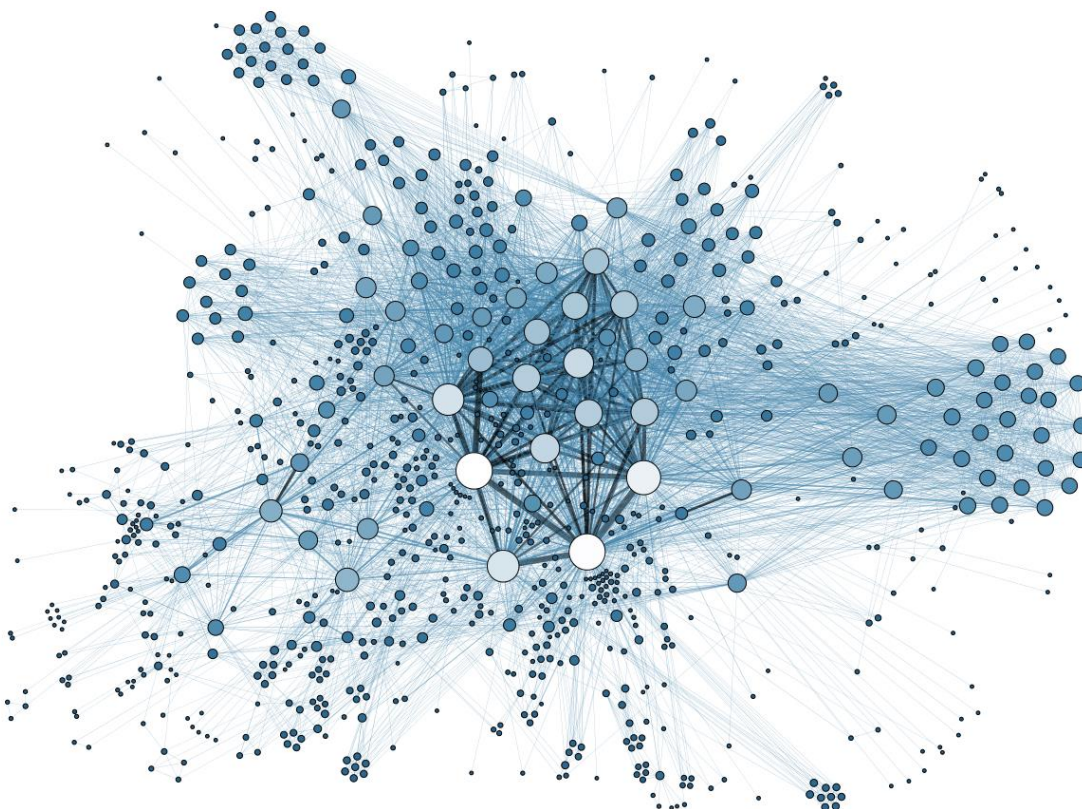


Alanen Jere

# Hajautetun MANET-verkon tilatallenteiden yhdistäminen yhdeksi tietokannaksi



Insinööri (AMK)

Tieto- ja viestintäteknikka

Kevät 2021



KAMK • University  
of Applied Sciences

## **Tiivistelmä**

**Tekijä:** Alanen Jere

**Työn nimi:** Hajautetun MANET-verkon tilatallenteiden yhdistäminen yhdeksi tietokannaksi

**Tutkintonimike:** Insinööri (AMK), tieto- ja viestintätekniikka

**Asiasanat:** JSON, Python, Kaavioteoria, Verkkoteoria, MANET

Työn tilaajana on Bittiumin Kajaanin yksikkö Bittium Wireless Oy. Bittium on oululainen yritys, jonka toiminta erikoistuu turvallisten viestintä- ja liitettävyyssratkaisujen kehittämiseen.

Opinnäytetyön aiheena oli perehtyä Python-ohjelmointikieleen, verkko- ja kaavioteoriaan, tutustua MANET-verkkojen tilatietojen tallentamiseen ja kehittää työkalu helpottamaan useiden tilatallennetiedostojen tulkintaa verkonvalvontatyökalulla.

Työkalun ohjelmointiin käytettiin Python-ohjelmointikieltä sen nopean ohjelmointi ja testaus syklin vuoksi. Lisäksi työkalun tuli tukea JSON-formaattia, että se on yhteensopiva olemassa olevan työkaluketjun kanssa.

Testaustulosten pohjalta havaittiin, että työkalu luo yhden tietokannan onnistuneesti käyttäjän antamien MANET-verkon tilatallenteiden perusteella, sekä verkon tilan tulkintaan vaadittavien tallenteiden vaatima koko vähentyi yhdistämisen avulla jopa 10 %:iin alkuperäisestä koosta. Tuloksista voidaan päätellä myös, että työkalua on vielä mahdollista jatkokehittää yhdistetyn tallenteen tarkentamiseksi sekä nopeuttamiseksi ja tekemällä käytettävästä käyttöliittymästä käyttäjäystävällisemmän.

## **Abstract**

**Author(s):** Alanen Jere

**Title of the Publication:** Combining MANET network record files to one database

**Degree Title:** Bachelor of Engineering, Information- and communication technologies

**Keywords:** JSON, Python, Graph theory, Network theory, Distributed network

This thesis was commissioned by Bittium Corporation unit of Bittium Wireless Oy located in Kajaani. Bittium is Finnish company headquartered in Oulu. Bittium specializes in the development of secured communications and connectivity solutions.

The purpose of this thesis was orientating to Python programming language, and to the network and graph theory. A further goal was to become familiar with MANET network recordings and to develop a tool to improve representation of the network, when multiple partial network topology snapshots are available.

The tool was done using Python programming language, because of its fast programming and testing cycle. Also, the tool had to support JSON-format, because of compatibility with the existing toolchain.

The test results indicate that the tool creates a database successfully based on the provided recordings and the needed storage space for the data was reduced to 10% of the original size. However, the results also show that the merge could still be improved to get an even more accurate merged database, faster merge time and more user-friendly user interface.

## Sisällys

1	Johdanto .....	1
2	Python-ohjelmointikieli .....	2
3	JSON.....	3
4	Kaavioteoria.....	4
4.1	Verkkoteoria .....	4
4.2	Verkkotopologia .....	5
4.3	Suuntaamaton verkko .....	5
4.4	Suunnattu verkko .....	6
4.5	Painotettu verkko.....	6
4.6	Verkon optimointi .....	7
5	MANET-verkko.....	8
6	Bittium TAC WIN -tuoteperhe .....	9
6.1	Bittium TAC WIN Tactical Router .....	9
6.2	Bittium TAC WIN Radio Head .....	9
7	Verkonvalvontatyökalu.....	10
8	Toteutus.....	11
8.1	Vaatusmäärittely .....	11
8.2	Suunnittelu .....	13
8.3	Kehitys .....	13
8.4	Testaus .....	14
8.4.1	Ensimmäinen testaus .....	14
8.4.2	Toinen testaus .....	18
9	Tulokset .....	19
9.1	Tulosten analysointi .....	19
9.2	Käytettävyys .....	19
9.3	Ongelmakohdat.....	20
9.4	Jatkokehitys.....	20

