Linuxi võrgutamine III

ETTEVALMISTUS

Selle harjutuse tegemiseks:
Võid kasutada põhjana harjutus "Linuxi võrgutamine II" osa III loodud virtuaalarvuteid
või
teed uuesti järgnevad tegevused:

Selle harjutuse sooritamiseks on vaja importida S-x.ova 3 korda.
Impordil pane "linnuke" kasti " reinitialize MAC addresses...

Ettevalmistus. Seadista VMid järgnevalt:
Ole tähelepanelik, kui seadete juures eksid, siis asjad käima ei lähe.

1. Impordi järjekorras saavad arvutid nimed S-x, S-x_1 ja Sx_2
2. **S-x**
    1. Lisa võrguadapter Adapter1 NAT
    2. Lisa võrguadapter Adapter2 "Internal network" intnet1 (lisa "1" adapteri nimele)
    3. **Igaks juhuks uuenda mac aadress "noolekestega" nupp MAC aadressi kasti taga**
    4. Lisa võrguadapter Adapter3 "Internal network" intnet2 (lisa "2" adapteri nimele)
    5. **Igaks juhuks uuenda mac aadress "noolekestega" nupp MAC aadressi kasti taga**
3. **S-x_1**
    1. Lisa võrguadapter Adapter1 "Internal network" intnet1 (lisa "1" adapteri nimele)
    2. **Igaks juhuks uuenda mac aadress "noolekestega" nupp MAC aadressi kasti taga**
4. **S-x_2**
    1. Lisa võrguadapter Adapter1 "Internal network" intnet2 (lisa "2" adapteri nimele)
    2. **Igaks juhuks uuenda mac aadress "noolekestega" nupp MAC aadressi kasti taga**

Kuna nendes virtuaalarvutites on vaikepaigalduses network-manager, tuleb kõigepealt tagada, et network-manager meid segama ei hakkaks.

S-x
Kommenteeri sisse eth0, lisa eth1 ja eth2 deklaratsioonid (inet manual)
S-x_1
Lisa eth0 deklaratsioon (inet manual)
Näide
auto eth0
iface eth0 inet manual
S-x_2
Jäta esialgu rahule

Tee võrguteenusele restart
(Võid ka network-manageri üldse seisma panna (service network-manager stop) või üldse süsteemist eemaldada apt-get remove –purge network-manager)

Omista masinatele oma-lemmik-töövõtet kasutades IP aadressid ja seadista vaikelüüs järgnevalt

```
S-x
eth0 dhcp
eth1 172.16.88.254/24
eth2 172.16.99.254/24
S-x_1
eth0 172.16.88.100/24
gw 172.16.88.254
```

S-x_2
Jäta esialgu rahule

Omista "klient" VM-idele toimiv nimeserver omal valikul (echo nameserver 8.8.8.8 > /etc/resolv.conf)

Lülita sisse marsruutimine VM-is S-x

```
sysctl -w net.ipv4.ip_forward=1
```

Seadista sNAT VM-is S-x

```
iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
```

Paigalda tshark
```
apt-get update
apt-get install tshark
```

Kontrolli seadete õigsust! Hetkel peab olukord olema selline:
Ping S-x ja S-x_1 vahel toimib!
Mõlemast VM-ist on võimalik pingida välisvõrku (google.com) nii nime kui IP järgi.
Oled valmis asja kallale asuma!

1. IP alias vs. VLAN 802.1q

Võrke on võimalik segmenteerida nii L3 (erinevad alamvõrgud) kui L2 tasmel (VLAN või segmendid) neid kahte kontseptsiooni ei tohi segamini ajada. Vaatleme seda protsessi lähemalt.

IP alias

S-x_1 lisame IP aliase

```
ip addr add 172.16.77.100/24 brd + dev eth0 label eth0:1
```
Vaata seadeid käsuga
```
ip addr
```
Näed, et eth0 juurde on lisandunud veel eth0:1 uue IP aadressiga
```
ping 172.16.77.254
```
(jäta käima)

S_x

```
ip addr add 172.16.77.254/24 brd + dev eth1 label eth1:1
```

Veendu S-x_1 aknas, et ping hakkas vastama.

