

**Mat-2.177 Operaatiotutkimuksen  
projektityöseminaari**

**Viestiverkon toimintaluotettavuuden arviointi**

**Loppuraportti 7.5.2006**

---

**Kohdeorganisaatio:**

Puolustusvoimien Teknillinen Tutkimuslaitos

**Yhteyshenkilö:**

Esa Lappi, esa.lappi@pvtt.mil.fi

**Ryhmä:**

Tuukka Sarvi (projektipäällikkö) 57679S

Juha Saloheimo 57739V

Jaakko Niemi 55114S

Tapio Heimo 76453U

# Sisällysluettelo

Johdanto .....	3
Työn lähtökohta ja tavoitteet .....	3
Raportin sisältö.....	3
Kohdeorganisaatio.....	4
Kirjallisuuskatsaus.....	4
Yleistä verkoista matematiikassa .....	4
Puolustusvoimien viestiverkot.....	5
Prikaatin viestiverkko .....	6
Elektroninen sodankäynti .....	7
Malli yhteyksien toimintatodennäköisyyksien määrittämiseksi.....	9
Yleistä Monte-Carlo-menetelmistä .....	9
Toimintatodennäköisyydet määrittävät algoritmit .....	10
lue_verkko.m .....	11
laske_TNmatriisi.m.....	12
laske_TN.m .....	13
Algoritmien vaatima laskenta-aika.....	14
Varianssianalyysi: tulosten epävarmuus.....	15
Tulosten epävarmuuden riippuvuus simulaatiokierrosten lukumäärästä.....	16
Epävarmuuden korrelaatio vastaavan yhteystodennäköisyyden kanssa.....	17
Malli tukiasemien tärkeyden määrittämiseksi keskeisyysmitoilla .....	20
Painojen laskeminen.....	20
Strength –keskeisyys.....	22
Lyhyimmän polun laskeminen .....	22
Closeness –keskeisyys .....	23
Betweenness –keskeisyys .....	23
keskeisyys.m.....	24
Mallin ominaisuuksien analyysi yksinkertaisessa verkossa .....	24
Sovellusesimerkki luotujen työkalujen käytöstä .....	29
Pohdinnat ja yhteenveto .....	36
Lähdeluettelo .....	37
Liitteet .....	38

Liite 1: MATLAB-tiedosto lue_verkko.m.....	38
Liite 2: MATLAB-tiedosto laske_TNmatriisi.m.....	39
Liite 3: MATLAB-tiedosto laske_TN.m .....	40
Liite 4: MATLAB-tiedosto keskeisyys.m .....	42
Liite 5: Kohdeorganisaatiolta saatu esimerkkiverkko.....	46

# Johdanto

## ***Työn lähtökohta ja tavoitteet***

Tämän projektityön tarkoituksena on tarkastella Suomen Puolustusvoimien viestiverkkojen toimintaluotettavuutta. Projekti on osa Teknillisen korkeakoulun kurssin Mat-2.177 Operaatiotyötutkimuksen projektityöseminaaria. Ongelmanasettajana toimii Puolustusvoimien Teknillinen Tutkimuslaitos (PvTT).

Viestiverkko on kriittinen osa tämän päivän sodankäyntiä. Operaatioiden onnistumisen edellytyksenä on usein, että komentopaikalta on toimiva viestiyhteys operaatioon osallistuviin joukkoihin. Hyökkäävän osapuolen on otettava suunnitelmissaan huomioon viestiverkon hyökkäykselle asettamat rajoitukset ja riskit. Ei ole mitään järkeä toteuttaa hyökkäysoperaatiota, jonka onnistuminen edellyttää viestiverkon toimivuutta, jos viestiverkon toiminta on hyvin epävarmaa tai riippuu muutaman helposti tuhottavan komponentin toiminnasta. Puolustavalle osapuolelle on vastaavasti erittäin hyödyllistä tietää toisen osapuolen viestiverkon heikot kohdat, joiden tuhoaminen tai toiminnan estäminen vaikeuttaa hyökkäystä.

Projektityön tavoitteena on luoda Puolustusvoimien käyttöön malleja ja työkaluja, jotka edesauttavat viestiverkon toimintaluotettavuuden arviointia ja viestiverkon tärkeimpien osien löytämistä. Tarkemmin määriteltynä projektin tavoiteasettelu on seuraava:

- Ensimmäinen tavoite: rakenna ja koodaa malli, jonka avulla voidaan määrittää annetun viestiverkon toimintatodennäköisyys, ts. todennäköisyys, että solmusta A saadaan yhteys solmuun B.
- Toinen tavoite: rakenna ja koodaa malli, jonka avulla voidaan tarkastella eri solmujen keskeisyyttä viestiverkon toiminnan kannalta.
- Muut tavoitteet: tuottaa hyödyllistä tietoa ja johtopäätelmiä viestiverkon toiminnasta viestiverkon rakenteen suunnittelua ja elektronista sodankäyntiä ajatellen.

## ***Raportin sisältö***

Raportti on suunnattu ensisijassa matematiikkaan perehtyneille, mutta olemme pyrkineet kirjoittamaan raportin siten, että matematiikkaan syvällisesti perehtymätönkin pystyy omaksumaan mallien peruseräatteen ja käytön. Tätä edesauttavat vaihe vaiheelta läpikäytävät käyttöesimerkit.

Raportti sisältää kirjallisuuskatsauksen Puolustusvoimissa käytettävissä olevista viestiverkoista sekä lyhyen johdannon matemaattiseen verkkoteoriaan. Raportissa käsitellään kahta luotua mallia, viestiverkon toimintatodennäköisyydet määrittävää mallia ja solmujen keskeisyyttä mittaavaa mallia. Luodut mallit vastaavat edellä esitettyä tavoiteasettelua.

Ensin käsitellään toimintatodennäköisyydet määrittävää mallia. Malli johdetaan Monte Carlo –simulaatiosta käsin, luodut algoritmit käydään läpi, tutkitaan algoritmien antamien tulosten virheitä sekä niiden laskenta-aikavaatimuksia. Solmujen keskeisyyttä mittaava malli johdetaan graafiteoriassa määritellyistä keskeisyysmitoista. Luotu malli käydään

kohta kohdalta läpi. Mallin ominaisuuksia analysoidaan yksinkertaisessa verkossa. Lopuksi on vielä sovellusesimerkki, joka valottaa sekä todennäköisyysmalliin että keskeisyysmalliin liittyvien työkalujen käyttöä ja niistä saatavien tulosten tulkintaa.

Raportissa pohditaan myös mallien hyödyllisyyttä käytännössä. Yritetään saadun datan valossa vetää johtopäätöksiä kuinka hyödyllisiä mallit ovat oikeita sotilasoperaatioita ajatellen. Tämän perusteella mietitään mallien puutteita ja mahdollisia jatkokehityskohteita.

## ***Kohdeorganisaatio***

Puolustusvoimien Teknillinen Tutkimuslaitos PvTT on pääosin pääesikunnan alainen osa Suomen puolustusvoimia, jonka tehtävänä on tuottaa puolustusvoimien ja maanpuolustuksen kehittämiseksi ja päätöksenteon tueksi elektroniikan ja informaatiotekniikan, suojan sekä asetekniikan tutkimuspalvelut. PvTT keskittyy niille tutkimuksen osa-alueille, joilla ei ole osaamista muissa tutkimuslaitoksissa, tai joilla puolustusvoimien oma tutkimustoiminta on välttämätöntä. Laitoksen tutkimuskohteina ovat aseet ja ampumatarvikkeet, räjähteet, sotavarustuksessa käytettävät materiaalit, epäkonventionaaliset aseet ja niiltä suojautuminen, optroniikka, häiveteknologia sekä elektroniikka- ja informaatioteknologia. Tämän projektityön tavoitteena on tukea elektroniikka- ja informaatiotekniikan osaston tutkimusta.

## **Kirjallisuuskatsaus**

Kirjallisuuskatsauksessa on lyhyt katsaus matemaattiseen verkkoteoriaan eli graafiteoriaan siltä osin kun sitä tässä työssä käytetään. Tämän jälkeen käsitellään Puolustusvoimien viestiverkkoja käytännön tasolla ja määritellään niihin liittyviä käsitteitä sekä kerrotaan tarvittavia taustatietoja esimerkiksi elektronisesta sodankäynnistä.

### ***Yleistä verkoista matematiikassa***

Verkko (so. graafi) koostuu solmuista ja niitä yhdistävistä linkeistä. Yksinkertaisimmassa tapauksessa verkko on suuntaamaton, eli jos solmusta  $i$  on linkki solmuun  $j$ , on myös solmusta  $j$  linkki solmuun  $i$ . Tässä työssä tarkastellaan kuitenkin suunnattuja verkkoja. Matemaattisesti verkko on kätevää esittää yhteismatriisilla  $A$ , jonka alkio  $A_{ij} = 1$ , jos solmusta  $i$  on linkki solmuun  $j$ . Verkon linkeihin voidaan myös liittää paino, jolloin puhutaan painotetuista verkoista. Yhteismatriisi on edelleen kätevä esitystapa, alkio  $A_{ij}$  vastaa nyt  $i$ :stä  $j$ :hin kulkevan linkin painoa. Tässä työssä linkkien painot vastaavat yhteystodennäköisyyksiä solmujen välillä. Toisinaan verkot on järkevää määritellä siten, että myös rinnakkaiset linkit ovat mahdollisia. Tällöin siis solmujen  $i$  ja  $j$  välillä voi olla useita linkejä. Näitä emme kuitenkaan tarvitse tässä projektityössä.

Matemaattisessa graafiteoriassa ei suoraan käsitellä tapausta, jossa eri solmujen toiminta on epävarmaa. Tässä työssä solmuillakin (eri pisteissä sijaitsevilla radiolla) on toimintatodennäköisyys. Tätä tietoa ei voi sisällyttää eri linkkien toimintatodennäköisyyksiin. Solmujen epävarma toiminta on siis otettava huomioon erikseen.

Verkoissa on hyödyllistä määritellä solmujen etäisyys linkkien painojen avulla. Esimerkiksi keskeisyystarkastelut vaativat tämän lähestymistavan. Painottamattomassa verkossa solmujen  $i$  ja  $j$  välisellä etäisyydellä tarkoitetaan pienintä mahdollista määrää linkkejä, joiden läpi kulkemalla päästään solmusta  $i$  solmuun  $j$ . Painotetuissa verkoissa etäisyys on hieman monimutkaisempi käsite. Linkin paino määritellään nimittäin lähes aina siten, että suuri paino viittaa voimakkaaseen yhteyteen linkin päätesolmujen välillä. Tämän takia linkin päätesolmujen välistä etäisyyttä ei ole järkevää määritellä linkin painona. Jos linkkien painot ovat mielivaltaisia positiivisia reaalitykkuja, määritellään linkin päiden välinen etäisyys usein painon käänteislukuna. Tässä projektityössä tarkastellaan kuitenkin verkkoja, joissa linkkien paino vastaa todennäköisyyttä, että lähetetty viesti kulkee linkin läpi, eli se on aina nollan ja yhden välillä. Todennäköisyys, että viesti kulkee kahden peräkkäisen linkin läpi, on linkkien painojen tulo. Tämän takia linkin päätesolmujen välinen etäisyys eli linkin pituus on järkevää määritellä painon käänteisluvun luonnollisena logaritmina, sillä logaritmiltaan tunnetusti pätee  $\log(ab) = \log(a) + \log(b)$ . Nyt mielivaltaisen solmun  $i$  etäisyys toisesta mielivaltaisesta solmusta  $j$  voidaan määritellä pienimpänä mahdollisena linkkien ”pituuksien” summana kuljettaessa  $i$ :stä  $j$ :hin.

### ***Puolustusvoimien viestiverkot***

Monipuoliset ja toimivat viestiyhteydet ovat ehto puolustusvoimien tehokkaalle toiminnalle. Puolustusvoimat tarvitsevat viestiyhteyksiä operaatioiden tukemiseen, joukkojen komentamiseen ja valvomiseen ja riittävän tilannetietoisuuden saavuttamiseen. Seuraava katsaus puolustusvoimien viestiverkkoihin perustuu pääosin artikkeliin [1].

Taisteluyksiköiden yhteistoiminta viestiyhteyksien avulla on ratkaisevaa digitaalisella taistelulentäällä. Suomen puolustusvoimat aloittivat 1990-luvun alussa ohjelmistoradioiden (SDR) tutkimuksen ja kehittämisen tulevaisuuden radiosysteemejä varten.

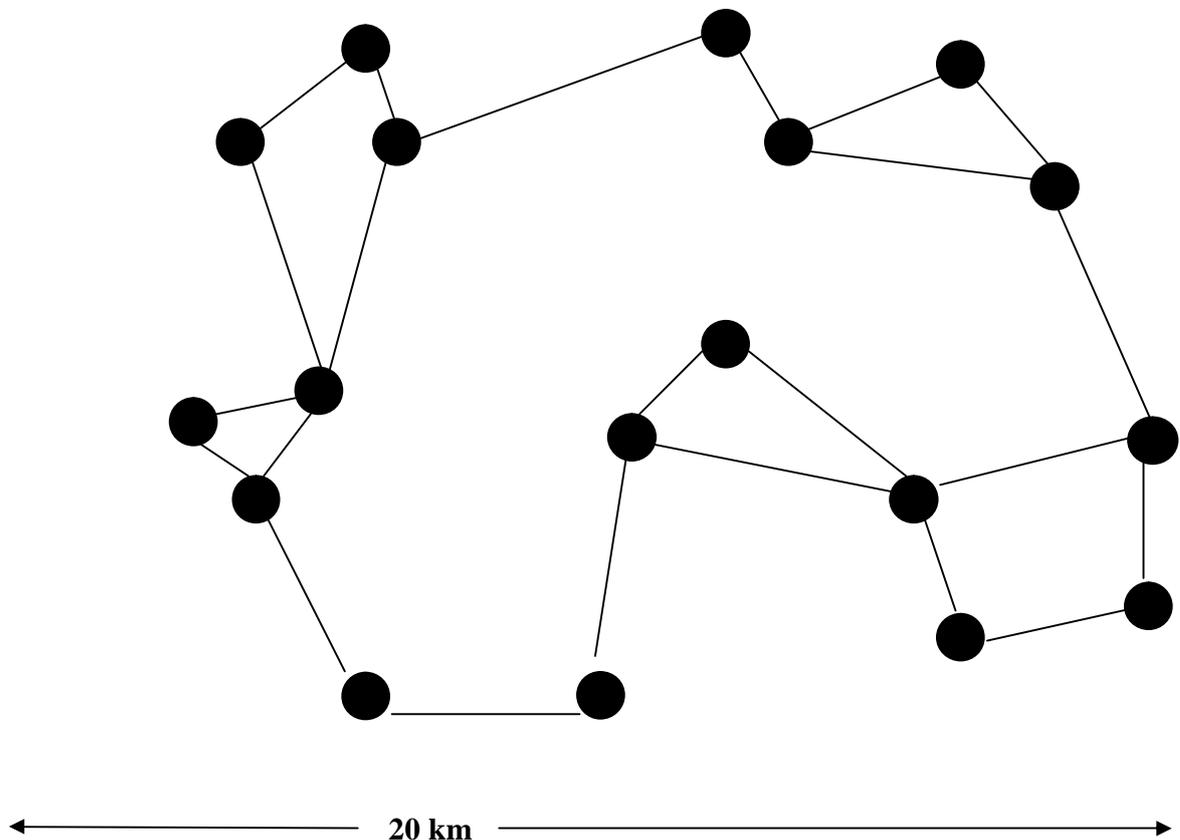
Ohjelmistoradiot muodostavat Ad Hoc –viestiverkon eli kommunikointi on langatonta verkon laitteiden välillä. Verkko muodostuu dynaamisesti langattomien laitteiden välille ilman etukäteen konfiguroitua rakennetta. Tällaisessa verkossa kommunikointi niiden laitteiden välillä, joihin ei ole suoraa radioyhteyttä, tapahtuu välisolmujen kautta. Ad Hoc –verkko muodostaa automaattisesti viestiliikenneyhteyden lähettävän ja vastaanottavan solmun välille ja viesti voi kulkea tällöin kaikkia mahdollisia viestiverkon reittejä pitkin. Ad hoc –verkkotekniikka mahdollistaa myös viestiverkon muodostamisen liikkuvien solmujen välille, mikä edesauttaa viestiyhteyksien ylläpitämisen liikkeessä olevien joukkojen välillä. Tämä on Ad Hoc –verkkojen yksi suurimmista eduista verrattuna vanhanaikaisiin staattisiin verkkoihin, jotka eivät toimi liikkeessä.

Ohjelmistoradio mahdollistaa eurooppalaisen kriisinhallinnan yhteisoperaatiot. Ohjelmistoradion merkitys on keskeinen puolustushaara-, aselaji- ja kansallisuusrajat ylittävässä yhteistoiminnassa ja yhteisoperaatioiden toteuttamisessa tulevaisuuden taistelulentäällä, sillä se tekee eri aselajien ja puolustushaarojen välisen yhteydenpidon helpommaksi. Se on osa verkkokeskeisen sodankäynnin mahdollistavia teknologioita ja tukee puolustusvoimien integroidun tiedustelu-, valvonta- ja johtamisjärjestelmän kehittämistä.

Puolustusvoimien ohjelmistoradio -hankkeessa kehitetään puolustusvoimille kaupallisia komponentteja hyödyntävä ohjelmistoradioalusta, suuren tiedonsiirtokyvyn mahdollistava mukautuva laajakaistainen aaltomuoto, paikannusaaltomuoto sekä adaptiivinen antennijärjestelmä.

