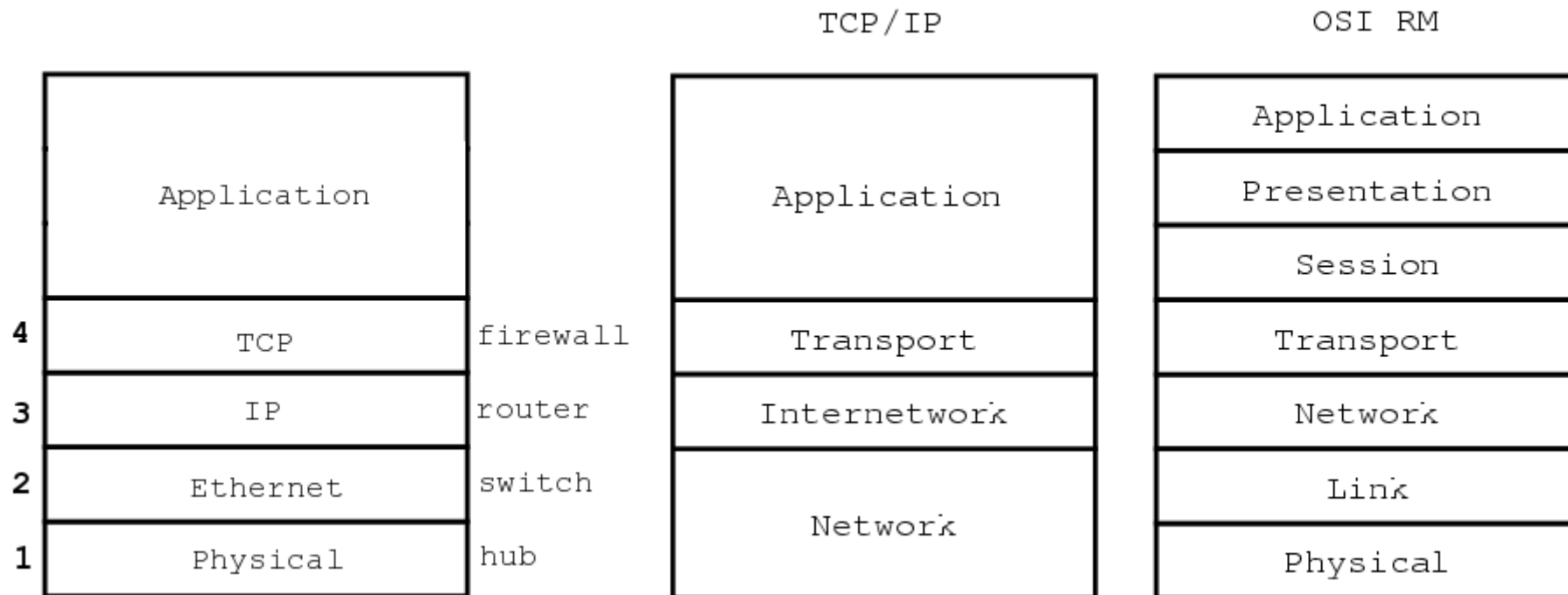


# IP protokol v linuxu - trocha teorie a hodně praxe - příkazy ip, iptables a další

petr.kopecky@linuxbox.cz

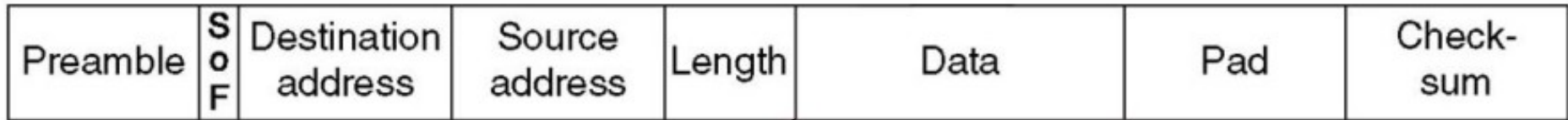
Úvod přednášky bude patřit zopakování vrstev TCP/IP protokolu a vazeb mezi nimi a na tomto základě bude vysvětleno použití základních nástrojů pro konfiguraci sítě, nastavení routování a firewallu. Součástí přednášky budou řešení některých situací z praxe a popis nástrojů či postupů při řešení problémů.

