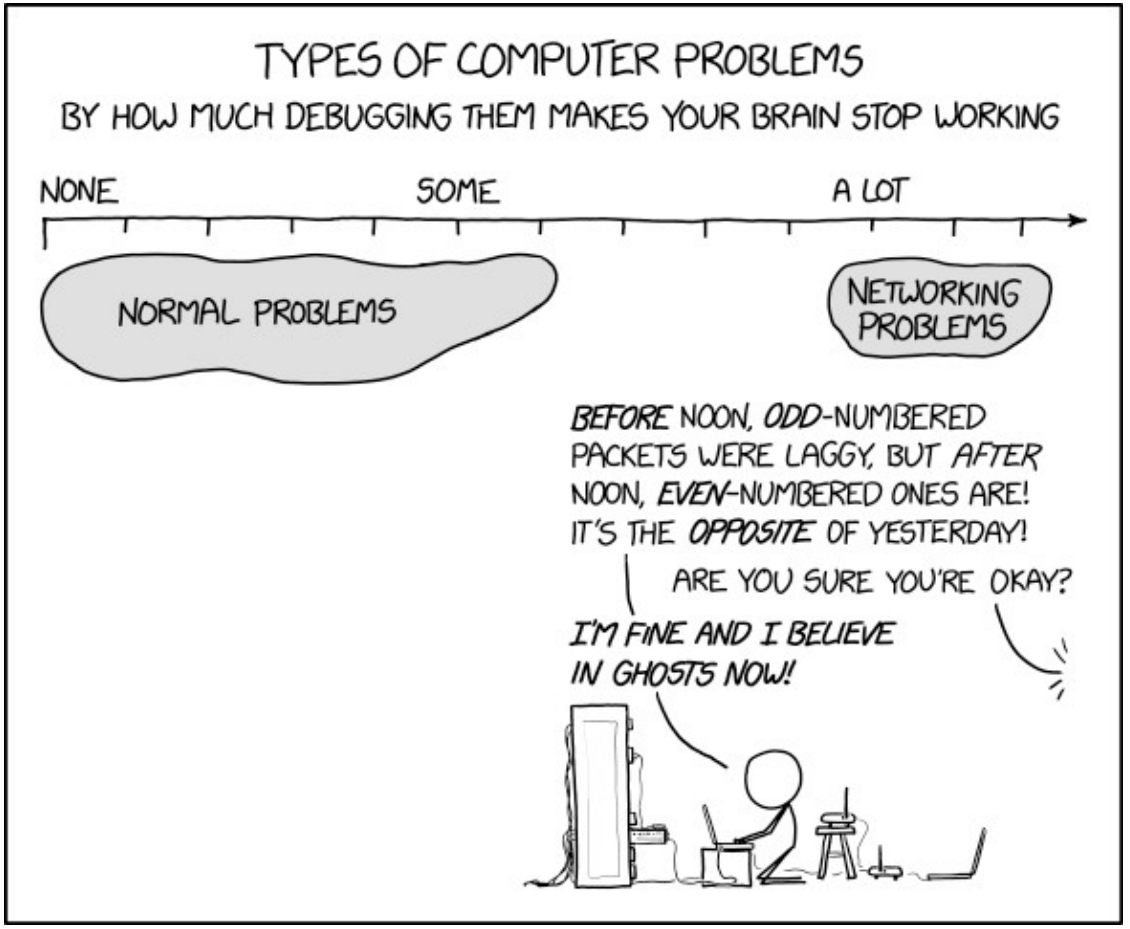
Reititys



https://xkcd.com/2259/

Reititys

- Miten löydetään kone jolla on tietty IP?

  (OSI layer 3)
- Samaan aliverkkoon pääsee suoraan, muuten tarvitaan reititystä
- Oletusreitin asetus (*yleensä* automaattinen):

  ip route add default via *reititin*

  missä *reititin* on välittävän koneen (gateway) osoite

Reititys

- Jos johonkin koneeseen tai verkkoalueeseen ei pääse oletusreitin kautta, sille pitää määritellä oma gateway. Esim.

  ip route add 192.168.127.0/24 via 172.21.208.17

  ohjaa aliverkkoon 192.168.127.0/24 lähtevät paketit lonka7 -koneen kautta.

  ip route add 192.168.127.19 via 172.21.208.17

  vastaavasti mutta vain yksittäiselle koneelle, jonka osoite on 192.168.127.19 (myös maski /32).