### Prikaatin viestiverkko

Seuraavaksi tarkastellaan prikaatin viestiverkon toimintaa vihamielisessä ympäristössä. Prikaatin Ad Hoc -verkon solmuissa on digitaaliset VHF-radiot prikaatin sisäiseen kommunikaatioon ja yhteistoimintaan siviilipuolen kanssa käytetään yleistä matkapuhelinverkkoa (UMTS). Kuvassa 1 on esitetty yksi esimerkki prikaatin viestiverkosta, jossa joukot ovat jakautuneet 20 x 20 km alueelle. Mustat ympyrät kuvastavat verkon tukiasemia ja viivat tukiasemien välille muodostuneista yhteyksistä. Yhteydet voivat olla vain toispuoleisia riippuen solmujen radioiden ominaisuuksista.



Kuva 1. Esimerkki prikaatin viestiverkosta

Viestintävälineet on kiinnitetty taistelujoneuvoihin, jotka voivat olla Ad Hoc – verkkotekniikan ansiosta liikkeessä. Jokainen solmu tukee kommunikaatiota muiden solmujen ja muiden päällekkäisten verkkojen kanssa yhteyden muodostuessa

automaattisesti solmujen välille. Eri siviiliverkkojen avulla (UMTS) tuetaan asevoimien ja siviilien yhteistoimintaa. Viestiverkon digitaaliset VHF-radiot voivat operoida joko yhdellä kanavalla tai taajuushyppytilassa. Jälkimmäisessä tilassa radio muuttaa taajuutta noin 100 kertaa sekunnissa VHF-radion taajuusalueella vaikeuttaen viestiverkon häirintää.

## **Elektroninen sodankäynti**

Elektroninen sodankäynti muodostuu kolmesta eri osa-alueesta: elektroninen tuki, elektroninen hyökkäys ja elektroninen suojele. Elektronisilla tukitoimilla tarkoitetaan elektromagneettisen energian lähteiden etsintään, tunnistamista ja paikantamista. Elektroninen hyökkäys tarkoittaa elektromagneettisen energian käyttöä hyökkäyksissä eri kohteita vastaan. Elektronisella suojelella pyritään henkilöiden, varusteiden ja kiinteiden kohteiden suojelemaan elektronisia hyökkäyksiä vastaan.

## **Viestiverkon häirintä**

Viestiverkkoa voidaan häiritä monilla eri menetelmillä. Häirinnän päätyypit ovat spot-, response-, swept-, comb- ja barrage - häirintä.

Spot-häirintää harjoittaessa häirintä kohdistetaan yhdelle kanavalle ja häirintäsignaalin kaistaleveys on tavallisesti sama kuin häiritävän kanavan kaistaleveys. Häirinnässä käytetään suunnattavia antennia, joilla maksimoidaan häirinnän vaikutusta viestin vastaanottajassa. Yhden kanavan häirintä maksimoi häirintävaikutusta samalla kanavalla operoiviin vastaanottajiin. Response-häirinnässä häiritsijä käyttää etsintävastaanotinta ja häirintää toteutetaan vain kun viestiliikennettä havaitaan tarkkailtavalla kanavalla.

Barrage-häirinnässä häirintäsignaali jakautuu tietyille taajuusalueelle usean eri kanavan ylle. Barrage-häirintä tarvitsee onnistuakseen vain vähän informaatiota häiritävän systeemin käyttämistä taajuuksista, mutta toisaalta häirinnän teho yhtä kanavaa kohden ei ole yhtä korkea kuin spot – häirinnässä, koska häirinnän teho jakautuu tasaisesti häiritävän taajuusalueen ylle. Comb-häirinnässä keskitytään häirinnässä samanaikaisesti useaan eri kanavaan. Tässä tapauksessa häirintäteho on korkeampi kuin barrage-häirinnässä, koska häirintä keskittyy tiettyihin kanaviin barrage-häirinnässä käytettävän taajuusalueen sijaan.

Swept -häirintä operoi tietyillä lähekkäisillä kanavilla ja häirintää toteutetaan suunnattujen antennien avulla. Swept-häirintä häiritsee vain yhtä kanavaa kerrallaan ja koko häirintäsignaalin lähettäjän teho keskittyy yhdelle kanavalle.

## **Häirintälaitteiden sijoitus**

Häirintäsignaalin lähettäjä voidaan sijoittaa monella tavalla. Eri sijoitusvaihtoehdot asettavat kuitenkin rajoitteita käytettävälle häiritsijälle painon, tilavuuden ja liikkuvuuden suhteen.

Yksi mahdollisuus on sijoittaa häirintälähetin taistelujoneuvoon. Ajoneuvoihin voidaan sijoittaa suhteellisen raskaita ja tehokkaita häirintä systeemejä verrattuna muihin liikkuviin ratkaisuihin. Ajoneuvojen liikkuvuus on kuitenkin rajoitettua taistelukentällä.

Häiritsijän kantamaa rajoittavat maastonmuodot erityisesti operoitaessa korkeilla taajuuksilla. Tavallisesti ajoneuvoissa käytettävien häirintälähettimien teho on 1 – 10 kW.

Toisaalta häirintälaitteisto voidaan sijoittaa helikopteriin tai miehittämättömään/normaaliin henkilöstön operoimaan lentokoneeseen. Ilmasta operoitaessa laitteiston liikkuvuus on suuri etu eivätkä maanmuodot vaikuta häirintälähettimen toimintaan. Ilma-alusten koko ja painon kantokyky aiheuttavat kuitenkin rajoituksia häirintälaitteiston painolle ja teholle, jolloin puhtaassa tehossa jäädään verrattuna maasta operoitaviin ratkaisuihin. Tavallisesti näiden laitteiden häirintäteho on noin 1kW luokkaa.

Kevyitä häirintälaitteistoja voidaan levittää tykistön avulla tai sotilaat voivat kantaa niitä mukanaan. Kevyet häirintälaitteet pitää asentaa lähelle häirit্তävää kohdetta riittävän häirintätehon saavuttamiseksi. Kevyiden häirintälähettimien teho vaihtelee 20 – 100 W:n välillä ja ne painavat normaalisti 10 – 15 kg.

### **Prikaatin viestiverkon toiminta häirinnän vaikutuksien alla**

Tehdyissä kokeissa on todettu, että UMTS siviiliverkon toiminta on hyvin vajavaista, jos elektronisen sodankäynnin keinoja käytetään verkkoa vastaan. Tehokkaalla siviilitahojen ja taisteluyksiköiden yhteistoiminnalla voidaan kuitenkin taata paremmat viestiyhteydet eri verkoissa toimiville. Laajalle taajuusalueelle kohdistuva häirintä ei ole kovin tehokasta prikaatin Ad-Hoc verkkoa vastaan, jos verkon radiot käyttävät vaihtotaajuuksimallia. Jatkuvat muutokset taajuudessa estävät tehokkaan häirinnän yksiköiden viestiyhteyksiä vastaan. Yhteen kanavaan keskittyvää häirintää voidaan käyttää, jos vastustaja saa selville käytettävän lähetyssavaimen tai jos häirinnällä kyetään rikkomaan radioiden tahdistus.

Pataljoonataso on prikaatin viestiverkon haavoittuvaisin osa. Yhteydet pataljoonan yksiköistä komentopaikalle ovat pitkät eivätkä yksiköt voi normaalissa tilanteessa ottaa muihin tukiasemiin yhteyttä. Jos vastustaja löytää viestiverkon kriittiset osat, se voi häiritä tiettyjä tärkeimpiä solmuja ja estää suunnatuilla antennilla kaikki pitkät radioyhteydet lamauttaen prikaatin viestitoiminnan. Ad Hoc –verkkoa vastaan toimittaessa häiritsijän olisi tehokasta käyttää useita eri häirintälähettimeä, jotka toimisivat samaan aikaan eri taajuusalueilla. Tällöin on kuitenkin mahdollista, että ongelmia syntyy oman viestiverkon ja häirinnän sekaantumisesta.

Viestiverkon toimintavarmuus paranisi, jos myös komppanioilla olisi omat Ad Hoc –solmut, joiden avulla ne voisivat olla yhteydessä muihin solmuihin VHF-alueella. Tällöin olisi hyvin vaikeaa löytää prikaatin viestiverkosta erityisen haavoittuvia osia. Toinen mahdollisuus olisi taktisten internet palveluiden käyttäminen yksikkötasolla. Myös tässä tapauksessa yhteydet muihin käyttäjiin muodostettaisiin tukiasemien kautta.

# Malli yhteyksien toimintatodennäköisyyksien määrittämiseksi

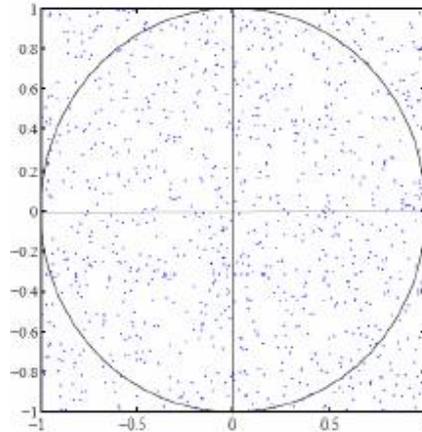
## *Yleistä Monte-Carlo-menetelmistä*

Monte-Carlo on yleisnimitys stokastisille algoritmeille, joita käytetään yleensä laskennallisesti raskaiden ongelmien ratkomiseen. Erityisen paljon Monte-Carlo menetelmiä käytetään laskennallisessa fysiikassa, klassinen esimerkki on moniulotteisten määrättyjen integraalien numeerinen ratkaiseminen.

Tässä projektityössä tarvitsemme vain kaikkein yksinkertaisinta Monte-Carlo menetelmää, ns. Hit-or-miss-menetelmää. Yksinkertaisuudestaan huolimatta Hit-or-miss on hyvä menetelmä määrättyjen integraalien laskemiseen. Sen perustoiminta idea tarvitsemassamme erityistapauksessa selvinnee seuraavan esimerkin avulla, vaikkakin matematiikkaan perehtymättömän voi olla vaikea löytää yhteyttä integraalilaskentaan.

Haluamme laskea yksikköympyrän, eli ympyrän, jonka säteen pituus on yksi, pinta-alan. Tämä voidaan tehdä Hit-or-miss-menetelmällä seuraavasti.

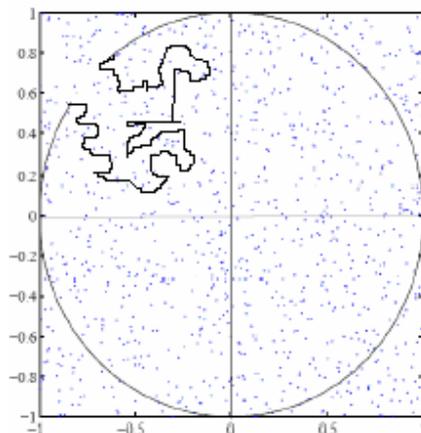
- 1) Piirrä ympyrä, jonka säde on yksi ja sen ympärille yksikköneliö.
- 2) "Heitele" satunnaisesti  $N_0$  kappaletta pisteitä neliön sisälle.
- 3) Laske ympyrän sisälle osuneiden pisteiden lukumäärä  $N$ .
- 4) Ympyrän pinta-alan osuus neliön pinta-alasta, joka tässä tapauksessa on myös ympyrän pinta-ala (sillä neliön pinta-ala on yksi), on yksinkertaisesti  $N_0/N$ .



Matemaattisesti ilmaistuna edellisessä approksimoitiin yksikköympyrän Lebesguen mitta. Yhteys integraalilaskentaan löytyy siitä että kyseinen tehtävä tarkoittaa integraalin

$$\int_{[0,1] \times [0,1]} f(x) dx \quad f(x)=1, \text{ kun } |x|<1 \text{ ja } f(x)=0 \text{ muualla} \quad (1)$$

ratkaisemista, jossa  $dx$  on Lebesguen mitta. Tämän oivaltaminen ei kuitenkaan ole välttämätöntä tulevan ymmärtämiseksi. Monte-Carlo menetelmien hyödyllisyys ja kätevyys eivät välttämättä tulleet esille vielä edellisestä esimerkistä, sillä kyseinen integraali on hyvin helppoa ratkaista analyttisesti. Jos kuitenkin halutaan arvioida alla olevassa kuvassa esitetyn kappaleen pinta-alaa, jossa ympyrän kaarta on hieman



modifioitu, eivät analyttiset menetelmät enää toimi. Hit-or-miss toimii aivan kuten ympyränkin tapauksessa, lasketaan taas kappaleen sisälle osuneiden pisteiden osuus kaikista pisteistä. Tässä tapauksessa standardit numeeriset menetelmätkin toimivat ilman ongelmia, mutta kuvitellaanpa seuraavaksi vastaava tehtävä epäsäännölliselle useampiulotteiselle kappaleelle. Deterministiset integrointimenetelmät kärsivät niin sanotusta moniulotteisuuden kirouksesta ("curse of dimensionality"): laskennan työmäärä kasvaa eksponentiaalisesti ulottuvuuden kasvaessa. Tämän takia näitä menetelmiä voidaan käyttää vain hyvin pieniulotteisiin integraaleihin. Hit-or-miss-menetelmä toimii kuitenkin aina samalla tavalla. Virhe käyttäytyy ulottuvuudesta riippumatta kuten  $1/\sqrt{N}$ , jossa  $N$  on "heitettyjen" pisteiden lukumäärä. Tämän ominaisuuden takia Monte-Carlo-menetelmät ovat usein ainoa mahdollinen keino ratkaista moniulotteinen integraali.

### ***Toimintatodennäköisyydet määrittävät algoritmit***

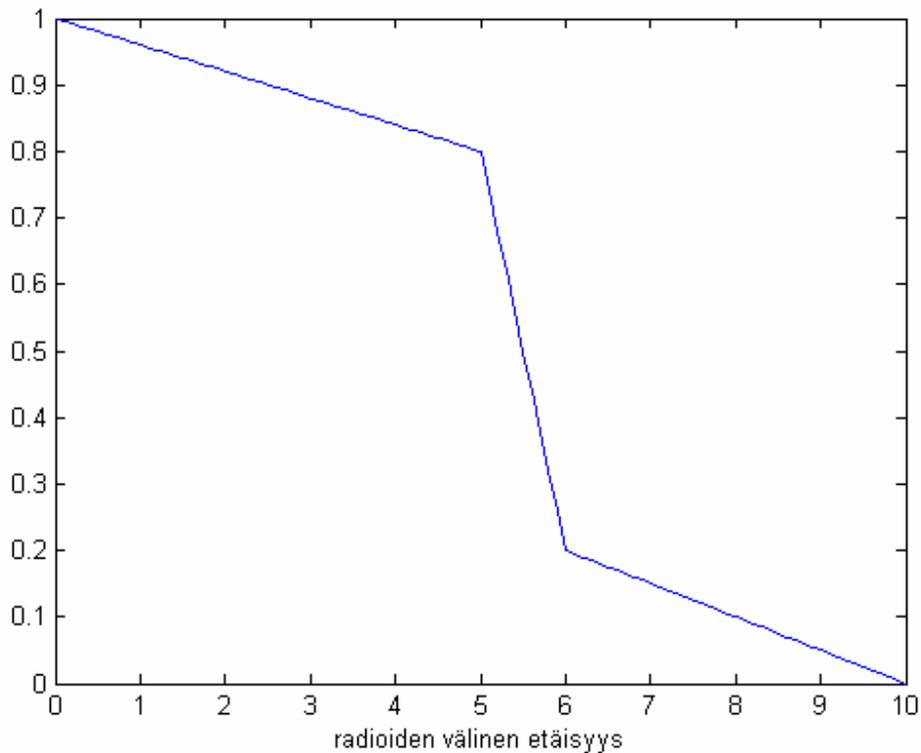
Tässä työssä määrittelemme viestiverkon radioiden välisten yhteyksien toimintatodennäköisyydet MATLAB-koodien avulla. MATLAB valittiin työkalujen toteuttamisympäristöksi, koska se on joustava ja tehokas vaihtoehto paljon matriisilaskentaa sisältävien algoritmien toteuttamiseen.

Olemme kehittäneet algoritmin, jolla voimme laskea arviot radioiden välisten linkkien toimintatodennäköisyyksille perustuen radioiden lähetystehoon ja niiden välisiin etäisyyksiin (lue\_verkko.m). Radioiden välisten yhteyksien toimintatodennäköisyyksien määrittämiseen käytämme Monte-Carlo -menetelmiä. Tarkastelemme kahta tapausta. Ensimmäisessä tapauksessa määritämme todennäköisyydet, joilla ns. runkoverkkoon kuuluvat radiot saavat yhteyden toisiinsa. Toisessa tapauksessa määritämme todennäköisyyden, jolla runkoverkkoon liittyvä radio saa yhteyden toiseen samaan verkkoon liittyvän radion kanssa. Tapauksesta tekee mielenkiintoisen se, että radiot voivat liittyä useampaan kuin yhteen tukiradioon. Olemme kehittäneet molemmille tapauksille Monte-Carlo-menetelmiin perustuvat algoritmit, jotka ottavat parametreikseen verkkoon kuuluvien radioiden ja niiden välisten suorien (ei siis muiden radioiden kautta kulkevien) yhteyksien toimintatodennäköisyydet, sekä iteraatiokierrosten, eli "heitettävien pisteiden" lukumäärän. Seuraavaksi näihin algoritmeihin liittyvien ajotiedostojen toiminnan.

