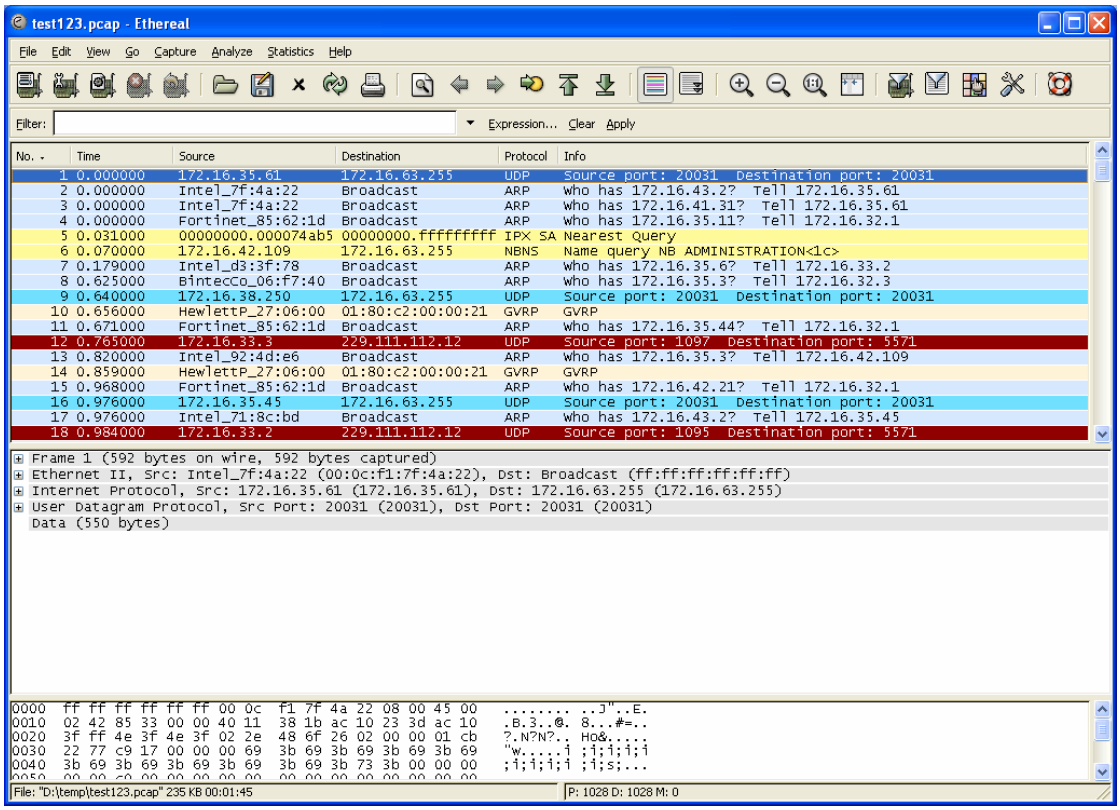




- The trace is started and captures all packets until trace is stopped
- For stopping the trace close the trace window or stop the DIME Tools



- Open the stored pcap-File with Ethereal / Wireshark.



- Ethereal has powerful filtering capabilites. For using them see http://www.ethereal.com/docs/ or http://www.wireshark.org/docs

2b.) Linux platform

The use of the linux version has two advantages in comparison to the windows version:

Realtime Trace
The output of the bricktrace-linux program can be directly send to ethereal. This means you can see the traced packets in realtime. With the windows version you first have to finish the trace and open the pcap trace files afterwards.

Prefilter possibility
The output of the bricktrace-linux can be filtered directly from the program itself. This is e.g. an advantage if the trace-session to the funkwerk device is running over a slow link, but a faster link should be traced (e.g. tracing a DSL connection over a ISDN management link).