Reititys

- Käytössä olevaa reititystaulua voi katsella:

  ip route [show]

  route [-n] # vanha

- Reitityksen poisto (argumentteina samat kuin ip route add -komennossa annettiin tai mitä ip route show näyttää):

  ip route delete ...

Reititys

- Jotta gateway-kone välittäisi paketteja eteenpäin se pitää erikseen sallia:

  echo 1 > /proc/sys/net/ipv4/ip_forward # tai

  sysctl -w net.ipv4.conf.all.forwarding=1

  - Tarkistus: cat /proc/sys/net/ipv4/ip_forward # tai

    sysctl net.ipv4.conf.all.forwarding
  - Asetus pysyväksi: net.ipv4.ip_forward=1 tiedostossa /etc/sysctl.conf
  - Gateway-koneen (softa)palomuurissa tarvitaan myös säännöt tätä varten

Reititys

- Gateway-koneen pitää yleensä olla omassa aliverkossa (tai sinne pitää määritellä reitti ensin, harvinaista).
- Kahden samaa privaattiosoitealuetta käyttävän aliverkon yhdistäminen reitittämällä ei onnistu; jos mahdollista kannattaa käyttää eri osoitealueita, muuten tarvitaan monimutkaisempia palomuurisäätöjä (NAT) tai ARP-proxyä tms.

Reititysesimerkki

- JY:n julkinen verkko: 130.234.0.0/16, labraverkon alue 130.234.208.0/23
- Kurssin sisäverkko: 172.21.0.0/16
  - Sisäverkon rajapalomuuri: 172.21.0.1, 130.234.254.76
  - Palomuurin julkinen verkkoalue: 130.234.208.0/23
  - Palomuurin sisäinen verkkoalue: 172.21.0.0/16
  - Sisäverkon dns/dhcp-palvelin: 172.21.0.4
  - Sisäverkon osa 172.21.208.0/23 NATattu (1-1 NAT) julkiselle alueelle 130.234.208.0/23

Reititysesimerkki

- lonka5: 130.234.208.15, 172.21.208.15, 192.168.125.1
  - Lonka5:n sisäinen (NAT-)verkko: 192.168.125.0/24
- lonka6: 130.234.208.16, 172.21.208.16, 192.168.126.1
  - Lonka6:n sisäinen (NAT-)verkko: 192.168.126.0/24
- lonka7: 130.234.208.17, 172.21.208.17, 192.168.127.1
  - lonka7:n sisäinen (NAT-)verkko: 192.168.127.0/24
- lonka8: 130.234.208.18, 172.21.208.18, 192.168.128.1
  - lonka8:n sisäinen (NAT-)verkko: 192.168.128.0/24

Reititysesimerkki

- Koneet:

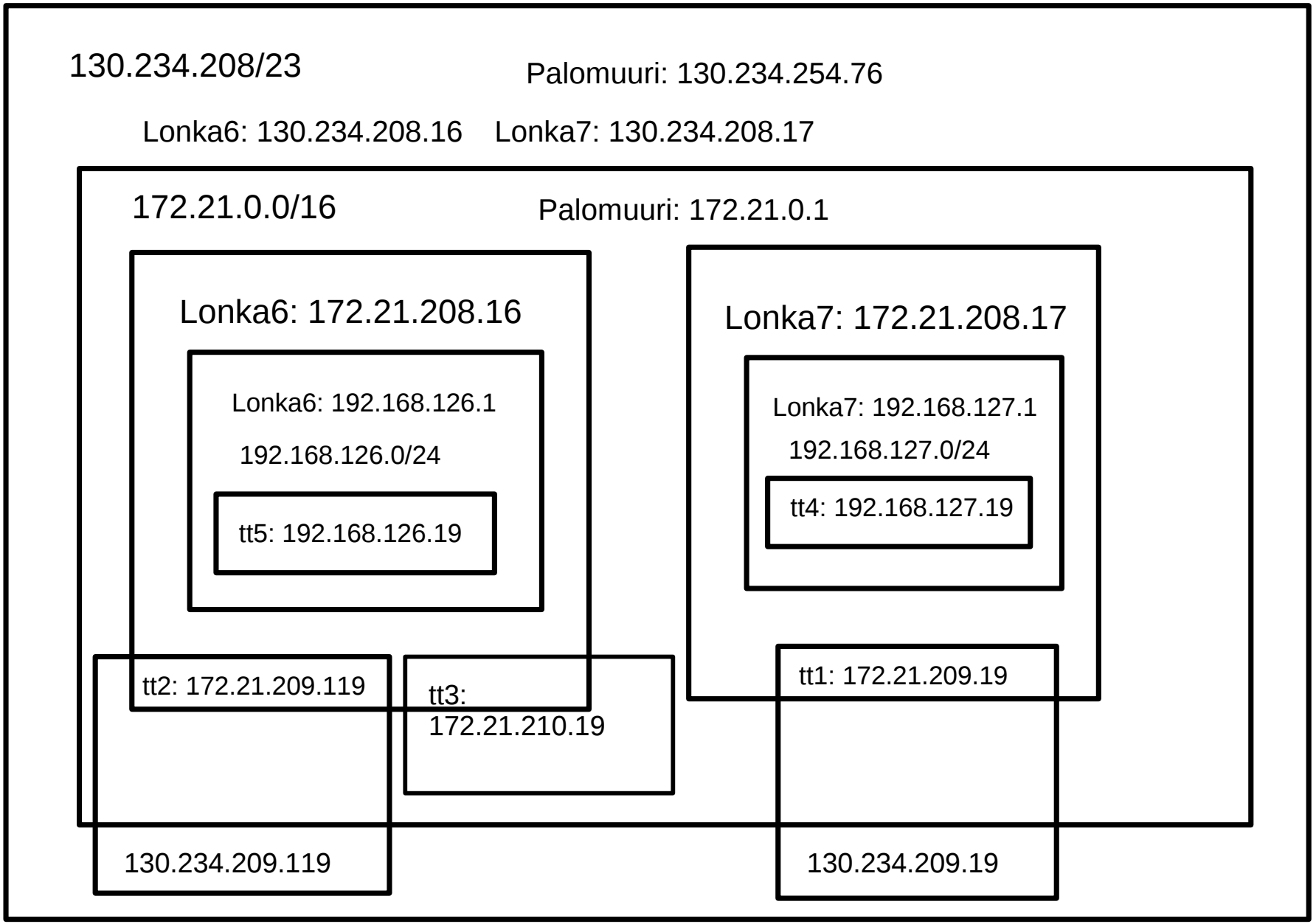
  tt1: 130.234.209.19, 172.21.209.19 (siltaverkko)

  tt2: 130.234.209.119, 172.21.209.119 (siltaverkko)

  tt3: 172.21.210.19 (siltaverkko, ei julkista IP:tä)

  tt4: 192.168.127.19 (NAT-verkko lonka7:n sisällä, ei julkista IP:tä eikä ulomman sisäverkon IP:tä)

  tt5: 192.168.126.19 (NAT-verkko lonka6:n sisällä, ei julkista IP:tä eikä ulomman sisäverkon IP:tä)


130.234.208/23
Palomuuri: 130.234.254.76
Lonka6: 130.234.208.16
Lonka7: 130.234.208.17
172.21.0.0/16
Palomuuri: 172.21.0.1
Lonka6: 172.21.208.16
Lonka6: 192.168.126.1
192.168.126.0/24
tt5: 192.168.126.19
Lonka7: 172.21.208.17
Lonka7: 192.168.127.1
192.168.127.0/24
tt4: 192.168.127.19
tt2: 172.21.209.119
tt3:
172.21.210.19
tt1: 172.21.209.19
130.234.209.119
130.234.209.19

Reititysesimerkki...

- Samassa aliverkossa olevat koneet näkevät toisensa suoraan
- Palomuuri reitittää omien aliverkkojensa ja julkiverkon välillä
- VM-alustakoneet (lonka*) reitittävät aliverkon 172.21 sisällä
- VM-alustakoneet reitittävät myös omien sisäverkkojensa (192.168.x) ja (omasta näkökulmastaan ulkoisen) aliverkon 172.21 välillä

Reititysesimerkki...