## **lue\_verkko.m**

Tämä ajotiedosto määrittää viestiverkkoon kuuluvien radioiden välisten linkkien toimintatodennäköisyydet radioiden etäisyyksien ja kantamien perusteella. MATLAB-koodi on esitetty liitteessä 1. Syöteenä algoritmille on annettava kaikkien viestiverkkoon kuuluvien radioiden sijainti, kantama ja toimintatodennäköisyys teksti- tai Excel-tiedostona. Algoritmi palauttaa käyttäjälle  $N \times N$ -matriisin, jossa ovat eri radioiden välisten suorien linkkien toimintatodennäköisyydet ja  $N \times 1$ -vektorin, joka sisältää viestiverkon radioiden toimintatodennäköisyydet. Esitetään seuraavaksi algoritmin toiminta vaiheittain.

- 1) Luetaan ensin käyttäjän antamasta tiedostosta radioiden koordinaatit, kantamat ja solmujen toimintatodennäköisyydet. Tiedosto on laadittava niin, että ensimmäisessä sarakkeessa ovat radioiden P-koordinaatit, toisessa sarakkeessa radioiden I-koordinaatit, kolmannessa radioiden kantamat ja neljännessä radioiden toimintatodennäköisyydet. Huomattavaa on, että sekä koordinaatit ja kantamat on annettava samassa yksikössä ja sarakkeet on erotettava tabulaturilla, jos tiedot annetaan txt-tiedostona. Koska vain radioiden etäisyyksillä on merkitystä, voidaan kartan nurkkaan määrittää nollapiste, jonka mukaan radioiden P- ja I-koordinaatit voidaan helposti määrittää. Tiedoston nimi luetaan parametrina. Tiedot voidaan syöttää myös Excel-tiedostona. Ajotiedosto tunnistaa tiedostonimen päätteen perusteella onko kyseessä .txt- vai .xls-tiedosto. Esimerkki tekstitiedostosta on annettu liitteessä 5 (kohdeorganisaatiolta saatu esimerkkiverkko).
- 2) Tallennetaan seuraavaksi radioiden toimintatodennäköisyydet solmut-vektoriin ja luodaan  $N \times N$ -matriisi radioiden etäisyyksien laskemista varten.
- 3) Lasketaan radioiden etäisyydet koordinaattien perusteella Pythagoraan lauseen avulla.
- 4) Lasketaan radioiden välisten linkkien toimintatodennäköisyydet. Tässä hyödynnetään apufunktiota `linkki_tn`, joka laskee kahden radion välisen linkin toimintatodennäköisyyden radioiden etäisyyden ja lähettävän radion kantaman perusteella. Linkin toimintatodennäköisyyttä on approksimoitu paloittain määrittelyllä lineaarisella funktiolla, jonka on esitetty graafisesti kuvassa 2.



Kuva 2. Viestiyhteyden toimintatodennäköisyyden approksimointi radioiden välisen etäisyyden (yksikkö 1000m) avulla.

Kuvassa 2 kantamaksi on annettu 5000 metriä. Kuten kuvasta näkyy, muodostuu funktio kolmesta eri osasta. Kantamaa pienemmillä etäisyyksillä todennäköisyys vaimenee hitaasti ja kantamaa vastaavalla etäisyydellä todennäköisyys on vielä 0.8. Tämän jälkeen todennäköisyys laskee jyrkästi ja etäisyydellä 1.2 x kantama toimintatodennäköisyys on enää 0.2. Tästä eteenpäin toimintatodennäköisyys alenee hitaasti kohti nollaa ja etäisyydellä 2 x kantama se häviää kokonaan.

Tämä funktio on hyvin yksinkertainen, mutta se palvelee tämän työn tarkoitusperiä riittävän hyvin. Tässä työssä sivuutamme linkin toimintatodennäköisyyteen liittyvän radioteknisen ongelmakentän ja jätämme näin huomioimatta mm. maaston vaikutuksen toimintatodennäköisyyteen. Funktio on kuitenkin laadittu niin, että sitä on helppo muuttaa ja parantaa tarkempien linkkien toimintatodennäköisyyksien saamiseksi.

### laske\_TNmatriisi.m

Tämä ajotiedosto (ks. liite 2) määrittää runkoverkkoon kuuluvien radioiden välisten yhteyksien toimintatodennäköisyydet. Parametreiksi annetaan yksittäisten radioiden toimintatodennäköisyydet  $N$ :n pituisessa vektorissa (solmut) ja radioiden välisten suorien yhteyksien toimintatodennäköisyydet  $N \times N$  kokoisessa matriisissa (linkit). Solmutvektorin perustaalla radioille voidaan ajatella indeksit  $1, \dots, N$ . Tämän lisäksi ajotiedostoa editoimalla voi halutessa muuttaa simulaatiokierrosten määrää (rounds). Itse algoritmi toimii siten, että arvotaan realisaatioita stokastisesta verkosta ja tallennetaan muistiin

kunkin realisaation tapauksessa toimivat yhteydet (Hit-or-miss). Lopuksi lasketaan kuinka suuressa osassa tapauksia yhteys toimi kunkin parin ja kummankin suunnan kohdalla. Tämä on toteutettu seuraavasti:

- 1) Luodaan  $N \times N$  matriisi (onnistuneet), jonka kaikki alkiot ovat nolliä toimivien yhteyksien lukumäärien tallentamiseen.
- 2) Aloitetaan silmukka, jonka kukin kierros vastaa yhtä iteraatiokierrosta.
- 3) Arvotaan, mitkä solmut toimivat, ja tallennetaan niiden indekset suuruusjärjestyksessä  $M$ -pituisen vektoriin (toimivat).
- 4) Arvotaan, mitkä toimivien solmujen välisistä yhteyksistä toimivat ja ilmoitetaan tulos  $M \times M$  matriisissa (runko), jonka indekset menevät samalla tavalla, kuin edellisessä kohdassa luodun vektorin. Matriisin alkion arvo 1 vastaa toimivaa suoraa yhteyttä ja 0 toimimatonta suoraa yhteyttä.
- 5) Korotetaan edellisessä kohdassa luotu matriisi  $M$ :nteen potenssiin, jolloin saadaan uusi  $M \times M$  matriisi (yhteydet). Matriisin alkion arvo 0 vastaa toimimatonta yhteyttä (suoraa ja epäsuoraa) ja nollasta eroava arvo toimivaa yhteyttä.
- 6) Tutkitaan mitkä yhteyksistä toimivat edellisessä kohdassa luodun matriisin avulla ja kasvatetaan toimivien yhteyksien tapauksessa 1. kohdassa luodun matriisin vastaavaa alkioita yhdellä.
- 7) Palataan kohtaan 2, jos iteraatiokierrosten lukumäärä on pienempi kuin haluttu, muuten edetään seuraavaan kohtaan.
- 8) Jaetaan 1. kohdassa luotu matriisi kierrosten lukumäärällä, jolloin saadaan  $N \times N$  matriisi, jonka alkiot kertovat yhteyksien toimintatodennäköisyyksien arviot.

Edellisen algoritmin 5. kohta saattaa vaikuttaa hieman epäselvältä. Kohdassa esitetty johtopäätös perustuu tunnettuun graafiteorian lauseeseen: Binäärisen yhteysmatriisin  $n:n$  potenssin alkio  $(i,j)$ , vastaa  $i$ :stä  $j$ :hin menevien  $n:n$  pituisten polkujen lukumäärää. Nyt emme ole kiinnostuneita polkujen lukumäärästä, vaan siitä onko radioiden välillä yhteyttä, eli yhtään polkua.

### **laske\_TN.m**

Tämä ajotiedosto (ks. liite 3) määrittää kahden runkoverkkoon liittyvän radion välisen yhteyden toimintatodennäköisyyden. Toinen radiosta on lähettäjä ja toinen vastaanottaja. Toimintatodennäköisyys on suunnattu todennäköisyys, että lähettäjä saa yhteyden vastaanottajaan (verkko on epäsymmetrinen). Verkkoon liittyvät radiot voivat liittyä mielivaltaiseen määrään radioita ja näiden ”liittymis-yhteyksien” oletetaan toimivan. Itse asiassa ajotiedosto siis määrittää todennäköisyyden, että jostakin runkoverkosta valitusta  $m$ :stä radiosta saadaan yhteys johonkin etukäteen valituista  $n$ :stä radiosta. Parametreinä algoritmi tarvitsee edellisessä osiossa esitellyn algoritmin parametrien lisäksi  $m:n$  pituisen vektorin, jossa kerrotaan mihin runkoverkon radioihin lähettäjä liittyy (liitos1) ja  $n:n$  pituisen vektorin (liitos2), jossa ilmoitetaan ne radiot, joihin vastaanottaja liittyy. Tämän jälkeen algoritmi toimii seuraavasti:

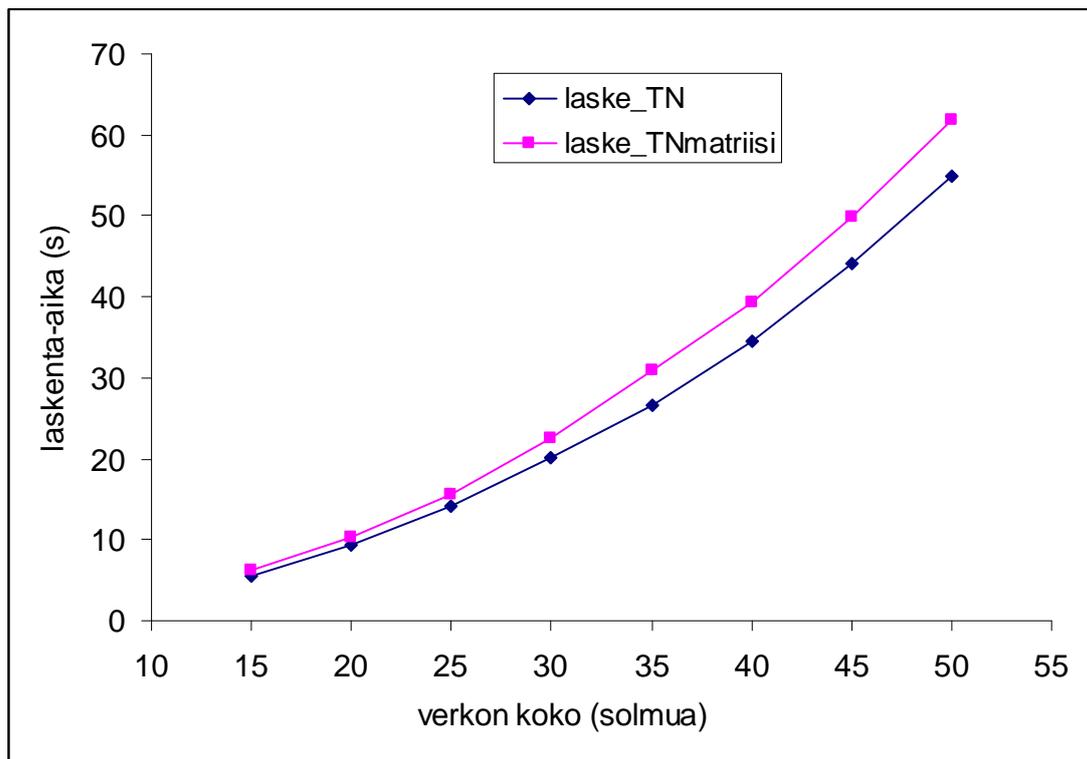
- 1) Luodaan  $N:n$  pituinen vektori (liitos1vek), jonka alkio on 1, jos lähettäjä liittyi kyseistä alkiota vastaavaan radioon, ja 0 muuten. Luodaan vastaava vektori (liitos2vek) myös vastaanottajan liitännöille.
- 2) Luodaan muuttuja (onnistuneet) onnistuneiden yhteysyritysten määrän tallentamista varten.
- 3) Aloitetaan silmukka, jonka kukin kierros vastaa yhtä iteraatiokierrosta.
- 4) Arvotaan, mitkä solmut toimivat, ja tallennetaan niiden indeksit suuruusjärjestyksessä  $M$ -pituisen vektoriin (toimivat). Luodaan myös vektorit (liitetyt1, liitetyt2), jotka kertovat liitäntöjen indeksit tässä vektorissa.
- 5) Arvotaan, mitkä toimivien solmujen välisistä yhteyksistä toimivat ja ilmoitetaan tulos  $M \times M$  matriisissa (runko), jonka indeksit menevät samalla tavalla, kuin edellisessä kohdassa luodun vektorin. Matriisin alkion arvo 1 vastaa toimivaa suoraa yhteyttä ja 0 toimimatonta suoraa yhteyttä.
- 6) Korotetaan edellisessä kohdassa luotu matriisi  $M:n$ teen potenssiin, jolloin saadaan uusi  $M \times M$  matriisi (yhteydet). Matriisin alkion arvo 0 vastaa toimimatonta yhteyttä (suoraa ja epäsuoraa) ja nollasta eroava arvo toimivaa yhteyttä.
- 7) Tutkitaan onko lähettävästä radiosta yhteyttä vastaanottavaan radioon ja kasvatetaan 2. kohdassa luodun muuttujan arvoa yhdellä jos yhteys on olemassa.
- 8) Palataan kohtaan 3, jos iteraatiokierrosten lukumäärä on pienempi kuin haluttu, muuten edetään seuraavaan kohtaan.
- 9) Jaetaan 2. kohdassa luodun muuttujan arvo kierrosten lukumäärällä- Saatu tulos on yhteyden toimintatodennäköisyyden arvio.

### ***Algoritmien vaatima laskenta-aika***

Kohdeorganisaatio asetti tiettyjä vaatimuksia algoritmien laskenta-aikojen suhteen. Kahden radion välisen yhteystodennäköisyyden määrittäminen tuli olla mahdollista reaaliaikaisesti (laskenta-aika  $< 1$  min). Toisaalta koko verkon yhteystodennäköisyyksien määrittäminen sai viedä enemmän aikaa. Nämä laskenta-aikavaateet koskevat normaalia pöytä tietokonetta.

Tarkastellaan siis algoritmeja laske\_TN.m ja laske\_TNmatriisi.m. Näistä edelliselle oli asetettu erityinen vaatimus nopeudesta. Kohdeorganisaatio soveltaa luotuja malleja viestiverkkoihin, joissa on 20 – 50 solmua. Laskenta-aika riippuu käytettävästä simulaatiokierrosten määrästä sekä analysoidavan verkon koosta. Riippuvuus simulaatiokierrosten määrästä on luonnollisesti lineaarinen, sillä simulaatiokierrokset ovat odotusarvoisesti samanlaisia (verkon stokastisuudesta johtuen ne eivät ole aina samanlaisia).

Simulaatiokierroksien vaikutusta kiinnostavampi tutkimuksen kohde on verkon koko. Analysoimme algoritmien laske\_TN.m ja laske\_TNmatriisi.m vaatimaa laskenta-aikaa generoimissamme verkoissa, joissa solmuja oli 15 – 50. Ajoimme kotitietokoneella kummankin algoritmin 10000 kierroksella kullakin verkon koolla (8 eri kokoa) kolme kertaa ja mittasimme laskentaan kuluneen ajan. Keskiarvot laskenta-ajoista eri verkon kooilla on esitetty kuvassa 3.



Kuva 3. Verkon koon vaikutus laskenta-aikaan.

Kuvasta nähdään, että koko verkon yhteystodennäköisyyksien laskemiseen (laske\_TNmatriisi) ei mene juurikaan kauemmin kuin yksittäisten verkkoon liittyneiden radioiden yhteystodennäköisyyden laskemiseen (laske\_TN). Tämä voi vaikuttaa yllättävältä, mutta selittyy sillä, että algoritmit eroavat toisistaan lähinnä yhteyksien tarkistamisen suhteen (tarkistetaan mistä on yhteys mihinkin). Laske\_TNmatriisi.m:ssä tarkistetaan kaikkien solmujen väliset yhteydet, kun taas laske\_TN.m:ssä vain lähettäjään ja vastaanottajaan yhteydessä olevien solmujen yhteydet. Tästä seuraa, että kaikkien yhteyksien todennäköisyyksien laskeminen vie vain vähän enemmän aikaa.

Kuvasta nähdään myös, että simulaatiokierroksia ollessa 10000, toteutuu kohdeorganisaation vaatimus laske\_TN.m:n laskenta-ajalle kaikissa verkon kokoluokissa. Laske\_TNmatriisi.m voi halutessa ajaa suuremmallakin simulaatiokierrosten määrällä, koska tässä tapauksessa käytetylle laskenta-ajalle ei ollut rajoituksia.

### ***Varianssianalyysi: tulosten epävarmuus***

Tutkimme verkon yhteystodennäköisyydet määrittävien algoritmien antamien tulosten epävarmuutta (laske\_TNmatriisi.m ja laske\_TN.m). Tutkimuksen kohteena on tulosten epävarmuuden riippuvuus simulaatiokierrosten lukumäärästä sekä vastaavan yhteystodennäköisyyden suuruudesta.

Tutkimme tässä tapauksessa pelkästään algoritmia, joka antaa verkon kaikkien solmuparien välisten yhteyksien todennäköisyydet (laske\_TNmatriisi.m). Tässä tapauksessa saadut tulokset on yleistettävissä myös algoritmiin, jossa verkkoon

liitytään(laske\_TN.m), sillä tämä on erikoistapaus, jossa käsitellään vain tiettyä solmuparia laajemmassa verkossa. Analyysin kohteeksi olemme valinneet kohdeorganisaatiolta saamamme esimerkkiverkon (ks. liite 5), sillä se vastaa parhaiten todellista sovellustilannetta.

## Tulosten epävarmuuden riippuvuus simulaatiokierrosten lukumäärästä

Tulosten epävarmuuden riippuvuus simulaatiokierrosten määrästä on erittäin kiinnostavaa algoritmin käytännön soveltamisen kannalta. Kun tulosten hajonta ja varianssi on määritetty tietylle simulaatiokierrosten lukumäärälle, tiedämme kuinka tarkkoja tuloksia algoritmi pystyy ko. kierrosmäärällä antamaan.

