

# 5G-PRIVAATTIVERKON ASENNUS JA KÄYTTÖÖNOTTO

Lapin ammattikorkeakoulu / AGRARSENSE-hanke

Parpala Samuli

Opinnäytetyö

Tieto- ja viestintäteknikan koulutus  
Insinööri (AMK)

2026

Tieto- ja viestintätekniiikan koulutus  
Insinööri (AMK)

---

<b>Tekijä</b>	Samuli Parpala	<b>Vuosi</b>	2026
<b>Ohjaaja</b>	Ari Karjalainen		
<b>Toimeksiantaja</b>	Lapin ammattikorkeakoulu / AGRARSENSE-hanke		
<b>Työn nimi</b>	5G-privaativerkon asennus ja käyttöönotto		
<b>Sivumäärä</b>	54 + 1		

---

Opinnäytetyön aiheena on privaativerkon rakennus ja Cumucoren toimittaman core-ohjelmiston asennus rakennetulle verkolle. Lopputulos on laajennus nykyisen Off-Grid 5G -järjestelmän rinnalle, jolloin käytössä on kaksi standalone-tekniologiaan perustuvaa toteutusta coren ajamiseksi: alkuperäinen toteutus, jossa Nokian core toimii ulkoisessa datakeskuksessa, sekä uusi toteutus, jossa Cumucoren core toimii paikallisesti. 5G SA -verkon rinnalle rakennettiin myös erillinen 4G-verkko, jonka core toimii paikallisesti.

Opinnäytetyön raportin alussa tutustutaan opinnäytetyön kannalta keskeisiin teknologioihin ja käytettäviin ohjelmistoihin, kuten 4G- ja 5G-teknologioihin, ydinverkkoihin, itsenäisiin ja ei-itsenäisiin verkkoihin, privaativerkkoihin sekä virtuaalikonejärjestelmiin. Erilaisten tekniikoiden ja ohjelmistojen teoriaan tutustutaan pienissä määrin myös käytännön osion aikana.

Seuraavassa vaiheessa syvennytään itse opinnäytetyön ydinaihepiiriin eli verkon rakennukseen sekä tarvittavien ohjelmistojen asennukseen ja käyttöönottoon. Osiossa käydään läpi virtuaalikoneiden isäntäkoneena toimivan laitteen soveltuvuuden varmistaminen, Ubuntu-käyttöjärjestelmän asennus isäntä- ja virtuaalikoneille, virtuaalikoneiden luonti, käytettävien laitteiden konfigurointi ja lopuksi verkon toiminnan todentaminen. Virtuaalikoneiden varmuuskopiointi on myös osakseen käyty raportin ylläpitoluvussa läpi.

Lopuksi tutustutaan SIM-korttien ohjelmointiin tutkimalla valmiiksi verkkoon ohjelmoitujen SIM-korttien tietoja ja soveltamalla tästä kertynyttä oppia uuden kortin ohjelmoimiseen rakennetussa verkossa toimivaksi.

Avainsanat

4G-tekniikka, 5G-tekniikka, langaton tekniikka, Linux, SIM-kortit, virtualisointi

Degree Programme in Information  
and Communication Technology  
Bachelor of Engineering

---

<b>Author</b>	Samuli Parpala	<b>Year</b>	2026
<b>Supervisor</b>	Ari Karjalainen		
<b>Commissioned by</b>	Lapland UAS / AGRARSENSE-project		
<b>Title</b>	Private 5G Network Installation and Deployment		
<b>Number of pages</b>	54 + 1		

---

The subject of this thesis is to build a private network and to install core software for it. Final product is an expansion to the current Off-Grid 5G system, which then allows for two standalone architecture-based solutions for running the core: the original solution, where Nokia core runs in an external datacenter and the new solution, where Cumucore core runs locally. An external 4G network with a local running core is built alongside the 5G SA network.

The essential technologies and software regarding the process are introduced at the beginning of the thesis report. These include 4G- and 5G technologies, core networks, standalone- and non-standalone networks, private networks and virtual machines. Various kinds of technology and software are also introduced briefly during the practical chapter of the report.

The next phase goes deeper into the core subject of the study, that is, building of the private network as well as installation and deployment of necessary software. This portion goes through inspecting the suitability of the device to be used as a host machine, installation of Ubuntu OS to host- and virtual machines, creation of virtual machines, configuration of implemented devices and verifying network functionality. Backing up the virtual machines is also partially gone through.

At the end of the practical portion of the thesis work, SIM-card programming is made familiar to a certain degree. Ready-made SIM-cards for the built network were examined and studied, and this knowledge was put to practice when programming a new SIM-card to function in built network.

**Keywords** 4G-technology, 5G technology, Linux, SIM-cards, virtualization, wireless technology

## SISÄLLYS

1	JOHDANTO .....	9
2	KÄYTETYT TEKNOLOGIAT .....	11
2.1	Laitteisto ja ohjelmisto .....	11
2.1.1	Laitteisto .....	11
2.1.2	Virtuaalikoneohjelmistot: QEMU-KVM ja Libvirt.....	12
2.1.3	Operointirajapinta: SSH.....	13
2.2	5G.....	14
2.3	Core.....	16
2.4	NSA- ja SA-verkot.....	17
2.5	Privaattiverkot.....	18
3	COREN ASENNUS .....	19
3.1	NUC-koneen vertailu coren laitteistovaatimukseen.....	19
3.2	Isäntäkoneen Ubuntu 22.04 LTS -käyttöjärjestelmän asennus.....	20
3.2.1	Valmistelut.....	20
3.2.2	Asennusprosessi .....	20
3.3	Virtuaalikoneet.....	21
3.3.1	Valmistelu .....	21
3.3.2	KVM-ohjelmistojen asennus ja virtuaalikoneen luonti.....	23
3.4	Core-ohjelmiston asennus asennusskriptillä.....	26
3.4.1	USB-laitteen kytkentä fyysisen koneen kautta virtuaalikoneelle ..	26
3.4.2	Asennusskriptin ajaminen.....	28
4	YLLÄPITO .....	29
4.1	SSH-yhteyksien konfigurointi.....	29
4.1.1	SSH-avainten generointi asiakaskoneella .....	30
4.1.2	SSH-avainten konfigurointi palvelinkoneelle.....	30
4.1.3	Autentikointimetodin konfigurointi .....	31
4.1.4	Reitittimen konfigurointi .....	33
4.2	IP-osoitteiden konfigurointi .....	33
4.2.1	Isäntäkoneen konfigurointi.....	35
4.2.2	Virtuaalikoneen konfigurointi .....	36
4.3	Verkon toiminnan todentaminen .....	37

4.4	Varmuuskopiointi .....	38
4.4.1	Virtuaalikoneen varmuuskopiointi .....	38
4.4.2	Varmuuskopion käyttö .....	40
5	SIM-KORTIT .....	41
5.1	pySim-prog.py .....	41
5.2	pySim-read.py .....	42
5.3	pySim-shell.py .....	43
5.4	SIM-kortin ohjelmointi .....	43
5.4.1	SIM-kortin ohjelmointi 4G-verkkoon.....	44
5.4.2	SIM-kortin ohjelmointi 5G-verkkoon.....	46
6	POHDINTA .....	48
	LÄHTEET.....	49
	LIITE .....	55

## KÄYTETYT LYHENTEET JA TERMIT

4G	Fourth Generation Cellular Network Technology, neljäs sukupolven langaton tiedonsiirtotekniikka
5G	Fifth Generation Cellular Network Technology, viidennen sukupolven langaton tiedonsiirtotekniikka
ACC	Access Control Class, käyttöoikeusluokka
AF	Application Function, sovellustoiminto
AMF	Access and Mobility Management Function
AUSF	Authentication Server Function
AVX	Advanced Vector Extensions, kehittyneet vektorilaajennukset
BTS	Base Transceiver Station, tukiasema
CC	Critical Communications
DHCP	Dynamic Host Configuration Protocol
DN	Data Network
EPC	Evolved Packet Core
FPLMN	Forbidden PLMN, kielletty PLMN
gNB	5G Node B
HPLMN	Home PLMN
HPLMNAcT	HPLMN with Access Technology
ICCID	Integrated Circuit Card Identifier
IMSI	International Mobile Subscriber Identifier, kansainvälinen matkaviestintilaajan tunnus
IoT	Internet of Things, esineiden internet
IP	Internet Protocol, internet-protokolla
ISO	International Standards Organization, kansainvälinen standardisoimisjärjestö
KVM	Kernel-based Virtual Machine, kernel-ytimeen pohjautuva virtuaalikone
LTE	Long-Term Evolution
LTS	Long-Term Support, pitkäaikaistuki
MCC	Mobile Country Code, maatunnus
MME	Mobility Management Entity

MNC	Mobile Network Code, matkaviestinverkkotunnus
MS	Mobile Station, matkaviestintälaite
MSISDN	MS International Subscriber Directory Number
NB	Node B
NE	Network Element, verkkoelementti
NEF	Network Exposure Function
NF	Network Function, verkkotoiminto
NG	Next Generation, uusi sukupolvi
NG-RAN	Next Generation Radio Access Network, uuden sukupolven radioliityntäverkko
NIC	Network Interface Card, verkkokortti
NR	New Radio
NRF	NF Repository Function
NR-Uu	New Radio UTRAN UE
NSA	Non-Standalone, ei-itsenäinen
NSSF	Network Slice Selection Function
NUC	Next Unit of Computing, seuraava laskennan yksikkö
OP	Operator Code
OPc	Operator Code (Derived)
OPLMN	Operator PLMN, operaattori PLMN
OPLMNwAcT	OPLMN selector with Access Technology
OS	Operating System, käyttöjärjestelmä
PCF	Policy Control Function
PLMN	Public Land Mobile Network, julkinen matkaviestinverkko
PLMNwAcT	User Controlled PLMN with Access Technology
QEMU	Quick Emulator
RAM	Random Access Memory, väliaikainen työmuisti
RAN	Radio Access Network, radioliityntäverkko
SA	Standalone, itsenäinen
SBA	Service-Based Architecture, palvelupohjainen arkkitehtuuri
SIM	Subscriber Identity Module
SMF	Session Management Function

SMSP	Short Message Service Parameters
SSD	Solid State Drive, puolijohdeasema
SSH	Secure Shell
UDM	Unified Data Management
UE	User Equipment, käyttäjälaite
UMTS	Universal Mobile Telecommunications System, maailmanlaajuinen liikkuvan televiestinnän järjestelmä
UPF	User Plane Function, käyttäjätason toiminto
URLLC	Ultra Reliable and Low Latency Communications, erittäin luotettava ja vähälatenssinen kommunikointi
USIM	Universal Subscriber Identity Module
UTRAN	UMTS Terrestrial Radio Access Network
VM	Virtual Machine, virtuaalikone
XDP	Express Data Path

## 1 JOHDANTO

Opinnäytetyö on toteutettu yhdessä Cumucoren kanssa Lapin ammattikorkeakoulun AGRARSENSE-hankkeelle. Valmis lopputulos on toimiva 5G-mobiili-privattiverkko sekä 4G-variantti samantyyppisestä verkosta, joita voidaan hyödyntää Lapin ammattikorkeakoulun IoT-laboratorion toiminnassa. Verkko on nykyisen Off-grid 5G network -järjestelmän laajennus. Opinnäytetyön aihepiiri on luonteva jatkumo kyberfyyssisten järjestelmien kehittäjän linjalle sen tarjotessa syvempää perehtymistä tietoliikennejärjestelmiin ja langattomiin verkkoihin. Verkon ohjelmallinen toteutus tapahtuu pääosin ilman graafista käyttöliittymää Linux-ympäristössä.

Nykyisen Off-grid 5G network -järjestelmän coren ohjaustason (*Control Plane*) verkkotoiminnot on keskitetty Nokian datakeskukseen Espooseen (Suutari-Jääskö 2025, 12). Cumucoren toimittaman coren asennus paikalliseen palvelinkoneeseen mahdollistaa näiden verkkotoimintojen suorittamisen paikallisella palvelinkoneella, jolloin yhteyttä ulkoiseen datakeskukseen ei tarvita. Näin mahdollistetaan täysin itsenäisen suljetun- eli privattiverkon toiminta, jota voidaan tarvittaessa liikuttaa haluttuun sijaintiin.

Raportti kattaa Cumucore-ohjelmiston asennuksen ja käyttöönoton eri vaiheet NUC13RNGi7-minutietokoneen sisään luotaville virtuaalikoneille, mukaan lukien laitteistovaatimukset core-ohjelmiston toiminnalle. Cumucore tekee osan core-ohjelmiston konfiguroinnista.

Kuvio 1 (s. 12) kuvastaa rakennettavan 4G-/5G core-verkon topologiasta rakennetta. Kaikki laitteet 10.10.10.0/24-verkossa ovat fyysisesti kytkettynä Switch 1:een. Suurin osa tämän verkon laitteista sijaitsee Lapin ammattikorkeakoulun Rovaniemen kampuksen IoT-laboratoriossa lukuun ottamatta Server Rackia ja 5G BTS:ää, jotka sijaitsevat Off-Grid 5G Mobile Network -peräkärjessä.

Tietoturvalliset etäkäyttömahdollisuudet ovat olennainen osa palvelimien parissa työskenneltäessä. Siihen työkaluna käytetään tässä tapauksessa SSH-protokollaa. SSH-protokollalle luodaan yksityiset avaimet asiakas- ja palvelinkoneiden

välille, jolloin yhteydenotto ei onnistu pelkästään IP-osoitteella, käyttäjätunnuksella ja salasanalla, vaan asiakaskoneen täytyy erikseen olla määriteltynä palvelimelle.

Lopuksi syvennytään pintapuolisesti SIM-korttien toimintaan ja ohjelmointiin käyttäen PySim-ohjelmasarjaa yhdessä SIM-kortinlukijan/-kirjoittajan ja Raspberry Pi 4:n kanssa. SIM-kortin sisällön tutkiminen ja kortin ohjelmointiin tutustuminen sisällytetään opinnäytetyöhön tietyin rajauksin. Tavoitteista poissuljetaan onnistunut opinnäytetyön aikana rakennetussa verkossa toimivan kortin ohjelmointi, sillä se poikkeaa osakseen opinnäytetyön ydinaihepiiristä.

Tämän opinnäytetyön käytännön osion toteutuksessa tekijä on hyödyntänyt tiedonhakyökaluna ChatGPT-versiota 4o. Kussakin tilanteessa tarkastettiin työkalun tarjoamat tiedonlähteet sen tuottamalle tekstille. Tiedonlähteistä valikoitiin luotettavat linkit, ja niiden perusteella muodostettiin tietoperusta pulman tai tilanteen ratkaisemiseksi. Tarvittaessa jatkettiin syvempää tiedonhakua valikoitujen tiedonlähteiden perusteella. Kuten tuntemattomat koodiblokit tai komennot yleisestikin, myös työkalun tuottamat koodiblokit ja komennot tulkittiin erinäisten tiedonlähteiden pohjalta, jotta ymmärrettiin mitä koodi tai komento tekee, ennen kuin sitä sovellettiin omaan käyttöön. Mitään salassa pidettävää, yksityistä tai arkaluonteista tietoa ei työkalulle annettu, vaan kunkin pulmatilanteen kohdalla, jossa työkalua hyödynnettiin, tilanne kuvailtiin työkalulle yleisiä esimerkkejä käyttäen.

## 2 KÄYTETYT TEKNOLOGIAT

Tässä luvussa käsitellään työssä käytettävää laitteistoa ja ohjelmistoa sekä työhön liittyvää teknologiaa. Aihealueet käsitellään pääpiirteittäin, mutta osaan niistä syvennytään lähemmin myöhemmissä luvuissa käytännön yhteydessä.