10	Yhteenveto .....	21
	Lähteet .....	22

## Symboliluettelo

Ad hoc	Tietokonetekniikassa ad hoc on langattomien lähiverkkojen välinen yhteystyyli, jossa langattomat laitteet toimivat ilman tukiaseman apua
Aikaleima	Solmulla tallennettu aika tietojen keräyshetkeltä
Flash	Adobe Flash
Intervalli	Tallennetietojen keräystiheys
IP	Internet Protocol
JSON	JavaScript Object Notation
LAN	Lähiverkko (Local Area Network)
Linkki	Solmujen välille muodostuva yhteys (Edge/Line/Link)
MANET	Langattomasti yhdistyvien mobiililaitteiden verkko (Mobile Ad hoc Network)
NumPy	Numeric Python
P2P	Point-to-Point Network
PMP	Point-to-Multipoint Network
SciPy	Scientific Python
SHDSL	Single-pair High-speed Digital Subscriber Line
Skripti/Script	Tietokoneen käyttämä komentosarjakieli. Käytetään myös kuvaamaan tietokoneen suorittamaa pientä ohjelmaa
Solmu	Kaaviossa oleva henkilö, tietokone tai asia, jolla on mahdollisuus olla yhteys toiseen solmuun (Node/Point/Vertices)
TAC WIN	Tactical Wireless IP Network

## 1 Johdanto

Bittium on oululainen yritys, jonka toiminta erikoistuu turvallisten viestintä- ja liitettävyyssratkaisujen kehittämiseen [2]. Aloitin Bittiumilla työt keväällä 2018 ja suoritin samalla harjoittelun. Opinnäytetyön aihe tuli samalla, kun minua siirrettiin yrityksen sisällä toiseen projektiin.

Työn tavoitteena on perehtyä Python-ohjelmointikieleen, verkko- ja kaavioteoriaan, Bittium TAC WIN -tuotteisiin ja kehittää työkalu, joka tukee hajautetuista MANET-verkoista tallennettujen tilatietojen tulkintaa.

Työkalun tulee pystyä yhdistämään useita hajautetun MANET-verkon solmuilta tallennettuja tilatallenteita yhdeksi tietokannaksi. Tämä auttaa ja nopeuttaa suurikokoisten verkkojen ja pitkäkestoisten tallenteiden tulkintaa sekä edesauttaa tilastoanalyysin tekemisen tallennetusta verkosta.

## 2 Python-ohjelmointikieli

Pythonin kehittäjä Guido van Rossum kehitti ensimmäisen version Python-kielestä 1980-luvulla. Tällöin kieli valmistettiin olemaan jatkoa ABC-ohjelmointikielelle korjaamalla ABC-kielen sisältämiä virheitä ja lisäämällä muutamia uusia ominaisuuksia. Guido van Rossum toimi Pythonin pääkehittäjänä vuoteen 2018 asti, jolloin hän ilmoitti pitävänsä ”lopullisen loman” tehtävistään ja siirtyi tällöin osaksi viiden henkilön johtoryhmää. [3]

Nykyään Python on monipuolinen, tulkattava avoimella lähdekoodilla oleva ohjelmointikieli. Se on helppo oppia kielen yksinkertaisen syntaksin ja korkeatasoisen tietorakenteen vuoksi. Kielen mukana tulee useita kirjastoja, jotka sisältävät monia eri funktioita ja algoritmeja tiedostojen, tietokantojen ja verkkojärjestelmien hallintaan. Kielelle on myös saatavilla vaatimaan tieteelliseen laskelmointiin olevat NumPy- ja SciPy-kirjastot, jotka mahdollistavat kilpailun MatLab-nimisen tieteellisen ohjelman kanssa. Lisäksi Pythonille on saatavilla useita muiden valmistamia avoimen lähdekoodin omaavia kirjastoja. Pythonia on myös mahdollista laajentaa käyttäen C- ja C++-ohjelmointikieliä. Lisäksi kieli on käytettävissä useilla eri käyttöjärjestelmillä (Windows, MacOS, Linux). [4; 5; 6]

Pythonia on mahdollista käyttää komentoriviltä ajettavissa skripteissä. Kielellä valmistettuja ohjelmia ei tarvitse kääntää ennen suorittamista. Tästä syystä ohjelmoinnin ja testauksen sykli on lyhyt ja ohjelman suoritus onnistuu nopeasti. [4; 5]



### 3 JSON

JSON on ECMA-404 standardilla oleva formaatti, joka on ihmiselle sekä koneelle helposti luettavaa ja kirjoitettavaa. JSON kehitettiin mahdollistamaan reaaliaikainen tukiaseman ja selaimen välinen kommunikointi ilman selaimen lisättäviä liitännäisiä, kuten Flashia tai Javaa, vuoden 2000 alkupuolella. Formaatti on yleisesti käytössä JavaScript-ohjelmointikielessä. Vuonna 2017 useat eri ohjelmointikieliset lisäsivät formaatin kirjoittamisen ja lukemisen osaksi ohjelmointikielten tietokantaa. [7]

JSON tukee useita perusohjelmointikielten käyttämiä tietotyyppisiä, kuten numerot, Unicode-merkkijonot, ryhmiä, Boolean (tosi/epätosi), objekteja ja null. Tietokanta koostuu avaimista ja arvoista, kuten kuvassa 1 oleva "etunimi" on avain ja "Matti" on tämän avaimen arvo. [7]

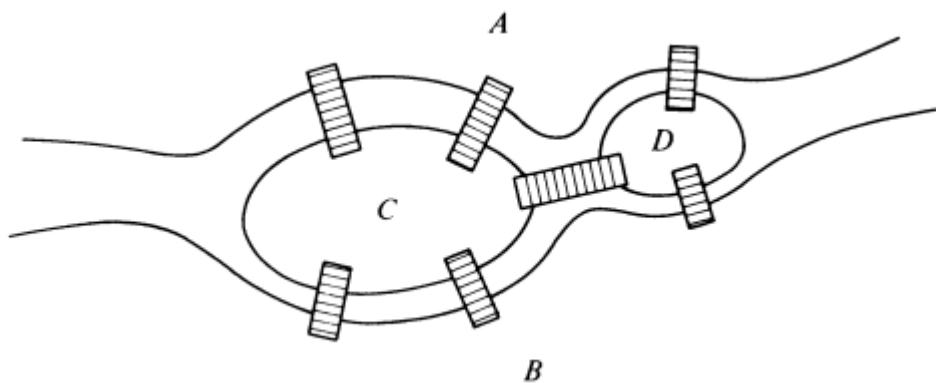
```
{
  "etunimi": "Matti",
  "sukunimi": "Muukalainen",
  "elossa": True,
  "ikä": 30,
  "osoite": {
    "kaupunki": "Helsinki",
    "postiosoite": "12345"
  },
  "vaimo": null,
  "lapset": []
}
```

Kuva 1. Perinteinen JSON-formaatti

Koska JSON-formaatti on tuettu useilla eri kielillä, formaatin käyttö mahdollistaa tiedostojen yhteensopivuuden eri kielillä tehtyjen ohjelmien ja työkalujen kanssa. Formaatin käyttö takaa myös helpon tiedon luettavuuden ihmiselle sekä ohjelmille tai työkaluille.