Ajoimme laske\_TNmatriisi.m-ajotiedostolla kohdeorganisaatiolta saamamme esimerkkiverkon läpi 1000 kertaa annetulle simulaatiokierrosten (rounds) määrälle. Simulaatiokierrosten määrinä käytimme 100, 1000, 3500, 7000 ja 10000:ta. Koska verkossa oli 22 solmua, saimme kustakin ajosta 1000 22x22-todennäköisyysmatriisia. Näistä määritimme aritmeettisen keskiarvon ja otoskeskihajonnan kullekin matriisin alkiolle. Keskiarvon ja otoskeskihajonnan laskemiseen käytettiin kaavoja (2) ja (3):

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_n \quad (2)$$

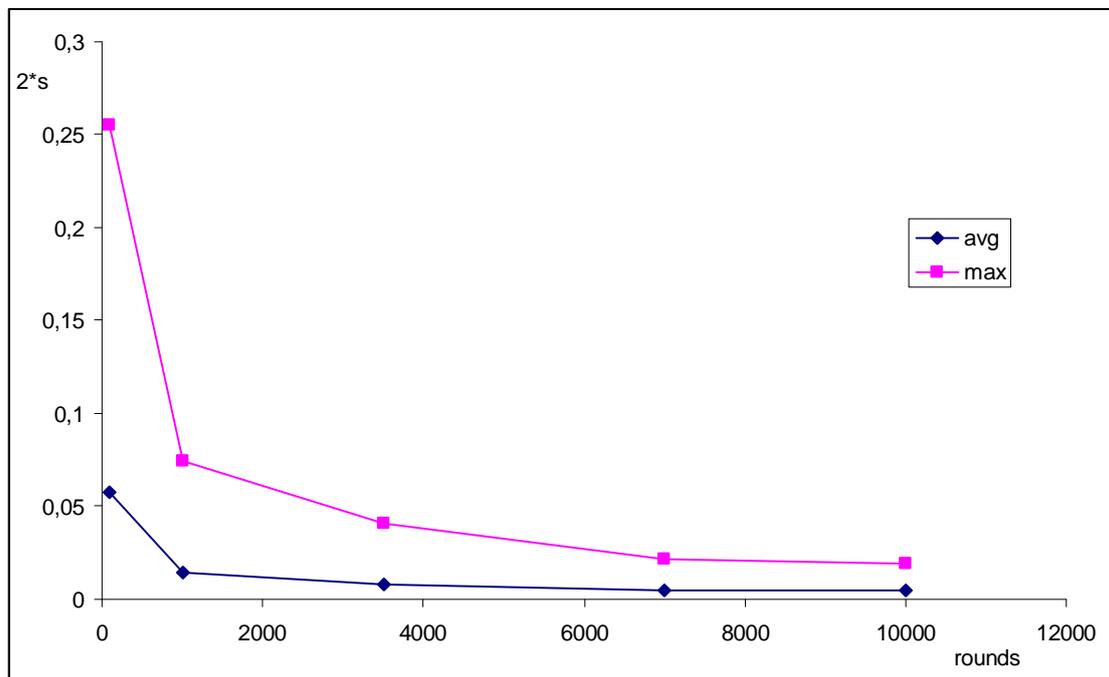
$$s = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2} \quad (3)$$

22x22-matriisia on epäkäytännöllistä käsitellä kokonaisuena, joten laskimme matriisin alkioiden keskihajontojen maksimin ja keskiarvon. Keskihajontojen minimi on kaikissa tapauksissa nolla, koska yhteystodennäköisyysmatriisin diagonaali-alkiot ovat aina ykkösiä. Näin ollen minimin analysointi ei tässä tapauksessa ole mielekäästä. Näistä laskimme 95 % luottamusväliä ( $2\sigma$  vastaa keskimäärin 95 % luottamusväliä). Pöytä tietokoneelta tehtyihin simulaatioajoihin laskenta-aikaa kului kaiken kaikkiaan noin kahdeksan tuntia. Saadut tulokset on esitetty taulukossa 1.

Taulukko 1. Tulosten keskihajonnat eri simulaatiokierrosten lukumäärillä.  $2s$  vastaa 95 % luottamusväliä. Taulukossa on annettu keskihajontojen keskiarvo ja maksimi yli todennäköisyysmatriisin alkioiden.

Simulaatiokierrosten lkm	$2s$ (keskiarvo)	$2s$ (maksimi)
100	0.057522	0.2554
1000	0.014324	0.074534
3500	0.007794	0.040898
7000	0.005197	0.021812
10000	0.004967	0.01916

Saadut tulokset on esitetty graafisesti kuvassa 4.



Kuva 4. 95 % luottamusvälin koko  $2s$  eri simulaatiokierrosten lukumäärillä. Max vastaa keskihajontojen maksimia ja avg keskiarvoa.

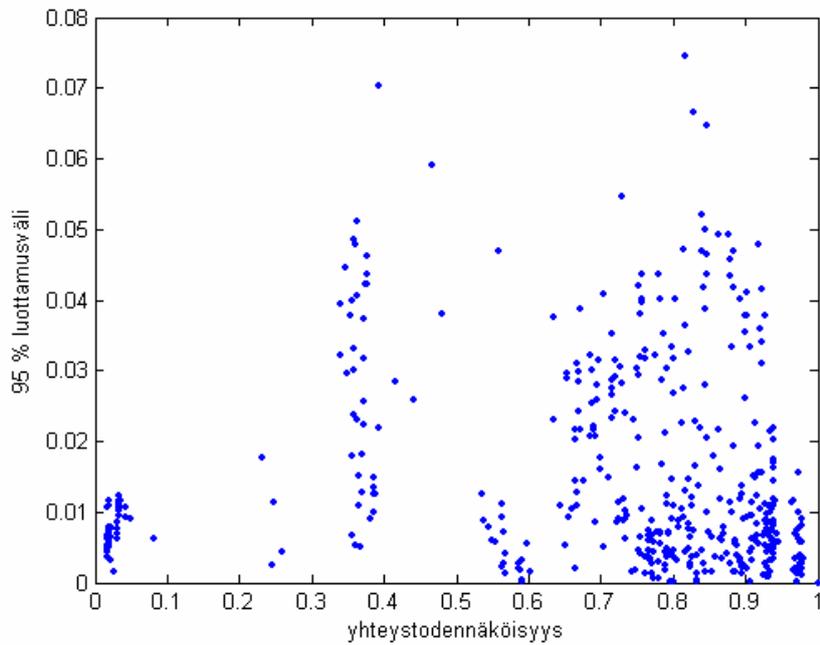
Kuvasta 4 nähdään, että 95 % luottamusvälin koko pienenee eksponentiaalisesti simulaatiokierrosten lukumäärän funktiona. Simulaatiokierrosten lisäämisestä 7000 10000:een ei ole läheskään niin paljon hyötyä kuin lisäyksestä 1000->3000. Tarkkailemalla taulukon 1 arvoja havaitaan, että keskimäärin tuloksissa saavutetaan kohdeorganisaation kannalta haluttu taso (0.02-tarkkuus) jo 1000 simulaatiokierroksen tapauksessa. Kuitenkin kun tarkkaillaan huonointa tapausta (maksimikeskihajonta), havaintaan, että tämä taso saavutetaan vasta 10000 simulaatiokierroksen tapauksessa. Maksimikeskihajonta edustaa yleistä tapausta. Näyttäisi siis siltä, että 10000 simulaatiokierrosta on sopiva simulaatiokierrosten lukumäärä käytännön sovelluksia ajatellen.

### **Epävarmuuden korrelaatio vastaavan yhteystodennäköisyyden kanssa**

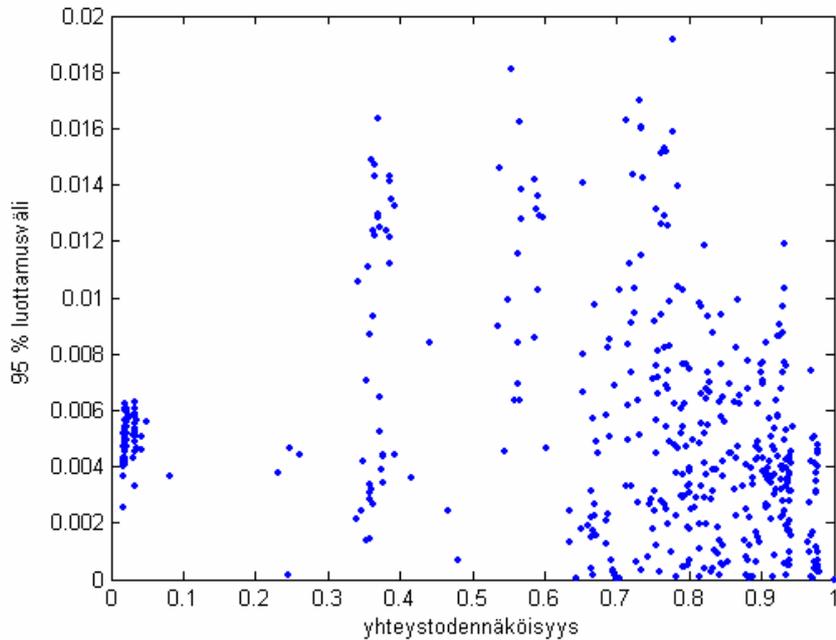
Tarkastellaan seuraavaksi onko yhteyden toimintatodennäköisyyden ja algoritmin antaman virheen koon välillä yhteyttä. Toisin sanoen, onko niin, että mitä suurempi yhteystodennäköisyys sitä suurempi todennäköisyyden epävarmuus? Jos asia olisi näin, suhteelliset virheet todennäköisyyksissä saattaisivat olla samaa luokkaa.

Tarkastellaan kutakin tapausta (simulaatiokierrosten lkm = 100, 1000, 3500, 7000, 10000) erikseen. Aiemmin saatuina tuloksina on yhteystodennäköisyysmatriisin keskiarvo ja vastaava otoskeskihajontamatriisi. Tulokset voidaan ryhmitellä pareiksi (yhteystodennäköisyys, 95 % luottamusvälin koko) ottamalla vastaavat alkiot edellä

mainituista matriiseista. Näin saadaan  $2^2 = 484$  mittausparia yhteystodennäköisyyden ja luottamusvälin koon yhteydestä. Asian tutkimiseksi piirrettiin viisi kappaletta (kullekin simulaatiokierrosten lukumäärälle) kuvaajia: 95 % luottamusväli yhteystodennäköisyyden funktiona. Alla on esitetty nämä kuvaajat tapauksissa: simulaatiokierrosten lkm = 1000 ja 10000.



Kuva 5. 95 % luottamusvälin koko ( $2 * \text{otoskeskihajonta}$ ) yhteystodennäköisyyden funktiona. Simulaatiokierrosten määrä oli 1000.



Kuva 6. 95 % luottamusvälin koko (2 \* otoskeskihajonta) yhteystodennäköisyyden funktiona. Simulaatiokierrosten määrä oli 10000.

Kuvaajien piirtämisen lisäksi havaintoaineistolle laskettiin korrelaatiot Pearsonin otoskorrelaatiota eli kaavaa (4) käyttäen:

$$r_{xy} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{(n-1)s_x s_y}, \text{ missä } s_x \text{ ja } s_y \text{ ovat otoskeskihajontoja.} \quad (4)$$

Lasketut otoskeskihajonnat eri tapauksissa on esitetty taulukossa 2.

Taulukko 2. Korrelaatio yhteystodennäköisyyden ja 95 % luottamusvälin koon (2 \* otoskeskihajonta) välillä eri simulaatiokierrosten lukumäärillä.

Simulaatiokierrosten lukumäärä	Korrelaatio todennäköisyyden ja luottamusvälin koon välillä
100	0.0359
1000	-0.0438
3500	-0.0701
7000	-0.0518
10000	-0.2296

Kuvia 5 ja 6 vastaavat kuvat muille simulaatiokierrosten määrille kuin 1000 ja 10000 olivat hyvin samankaltaisia kuin edellä esitetyt kuvat. Kuvista 5 ja 6 havaitaan selvästi, että yhteystodennäköisyyden ja luottamusvälin koon välillä ei ole suoraa yhteyttä. Tämä nähdään myös tarkastelemalla taulukon 2 korrelaatiota. Kaikki korrelaatio ovat lähellä nollaa. Ainoa korrelaation arvo, joka poikkeaa nolasta merkittävästi, on viimeinen tapaus, jolloin simulaatiokierrosten lukumäärä oli 10000 – tällöin korrelaatio oli -0.2269, joka ei ole suuri arvo. Kuvien ja taulukon perusteella voidaan siis sanoa, yhteystodennäköisyyden epävarmuuden (luottamusväli) ja yhteystodennäköisyyden suuruuden välillä ei ole selvää yhteyttä.

Kuvien 5 ja 6 sekä muista tapauksista piirrettyjen kuvaajien perusteella näyttäisi siltä, että suurimmat epävarmuudet sijoittuvat todennäköisyysvälin keskelle. Välin päissä ( $p \in [0,0.1]$  ja  $p \in [0.9,1.0]$ ) epävarmuudet näyttäisivät olevan pienempiä. Kohdeorganisaation kannalta kiinnostava on erityisesti loppupää  $p \in [0.9,1.0]$ , sillä tällöin todennäköisyyden toisellakin desimaalilla on merkitystä. Tulokset näyttävät tässä mielessä suotuisilta, koska epävarmuus pienenee todennäköisyysvälin loppupäässä. Kuitenkin täytyy muistaa, että analyysi koskee vain yhtä verkkoa, eikä tällöin kata yleistä tapauksia.

## **Malli tukiasemien tärkeyden määrittämiseksi keskeisyysmitoilla**

Keskeisyysmitoilla pyritään vastaamaan kysymykseen, ”Mikä solmu tai mitkä solmut ovat tärkeimmät/keskeisimmät solmut verkossa?”, tai nimenomaan tässä tapauksessa ”Minkä tukiasemien toiminta tulisi turvata ensisijaisesti?” Keskeisyysmittoja käytetään yleisesti verkkoteoriassa kuvaamaan verkon solmujen tärkeyttä. Keskeisyysmittoja on useita erilaisia ja ne mittaavat solmujen tärkeyttä hieman eri tavalla [1]. Tässä työssä käytämme solmujen keskeisyyden määrittämisessä seuraavia mittoja: *strength*, *closeness* ja *betweenness*. Näistä mitoista *betweenness* on monimutkaisin ja sopii parhaiten armeijan viestiverkon tukiasemien tärkeyden mittaamiseen.

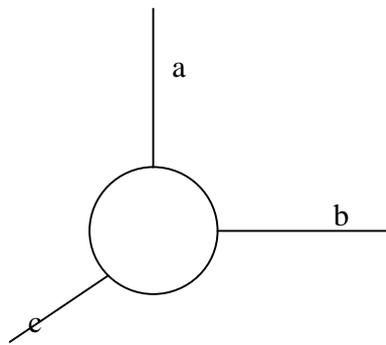
### ***Painojen laskeminen***

Edellä esitetyt keskeisyysmitat voidaan laskea sekä painottamattomille että painotetuille verkoille. Koska viestiverkon linkeille on määritelty toimintatodennäköisyydet ja ne vaihtelevat huomattavasti eri linkkien välillä, on selvää, että yhteydet solmujen välillä eivät ole samanarvoisia, ja näin ollen keskeisyysmittojen laskemisessa kannattaa ottaa huomioon linkkien toimintatodennäköisyyksien perusteella määritetyt painot.

*Closeness* ja *betweenness* -algoritmeissa etäisyysmitan laskemisessa käytetään linkin päätesolmujen välistä etäisyyttä, eli pituutta, joka yleensä määritellään painon käänteislukuna. Kuten aikaisemmin on todettu, tässä tapauksessa linkin päätesolmujen välinen etäisyys ( $l$ ) on järkevää määritellä linkin toimintatodennäköisyyden ( $p$ ) negatiivisena luonnollisena logaritmina, eli  $l = -\ln(p)$ . Logaritmillemhan tunnetusti pätee  $\log(ab) = \log(a) + \log(b)$  ja näin ollen useamman solmun kautta kulkevan reitin pituus on verrannollinen reitin toimintatodennäköisyyteen, joka lasketaan kertomalla yksittäisten linkkien toimintatodennäköisyydet keskenään. Tässä lähestymistavassa myös

erikoistapaukset, jossa linkin toimintatodennäköisyys on nolla tai yksi, tulee otettua järkevästi huomioon. Kun linkin toimintatodennäköisyys on yksi, eli linkki toimii varmasti, linkin päätepisteissä sijaitsevien solmujen etäisyydeksi toisistaan saadaan nolla, eli viesti saadaan aina menemään solmusta toiseen. Kun linkin toimintatodennäköisyys on nolla, eli solmusta toiseen ei saada suoraa yhteyttä, solmujen väliseksi etäisyydeksi saadaan ääretön.

*Strength* –keskeisyysmitan laskemisessa tarvitaan taas linkin etäisyyden sijasta linkin painoa. Paino ( $w$ ) saadaan laskettua helposti etäisyyden ( $l$ ) käänteislukuna, eli  $w = 1/l = 1/(-\ln(p))$ .



Kuva 7. Solmujen painojen laskeminen.