### 2.1 Laitteisto ja ohjelmisto

#### 2.1.1 Laitteisto

Tarkastellaan kuviota 1. Isäntäkoneena toimiva NUC13RNGi7-tietokone nähdään kuviossa nimikkeellä "HOST-PC". Isäntäkoneelle luodaan kaksi kappaletta virtuaalikoneita, joista toinen ajaa 4G-palvelinta ja toinen 5G-palvelinta. Virtuaalikoneet nähdään kuviossa nimikkeillä "4G-Core" ja "5G-Core". Isäntäkoneen soveltuvuus käyttötarkoitukseen on käyty läpi luvussa 3.1. (Uusitalo 2025d.)

4G-palvelimen tukiasema on kuvion alaosassa nimikkeellä "FZM / 4G-Pico-bbu" ja 5G-palvelimen tukiasema kuvion yläosassa vasemmalla nimikkeellä "5G BTS". Etäyhteydet verkkoon kulkevat Telewell 4G-reitittimen kautta, joka nähdään kuvion yläosassa oikealla. (Uusitalo 2025d.)

Tukiasemien konfigurointi tapahtuu hallintakoneena toimivan tietokoneen avulla, joka nähdään kuviossa nimikkeellä "Management PC". Kuten jo luvussa 1 mainittiin, kaikki verkon laitteet lukuun ottamatta core-verkkoihin liitettyjä käyttäjälaitteita, ovat Ethernet-kaapelilla kytkettynä verkkokyttimeen "Switch 1". (Uusitalo 2025d.)



miessa KVM vastaa prosessorikiihdytyksestä sekä muistiin liittyvistä toimenpiteistä ja QEMU käsittelee oheis- ja muiden laitteiden emuloinnin. (Hoke & Snyder 2025.)

**KVM** on *kernelmoduuli*, joka toimittaa virtualisoinnin kannalta oleelliset ominaisuudet käyttäen samalla hyödykseen prosessorin laitteistoavusteisia virtualisointilaajennuksia. Se mahdollistaa Linux *kernelin* toimimisen *type-1-hypervisorina*, virtuaalikoneiden luomisen ja hallinnan sekä virtuaalikoneiden toiminnan mukaan lukien prosessori- ja muistitoimenpiteet lähes natiivilla käytettävissä olevan fyysisen prosessorin teholla. KVM on suoraan vuorovaikutuksessa fyysisen laitteiston kanssa tarjoten näin tarvittavan infrastruktuurin täyden virtualisoinnin mahdollistamiseksi. (Hoke & Snyder 2025.)

**Libvirt** toimii hallintarajapintana monille virtualisointialustoille, kuten KVM/QEMUlle tässä tapauksessa. Sen kautta voidaan hallita muun muassa seuraavaa:

- virtuaalikoneen luonti
- virtuaalikoneen käyttö komentokehoteella tai graafisella käyttöliittymällä sekä virtuaalikoneen käynnistys, pysäytys ja sammutus
- virtuaaliverkkoasetukset, kuten NAT-tyypin tai verkkosillan konfigurointi
- etäyhteyksien, kuten SSH-yhteyksien hallinta. (Hoke & Snyder 2025.)

### 2.1.3 Operointirajapinta: SSH

SSH on ohjelma, jota voidaan käyttää etäyhteyden ottamiseen laitteelta toiselle. Sen sanotaan tarjoavan turvallinen yhteys kahden toisillensa tuntemattoman epäluotettavan laitteen välille epäluotettavassa verkossa. (Beck, ym. 2013a.)

SSH on hyvin käytännöllinen ja turvallinen vaihtoehto tekstikäyttöliittymäpohjaisten laitteiden, kuten palvelinkoneiden etähallintaan. Henkilökohtaisesti en ole kokeillut onnistuuko SSH:n avulla välittää livekuvaa, mutta muutamien lähteiden tutkinnan perusteella voisi sanoa, että ei se ole mahdotontakaan (Wellesley College 2019; Salmela 2026).

## 2.2 5G

5G tulee sanoista fifth generation eli viides sukupolvi, ja se on nykypäivänä maailmanlaajuisen langattomien verkkojen uusinta teknologiaa. 5G:n ja 4G:n eroja on muun muassa arvioitu enimmäistiedonsiirtonopeus, joka on 5G:llä noin kymmenkertainen 4G:hen verrattuna. 5G:n arvioitu enimmäisnopeus on jopa kymmenen gigabittiä sekunnissa, kun taas 4G-teknologialla enimmäisnopeus on noin yksi gigabitti sekunnissa. (Takko 2025.)

5G tuo mukanaan lukuisia parannuksia langattoman verkon nopeuteen ja luotettavuuteen liittyen. Näihin kuuluu muun muassa CC- ja URLLC-palvelut, jotka tulevat esille esimerkiksi prosessiautomaation yhteydessä, jossa vaaditaan 99,9999:n prosentin vahvuista luotettavuusprosenttia, 50:n millisekunnin päästä päähän viivettä sekä tiedonsiirtonopeutta, joka ylittää sataan megabittiin sekunnissa. (Sultan 2022.)

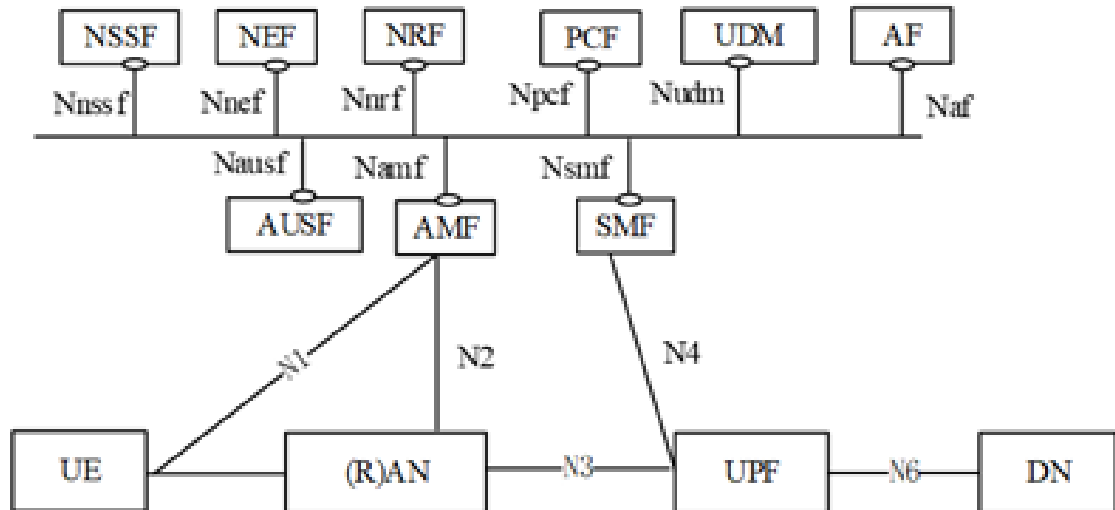
Uusien 5G-ominaisuuksien lisäksi 5G-järjestelmä tukee lähes kaikkia 4G LTE-ominaisuuksia. 5G:ltä löytyy myös EPC-tuki 4G-verkon kanssa, kuten edeltäjänsä 4G:lläkin on 2G-, 3G-, WLAN-, WiMax-, Ethernet-, DSL- ja kaapeliverkkojen välillä. (Sultan 2022; Awati 2024.)

5G-järjestelmän peruskaava noudattaa samaa mallia edeltäjiensä kanssa. Kuten kuviossa 2 nähdään, siihen kuuluu matkaviestintälaitteesta (MS) ja USIMistä muodostuva käyttäjän laite, gNB:n sisältävä uuden sukupolven radioliityntäverkko (NG-RAN) sekä 5G-ydinverkko (5GC), jonka sisällä AMF- ja UPF-funktiot toimivat. (Sultan 2022.)



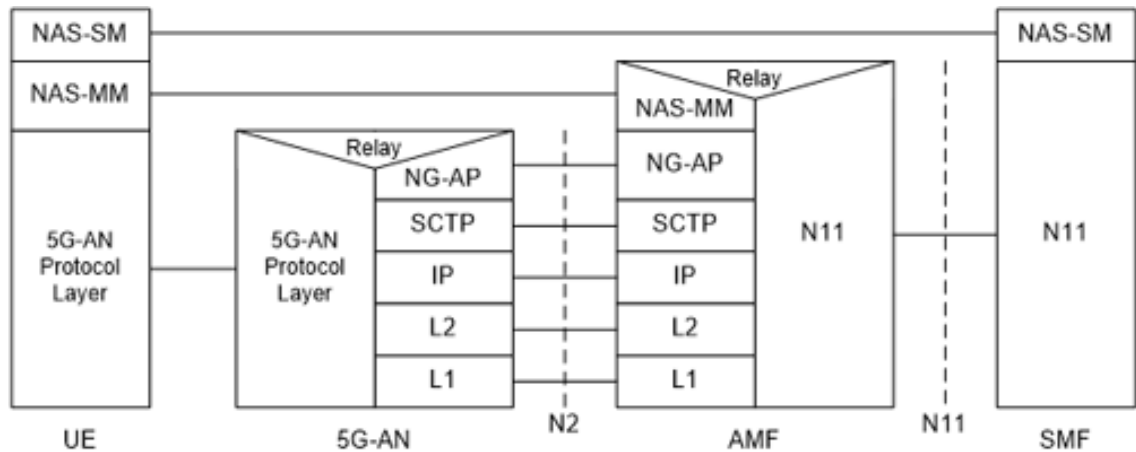
Kuvio 2. 5G-järjestelmän yleiskatsaus (Sultan 2022)

Kuten NB, myös UEn ja NG-RANin välisen radioliitännän termin NR-Uu jälkimmäinen osa periytyy aiemmilla sukupolvilla käytetyistä termeistä (Sultan 2022). Uu-lyhenne tarkoittaa radioliityntää UE:n ja UTRANin välillä (Bhatti, A., Haider, M. & Kirmani, A. 2007, 1). NG-RANin ja 5GC:n välinen viitepiste NG muodostuu useista rajapinnoista, yleensä N2- sekä N3-rajapinnoista, jotka nähdään kuviossa 3 (Sultan 2022).



Kuvio 3. 5GS-arkkitehtuuri (Sultan 2022)

5GC-arkkitehtuuri perustuu palvelupohjaiseen arkkitehtuuriin (SBA), jossa elementeistä käytetään verkkoelementti-nimityksen (NE) sijaan nimitystä verkkotoiminto (NF). Kuviossa 3 pohjalla nähtävät verkkotoiminnot UE, RAN, UPF ja DN kuuluvat käyttäjätasoon (*User Plane*), kuvion ylemmässä osassa nähdään vastaavasti *signalling plane*lle oleelliset verkkotoiminnot. Kuviossa 4 nähdään ohjaustason protokollapino (*protocol stack*). Ohjaustaso muodostuu välille UE-AMF & UE-SMF. (Sultan 2022.)



Kuvio 4. Ohjaustason protokollapino UE:n, 5G-AN:n, AMF:n ja SMF:n välillä (Sultan 2022)

Kun UE tekee yhteydenottoyrityksen verkkoon, AMF auktorisoi pääsyn ja seuraa, mille tukiasemalle UE ottaa yhteyden. SMF allokoii IP-osoitteen jokaiselle AMF:n auktorisoimalle UE:lle ja kommunikoi UPF:n kanssa ylläpitääkseen laitekohtaisia sessioita. (Davie, B., Peterson, L. & Sunay, O. 2023.)

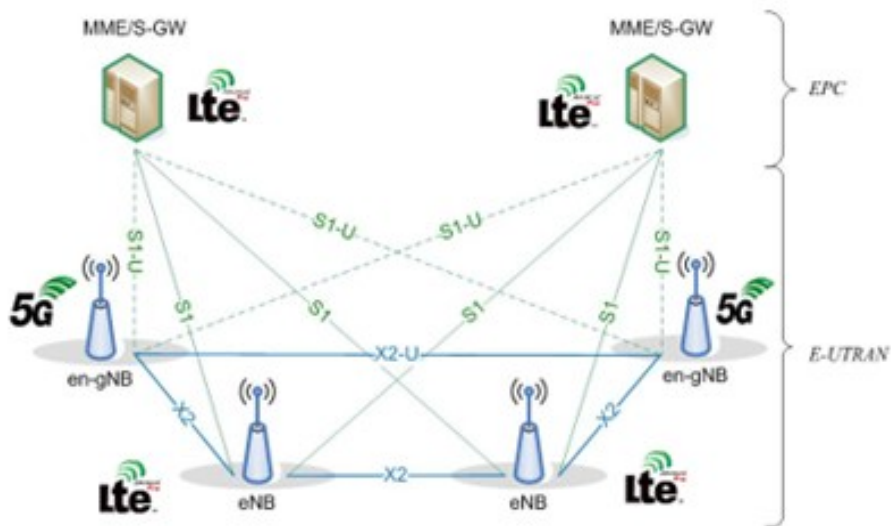
### 2.3 Core

Lapin ammattikorkeakoulun kampuksen verkon coren ohjaustaso sijaitsee Nokian datakeskuksella Espoossa ja muut komponentit, kuten reunalaskenta- ja UPF-palvelimet, on sijoitettu Rovaniemen kampukselle (Suutari-Jääskö 2025, 9, 12). Tämän opinnäytetyön yhteydessä rakennettavan verkon kaikki komponentit sijaitsevat Rovaniemen kampuksella. Tällaisen järjestelmän integroiminen esimerkiksi osaksi Lapin ammattikorkeakoulun Off-Grid 5G Mobile Network -järjestelmän peräkäräyksi mahdollistaisi täysin itsenäisesti toimivan verkon pystytyksen ilman yhteyttä ulkoiseen datakeskukseen.

Tätä rakenne-eroa ei pidä sekoittaa NSA- ja SA-verkkojen arkkitehtuurieroihin. Lapin ammattikorkeakoulun 5G-verkko on arkkitehtuuriltaan yhä kuvion 6 (s. 17) mukainen SA-verkko. Verkon core on hajautettu niin, että käyttäjätaso toimii paikallisesti ja ohjaustaso sijaitsee Espoossa. (Karjalainen 2026.)

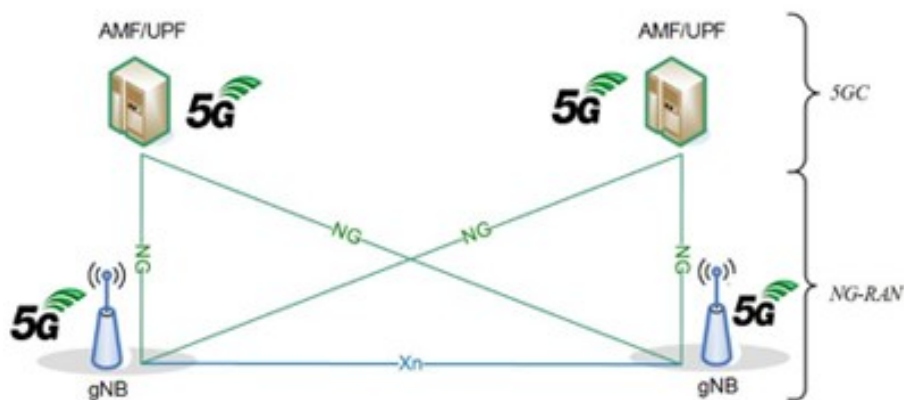
## 2.4 NSA- ja SA-verkot

Kuten kuviossa 5 on kuvattu, NSA-arkkitehtuurissa verkossa 5G RAN ja NR ovat sidoksissa 4G LTE- ja EPC-infrastruktuurin ydinverkkoon. Tämä konfiguraatio mahdollistaa 5G:n nopeuden hyödyntämisen 4G-verkossa. (Sultan 2022.)