- Siltaverkkoa käytettäessä virtuaalikoneen oletusreittinä voi käyttää joko palomuuria (172.21.0.1) tai jotakin alustakoneista (tyypillisesti alustakonetta jossa se itse on).
- Jos VM on NATattuna alustakoneen sisäisessä (192.168.x) -verkossa, oletusreitin pitää olla alustakoneen sisäinen osoite (192.168.x.1)

Reititysesimerkki...

- Alustakoneiden (NAtattuihin) sisäverkkoihin pääsee vain ko. alustakoneen kautta, siis käyttämällä sitä joko oletusreittinä tai erikseen määriteltynä reittinä kyseiselle sisäverkolle (kumpikin edellyttää, että alustakoneen palomuuri sallii tämän, ulkoverkosta tultaessa myös rajapalomuuri(t))

Reititysesimerkki...

- Koneet tt1, tt2 ja tt3 pääsevät toisiinsa suoraan, muualle käyttäen (oletus)reitittimenä joko palomuuria tai alustakoneitaan, koska niissä on käytössä siltaverkko (bridged net)
- Koneissa tt4 ja tt5 on käytössä NAT-verkko, josta ne näkevät suoraan vain oman alustakoneensa sisäisen (NATatun) verkon (192.168.x.0), oletusreittinä toimii vain aina oman alustakoneestaan sisäinen IP (192.168.x.1)

Reititysesimerkki...

- Alustakoneet tarjoavat sisäverkkoonsa myös nimipalvelun, joten tt4:ssä nimipalvelinkin on 192.168.127.1; se voisi käyttää ulkoistakin nimipalvelua (esim. 172.21.0.4), mutta silloin se ei saisi omaa nimeään (eikä muita NAT-verkkonsa koneita) nimipalvelusta

Reititysesimerkki...

- Jotta NAT-sisäverkon ulkopuoliset labraverkon koneet pääsisivät sen sisälle, niissä pitää määritellä reitittimeksi sinne ko. alustakone (tt4:n tapauksessa siis lonka7, ja nimenomaan sen ”ulompi sisäinen” IP 172.21.208.17, tt5 vastaavasti lonka6)
- Labraverkon ulkopuolisille koneille pitäisi vastaavasti määritellä reitittimeksi alustakoneen julkinen IP (130.234.208.17)

Reititysesimerkki...

- Alustakoneen käyttäminen oletusreittinä toimii myös siltaverkon koneiden (tt1, tt2 &c) kanssa – kunhan ko. kone pysyy päällä (potentiaalinen ongelma migraation kanssa)
- Reititys ei vaikuta nimipalveluun, sisäverkkojen koneet pitää lisätä lokaaliin /etc/hosts -tiedostoon ellei niitä ole "globaalissa" nimipalvelussa (labraverkon palvelimessa 172.21.0.4).

Reititysesimerkki...

- Kone tt4 näkee oletusasetuksilla kaiken paitsi lonka[568]:n sisäverkkoja 192.168.12[568].0 (ja siten tt5:ttä), vastaavasti tt5 näkee kaiken paitsi lonka[578]:n sisäverkkoja (ja tt4:ää).
- Toisin sanoen tt4 näkee lonka7:n sisäverkon (192.168.127.0/24), labran sisäverkon (172.21.0.0/16) ja ulkomaailman (1-to-many NAT), mutta se itse näkyy vain lonka7:n sisällä ja lonka7:n kautta reitittäville koneille.

Reititysesimerkki...

- Jotta kone tt1 näkisi myös tt5:n, kaksi vaihtoehtoa:
  - Oletusreitiksi lonka6 (172.21.208.16)

    gateway 172.21.208.16
  - Erillinen reititys lonka6:n kautta sen sisäverkkoon (tai haluttaessa vain tt5:een)

    gateway 172.21.0.1

    ip route add 192.168.126.0/24 via 172.21.208.16

Reititysesimerkki...

- Jotta kone tt4 näkisi tt5:n:
  - Oletusreitin oltava kuitenkin aina lonka7 (192.168.127.1)
  - Aliverkolle 192.168.126.0/16 määriteltävä erikseen reitti, reitittimenä lonka6 (172.21.208.16)

    gateway 192.168.127.1

    ip route add 192.168.126.0/16 via 172.21.208.16

Reititysesimerkki...

