4. Síťování

- dříve se používaly protokoly uucp (unix to unix copy)
- existuje řada síťových protokolů
    - IP
    - IPx (pro sdílení souborů apod.)
    - X.25 (telekomunikační)
    - AX.25 (zjednodušená verze x.25, používali radioamatéři)
    - dále stacky BT, ATM, PPP (point-to-point),…

- **MAC adresa**
    - fyzická adresa ethernetového protokolu karty
    - 6 bitové číslo, odděluje se dvojtečkami
    - např. 00:20:4B:5C:6D:7E

Protokol IP

- existují dvě verze – IPv4 a IPv6
- je to stack (protokolová sada), implementuje více vrstev OSI
- ethernet, wifi,…
- vždy se objevují síťová zařízení, která nejsou v /dev, ale samostatně
- existují dva způsoby implementace IP
    - socket (BSD, Linux, Mac)
    - stream (SvRxm, Solaris,…)

- adresy se přidělují ve formě a.b.c.d. (8bit čísla)
    - a,b,c = síťová adresa, přiděluje se pevně k switchi (192.168.0)
    - d = host-adress
    - síť třídy C (má pevně dané tři bajty), méně než 256 počítačů
    - + se to ještě může omezit (např. 192.207 --> 16bitů --> maska /28)
    - dvě adresy nemůže používat žádný stroj
        - samé nuly v hostpart (network adress)
        - samé jedničky v hostpart (broadcast adress)

Příkaz ifconfig

- **ifconfig -a**
    - *vypíše všechna síťová zařízení, o kterých kernel ví (a jejich stav)*

    - určitě tam bude **lo/lo0 (local loopback)**
        - provoz vnitřních spojení v rámci vlastního serveru
        - adresa 127.0.0.1, jméno localhost [NEMĚNIT]

    - **dummy0** - na testing (grounding, drží IP adresu)

    - **eth(x)**
        - zařízení, jsou zandavatelná/vyndavatelná
        - jednoznačně identifikována MAC adresou

        - většinou drátové, ale mohou být i bezdrátové (ty jsou ale většinou **wlan(x)**)

    - **ppp(x), tun(x), gre(x), ipip, sit(x)**,…

- dá se s ním nastavit prakticky vše, co se týká základních vlastností existujícího rozhraní
- parametry nezačínají pomlčkami, jsou tam klíčová slova
- 3 základní věci, které se většinou nastavují - IP adresa, netmask, broadcast adresa

Konfigurace síťových rozhraní

- ***ifconfig (jméno-zařízení) (IP-adresa) netmask (maska-sítě) broadcast (broadcast-adresa)***

- nutno zadat i default gateway (pro přístup k DNS apod.) – nelze přes ifcongig, ale přes route

- **route add default gw (IP-adresa)**
    - všechny packety, které nevedou do strojů v lokální síti, se směřují tam

- pak se musí nastavit ještě **DNS**
    - ne příkazem, nastavení je v souborech
    - /etc/resolv.conf
        - textový soubor
        - obsahuje alespoň 1 řádek (nameserver + IP adresa)
        - většinou i domain název-domény (pro vyhledávání nekvalifikovaných jmen)

- jména lze uložit i mimo oficiální DNS
    - /etc/hosts
    - IP-adresa jména-stroje

5. Pokročilé síťování

Příkaz ip

- prakticky vše, co se týká síťování v linuxu
- rozhraní (link), addr, route (objekty směrování), pravidla, neighbours, tunel, multicast,…
- pro konfiguraci bezdrátu se používá spíše iwconfig či wpa_supplicant (démon)

IP link

- promiskuitnost rozhraní
- přejmenovat rozhraní (ip link set eth0)
- a mnoho dalšího…

Routování

- v linuxu je mnoho routovacích tabulek (většinou 256), ale většinou se používá default
- default má číslo 255

ip rule

- pravidla
- selektor (matchování packetů), co se má stát

- selektor může být FROM, TO, ToS (type of service), pref, firewall mark,…
- z jaké sítě pochází, kam směřuje,…
- action má table identifikátor-tabulky

Tunel

- umožňuje zakládat, spravovat a rušit zařízení, která nemají fyzický konektor, ale tunelují své pakety pomocí zařízení, které je má
- **tunel add type typ-tunelu local local-ip remote remote-ip**
    - typ = gre, ipip, ip6ip,…
    - local = moje adresa, remote = přijímač paketů

Bridgování / switching

- na 2. vrstvě OSI modelu
- příkaz **brctl addbr**
    - lze s ním vytvářet a nastavovat bridge (= bridge control)
    - vytvoří prázdný bridge

- **brctl addif jmeno-bridge jmeno-rozhrani**
    - např. brctl addif br0 eth0, pak znovu pro eth1

- bridge mohou mít nastavenou IP adresu

iptables

- schopnost filtrace paketů, něco jako firewall
- defaultně jsou v jádru 2 – filters a NAT
- nastavení NAT (schováme za jednu IP adresu celou síť)
- práce nad řetězi pravidel (chain)
- pravidla se vyhodnocují chronologicky
- **filter – 3 chainy**
    - INPUT – pro packety pro nás stroj
    - OUTPUT – odchozí packety
    - FORWARD – jen pro packety routované
- **NAT**
    - PREROUTING
    - POSTROUTING
    - MANGLE
- pravidla se skládají z matching-part a target
- např. iptables -A INPUT -s 192.168.5.0./24 -d port80 -j ACCEPT (/DROP, REJECT)

Port-forwarding

- **ssh -l 1080:IP7:80 user@IP2**
    - port 80 z dané sítě se forwardne na můj port 1080
    - ntb bude port 1080 vysílat do sítě
    - jinak funguje jen z toho notebooku
    - http:1080//loclahost
- není třeba root