TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

JARI KÖNÖNEN

# LUOTETTAVUUS AD HOC -VERKOISSA

Diplomityö

Tarkastajat: professori Pekka Loula ja yliopisto- opettaja Matti Monnonen Tarkastajat ja aihe hyväksytty TYY:n talouden ja rakentamisen tiedekunnan kokouksessa 8 kesäkuuta 2016

# TIIVISTELMÄ

**Jari Könönen**: Luotettavuus ad hoc -verkoissa
Tampereen teknillinen yliopisto
Diplomityö, 59 sivua
Kesäkuu 2016
Johtamisen ja Tietotekniikan diplomi-insinöörin tutkinto-ohjelma
Pääaine: Tietoverkkojen hallinta ja tietoturva
Tarkastajat: professori Pekka Loula ja yliopisto-opettaja Matti Monnonen

Avainsanat: Luotettavuus, Ad hoc, manet, reititys, protokolla

Ad hoc -verkko on verkko, joka rakennetaan ja muodostetaan kahden tai useamman tietoteknisen laitteen välille ilman tukiasemaa. Ad hoc -verkon laitteet kommunikoivat keskenään ilman erillistä tukiasemaa ja jos verkossa useampia laitteita ja kaikki laitteet eivät ole toistensa kuuluvuusalueella silloin laitteet toimivat myös reitittiminä toimittamaan viestit oikeille vastaanottajille. Ad hoc -verkko voi olla langallinen tai langaton verkko tai niiden yhdistelmä. Jos ad hoc -verkko on langaton, niin se tosi edullinen rakentaa, kun ei tarvita erillisiä laitteita reitittämiseen.

Luotettavuus on tärkeä asia elämässä, niin se on myös viestinnässäkin erityisesti verkkoviestinnässä. Työn tavoitteena on selvittää, miten luotettavuus toteutetaan ad hoc -verkoissa. Tutustutaan, mitä luotettavuus tarkoittaa ja miten luotettavuuden eri osa-alueet toteutetaan. Käydään läpi joitain ad hoc -verkkoratkaisuja. Mobiili ad hoc -verkko luo reitittämiseen haastetta liikkuvuuden takia. Mobiililaitteissa on usein vähän muistikapasiteettia ja laitteiden virrankulutus voi olla korkea. Työssä käsitellään reititysprotokollia varsinkin langattomaan tekniikkaan liittyen. Työ esittelee erilaisia reititysprotokollia ja miten reititysprotokollat voidaan luokitella ominaisuuksiensa tai toimintatapansa perusteella. Esimerkkeinä esitellään useita protokollia pääpiirteittäin ja lähemmin perehdytään reititysprotokollien kahteen laatupalveluun, joilla parannetaan reitityksen laatua ja sitä kautta reitityksen luotettavuutta. Reititysprotokollien tarkoituksena on toimia niin, että paketit toimitettaisiin perille vastaanottajalle lyhyintä, nopeinta ja luotettavinta reittiä pitkin. Työssä esitetään, miten reititysprotokollat hoitavat reitityksen ja siihen liittyvät asiat kuten esimerkiksi ruuhkanhallinnan, verkon kuormituksen, pakettien ja reititystietojen päivitykset. Toiset reititysprotokollat soveltuvat paremmin pieniin verkkoihin ja toiset taas vähän paremmin laajempiin verkkoihin. Ja siinäkin on isoja eroja, miten reititysprotokollat selviävät verkon topologiamuutoksista.

# ABSTRACT

**Jari Könönen**: Reliability in Ad Hoc Networks
Tampere University of Technology
Master of Science Thesis, 59 pages
June 2016
Master's Degree Programme in Management and Information Technology
Major: Network Management and Information Security
Examiner: Professor Pekka Loula and University-teacher Matti Monnonen

Keywords: Reliability, Ad Hoc, Manet, Routing, Protocol

The ad hoc network is a network that is being built and will be formed between two or more computer equipment without using an access point. The ad hoc network devices communicate with each other without an access point, and if the number of devices on the network and all of the devices are not in each other's coverage area when the equipment also work as routers to deliver messages to the correct recipients. Ad hoc network can be wired or wireless network, or a combination of the two. If the ad hoc network is wireless, so it's very cheap to build, there is no need to route the separate devices.