Kuva 7 esittää yksinkertaisen esimerkin, jossa solmuun tulee kolme linkkiä. Linkkien toimintatodennäköisyydet ovat  $a$ ,  $b$  ja  $c$ . Linkkien pituudet lasketaan seuraavasti:

$$l_a = -\ln a \quad l_b = -\ln b \quad l_c = -\ln c$$

Ja painot vastaavasti:

$$w_a = \frac{1}{l_a} = -\frac{1}{\ln a} \quad w_b = \frac{1}{l_b} = -\frac{1}{\ln b} \quad w_c = \frac{1}{l_c} = -\frac{1}{\ln c}$$

Jos  $a = 0.9$ ,  $b = 0.5$  ja  $c = 0.1$ , niin

$$l_a = 0.11 \quad l_b = 0.69 \quad l_c = 2.30 \quad \text{ja} \quad w_a = 9.49 \quad w_b = 1.44 \quad w_c = 0.43$$

Eli linkin  $a$  pituus on selvästi pienempi kuin linkin  $b$  pituus, jonka pituus on puolestaan selvästi pienempi kuin linkin  $c$  pituus. Vastaavasta linkin  $a$  paino on selvästi suurempi kuin  $b$ :llä, jonka paino puolestaan on selkeästi  $c$ :n painoa suurempi. Eli linkin toimintatodennäköisyyden suuruus vaikuttaa merkittävästi linkin pituuteen ja painoon, kuten pitääkin.

## **Strength –keskeisyys**

Painottamattomilla verkoilla yksinkertaisin keskeisyysmitta on *degree*. *Degree* –mitta kertoo sen montako linkkiä on kiinnitetty kyseiseen solmuun. Painotetuilla verkoilla *degree* –mittaa vastaava mitta on *strength*. *Strength* –mitta toimii samalla tavalla kuin *degree*, mutta nyt linkkien painot otetaan huomioon. *Strength* –mitta on siis kaikkien kyseiseen solmuun kiinnittyneiden linkkien painojen summa. Tässä työssä tarkasteltu viestiverkko ei ole symmetrinen, vaan solmuun tulevan linkin toimintatodennäköisyys voi olla eri kuin solmusta lähtevän linkin toimintatodennäköisyys. Näin ollen tarkastellaan erikseen kahta mitta: *in-strength* ja *out-strength*. *In-strength* kuvaa solmuun tulevien lenkkien vahvuutta ja *out-strength* solmusta lähtevien linkkien vahvuutta.

Tarkastellaan jälleen kuvan 7 tapausta, jossa solmuun on kiinnittynyt kolme linkkiä. Linkkien toimintatodennäköisyydet ovat a, b ja c. Solmun *strength* –mitta lasketaan:

$$s = w_a + w_b + w_c$$

Tarkastellaan tapausta, jossa solmuun tulevien linkkien toimintatodennäköisyydet ovat a = 0.9, b = 0.5 ja c = 0.1 ja solmusta lähtevien linkkien a = b = c = 0.5. Näillä arvoilla saadaan:

$$s_{in} = w_{a,in} + w_{b,in} + w_{c,in} = 9.49 + 1.44 + 0.43 = 11.36$$

$$s_{out} = w_{a,out} + w_{b,out} + w_{c,out} = 1.44 + 1.44 + 1.44 = 4.32$$

Huomataan, että *in-strength* ja *out-strength* saavat merkittävästi erilaiset arvot, vaikka linkkien toimintatodennäköisyyksien summa on sama. Tämä johtuu siitä, että lähellä yhtä olevat todennäköisyydet saavat hyvin suuren painon, koska yhteys toimii suurella varmuudella. Pituuksia laskettaessa erot eivät ole aivan näin suuria.

*Strength* –mitat antavat jonkinlaisen käsityksen siitä, mitkä solmut ovat keskeisiä viestiverkossa, mutta keskeisyysanalyysissä näitä arvoja ei pelkästään kannata kovin paljoa tuijottaa. Ne toimivat lähinnä muiden mittojen tukena ja vertailukohteina.

## **Lyhyimmän polun laskeminen**

Sekä *closeness* että *betweenness* -keskeisyysmitat perustuvat *geodesic pathin*, eli lyhyimmän polun laskemiseen. Painottamattomilla verkoilla polun pituus tarkoittaa sitä kuinka monta linkkiä on kyseisellä polulla kuljettaessa solmusta A solmuun B. Lyhin polku on puolestaan se polku, jota pitkin päästään vähimmällä linkkimäärällä solmusta A solmuun B. Painotetuilla verkoilla ei tarkastella solmujen välillä olevien linkkien määrää, vaan linkkien pituuksia. Lyhin polku on siis se, jota pitkin kuljettaessa linkkien pituuksien summa on lyhyempi kuin millään muulla polulla kyseisten solmujen välillä. Voi olla, että lyhyimpiä polkuja on myös kaksi tai useampia.

Lyhyimpien polkujen laskemisessa käytimme Dijkstran algoritmia [2], joka laskee kaikki etäisyydet annetusta aloitussolmusta i seuraavasti:

- 1) Etäisyydet solmusta i muihin solmuihin tallennetaan taulukkoon ja jokaisen solmun kohdalle merkitään joko ”exact”, joka tarkoittaa, että etäisyys solmujen välillä on laskettu tarkasti, tai ”estimated”, joka tarkoittaa, että etäisyys solmujen välillä on

- arvioitu. Aloitetaan siten, että merkitään solmun  $i$  estimoiduksi etäisyydeksi 0 ja muiden solmujen estimoiduiksi etäisyyksiksi  $\infty$ .
- 2) Valitaan niistä solmuista, joiden merkki on ”estimated”, se solmu, jonka estimoitu etäisyys on pienin ja merkitään kyseisen solmun kohdalle ”exact”.
  - 3) Tutkitaan tarkasteltavan solmun välittömiä naapureita, eli niitä solmuja mistä on suora yhteys tarkasteltavaan solmuun. Lisätään tarkasteltavan solmun etäisyyteen sen ja naapurisolmun välinen etäisyys ja muutetaan saatu summa naapurisolmun estimoiduksi etäisyydeksi, jos se on pienempi kuin kyseisten solmujen nykyinen estimoitu etäisyys.
  - 4) Toistetaan askelta 2 niin kauan, kunnes ”estimated” –merkittyjä solmuja ei enää ole.

### **Closeness –keskeisyys**

Solmun  $i$  *closeness* –keskeisyys on keskimääräinen geodesinen etäisyys solmusta  $i$  muihin solmuihin, eli lyhyimpien polkujen keskiarvo. *Closeness* –mitta on sitä pienempi mitä keskeisempi solmu, eli solmusta pääsee keskimäärin sitä nopeammin muihin solmuihin mitä pienempi sen *closeness*-arvo on. Kun lyhyimmät polut kaikkien solmujen välillä on laskettu edellä esitetyillä algoritmeilla, *closeness* –mitta jokaiselle solmulle on helppo laskea ottamalla keskiarvo solmun  $i$  laskettujen lyhyimpien polkujen arvoista.

### **Betweenness –keskeisyys**

*Betweenness* –mitta kuvaa sitä kuinka monella lyhyimmällä polulla solmu  $i$  sijaitsee, kun tarkastellaan lyhyimpiä etäisyyksiä jokaisen kahden verkon solmun välillä. Tämä mitta kuvaa sitä mitkä ovat verkon vaikutusvaltaisimmat solmut, eli minkä solmujen kautta viesti useimmiten liikkuu. Mitä suurempi *betweenness* –arvo kyseisellä solmulla on sitä keskeisempi solmu on. Jos korkeimman *betweenness*- arvon omaava solmu poistetaan verkosta, verkon solmujen väliset etäisyydet kasvavat yleensä kaikista eniten (eli tässä tapauksessa viestiverkon toimintatodennäköisyys pienenee eniten). *Betweenness* –mittaa voidaanakin pitää varsin hyvänä keskeisyysmittana, kun tarkastellaan viestiverkon toimintatodennäköisyyksiä.

*Betweenness* –mitan laskemisessa käytimme Newmanin algoritmia [2], jolla laskenta-aika saadaan supistettua arvoon  $O(mn)$ , koska laskutoimituksia tarvitaan vain  $m$  kertaa  $n$ . Yksinkertaisella algoritmeilla, jossa kaikki polut käydään erikseen läpi, tarvitaan  $mn^2$  laskutoimitusta ja laskenta-aika on  $O(mn^2)$ . Näin ollen käyttämällä Newmanin algoritmia laskuaikaa saadaan supistettua huomattavasti. Newmanin algoritmin toimintaperiaate on seuraava:

- 1) Lyhyimmät polut solmusta  $j$  kaikkiin muihin solmuihin lasketaan käyttämällä Dijkstran algoritmia painotetuille verkoille.
- 2) Jokaiselle solmulle  $k$  asetetaan muuttaja  $b_k$ , jolle annetaan arvoksi 1.
- 3) Kaikki solmut  $k$  käydään läpi aloittaen siitä solmusta, jonka etäisyys solmuun  $j$  on suurin. Tämän solmun  $b_k$  –arvo lisätään jokaiseen kyseisen solmun edeltäjäsolmuun, eli niihin solmuihin joihin kyseisestä solmusta on suora yhteys. Jos  $k$ :lla on enemmän kuin yksi edeltäjä arvo  $b_k$  jaetaan tasan kaikkien edeltäjien kesken. Esimerkiksi, jos solmulla  $k$  on kaksi lyhintä polkua, molempiin edeltäjiin lisätään arvo  $b_k/2$ .

- 4) Kun kaikki solmut on käyty läpi, tuloksena saadut arvot  $b_k$  kuvaavat sitä kuinka monella lyhyimmällä polulla solmu  $k$  on mukana, kun tutkitaan solmun  $j$  lyhyimpiä polkuja. Polun päätepisteet lasketaan myös kuuluvaksi lyhyimpään polkuun. Arvoa  $b_k$  pidetään juoksevana ja *betweenness*-mitta lasketaan kaikille mahdollisille  $n$ :lle arvolle solmusta  $j$ . Lopulliset juoksevat  $b_k$ -arvot ovat tarkka *betweenness*-mitta jokaiselle solmulle  $k$ .

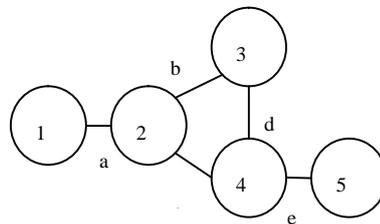
### **keskeisyys.m**

Keskeisyysmittoja laskeva MATLAB-funktio on nimeltään *keskeisyys.m* (ks. liite 4).

Funktiolle annetaan syötteenä linkit-matriisi, joka sisältää linkkien toimintatodennäköisyydet. Funktio palauttaa *betweenness*-arvovektorin (*betw*), jossa on kaikkien solmujen *betweenness*-mitta. Lisäksi funktio palauttaa haluttaessa *closeness* (*cl*)-, *out-strength* (*str\_out*)- ja *in-strength* (*str\_in*)-arvot vektorimuodossa. Funktio piirtää myös kuvaajat kustakin keskeisyysmitasta. Kuvaajissa on esitetty kunkin solmun prosentuaalinen keskeisyysmitan osuus verrattuna muihin solmuihin.

### **Mallin ominaisuuksien analyysi yksinkertaisessa verkossa**

Kuva 8 esittää yksinkertaisen verkon, jossa on viisi solmua ja viisi linkkiä. Tämän verkon tapauksessa esitetään miten kuvan verkosta voidaan laskea edellä esitetyt keskeisyysmitat *strength*, *closeness* ja *betweenness*. Lisäksi keskeisyysmittoja vertaillaan keskenään ja tehdään johtopäätöksiä linkkien keskeisyydestä.



Kuva 8. Esimerkkiverkko.

Linkkien toimintatodennäköisyydet ovat seuraavanlaiset:

$$P = \begin{bmatrix} 1 & 0.9 & 0 & 0 & 0 \\ 0.9 & 1 & 0.7 & 0.8 & 0 \\ 0 & 0.3 & 1 & 0.2 & 0 \\ 0 & 0.8 & 0.6 & 1 & 0.9 \\ 0 & 0 & 0 & 0.9 & 1 \end{bmatrix}$$

Lasketaan toimintatodennäköisyyksien avulla linkkien pituudet. Saadaan seuraavanlainen matriisi:

$$L = \begin{bmatrix} 0 & 0.1054 & \text{Inf} & \text{Inf} & \text{Inf} \\ 0.1054 & 0 & 0.3567 & 0.2231 & \text{Inf} \\ \text{Inf} & 1.2040 & 0 & 1.6094 & \text{Inf} \\ \text{Inf} & 0.2231 & 0.5108 & 0 & 0.1054 \\ \text{Inf} & \text{Inf} & \text{Inf} & 0.1054 & 0 \end{bmatrix}$$

Kun linkkien pituudet muutetaan painoiksi, saadaan painomatriisi:

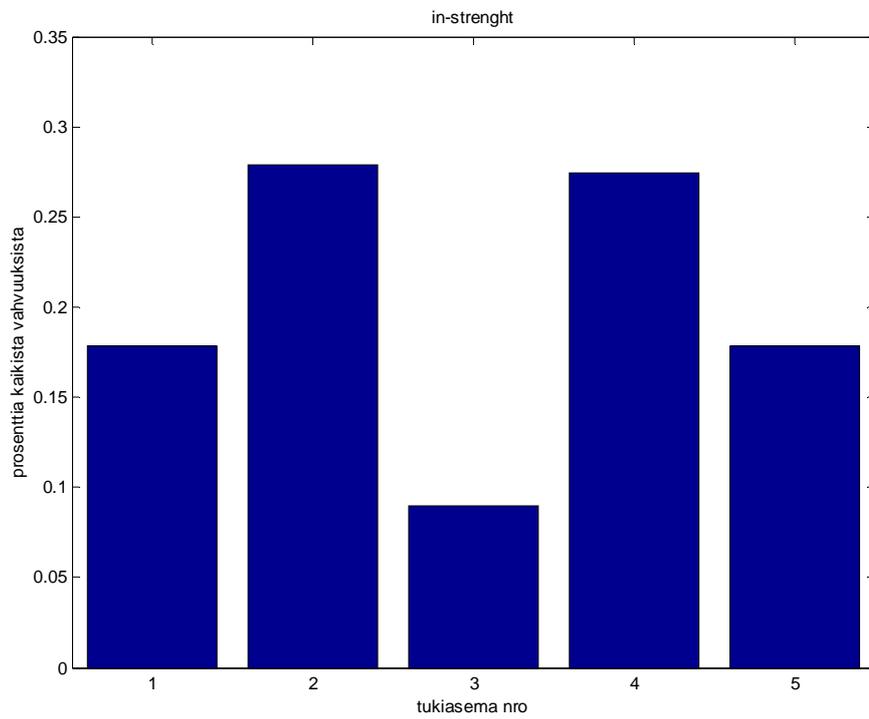
$$W = \begin{bmatrix} 0 & 9.4912 & 0 & 0 & 0 \\ 9.4912 & 0 & 2.8037 & 4.4814 & 0 \\ 0 & 0.8306 & 0 & 0.6213 & 0 \\ 0 & 4.4814 & 1.9576 & 0 & 9.4912 \\ 0 & 0 & 0 & 9.4912 & 0 \end{bmatrix}$$

*In-strength* ja *out-strength* saadaan laskettua painomatriisista helposti. Solmuun tulevien linkkien vahvuudet, eli *in-strength* solmulle  $i$  saadaan laskemalla sarakkeen  $i$  alkioden arvot yhteen. Solmusta lähtevien linkkien vahvuudet, eli *out-strength* solmulle  $i$  saadaan puolestaan laskemalla rivin  $i$  alkioden summa. Painomatriisi  $W$ :sta saadaan seuraavat vahvuudet:

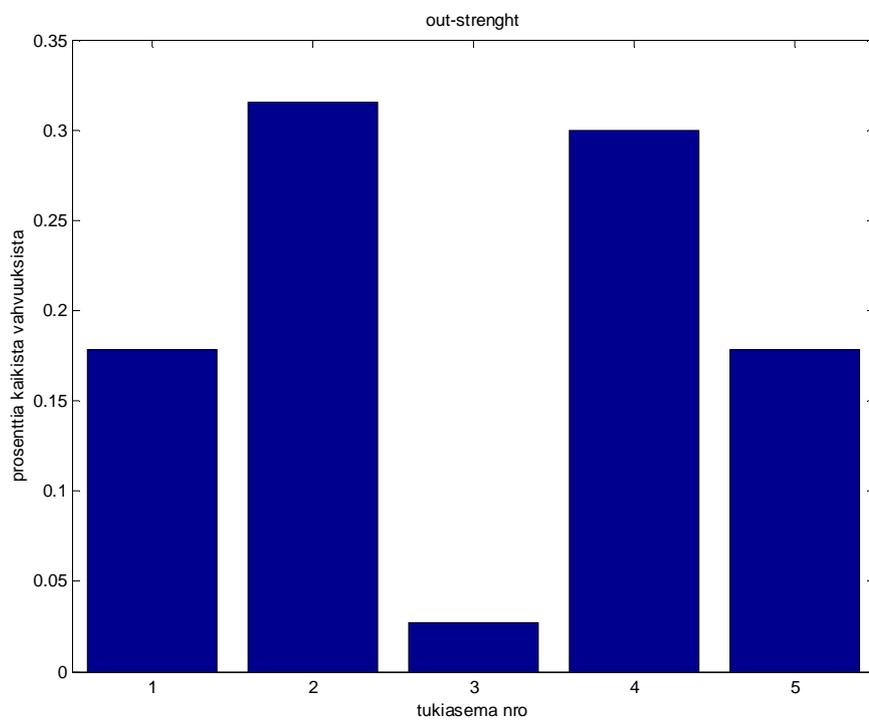
$$\textit{out-strength: } s_{o1} = 9.49 \quad s_{o2} = 16.77 \quad s_{o3} = 1.45 \quad s_{o4} = 15.93 \quad s_{o5} = 9.49$$

$$\textit{in-strength: } s_{i1} = 9.49 \quad s_{i2} = 14.80 \quad s_{i3} = 4.76 \quad s_{i4} = 14.59 \quad s_{i5} = 9.49$$

Huomataan, että solmun kolme *in-strength* arvo on selvästi suurempi kuin *out-strength* arvo. Näin pitääkin olla, koska solmuun tulevien linkkien todennäköisyydet ovat selvästi suuremmat kuin solmuista lähtevien linkkien todennäköisyys. Kuva 9 esittää prosentuaaliset *in-strength*-osuudet kullakin solmulla ja kuva 10 prosentuaaliset *out-strength*-osuudet. Kuvista huomataan vieläkin selvimmin, että solmu 3 on *strength* –mitoille mitattuna selkeästi vähiten keskeisin. Kun katsotaan verkon rakennetta ja linkkien todennäköisyysmatriisia, huomataan, että linkkien todennäköisyydet vaikuttavat *strength*-arvoon huomattavasti solmujen sijaintia enemmän.