- Reitit saa pysyviksi (bootissa automaattisesti asettuviksi) eri tavalla riippuen siitä onko käytössä vanha ifupdown (Ubuntu 17.04 ja vanhemmat, myös niistä päivitettynä uudemmat) vai uudempi netplan (Ubuntu 17.10 ja uudemmat uutena asennuksena); voi vaihtaa myöhemminkin:

  apt install ifupdown; apt purge netplan.io
- ifupdown-asennuksessa verkkomääritykset tehdään tiedostossa /etc/network/interfaces
- netplan-asennuksessa määrittelytiedosto on yleensä /etc/netplan/01-netcfg.yaml

  (varoitus: tab-merkit eivät toimi yaml-tiedostoissa!)

Reititysesimerkki, ifupdown

- /etc/network/interfaces tt1:ssä:

```
...
auto ens3
iface ens3 inet static
    address 172.21.209.19
    netmask 255.255.0.0
    broadcast 172.21.255.255
    gateway 172.21.0.1
    up ip route add 192.168.127.0/24 via 172.21.208.17
```

Reititysesimerkki, netplan

- /etc/netplan/01-netcfg.yaml tt2:ssa:

```
...
ethernets:
    ens3:
        addresses: [ 172.21.209.119/16 ]
        gateway4: 172.21.0.1
        routes:
          - to: 192.168.127.0/24
            via: 172.21.208.17
```

ip

- Yleiskomento verkkoasetusten säätämiseen, korvaa vanhat erilliset komennot komennot ifconfig, route, netstat ja arp (mutta Linux-spesifi, jälkimmäisiä tarvitaan yhä mm. *BSD:ssä, ja ne toimivat useimmissa Linux-jakeluissakin yhä)

ip

- Paljon toimintoja, yleinen syntaksi

  ip [*optiot*] *kohde* { *komento* }

  missä kohde voi olla jokin seuraavista:

  link, address, addrlabel, route, rule, neighbour, ntable, tunnel, tuntap, maddr, mroute, mrule, monitor, xfrm, netns, l2tp, tcp_metrics

  (useimmat voi lyhentää, esim. ”ip ro” jne)

ip

- Yleisiä optioita -r[esolve]: hae nimet DNS:stä (vanhoissa route, netstat jne komennoissa dns-haku tehtiin oletuksena ellei -n -optiota annettu, ip:ssä käytetään numeroita ellei -r:ää ole annettu)
- -s[tatistics], -h[uman-readable], -d[etails], -c[olor], -br[ief]
- -o[neline] (hyödyllinen mm. grepin ja wc:n kanssa)
- Komentoja voi lukea tiedostosta optiolla -b[atch] *file* (jatkaa virheiden jälkeen jos annetaan myös -force)
- Paljon muitakin optioita

ip route

- Reititystaulun tutkimiseen ja muokkaamiseen (korvaa vanhan route-komennon)**.** Yleisimmät käyttötavat ja joitakin esimerkkejä (ks. man ip-route):

  ip route [show]

  - Näyttää reititystaulun; lyhennettävissä esim. ”ip ro”

  ip rou add *address*[/*mask*] via *router* [dev *device*]

  ip route add 192.168.122.0/24 via 172.21.208.17 dev eth0

  ip ro delete ...

  - Argumentit kuten add-komennossa (tai kuten show näyttää)

ip route

ip route change ...

ip route change default via 172.21.208.17

ip route replace ...

- Kuten change mutta ei valita vaikka reittiä ei ennestään olisi

ip route get *address*

- Selvittää reitin yksittäiseen osoitteeseen

ip route save >*file*

- Kuten show mutta tulostus binäärimömmö, jota restore ymmärtää

ip route restore <*file*

ip link

- Verkkolaitteiden asetukset (korvaa osin ifconfig-komennon). Paljon alakomentoja ja optioita (ks. man ip-link), yleisimpiä:

  ip link [show]