## 4 Kaavioteoria

Kaavioteoria eli graafiteoria on matematiikan osa-alue, joka käsittelee solmujen yhdistämistä linkejä käyttäen. Teoriaa on ensimmäisen kerran käyttänyt Leonhard Euler vuonna 1736, jolloin hän todisti teorian avulla Königsbergin seitsemän sillan ongelman mahdottomaksi. Ongelman tarkoitus oli luoda reitti, joka kulkee kaupungin läpi ylittämällä jokainen silta yhden kerran ja päätyä takaisin lähtöpisteeseen. Tämä todistettiin mahdottomaksi, koska graafissa on solmuja, joiden linkkien määrä on pariton. Myöhemmin Cauchy ja L'Huilier yleistivät teoriaa matematiikkaan, joka tuli tunnetuksi topologiana. [8]



Kuva 2. Königsbergin seitsemän sillan ongelma [8]

Nykyään teoriaa käytetään tietokonetekniikassa kuvaamaan esimerkiksi kaavioita ja yhteyksiä. Teoriaa on myös mahdollista käyttää fysiikassa, kemiassa, matematiikassa tai muissa aiheissa, joissa kaavio on mahdollista muodostaa.

### 4.1 Verkkoteoria

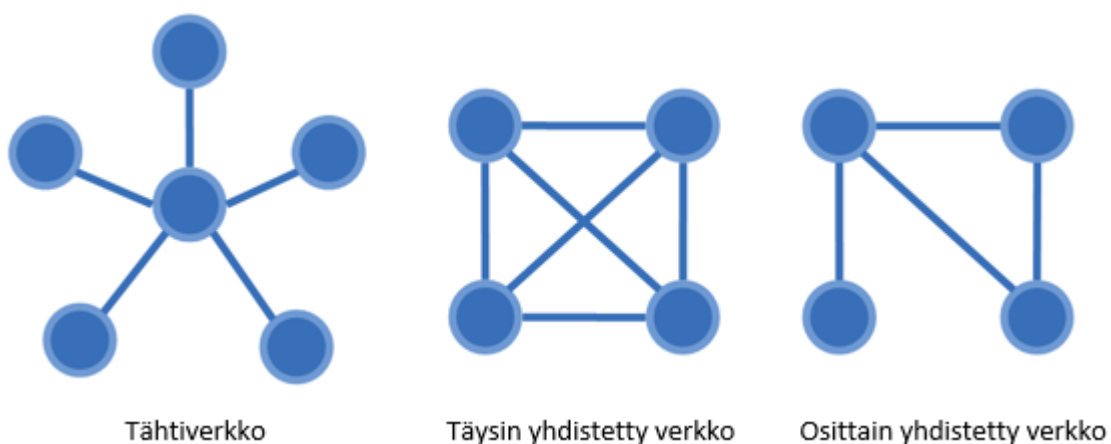
Verkkoteoria on osa kaavioteoriaa, jonka mukaan verkon rakennetta on mahdollista kuvata kaavion avulla. Verkkoja kuvatessa kaavioina linkit ovat langallisessa verkossa käytetyt johtimet ja langattomassa verkossa käytetyt radioaallot, joita käytetään tiedon siirrossa solmujen välillä. Solmut kuvaavat puolestaan laitteita, jotka lähettävät ja vastaanottavat verkossa liikkuvan tiedon linkejä käyttäen. Yleensä verkkokaaviossa solmuina toimivat tietokone, reititin tai kytkin.

Jokainen verkkomalli on mahdollista jakaa yksinkertaisiin tai moniverkkoihin. Yksinkertaisella verkolla tarkoitetaan verkkoa, joka ei sisällä kahden solmun välillä enempää kuin yhden linkin. Moniverkko voi sisältää useita linkkejä kahden solmun välillä. [9]

#### 4.2 Verkkotopologia

Verkkotopologia on listaus verkon sisältämistä tiedosta, kuten linkeistä ja solmuista. Topologia on verkon rakenteen kuvaus fyysisesti tai loogisesti. Fyysinen kuvaus sisältää verkon komponenttien osien tiedot, kuten laitteiden sijainnit. Looginen kuvaus sisältää tiedot verkon sisäisestä liikenteestä. Verkon fyysinen topologia ei välttämättä ole aina sama kuin looginen topologia. [9]

Kuvassa 3 näkyvät yleisesti tunnetuimmat verkon topologiat, jotka ovat tähtiverkko, täysin yhdistetty verkko, osittain yhdistetty verkko ja P2P-verkko. [9]



Kuva 3. Yleiset verkkotopologiatyytit

#### 4.3 Suuntaamaton verkko

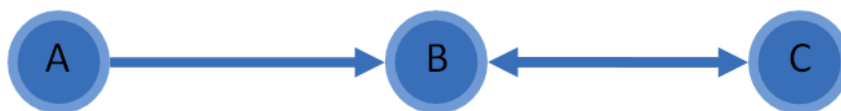
Suuntaamaton verkkomalli tarkoittaa verkkorakennetta, jossa linkki tapahtuu kahden solmun välillä molempien osapuolien osallistumisen perusteella. Tällainen verkko muodostuu esimerkiksi kuvassa 4, kun henkilöt A ja B käättelevät toisiaan. [9]



Kuva 4. Kahden solmun ja yhden linkin suuntaamaton yksinkertainen verkko, P2P-verkko

#### 4.4 Suunnattu verkko

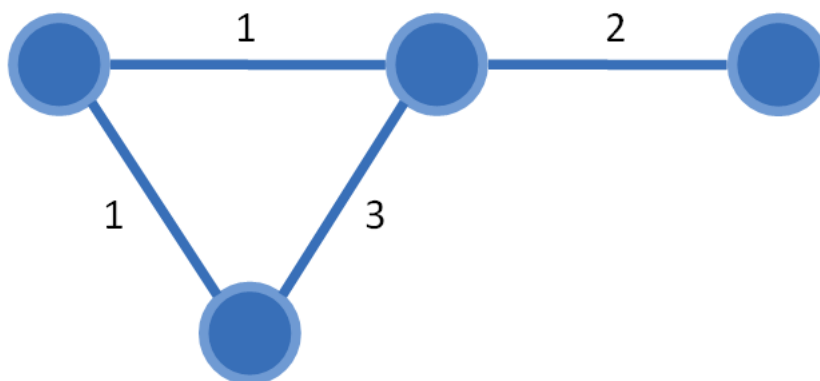
Suunnatulla verkolla tarkoitetaan verkkoa, jonka linkeillä on olemassa suunta solmujen välillä. Tällainen verkko muodostuu esimerkiksi kuvassa 5, kun henkilö A katsoo henkilöä B. Tämä johtuu siitä, että katsominen ei välttämättä vaadi molempien osapuolien osallistumista. Kyseisessä verkossa muodostuu myös suunnattu verkko henkilöiden B ja C välille samasta syystä, vaikka molemmat osapuolet katsoisivat toisiaan. [9]



Kuva 5. Kolmen solmun ja linkin suunnattu moniverkko

#### 4.5 Painotettu verkko

Verkkomallissa määritetään jokaiselle linkille arvo kuvaamaan linkin hintaa, pituutta tai tilavuutta. Tätä mallia käytetään esimerkiksi lyhyimmän matkan ja matkustavan myyjän ongelmissa, joiden tarkoitus on löytää mahdollisimman lyhyt reitti vaadittujen solmujen välillä. [9]



Kuva 6. Neljän solmun ja linkin painotettu yksinkertainen verkko [9]