You can see the usage of the bricktrace-linux program with all its options with "`bricktrace-linux -?`".

```
user@linux:~/bricktrace-linux> bricktrace-linux -?

Bintec/Funkwerk remote interface tracer ($Revision: 2.43 $)

Usage:
  bricktrace-linux [opts] <routerip> [<channel> <unit> <slot> or <ifindex>]
        -h      hexadecimal output (-! for full length)
        -2      layer 2 output
        -3      layer 3 output
        -a      asynchronous HDLC (B-Channel only)
        -e      ETS300075 (EuroFileTransfer) output (B-channel only)
        -F      FAX (B-Channel only)
        -A      FAX + AT Commands (B-Channel only)
        -D      delta time
        -p      PPP (B-Channel only)
        -f      Frame Relay (B-Channel only)
        -i      IP output
        -N      Novell(c) IPX output
        -t      ascii text output (B-Channel only)
        -x      raw dump mode
        -X      asynchronous PPP over X.75
        -T <tei>        set tei filter (D-Channel only)
        -c <cref>       set callref filter (D-Channel only)
        -r <cnt>        capture only cnt bytes per paket
        -v      increase debug verbose level
        -V 1..3 trace protocol version (default: 3)
        -P<port>        specify trace tcp port (default: 7000)
        -I ipsrc:ipdst:proto:srcport:dstport      IPsession filter
        -B ip1:ip2:proto:port1:port2     bidirect IPsession filter
        -o      OR for LAN filter
        --src=<addr>    LAN filter for source MAC address
        --dst=<addr>    LAN filter for destination MAC address
        --llc           LAN filter for LLC packets
        --help          extended help (environ vars & filter)
        --vpi=<vci>     VPI for ADSL connections
        --vci=<vpi>     VCI for ADSL connections
        --ethereal      start ethereal (implies --pcap-pipe)
        --pcap-pipe     write data in pcap-format into named pipe
        --pcap-file     write data in pcap-format into file
        --ofile=<fname> out filename (pipe/file)
        --pwd=<passwd>  remote admin-password
```

```
        <routerip>      trace host (router's name or IP-address)
        <channel>       0 = D-Channel or no ISDN, 1..31 = Bx-Channel
        <unit>          0..15
        <slot>          0..9
        <ifindex>       interface index (instead of chan/unit/slot)
        if no chan/unit/slot or ifindex given: list all interfaces


Examples:
    bricktrace-linux router                    : list all interfaces
    bricktrace-linux router 0 1 2              : D-Channel(0) of ISDN Slot 2,
Unit 1
    bricktrace-linux router 1000               : LAN Interface 1000 (Slot 1)
    bricktrace-linux router 100001             : virtual IPsec interface 100001
    bricktrace-linux --ethereal router 1000    : write PCAP & start ethereal
    bricktrace-linux --pcap-file router 1000   : write PCAP file

user@linux:~/bricktrace-linux>
```

For finding out the traceable interfaces of the device, use the command without “ifindex”.

```
user@linux:~> bricktrace-linux --pwd funkwerk 192.168.1.1
bricktrace-linux: connected to 192.168.1.1:7000
Ifc:    1000 Type:   7 (LAN 802.3)
Ifc:    5000 Type:   7 (LAN 802.3)
Ifc:    2000 Type:   4 (WLAN)
Ifc:    3000 Type:   3 (ATM)
Ifc:    4000 Type:   0 (ISDN D-channel)
Ifc:   50000 Type:   7 (LAN 802.3)
Ifc:  200000 Type:   7 (LAN 802.3)
 end
user@linux:~>
```

For resolving the interface index values (Ifc) use the “ifstat” command on the telnet console to the router (not on the linux machine!)

```
r232bw:> ifstat
Index  Descr         Type Mtu  Speed St Ipkts   Ies Opkts   Oes PhyAddr/ChgTime
000000 REFUSE        othr 8192     0 up 0       0   0       0     0 00:00:00
000001 LOCAL         othr 8192     0 up 0       0   0       0     0 00:00:00
000002 IGNORE        othr 8192     0 up 0       0   0       0     0 00:00:00
001000 en1-0         eth  1500  100M up 1962248 0   3015    0   00:a0:f9:09:7d:f8
001001 en1-0-llc     eth  1496  100M up 186     0   0       0   00:a0:f9:09:7d:f8
001002 en1-0-snap    eth  1492  100M up 139     0   0       0   00:a0:f9:09:7d:f8
005000 en5-0         eth  1500  100M up 501     0   484     0   00:a0:f9:09:7d:f8
005001 en5-0-llc     eth  1496  100M up 0       0   0       0   00:a0:f9:09:7d:f8
005002 en5-0-snap    eth  1492  100M up 0       0   0       0   00:a0:f9:09:7d:f8
050000 ethoa50-0     eth  1500   10M dn 0       0   0       0   00:a0:f9:89:7d:f8
050001 ethoa50-0-ll  eth  1496   10M dn 0       0   0       0   00:a0:f9:89:7d:f8
050002 ethoa50-0-sn  eth  1492   10M dn 0       0   0       0   00:a0:f9:89:7d:f8
200000 vss1-0        eth  1500   54M dn 0       0   0       0   00:00:00:00:00:00
200001 vss1-0-llc    eth  1496   54M dn 0       0   0       0   00:00:00:00:00:00
200002 vss1-0-snap   eth  1492   54M dn 0       0   0       0   00:00:00:00:00:00
  total: 15
r232bw:>
```