  - "show" oletus, voi jättää pois ellei lisäargumentteja ole

  ip -statistics link show

  - vrt. ip -s -br li, ip -s -d li

  ip lin set dev eth0 up

  ip li set dev eth0 arp off

ip address

- Verkko-osoitteiden asetukset (ks. man ip-address):

  ip [-br] address [show] # -br[ief] yleensä hyvä

  - ”show” voidaan jättää pois ellei lisäargumentteja ole

  ip addr show eth0

  ip -br a s up

  ip ad add 192.168.5.6/24 dev eth0

  - ip alias = toinen ip samalle interfacelle

  ip ad broadcast 192.168.5.255 dev eth0

  ip ad delete 192.168.5.6/24 dev eth0

ip neighbour

- Kuvaus IP → MAC (hardware) address
- Naapuritaulun käsittely korvaa vanhan arp-komennon ja tarjoaa vastaavan toiminnallisuuden myös IPv6:een
- Naapuritaulussa (IPv4:ssä myös ARP-taulu, sanoista Address Resolution Protocol) on tieto siitä, mikä IP vastaa mitäkin hardware-osoitetta ja interfacea; päivittyy yleensä automaattisesti
- Hyödyllinen esim. IP-päällekkäisyyksiä jäljitettäessä

ip neighbour

- Naapuritaulua voi tutkia ja muokata, esim (optioita on paljon enemmänkin, ks. man ip-neighbour):

  ip [-s] neighbour [show]

  ip neighb flush dev eth0

  ip neigh add *ip-address hw-address* dev *device*

  ip neig change ...

  ip nei replace...

  ip ne delete ...

ip neighbour esim.

- Mitä lonka5:n interfacea virtuaalikone tt1 käyttää?

  host tt1 # → 172.21.209.19

  ip neigh | grep 172.21.209.19

  172.21.209.19 dev br0 lladdr 52:**54:00:4c:6a:19**

  ip -oneline link | grep 54:00:4c:6a:19 # alku 52: -pois

  17: **vnet0**: <BROADCAST,MULTICAST,UP,…
- Em. toimii vaikkei olisi oikeuksia virsh-komentoon, jolla tiedon saa helpomminkin:

  virsh domiflist tt1

ARP proxy

- Kone reagoi toiselle (”fyysisen” aliverkon ulkopuolella olevalle) koneelle (koneille) tarkoitettuihin paketteihin kuin olisi se itse ja ohjaa ne perille (esim. toisen verkkokorttinsa kautta, jotain tunnelia pitkin tms).
- Mahdollistaa aliverkon jakamisen osiin tai kahden samaa IP-avaruutta käyttävän aliverkon yhdistämisen läpinäkyvästi: aliverkon muiden koneiden ei tarvitse tehdä mitään reitityksiä tms tai edes tietää järjestelystä.
- Harvoin tarpeen nykyisin.

SSH port forwarding

- Ssh osaa välittää pääteyhteyden lisäksi melkein mitä vain tcp/ip-liikennettä
- Mahdollistaa yksittäisten porttien ohjaamisen (reitittämisen) eri koneille
- Edellyttää että ssh-yhteys muodostettavissa, IP-tason reitityksen (ja tcp:n) pitää siis toimia ensin
- Yhteys kryptografisesti salattu ja autentikoitu

Port forwarding: local → remote

-L [bind_address:]port:host:hostport

Ohjataan lokaalin koneen portti etäkoneen porttiin: lokaalin koneen porttia voidaan käyttää kuin se olisi etäkoneessa. Huom. *host* ei yleensä ole sama kone johon ssh-yhteys otetaan, vaan jokin kone sen verkossa. Jos se on palomuurin takana, nimipalvelu ei yleensä toimi vaan tarvitaan IP.

Port forwarding: remote → local

-R [bind_address:]port:host:hostport

Ohjataan etäkoneen portti lokaalin koneen porttiin: etäkone voi käyttää lokaalin koneen porttia kuin omaansa. Tässä *host* on jokin kone lokaalin koneen kanssa samassa verkossa (ja nimipalvelukin toimii).

Port forwarding options

-N Ei suoriteta mitään komentoa (vain porttiohjaus)

-f Jätetään ssh taustalle (mahd. salasanakyselyn jälkeen)

-n sulje ssh:n stdin (ei toimi salasanakyselyn kanssa)

-g Salli etäyhteydet lokaaleihin forwardoituihin portteihin

-A salli (-a kiellä) ssh-agentin forwardointi

(enemmänkin on...)