Kuvio 5. NSA-arkkitehtuuri (Sultan 2022)

SA-verkko taas ei toimiakseen tarvitse enää ollenkaan 4G-komponentteja, kuten kuviossa 6 nähdään. 4G-core on korvattu 5G-corella, jolloin myös AMF korvaa MME:n. 4G:n MME ja 5G:n AMF toimittavat jokseenkin samaa roolia kussakin verkossa, mutta eivät ole aivan identtisiä keskenään. (Sultan 2022.)



Kuvio 6. SA-arkkitehtuuri (Sultan 2022)

## 2.5 Privaattiverkot

Privaattiverkolla voidaan tarkoittaa esimerkiksi organisaation sisäistä verkkoa, joka ei välttämättä ole yhteydessä ulkoisiin verkkoihin. Tällaisia organisaatioita voivat olla esimerkiksi tehtaat, sairaalat, viranomaiskohteet tai muunlaiset organisaatiot, joissa sisäisellä televiestinnällä on merkittävä rooli. Privaattiverkko on organisaation tietoturvan kannalta turvallisempi vaihtoehto julkiseen verkkoon verrattuna. (Karjalainen 2026.)

Privaattiverkon, kuten muidenkin verkkojen sisällä voidaan tietoliikennettä priorisoida tärkeysasteen mukaan. Esimerkiksi vaaratiedotteet ja hätäpuhelut voisivat hyvin olla prioriteettilistan kärjessä, jolloin tällaisen viestin lähtiessä liikkeelle verkossa hidastetaan muuta verkkoliikennettä tarpeen mukaan varmistaen vaaratiedotteen tai hätäpuhelun läpipääsy. Kriisitilanteissa voidaan myös kaupallisten verkkojen, kuten Elisan kanssa soveltaa tällaista käytäntöä varaten tärkeälle verkkoliikenteelle kapasiteettia, jolloin yksityiskäyttäjien verkkokapasiteettia saatetaan joutua pudottamaan. (Karjalainen 2026.)

### 3 COREN ASENNUS

#### 3.1 NUC-koneen vertailu coren laitteistovaatimuksiin

Käytettävän laitteiston tulee täyttää valmistajan määrittämät laitteistovaatimukset. Taulukko 1 toimii vertailuna valmistajan määrittämien 4G- ja 5G-core-palvelinkoneiden laitteistovaatimusten sekä isäntäkoneena toimivan NUC 13 -minitietokoneen teknisten tietojen välillä. Verkkokorteista käytetään taulukoissa termiä ”NIC”.

Taulukko 1. Coren valmistajan laitteistovaatimukset ja käytetyn laitteiston ominaisuuksien vertailu (Larabel 2021; ASUSTeK Computer Inc. 2023; Cumucore 2023a, 8; 2023b, 10; Jimm’s PC-Store Oy 2025)

	4G:	5G:	NUC:
Ubuntu versio:	22.04 Server LTS 6.5	22.04 Server	22.04
Prossessorin AVX tuki:	Vaadittu		AVX2
RAM (GB):	>= 4	>=8	32
Levytila (GB) ja tyyppi:	100 HDD	100 HDD	1000 SSD
Ytimet:	4×64bit	4×64bit	16×64bit
Verkkokortti:	Nopea XDP-yhteensopivat ajurit	3×NIC (10Gbps for 5G)	Ks. taulukko 2

Taulukossa 2 nähdään tarkemmat tiedot NUC 13 -minitietokoneeseen asennetuista verkkokorteista. Cumucore-dokumentaation mukaan (Cumucore 2023a, 9; 2023b, 10, 11) kolme verkkoliitäntää on hyvä olla, mutta kahdellakin pärjää tarvittaessa.

Taulukko 2. NUC 13 -minitietokoneen verkkokorttien tekniset tiedot (Intel Corporation 2022a; 2022b; Marvell 2022)

	Malli:	Liitäntä:	Kapasiteetti:	Kellotaajuus:
NIC1:	Intel I226-V (XDP tuki löytyy)	Ethernet	1Gb	33MHz
NIC2:	Intel Killer Wi-Fi 6E AX1690	Langaton	~3Gb	33MHz
NIC3:	Marvell AQC113	Ethernet	10Gb	33MHz

Taulukko 3 havainnollistaa käytettävissä olevan levytilan sekä RAM-muistin jalkoa. Isäntäkoneen kattavat resurssit näiden parametrien suhteen mahdollistavat

minimivaatimuksien ylittävän muistin ja levytilan allokoinnin 4G- ja 5G-virtuaalikoneille. Näin varmistetaan virtuaalikoneiden jouheva toiminta.

Taulukko 3. Levytilan ja RAM-muistin jako

	4G:	5G:	NUC:
RAM (GB):	8	12	12
Levytila (GB):	200	200	600

## 3.2 Isäntäkoneen Ubuntu 22.04 LTS -käyttöjärjestelmän asennus

### 3.2.1 Valmistelut

Ennen Ubuntu-käyttöjärjestelmän asennusta isäntäkoneelle käytettävä osio täytyi formatoida. Tässä vaiheessa on yleisesti hyvä muistaa, että mikäli osiolla on jotain, mitä täytyy ottaa talteen, täytyy se tehdä ennen osion formatointia.

Pelkkä käyttöjärjestelmän asennus ei pyyhi levyä tai levyille luotua osiota täysin tyhjäksi. Tämä saattaa johtaa erinäisiin virhetilanteisiin, esimerkiksi sellaiseen, jossa asennusprosessin aikana luodulla käyttäjällä ei päässyt kirjautumaan sisään.

### 3.2.2 Asennusprosessi

Aluksi kirjoitettiin (*flash*) Ubuntun sivuilta (Ubuntu 2024) löytyvä Ubuntu 22.04 server -käyttöjärjestelmä Balena Etcher -ohjelmaa (Balena 2025) käyttäen USB-muistitikulle. Kun kirjoitus oli valmis, kytkettiin NUC-koneesta virta pois, asetettiin juuri kirjoitettu muistitikku NUC-koneen USB-porttiin ja käynnistettiin kone.

Asennus oli melko suoraviivainen, mutta muutama asia täytyy ottaa huomioon:

- Asennettaessa täytyi valita vaihtoehto "Ubuntu server with HWE kernel"
- kohdassa "Guided storage configuration" täytyi tässä tapauksessa valita "Custom storage layout", sillä asennuksessa käytettiin koko levyn sijasta levyn osiota

- "Custom storage layout" -valikosta valittiin, mille osiolle käyttöjärjestelmä haluttiin asentaa. "Mount to" -kohdasta avattiin tietue "Edit" → "Other" ja jätettiin kohta sellaisekseen, jolloin siinä näkyi pelkästään vinoviiva eli "/"-merkki.

Mikäli käytettävät SSH-avaimet olisivat jo generoituna, voitaisiin ne tässä vaiheessa syöttää järjestelmälle, jolloin pelkällä salasanalla autentikointikin voitaisiin ottaa jo tässä vaiheessa pois käytöstä. Pelkällä salasanalla autentikointi jätettiin kuitenkin vielä päälle, sillä SSH-avainten konfigurointi tehdään jälkeinpäin. Käyttöjärjestelmä oli tämän jälkeen valmis käytettäväksi.

### 3.3 Virtuaalikoneet

#### 3.3.1 Valmistelu

Laitteen yhteensopivuus virtuaalikoneohjelmistojen osalta oli syytä varmistaa, ennen kuin jatkettiin eteenpäin. Suoritettiin isäntäkoneella komento `"egrep -c '(vmx|svm)' /proc/cpuinfo"`. Jos komento antaa vastaukseksi numeron 1 tai suuremman, se tarkoittaa, että prosessorilla on virtualisointituki. Komennon tuloste indikoi, että laitteella on virtualisointituki. Komennolla `"sudo kvm-ok"` tarkastettiin vielä, että prosessori tukee KVM-laitteistokiihdytystä. Vastaukseksi saatiin `"INFO /dev/kvm exists \ KVM acceleration can be used"` tarkoittaen siis, että KVM-laitteistokiihdytystä voidaan käyttää. (Aleksic 2024; DeepWiki 2025.)

Tilanteessa, jossa komento `"sudo kvm-ok"` palauttaa virheviestin, täytyy laitteelle asentaa CPU-Checker-ohjelma. Ohjelman asennuksen jälkeen voidaan ajaa `"sudo kvm-ok"` uudelleen. (Aleksic 2024.)

**ISO-kuvan** virtuaalikoneita varten voi ladata suoraan Ubuntun omilta sivuilta Linuxin `"wget"`-komennolla. Virtuaalikoneissa käytettävät ISO-kuvat suositellaan lataamaan `"/var/lib/libvirt/images"`-hakemistoon. Siirryttiin siis komennolla `"cd /var/lib/libvirt/images"` haluttuun hakemistoon ja ladattiin ISO-kuva komennolla `"sudo wget https://releases.ubuntu.com/jammy/ubuntu-22.04.5-live-server-amd64.iso"`. (Korolev 2023.)

Jotta Virtual Machine Manager voi käyttää ladattua ISO-kuvaa virtuaalikoneen käyttöjärjestelmän asennukseen, se täytyy ensin asettaa loop-laitteeksi. Aloitettiin *losetup*-komennolla, joka näyttää käytössä olevat loop-laitteet (Canonical 2025a). Kuviossa 7 nähdään, että silmukat (*loop*) 1–5 olivat käytössä, joten silmukka numero 6 oli seuraava käytettävissä oleva vaihtoehto.

```

agranucserver@agranucserver:~$ losetup
NAME          SIZELIMIT OFFSET  AUTOCLEAR RO  BACK-FILE                                  DIO LOG-SEC
/dev/loop1    0         0       1 1 /var/lib/snapd/snaps/core20_2582.snap     0   512
/dev/loop4    0         0       1 1 /var/lib/snapd/snaps/snapd_21759.snap    0   512
/dev/loop2    0         0       1 1 /var/lib/snapd/snaps/lxd_29351.snap      0   512
/dev/loop0    0         0       1 1 /var/lib/snapd/snaps/core20_2318.snap    0   512
/dev/loop5    0         0       1 1 /var/lib/snapd/snaps/snapd_24505.snap    0   512
/dev/loop3    0         0       1 1 /var/lib/snapd/snaps/lxd_31333.snap      0   512

```

Kuvio 7. Loop-laitteet

Siirryttiin komennolla *cd /var/lib/libvirt/images* hakemistoon, jossa ISO-kuva sijaisi. Tämän jälkeen ISO-kuva asetettiin silmukkaan numero 6 komennolla *sudo losetup /dev/loop6 ubuntu-22.04.5-live-server-amd64.iso* (Bruden 2025). Tarkastettiin vielä lopuksi *losetup*-komentoa käyttäen, että ISO-kuva näkyy silmukassa numero 6, kuten kuviossa 8 nähdään.

```

agranucserver@agranucserver:/var/lib/libvirt/images$ losetup
NAME          SIZELIMIT OFFSET  AUTOCLEAR RO  BACK-FILE                                  DIO LOG-SEC
/dev/loop1    0         0       1 1 /var/lib/snapd/snaps/core20_2582.snap     0   512
/dev/loop6    0         0       0 0 /var/lib/libvirt/images/ubuntu-22.04.5-live-server-amd64.iso 0   512
/dev/loop4    0         0       1 1 /var/lib/snapd/snaps/snapd_21759.snap    0   512
/dev/loop2    0         0       1 1 /var/lib/snapd/snaps/lxd_29351.snap      0   512
/dev/loop0    0         0       1 1 /var/lib/snapd/snaps/core20_2318.snap    0   512
/dev/loop5    0         0       1 1 /var/lib/snapd/snaps/snapd_24505.snap    0   512
/dev/loop3    0         0       1 1 /var/lib/snapd/snaps/lxd_31333.snap      0   512

```

Kuvio 8. ISO-kuva loop-laitteena

Tämän jälkeen voitiin suorittaa myös kiinnitys eli mount-toiminto loop-laitteeksi asetetulle ISO-kuvalle komennolla *sudo mount /var/lib/libvirt/images/ubuntu-22.04.5-live-server-amd64.iso /mnt*. Komento kiinnittää ISO-kuvan *mnt*-hakemistoon ja näin ollen mahdollistaa ISO-kuvan sisällön tutkimisen ja päästiin varmistamaan, että *casper*-hakemistosta löytyy tarvittavat tiedostot *vmlinuz* ja *initrd*. Tiedostot ovat oleellisia onnistuneen asennuksen kannalta.

Tarkastuksen jälkeen voitiin irrottaa ISO-kuvan kiinnitys komennolla *sudo umount /mnt*. ISO-kuva on valmis käytettäväksi virtuaalikoneen luonnin yhteydessä.

**Vmlinuz**-nimitys tulee termistä ”Virtual Memory Linux gZip” tarkoittaen, että tiedosto on Gzip-työkalulla pakattu Linux *kernel* (Bellavue Linux 2005; Free Software Foundation 2020). *Kernel* toimii keskeisenä komponenttina tietokoneen laitteiston ja prosessien välillä (Red Hat 2019).

**Initrd**-lyhenne tulee sanoista ”Initial Ram Disk”. Initrd tarjoaa väliaikaisen juuritiedostojärjestelmän sekä sisältää *kernelin* käynnistyksen kannalta oleelliset tiedostot, moduulit ja ajurit. Suoritettuaan tarvittavat toimenpiteet *kernel* voi jatkaa käynnistystä kiinnittäen varsinaisen juuritiedostojärjestelmän ja irrottaen initrdin luoman väliaikaisjuuritiedostojärjestelmän. (Almesberger & Lermen 2000; Adjei 2023.)

### 3.3.2 KVM-ohjelmistojen asennus ja virtuaalikoneen luonti

Tämä luku sisältää tarvittavien virtuaalikoneohjelmistojen asennuksen, käyttäjän lisäämisen virtuaalikoneiden käyttäjäryhmälle ja lopuksi itse virtuaalikoneen luomisen. Ensiksi asennettiin QEMU:n toimittama KVM ohjelmisto, libvirt taustaprosessi- ja hallintapaketit sekä Linux ethernet -siltapaketti komennolla ”*sudo apt install qemu-kvm libvirt-daemon-system libvirt-clients bridge-utils -y*” (Aleksic 2024; SPI Inc. 2026a; 2026b; 2026c).

Seuraavaksi lisättiin käyttäjä ”*agranucserver*” ryhmiin ”*libvirt*” ja ”*kvm*” komendoilla ”*sudo adduser 'agranucserver' libvirt*” ja ”*sudo adduser 'agranucserver' kvm*”. Vain näiden ryhmien jäsenillä on pääsy virtuaalikoneisiin, joten ainakin yksi käyttäjä täytyy lisätä (Aleksic 2024). Näillä käyttäjillä ei ole tekemistä virtuaalikoneilla luotujen käyttäjien kanssa, vaan virtuaalikoneilla on omat koneen sisäiset käyttäjänsä.

Komennolla ”*sudo systemctl status libvirtd*” varmistettiin, että libvirt on asennettu onnistuneesti. Komennon tuloste näytti tilaa ”*active (running)*”, joten libvirtin asennus on onnistunut ja palvelu on käynnissä. Mikäli tuloste näyttää, että palvelu ei ole käynnissä, voidaan se käynnistää komennolla ”*sudo systemctl enable --now libvirtd*”. (Aleksic 2024.)

Ennen virtuaalikoneen varsinaista luontia täytyi asentaa vielä Virtual Machine Manager, joka mahdollistaa varsinaisten virtuaalikoneiden luomisen ja niiden hallinnan. Asennus suoritettiin komennolla ”*sudo apt install virt-manager -y*”. (Aleksic 2024; Canonical Group Ltd. 2025.)