#### 4.6 Verkon optimointi

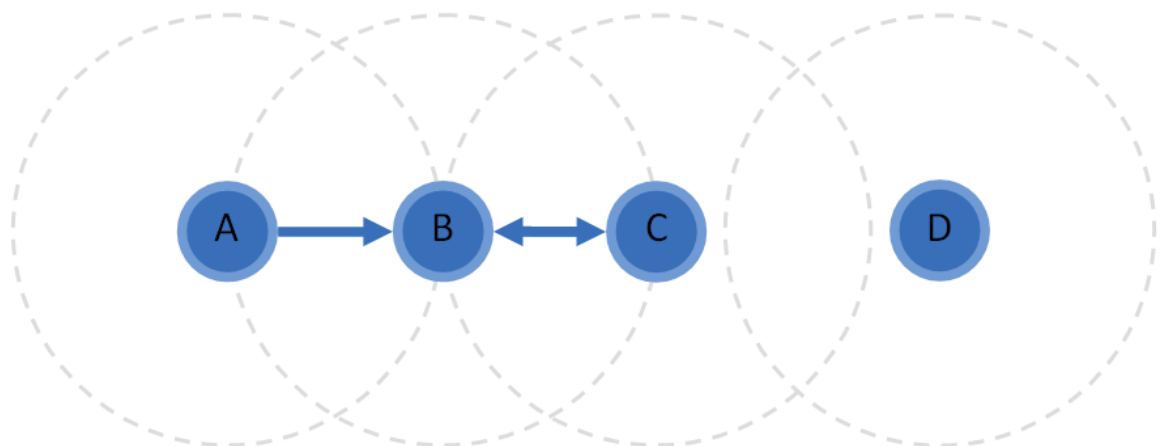
Kaavioteorialla ja verkkoteorialla on myös verkkojen kuvaamisen ja havainnollistamisen lisäksi mahdollista pyrkiä optimoimaan kyseisiä verkkoja. Verkon rakenteen optimoinnissa voidaan käyttää lyhyimmän matkan etsimistä, solmujen sijaintien sekä linkkien analysointia.

## 5 MANET-verkko

MANET on langaton verkko, joka muodostetaan tiettyä tarkoitusta varten. Koska verkko on ad hoc -pohjainen, se ei tarvitse olemassa olevaa infrastruktuuria toimiakseen. Verkossa ei ole tukiasemia vaan jokainen verkon solmu toimii myös reitittimenä. Tämän ansiosta verkon on mahdollista muodostaa vapaasti verkkotopologia, joka voi muuttua arvaamattomasti. [10]

Jokainen verkon solmu kuuntelee toisien solmujen lähettämiä merkkisignaaleja. Samalla solmu lähettää omaa merkkisignaalia määritetyin väliajoin. Kun solmu vastaanottaa toiselta solmulta merkkiviestin, se lisää vastaanotetun solmun naapuritietoihin. Koska verkon solmut lähettävät merkkisignaaleja, jotka eivät välttämättä tarvitse vastausta tai saavuta toista solmua, MANET-verkko on suunnatun verkkomallin mukainen verkko. [10]

Kuvan 7 esimerkissä solmut A, B, C ja D ovat lähettäneet merkkisignaalin (katkoviivalla merkitty alue). Solmu B on vastaanottanut solmujen A ja C lähettämän merkkisignaalin ja on täten tietoinen kyseisten naapurisolmujen olemassaolosta. Lisäksi solmu C on vastaanottanut solmun B lähettämän merkkisignaalin. Solmu A ei kuitenkaan ole saanut syystä  $n$  merkkisignaalia solmulta B. Solmu D on ulkona muiden solmujen lähettämien merkkisignaalien kantamasta, eikä sen kantamassa ole muita solmuja. Tästä syystä solmun D lähettämää merkkisignaalia ei voida vastaanottaa muilla solmuilla, eikä se voi vastaanottaa muiden solmujen lähettämää merkkisignaalia. Myös solmut A ja C ovat ulkona toistensa kantoalueelta. [10]



Kuva 7. MANET-verkon graafin esimerkki

## 6 Bittium TAC WIN -tuoteperhe

TAC WIN -järjestelmän avulla on mahdollista muodostaa nopea langaton IP-verkko tukemaan kommunikointia. Järjestelmä toimii useilla eri taajuuksilla ja verkkorakenteilla, joiden ansiosta se on kustannustehokas ja helppo käyttää. Radioilla etäisyyttä toisistaan voi olla jopa 45 km linkki tai P2P-käytössä. [11]



Kuva 8. Bittium TAC WIN Tactical Router ja Bittium TAC WIN Radio Head I [12]

### 6.1 Bittium TAC WIN Tactical Router

Reititin tukee useita eri langallisia (LAN, SHDSL, valokuitu) ja langattomia yhteyksiä samaan aikaan. Langattomien yhteyksien muodostamiseen käytetään Bittium TAC WIN Radio Head -tuotteita (Radio Head I/III/IV). [11]

### 6.2 Bittium TAC WIN Radio Head

Bittium TAC WIN Radio Head -tuotteet toimivat NATO I-, III- ja IV- taajuusalueilla. Radio Head -tuotteet mahdollistavat MANET-, P2P- ja PMP-langattomien verkkojen muodostamisen. [11]

## 7 Verkonvalvontatyökalu

Verkonvalvontatyökalu on Bittiumin kehittämä ja käyttämä kehitystyökalupaketti, joka mahdollistaa hajautettujen verkkojen tarkastelun visuaalisesti reaaliajassa sekä verkon tallentamisen ja tallenteiden tarkastelun käyttäen verkkoteoriaa (kuva 9). [13]



Kuva 9. Verkonvalvontatyökalu [13]

Tallenteet sisältävät tallennuksen suorittaneen solmun keräämiä tilatietoja verkosta. Tämä mahdollistaa verkon tilan tarkastelemisen useilla eri ajanhetkillä. [13]



## 8 Toteutus

### 8.1 Vaatimusmäärittely

Työkalun tulisi pystyä yhdistämään useita MANET-verkosta tallennettuja tilatiedostoja yhdeksi tietokannaksi. Tallenteet voivat olla otettu vaihtelevilla ajanhetkillä eri puolelta verkkoa. Lisäksi jokin solmu voi sijaita ruuhkaisessa kohdassa verkkoa, jolloin tallennetta varten olevat tiedot voivat olla puutteellisia tai puuttua kokonaan. Työkalulle syötettyjen tiedostojen rakenne tulee olla JSON-formaatin mukainen ja sisältää seuraavat tiedot:

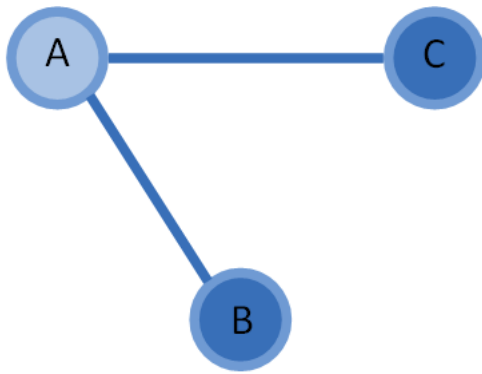
- aikaleimoja, jolloin tilatiedot on kerätty
- solmujen tiedot kyseisillä ajanhetkillä, esimerkiksi tiedossa olevat naapurisolmut
- linkkien tiedot kyseisillä ajanhetkillä, esimerkiksi yhteyden laatu

Yhdistämisen aikana työkalun tulisi huomioida yllä mainittujen tietojen perusteella tietojen laatu ja varmistaa tietojen yhdistyminen oikean aikaleiman mukaan. Työkalun tuottaman yhdistetyn tietokannan tulee olla JSON-formaatin mukainen.