# Vrstvy síťových protokolů

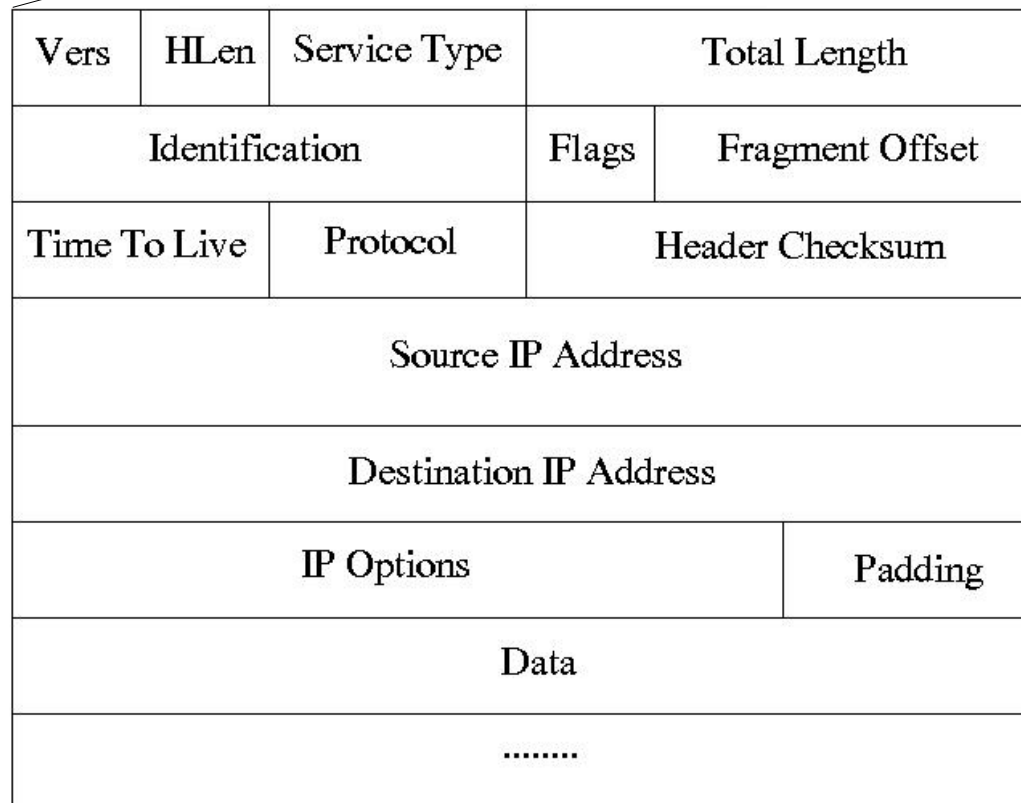


# Anatomie paketu

ethernetový paket (2. vrstva)