For tracing a certain interface and directly show the trace in ASCII format on the your console add the interface index (also called <ifindex> or <ifc):

```
user@linux:~> bricktrace-linux --pwd funkwerk 192.168.1.1 1000
bricktrace-linux: connected to 192.168.1.1:1000
Ifc:1000 (Chan:0 Unit:0 Slot:1) Type: 7 (LAN 802.3)

030095.193 R DATA[0060]
      0000: ff ff ff ff ff ff 00 03  47 4d c5 45 08 06 00 01  ........GM.E....
      0010: 08 00 06 04 00 01 00 03  47 4d c5 45 c0 a8 01 64  ........GM.E...d
      0020: 00 00 00 00 00 00                                 ......
             Arp Request: Who is 192.168.1.1 ?  Tell: 192.168.1.100

030095.193 X DATA[0042]
      0000: 00 03 47 4d c5 45 00 a0  f9 09 7d f8 08 06 00 01  ..GM.E....}.....
      0010: 08 00 06 04 00 02 00 a0  f9 09 7d f8 c0 a8 01 01  ..........}.....
      0020: 00 03 47 4d c5 45                                 ..GM.E
             Arp Reply: 192.168.1.1 is 00:a0:f9:09:7d:f8

030095.193 R DATA[0098]
      0000: 00 a0 f9 09 7d f8 00 03  47 4d c5 45 08 00 45 00  ....}...GM.E..E.
      0010: 00 54 09 51 40 00 40 01  ad a2 c0 a8 01 64 c0 a8  .T.Q@.@......d..
      0020: 01 01 08 00 d0 da                                 ......
             IP-Packet from 192.168.1.100 to 192.168.1.1  protocol ICMP
             ICMP-Message , type echo request

030095.193 X DATA[0098]
      0000: 00 03 47 4d c5 45 00 a0  f9 09 7d f8 08 00 45 00  ..GM.E....}...E.
      0010: 00 54 0b 18 40 00 3f 01  ac db c0 a8 01 01 c0 a8  .T..@.?.........
      0020: 01 64 00 00 d8 da                                 .d....
             IP-Packet from 192.168.1.1 to 192.168.1.100  protocol ICMP
             ICMP-Message , type echo reply

user@linux:~>
```

For filtering the trace-output use the options “-I” and “-B”.

The syntax is:

```
      -I ipsrc:ipdst:proto:srcport:dstport     IPsession filter
      -B ip1:ip2:proto:port1:port2             bidirect IPsession filter
```

Example: Tracing only ICMP packets (IP protocol 1):

```
bricktrace-linux --pwd funkwerk -I ::1 192.168.1.1 1000
```

Example: Tracing only telnet packets (TCP (IP protocol 6), Port 23)

```
bricktrace-linux --pwd funkwerk -B ::6:23 192.168.1.1 1000
```

Example: Tracing only packets between two host IP addresses:

```
bricktrace-linux --pwd funkwerk -B 192.168.1.1:192.168.1.100 192.168.1.1 1000
```

Using Ethereal / Wireshark with bricktrace-linux

For sending the trace to a ethereal/wireshark readable file, use the options “—pcap-file” and “-ofile=<filename>”

```
bricktrace-linux --pwd funkwerk --pcap-file --ofile=testtrace.pcap 192.168.1.1
1000
```

Open the file with Ethereal / Wireshark.

For sending the trace in **realtime** to ethereal/wireshark, use the options “—ethereal”. All output is piped to ethereal.