Seuraavaksi voitiin suorittaa itse virtuaalikoneen luominen. Luominen tapahtui yhdellä komennolla (Korolev 2023; Aleksic 2024; Canonical 2025c), joka nähdään kuviossa 9. Komennolla luodaan virtuaalikone, jolle asennetaan 4G core -paketti, joten asennuksen parametrien täytyi vastata taulukossa 1 (s. 19) määriteltyjä 4G-koneen laitteistovaatimuksia.

```

agranucserver@agranucserver:~/var/lib/libvirt/images$ sudo virt-install \
> --name=vm4g \
> --description='VM for running 4G core' \
> --ram=8192 \
> --vcpus=4 \
> --location /var/lib/libvirt/images/ubuntu-22.04.5-live-server-amd64.iso,kernel=casper/vmlinuz,initrd=casper/initrd \
> --disk path=/var/lib/libvirt/images/CoreVM/vm4g.qcow2,size=200 \
> --graphics none \
> --os-variant ubuntu22.04 \
> --extra-args='console=ttyS0,115200n8 serial'|

```

Kuvio 9. Virtuaalikoneen luomiskomento

Komennon antamisen jälkeen odoteltiin hetki, kun järjestelmä loi virtuaalikonetta. Kun luomisprosessi on valmis, Ubuntun asennusnäkyvä avautuu näytölle. Ubuntun asennus virtuaalikoneelle suoritettiin samoin kuin luvussa 3.2.2 on ohjeistettu.

Ubuntun virtuaalikoneelle asennuksen jälkeen näytölle ilmaantui viesti ”*Failed unmounting cdrom*”. Tämä aiheutti hieman hämmennystä, mutta poistui painamalla enter-näppäintä. Virtuaalikone on tämän jälkeen käyttövalmis.

Seuraavaksi luotiin 5G-virtuaalikone 4G-virtuaalikoneen rinnalle. 5G-virtuaalikoneen luonti tapahtui kuvion 9 komentoa mukailleen, mutta parametrit täytyi muokata vastaamaan taulukossa 1 (s. 19) kuvattuja 5G-koneen laitteistovaatimuksia.

Ubuntua asennettaessa virtuaalikoneelle näytölle ilmaantui kysymys, että jatkaanko *basic modella* vai vaihdetaanko *rich modeen*. 4G-virtuaalikoneen luonnin yhteydessä valitsin ”Switch to rich mode” ja 5G-virtuaalikoneen luonnin yhteydessä valitsin ”Continue with basic mode”. En huomannut juurikaan eroa näiden kahden käyttöliittymän välillä.

**RAM-muistin ja levytilan allokoinnissa** täytyy ottaa huomioon tiettyjä asioita. Kuviossa 9 RAM-muistin allokointia indikoiva "--ram"-parametri ei ole samalla tavalla kokonaisluku, kuten "--disk"-parametrin levytilan allokointi, jonne syötetään haluttu arvo gigatavuina. Aleksicin (2024) ja Korolevin (2023) esimerkkikuvissa nähdään RAM-muistin arvoina 961 ja 2048. Esimerkeistä päätettiin, että RAM-muistin arvo täytyy mahdollisesti syöttää binäärille omaisena 2-järjestelmän kantelukuna, vaikka luku 961 ei siihen lukeudukaan, mutta 2048 lukeutuu. Asia varmistui Ubuntun virallisesta manuaalista, jossa todetaan, että arvo täytyy syöttää MiB-arvona eli mebitavuina (Canonical 2025c) sekä TechTargetin artikkelista, jossa todetaan, että gibitavu on 1024 MiB (Peterson 2023).

Taulukosta 1 (s. 19) nähdään, että 4G-kone tarvitsee vähintään neljä gigatavua ja 5G-kone vähintään kahdeksan gigatavua RAM-muistia. Isäntäkoneen riittävä RAM-muistin määrä mahdollisti sen, että virtuaalikoneille voitiin allokoida enemmänkin RAM-muistia.

4G-koneelle allokointiin kahdeksan gigatavua eli noin 8000 MiB ja 5G-koneelle 12 gigatavua eli noin 12 000 MiB. Tarvittavat MiB-arvot kunkin koneen RAM-muistia määrittävälle parametrille voidaan laskea kaavan 1 (GTM 2021) mukaan, jolloin kahdeksan gibitavua vastaa 8192 mebitavua ja 12 gibitavua vastaa 12 288 mebitavua.

$$MiB = GiB \times 2^{10} \tag{1}$$

missä

*MiB* on mebitavumäärä, joka halutaan saada selville  
*GiB* on gibitavumäärä, josta mebitavumäärä lasketaan.

Samasta lähteestä löytyy myös kaava 2 (GTM 2021), jolla sivuston mukaan voidaan laskea MiB-arvo GB-arvosta. Kaava 2 vaikuttaa teoreettisesti korrektimmalta, mutta kaavalla 1 lasketut arvot vastasivat lähemmin aiemmin tarkastelemiäni esimerkkien "--memory"-parametrin arvoja (Heimes 2020; Korolev 2023), joten jatkoin tällä perusteella kaavan 1 käyttöä. Kaavan 2 mukaan kahdeksan gigatavun arvoksi olisi tullut noin 7629 mebitavua ja 12 gigatavun arvoksi noin 11 444 mebitavua.

$$MiB = GB \times \frac{10^9}{2^{20}} \quad (2)$$

missä

*MiB* on mebitavumäärä, joka halutaan saada selville  
*GB* on gigatavumäärä, josta mebitavumäärä lasketaan.

Myönnettäköön, että mega- ja mebitavuista kirjoittaessa on saattanut tulla sekaannuksia tavu- ja bitti-termien kesken, sillä tietoa haettiin merkittyjen englanninkielisten lähteiden lisäksi myös suomenkielisestä lähteestä Lenovon sivustolta (Lenovo 2026). IEEE:n standardissa 1541–2002 (2009, 1) mainitaan, että *byte* eli tavu merkitään isolla B-kirjaimella. W3Schools (2026) sivuston mukaan pientä b-kirjainta käytetään kuvaamaan bittiiä, mutta sitä ei tietääkseni ole standardeissa määritely, vaikka se yleisessä käytössä onkin.

### 3.4 Core-ohjelmiston asennus asennusskriptillä

#### 3.4.1 USB-laitteen kytkentä fyysisen koneen kautta virtuaalikoneelle

Cumucoren toimittama core-ohjelmisto asennettiin USB-muistitikulta virtuaalikoneelle. Isäntäkoneeseen kytketyt USB-oheislaitteet, lukuun ottamatta hiirtä ja näppäimistöä, eivät oletuksena ole näkyviä virtuaalikoneille, vaan niille täytyy suorittaa USB-läpikulkutoiminto (*USB-passthrough*) (KVM contributors 2010).

Prosessia jatkettiin isäntäkoneen puolella. Ensimmäisenä tarkistettiin käytettävän USB-laitteen ”vendor ID”- ja ”product ID”-tietueet komennolla ”*lsusb*”. Kuviossa 10 nähdään yksi ainoa Kingston-valmistajan laite, joten voitiin päätellä, että se on äsken isäntäkoneeseen asetettu Kingstonin muistitikku. Samalla rivillä ”ID”-tagin jälkeen näkyy merkintä 0951:1666. Tämä tarkoittaa, että laitteen vendor ID on 0951 ja product ID on 1666. Näitä tietueita tarvitaan myöhemmin suoritettavan USB-läpikulkutoiminnon suorittamiseksi. (Wragg 2009.)

```

agranucserver@agranucserver:~$ lsusb
Bus 004 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 003 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 002 Device 003: ID 0951:1666 Kingston Technology DataTraveler 100 G3/G4/SE9 G2/50
Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 001 Device 005: ID 0bda:4c60 Realtek Semiconductor Corp. USB Audio
Bus 001 Device 004: ID 067b:23a3 Prolific Technology, Inc. USB-Serial Controller
Bus 001 Device 003: ID 0461:4e22 Primax Electronics, Ltd Dell Mouse, 2 Buttons, Modell: MS111-P
Bus 001 Device 002: ID 413c:2113 Dell Computer Corp. KB216 Wired Keyboard
Bus 001 Device 006: ID 8087:0033 Intel Corp.
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub

```

Kuvio 10. ”*lsusb*”-komennon tuloste

Seuraavissa vaiheissa käydään prosessia läpi 4G-virtuaalikoneen osalta. 5G-virtuaalikoneelle tehtävät toimenpiteet ovat lähes suoraan verrannolliset 4G-virtuaalikoneelle tehtäviin toimenpiteisiin, eikä niitä täten käydä erikseen läpi.

Jatkettiin siis isäntäkoneen puolella eteenpäin sammuttamalla virtuaalikone komennolla `"virsh shutdown vm4g"`. Virtuaalikoneen onnistunut sammutus voidaan hetken päästä tarkistaa komennolla `"virsh list --all"`. Komennon tuloste näyttää kaikki käynnissä sekä sammuksissa olevat virtuaalikoneet. Virtuaalikoneen nimen oikealla puolen lukee joko `"running"` tai `"shut off"`, joka indikoi onko virtuaalikone käynnissä vai ei.

Terminaaliin annettiin komento `"virsh edit vm4g"`, jolla päästään käsiksi virtuaalikoneen domain.xml-tiedostoon. Domain.xml-tiedosto sisältää perustavat määritelmät virtuaalikoneen toiminnalle mukaan lukien tiedon siitä, mitä *hypervisoria* virtuaalikone käyttää toimiakseen, virtuaalikoneen nimen sekä -ID:n ja mitä oheislaitteita virtuaalikoneelle määritellään näkyväksi, mukaan lukien USB-muistikut. Domain.xml-tiedostoon lisättiin alla olevassa esimerkissä turkoosilla korostetut rivit `<domain>`-lohkon sisällä olevan `<devices>`-lohkon sisään. (Wragg 2009; Libvirt 2026a.)

```
<domain type='kvm'>
  <name>vm4g</name>
  ...
  <devices>
    ...
    <hostdev mode='subsystem' type='usb'>
      <source>
        <vendor id='0x0951'>
        <product id='0x1666'>
      </source>
    </hostdev>
  </devices>
</domain>
```

Kun muutokset oli tallennettu, käynnistettiin virtuaalikone komennolla `"virsh start vm4g --console"`. Käytettävä parametri `"--console"` avaa samalla käynnistettävän virtuaalikoneen komentoliittymän. Tässä kohdassa vaihdettiin siis isäntäkoneesta virtuaalikoneeseen, joten seuraavat vaiheet suoritettiin vm4g-virtuaalikoneella.

Komennolla *"lsblk"* voidaan tarkistaa, että *domain.xml*-tiedostoon lisätty USB-laite näkyy virtuaalikoneelle. Tuloste näytti, että USB-laite on kytketty *"/dev/"*-hakemistoon nimikkeellä *"sda1"*. (Broz & Zak 2017.)

### 3.4.2 Asennusskriptin ajaminen

Kun USB-laite oli saatu näkyväksi virtuaalikoneelle, voitiin sille suorittaa kiinnitystoiminto, joka avaa pääsyn USB-laitteen sisältöön. USB-laite kiinnitettiin *"media"*-hakemiston sisälle luotuun *"usb-drive"*-hakemistoon komennolla *"sudo mount /dev/sda1 /media/usb-drive"*. (Rendek 2025.)

Ennen tiedostojen kopiointia, virtuaalikoneen *"home"*-hakemistoon luotiin tarvittaville tiedostoille uusi hakemisto komennolla *"mkdir corefiles"*. Tarvittavat tiedostot kopioitiin USB-laitteelta virtuaalikoneelle Linuxin *"cp"*-komentoa hyödyntäen. Kopioinnin onnistuminen varmistettiin vielä menemällä *"corefiles"*-hakemistoon ja silmäilemällä tiedostoja varmentuen, että kaikki tarvittavat tiedostot löytyvät hakemistosta. Kopioinnin ollessa valmis voitiin USB-laite irrottaa kiinnityksestä komennolla *"sudo umount /dev/sda1"*. Tämän jälkeen USB-laite voidaan myös fyysisesti irrottaa laitteesta.

Tässä vaiheessa siirryttiin takaisin isäntäkoneen puolelle, sillä virtuaalikoneen *domain.xml*-tiedostosta täytyi käydä poistamassa sinne luvussa 3.4.1 konfiguroitu USB-laite. Mikäli tämä vaihe jää suorittamatta, virtuaalikone ei käynnisty ilman, että tämä USB-laite on kytkettynä isäntäkoneeseen, sillä virtuaalikone yrittää hakea USB-laitetta käynnistyessään, eikä sen poissa ollessa voi käynnistyä.

Toistettiin luvussa 3.4.1 ohjeistettu virtuaalikoneen sammutus ja *domain.xml*-tiedoston avaaminen. Poistettiin turkoosilla maalatut rivit, tallennettiin muutokset ja käynnistettiin virtuaalikone uudelleen.

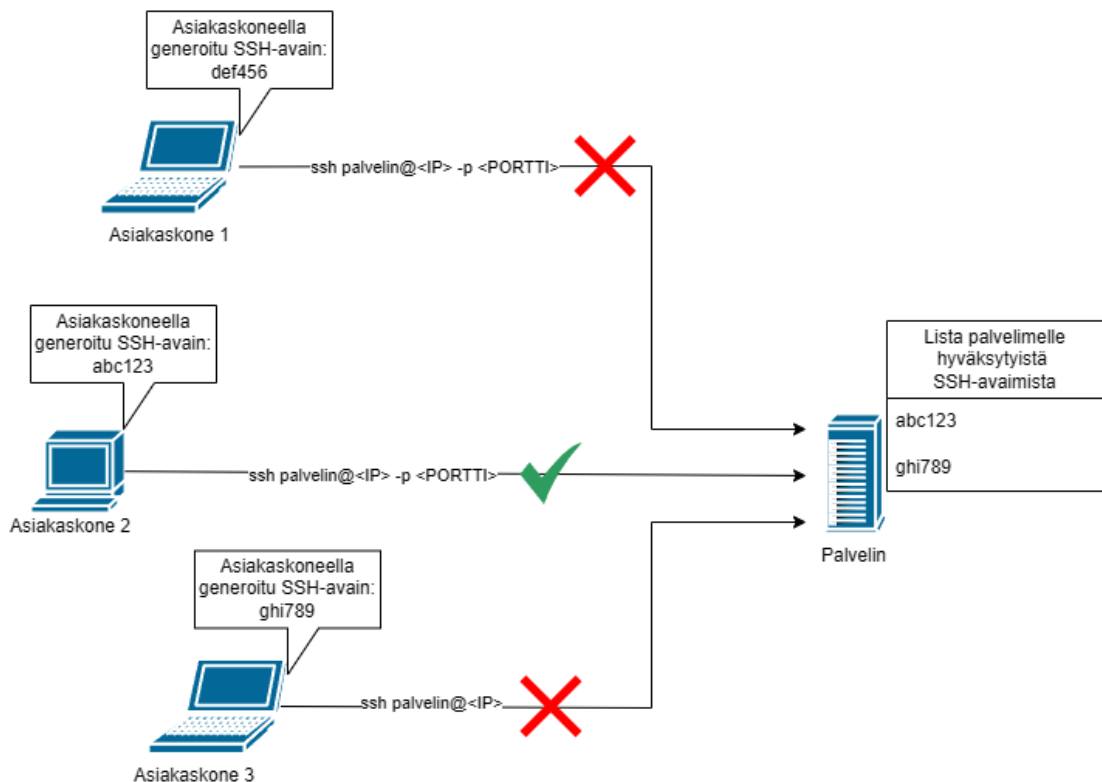
Prosessia jatkettiin virtuaalikoneen puolella tekemällä aiemmin *"corefiles"*-hakemistoon kopioidusta shell-skriptistä suoritettava eli *executable* *"sudo chmod -x"*-komentoa hyödyntäen. Toimenpiteen jälkeen skripti ajettiin *"sudo bash"*-komentoa käyttäen. Komentojen loppuun lisätään shell-skriptin tiedostonimi *".sh"*-päätteinen. (Levine 2025.) Skriptin ajon jälkeen virtuaalikone käynnistyi uudelleen.