Ava S-x_1 aknas teine konsool (alt F2), logi sisse ja käivita

```
ping 172.16.88.254
```
(jäta käima)
S_x

käivita tshark (kui alustasid „puhtalt lehelt" siis apt-get update, apt-get install tshark)

```
tshark -i eth1
```

Mida näed väljundist. Näed, et ühe liidese peal liiguvad 2 erinevasse IP alamvõrku kuuluvad

paketid. Seega pole tegemist „päris” liidesega, vaid samasse L2 segmenti on pandud mitu L3 alamvõrku.

2. VLAN (802.1q)

Kui IP alias tekitas ainult nimeliselt virtuaalse liidese, siis VLAN tagged liides on „päris” liides ja L2 kihis eraldatud.

S-x_1

katkesta kõik pingid ja lisa kerneli moodul VLAN toe jaoks

```
modprobe 8021q
```
Veendu, et moodul sai laetud
```
lsmod
```

Lisame uue „tagged” liidese

```
ip link add link eth0 name eth0.66 type vlan id 66
```

Käsk tekitab virtuaalse L2 liidese, mis hakkab eetrikaadreid märkima märgendiga 66 (VLAN nr 66) (me ei kasuta TAGi 1, kuna see kuulub nn. "default VLANile ja selle kasutamine on pea alati ebasoovitav, Samuti ei kasuta me VLAN 0 kuna see on eriotstarbeline.). Iseenesest võiksime kasutada suvalist märgendit 2–4096. Antud juhul kasutame me 66 kuna meil juba on võrgud ...88..., ...99... ja ...77... Hiljem paneme sellesse etherneti segmenti IP alamvõrgu ...66... Selguse mõttes on VLAN tagi numbri valmisel mõistlik hoida seal mingit „vihjet” milline IP alamvõrk seal võiks resideeruda. Rõhutan, see on selguse mõttes ja puhtalt oma vaba valik, et hiljem oleks elu lihtsam. L2 TAGi number ja IP subnet ei ole tehniliselt mingil moel seotud.

Paneme veel tähele, et tegelikult ei sunni mitte miski meid tegema tagged liidest sama nimega (name) milliste TAGidega eetrikaadreid ta hakkab edastama (vlan id). Aga teha need kaks erinevad on ausalt ütelda tõeline sigadus sellele, kes pärast teid peab süsteemi haldama hakkama ja ise võib ka sellega endale valusasti jalga tulistada. Seega hoiame ikka liidese nime ja TAGi numbri sünkroonis nagu viisakad inimesed kunagi.

Veendume, et liides tõepoolest tekkis

```
ip link show
```

Veel saame siit väljundist välja lugeda, et meie eelmises osas tehtud „liides” eth0:1 ei ole tegelikult liides „etherneti” tähenduses ja seega teda selles nimekirjas näha ei ole.

Omistame liidesele aadressi

```
ip addr add 172.16.66.100/24 brd + dev eth0.66
```

Pane ühes konsoolis marsruuteri poole teele

```
ping 172.16.77.254
```
ja teises
```
ping 172.16.66.254
```
(jäta käima)

S-x
Sooritame siin analoogsed tegevused

```
modprobe 8021q
ip link add link eth1 name eth1.66 type vlan id 66
ip addr add 172.16.66.254/24 brd + dev eth1.66
```

Veendu S-x_1, et ping hakkas vastama. Kui ei hakanud, kontrolli oma seadeid, veendu et liides on mõlemas masinas ka reaalselt „up".

S-x_1
käivita tshark

```
tshark -i eth0
```

Näed mõlemasse võrku kuuluvad ping pakette ja vastuseid

```
tshark -i eth0.66
```

Näed ainult VLAN66 kuuluvad ping pakette ja vastuseid.

Kui tegid kõik õigesti, oled edukalt loonud virtuaalse kohtvõrgusegmendi.
Võrgus on 1 "füüsiline" L2 segment (intnet1) üks virtuaalne segment VLAN66. L3 (IP) tasemel on kokku 4 IP alamvõrku.

Linuxi võrgutamine – networkmanager way

Kasutatavad VM-d on S-x (eelnevalt seadistatud) ja S-x_2

Käivita **S-x_2** ja veendu, et vaikekonfiguratsioonis on liideste kirjeldused /etc/network/interfaces sees välja kommenteeritud ja seega on liidesed network-manageri hallata. (loopback liides jäta rahule!)