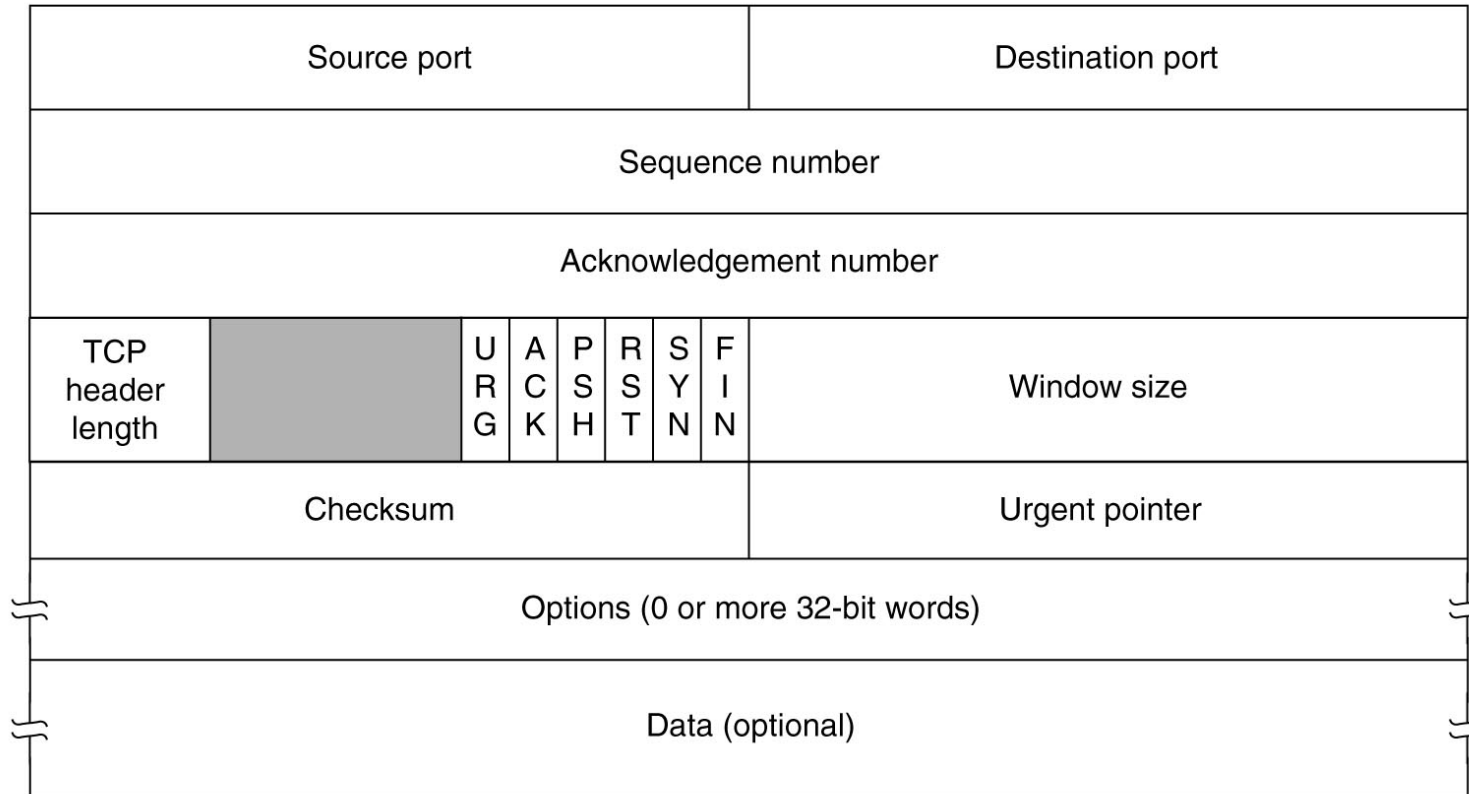


ip paket (3. vrstva)



# Anatomie paketu

tcp protokol (4. vrstva)



## Vazba mezi 2. a 3. vrstvou

- sdílené medium (ethernet) vs. spojení bod – bod
  - sdílené medium vyžaduje adresaci na 2.vrstvě
  - adresami jsou MAC adresy ethernetového rámce
  - unicast/multicast/broadcast se řeší na ip i ethernet vrstvě
  - unicast paket se zdrojovou i cílovou adresou, putující od jedné stanice přímo ke druhé
  - multicast (pouze pro info) pro skupinový příjem
  - broadcast pro všechny příjemce na sdíleném mediu – ( stejný hub, switch, VLAN )
- na 2. vrstvě je příkladem ARP protokol  
na 3. vrstvě je příkladem DHCP protokol
- můstkem mezi 2. a 3. vrstvou je ARP