Kuva 9. Verkon solmujen in-strenght-arvojen prosentuaaliset osuudet.



Kuva 10. Verkon solmujen out-strenght-arvojen prosentuaaliset osuudet.

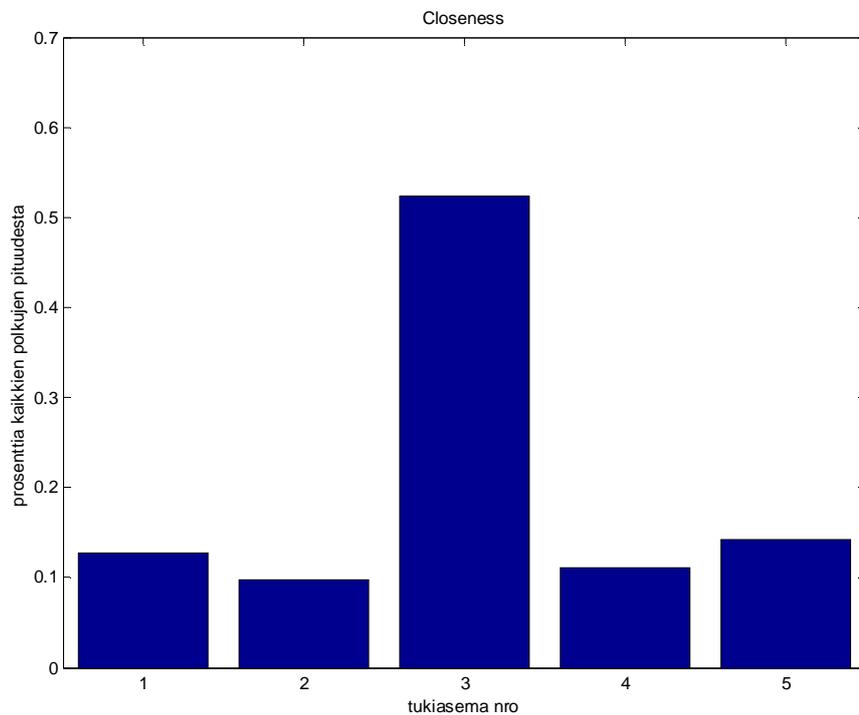
Nyt siirrytään *closeness* ja *betweenness* -arvojen tarkasteluun. Lasketaan ensiksi lyhyimmät polut Dijkstran algoritmilla. Saadaan seuraavanlainen matriisi:

$$SP = \begin{bmatrix} 0 & 0.1054 & 0.4620 & 0.3285 & 0.4339 \\ 0.1054 & 0 & 0.3567 & 0.2231 & 0.3285 \\ 1.3093 & 1.2040 & 0 & 1.4271 & 1.5325 \\ 0.3285 & 0.2231 & 0.5108 & 0 & 0.1054 \\ 0.4339 & 0.3285 & 0.6162 & 0.1054 & 0 \end{bmatrix}$$

*Closeness*-arvo solmulle  $i$  saadaan laskettua summaamalla yhteen rivin  $i$  kaikki alkiot. Solmuille saadaan seuraavat *closeness*-arvot:

$$c_1 = 1.32 \quad c_2 = 1.01 \quad c_3 = 5.47 \quad c_4 = 1.17 \quad c_5 = 1.48$$

Kuten arvoista nähdään, solmu 3 saa selvästi suurimman arvon, joka tässä tapauksessa tarkoittaa sitä, että se on keskimäärin kauimpana muista solmuista. Kuva 11 näyttää solmujen prosentuaaliset *closeness* -osuudet (tässä siis mitä pienempi prosenttiosuus sitä keskeisempi solmu). Kun vertaillaan tätä kuvaa *strength* -osuuksien kuviin, huomataan, että molemmat keskeisyysmitat antavat hyvin samanlaisen tuloksen solmu 3 on selkeästi muita epäkeskeisempi ja solmut 2 ja 4 ovat jonkin verran keskeisimpiä kuin solmut 1 ja 5.

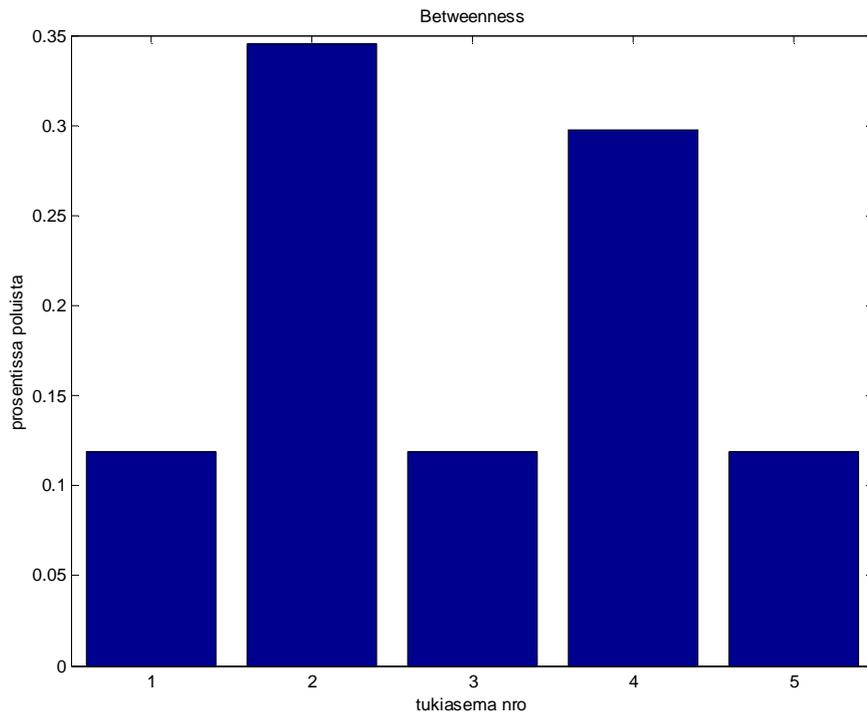


Kuva 11. Verkon solmujen closeness-arvojen prosentuaaliset osuudet.

*Betweenness* –arvot saadaan laskettua lyhyimpien polkujen matriisin avulla Newmanin algoritmillä. Tulokseksi saadaan:

$$b_1 = 5.0 \quad b_2 = 14.5 \quad b_3 = 5.0 \quad b_4 = 12.5 \quad b_5 = 5.0$$

Arvoja tarkastelemalla huomataan, että solmut 2 ja 4 ovat selvästi keskeisimmät ja solmut 1, 3 ja 5 ovat kaikki yhtä epäkeskeisiä. Kuva 12 esittää *betweenness* –mittojen prosentuaaliset osuudet kussakin solmussa. Tämä tulos eroaa merkittävästi *strength*- ja *closeness* –mitoista, joiden mukaan solmu 3 on selvästi epäkeskeisin. Kun tarkastellaan verkon rakennetta (kuva 8), *betweenness* –mitan keskeisyysarvio tuntuu huomattavasti järkevämältä kuin muiden keskeisyysmittojen tulokset. Kuvasta nähdään, että solmut 2 ja 4 ovat verkossa keskeisimmässä asemassa. Myös solmu 3 näyttäisi olevan keskeisemmässä asemassa kuin solmut 1 ja 5. Kun tarkastellaan toimintatodennäköisyysmatriisia huomataan kuitenkin, että solmun 3 linkkien toimintatodennäköisyydet ovat selvästi pienemmät kuin muiden solmujen välillä. Muut keskeisyysmitat painottavat näitä pieniä toimintatodennäköisyyksiä huomattavasti enemmän kuin verkon yleistä rakennetta. *Betweenness* –mitta näyttäisi taas pystyvän ottamaan huomioon sekä verkon rakenteen, että linkkien toimintatodennäköisyydet solmujen keskeisyyttä määrittettäessä. Näin ollen voidaan todeta, että viestiverkon solmujen keskeisyyttä tutkittaessa päähuomio kannattaa kiinnittää *betweenness*- arvoihin ja *strength*- ja *closeness*- arvoja kannattaa pitää lähinnä lisäinformaatiota antavina.



Kuva 12. Verkon solmujen *betweenness*-arvojen prosentuaaliset osuudet.

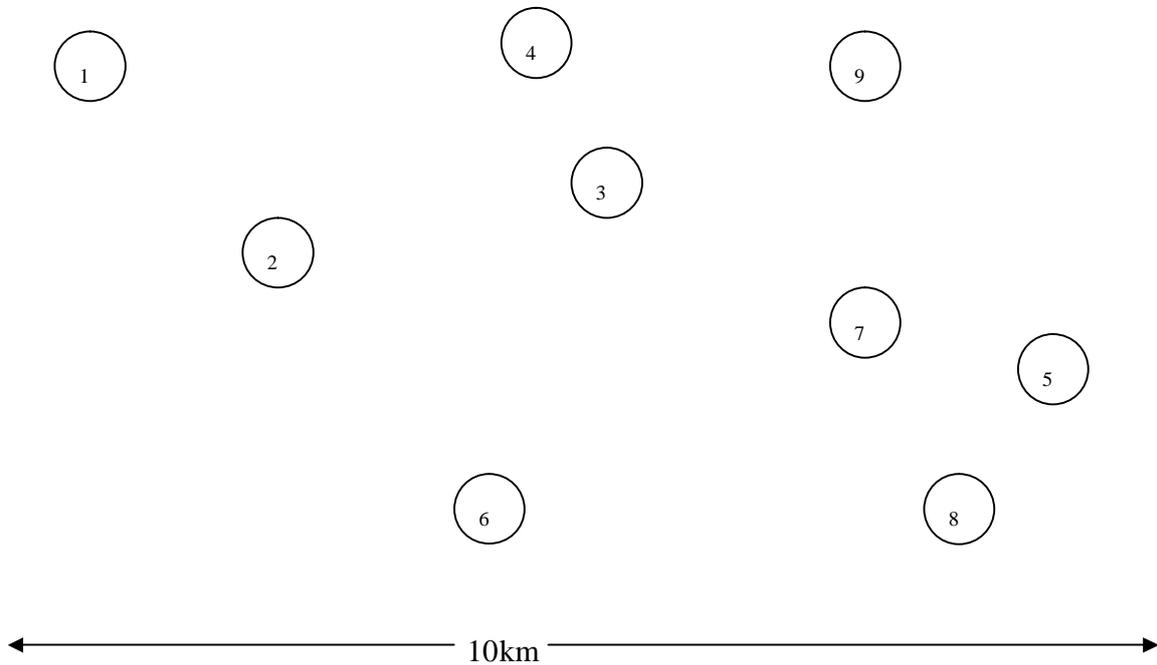
## Sovellusesimerkki luotujen työkalujen käytöstä

Havainnollistetaan seuraavaksi tekemiemme funktioiden toimintaa rajoitetun esimerkin voimin. Sovellamme MATLAB-algoritmeja yksinkertaiseen viestiverkkoesimerkkiin, jolloin saamme laskettua annetulle viestiverkolle erinäisiä tunnuslukuja. Tämän informaation avulla pyrimme analysoimaan viestiverkon ominaisuuksia kuten solmujen ja linkkien tärkeyksiä ja viestiverkon kriittisiä viestipolkuja. Analyysin perusteella pyrimme lopuksi ottamaan kantaa siihen, miten saadun informaation perusteella verkkoa kannattaisi vahvistaa, tai hyökkäysnäkökulmasta mitä yhteyksiä ja solmuja vastaan olisi järkevintä hyökätä suurimman häiriön aikaansaamiseksi.

Otamme tarkastelun alle viestiverkon, joka koostuu yhteensä 9 tukiasemasta. Kyseinen viestiverkko ei vastaa prikaatin viestiverkkoa, joka koostuu yli 20 solmusta. Suppeampi esimerkki havainnollistaa kuitenkin paremmin algoritmien toimintaa ja analyysissä saatavat tulokset ja päätelmät pätevät myös suuremmille viestiverkoille. Alla on esitetty tekstitiedosto, joka sisältää verkon tiedot sarakkeittain järjestyksessä P- ja I-koordinaatit (metreinä), tukiasemien radioiden kantamat (metreinä) ja tukiasemien toimintatodennäköisyydet.

860	7540	3000	0.97
2430	5320	3000	0.97
5100	5830	4000	0.97
4980	8010	3000	0.97
9750	2520	1500	0.95
4530	980	3000	0.97
8320	3880	3000	0.97
9010	540	1500	0.95
8420	7860	3000	0.97

Kuvaan 13 on havainnollistettu viestiverkon rakennetta sijoittamalla tukiasemat annettuja koordinaatteja vastaaville paikoille.



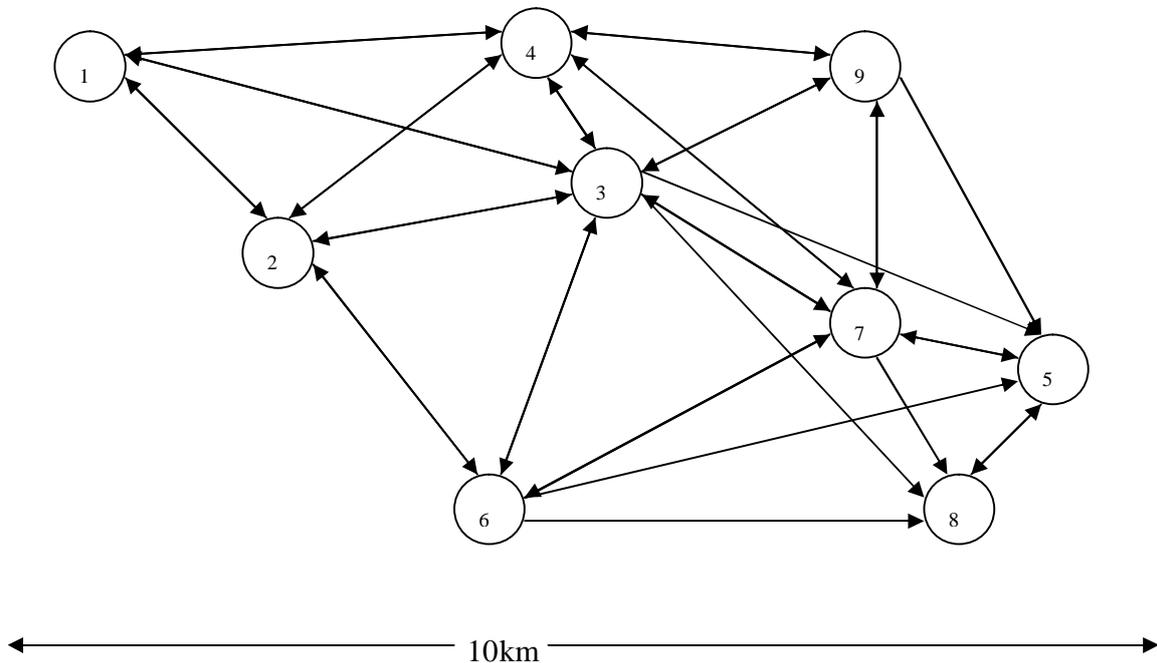
Kuva 13. Tarkasteltava verkko kartalla.

Syötetään seuraavaksi tekstitiedosto lue\_verkko.m-ajotiedostoon. Taulukossa 3 on esitetty algoritmista saadut linkkien toimintatodennäköisyydet. Esimerkiksi matriisin ensimmäisellä riviltä voidaan lukea tukiasemasta 1 lähtevien linkkien toimintatodennäköisyydet. Diagonaalilla on aina ykkösiä, koska solmulla on aina yhteys itsensä kanssa. Huomattavaa on, että matriisi ei ole symmetrinen, joka tarkoittaa sitä, että radiosta A ei välttämättä ole yhteyttä radioon B, vaikka yhteys on olemassa radiosta B radioon.

Taulukko 3. Esimerkkiverkon linkkien toimintatodennäköisyydet (alkio  $a_{ij}$  vastaa yhteyttä solmusta  $i$  solmuun  $j$ ).

1,00	0,82	0,12	0,15	0,00	0,00	0,00	0,00	0,00
0,82	1,00	0,82	0,19	0,00	0,10	0,00	0,00	0,00
0,37	0,86	1,00	0,89	0,14	0,19	0,81	0,09	0,81
0,15	0,19	0,85	1,00	0,00	0,00	0,06	0,00	0,36
0,00	0,00	0,00	0,00	1,00	0,00	0,17	0,15	0,00
0,00	0,10	0,09	0,00	0,05	1,00	0,10	0,12	0,00
0,00	0,00	0,19	0,06	0,87	0,10	1,00	0,39	0,17
0,00	0,00	0,00	0,00	0,15	0,00	0,00	1,00	0,00
0,00	0,00	0,18	0,36	0,04	0,00	0,17	0,00	1,00

Kuvassa 14 on esitetty graafisesti tukiasemien välille muodostuneet yhteydet. Nuolen suunta ilmaisee mihin suuntaan linkki toimii.