Networkmanageri seadistamiseks käsurealt on utiliit nmcli
nwcli süntaks erineb oluliselt „klassikaliste" utiliitide „võtmete" süsteemist ja meenutab natuke OOP programeerimiskeelt.
Metasüntaks
nmcli [võtmed] [objekt] parameeter väärtus käsud

Erakordselt hea ja näideterikas nmcli juhend
https://docs.fedoraproject.org/en-US/Fedora/20/html/Networking_Guide/sec-Using_the_NetworkManager_Command_Line_Tool_nmcli.html

S-x_2

Veendu, et teenus töötab

```
service network-manager status
```

Vaatame võrguühenduse seadeid

```
nmcli dev status
```

Peaksid nägema midagi umbes sellist:

```
eth0    ethernet  connecting (getting IP configuration)  Wired connection 1
```
Siit võib järeldada, et networkmanager on leidnud ühe füüsilise liidese ja püüab nüüd sellele DHCP abil aadressi hankida. Meil DHCP serverit ei ole, seega tuleb liides seadistada manuaalselt.

Püüame liidesele IP aadressi omistada

```
nmcli con add ifname eth0 type ethernet ip4 172.16.99.100/24 gw4 172.16.99.254
```
„con add“ -- lisa ühendus
ifname – millist liidest kasutada
type ethernet – ühenduse tüüp
ip4 – ipv4 aadress
gw4 – vaikelüüsi aadress.
Peaksid nägema umbes sellist pilti.
```
Connection 'ethernet-eth0' (3a8ac0ec-eab1-4450-bfe2-1357d270e4b6) successfully
added.
```
Proovime pingida vaikelüüsi

```
ping 172.16.99.254
Network unreachable
```

Mis jama see siis nüüd on? Vaatame järele mis iproute arvab.

```
ip addr
```

Näe kus tegelane, raporteeris „success“ aga ip aadressi polegi. Vaatame, mida networkmanager ise asjast arvab.

```
root@S-x:~# nmcli con show
NAME                UUID                                  TYPE            DEVICE

ethernet-eth0       a3d4141e-2570-460c-bb5e-596a9a0d4181  802-3-ethernet  --

Wired connection 1  87630ab6-0e9d-4a83-aace-d5b495b18430  802-3-ethernet  eth0

root@S-x:~# nmcli dev status
DEVICE  TYPE      STATE                                  CONNECTION
eth0    ethernet  connecting (getting IP configuration)  Wired connection 1
lo      loopback  unmanaged                              --
root@S-x:~# _
```

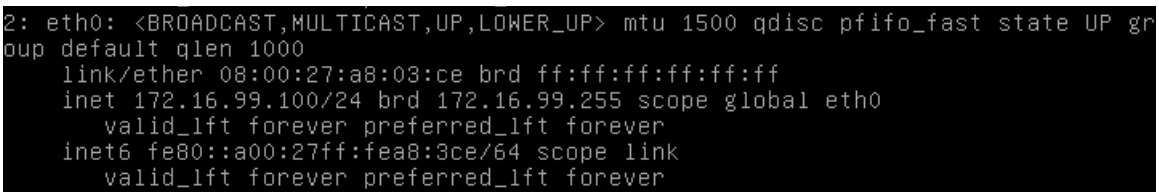
Mida sellest pildist järeldame. Mõtleme natuke järele ja saame aru, et nwmanager „omaenese tarkusest“ on juba tekitanud ühenduse nimega „Wired connection 1“ ja arvab, et see peab ilmtingimata DHCP abil ühenduse saama. Loomulikult meil eth0 enam kasutada ei lubata. Paneme tähele, et networkmanager lisab veel ühe (õppejõu meelest täiesti tarbetu) abstraktsioonikihi nimega „connection“.

Nüüd peame kuidagi nwmanagerile koha kätte näitama ja sellisele isetegevusele piiri panema. Võtame mittevajaliku ühenduse maha (NB jutumärgid!)

```
nmcli con down „Wired connection 1“
nmcli con del „Wired connection 1“
```

(keda huvitab, võib proovida, milliseid inetusi networkmanager talle ütleb, kui proovida kustutada liidest seda eelnevalt maha võtmata)

Peale vastuolulise seadistuse likvideerimist peaks vastu vaatama selline, oluliselt meeldivam pilt



Ka ping gw suunas peaks saama nüüd vastuse. Samuti 8.8.8.8 (kui marsruuteri ikka õigesti seadistasid). Kui vastust ikka ei tule, kontrolli üle, et sul oleks ainult üks „connection" nimega