## 4 YLLÄPITO

### 4.1 SSH-yhteyksien konfigurointi

Tässä vaiheessa prosessia täytyy OpenSSH server olla asennettuna. Yleensä se asennetaan jo Ubuntu-käyttöjärjestelmän asennuksen yhteydessä, mutta mikäli se on jäänyt siinä vaiheessa tekemättä, voi sen asentaa jälkikäteenkin komennolla *"sudo apt install openssh-server"*.

Kuvio 11 kuvastaa, kuinka SSH-yhteydet toimivat konfiguraatioiden ollessa valmiit. Asiakaskoneen 1 SSH-avainta ei ole konfiguroitu palvelimelle, joten SSH-yhteydenottoyritys epäonnistuu, vaikka asiakaskone on määritellyt tarvittavan porttinumeron. Asiakaskoneen 2 SSH-avain on konfiguroitu palvelimelle ja se on yhteyttä muodostaessaan määrittänyt oikean porttinumeron, joten yhteydenottoyritys hyväksytään. Myös asiakaskoneen 3 SSH-avain on konfiguroitu palvelimelle, mutta se ei ole yhteydenottoyrityksen yhteydessä määrittänyt tarvittavaa porttinumeroa, joten yhteydenottoyritys epäonnistuu.



Kuvio 11. SSH:n toiminta palvelimen ja asiakaskoneiden välillä

#### 4.1.1 SSH-avainten generointi asiakaskoneella

Asiakaskoneesta puhuttaessa tarkoitetaan laitetta, jolta halutaan ottaa etäyhteys palvelimelle. Tässä tapauksessa virtuaalikone toimitti palvelinkoneen roolia ja henkilökohtaisella työpisteellä sijaitseva tietokone toimi asiakaskoneena. Tämän luvun toimenpiteet suoritettiin jokaisella asiakaskoneella, jolta haluttiin SSH-yhteys palvelimelle, sillä järjestelmän SSH-yhteydet konfiguroitiin kokonaisuudessaan niin, että vain asiakaskoneilta, joiden laitekohtainen SSH-avain on konfiguroitu palvelimen hyväksytyjen laitteiden luetteloon, on mahdollista ottaa SSH-yhteys palvelimelle.

Asiakaskoneelta avattiin komentokehote, ja sille luotiin SSH-avain komennolla *"ssh-keygen"*. Jos SSH-avain on jo aiemmin luotu samalla laitteella, tämä komento ylikirjoittaa vanhan avaimen. Jos näin on, niin kannattaa silloin käyttää jo olemassa olevaa avainta, jättää mainittu komento suorittamatta ja jatkaa seuraavaan kappaleeseen. Generointivaiheessa kysytään, haluaako käyttäjä luoda tunnuksaan SSH-kirjautumista varten. Tämä ei ole pakollinen, mutta tietoturvasyistä tunnuksana on suotavaa asettaa. (Garnett & Kurup 2022.)

Generoidut SSH-avaimet löytyvät hakemistosta *"~/.ssh"*. Siirryttiin hakemistoon komennolla *"cd ~/.ssh"* ja suoritettiin siellä komento *"type id\_ed25519.pub"*. Windowsin komentokehoteessa *"type"*-komento vastaa Linuxin *"cat"*-komentoa, joka tulostaa parametrina annetun tiedoston sisällön komentokehoteeseen. Annettu *"type"*-komento tulostaa siis tässä tapauksessa *"id\_ed25519.pub"*-tiedoston sisällön eli laitekohtaisen SSH-avaimen. (Garnett & Kurup 2022; Microsoft 2023.) Tiedoston sisältö kopioitiin leikepöydälle seuraavan luvun vaiheita varten.

#### 4.1.2 SSH-avainten konfigurointi palvelinkoneelle

Aloitettiin tarkistamalla, löytyykö palvelinkoneelta *".ssh"*-hakemistoa suorittamalla komento *"ls ~/.ssh"*. Vasta-asennetulla käyttöjärjestelmällä, jolle ei ole vielä konfiguroitu SSH-yhteyksiä, ei yleensä tätä hakemistoa vielä löydy. Jos hakemistoa ei löydy, voidaan se luoda komennolla *"mkdir -p ~/.ssh"*. (Garnett & Kurup 2022.) SSH-avaimen kopioimiseen asiakaskoneelta palvelinkoneelle on monta eri tapaa, joista työn aikana käytettiin kahta vaihtoehtoa.

**Manuaalinen kopiointi** tapahtuu seuraavasti. Annetaan palvelinkoneella komento "*sudo nano ~/.ssh/authorized\_keys*". Komento luo "ssh"-hakemiston sisään tiedoston "authorized\_keys" ja avaa tiedoston "GNU nano" -tekstieditorissa. Kirjoitetaan "id\_ed25519.pub"-tiedoston sisältö käsin tähän tiedostoon ja tallennetaan muutokset.

**Avaimen kopiointi SSH-yhteyden välityksellä** onnistuu vain, mikäli SSH-yhteyden pelkällä salasanalla autentikointi on sallittu. Luvussa 3.2.2 mainittiin, että pelkällä salasanalla autentikointi jätettiin toistaiseksi päälle, joten SSH-avaimen lisäys palvelinkoneelle onnistuu myös tällä metodilla. Metodi on muuten sama kuin manuaalinen kopiointi, mutta manuaalisen kirjoituksen sijaan otetaan SSH-yhteys asiakaskoneelta palvelinkoneelle ja liitetään kopioitu teksti leikepöydältä "authorized\_keys"-tiedostoon. (Garnett & Kurup 2022.)

#### 4.1.3 Autentikointimetodin konfigurointi

Tässä vaiheessa muutettiin SSH-autentikointi niin, että pelkällä salasanalla autentikointi ei enää ole mahdollista. Toimenpide vaati kolmen konfiguraatitiedoston muokkauksen.

Ensiksi avattiin tiedosto, jossa määritellään pelkällä salasanalla autentikointi. Suoritettiin komento "*sudo nano /etc/ssh/sshd\_config*", joka avaa "sshd\_config"-tekstitiedoston tekstieditorissa. Konfiguraatitiedostossa on kommentoituna tietue, "*PasswordAuthentication no*". (Garnett & Kurup 2022.)

Tietueiden ollessa kommentoituna konfiguraatio ohittaa nämä tietueet. Menetelmä on suoraan verrannollinen kommentointiin kooditiedostoissa muun muassa Pythonia tai C-pohjaista koodia kirjoittaessa.

Poistettiin rivin kommentiksi määrittävä ristikkomerkki (#) rivin alusta, jolloin konfiguraatietietue muuttuu aktiiviseksi. Tallennettiin muutokset ja poistettiin tekstieditorista. (Garnett & Kurup 2022.)

Seuraavana suoritettiin komento "*sudo nano /etc/cloud/cloud.cfg.d/99-disable-password-auth.cfg*", joka avaa tekstieditorissa tiedoston "99-disable-password-auth.cfg". Tiedostoon kirjoitettiin rivi "*ssh\_pwauth: 0*". Ensimmäisessä vaiheessa

muokattu "sshd\_config" tiedosto palautuu oletuksena oletusasetuksiin aina virtuaalikoneen käynnistyessä uudelleen. Tämän konfiguraatitiedoston tällaiseksi muokkaamisen johdosta "sshd\_config"-tiedosto ei enää palaudu oletusasetuksiin uudelleenkäynnistyksen yhteydessä. (Uusitalo 2025a.)

Kolmanneksi tarkastettiin "50-cloud-init.conf"-tiedosto "sshd\_config.d"-hakemistosta komennolla "*sudo nano /etc/ssh/sshd\_config.d/50-cloud-init.conf*". Mikäli tiedosto sisältää tekstirivin, jossa lukee "*PasswordAuthentication yes*", täytyy rivi asettaa kommentoiduksi asettamalla ristikkomerkki rivin ensimmäiseksi merkiksi. Tallennetaan vielä muutokset. (Alves 2022.)

Lopuksi suoritettiin vielä komento "*sudo sshd -T | grep -E 'passwordauthentication|pubkeyauthentication'*", joka näyttää autentikointimethodien tilan. "*Sshd*"-toiminto on OpenSSH-palvelun taustaprosessi, joka muun muassa kuuntelee käyttäjien välisiä SSH-yhteyksiä. "-T"-parametri määrittää sshd-komennon toimimaan testitilassa ja tulostaa konfiguraatioparametrit komentokehoteeseen. (Beck ym. 2013b.) Tämän jälkeen annettava pystyviiva toimii välittäjänä, joka ottaa tulosteen merkin vasemmalla puolella olevasta komennosta ja antaa sen merkin oikealla puolella olevalle komennolle käsiteltäväksi (Hess 2019).

"*Grep*"-toiminnolla voidaan hakea esimerkiksi tiettyä merkkijonoa. Tässä tapauksessa sitä käytettiin hakemaan merkkijonoja "passwordauthentication" ja "pubkeyauthentication". Kaksi eri merkkijonoa voidaan määrittää hakukohteeksi asettamalla pystyviiva hakukohteina olevien merkkijonojen väliin. (Kerrisk 2026a.)

Komennon tuloste näytti seuraavalta, joten kaikki on kunnossa ja voitiin jatkaa eteenpäin:

```
pubkeyauthentication yes  
passwordauthentication no
```

#### 4.1.4 Reitittimen konfigurointi

Tietoturvallisuuden lisäämiseksi lähiverkossa olevalle 4G-reitittimelle, jonka lävitse myös yhteys ulkoiseen internetiin tapahtuu, konfiguroitiin porttiohjaus (*port forwarding*) ohjaamaan IP-liikenne julkisen portin 8022 kautta yksityiseen porttiin numero 22. Taulukossa 4 nähdään katsaus 4G-reitittimen konfiguraatioista, jossa myös porttiohjaus on konfiguroituna.

Taulukko 4. 4G-reitittimen konfiguraatio

Staattinen reititys			
Kohde IP		Yhdyskäytävä	
10.6.8.0		10.10.10.4	
10.200.0.0		10.10.10.4	
Porttiohjaus			
Palvelimen IP-osoite	Lähde IP-osoite	Julkinen portti	Yksityinen portti
10.10.10.4	Mikä tahansa	8022	22

#### 4.2 IP-osoitteiden konfigurointi

Lähiverkko rakennettiin taulukon 5 mukaisesti 10.10.10.0/24-aliverkkoon. Laitteiden staattiset IP-osoitteet asetettiin järjestelmällisesti niin, että tietyillä laitteilla on tietyntyyppiset IP-osoitteet. 4G- ja 5G-virtuaalikoneiden IP-osoitteet päättyvät teknologiasukupolven mukaisen numeron mukaan numeroihin 4 ja 5. Nämä virtuaalikoneet toimivat isäntäkoneen päällä, joten isäntäkoneelle oli luonnollista asettaa numeroon 3 päättyvä IP-osoite. Kunkin verkon tukiaseman IP-osoitteiden päätteet vastaavat myös käytettävää teknologiaa, mutta koska ".4"- ja ".5"-päätteiset IP-osoitteet olivat jo käytössä, valikoitiin tukiasemien IP-osoitteiksi ".40"- ja ".50"-päätteiset osoitteet.

Taulukko 5. Verkon IP-infrastruktuuri (Uusitalo 2025c)

Hallinta				
IP-osoite	Aliverkon maski	Laite	Tarkoitus	Lisätieto
10.123.10.0	/24	Verkko		
10.10.10.1	/24	4G-reititin	Etähallinta, yhdyskäytävä ( <i>gateway</i> )	
10.10.10.40	/24	4G-Tukiasema		
10.10.10.50	/24	5G-Tukiasema		
10.10.10.3	/24	Isäntäkone		
10.10.10.4	/24	4G-Core (MME)		
10.10.10.5	/24	5G-Core (AMF)		
10.10.10.129	/24	Hallintakone		Vaihtoehtoisesti jotain DHCP alueelta 10.10.10.140-180
10.10.10.255	/24	Yleislähetys ( <i>broadcast</i> )		
RAN				
IP-osoite	Aliverkon maski	Laite	Tarkoitus	Lisätieto
10.200.0.0	/24		Päätelaitteille varattu verkko	

IP-forward-toiminto täytyi asettaa päälle niin isäntäkoneesta kuin virtuaalikoneistakin seuraavilla toimenpiteillä. Ensiksi avattiin tarvittava konfiguraatiodosto komennolla `"sudo nano /etc/sysctl.conf"` ja poistettiin sieltä ristikkomerkki eli kommentointimerkki `"net.ipv4.ip_forward=1"`-merkkijonon alusta. Tallennettiin muutokset, poistuttiin tekstieditorista ja laitettiin päivitetty konfiguraatio toimintaan komennolla `"sudo sysctl -p /etc/sysctl.conf"`. (Nanni 2012.)

Staattisia IP-osoitteita käytettäessä täytyy DHCP-palvelu poistaa käytöstä niin isäntä- kuin virtuaalikoneilta. Laitekohtaiset IP-osoite- ja DHCP-konfiguraatiot nähdään seuraavien lukujen kuvioista 12 (s. 35) ja 14 (s. 37).

#### 4.2.1 Isäntäkoneen konfigurointi

Isäntäkone on yhteydessä yhdyskäytävänä toimivaan 4G-reitittimeen. 4G-reititin löytyy IP-osoitteen 10.10.10.1 takaa. Uusitalon (2025a) mukaan isäntäkoneelta täytyy luoda silta virtuaalikoneille, jotta IP-liikenne voidaan reitittää kulkemaan virtuaalikoneille, näin ollen isäntäkoneen toimiessa ikään kuin reitittimenä virtuaalikoneille. Suoritettiin isäntäkoneella komento ”*sudo nano /etc/netplan/50-cloud-init.yaml*” ja muokattiin tiedostosta löytyvät konfiguraatiot kuvion 12 mukaiseksi (Trudel-Lapierre 2018; Lembo 2020; Märdian 2022; 2024).

```
# This file is generated from information provided by the datasource. Changes
# to it will not persist across an instance reboot. To disable cloud-init's
# network configuration capabilities, write a file
# /etc/cloud/cloud.cfg.d/99-disable-network-config.cfg with the following:
# network: {config: disabled}
network:
  ethernets:
    enp2s0:
      dhcp4: false
  version: 2
# wifis:
#   wlp0s20f3:
#     access-points:
#       WIFI NAME:
#         password: PASSWORD
#     dhcp4: true
  bridges:
    br0:
      interfaces: [enp2s0]
      dhcp4: false
      addresses: [10.10.10.3/24]
      routes:
        - to: default
          via: 10.10.10.1
      nameservers:
        addresses: [8.8.8.8, 8.8.4.4]
```

Kuvio 12. Isäntäkoneen verkkokonfiguraatiotiedosto

Kuviossa 12 näkyy pois kommentoituna WiFi-verkon yhdistämiseen vaadittava konfiguraatioblokki. Blokki on kommentoitu pois, sillä WiFi-yhteyttä ei tarvita normaalikäytön aikana, mutta jätettiin käyttövalmis pohja WiFi-yhteyden konfigurointia varten, mikäli sitä joskus tarvitsee.