Kuva 14. Viestiverkkoon muodostuvat linkit.

Kuva 14 paljastaa miten epäsymmetrinen muodostuva viestiverkko on. Vaikka esimerkkiverkon koko on suhteellisen pieni, on päätelmien teko ilman aputyökaluja hyvin vaikeaa ellei mahdotonta.

Syötetään seuraavaksi linkkien toimintatodennäköisyydet sisältävä taulukko 3 ja tekstitiedostossa ilmenevät solmujen toimintatodennäköisyydet laske\_TNmatriisi.m-komentotiedostoon ja lasketaan kaikkien solmujen välisten yhteyksien toimintatodennäköisyydet. Alla oleva taulukko 4 sisältää nämä todennäköisyydet.

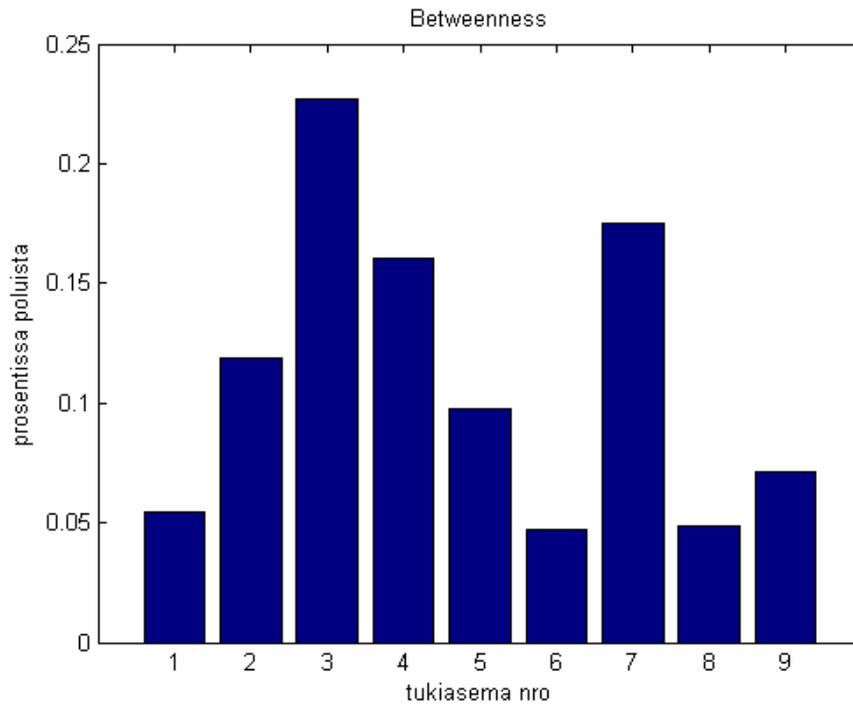
Taulukko 4. Solmujen välisten yhteyksien toimintatodennäköisyydet

1,00	0,81	0,71	0,68	0,52	0,25	0,59	0,32	0,62
0,84	1,00	0,83	0,78	0,61	0,29	0,69	0,37	0,72
0,79	0,87	1,00	0,89	0,71	0,31	0,80	0,43	0,84
0,71	0,78	0,84	1,00	0,63	0,27	0,71	0,39	0,77
0,04	0,05	0,05	0,05	1,00	0,03	0,16	0,19	0,06
0,16	0,18	0,18	0,17	0,23	1,00	0,22	0,21	0,16
0,22	0,25	0,27	0,27	0,81	0,16	1,00	0,45	0,32
0,01	0,01	0,01	0,01	0,14	0,00	0,02	1,00	0,01
0,35	0,38	0,42	0,44	0,39	0,14	0,42	0,22	1,00

Koska matriisissa ei ole yhtään nollia, viestiverkon jokaisesta solmusta päästään jokaiseen solmuun. Jos matriisi sisältäisi nollia, joku tai jotkut solmut olisivat eristyksissä muusta viestiverkosta ja viestiverkko olisi rakenteeltaan susi. Tällöin olisi joko lisättävä viestiverkkoon solmuja tai järjesteltävä viestiverkon solmut uudelleen.

Esimerkki viestiverkossa näyttäisi olevan ongelmia etenkin solmujen 5 ja 8 kanssa, joista lähtevät yhteyksien toimintatodennäköisyydet ovat luokkaa  $<0.1$ . Tämä selittynee sillä, että solmut 5 ja 8 ovat liittyneitä muihin solmuihin vain solmun 7 kautta, jolloin ne ovat hyvin vahvasti riippuvaisia siitä. Lisäksi solmulla 8 on yhteys vain solmuun 5, mikä pienentää sen yhteyksien toimintatodennäköisyyksiä huomattavasti. Viestiverkko ei toimi riittävällä varmuudella näiden solmujen osalta ja tästä syystä solmujen 5, 6 ja 7 radioiden tehoja olisi joko nostettava tai sitten nämä solmut olisi sijoitettava keskeisemmälle paikalle viestiverkossa. Solmujen välillä on siis havaittavissa selviä eroja, ääriesimerkkinä solmut 3 ja 8, joista solmu 3 saa viestiverkon parhaat yhteyksien toimintatodennäköisyydet. Tähän syynä ovat solmujen radioiden kantamien erot ja tukiasemien toimintatodennäköisyyksien eroavaisuus. Lisäksi solmulla 3 on paljon keskeisempi asema verkossa, koska sillä on suora linkkiyhteys viestiverkon kaikkiin muihin solmuihin, kun solmu 8 on yhteydessä vain solmuun 5. Solmun keskeinen sijainti näyttäisi siis korreloivan vahvasti siitä lähtevien yhteyksien toimintatodennäköisyyksien kanssa. Viestiverkon kriittiset alueet sijaitsevat viestiverkon laidoilla, jossa sijaitsevilla solmuilla on vain muutama yhteys muihin tukiasemiin. Taulukosta 4 luonnollisesti nähdään, että mitä pitempi matka kahden solmun välillä on, sitä pienempi on yhteyden toimintatodennäköisyys.

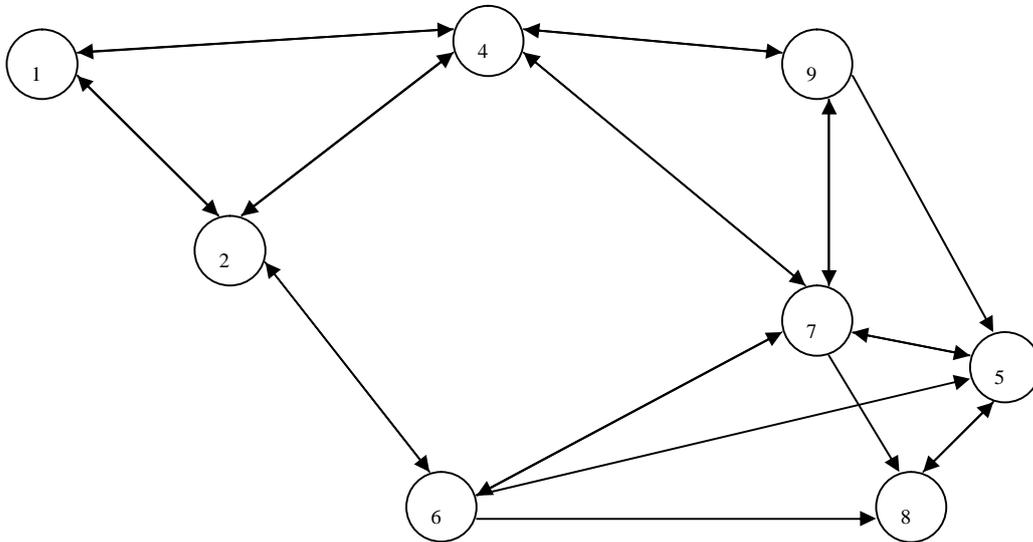
Lasketaan vielä betweenness-algoritmillä keskeisyysmitat viestiverkon solmuille. Kuvassa 15 on esitetty Betweenness-mitat verkon solmuille.



Kuva 15. Betweenness-mitat viestiverkon solmuille.

Betweenness analyysin mukaan solmut 3 ja 7 olisivat tärkeimpiä solmuja viestiverkossa. Tämä on hyvinkin järkevä tulos sillä ilman solmua 7 solmut 8 ja 9 eristyisivät kokonaan muusta viestiverkosta. Solmu 3:n tärkeys selittyy sen keskeisyydellä viestiverkossa. Näiden tulosten valossa ensisijaisen tärkeää viestiverkon toiminnan kannalta olisi solmujen 3 ja 7 toiminnan turvaaminen. Vastaavasti vihollisen toiminnan kannalta ajateltuna häirintätoimet tulisi kohdistaa ennen kaikkea näihin kahteen solmuun suurimman vaikutuksen aikaansaamiseksi.

Tarkastellaan solmun 3 tärkeyttä viestiverkon kokonaistoiminnan kannalta poistamalla se viestiverkosta. Tämä havainnollistaa tilannetta, jossa vihollinen häiritsee onnistuneesti solmun 3 toimintaa. Kuva 16 esittää viestiverkkoa tässä uudessa tilanteessa.



Kuva 16. Esimerkkiviestiverkko ilman solmua 3.

Taulukossa 5 on esitetty solmujen välisten yhteyksien toimintatodennäköisyydet, kun solmu 3 on poistettu viestiverkosta.

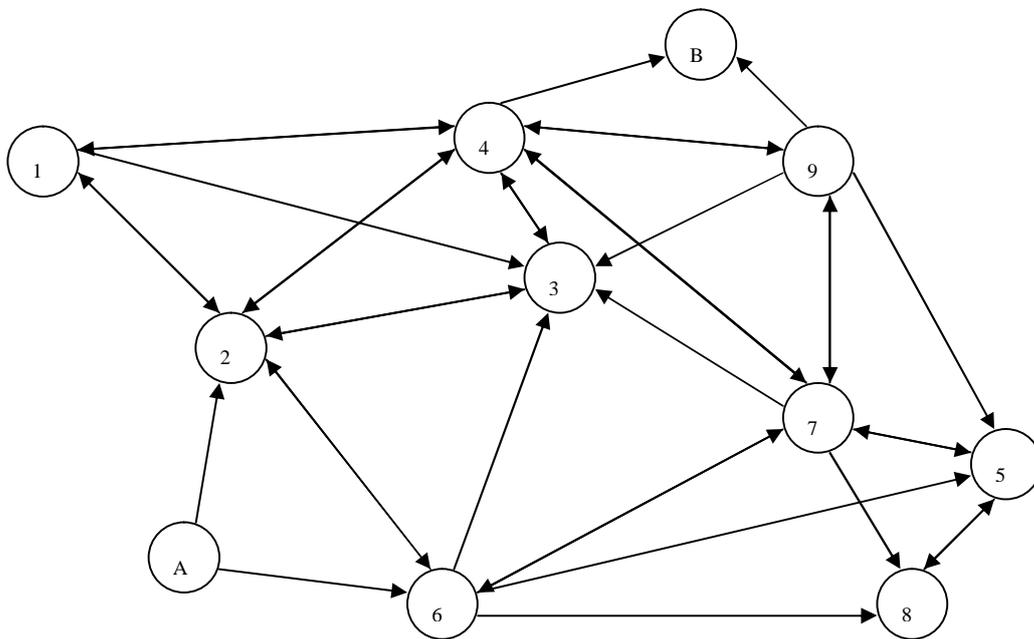
Taulukko 5. Solmujen väliset toimintatodennäköisyydet ilman solmua 3.

1,00	0,78	0,27	0,04	0,08	0,04	0,03	0,10
0,78	1,00	0,28	0,04	0,09	0,04	0,03	0,10
0,27	0,28	1,00	0,10	0,04	0,11	0,05	0,34
0,01	0,01	0,02	1,00	0,02	0,16	0,19	0,03
0,08	0,09	0,04	0,15	1,00	0,11	0,16	0,03
0,04	0,04	0,11	0,81	0,10	1,00	0,43	0,17
0,00	0,00	0,00	0,14	0,00	0,02	1,00	0,00
0,10	0,10	0,35	0,18	0,03	0,19	0,09	1,00

Vertaamalla taulukkoa 5 taulukkoon 4 nähdään, että solmun 3 poistaminen laskee vahvasti melkein kaikkien muiden yhteyksien toimintatodennäköisyyksiä. Tämä tukee Betweenness-algoritmin tulosta. Viestiverkkoesimerkissä vain yhden solmun poistaminen aiheuttaa käytännössä koko viestiverkon luhistumisen. Solmun 3 rooli viestiverkossa on

tässä mielessä ylikorostunut ja sen tueksi keskeiseen asemaan olisi siirrettävä joku toinen solmu tai koko viestiverkon rakenne olisi mietittävä uudelleen. Analyysin perusteella voidaan vetää johtopäätös, että Betweenness-algoritmi kuvaa hyvin viestiverkon solmujen tärkeyttä ja sen antamien tulosten perusteella voidaan priorisoida solmujen suojausta tai toisaalta viestiverkkoa vastaan harjoitettavaa häirintää.

Demonstroidaan vielä funktion laske\_TN.m toimintaa tarkastelemalla tilannetta, jossa radio A liittyy solmuihin 2 ja 6 lähettääkseen viestin radiolle B, joka on liittynyt solmuihin 4 ja 9. Lasketaan yhteyden toimintatodennäköisyys laske\_TN.m-komentotiedostolla, antamalla funktiolle jälleen parametreina linkkien ja solmujen toimintatodennäköisyydet ja vektorit, jotka kertovat mihin solmuihin radiot liittyvät ( $A = [2\ 6]$  ja  $B = [4\ 9]$ ). Kuva 17 ilmentää tilannetta.



Kuva 17. Kaksi erillistä radiota liittyy runkoverkkoon.

Tulokseksi saadaan 0.81, joka on suhteellisen hyvä yhteyden toimintatodennäköisyys. Liittymällä useampaan runkoverkon solmuun saavutetaan suurempi yhteyden toimintatodennäköisyys, kun liityttäisiin vain yhteen solmuun.

## Pohdinnat ja yhteenveto

Projektityön tuloksena saatiin kaksi erillistä mallia: malli viestiverkon todennäköisyyksien laskemiseksi ja malli viestiverkon solmujen tärkeyden mittaamiseksi.

Viestiverkon toimintatodennäköisyydet määrittävä malli näistä toteutettiin Monte Carlo –simulaatioon perustuen. Alun perin idean oli lähestyä ongelmaa minimikatkosjoukkojen kautta, mutta tämä lähestymistapa hylättiin, kun oivallettiin, että Monte Carlo –simulaatiolla haluttuihin tuloksiin päästää huomattavasti yksinkertaisemmin ja kätevämmiin. Tähän malliin sisältyy kolme MATLAB:lla ajettavaa tiedostoa: lue\_verkko.m, laske\_TNmatriisi.m ja laske\_TN.m. Näiden avulla voidaan koordinatti- ja kantamatietoina annetulle viestiverkolle laskea yhteystodennäköisyysmatriisi (kertoo todennäköisyyden, että solmusta  $i$  saadaan yhteys  $j$ :hin) sekä määrittää kahden runkoverkkoon liittyvän radion yhteystodennäköisyys. Kohdeorganisaation antamiin tavoitteisiin laskenta-ajan ja laskentatarkkuuden suhteen päästiin valitsemalla simulaatiokierrosten lukumääräksi 10000. Tällöin laskenta-aika pöytätietokoneessa on 20-50 solmua sisältävissä verkoissa luokka 5 – 60s molempien algoritmien tapauksessa. Algoritmien antama tarkkuus on tällöin 0.02 (95 % havainnoista kuuluu välille  $p \pm 0.02$ ).

Malli viestiverkon tukiasemien (solmujen) tärkeyden mittaamiseksi toteutettiin keskeisyysmittoja käyttämällä. Malliin liittyy MATLAB-tiedosto keskeisyys.m, joka laskee annetun verkon solmuille keskeisyysmitat out-strength, in-strength, closeness ja betweenness. Näistä Puolustusvoimien viestiverkon tapauksessa käyttökelpoisin on betweenness. Betweenness-arvo mittaa tehokkaasti solmujen tärkeyttä verkossa, kuten havaittiin edellä esitetystä sovellusesimerkissä, joissa suurimman betweenness-arvon omaavan solmun poistaminen verkosta aiheutti sen, että kaikki yhteystodennäköisyydet laskivat merkittävästi.

Raportin johdannossa määritettyihin tavoitteisiin on siis päästy. Muina tavoitteina mainittu hyödyllisen tiedon tuottaminen esim. viestiverkkojen suunnittelua ja elektronista sodankäyntiä varten toteutui osittain. Raportin koko ei antanut periksi laajamittaista analyysia otollisesta verkon rakenteesta yms.; siispä kohdeorganisaation kannalta hyödyllistä tietoa on tuotettu vain mallien ja sovellusesimerkkien muodossa.

Valitsemamme lähestymistavat ongelmiin (Monte Carlo –simulaatio ja keskeisyys-mitat) ovat erittäin onnistuneita. Vaikka luodut mallit ovat onnistuneita ja sopivat hyvin tarkoituksiinsa, voidaan niitä halutessa kehittää edelleen. On mahdollista, että todennäköisyysmallien käyttämää laskenta-aikaa voidaan pienentää siirtymällä esim. C-ohjelmointikieleen. Todennäköisyyksien tarkkuutta on kuitenkin vaikea kasvattaa, mikäli pitäydytään Monte Carlo –menetelmissä. Kuitenkin verkon todennäköisyyksien laskemiseen Monte Carlo –menetelmä on melko varmasti paras lähestymistapa.