Järgnevalt tuleks paika panna DNS serveri aadress.

```
nmcli con mod ethernet-eth0 ipv4.dns 8.8.8.8
```

con mod – modifitseerime olemasoleva ühenduse seadeid
ipv4.dns – DNS serveri aadress (võim määrata korraga ka mitu, tühikuga eraldatud, mitme aadressi korral tuleb panna nad jutumärkidesse, näiteks „8.8.4.4 8.8.8.8")
(Miks aadress on sünaksiga ip4 ja dns sünaksiga ip**v**4 – jumal üksi teab. Linuxi kurikuulus inkonsistentsus)

Proovi mõnda hosti nime järgi pingida? Saad vastuse? Vägev networkmanager ei suuda ju seadeid jooskvalt modifitseerida. Meil tuleb teha veeltähele

```
nmcli con down „ethernet-eth0"
nmcli con up „ethernet-eth0"
```

Nüüd võiks ka ping nime pihta vastuse saada.

Ainuke positiivne asi kogu selle aju valutama paneva „loogika" juures on, et mida te nmcliga ka seadistate see nii ka jääb. Muutused salvestatakse automaatselt on ja persistentsed.
Proovi järele, tee näiteks S-x_2-le reboot ja veendu, et ip seaded on jätkuvalt paigas.

IP alias ja 802.1q

Lisame nmcli abil ip aliase

```
nmcli con mod „ethernet-eth0" +ipv4.addresses 172.16.55.100/24
```

Pane tähele plussikest, mis annab nmcli-le teada, et aadress tuleb lisada (mitte olemasolev asendada)
Loomulikult pead liidesele veel ka taaskävituse tegema.

Mõtle, millisele liidesele ja millise aadressi (iproute abil) peaksid lisama „marsruuteris" S-x nii et S-x ja S-x_2 TCP ühendus toimiks ...55... alamvõrgu kaudu. (Abiks on esimene IP aliase harjutus) (tee seda!)

802.1q VLAN
Enne tegutsema asumist veendume, et 8021q moodul **EI OLEKS** laetud

```
lsmod | grep 802
```

Kui me midagi vastuseks ei saa, on kõik ok. Kui mingil eksikombel peaks see moodul olema laetud, siis eemalda käsuga

```
rmmod 8021q
```

Lisame VLAN44 tagitud eetrikaadrite jaoks sobiva ühenduse. Paneme tähele, et kuna vlan adapter on eraldiseisev (mis sellest, et mitte füüsiline) ethernet adapter, siis peame selle jaoks looma ka uue ühenduse (connection)

```
nmcli con add con-name VLAN44 type vlan ifname eth0.44 dev eth0 id 44
```

Erisused: seekord määrame ise ühendusele nime, samuti paneme loodavale liidesele „klassikalise“ nime eth0.44 (kuigi selle nimeks võiks ka „saabas“ määrata aga viisakad inimesed nii ei tee).

Vaatame üle

```
nmcli dev show
nmcli con show
```

```
lsmod | grep 802
```

Näeme ühte nwmanageri kahtlasevõitu „väärtust“ -- koostöös udeviga suudeti „vajadusepõhiselt“ laadida 802.1q toe moodul automaatselt.

```
nmcli con show VLAN44
```

Liides ja ühendus on, aga loomulikult on tõbras meile vaikimisi seadena DHCP (auto) toppinud. Ilmselt jäi seadistustes midagi puudu. Seekord teeme nii. Kõigepealt likvideerime selle liidese...

```
nmcli con down VLAN44
nmcli con del VLAN44
```

Loome uuesti täiskomplektis

```
nmcli con add con-name vlan44 type vlan ifname eth0.44 dev eth0 id 44 ip4
172.16.44.100/24 gw4 172.16.44.254
```

Ehk siis loome uue ühenduse nimega vlan44, tüüpi 802.1q, loome uue vlan liidese nimega eth0.44 , ütleme, et füüsiline liides kus sellise märgendiga (tag) eetrikaadrid liiguvad on eth0 ja lõpuks lisame ka kõige tähtsama – vlan identifikaatori mille järgi sellesse vlani kuuluvad kaadrid ära tuntakse – „44“. Lisaks määrame kohe ka IP ja vaikelüüsi.