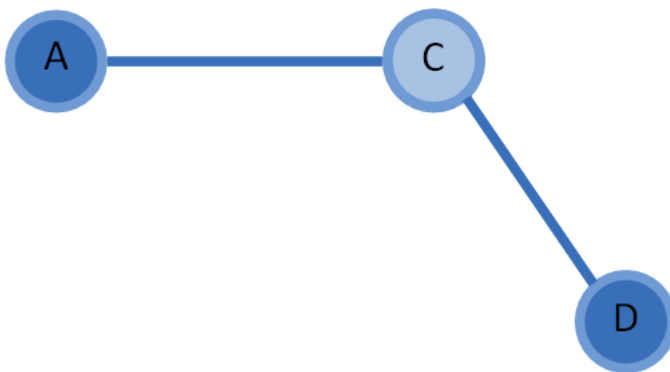
Pelkkä aikaleiman käyttö ei kuitenkaan mahdollista parhaimman laatuista tulosta, koska aika solmujen välillä voi vaihdella. Tämä ei kuitenkaan estä aikaleiman käyttämistä yhdistämisessä. Lisäksi tallenteiden intervalli voi vaihdella, jos se on määritetty eri arvolla solmun aloittaessa tallentamaan verkon tilatietoja.

Lopullisen tietokannan kuuluisi sisältää ajantasaisesti kaikkien solmujen tilatiedot, vaikka linkki solmuun olisi menetetty. Tämä kuitenkin edellyttää, että ainakin yksi tilatallenne sisältää muulloin puuttuvan solmun tilatiedot vaadituille ajanhetkille. Yhdistetty tietokanta vähentäisi myös duplikaattitietoja solmuista ja auttaisi tilatietojen tulkinnassa, koska kaikki tarvittavat tilatiedot olisivat yhdessä tietokannassa.

Esimerkki tilanteesta: Kuvassa 10 on kuvattu solmun A tallentama verkon tila ajanhetkellä  $x$ , ja kuvassa 11 on kuvattu solmun C tallentama verkon tila samalla hetkellä. Kuitenkin solmu A ei ole saanut tilatietoja solmusta D ja solmu E on ollut ulkona verkon kantamasta (kuva 12) ajanhetkellä  $x$ . Näiden kaikkien solmujen tilatiedot tulisi olla saatavilla yhdessä tietokannassa, jotta verkon tilatiedot olisi helpompi tulkita kyseisellä ajanhetkellä (kuva 13).



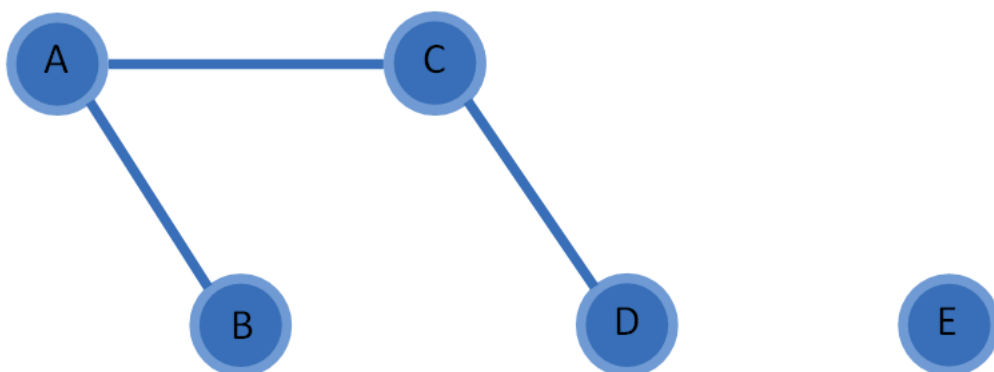
Kuva 10. Solmun A tallentama verkko ajanhetkellä  $x$



Kuva 11. Solmun C tallentama verkko ajanhetkellä  $x$



Kuva 12. Solmun E tallentama verkko ajanhetkellä  $x$



Kuva 13. Yhdistetty tietokanta verkosta, joka sisältää kaikki solmut ajanhetkeltä  $x$

Työkalu tulee toteuttaa käyttäen Python-ohjelmointikieltä (Python 3) ja sen tulee toimia Linux-käyttöjärjestelmässä.

## 8.2 Suunnittelu

Tallenteiden verkkotiedot ovat suunnatun verkon mukaiset, mutta työkalun selkeyden vuoksi näitä tilatietoja käsitellään suuntaamattoman verkon mukaisesti. Tällöin työkalu sallii linkkien muodostumisen, kun molemmat solmut ovat tietoisia toisistaan.

Yhdistämisen tarkentamiseksi käyttäjän olisi hyvä pystyä määrittämään toleranssi ajalle. Tämän jälkeen aikaleimojen käyttö olisi mahdollista tiedostojen yhdistämiseen, käyttäjän määrittämän toleranssin mukaan. Toleranssi myös mahdollistaisi nopeamman ja tarkan suorituksen jos sopivaa aikaleimaa tallenteesta ei ole löydettävissä.

Yhdistettäessä tilatietoja solmun oma tallenne oletetaan sisältävän parhaimman laatuinen tilatieto kyseisestä solmusta halutulta ajanhetkeltä. Lisäksi tulisi selvittää onko tallennetiedostolla tilatietoja solmuista, jotka eivät ole vielä mukana tai tallenteen tilatieto on paremman laatuista kuin tulostiedostossa. Solmun omasta tallenteesta lisättyjä solmun tilatietoja ei saa ylikirjoittaa myöhempien tallenteiden tilatiedoilla.