Kuviossa 12 nähdään myös teksti, joka kertoo, että konfiguraatiotiedostoon tehdyt muutokset eivät ole pysyviä, vaan muuttuvat takaisin oletusasetuksiin laitteen uudelleenkäynnistyksen yhteydessä. Muutoksien säilyttäminen myös uudelleenkäynnistyksen jälkeen edellyttää uuden konfiguraatiotiedoston luomista ”/etc/cloud/cloud.cfg.d/”-hakemistoon. Suoritettiin komento ”*sudo nano*

`/etc/cloud/cloud.cfg.d/99-disable-network-config.cfg`” ja kirjoitettiin avautuneeseen konfiguraatiodostoon merkkijono `network: {config: disabled}`”. Tallennettiin muutokset ja suoritettiin tekstieditorin sulkeuduttua komennot `“sudo netplan try”` ja `“sudo netplan apply”`. Ensimmäinen komento tarkastaa `“50-cloud-init.yaml”`-konfiguraatiodoston toimivuuden ja toinen ottaa uudet asetukset käyttöön (Lembo 2020; Uusitalo 2025a).

Isäntäkoneen puolella tehtiin vielä yksi toimenpide. Varmistettiin, että virtuaalikone on sammuksissa komennolla `“virsh list --all”` ja avattiin tämän jälkeen virtuaalikoneen `domain.xml` komennolla `“virsh edit vm4g”`. Etsittiin `domain.xml`-tiedostosta kohta `<interface type='network'>` ja muokattiin se vastaamaan kuviossa 13 esitettäviä parametreja. Käynnistettiin lopuksi virtuaalikone komennolla `“virsh start vm4g”`. (Foerster & Smythies 2018; Libvirt 2026a; 2026b.)

```
<interface type='bridge'>
  <mac address='52:54:00:fe:73:ac' />
  <source bridge='br0' />
  <model type='virtio' />
  <address type='pci' domain='0x0000' bus='0x01' slot='0x00' function='0x0' />
</interface>
```

Kuvio 13. vm4g-virtuaalikoneen `domain.xml`-tiedoston `interface`-lohko

#### 4.2.2 Virtuaalikoneen konfigurointi

Virtuaalikoneen tavanomaiset `netplan`-verkkoasetukset konfiguroitiin niin, että virtuaalikone on yhteydessä lähiverkkoon. Suoritettiin komento `“sudo nano /etc/netplan/50-cloud-init.yaml”`. Avautuneeseen konfiguraatiodostoon konfiguroitiin yhdyskäytävä (*gateway*), laitteen IP-osoite sekä nimipalvelu (*nameservers*) kuvion 14 mukaisesti.

```
# This file is generated from information provided by the datasource. Changes
# to it will not persist across an instance reboot. To disable cloud-init's
# network configuration capabilities, write a file
# /etc/cloud/cloud.cfg.d/99-disable-network-config.cfg with the following:
# network: {config: disabled}
network:
  ethernets:
    enp1s0:
      dhcp4: false
      addresses: [10.10.10.4/24]
      routes:
        - to: default
          via: 10.10.10.1
      nameservers:
        addresses: [8.8.8.8, 8.8.4.4]
  version: 2
```

Kuvio 14. Virtuaalikoneen verkkokonfiguraatiotiedosto

Samoin kuin luvussa 4.2.1 isäntäkoneella, täytyi virtuaalikoneellakin luoda ja konfiguroida "99-disable-network-config.cfg"-konfiguraatiotiedosto. Suoritettiin komento "*sudo nano /etc/cloud/cloud.cfg.d/99-disable-network-config.cfg*" ja kirjoitettiin sinne rivi "*network: {config: disabled}*". Päivitetyt konfiguraatiot otettiin vielä käyttöön komennoilla "*sudo netplan try*" ja "*sudo netplan apply*".

### 4.3 Verkon toiminnan todentaminen

Verkkojen toiminnan todentamiseen käytettiin OnePlus 11 -matkapuhelinta sekä Milesight 5G Dongle -, Teltonika RUT200 - ja Teltonika RUTX50 -reitittimiä. Näistä RUT200 on ainoa 4G-laite, eikä näin ollen sovellu 5G-verkossa käytettäväksi.

Toiminnan todentamiseen käytettiin Linuxin "*ping*"-komentoa (Kerrisk 2026c) sekä isäntäkoneella pyörivää OpenSpeedTest-ohjelmaa (OpenSpeedTest 2023). Matkapuhelimella käytettiin NetMonster-sovellusta (Mroček 2026) ping-komennon suorittamiseen.

4G-coren MME- ja 5G-coren AMF-lokeja seurattiin Linuxin journalctl-työkalua ja Byobu-terminaalimultiplekseriä hyödyntäen. Journalctl-työkalulla voidaan tarkastella systemd-journald.servicen ja systemd-journald-remote.servicen kirjaamia lokimerkintöjä tietyltä aikaväliltä tai reaaliajassa (Kerrisk 2026b). Byobulla voidaan kuvion 15 mukaan jakaa terminaali-ikkuna useampaan osaan ja suorittaa eri toimintoja eri ikkunoissa samanaikaisesti (Kirkland 2020).

```

Alt-Left/Right      Move focus among windows
Alt-Up/Down         Move focus among sessions
Shift-Left/Right/Up/Down  Move focus among splits
Shift-F3/F4         Move focus among splits
Ctrl-F3/F4          Move a split
Ctrl-Shift-F3/F4    Move a window
Shift-Alt-Left/Right/Up/Down  Resize a split
F5                  Reload profile, refresh status
Alt-F5              Toggle UTF-8 support, refresh status
Shift-F5            Toggle through status lines
Ctrl-F5             Reconnect ssh/gpg/dbus sockets
Ctrl-Shift-F5       Change status bar's color randomly
F6                  Detach session and then logout
Shift-F6            Detach session and do not logout
Alt-F6              Detach all clients but yourself
Ctrl-F6             Kill split in focus
F7                  Enter scrollback history
Alt-PageUp/PageDown  Enter and move through scrollback
Shift-F7            Save history to $BYOBU_RUN_DIR/printscre
en
F8                  Rename the current window
Ctrl-F8             Rename the current session
Shift-F8            Toggle through split arrangements
Alt-Shift-F8        Restore a split-pane layout
Ctrl-Shift-F8       Save the current split-pane layout
F9                  Launch byobu-config window
Ctrl-F9             Enter command and run in all windows
Shift-F9            Enter command and run in all splits
F10                 * Used by X11 *
F11                 * Used by X11 *
Alt-F11             Expand split to a full window
Shift-F11           Zoom into a split, zoom out of a split
Ctrl-F11            Join window into a vertical split
F12                 Escape sequence
Shift-F12           Toggle on/off Byobu's keybindings
Alt-F12             Toggle on/off Byobu's mouse support
Ctrl-Shift-F12     Mondrian squares
(END)

```

```

kirkland@x230:~) # you can also "break" a split out to a window
kirkland@x230:~) # of it's own, permanently
kirkland@x230:~) # Alt-F11
kirkland@x230:~) █

BYOBU_WINDOW kirkland@x230:~)
S="$BYOBU_CONFIG_DIR/windows.tmux"
    fl
    ;;
    screen)
        # Allow override of
        default window list, with BYOBU_WIND
        OWS environment variable
        CUSTOM_WINDOW_SET=0
        if [ -s "$BYOBU_WIND
        kirkland@x230:~)
            CUSTOM_WINDO
M_SET=1
        elif [ -s "$BYOBU_CO
NFIG_DIR/windows.SBYOBU_WINDOWS" ];
        then
            CUSTOM_WINDO
M_SET=1
126,2-9 59%

```

14.04 deno 0: bash- 1:heIp+ 2:top# e636 81 19h2m 3264rpm 77%+ 0.04 4x2.3GHz 15.465% kirkland@x230 10.0.47.2 08:44UTC 2014-08-09 10:44:43

Kuvio 15. Byobu-terminaalimultiplekseri (Kirkland 2014)

## 4.4 Varmuuskopiointi

### 4.4.1 Virtuaalikoneen varmuuskopiointi

Beit-Halahmin (2023a) mukaan yksi yksinkertainen tapa luoda varmuuskopio KVM-virtuaalikoneesta on kopioida virtuaalikoneen tiedot sisältävä qcow2-tiedosto sekä tietyt konfiguraatiot sisältävä domain.xml-tiedosto ulkoiselle muistilaitteelle. Virtuaalikoneet saattavat olla isoja tiedostokooltaan, kuten tässä tapauksessa virtuaalikoneen ollessa 201:n gigatavun kokoinen. On siis syytä varmistaa, että käytettävällä muistilaitteella on riittävästi kapasiteettia varmuuskopioitavien virtuaalikoneiden varastointiin. Kirjaututtiin siis sisään isäntäkoneeseen ja tarkastettiin virtuaalikoneen qcow2-tiedoston koko komennolla `sudo ls -lh /var/lib/libvirt/images/CoreVM/vm4g.qcow2`. Kuviossa 16 nähdään, että komento listaa kohdetiedoston tiedot ihmissilmälle selkeästi luettavassa muodossa (Canonical 2025b).

```

agranucserver@agranucserver:~$ sudo ls -lh /var/lib/libvirt/images/CoreVM/vm4g.qcow2
[sudo] password for agranucserver:
-rw----- 1 libvirt-qemu kvm 201G Aug 28 13:54 /var/lib/libvirt/images/CoreVM/vm4g.qcow2

```

Kuvio 16. Komennon tuloste

Ulkoisena muistilaitteena, johon varmuuskopiot otetaan, käytettiin LaCie Rugged USB-C 2TB SSD:tä. Aluksi virtuaalikoneet täytyi sammuttaa, sammutuksen jälkeen voitiin kytkeä ja kiinnittää muistilaitte isäntäkoneeseen. Kun tämän jälkeen yritettiin luoda hakemistoa virtuaalikoneiden varmuuskopioita varten, saatiin virheviesti, jossa kerrottiin, että riittäviä oikeuksia ei löydy. Tiedonhaun ja selvittelyn jälkeen selvisi, että ulkoinen muistilaitte täytyy olla formatoituna exFat-muotoon, jotta muistilaitteelle kirjoittaminen Ubuntu-järjestelmän kautta onnistuu (Uusitalo 2025b). Tämän lisäksi kiinnitys suoritettiin varmuuden vuoksi tarkemmalla mount-komennolla `"sudo mount -t exfat -o uid=$UID,gid=$(id -g),umask=22 /dev/sda1 /media/usb-drive"` (Uusitalo 2025b). Komento määrittää tiedostojärjestelmäksi exFat-muodon, asettaa nykyisen käyttäjän sekä käyttäjäryhmän laitteen omistajaksi ja asettaa luku- sekä kirjoitusoikeudet laitteeseen näille osapuolille. Suoritettiin vielä komennot `"cd /media/usb-drive"` ja `"mkdir -m 777 VM4G"`, joilla siirrytään "usb-drive"-hakemistoon ja luodaan sinne "VM4G"-hakemisto tarvittavilla oikeuksilla.

Valmistelujen jälkeen päästiin suorittamaan itse varmuuskopion tekeminen. Ensiksi suoritettiin komento `"virsh dumpxml vm4g > /media/usb-drive/VM4G/vm4g.xml"`, joka kopioi virtuaalikoneen domain.xml-tiedoston haluttuun hakemistoon. Seuraavaksi suoritettiin komento `"sudo cp /var/lib/libvirt/images/CoreVM/vm4g.qcow2 /media/usb-drive/VM4G/"`, joka kopioi virtuaalikoneen qcow2-tiedoston haluttuun hakemistoon. (Beit-Halahmi 2023a.) Viimeisimmän komennon antamisen jälkeen odotettiin noin 1–2 tuntia varmuuskopion valmistumista.

5G-virtuaalikoneen kohdalla suoritettiin samat toimenpiteet sillä erolla, että 5G-virtuaalikoneelle luotiin ulkoiselle muistilaitteelle oma hakemisto nimellä "VM5G" ja XML-tiedosto nimettiin 5G-virtuaalikoneen mukaan. Varmuuskopioinnin jälkeen ulkoisen muistilaitteen kiinnitys voitiin poistaa umount-komennolla, kytkeä muistilaitte irti isäntäkoneesta ja käynnistää virtuaalikoneet.

#### 4.4.2 Varmuuskopion käyttö

Varmuuskopion käyttöönottoa ei opinnäytetyön toteutuksen aikana kokeiltu käytännössä. Tämän luvun sisältö perustuu Beit-Halahmin (2023b) julkaisemaan kirjoitukseen KVM-virtuaalikoneen varmuuskopion käyttöönottamisesta.

Aluksi täytyy varmistaa, että käytettävällä laitteella on riittävästi muistikapasiteettia virtuaalikoneen siirtämiseksi. Tämän jälkeen voidaan kytkeä varmuuskopion sisältävä muistilaitte kiinni kohdelaitteeseen ja suorittaa kiinnitystoiminto.

Komennolla `cp /media/usb-drive/VM4G/vm4g.xml /path/to/VM/location` kopioidaan virtuaalikoneen domain.xml-tiedosto haluttuun kohteeseen. Määritetään virtuaalikoneen asetukset XML-tiedostosta komennolla `virsh define /path/to/VM/vm4g.xml`.

Avataan virtuaalikoneen domain.xml-tiedosto komennolla `virsh edit vm4g.xml` ja tarkastetaan kuvion 17 esimerkin mukaan, että `<source>`-kohdassa määritelty virtuaalikoneen qcow2-tiedoston sijainti vastaa sijaintia, johon qcow2-tiedosto halutaan kopioida. Kun tämä on varmistettu, voidaan qcow2-tiedosto kopioida ulkoiselta muistilaitteelta domain.xml-tiedostossa mainittuun sijaintiin `cp`-komentoa käyttämällä.

```
<devices>
  <emulator>/usr/bin/qemu-system-x86_64</emulator>
  <disk type='file' device='disk'>
    <driver name='qemu' type='qcow2' discard='unmap' />
    <source file='/var/lib/libvirt/images/CoreVM/vm4g.qcow2' />
    <target dev='vda' bus='virtio' />
    <address type='pci' domain='0x0000' bus='0x04' slot='0x00' function='0x0' />
  </disk>
```

Kuvio 17 vm4g.xml-tiedoston devices-lohko

Varmuuskopion tulisi teoreettisesti tämän jälkeen olla valmis käytettäväksi. Virtuaalikoneen voi käynnistää ja ulkoisen muistilaitteen voi kytkeä irti isäntäkoneesta.

## 5 SIM-KORTIT

SIM-korttien lukemiseen ja kirjoittamiseen käytettiin HID Omnikey 3121 -älykortinlukijaa, joka oli kytkettynä Raspberry Pi 4 -tietokoneeseen. Raspberrille asennettiin pySim-työkalusarja, jotta kortinlukijaa pystyttiin hyödyntämään SIM-korttien lukemiseen ja kirjoittamiseen. Ohjelmoitavat kortit olivat ETSI/3GPP standardin kanssa yhteensopivia sysmocom sysmoISIM-SJA5 SIM/UICC/USIM/ISIM/HPSIM -kortteja (sysmocom 2026).

Suunnitelmana oli lukea ja tarkastella valmiiksi verkkoon ohjelmoidun SIM-kortin tietueita, tutustua tietueisiin ja oppia, mikä merkitys milläkin tietueella on SIM-kortin toiminnan kannalta ja soveltaa kertynyttä oppia tämän jälkeen sysmocom sysmoISIM-SJA5-S17 -kortin ohjelmoimiseen. PySimin sisältämistä skripteistä käytettiin pääasiassa kolmea skriptiä, joista lisää alaluvuissa.

### 5.1 pySim-prog.py

Komennolla voidaan kirjoittaa parametrina annettavia tietueita kortille. Komennon syntaksi näyttää esimerkiksi seuraavanlaiselta: *"pySim-prog.py -p <PC/SC laite-numero> --pin-adm <ADM PIN> --name <operaattorin nimi> --ki <Ki-arvo> --mcc <MCC-tunnus> --mnc <MNC-tunnus> --iccid <ICCID-arvo> --imsi <IMSI-arvo> --acc <ACC-numero>"* (Chlosta, Herle, Maier, Munaut & Welte 2025, 54; Welte 2024b, 16). Esimerkissä kulmasulkeet ja niiden sisältö poistetaan ja paikalle kirjoitetaan tarvittavat tiedot.