Lisa iseseisevalt, analoogiliselt teisele harjutusele (iproute abil) „marsruuteris” S-x õigele liidesele samuti vlan44. Lisab sobiv IP aadress ...44... võrgust ja veendu, et saad ühenduse selle vlani ja sobiva IP võrgu kaudu. Kui hakkama ei saa, küsi nõu.

Static route ja networkmanager

Proovime lisada ka ühe static route. NB! See eeldab, et oled oma 44 vlani ühenduma saanud! Juhendit uurides komistame kohe järgmise inkonsistentsuse otsa. nmcli abil käsurealt staatilist marsruuti (route) lisada ei saagi. Kõigepealt peame käivtama nmcli interaktiivses režiimis.

```
nmcli con edit „vlan44“
```

Näed nmcli viipa

```
nmcli>
```

sinna sõnume järgmiselt

```
set ipv4.routes 8.8.8.8/32 172.16.44.254
```

Hostiga mille IP on 8.8.8.8 maskiga 32 (host mask!!!) tuleb ühendust võtta marsruuteri kadudu

mille IP-ks on 172.16.44.254 (Sa ju ikka panid marsruuteris liidesele eth2.44 just selle IP eelmise harjutuse lõpetuseks :))

```
verify
```

Kui näpuvigu ei teinud, teatab nmcli „verify connection:OK“

```
save persistent
```

Kui muidu salvestatakse muudatused automaatselt, siis editori režiimis tuleb need miskipärast salvestada? Ja lisaks „persistentsele” on veel mingi MITTEpersistentne salvestamine. Confusing!

Väljume käsuga

```
quit
```

ja kuna me seda nwmanageri sugugi ei usalda, siis kontrollime üle käsuga

```
ip route
```

No muidugi pole meil mingit routet, seega

```
nmcli con down vlan44
nmcli con up vlan44
```

Nüüd on pilt juba parem ja testiks võime veel kontrollida

```
ping 8.8.8.8
```

Kui tegid kõik õigesti, siis saad pingile ka vastuse.
Kävita S-x peal tshark ja veendu, et ICMP paketid 8.8.8.8 pihta tulevad justnimelt läbi eth2.44 liidese!

Pärast tänast harjutust võib igaüks juba ise otsustada, kas talle meeldib rohkem iproute või networkmanager. Paraku osades uuemates distrotes (CentOS) on vanakooli riistad juba välja jäetud ja näiteks „ifconfig“ utiliiti enam ei olegi. Ilmselt pole ka teil selle rakenduse eest pääsu, meeldib see teile siis või mitte.
NB! Kuigi networkmanager on „tagasiühilduv“ skriptipõhiste seadistusmudelitega on **TUNGIVALT EBASOOVITAV** neid läbisegi kasutada. See on kõikvõimalike tarbetute probleemide allikas. Peate ise otsustama, kas annate networkmanagerile kinga ja teete asja „0ldsk00l“ moel või siis kasutate 100% networkmanageri oma võrguühenduste haldamiseks.

Kui aju väga veel ei valuta

Proovi lahedat viga.
S-x_1 teeme **tahtliku** näpuka

```
nmcli con add con-name VLAN3 type vlan ifname eth0.3 dev eht0 id 3
```

(Kas näed kus on viga?) Sisesta käsk ja imetle, kuidas Networkmanager raporteerib edukast ühenduse loomisest.

```
nmcli dev status
nmcli con show
nmcli con show VLAN3
```

Kes ikka veel ei märganud, siis liidese nimi, kuhu me VLAN liidese lisasime on valesti kirjutatud.
Proovime oma viga parandada (samuti mitte kõige mõistlikumal moel)

```
nmcli con add con-name VLAN3 type vlan ifname eth0.3 dev eth0 id 3

nmcli dev status
nmcli con show
```

Näe kus lugu, nüüd meil on KAKS liidest sama nimega, neist üks vigane ja teine korrektne
(mõistlik oleks olnud ütelda con mod ... jne)
Katsume vigasest liidesest lahti saada

```
nmcli con del VLAN3
nmcli con show
```

Uh-oh! Mõlemad liidesed läksid ära. Nii vigane kui korrektne.
Olgu see hoiatavaks meeldetuletuseks, et admini näpuvigadel on koledad tagajärjed ja isegi „sõbraliku“ moega utiliidid ei oska neid tihtipeale parandada.