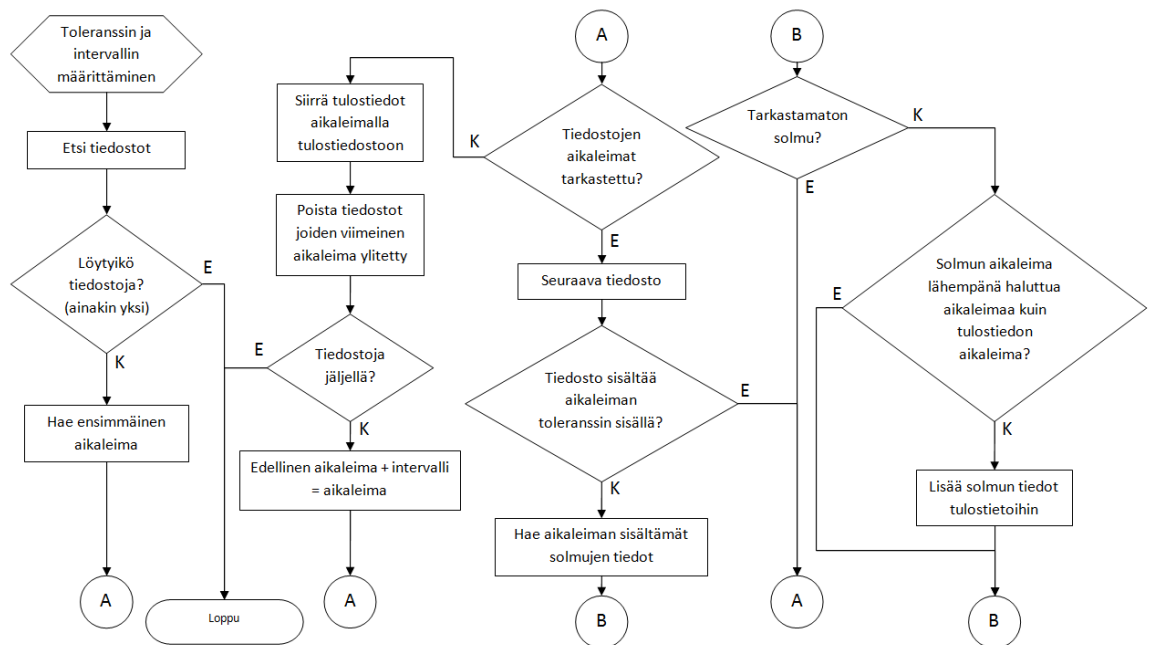
## 8.3 Kehitys

Käyttäjä voi halutessaan määrittää työkalua käynnistäessä intervallin ja toleranssin, joita käytetään tiedostojen yhdistämisessä. Kuitenkin jos käyttäjä ei ole määrittänyt parametria työkalua käynnistäessä, työkalu käyttää alkuperäistä arvoja kyseiselle parametrille. Parametrit on alustettu takaamaan tarkka mutta kuitenkin pieniä heittoja salliva yhdistäminen.

Aluksi työkalu päättelee oikean laatuiset tiedostot käyttäjän määrittämästä kansioista. Määritetyn kansion sisällä on mahdollista olla myös lisää kansioita, joten nämä kuuluu käydä myös läpi. Työkalu tarkistaa tiedoston nimen, tyyppin ja tiedoston sisällön siltä varalta, että tiedosto ei ole jostain syystä tyhjä. Samalla työkalu hakee hyväksytyistä tiedostoista ensimmäisen aikaleiman. Tämän jälkeen yhdistettyyn tietokantaan haluttavaa aikaleimaa kasvatetaan määritetyllä intervallilla. Jos saatavilla olevista tiedostoista ei löydy haluttua aikaleimaa, työkalu etsii seuraavan saatavilla olevan aikaleiman ja alkaa kasvattaa kyseistä aikaleimaa intervallin mukaan.

Tämän jälkeen työkalu alkaa suorittaa tallenteiden yhdistämistä aloittamalla aikaisemmin löydetystä ensimmäisestä aikaleimasta. Työkalu käyttää jokaisesta tiedostosta mahdollisimman lähellä yhdistettävään tiedostoon haluttua aikaleimaa. Jos tiedosto ei sisällä haluttua aikaleimaa

määritetyn toleranssin sisällä, työkalu alkaa tarkastella seuraavaa tiedostoa. Aikaleimoille, jotka ovat määritetyn toleranssin sisällä. Työkalu hakee kaikki solmut, joita ei ole vielä lisätty tulostietoihin tai joiden aikaleima on lähempänä yhdistettävään tiedostoon haluttua aikaleimaa kuin edellinen sallittu aikaleima. Kun kaikki tiedoston aikaleiman sisältämät solmut on tarkastettu, lisätään mahdolliset lisättävät tiedot tulostietoihin. Kun tallenteen viimeinen aikaleima ylittyy, tallenne poistetaan listalta.



Kuva 14. Vuokaavio työkalun logiikasta

## 8.4 Testaus

Testauksia on suoritettu useampia kertoja kehityksen aikana. Kuitenkin tässä osiossa käydään läpi kaksi testiä, jotka on suoritettu kehityksen jälkeen. Testeissä käytetään useita tallenteita usealta eri solmulta, vähintään kaksi tallennetta jokaiselta solmulta.

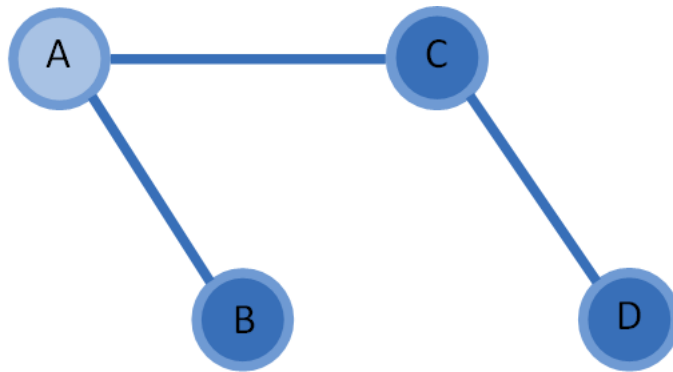
### 8.4.1 Ensimmäinen testaus

Testissä parametreina käytettiin alkuperäisiä arvoja, jotka on määritetty työkaluun. Tällöin arvot ovat 60 sekunnin intervalli ja  $\pm 60$  sekunnin toleranssi.

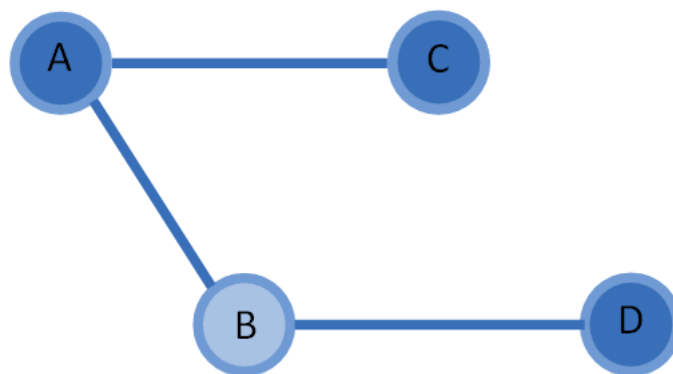
Työkalun antaman palautteen perusteella yhdistäminen onnistui ilman ongelmia. Palautteesta on nähtävillä myös, että jokainen solmu on päivittynyt aikaleimojen mukaisesti sallitun toleranssin mukaan.

Verkonvalvontatyökalussa näkyvillä olevat solmujen tiedot päivittyvät syötetiedostojen mukaan ja aikaisemmin puuttuvat tiedot tukiasemilta ovat saatavilla. Kuitenkin linkit eivät ole aina visuaalisesti oikein. Tämä johtuu solmujen eroavista aikaleimoista, jos näille on määritetty eri kellonaika.

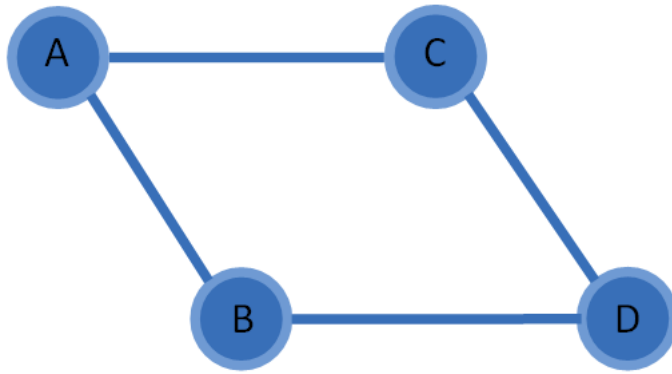
Esimerkki tilanteesta: Solmu A on tallentanut ajanhetkellä  $y$  verkon (kuva 15) ja solmu B tallentanut verkko (kuva 16) on hieman ohi ajanhetkestä  $y$ , mutta kuitenkin sallitun toleranssirajan sisällä, jonka vuoksi yhdistetyssä tietokannassa ajanhetki  $y$  sisältää molempien tallenteiden linkkien tiedot solmulle D (kuva 17).