# Příklad komunikace

```
# ip addr
1: eth0: <BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast qlen 1000
link/ether 00:ff:03:e9:5a:00 brd ff:ff:ff:ff:ff:ff
inet 10.76.66.166/24 brd 10.76.66.255 scope global eth0
2: lo: <LOOPBACK,UP> mtu 16436 qdisc noqueue
link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
inet 127.0.0.1/8 brd 127.255.255.255 scope host lo

# ip route
10.76.66.0/24 dev eth0 proto kernel scope link src 10.76.66.166
169.254.0.0/16 dev eth0 scope link
default via 10.76.66.1 dev eth0

# nc -z -v 10.76.66.10 80
# tethereal -n -i eth0 -f 'host 10.76.66.10'
00:ff:03:e9:5a:00 -> ff:ff:ff:ff:ff:ff ARP Who has 10.76.66.10? Tell 10.76.66.166
00:0f:3d:f4:5b:52 -> 00:ff:03:e9:5a:00 ARP 10.76.66.10 is at 00:0f:3d:f4:5b:52
10.76.66.166 -> 10.76.66.10 TCP 32768 > 80 [SYN] Seq=0 Ack=0 Win=5840 Len=0
10.76.66.10 -> 10.76.66.166 TCP 80 > 32768 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0
10.76.66.166 -> 10.76.66.10 TCP 32768 > 80 [ACK] Seq=1 Ack=1 Win=5840 Len=0
10.76.66.166 -> 10.76.66.10 TCP 32768 > 80 [FIN, ACK] Seq=1 Ack=1 Win=5840 Len=0
10.76.66.10 -> 10.76.66.166 TCP 80 > 32768 [FIN, ACK] Seq=1 Ack=2 Win=5792 Len=0
10.76.66.166 -> 10.76.66.10 TCP 32768 > 80 [ACK] Seq=2 Ack=2 Win=5840 Len=0

# nc -z -v 158.196.149.9 80
# tethereal -n -i eth0 -f 'host 158.196.149.9 or arp'
00:ff:03:e9:5a:00 -> ff:ff:ff:ff:ff:ff ARP Who has 10.76.66.1? Tell 10.76.66.166
00:10:dc:2c:b7:0a -> 00:ff:03:e9:5a:00 ARP 10.76.66.1 is at 00:10:dc:2c:b7:0a
10.76.66.166 -> 158.196.149.9 TCP 32769 > 80 [SYN] Seq=0 Ack=0 Win=5840 Len=0
158.196.149.9 -> 10.76.66.166 TCP 80 > 32769 [RST, ACK] Seq=0 Ack=0 Win=0 Len=0
```

## Rozhraní v linuxu

lo loopback, není routovatelný

eth ethernetová karta

vlan VLAN interface

br bridge (switch)

tun,tap,ppp,sl,... SW rozhraní pro VPN

dummy,imq SW rozhraní

alias/label fiktivní rozhraní pro sekundární IP adresy

# Příkaz ip

plně nahrazuje příkazy ifconfig, route, arp

<http://lartc.org>

ip link	„nahození“ rozhraní, nastavení MAC adresy
ip neigh	manipulace s ARP tabulkou
ip addr	manipulace s IP adresami
ip route	manipulace s routovacími tabulkami
ip rule	manipulace s routovacími pravidly
ip tunnel	nastavení ipip, sit a gre tunelů
ip maddr	multicast
ip mroute	multicast
ip monitor	pro ladění – lépe rtmon

## IP adresy a masky

192.168.1.66/255.255.255.248

192.168.1.66/29

$256 - 248 = 8$

$8 = 2^3$

$n = 3$   $m = (32 - 3) = 29$

192	168	1	66
11000000	10101000	00000001	010000 <b>10</b>
255	255	255	248
11111111	11111111	11111111	11111 <b>000</b>
192	168	1	64
11000000	10101000	00000001	01000 <b>000</b>

ipcalc (pouze pro ověření, nebo do skriptů :)

## Adresy používané v LAN:

10.0.0.0/8            10.0.0.0/255.0.0.0  
172.16.0.0/12        172.16.0.0/255.240.0.0  
192.168.0.0/16    192.168.0.0/255.255.0.0

pro další příklady budeme uvažovat stroj, jenž má 2 rozhraní:

LAN:            eth0    192.168.1.1/24

Internet:    eth1    172.16.1.2/29

default route přes    172.16.1.1

# příkaz iptables

<http://netfilter.org>

pracuje s:

table tabulky filter, nat, mangle

chain seznamy pevné i uživatelské

match filtry

target akce

#jednořádkový firewall:

```
iptables -A INPUT -m state --state NEW,INVALID -j DROP
```

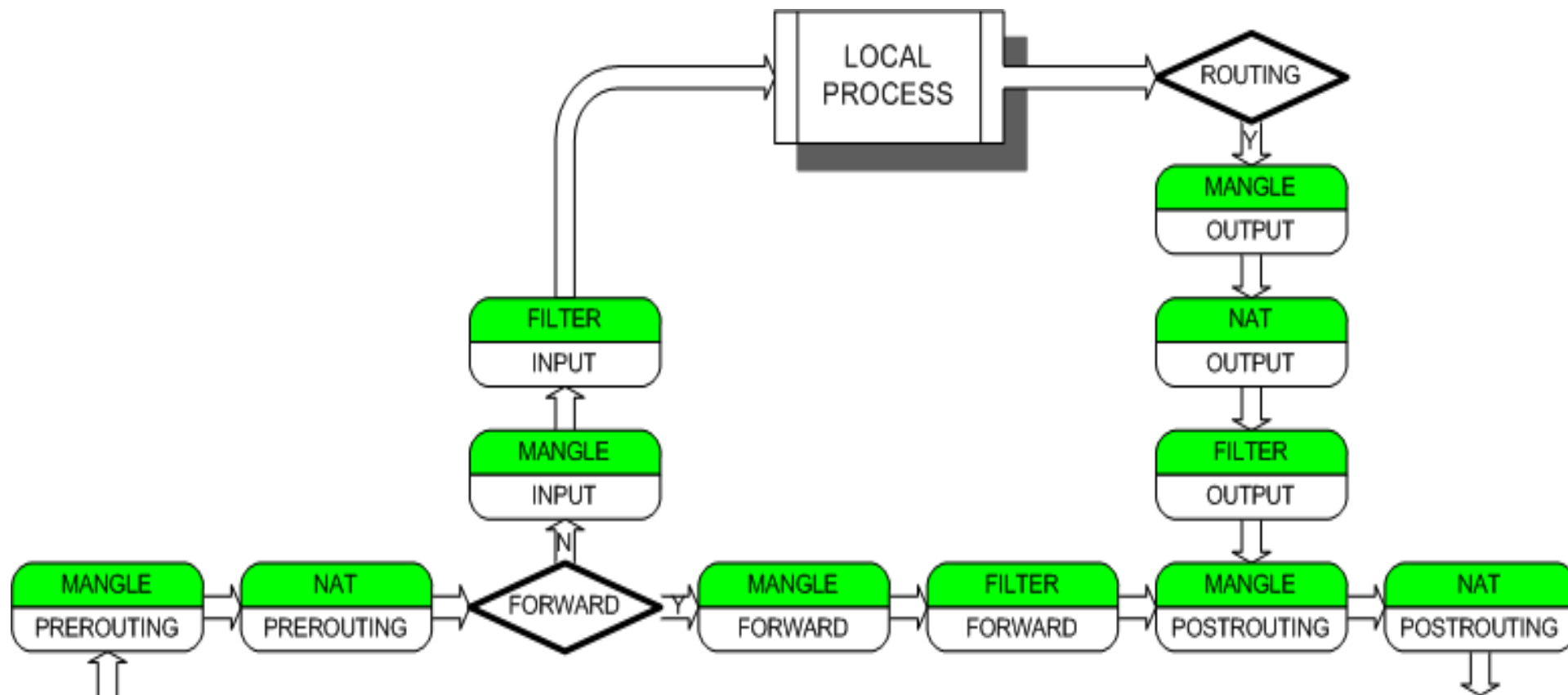
#s podporou NAT:

```
iptables -t nat -A POSTROUTING -o eth1 -s ! 172.16.1.2 \  
-m conntrack --ctstate NEW -j SNAT -to 172.16.1.2
```

```
#iptables-save
*mangle
:PREROUTING ACCEPT [0:0]
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
:POSTROUTING ACCEPT [0:0]
COMMIT
*nat
:PREROUTING ACCEPT [0:0]
:POSTROUTING ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
COMMIT
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A INPUT -m state --state INVALID,NEW -j DROP
COMMIT

#iptables -v -n --line -L POSTROUTING -t nat
```