Solmujen tärkeyttä mittaavaa malli nojaa betweenness-mitan arvioon solmujen tärkeydestä. Betweenness-mitan heikkoutena voidaan pitää sitä, että se ottaa huomioon pelkästään lyhyintä (todennäköisintä) polkua kulkevat reitit ja niiden varrella olevat solmut. Viesti kulkee luonnollisesti myös muita kuin lyhyimpiä reittejä pitkin, näitä reittejä ei kuitenkaan oteta solmujen keskeisyyttä määritettäessä huomioon. Betweenness-algoritmin muunnelmia, jotka ottavat myös muut reitit huomioon, on kehitelty esim.

Newmanin satunnaiskävelyyn perustuva kehittyneempi betweenness-keskeisyys [3]. Käytännössä näiden algoritmien antamat tulokset eivät kuitenkaan poikkea merkittävästi käyttämästämme lyhyimpiin polkuihin perustuvasta algoritmista. Ominaisarvovektoriajotelmalla perustuva eigenvector-keskeisyys [1] on myös yleisesti käytetty keskeisyysmitta, joka voisi betweenness-keskeisyyden lisäksi tuoda myös olennaista tietoa viestiverkon tukiasemien keskeisyydestä. Algoritmin soveltamisessa suunnatuille verkoille on kuitenkin omat ongelmansa.

Luotuja malleja voidaan käyttää suoraan avuksi Puolustusvoimien viestiverkoissa. Nämä uudentyyppiset verkot ovat ns. ad hoc-verkkoja, joissa viesti voi kulkea solmujen välillä kaikkia sallittuja reittejä pitkin. Mallien avulla voidaan tutkia esimerkiksi optimaalista viestiverkon rakennetta (mikä viestiverkko kestää parhaiten häirinnän ja muut hyökkäykset?) ja elektronisen sodankäynnin vaikutusta prikaatin toimintaan. Eräs kiinnostava lähestymistapa jälkimmäiseen olisi esim. tutkia skenaarioanalyysin kautta elektronisen häirinnän vaikutusta viestiverkon ja koko prikaatin toimintaedellytyksiin. Eri skenaarioissa voitaisiin luoda todenmukaisia elektronisia hyökkäyksiä, joissa hyökkääjä estää tai häiritsee tiettyjen linkkien ja solmujen toimintaa.

## Lähdeluettelo

[1] Jari Seppälä, Electronic Warfare in future communication environment, *Military technology research No 21 in Report series 1, National Defence College, Department of Military Technology* (2005), ISBN: 951-25-1623-3

[2] Pertti Laininen; 1998. Todennäköisyys ja sen tilastollinen soveltaminen. Helsinki. Hakapaino Oy (1999). ISBN: 951-672-279-2

[3] Mat-2.104 Tilastollisen analyysin perusteet luentokalvot:  
<http://www.sal.tkk.fi/Opinnot/Mat-2.104/>

[4] Mat-2.170 Simulointi luentokalvot: <http://www.sal.tkk.fi/Opinnot/Mat-2.104/>

[5] M.E. J. Newman, The mathematics of networks, *Department of Physics and Center for the Study of Complex Systems, University of Michigan, Ann Arbor, MI 48109-1040*

[6] M. E. J. Newman, Scientific collaboration networks. II. Shortest paths, weighted networks, and centrality, *Physical Review E*, 64 (2001)

[7] M. E. J. Newman, A measure of betweenness centrality based on random walks, *Department of Physics and Center for the Study of Complex Systems, University of Michigan, Ann Arbor, MI 48109-1020*

[8] <http://www.mil.fi/paaesikunta/tiedotteet/994.dsp>: "Ohjelmistoradio mahdollistaa yhteisoperaatiot eurooppalaisessa kriisinhallinnassa"

[9] [http://www.proessori.fi/es04/ARKISTO/SPONTAANIT\\_VERKOT.HTM](http://www.proessori.fi/es04/ARKISTO/SPONTAANIT_VERKOT.HTM):  
"Liikkuvuus irti kahleistaan"

# Liitteet

## Liite 1: MATLAB-tiedosto lue\_verkko.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% lue_verkko.m
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Laskee koordinaatti- ja kantamatiedoista linkkien ja solmujen toiminta-
% todennäköisyydet.
% Parametrit: teksti- tai exel-tiedosto
% Palautusarvot: solmut (solmujen toimintatodennäköisyydet) ja linkit
% (linkkien toimintatodennäköisyydet)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [solmut, linkit] = lue_verkko(a)
%luetaan data
b = length(a);

if('t' == a(b))
matriisi = load(a);
else
matriisi = xlsread(a);
end;

[M,N] = size(matriisi);
%solmujen toimintatodennäköisyydet
solmut = matriisi(:,4);
%lasketaan eri solmujen väliset etäisyydet
etaisyydet = zeros(M);

for m = 1:M
    for n = 1:M
        etaisyydet(m,n) = sqrt((matriisi(m,1) - matriisi(n,1))^2 + (matriisi(m,2) - matriisi(n,2))^2);
    end
end

%lasketaan linkkien toimintatodennäköisyydet voimaa-funktion avulla
linkit = zeros(M);
for m = 1:M
    for n = 1:M
        %lasketaan nimen omaan tn., että on yhteys m:stä n:ään
        %nämä eivät ole symmetrisiä
        linkit(m,n) = linkki_tn(matriisi(m,3), etaisyydet(m,n));
    end
end
```

```
end
```

```
function tod = linkki_tn(kantama, etaisuus)  
%aprosimoidaan paloitteisella lineaarisella funktiolla  
% etäisyydellä 0 todennäköisyys 1, kantaman päässä tod 0.8,  
%1.2kantaman päässä tod 0.2 ja 2*kantama tod on 0  
  
if(etaisuus > 2 * kantama)  
    tod = 0;  
elseif(etaisuus <= kantama)  
    tod = ((-0.2*(etaisuus / kantama) + 1);  
elseif((etaisuus > kantama) && (etaisuus <= 1.2*kantama))  
    tod = - 3 * etaisuus / kantama + 3.8;  
else  
    tod = -0.25 * etaisuus / kantama + 0.5;  
end;
```

## ***Liite 2: MATLAB-tiedosto laske\_TNmatriisi.m***

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
% laske_TNmatriisi.m  
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
% Laskee verkon kokonaistoimintatodennäköisyydet.  
% Parametrit: solmut (solmujen toimintatodennäköisyydet) ja linkit  
% (linkkien toimintatodennäköisyydet).  
% Palautusarvot: TNmatriisi (yhteystodennäköisyysmatriisi, jonka alkio  
% a_ij sisältää todennäköisyyden, että solmusta i saadaan yhteys j:hin).  
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
function TNmatriisi = laske_TNmatriisi(solmut,linkit)  
  
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
% Alkuarvot %%%%%%%%%  
% Simulaatiokierrosten lukumäärä  
rounds=10000;  
  
%Iteraatio alkaa  
onnistuneet=zeros(length(solmut));  
for round=1:rounds,  
  
    toimivat=[];  
    %arvotaan toimivat solmut  
    for i=1:length(solmut),  
        if solmut(i) > rand(1)
```

```

        toimivat=[toimivat,i];
    end
end

%Muodostetaan matriisi vain toimiville solmuille
runko=zeros(length(toimivat));
for i=1:length(toimivat),
    for j=1:length(toimivat),
        if linkit(toimivat(i),toimivat(j)) >= rand(1)
            runko(i,j)=1;
        end
    end
end
end

%Lasketaan mistä solmusta on yhteys mihinkin solmuun
yhteydet=runko^length(toimivat);
%Tutkitaan mitkä palat yhteydessä
for i=1:length(toimivat),
    for j=1:length(toimivat),
        if yhteydet(i,j) > 0
            onnistuneet(toimivat(i),toimivat(j)) = onnistuneet(toimivat(i),toimivat(j)) + 1;
        end
    end
end
end
end

%Lasketaan viestiverkon toimintatodennäköisyysarvio
TNmatriisi = onnistuneet / rounds;

% asetetaan vielä TN_matriisiin diagonaalit ykkösiksi
for i=1:length(solmut)
    TNmatriisi(i,i)=1;
end
end

```

### ***Liite 3: MATLAB-tiedosto laske\_TN.m***

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% laske_TN.m
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Laskee todennäköisyyden, että radiosta 1 saadaan yhteys radioon 2.
% Parametrit: solmut (solmujen toimintatodennäköisyydet), linkit
% (linkkien toimintatodennäköisyydet), liitos1 (vaakavektori, joka sisältää
% niiden solmujen numerot, joihin radio1 on liittynyt) ja liitos2

```

```

% (vaakavektori, joka sisältää niiden solmujen numerot, joihin radio2 on
% liittynyt).
% Palautusarvot: TN (yhteystodennäköisyys).
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function TN = laske_TN(solmut,linkit,liitos1,liitos2)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Alkuarvot %%%%%%%%%
% simulaatiokierrosten lukumäärä
rounds=10000;

%Muodostetaan bin-vektori, jossa 1 jos solmuun liitytään
liitos1vek=zeros(length(solmut));
liitos2vek=zeros(length(solmut));
for i=1:length(liitos1),
    liitos1vek(liitos1(i))=1;
end

for i=1:length(liitos2),
    liitos2vek(liitos2(i))=1;
end

%Iteraatio alkaa
onnistuneet=0;
for round=1:rounds,

    toimivat=[];
    liitetyt1=[]; % sisältää indeksit toimivat-vektorissa
    liitetyt2=[];

    %arvotaan toimivat solmut
    for i=1:length(solmut),
        if solmut(i) > rand(1)
            toimivat=[toimivat,i];
            if liitos1vek(i) == 1
                liitetyt1=[liitetyt1,length(toimivat)];
            end
            if liitos2vek(i) == 1
                liitetyt2=[liitetyt2,length(toimivat)];
            end
        end
    end

    %Muodostetaan matriisi vain toimiville solmuille

```

```

runko=zeros(length(toimivat));
for i=1:length(toimivat),
    for j=1:length(toimivat),
        if linkit(toimivat(i),toimivat(j)) >= rand(1)
            runko(i,j)=1;
        end
    end
end

%Lasketaan mistä solmusta on yhteys mihinkin solmuun
yhteydet=runko^length(toimivat);

%Tutkitaan kulkeeko viesti arvotussa verkossa liitos1 -> liitos2
apu=0;
for i=1:length(liitetyt1),
    for j=1:length(liitetyt2),
        apu = apu + yhteydet(liitetyt1(i),liitetyt2(j));
    end
end

%Jos viesti kulkee, kasvatetaan onnistuneita
if apu > 0
    onnistuneet=onnistuneet+1;
end
end

%Lasketaan yhteyden toimintatodennäköisyysarvio
TN = onnistuneet / rounds;

```

## ***Liite 4: MATLAB-tiedosto keskeisyys.m***

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% keskeisyys.m
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Laskee halutut keskeisyysmitat annetun verkon solmuille ja piirtää
% kuvaajat mittojen prosentuaalisesta jakaumasta yli solmujen.
% Parametrit: linkit (linkkien toimintatodennäköisyydet).
% Palautusarvot: betw (betweenness), cl (closeness)
% str_out(out-strength), str_in(in-strength).
% Palautusarvot ovat vektoreita, jossa on keskeisyysmittojen arvot
% kullekin solmulle.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [betw,cl,str_out,str_in] = keskeisyys(linkit)

```

```

koko=size(linkit,1); %linkkimatriisin koko
sp=zeros(koko);      %alustetaan shortest paths -matriisi

%lasketaan linkeille painot
for(i=1:koko)
    for(j=1:koko)
        painot(i,j)=-log(linkit(i,j));
    end
end

%%%% SHORTEST PATHS
for(i=1:koko)
    solmut=zeros(koko,1);      %alustetaan solmut
    lapikaynti=zeros(koko,1); %0, jos estimated, 1, jos exact
    apulk=solmut;              %apumuuttuja, jotta löydetään helposti pienin solmu, jota ei ole käyty
jo läpi

%muutetaan tutkittava solmu nolllaksi ja muut äärettömäksi
    for(j=1:koko)
        solmut(j,1)=inf;
        apulk(j,1)=inf;
    end
    solmut(i,1)=0;
    apulk(i,1)=0;

%Käydään vuorotellen kaikki solmut läpi, aloittaen pienimmästä ei-läpikäydystä solmusta
    while (min(lapikaynti) == 0)
        [y,ind]=min(apulk);

%Käydään läpi naapurisolmut, ja muutetaan tarvittaessa arvoa
        for(k=1:koko)
            if(painot(k,ind)+y < solmut(k,1))
                solmut(k,1)=painot(k,ind)+y;
                apulk(k,1)=painot(k,ind)+y;
            end
        end
        lapikaynti(ind)=1; %Merkitään, että kyseinen solmu on käyty läpi
        apulk(ind)=inf;
    end

%Tallennetaan solmun i shortest paths
    sp(:,i)=solmut;

```

```

end

%%% BETWEENNESS
betw=zeros(koko,1);      %alustetaan 'betweenness-laskuri'
betw_solmut=zeros(koko);
%muodostetaan yhteydet kertova matriisi (1, jos on yhteys, muuten 0)
for(i=1:koko)
    for(j=1:koko)
        if(linkit(i,j)~=0)
            yhteydet(i,j)=1;
        end
    end
end
end

%Käydään kaikki solmut järjestyksessä läpi
for(i=1:koko)
    b=ones(koko,1);      %annetaan kaikkien solmujen b-arvoksi 1
    spi=sp(i,:);
    [spis,ind]=sort(spi); %lajitellaan shortest pathsien pituuden mukaan
    kaytylapi=zeros(koko,1);%alustetaan taulukko läpikäydyille solmuille
    kaytylapi(i)=1;      %ja merkitään tutkittava solmu läpikäydyksi

    %Tutkitaan lähimmät naapurit aloittaen kauimmasta solmusta
    for(j=0:koko-1)
        kaytylapi(ind(koko-j),1)=1; %merkitään, että kaukaisin solmu on käyty läpi
        predec=0;

        %Lasketaan monta edeltäjää kauimmaisella solmulla on
        for(k=1:koko)
            if(yhteydet(k,ind(koko-j))==1 && kaytylapi(k,1)==0)
                predec=predec+1;
            end
        end

        %Kasvatetaan b-arvoa, jos solmu on tutkittavan ja kauimmaisen solmun välisellä lyhimällä reitillä
        %b-arvoa kasvatetaan luvun tutkittavan solmun b-arvo/edeltäjien määrä verran
        for(k=1:koko)
            if(yhteydet(k,ind(koko-j))==1 && kaytylapi(k,1)==0)
                b(k)=b(k)+b(ind(koko-j))/predec;
            end
        end
    end
end
betw=betw+b; %lisätään betw-lukuun tutkitun solmun betweenness-arvot

```

```

end

%lasketaan betweenness_prosentit
betw_sum=sum(betw);
betw_pct=betw/betw_sum;
figure(1);
bar(betw_pct);
title('Betweenness');
ylabel('prosentissa poluista');
xlabel('tukiasema nro');

%%%% CLOSENESS
%lasketaan shortest pathsien summat kussakin solmussa
cl=(sum(sp));
cl_sum=sum(cl);
cl_pct=cl/cl_sum;
figure(2);
bar(cl_pct);
title('Closeness');
ylabel('prosenttia kaikkien polkujen pituudesta');
xlabel('tukiasema nro');

%%%% STRENGTH
%muutetaan painot siten, että vahvalla linkillä on suuri paino ja heikolla pieni
vahvuus=zeros(koko);
for(i=1:koko)
    for(j=1:koko)
        if(painot(i,j)~=0)
            vahvuus(i,j)=1/painot(i,j);
        end
    end
end

%lasketaan out- ja in-strengthit
str_in=sum(vahvuus);
str_out=sum(vahvuus');

%lasketaan prosentit ja piirretään kuvat
str_out_sum=sum(str_out);
str_out_pct=str_out/str_out_sum;
figure(3);
bar(str_out_pct);
title('out-strength');

```

```

ylabel('prosenttia kaikista vahvuuksista');
xlabel('tukiasema nro');

str_in_sum=sum(str_in);
str_in_pct=str_in/str_in_sum;
figure(4);
bar(str_in_pct);
title('in-strenght');
ylabel('prosenttia kaikista vahvuuksista');
xlabel('tukiasema nro');

```

### ***Liite 5: Kohdeorganisaatiolta saatu esimerkkiverkko.***

Tiedot ovat sarakkeittain järjestyksessä P-koordinaatti, I-koordinaatti, kantama ja radion toimintatodennäköisyys. P-koordinaatti, I-koordinaatti ja kantama on annettu metreissä.

9100	15200	10000	0.99
15400	21100	10000	0.99
18300	12200	10000	0.99
24100	18500	10000	0.99
30100	12700	10000	0.99
33200	18000	10000	0.99
5800	17200	5000	0.95
11900	27200	5000	0.95
13600	25400	5000	0.95
24100	22300	5000	0.95
29800	20600	5000	0.95
35400	22300	5000	0.95
39000	14700	5000	0.95
34400	14300	5000	0.95
18100	6800	5000	0.95
23200	10300	5000	0.95
14800	18900	5000	0.95
12100	9700	5000	0.95
41800	20800	2000	0.92
41700	19900	2000	0.97
39100	19600	2000	0.92
40200	16500	2000	0.91