Kuva 15. Solmun A tallentama verkko ajanhetkellä  $y$



Kuva 16. Solmun B tallentama verkko ajanhetkeltä  $y$  sallitulla toleranssilla

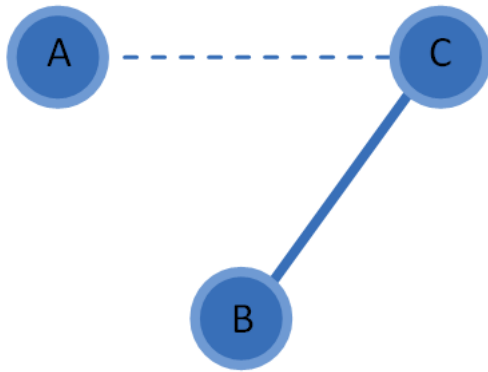


Kuva 17. Yhdistetty tietokanta verkosta ajanhetkeltä  $y$

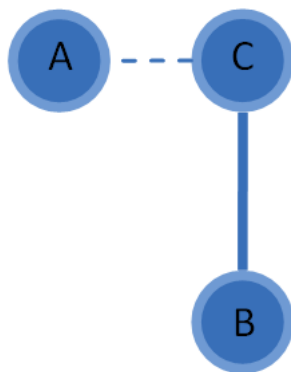
Tämä ongelma on mahdollista korjata siten, että käytetään tallennetta, jonka aikaleima on lähempänä haluttua aikaleimaa. Tällöin solmun B tallentamat solmun D tiedot ja linkki solmuun D hylätään ja yhdistetty tietokanta tulee sisältämään solmun A tallentamat tilatiedot solmusta D ja sen linkeistä.

Yllä mainitun ongelman lisäksi verkkovalvontatyökalulla on havaittavissa, että solmun tilatietojen yhdistämisessä on mahdollista tapahtua virhe. Virheellisten tilatietojen lisäys tapahtuu, kun solmun yhteys palautuu tämän menettämisen jälkeen ja solmulta ei saa olla omaa tallennetta kyseisiltä ajanhetkiltä.

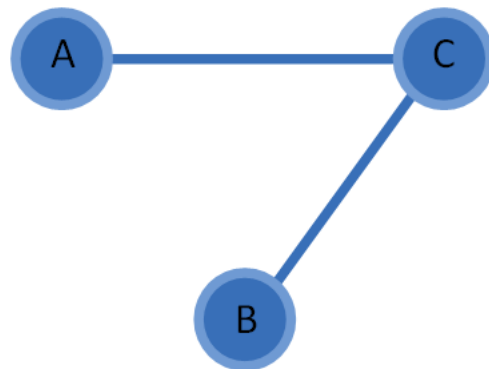
Esimerkki tilanteesta: Solmu A on menettänyt yhteyden solmuun C ajanhetkellä  $z$  (kuva 18). Solmun A tilatallenteen aikaleima on lähempänä haluttua aikaleimaa  $z$  kuin solmun B tilatallenteen aikaleima. Solmulta C ei ole omaa tilatallennetta halutuilta ajanhetkiltä, joten yhdistämiseen käytetään solmujen A ja B sisältämiä tilatietoja solmusta C. Koska solmulla A ei ole yhteyttä solmuun C, käytetään yhdistetyssä tietokannassa (kuva 18) solmun B tallentamia tilatietoja solmusta C. Solmun C tilatiedot vaihtuvat ajanhetkellä  $z + 1$  ennen kuin solmujen A ja C välinen yhteys on saatavilla (kuva 19). Kun solmujen A ja C välinen yhteys on taas saatavilla solmun A tallenteessa ajanhetkellä  $z + 2$ , solmu C päivittyy aluksi tilatietojen mukaan, jolloin A ja C yhteys oli menetetty (kuva 20). Tämän jälkeen ajanhetkellä  $z + 3$  solmun C tilatiedot päivittyvät tietokantaan ajanhetkellisesti oikein, käyttäen solmun A tallennetta (kuva 21).



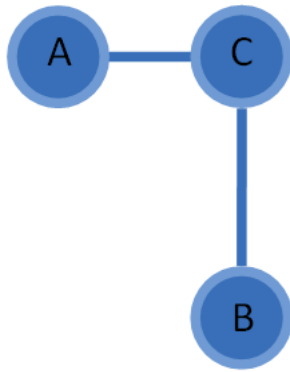
Kuva 18. Yhdistetyn tietokannan verkko ajanhetkeltä  $z$



Kuva 19. Yhdistetyn tietokannan verkko ajanhetkeltä  $z + 1$



Kuva 20. Yhdistetyn tietokannan verkko ajanhetkeltä  $z + 2$



Kuva 21. Yhdistetyn tietokannan verkko ajanhetkeltä  $z + 3$

Korjaus tilanteelle on hylätä solmun C tilatieto solmun A tallenteesta ja käyttää tilatietoa solmun B tallenteesta ajanhetkellä  $z + 2$ .

#### 8.4.2 Toinen testaus

Testissä käytettiin 300 sekunnin intervallia ja  $\pm 300$  sekunnin toleranssia.

Työkalun antaman palautteen perusteella yhdistäminen onnistui ilman ongelmia. Jokainen solmu on päivittyneet viiden minuutin välein tulostiedostoon.

Testissä ilmenee sama linkkiongelma kuin ensimmäisessä testissä. Tätä on kuitenkin osittain hankala huomata suuremman intervallin vuoksi. Tilatieto-ongelma ei kuitenkaan testissä ilmene. Tämä johtuu tietokannan luomisesta suuremmalla intervallilla, jonka takia ensimmäisessä testissä ollut ongelma ei ole näkyvillä. Kuitenkin tilatieto-ongelma on mahdollinen, mutta se on harvinaisempi suuren intervallin vuoksi.



## 9 Tulokset

### 9.1 Tulosten analysointi

Työkalu mahdollistaa useiden MANET-verkkojen tilatallennetiedostojen yhdistämisen yhdeksi tietokannaksi. Yhdistetyssä tietokannassa on nähtävissä pieniä eroja aikojen välillä, mutta tämä johtuu pääasiassa toleranssin sallimista arvoista.

Kokonaisen MANET-verkon tilan tulkintaan vaadittavien tallenteiden vaatima koko vähentyy yhdistämisen avulla jopa 10 %:iin alkuperäisestä koosta. Tilatietojen tulkinta onnistuu tämän ansiosta myös nopeammin.

Intervallin ollessa määritettynä liian suureksi verkon tulkinnasta tulee hankalaa. Intervallin määrittäminen liian pienellä arvolla pidentää työkalun suoritusaikaa; suurempaa haittaa asiasta ei aiheudu. Tämä johtuu siitä, että jokainen tallenne täytyy tarkistaa, sisältävätkö nämä haluttua aikaleimaa. Toleranssin ollessa määritetty liian suureksi ongelmaa ei tulisi syntyä, koska työkalu pyrkii käyttämään mahdollisimman lähellä haluttua aikaleimaa olevaa aikaleimaa. Kuitenkin toleranssin ollessa määritetty liian pieneksi on mahdollista, että tilatietoja putoaa pois, jos intervalli on myös suuri.