Local port forwarding esimerkki

- Esimerkki: kotiverkossa on kone johon pääsee ssh:lla (”koti1”, sisäinen IP 192.168.1.5) ja toinen kone jossa pyörii ”koti-intranet” (”kotiwww”, 192.168.1.6). Komento (ulkopuolisesta koneesta):

  ssh koti1 -L 8080:192.168.1.6:80 -N -f

  ja selaimessa http://localhost:8080

  näkee koti-intranetin, mutta vain koneessa, jossa ssh pyörii.

Local port forwarding esimerkki

- Jos portti halutaan jakaa muillekin, asetettava bind_address, esim.

  ssh koti1 -L '*:8080:192.168.1.6:80' -g -N -f

  tai se voi olla määrätty ssh_config- tai .ssh/config tiedostossa: GatewayPorts=yes
- Etäkoneelle pitää yleensä käyttää IP:tä jos se on palomuurin takana (nimipalvelukutsu tehdään lokaalissa koneessa, joka ei tiedä palomuurin takaisia nimiä)
- -L -optioita voi antaa useita (eri porteilla ja koneilla)

Remote port forwarding esimerkki

- Esimerkki 1: Sama tilanne kuin edellä, mutta mennäänkin sisältä ulos avaamaan portti (kotikoneesta):

  ssh -R 8080:kotiwww:80 ulkokone

  ja jälleen (ulkokoneen!) portissa 8080 näkyy koti-intranet.
- Jälleen jako muille koneille bind_address -asetuksella:

  ssh -R '*:'8080:kotiwww:80 ulkokone

Remote port forwarding esimerkki

- Edellyttää, että forwarding on sallittu ulkokoneen sshd_config -tiedostossa: GatewayPorts=yes (huom. eri merkitys kuin ssh_config'issa!)
- Tässä nimipalvelu toimii kuten sisäverkossa yleensäkin, koti-intranetille voi käyttää nimeä (”kotiwww”)
- -R optioitakin voi antaa useita eri koneille ja porteille

Remote port forwarding esim2

- Esimerkki 2: palvelin mokkulan päässä, dynaaminen IP, NAT, puhelinyhtiö ei salli yhteyksiä sisäänpäin. Avataan käänteinen ssh- ja http-yhteys etäkoneesta:

  ssh -R 52022:localhost:22 -R 8080:localhost:80 julkikone

  ja julkikoneesta pääsee takaisin:

  ssh -p 52022 localhost

  wget http://localhost:8080/index.html

Remote port forwarding: autossh

- Usein kätevä **autossh**-ohjelman kanssa käynnistettynä /etc/rc.local -tiedostosta tyyliin

  autossh -M 50023 -f -N -g -R 50022:localhost:22 user@ulkokone

  (-M on autossh:n oma optio, jolla valitaan portti jota se käyttää lähinnä seuraavan (tässä 50024) kanssa yhteyden päälläpysymisen testaamiseen, muut se välittää ssh:lle. Vrt. TCPKeepAlive, ClientAliveCountMax ja ClientAliveInterval, ServerAliveCountMax, ServerAliveInterval)

Ssh: X forwarding

- ”X Window System”
- Etäkone pääsee käsiksi lokaaliin näyttöön &c (terminologia: ”X server” = lokaali kone jossa näyttö on, ”client” etäkoneen ohjelma, joka sitä käyttää)
- Etäkone pääsee myös hiireen ja näppäimistöön käsiksi, vaarallista
- Optiot -X, -Y, -x
- Ympäristömuuttuja $DISPLAY

Ssh: X forwarding

-Y rajoittamaton lokaalin näytön etäkäyttö; vaarallinen, etäkone voi tehdä ”mitä vain”, kaapata näppäimistön, sotkea lokaaleja (X-)ohjelmia jne (aikojen alussa tämä oli ainoa vaihtoehto...)

-X rajoitettu ja turvallisempi: etäkoneen ohjelmat eivät pääse suoraan käsiksi lokaaleihin ohjelmiin ja niiden käynnistymisaika on rajoitettu xauth-tokenilla. Silti myös -X on vaarallinen, etäkone pääsee helposti mm. seuraamaan näppäimistöä

-x estää X-forwardoinnin (jos se on oletuksena päällä)

Tilapäisesti kumottavissa (esim. aliprosessille) tyhjentämällä ympäristömuuttuja $DISPLAY (DISPLAY= ./proggis)