Reliability is an important thing in life, so it is also with regard, in particular, the network communication. The aim is to find out how reliability is carried out ad hoc networks. You will learn about what reliability means and how the various aspects of the reliability of the areas will be implemented. We will go through some of the ad-hoc network solutions. Mobile ad hoc network routing, create a challenge due to the movement. Mobile devices often have little memory capacity and power consumption of the devices may be high. The work deals with routing protocols, especially in the wireless technology. The work presents a variety of routing protocols and how routing protocol can be classified on the basis of their properties or their approach. Examples will be presented in a number of protocols and a closer look at the focus on the quality of the service in the routing protocols into two, to improve the quality and reliability of the route through the route. Routing protocols are designed to work in such a way that the packages would be put on the shortest, fastest, and most reliable delivery to the recipient. The work shows how the routing protocols manage routing and related issues such as, for example, the queue manager, network load balancing, packets and routing information updates. Some routing protocols are more suitable for small networks and others are a bit better in broader networks. And even if there are big differences in how the changes in the network topology of routing protocols will survive.

# ALKUSANAT

Tämä on tutkimustyö, jonka aiheena on luotettavuus ad hoc -verkoissa. Työ tehtiin pääsääntöisesti kevään ja syksyn 2016 välisenä aikana.

Haluan kiittää kaikkia tutkimustyöhöni jollain tavalla liittyviä tahoja. Kiitokset työntarkastajille Pekka Loulalle ja Matti Monnoselle. Erityisesti kiitos Pekka Loulalle ohjauksesta ja kärsivällisyydestä.

Porissa, 17.11.2016

Jari Könönen

# SISÄLLYSLUETTELO

# LYHENTEET JA MERKINNÄT

| | |
|---|---|
| ACK | Acknowledgement |
| ACL | Asynchronous Connection -Less |
| ADMR | Adaptive Demand-driven Multicast Routing |
| AES | Advanced Encryption Standard |
| AG | Anonymous Gossip |
| AODV | Ad hoc On-demand Distance Vector |
| AOMDV | Ad hoc On-demand Multipath Distance Vector Routing |
| AP | Access Point |
| ATIM | Announcement Traffic Indication |
| CAST | Encryption algorithm |
| CBR | Content-Based Routing |
| CEDAR | Core-extraction Distributed Ad Hoc Routing |
| CHAMP | Caching and Multipath Routing Protocol |
| DAG | directed asyclic graphs |
| dBm | Desibel-milliwatts |
| DCMP | Dynamic Core Based Multicast Routing |
| DDR | Distributed dynamic routing |
| DEAR | Device and Energy Aware Routing |
| DES | Data Encryption Standard |
| DGR | Direction Guided Routing |
| DNS | Domain Name System |
| DREAM | Distance Routing Effect Algorithm for Mobility |
| DSDV | Destination-Sequenced Distance Vector Routing Protocol |
| DSR | Dynamic Source Routing |
| DSRC | Dedicated Short-Range Communications |
| DST | Distributed Spanning Tree shuttling |
| FHSS | Frequency Hopping Spread Spectrum |
| FIFO | First In First Out |
| FSR | Fisheye State Routing |
| GAMER | Geocast Adaptive Mesh Environment for Routing |
| GFSK | Gaussian Frequensy Shift-keying |
| GPS | Generalized Processor Sharing |
| GPS | Global Positioning System |
| GREP | Gossip Reply Packet |
| GREQ | Gossip Request Packet |
| GSR | Global state routing |
| HCI | Host Controller Interface |
| HSR | Hierarchical State Routing |
| HTF | Hybrid Tree Flooding |
| HWMP | Hybrid Wireless Mesh Protocol |
| IARP | IntrAzone Routing Protocol |
| IBSS | Independent Basic Service Set |
| ID | Identifier |
| IDEA | International Data Encryption Algorithm |
| IEEE | Institute of Electrical and Electronics Engineers |
| IERP | IntErzone Routing Protocol |