Työkalun on mahdollista muodostaa kokonainen kuva verkosta (linkkien kanssa), vaikka tallenteet sisältäisivät vain tallennuksen tehneen solmun, linkkien ja naapurisolmujen tiedot. Jos tallenteiden sisältöä on rajattu vain solmun omiin tietoihin, jäävät linkkien tiedot pois. Kuitenkin tietokannan luominen tulisi olla mahdollista, mutta ilman linkkien tietoja.

### 9.2 Käytettävyys

Työkalun käyttö vaatii käyttäjältä vähintään syötekansion, joka sisältää tarvittavat tallennetiedostot verkosta. Lisäksi valinnallisia parametreja ei ole useita, joten työkalun käyttö on kohtuullisen yksinkertainen. Työkalu toimii komentoriviltä ja käyttäjä voi kerätä halutessaan työkalun antamat tiedot suorituksesta tiedostoon talteen. Työkalu on testattu toimivaksi ainoastaan Linux-käyttöjärjestelmällä.

### 9.3 Ongelmakohdat

Työkalun suorittamiseen kuluu paljon aikaa, mutta tämä johtuu pääasiassa suuresta määrästä yhdistettäviä tallennetiedostoja. Työkalua on mahdollista nopeuttaa käyttämällä monisäikeistä suoritusta yksisäikeisen suorituksen sijaan, joka vaatii eri ohjelmointikielen käyttöä.

Lisäksi jos tallenteissa on solmu, jolla ei ole olemassa omaa tallennetta, saa solmu tietonsa lähimpänä haluttua aikaleimaa olevasta tallenteesta. Tämä aiheuttaa ongelman solmun tilatietojen kanssa, kun lähimpänä haluttua aikaleimaa oleva tallenne menettää yhteyden vaadittuun solmuun.

### 9.4 Jatkokehitys

Työkalu olisi hyvä tarkistaa aikaleimat naapurisolmujen osalta, jos sama naapuri on saamassa arvoja kahdesta eri tallenteesta, jotta olisi mahdollista käyttää lähimpänä haluttua aikaleimaa olevaa vaihtoehtoa.

Työkalun suoritusta olisi myös mahdollista nopeuttaa tekemällä työkalusta monisäikeinen, jolloin käytettävä ohjelmointikieli olisi vaihdettava.

Työkalun käyttöliittymää olisi mahdollista kehittää niin, että nähtävissä olisi arvioitu yhdistämisen kesto ja kuinka paljon yhdistämistä on jo suoritettu.

Lisäksi tiedostojen yhdistäminen yhdeksi tietokannaksi edesauttaa tilastoanalyysien tekemistä tallennettujen verkkojen tiloista.

## 10 Yhteenveto

Työn tavoitteena oli kehittää työkalu tukemaan MANET-verkkojen solmujen tallentamien tilatiedostojen tulkintaa, yhdistämällä nämä yhdeksi tietokannaksi. Työ oli pääasiassa ohjelmointi- ja testauspainotteinen, koska työkalu täytyi tehdä alkupisteestä lähtien ja sen tuottamien tietokantojen oli oltava yhteensopivia olemassa olevan työkaluketjun kanssa. Tätä prosessia nopeutti Python-ohjelmointikielen käyttö, kielen nopean ohjelmoinnin ja testauksen syklin ansiosta.

Työtä varten tarvittavat työkalut, tiedot ja tallennetiedostot olivat hyvin saatavilla Bitiumilta. Aikaisemmin kerättyjen tallenteiden avulla oli mahdollista testata ja todeta työkalun toiminta toivotuilla ominaisuuksilla.

Työn lopputuloksista havaittiin, että työkalu suoriutui vaaditusta tehtävästä. Samalla havaittiin, että työkalua on mahdollista jatkokehittää, jotta tiedostojen yhdistämisestä saadaan tarkennettua, nopeutettua sekä tekemällä enemmän käyttäjäystävällinen käyttöliittymä. Lisäksi työkalu edesauttaa tilastoanalyysin tekemistä tallennetuista verkoista.

## Lähteet

1. Kansikuva. Internet lähde. Saatavilla:  
[https://en.wikipedia.org/wiki/File:Social\\_Network\\_Analysis\\_Visualization.png](https://en.wikipedia.org/wiki/File:Social_Network_Analysis_Visualization.png) Viitattu:  
29.05.2019
2. Bittium. Company overview. Internet lähde. Saatavilla:  
<https://www.bittium.com/about-bittium/facts-figures/company-overview> Viitattu:  
08.05.2019
3. Python-ohjelmointikieli. Yleistä tietoa. Internet lähde. Saatavilla:  
[https://en.wikipedia.org/wiki/Python\\_\(programming\\_language\)](https://en.wikipedia.org/wiki/Python_(programming_language)) Viitattu: 10.05.2019
4. Python. Python 2 documentation. Internet lähde. Saatavilla:  
<https://docs.python.org/2/index.html> Viitattu: 14.05.2019
5. Python. Python 3 documentation. Internet lähde. Saatavilla:  
<https://docs.python.org/3.5/index.html> Viitattu: 14.05.2019
6. Shell, Scott (17.05.2014). An Introduction to Python for scientific computing. Internet lähde. Saatavilla: <https://sites.engineering.ucsb.edu/~shell/che210d/python.pdf> Viitattu: 10.05.2019
7. JSON. Introducing JSON. Internet lähde. Saatavilla: <http://json.org/> Viitattu: 21.05.2019
8. Kaavioteoria. Internet lähde. Saatavilla:  
<https://www.edutechlearners.com/download/Graphtheory.pdf> Viitattu: 15.05.2019
9. Kimmo Soramäki, Samantha Cook. Network theory and Financial risk, eBook. Saatavilla:  
[https://www.amazon.com/Network-Theory-Financial-Kimmo-Soramaki-ebook/dp/B01D370ZU6/ref=tmm\\_kin\\_swatch\\_0?encoding=UTF8&qid=&sr=](https://www.amazon.com/Network-Theory-Financial-Kimmo-Soramaki-ebook/dp/B01D370ZU6/ref=tmm_kin_swatch_0?encoding=UTF8&qid=&sr=) Viitattu:  
10.04.2021
10. Yan Zhang, Jijun Luo, Honglin Hu. Wireless mesh networking, eBook. Saatavilla:  
[https://www.amazon.com/Wireless-Mesh-Networking-Architectures-Communications-ebook-dp-B009AIU4YK/dp/B009AIU4YK/ref=mt\\_other?encoding=UTF8&me=&qid=](https://www.amazon.com/Wireless-Mesh-Networking-Architectures-Communications-ebook-dp-B009AIU4YK/dp/B009AIU4YK/ref=mt_other?encoding=UTF8&me=&qid=) Viitattu: 24.02.2021
11. Bittium. Products overview. Internet lähde. Saatavilla:  
<https://www.bittium.com/index.php?id=1405#> Viitattu: 15.05.2019
12. Bittium. Tactical IP networks. Internet lähde. Saatavilla:  
<https://www.bittium.com/tactical-communications/tactical-ip-networks> Viitattu:  
14.05.2019
13. Verkkovalvontatyökalu. Internet lähde. Saatavilla:  
<https://www.bittium.com/index.php?id=1405#network-manager> Viitattu:13.05.2019