Parametri **-p** määrittää käytettävän PC/SC-tyyppisen laitteen numeron. PC/SC-laitteen numeron voi tarkastaa komennolla *"pcsc scan -r"*.

**ADM PIN** -numeroa käytetään autentikointimetodina, se toimitetaan kortin mukana. Parametrin **--name** perään kirjoitetaan verkon operaattorin nimi.

**Ki**-arvoa käytetään yhdessä OP-arvon kanssa laskemaan kortille OPc-arvo. OPc on 128-bittinen arvo, jota käytetään tiettyjen funktioiden laskelmoineissa (3GPP 2025, 7, 9). OP on operaattorikohtainen tunniste, jolla määritellään algoritmien toiminnan ero eri operaattoreiden välillä (Welte 2024b, 3). Tarvittava OP löytyi

Cumucoren web-hallintapaneelista. Ki on salainen korttikohtainen verkkoautentikaatioavain (Welte 2024b, 17), tarvittava Ki toimitettiin sysmocom-korttien mukana.

**MCC-tunnus** on maakohtainen kolminumeroinen tunniste. Esimerkiksi Suomessa tämä on 244. **MNC-tunnus** on mobiiliverkkotunniste, joka yhdessä MCC:n kanssa luo operaattorikohtaisen tunnisteen. (MCC-MNC 2026.)

**ICCID** on ITU-T:n suosituksessa E.118 määritelty ISO/IEC 7812:een pohjautuva sirukorttikohtainen numero. ICCID:tä ei lähetetä radioliityntärajapinnan kautta, joten sillä ei Welten mukaan ole yksityisten matkapuhelinverkkojen operoinnin yhteydessä niinkään merkittävää roolia. (Welte 2024b, 5.)

**IMSI** on oletusarvoisesti uniikki 3GPP-verkkoteknologian tilaajakohtainen numero. Julkisissa verkoissa IMSIn täytyy olla uniikki, mutta privaattiverkoissa ei välttämättä. IMSIn ensimmäiset 5–6 numeroa muodostuvat yleensä kortin MCC- ja MNC-arvoista. (Welte 2024b, 5.) Käytettyjen korttien IMSI muodostui verkon MCC- ja MNC-arvoista sekä korttikohtaisesta ennalta määritellystä päätteestä. Valmiiksi verkkoon konfiguroitujen korttien IMSIen päätteet oli numeroitu numerosta 001 numeroon 016, joten ohjelmoitavan kortin IMSIin päätteeksi valittiin 017.

Tukiasema lähettää bittimaskin hyväksymää **ACC**:ta. Vain tilaajat (*subscriber*), joilla on sama ACC kuin tukiaseman lähettämä, voivat käyttää tukiaseman luo-  
maa verkkoa (Welte 2024b, 33).

## 5.2 pySim-read.py

Komennolla voidaan lukea suuri osa kortinlukijaan asetetun kortin tärkeimmistä tiedoista. Komento ei kuitenkaan näytä sellaista tietoa, joka vaatii autentikointia tiedon tarkastelemiseen. (Chlosta ym. 2025.)

Käytin tätä skriptiä hyödyksi tutkiessani valmiiksi verkkoon ohjelmoidun SIM-kortin tietueita. Vertailin valmiin kortin tietueita ”tyhjän” eli ohjelmoimattoman sysmocom-kortin tietoihin ja tämän pohjalta selvittelin, mitä mikäkin tietue merkitsee ja miksi se on kortin toiminnan kannalta oleellinen tieto.

### 5.3 pySim-shell.py

PySim-shell luotiin pySim-progin jälkeen SIM-kortille määriteltävien parametrien määrän kasvaessa sen verran suureksi, että kaikkien parametrien välittäminen yhteen komentoon ei enää ollut suotuisaa (Chlosta ym. 2025, 54). PySim-shellillä pääsee siis tarkemmin tutkimaan kortin sisältöä ja muokkaamaan kortin tietoja.

Ohjelma voidaan käynnistää mukaillen otsikkoa 5.3 seuraavasti: `python3 pySim-shell.py -p 0`. Komennon parametri `-p 0` on sama kuin luvussa 5.1 on käsitelty. Ohjelma aukeaa hakemistoon `00:MF`, kuten kuviossa 18 nähdään.

```

[laforge@nataraja%pts/14 (20:14) ~/20231220-osmodevcall/pysim > ./pySim-shell.py -p 0
Using PC/SC reader number 0
Waiting for card...
Info: Card is of type: UICC
Detected UICC Add-on "SIM"
Detected UICC Add-on "GSM-R"
Detected UICC Add-on "RUIM"
AIDs on card:
  USIM: a0000000871002ffffffff8907090000 (EF.DIR)
  ISIM: a0000000871004ffffffff8907090000 (EF.DIR)
  ARA-M: a000000015141434c00
Detected CardModel: SysmocomSJA5
welcome to pySim-shell!
(C) 2021-2023 by Harald Welte, sysmocom - s.f.m.c. GmbH and contributors
Online manual available at https://downloads.osmocom.org/docs/pysim/master/html/shell.html
pySIM-shell (00:MF)>

```

Kuvio 18. pySim-shell-käyttöliittymä (Welte 2024a)

PySim-shellissä liikutaan kortin sisällä hakemistosta toiseen samaan tapaan kuin tekstikäyttöliittymien kanssa yleensäkin. Kehotteessa olevan hakemiston sisällön voi tarkastaa `dir`-komennolla ja liikkuminen tapahtuu `select`-komennolla.

### 5.4 SIM-kortin ohjelmointi

Ajatuksena oli, että yritetään ensin saada kortti toimimaan 4G-verkossa. Mikäli tämä onnistuu, jatketaan ohjelmointiyritystä niin, että saataisiin kortti toimimaan myös 5G-verkossa. Luvussa 5.4.1 käydään läpi kortin ohjelmoimiseen yleisesti liittyvää asiaa ja luvussa 5.4.2 käydään läpi, mitä tähän tehdään lisäksi, jotta saataisiin kortti yhdistymään 5G-verkkoon.



Vasta ohjelmoidun kortin pySim-read.py-tulostetta verratessa valmiiksi ohjelmoidun kortin tulosteeseen voidaan todeta, että kaikkia tarvittavia tietueita ei voida ohjelmoida pelkästään pySim-prog.py-ohjelmalla. Esimerkiksi kuviossa 20 nähtävä FPLMN-tietueen määrittäminen täytyi jälkikäteen tehdä pySim-shell-ohjelman kautta, kuten myös MSISDN-, SMSP-, HPLMNAcT-, OPLMNwAcT-, PLMNwAcT-, "Show in HPLMN" - ja "Hide in OPLMN" -tietueiden määrittäminen. Tietueiden muokkaaminen vaatii yleensä ADM PIN -numeron varmentamisen ennen muokkausta. Tämä tapahtuu komennolla "verify\_adm <ADM PIN>" (Chlosta ym. 2025, 5).

Kuviossa 20 vasemmalla nähdään valmiiksi ohjelmoidun kortin FPLMN-määrittäykset ja oikealla vasta pySim-prog-ohjelmalla ohjelmoidun kortin vajavaiset FPLMN-määrittäykset. FPLMN määrittää kielletyt verkot eli sellaiset MCC/MNC-yhdistelmät, joihin kortti ei yritä yhdistää. Tässä vaiheessa otettiin vielä vain mallia valmiiksi ohjelmoidusta kortista, joten uudelle kortille ohjelmoitiin samat kolme kiellettyä verkkoa, jotka valmiiksi ohjelmoidulta kortiltakin löytyi.

FPLMN:	FPLMN:
42f419 # MCC: 244 MNC: 91	ffffff # unused
42f464 # MCC: 244 MNC: 46	ffffff # unused
42f454 # MCC: 244 MNC: 45	ffffff # unused
ffffff # unused	ffffff # unused

Kuvio 20. FPLMN-määrittäykset

Näkisin, että esimerkiksi jollekin asiakasyritykselle räätälöityjä kortteja ohjelmoitaessa FPLMN-tietueeseen määriteltäisiin kortin käyttöalueella olevat verkot, joihin kortin ei haluta tekevän yhteydenottoyrityksiä. FPLMN-lista on hyvä määrittää myös siksi, että sillä säästetään aikaa laitteen uudelleen käynnistyessä sen ohitettaessa mahdolliset yhteydenottoyritykset FPLMN-listalle määritettyihin verkkoihin (Apostu 2024).

Tietueiden muuttaminen pySim-shellillä tapahtuu liikkumalla ensin "select"-komennolla hakemistoon, jossa tietue sijaitsee ja käyttämällä seuraavanlaisia komentoja komentokohtaisin parametrein:

- *update\_binary*
- *update\_record*
- *update\_binary\_decoded*

- *update\_record\_decoded*
- *edit\_binary\_decoded*
- *edit\_record\_decoded*

”\_decoded”-päätte avaa tietueen JSON-muodossa. Ilman tätä päätettä voidaan muokata tietueen heksa-enkoodattua binääriä (Chlosta ym. 2025, 1).

Kortti täytyi lisätä vielä Cumucoren web-hallintapaneelin kautta 4G- ja 5G-corelle tilaajaksi. Kortti voi ottaa yhteyden verkkoon vain, mikäli se on lisätty verkolle tilaajaksi.

Seuraavaksi voitiin kokeilla yhdistää vasta ohjelmoitua korttia 4G-verkkoon. Kortti asetettiin RUT 200 -reitittimen sisään ja verkkoon yhdistäminen tehtiin samoin kuin valmiiksi verkkoon ohjelmoitujenkin korttien kanssa. 017-kortti toimi verkossa moitteettomasti, joten kortin ohjelmoiminen voitiin todeta siltä osin onnistuneeksi.

#### 5.4.2 SIM-kortin ohjelmointi 5G-verkkoon

5G-verkkoon ei vielä tässä vaiheessa prosessia oltu onnistuttu yhdistämään laitteita, vaikka core-palvelimen AMF-lokeja tarkastellessa on nähty yhteydenottoyrityksiä eri laitteilta. Tämä johtui mahdollisesti siitä, että tähän asti käytetyt valmiiksi ohjelmoidut SIM-kortit ovat LTE-kortteja, eivätkä näin ollen olisi yhteensopivia 5G:lle ominaisen NR-verkon kanssa. Käytettävällä SIM-kortilla täytyy olla yhteensopivuus vähintään 3GPP:n standardijulkaisun 16 kanssa toimiakseen NR-verkossa. Käytettävä sysmocom-kortti täyttää tämän vaatimuksen, joten ohjelmointiyritystä jatkettiin 5G-verkkoon.

Ymmärrykseni mukaan SUCI-laskenta täytyy ottaa käyttöön yhdistettäessä 5G SA -verkkoon. Sysmocomin pySim-manuaalissa neuvotaan, että tämä toiminto voidaan ottaa käyttöön sysmolSIM-SJA5-S17-korttimallissa aktivoimalla palvelut 124 ja 125 (Chlosta ym. 2025, 8).

Tämä tapahtui avaamalla pySim-shell, kun 017-numeroinen kortti oli lukijassa paikallaan. Aluksi liikuttiin ”ADF.USIM/EF.UST”-hakemistoon komennoilla ”*select*

*ADF.USIM* ja *"select EF.UST"*. Tämän jälkeen annettiin komennot *"ust\_service\_activate 124"* ja *"ust\_service\_activate 125"*.

SUCI-laskennan toimivuus voidaan varmistaa ajamalla komento *"get\_identity"* *"ADF.USIM"*-hakemistossa. Komennon tulisi antaa tulosteena seuraavan tyyppistä: *"SUCI TLV Data Object: <merkkijono>"* (Chlosta ym. 2025, 8). Komento ei kuitenkaan antanut tämäntyyppistä tulostetta, vaan sen sijasta virheviestin *"EXCEPTION of type 'SwMatchError' occurred with message: SW match failed! Expected 9000 and got 6985: Command not allowed - Conditions of use not satisfied"*. Ensimmäisenä tuli mieleen, että komennon suorittaminen vaatii ADM PIN -numeron varmentamisen ennen mainittua komentoa luvussa 5.4.1 mainitulla komennolla *"verify\_adm"*. ADM PIN -numeron varmentamisenkin jälkeen saatiin kuitenkin sama virheviesti.

SUCI laskennassa on siis mahdollisesti jotain vielä vialla. Kokeilin kaikesta huolimatta yhdistää korttia 5G-verkkoon RUTX50-reitittimen avulla, mutta valitettavasti tuloksetta.

## 6 POHDINTA

Opinnäytetyön prosessi eteni jokseenkin tavanomaisesta poiketen, sillä käytännön osa suoritettiin touko–joulukuun välillä vuonna 2025, mutta opinnäytetyö aloitettiin virallisesti vasta tammikuussa vuonna 2026. Kirjoitin dokumentaatiota ja muistiinpanoja tehdyistä asioista koko käytännön prosessin ajan ja laadin näiden pohjalta tämän opinnäytetyön raportin. Työvaiheiden jälkikäteen muistelu tuotti toisinaan haasteita, mutta jokainen mietityttänyt seikka selvisi kyllä. Kirjoittamani dokumentaatio ja muistiinpanot toimivat hyvänä muistinvirkistykseenä, joskin joi-tain seikkoja täytyi jälkikäteen tarkastaa vielä paikan päältä. En osaa sanoa, olisi-ko opinnäytetyön työstäminen tavanomaiseen tyyliin ollut helpompi lähestymis-tapa.

Alussa määriteltyihin tavoitteisiin päästiin hyvin, sillä tuloksena on toimiva 4G-verkko ja itsenäisen 5G-verkon runko. 5G-verkkoon ei ikävä kyllä toistaiseksi tuntemattomasta syystä onnistuttu yhdistämään laitteita, vaikka vianetsintää ja on-gelmanratkaisua työstettiin sitkeästi. Niin kuin luvussa 5.4.2 mainittiinkin, tämä saattoi johtua siitä, että käytössä olleet valmiiksi ohjelmoidut kortit eivät olleet 3GPP standardijulkaisun 16 kanssa yhteensopivia, eikä oma SIM-kortin ohjel-mointiyritys ratkaissut asiaa. Uskon kuitenkin, että ongelma selviää vielä tulevai-suudessa.

SIM-korttien aihepiiriin katsaus tuotti kuitenkin tulosta, sillä yksi kortti saatiin on-nistuneesti ohjelmoitua toimimaan ainakin opinnäytetyön yhteydessä rakenne-tussa 4G-verkossa. Opittuja metodeja soveltamalla voi jatkossa ohjelmoida use-ampiakin kortteja toimimaan samassa verkossa, mikäli tarve vaatii.

Opinnäytetyön tuloksena olevaa kokonaisuutta voidaan hyvin hyödyntää Rova-niemen IoT-laboratorion toiminnassa. Toimiva itsenäinen verkko voidaan esimer-kiksi laajentaa osaksi nykyistä Off-Grid 5G -järjestelmää ja verkkoon voidaan oh-jelmoida räätälöityjä SIM-kortteja.

## LÄHTEET

3GPP 2025. 3GPP TS 35.206. Viitattu 16.3.2026

<https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=2391>.

Adjei, E. 2023. Understanding the initrd and vmlinuz in Linux Boot Process.

Viitattu 26.1.2026 [https://dev.to/er\\_dward/understanding-the-initrd-and-vmlinuz-in-linux-boot-process-534f](https://dev.to/er_dward/understanding-the-initrd-and-vmlinuz-in-linux-boot-process-534f).

Aleksic, M. 2024. How to install KVM on Ubuntu. Viitattu 3.6.2025

<https://phoenixnap.com/kb/ubuntu-install-kvm>.