| | |
|---|---|
| ISM | Industrial, Scientific, Medical |
| ITS | Intelligent Transportation System |
| L2CAP | Logical Link Control and Adaptation Protocol |
| LAR | Location-aided Routing |
| LMP | Link Manager Protocol |
| LORA | Least-Overhead Routing Approach |
| LSP | Link State Packet |
| MAC | Message Authentication Code |
| MANET | Mobile Ad Hoc Network |
| MBSS | Mesh Basic Service Set |
| MD | Message Digest |
| MEHDSR | Minimum Energy Hierarchical Dynamic Source Routing |
| MESS | Mesh Extended Service Set |
| MSTA | Mesh Station |
| mW | Milliwatt |
| NACK | Negative Acknowledgement |
| NIST | National Institute of Standards and Technology |
| NSA | National Security Agency |
| NTBMR | Neighbor Table Based Multipath Routing |
| NTP | Network Time Protocol |
| NeSt | Network Status |
| OGRP | Self-healing On-demand Geographic Path Routing Protocol |
| OLSR | Optimized Link State Protocol |
| ORA | Optimum Routing Approach |
| OSI | Open Systems Interconnection |
| PIDIS | Protocol-independent Packet Delivery Improvement Service |
| RADIUS | Remote Authentication Dial In User Service |
| RBS | Roadside Base Station |
| RDG | Route-Driven Gossip |
| RREP | Route Reply |
| RREQ | Route Request |
| RERR | Route Error |
| REEF | REliable and Efficient Forwarding |
| RFCOMM | Radio Frequency Communication |
| RSA | Rivest, Shamir, Adleman |
| RSU | Roadside Unit |
| SCO | Synchronous Connection- Oriented |
| SDP | Service Discover Protocol |
| SHA | Securer Hash Algorithm |
| SI | Swarm intelligence |
| SNMPv2 | Simple Network Management Protocol |
| SSDP | Simple Service Discovery Protocol |
| SSID | Service Set Identifier |
| STAR | Source-tree adaptive routing |
| TORA | Temporally- ordered routing algorithm |
| TTL | Time To Life |
| UDP | User Datagram Protocol |
| V2I | Vehiche to Intrastructure |
| V2V | Vehiche to Vehiche |
| VANET | Vehicular Ad Hoc Network |

| | |
|---|---|
| WMESH | Wireless Mesh Network |
| WPAN | Wireless Persona Area Network |
| WRP | Wireless Routing Protocol |
| ZHLS | Zone-based hierarchical link state |
| ZRP | Zone Routing Protocol |

# 1. JOHDANTO

## 1.1 Työn tavoite

Tämä on tutkimus ad hoc -verkkojen luotettavuudesta. Tutkimus ei ole kattava vaan pikemminkin vain pintaraapaisu laajasta ja kehittyvästä aihealueesta. Halusin tutkia, miten ad hoc -verkkoon luodaan luotettavuus sen ominaisuuksien takia. Verkkoonhan voi liittyä kuka tahansa, jos verkko on avoin ad hoc -verkko. Ad hoc -verkko voi olla myös suljettu, silloin verkkoon voi liittyä vain ennakkoon sovitut laitteet. Työssä tutkitaan, miten luotettavuus ja siihen vaikuttavat tekijät on otettu huomioon ad hoc – verkoissa ja niitä suunniteltaessa. Ad hoc -verkko on verkko, joka muodostetaan kahden tai useamman laitteen välille ilman erillistä tukiasemaa. Laitteet siis kommunikoivat keskenään ja välittävät viestejä toisilleen. Jos verkko muodostuu useammasta kuin kahdesta laitteesta, tulee verkko heti monimutkaisemmaksi ja vähän vaikeammaksi toteuttaa. Jos kaksi laitetta kommunikoi keskenään ja ne ovat toisten kuuluvuusalueella, ei ole mitään ongelmaa verkon muodostamisen suhteen. Mutta jos laitteet ovat toistensa kuuluvuusalueen ulkopuolella, silloin kommunikointi tapahtuu jonkin toisen laitteen tai laitteiden välityksellä. Tämän toisen laitteen täytyy osata toimittaa viestit kommunikoivien laitteiden välillä. Tätä toimenpidettä, että kolmas laite välittää viestit eteenpäin kommunikoivien laitteiden välillä, sanotaan reitittämiseksi. Ad hoc -verkossa laitteiden ominaisuuksiin pitää kuulua reitityksen osaaminen. Ad hoc -verkko voi olla rakennettu ihan kahdesta laitteesta tai määräämättömästä määrästä laitteita. Verkko voi olla langallinen, jolloin laitteet yhdistetään toisiinsa kaapelin avulla. Usein verkosta osa voi olla langatonta tekniikkaa ja tänä päivänä koko verkkokin voi olla täysin langatonta tekniikkaa käyttävä ja usein onkin. Työn tavoitteena on saada käsitys, miten luotettavuus ja siihen liittyvät tekijät saadaan toimimaan varsinkin manet-verkossa (mobile ad hoc network) sen liikkuvuuden ja muiden luotettavuutta haittaavien tekijöiden takia. Työn tavoitteena on saada kuva siitä, miten eri reititysprotokollien ominaisuudet vaikuttavat viestien välittämiseen ja siihen, miten toiset reititysprotokollat voivat toimia paremmin riippuen verkon koosta tai protokollien tavasta toimia viestejä eteenpäin lähetettäessä. Erityisesti, miten joku reititysprotokolla voi toimia paremmin jollain tietyllä tavalla tai alueella. Miten eri ominaisuudet tekevät toisista protokollista parempia kuin toisista. Toiset vaan soveltuvat toisia paremmin tiettyihin olosuhteisiin riippuen vaatimuksista. Luotettavuus koostuu monien eri tekijöiden summasta, johon vaikuttavat niin sisäiset kuin ulkoiset tekijät.