# Cesta paketu linuxem (z pohledu iptables)



[http://ebtables.sourceforge.net/br\\_fw\\_ia/bridge3b.png](http://ebtables.sourceforge.net/br_fw_ia/bridge3b.png)

[http://ebtables.sourceforge.net/br\\_fw\\_ia/PacketFlow.png](http://ebtables.sourceforge.net/br_fw_ia/PacketFlow.png)

# příklad dvou poskytovatelů internetu

```
ip link set up dev eth0
ip link set up dev eth1
ip link set up dev eth2
```

```
echo 1 >/proc/sys/net/ipv4/ip_forward
```

```
ip addr add 127.0.0.1/8 brd + dev lo
ip addr add 192.168.1.1/24 brd + dev eth0
ip addr add 172.16.1.2/29 brd + dev eth1
ip addr add 172.16.2.2/29 brd + dev eth2
```

```
ip route add default via 172.16.1.1
```

```
#ip route
192.168.1.0/24 dev eth0 proto kernel scope link src 192.168.1.1
172.26.1.0/24 dev eth1 proto kernel scope link src 172.26.1.2
172.26.2.0/24 dev eth2 proto kernel scope link src 172.26.2.2
default via 172.26.1.1 dev eth1
```

```
#/etc/iproute2/rt_tables - umznuje pouzivat pojmenovane tabulky

# musim vytvorit novou tabulku celou - tedy vcetne cesty na default route
ip route add 172.26.2.0/24 dev eth2 proto static src 172.26.2.2 table 111
ip route add default via 172.26.2.1 dev eth2 proto static table 111

# do tabulky se dostanu nap íklad pomoci znacky, kterou pripravim firewallem
ip rule add priority 10001 fwmark 0x1 lookup 111
# ip rule add from 172.26.2.2 lookup 111

#ip rule show # zkontrolujeme
0:      from all lookup local
10001:  from all fwmark 0x1 lookup 111
32766:  from all lookup main
32767:  from all lookup default

# prichozi pozadavek na iface eth2 na http protokol smeruji dovnitr
iptables -t nat -A PREROUTING -i eth2 -p tcp --dport 80 \
-m conntrack --ctstate NEW -j DNAT --to 192.168.1.33

# odchozi paket na stroj 192.168.1.33 premapuji na src ip 192.168.1.1
iptables -t nat -A POSTROUTING -o eth0 -d 192.168.1.33 -s ! 192.168.1.0 \
-m conntrack --ctstate NEW -j SNAT --to 192.168.1.1

# vysledke je komunikace se 4 ip adresami
# 10.0.0.1 -> 172.16.2.2 | FW | 192.168.1.1 -> 192.168.1.33
# origsrc      origdst      replydst      replysrc
```

```
tcp      6 116 TIME_WAIT src=127.0.0.1 dst=127.0.0.1 sport=47494 dport=6000
packets=4 bytes=216 src=127.0.0.1 dst=127.0.0.1 sport=6000 dport=47494 packets=3
bytes=164 [ASSURED] mark=0 use=1
```

```
# odchozi SMTP provoz pujde p es jineho poskytovatele
```

```
iptables -t mangle -A PREROUTING -i eth0 -d ! 192.168.1.0/24 -p tcp --dport 25\
-m conntrack --ctstate NEW -j MARK --set-mark 0x1
```

```
# musim zajistit, ze spojeni navazane p es druheho poskytovatele bude timto
smerem pokracovat
```

```
iptables -t mangle -A PREROUTING -i ! eth2 \
-m conntrack --ctorigdst 172.16.2.2 -j MARK --set-mark 0x1
```

```
iptables -t mangle -A PREROUTING -i ! eth2 \
-m conntrack --ctreplydst 172.16.2.2 -j MARK --set-mark 0x1
```