Almesberger, W. & Lermen, H. 2000. Using the initial RAM disk (initrd). Viitattu

26.1.2026 <https://docs.kernel.org/admin-guide/initrd.html>.

Alves, M. 2022. Unable to disable password authentication over SSH. Viitattu

5.2.2026

<https://askubuntu.com/questions/1440290/unable-to-disable-password-authentication-over-ssh>.

Apostu, D. 2024. What is an FPLMN list? Viitattu 13.4.2026

<https://simbase.com/blog/what-is-an-fplmn-list>.

ASUSTeK Computer Inc. 2023. Asus NUC 13 Extreme Kit Tech Specs. Viitattu

23.5.2025 <https://www.asus.com/displays-desktops/nucs/nuc-kits/nuc-13-extreme-kit/techspec/>.

Awati, R. 2024. Evolved Packet Core (EPC). Viitattu 4.3.2026

<https://www.techtarget.com/searchnetworking/definition/Evolved-Packet-Core-EPC>.

Balena 2025. BalenaEtcher. Viitattu 23.1.2026 <https://etcher.balena.io/>.

Beck, B., Campbell, A., De Raadt, T., Friedl, M., Provos, N., Song, D. & Ylönen, T. 2013a. ssh. Viitattu 12.3.2026 <https://linux.die.net/man/1/ssh>.

– 2013b. sshd. Viitattu 2.2.2026 <https://linux.die.net/man/8/sshd>.

Beit-Halahmi, D. 2023a. KVM backups with Virt-Manager. Viitattu 6.2.2026

<https://itsfoss.com/community/t/kvm-backups-with-virt-manager/10654>.

– 2023b. Restoring a KVM Backup with Virt-Manager. Viitattu 10.4.2026

<https://itsfoss.com/community/t/restoring-a-kvm-backup-with-virt-manager/10655>.

Bellavue Linux 2005. vmlinuz Definition. Viitattu 26.1.2026

<https://www.linfo.org/vmlinuz.html>.

Bhatti, A., Haider, M. & Kirmani, A. 2007. Radio Resource Management In 3G UMTS Networks. Opinnäytetyö, Blekinge Institute of Technology. Viitattu

5.3.2026 [https://www.diva-](https://www.diva-portal.org/smash/get/diva2:832085/FULLTEXT01.pdf)

[portal.org/smash/get/diva2:832085/FULLTEXT01.pdf](https://www.diva-portal.org/smash/get/diva2:832085/FULLTEXT01.pdf).

Broz, M. & Zak, K. 2017. lsblk. Viitattu 11.2.2026  
<https://linux.die.net/man/8/lsblk>.

Bruden 2025. Using the Linux Loop Device. Viitattu 26.1.2026  
<https://www.brudenossg.com/tip4.php>.

Canonical 2025a. Ubuntu manuals, losetup. Viitattu 26.1.2026  
<https://manpages.ubuntu.com/manpages/resolute/en/man8/losetup.8.html>.

– 2025b. Ubuntu manuals, ls. Viitattu 6.2.2026  
<https://manpages.ubuntu.com/manpages/jammy/man1/ls.1.html>.

– 2025c. Ubuntu manuals, virt-install. Viitattu 28.1.2026  
<https://manpages.ubuntu.com/manpages/noble/man1/virt-install.1.html>.

Canonical Group Ltd. 2025. Virtual Machine Manager. Viitattu 27.1.2026  
<https://documentation.ubuntu.com/server/how-to/virtualisation/virtual-machine-manager/>.

Chlosta, M., Herle, S., Maier, P., Munaut, S & Welte, H. 2025. osmopysim-usermanual. Viitattu 20.4.2026  
<https://downloads.osmocom.org/docs/pysim/master/osmopysim-usermanual.pdf>.

Cumucore 2023a. 4G EPC Packet Core Specifications. Viitattu 21.3.2026  
<https://gsma.my.site.com/mwcoem/servlet/servlet.FileDownload?file=00P6900003SgnHzEAJ>.

– 2023b. 5G NC Specifications. Viitattu 21.3.2026  
<https://gsma.my.site.com/mwcoem/servlet/servlet.FileDownload?file=00P6900003SgnI9EAJ>.

Davie, B., Peterson, L. & Sunay, O. 2023. Private 5G: A Systems Approach. Viitattu 10.3.2026  
<https://5g.systemsapproach.org/core.html>.

DeepWiki 2025. KVM Acceleration. Viitattu 23.1.2026  
<https://deepwiki.com/alephsecurity/xnu-qemu-arm64/3.4-kvm-acceleration>.

Foerster, D. & Smythies, D. 2018. KVM Network Bridge to assign Static IP. Viitattu 5.2.2026  
<https://askubuntu.com/questions/638162/kvm-network-bridge-to-assign-static-ip/638269#638269>.

Free Software Foundation 2020. GNU Gzip. Viitattu 26.1.2026  
<https://www.gnu.org/software/gzip/>.

Garnett, A. & Kurup, M. 2022. How to Set Up SSH Keys on Ubuntu 22.04. Viitattu 2.2.2026  
<https://www.digitalocean.com/community/tutorials/how-to-set-up-ssh-keys-on-ubuntu-22-04>.

GTM 2021. Gigabytes (GB) to Mebibytes (MiB) conversion. Viitattu 28.1.2026  
<https://www.xconvert.com/unit-converter/gigabytes-to-mebibytes>.

Heimes, F. 2020. A dozen different ways of running KVM virtual machines on Ubuntu Server. Viitattu 28.1.2025 <https://ubuntu-on-big-iron.blogspot.com/2020/08/12-different-ways-of-running-ubuntu-server-on-kvm.html>.

Hess, K. 2019. Working with pipes on Linux command line. Viitattu 2.2.2026 <https://www.redhat.com/en/blog/pipes-command-line-linux>.

Hoke, K. & Snyder, J. 2025. Learn about KVM virtualization with QEMU and Libvirt. Viitattu 13.4.2026 <https://docs.netapp.com/us-en/netapp-solutions-virtualization/kvm/kvm-overview.html>.

IEEE 1541-2002 2009. IEEE Standards for Prefixes for Binary Multiples. Viitattu 31.3.2026 <https://doi.org/10.1109/IEEESTD.2009.5254933>.

Intel Corporation 2022a. Intel Ethernet Controller I226-V Specifications. Viitattu 23.1.2026 <https://www.intel.com/content/www/us/en/products/sku/210599/intel-ethernet-controller-i226v/specifications.html>.

– 2022b. Intel Killer Wi-Fi 6E AX1690 Specifications. Viitattu 23.1.2026 <https://www.intel.com/content/www/us/en/products/sku/217241/intel-killer-wifi-6e-ax1690-is/specifications.html>.

Jimm's PC-Store Oy 2025. ASUS NUC13RNGi7 tuotesivu. Viitattu 23.1.2026 <https://www.jimms.fi/fi/Product/Show/199531/90ab3rng-mr8120/asus-nuc-13-extreme-kit-nuc13rngi7-minipc-barebone>.

Karjalainen, A. 2026. Lapin ammattikorkeakoulu Oy. Asiantuntijan haastattelu 12.3.2026.

Kerrisk, M. 2026a. grep(1) – Linux manual page. Viitattu 2.2.2026 <https://man7.org/linux/man-pages/man1/grep.1.html>.

– 2026b. journalctl(1) – Linux manual page. Viitattu 1.4.2026 <https://man7.org/linux/man-pages/man1/journalctl.1.html>.

– 2026c. ping(8) – Linux manual page. Viitattu 31.3.2026 <https://man7.org/linux/man-pages/man8/ping.8.html>.

Kirkland, D. 2014. Learn Byobu while listening to Mozart. Viitattu 1.4.2026 <https://youtu.be/NawuGmcvKus?si=aXbKr7i8rKIRMOEA&t=476>.

– 2020. Byobu – ABOUT THE PROJECT. Viitattu 1.4.2026 <https://www.byobu.org/about>.

Korolev, S. 2023. How to Install Ubuntu 22.04 Virtual Machine on KVM. Viitattu 28.1.2026 <https://www.wpdiaaries.com/ubuntu-on-kvm/#:~:text=1.,see%20the%20file%20casper/vmlinuz%20>.

KVM contributors 2010. USB. Viitattu 28.1.2026 <https://linux-kvm.org/page/USB>.

- Larabel, M. 2021. Intel IGC Network Driver Queues XDP Support For Linux 5.13. Viitattu 23.5.2025 & 23.1.2026 <https://www.phoronix.com/news/Intel-IGX-Gets-XDP>.
- Lembo, P. 2020. Ubuntu Server Setup – with netplan. Viitattu 5.2.2026 <https://gist.github.com/plembo/52aa4e442943f68d86ec0ce31de8e0b0>.
- Lenovo 2026. Mikä on megatavu (MB)? Viitattu 30.3.2026 <https://www.lenovo.com/fi/fi/glossary/what-is-megabyte/>.
- Levine, N. 2025. How to Run an INSTALL.sh Script on Linux in 4 Easy Steps. Viitattu 29.1.2026 <https://www.wikihow.com/Execute-INSTALL.sh-Files-in-Linux-Using-Terminal>.
- Libvirt 2026a. Domain XML format. Viitattu 5.2.2026 <https://libvirt.org/formatdomain.html>.
- 2026b. PCI addresses in domain XML and quest OS. Viitattu 5.2.2026 <https://www.libvirt.org/pci-addresses.html>.
- Marvell 2022. Product Brief – Marvell® Scalable mGig AQC113/AQC114/AQC114CS/AQC115C. Viitattu 11.2.2026 <https://www.marvell.com/content/dam/marvell/en/public-collateral/ethernet-adaptersandcontrollers/marvell-fastLinq-edge-product-brief.pdf>.
- MCC-MNC 2026. MCC-MNC Database. Viitattu 16.3.2026 <https://mcc-mnc.com>.
- Microsoft 2023. type. Viitattu 30.1.2026 <https://learn.microsoft.com/en-us/windows-server/administration/windows-commands/type>.
- Mroček, M. 2026. NetMonster. Viitattu 31.3.2026 <https://netmonster.app/>.
- Märdian, L. 2022. Allow setting the regulatory domain. Viitattu 5.2.2026 <https://github.com/canonical/netplan/blob/main/examples/wireless.yaml>.
- 2024. Implementation of ip route advmss options for Systemd networkd and Network manager. Viitattu 5.2.2026 <https://github.com/canonical/netplan/blob/main/examples/static-routes.yaml>.
- Nanni, D. 2012. How to enable IP forwarding in Linux. Viitattu 5.2.2026 <https://www.xmodulo.com/how-to-enable-ip-forwarding-in-linux.html>.
- OpenSpeedTest 2023. Free and Open-Source Self-Hosted HTML5 SpeedTest. Viitattu 31.3.2026 <https://openspeedtest.com/selfhosted-speedtest#Source-Code-Docker>.
- Peterson, S. 2023. What is a mebibyte (MiB)? Viitattu 28.1.2026 <https://www.techtarget.com/searchstorage/definition/mebibyte-MiB>.
- Red Hat 2019. What is the Linux kernel? Viitattu 26.1.2026 <https://www.redhat.com/en/topics/linux/what-is-the-linux-kernel>.

Rendek, L. 2025. How to mount USB drive in Linux. Viitattu 11.2.2026  
<https://linuxconfig.org/howto-mount-usb-drive-in-linux>.

Salmela, H. 2026. SSH-protokollan -X-parametrin käyttö. Yksityinen WhatsApp-viesti 13.4.2026. Viestin saaja: Samuli Parpala.

SPI Inc. 2026a. Utilities for configuring the Linux Ethernet bridge. Viitattu 26.1.2026 <https://packages.debian.org/bookworm/bridge-utils>.

– 2026b. Virtualization daemon – typical QEMU-based deployment. Viitattu 26.1.2026 <https://packages.debian.org/sid/libvirt-daemon-system>.

– 2026c. Virtualization library – clients. Viitattu 26.1.2026  
<https://packages.debian.org/sid/libvirt-clients>.

Sultan, A. 2022. 5G System Overview. Viitattu 20.4.2026  
<https://www.3gpp.org/technologies/5g-system-overview>.

Suutari-Jääskö, E. 2025. 5G-käsikirja. Opinnäytetyö, Lapin ammattikorkeakoulu. Viitattu 12.3.2025 <https://urn.fi/URN:NBN:fi:amk-2025052615920>.

sysmocom 2026. sysmoSIM-SJA5 programmable SIM/USIM/ISIM/HPSIM cards. Viitattu 12.3.2026 <https://sysmocom.de/products/sim/sysmoisim-sja5/index.html>.

Takko, A. 2025. Kaikki mitä sinun pitää tietää 5G:stä. Viitattu 13.4.2026  
<https://mobiilitutka.fi/5g/>.

Trudel-Lapierre, M. 2018. Examples: make sure the bridge example is self-contained. Viitattu 5.2.2026  
<https://github.com/canonical/netplan/blob/main/examples/static-routes.yaml>.

Ubuntu 2024. Ubuntu 22.04.5 LTS download page 2024. Viitattu 23.1.2026  
<https://releases.ubuntu.com/jammy/>.

Uusitalo, T. 2025a. Lapin ammattikorkeakoulu Oy. Asiantuntijan haastattelu 27.6.2025.

– 2025b. Lapin ammattikorkeakoulu Oy. Asiantuntijan haastattelu 4.9.2025.

– 2025c. Lapin ammattikorkeakoulu Oy. Yksityinen Teams-viesti 10.6.2025. Viestin saaja: Samuli Parpala.

– 2025d. Lapin ammattikorkeakoulu Oy. Yksityinen Teams-viesti 26.6.2025. Viestin saaja: Samuli Parpala.

W3Schools 2026. Bits and Bytes in Programming. Viitattu 31.3.2026  
[https://www.w3schools.com/programming/prog\\_bits\\_and\\_bytes.php](https://www.w3schools.com/programming/prog_bits_and_bytes.php).

Wellesley College 2019. Principles of Programming Languages: Remote Access with SSH. Viitattu 17.4.2026  
<https://cs.wellesley.edu/~cs251/f19/docs/common/ssh/#x>.

Welte, H. 2024a. OsmoDevCall - Using pySim-shell on sysmoSIM-SJA5 cards. Viitattu 20.3.2026  
<https://youtu.be/tGSYoUq1I58?si=oJYzsPiGSKQH2joW&t=159>.

– 2024b. sysmoSIM-SJA5 User Manual. Viitattu 16.3.2026  
<https://sysmocom.de/manuals/sysmoisim-sja5-manual.pdf>.

Wragg, D. 2009. Using USB pass-through under libvirt and KVM. Viitattu 29.1.2026 <https://david.wragg.org/blog/2009/03/using-usb-pass-through-under-libvirt.html>.

## LIITE

### Liite 1. Hyödyllisiä komentoja “virsh” -käyttöliittymälle

A list of useful commands for operating and finding your way around virtual machines and VM environment.

#### **Command prompt commands**

*virsh* – Enter interactive virtualization terminal

Note: When in interactive virtualization terminal, the following commands should be given without the “virsh” prefix.

*virsh start vmname* – Start a VM

*virsh console vmname* – Open the serial console of an already running VM through console

*(Note: Console screen might show as empty even though the prompt is active.)*

*virsh start vmname --console* – Start a VM and enter a login screen through console

*virsh shutdown vmname* – Shut down a VM

*virsh reboot vmname* – Reboot a VM

*virsh list* – List running VMs

*virsh list --all* – List all VMs (Running and not running)

*virsh edit vmname* – Open VM domain xml file in editor

#### **Keyboard commands**

Ctrl+5 – Exit VM console without shutting down the VM when using SSH

Ctrl+9 – Exit VM console without shutting down the VM when working locally

Ctrl+C – Exit interactive virtualization terminal (“*exit*” & “*quit*” commands also work)