## 1.2 Työn sisältö

Luvussa 2 käsitellään luotettavuutta ja kaikkea mitä luotettavuus pitää sisällään. Luotettavuus on tärkeä osa kommunikointia ihmisten välillä riippumatta siitä, tapahtuuko se kasvokkain vai verkon välityksellä. Verkon luotettavuus on erityisesti tutkinnan alla. Käydään läpi, miten luotettavuus hoidetaan verkkomaailmassa. Luvussa kerrotaan, mistä luotettavuus koostuu ja miten ne toteutetaan parhaalla mahdollisella tavalla verkon resursseilla.

Ad hoc -verkosta on paljon erilaisia variaatioita. Tutustun ja tutustutan lukijat tässä tutkimuksen luvussa 3 mesh-, manet-, vanet- ja bluetooth-verkkoihin. Muut ad hoc -verkon variaatiot esimerkiksi sensoriverkot ja Zigbee jätettiin suosiolla pois tästä tutkimuksesta. Mesh-verkko on verkko, jossa kaikki verkon laitteet kommunikoivat kaikkien laitteiden kanssa ja ilman erillistä tukiasemaa ad hoc -verkkojen tapaan. Jos mesh-verkko on langallinen ja täydellinen toteutukseltaan, niin silloin jokaisesta laitteesta lähtee kaapeli jokaiseen verkon laitteeseen. Tämä on kallis ja hankala toteuttaa käytännössä. Usein mesh-verkko onkin langaton tai ainakin osittain langaton ja vain toteutettu yhteyksillä joita tarvitaan. Manet-verkko käydään paremmin läpi protokollien välityksellä. Vanet-verkko on ajoneuvoihin suunnattu langaton verkko. Vanet tulee sanoista Vehicular Ad Hoc Network ja se on vain ajoneuvoihin liittyvä verkkoratkaisu. Bluetooth on taas lyhyiden etäisyyksien verkko. Verkkoihin liittyy aina tietoturvaongelmia tai paremminkin tiedon turvaamiseen liittyviä asioita. Joitakin verkkoihin liittyviä suoranaisia uhkia lähinnä pintapuolisesti käsitellään luvun loppupuolella.

Myöhemmin luvussa 4 käydään läpi reititys ja reititysprotokollat tai oikeastaan vain osa niistä, koska reititysprotokollia on kymmeniä. Olen valinnut tähän mielestäni muutaman tärkeimmän ja ehkä eniten käytetyn. Erityisesti keskitytään manet –verkon reititykseen. Luotettavalle tiedonsiirrolle tulee haastetta manet -verkossa, koska lähettävä laite ja vastaanottava laite sekä reitityksen käytetyt laitteet ovat usein liikkuvia. Liikkuvuudesta huolimatta viestit pitää saada luotettavasti ja mahdollisimman nopeasti perille. Luvun lopussa esitellään myös laatupalveluun perustuvia reititysprotokolliin saatavia palveluita, joilla parannetaan reitityksen laatua ja luotettavuutta pakettien läpimenoon koko verkon alueella. Näistä palveluista tai mekanismeista olen valinnut kaksi, jotka käydään tarkemmin läpi.