## **další nástroje**

`/proc/net`

`/proc/sys/net/ipv4`

`Documentation/networking/ip-sysctl.txt`

`/proc/sys/net/ipv4/ip_forward`

`/proc/sys/net/ipv4/conf/eth0/proxy_arp`

## **mii-tool, ethtool**

```
# ethtool -i eth0
driver: natsemi
version: 1.07+LK1.0.17
firmware-version:
bus-info: 0000:00:12.0
```

```
# ethtool eth0
```

```
Settings for eth0:
```

```
Supported ports: [ TP MII FIBRE ]
Supported link modes:   10baseT/Half 10baseT/Full
                        100baseT/Half 100baseT/Full

Supports auto-negotiation: Yes
Advertised link modes:  10baseT/Half 10baseT/Full
                        100baseT/Half 100baseT/Full

Advertised auto-negotiation: Yes
Speed: 10Mb/s
Duplex: Hal
Port: Twisted Pair
PHYAD: 1
Transceiver: internal
Auto-negotiation: on
Supports Wake-on: pumbags
Wake-on: ub
SecureOn password: 00:00:00:00:00:00
Current message level: 0x000040c5 (16581)
Link detected: no
```

## **ping**

```
-n # nezapominat  
-c # po et paketu  
-s # velikost paketu  
-I # zdrojova IP nebo interface
```

## **nc**

```
# netcat  
nc -z -v localhost 22 # test otevreného portu  
nc localhost smtp # komunikace
```

## **nmap**

```
nmap -sP 192.168.1.0/24  
nmap -sT -p 22 -P0 192.168.1.0/24  
nmap -O 192.168.1.0/24
```

## **dig**

```
# práce s DNS záznamy
```

## **netstat**

```
#netstat -ltupn
```

Active Internet connections (only servers)

Proto	Recv-Q	Send-Q	Local Address	Foreign Address	State	PID/Program name
tcp	0	0	0.0.0.0:6000	0.0.0.0:*	LISTEN	2183/X
tcp	0	0	:::6000	:::*	LISTEN	2183/X
tcp	0	0	:::22	:::*	LISTEN	1857/sshd
udp	0	0	0.0.0.0:68	0.0.0.0:*		1679/dhclient

## lsof

lsof -i

COMMAND	PID	USER	FD	TYPE	DEVICE	SIZE	NODE	NAME
dhclient	1679	root	4u	IPv4	4769		UDP	*:bootpc
sshd	1857	root	3u	IPv6	5110		TCP	*:ssh (LISTEN)
X	2183	root	1u	IPv6	6174		TCP	*:x11 (LISTEN)
X	2183	root	3u	IPv4	6175		TCP	*:x11 (LISTEN)

## tcpdump

-v -n

-i # interface (promiscuous mode)

-w # zapis do souboru

-s 0 # delka zaznamenavane casti paketu

tcpdump -v -n -i eth0 not tcp port 22

## tetherreal

textova verze ethereal

vidi mnohem hlouběji do komunikace

## brctl

nastavení bridge

## **vconfig**

nastaveni vlan

## **arping**

ping „na 2. vrstve“

## **ebtables**

„iptables na 2. vrstve“

## **arptables**

manipulace s arp tabulkami, rizeni odpovedi

# net-snmp

```
# /etc/snmp/snmpd.conf
```

```
com2sec local localhost MyROGroup
com2sec mynetwork 192.168.1.0/24 MyRWGroup
```

```
access MyROGroup "" any noauth 0 all none none
access MyRWGroup "" any noauth 0 all all all
```

```
#service snmpd start
```

```
# test dostupnosti
snmpwalk -v 2 -c public localhost systém
```

```
# vypis rozhrani
snmptable -v 2 -c public localhost ifTable
```