.

# 2. LUOTETTAVUUS

Luotettavuus on käsitteenä siinä mielessä hankala, ettei sille ole virallista määritelmää. Jokaisella ihmisellä on oma käsitys, mitä luotettavuudella tarkoitetaan. Luotettavuudella voidaan tarkoittaa monia eri asioita. Voidaan puhua, että joku ihminen on luotettava, jokin laite on luotettava, tieto on luotettava ja se on peräisin luotettavalta taholta. Luotettava henkilö on rehellinen eikä huijaa tai valehtele muille ihmisille. Jos ihmiseen voi luottaa, hän on luotettava. Sama koskee tekniikkaa, tekniikka on luotettava, kun se toimii halutulla tavalla. Tekniikalla on ominaisuus, joka on tekniikan pettäminen. Sama ominaisuus on ihmisilläkin. Ihmiset voivat olla epärehellisiä, voivat sairastua, voivat vihastua ja käyttäytyä poikkeavasti normaalista käytöksestä. Ihmiset ovat inhimillisiä, laitteet eivät niinkään. Luotettavuus täytyisi määritellä jotenkin, että tiedettäisiin, milloin laitteen tai ihmisen toiminta on normaalia tai oikeammin halutun kaltaista. Kun tietotekniikassa puhutaan luotettavuudesta, silloin usein tarkoitetaan, että laitteet ovat luotettavia, verkko toimii halutulla tavalla, tieto on luotettavaa ja ihmiset ovat luotettavia. Kaikki toimivat silloin oikein ja oikealla tavalla. Luotettavuus voitaisi sen perusteella määritellä, että se on oikeanlainen. Se on määritelmänä aivan liian epämääräinen. Voisi ajatella, että luotaisiin säännöt siitä, milloin joku asia on luotettava. Tarkkailtaisiin luotettavuutta ja kehitettäisiin metriikkoja siitä, miten sitä mitattaisiin. Monella tavallahan verkon ominaisuuksia jo mitataan. Tunnettuja metriikkoja ovat esimerkiksi läpimeno-aika-mittarit, viive per solmu, kadonneet paketit ja ruuhkista toipumisajat.

Tässä tutkimustyössä käsitellään ad hoc -verkon luotettavuutta. Kaikki asiat, mitä seuraavissa aliluvuissa käsitellään ja käydään läpi, ovat asioita ja käsitteitä, jotka omalla tavallaan lisäävät ad hoc -verkon luotettavuutta. Lisäävät ne muitakin tärkeitä ominaisuuksia verkkoon ja verkon liikenteeseen. Tuovat ihan perusasiat verkkoliikenteeseen, jotka pitäisi olla itsestään selvyyksiä verkkoliikenteessä. Ad hoc -verkko voi olla lankaverkko, mutta myös langaton, mikä tänä päivänä se useimmiten onkin tai ainakin osittain. Langattomuus tuo haasteita myös verkon luotettavuuteen, eikä ainoastaan liikkuvien laitteiden ja koko ajan muuttuvan verkon rakenteen takia. Reititys ja mahdollisesti jatkuva solmujen vaihtuminen luovat reititysprotokollille erityisiä haasteita. Se myös on luotettavuutta, että paketit silti toimitetaan perille vaikeista olosuhteista huolimatta. Ympäristötekijät voivat vaikuttaa pakettien siirtoon ja perillepääsyyn. Tällaisia ovat esimerkiksi maasto, ilmanpaine, katvealueet ja kaupungissa korkeat rakennukset. Luotettavuutta on se, että häiriötekijöistä huolimatta viestit saadaan toimitettua oikeille vastaanottajille. Luotettavuuden mittarina voisi toimia se, miten reititysprotokolla selviää verkon yllättävistä tilanteista toimittamalla paketit perille ilman suurempia viiveitä ja selviämällä linkkikatkoksista tai solmujen katoamistilanteista löytämällä uusia reittejä kohtuullisilla aikaviiveillä. Solmujen katoamistapauksissa ja linkkikatkostapauksissa monipolkureititysprotokollat ovat vahvoilla, koska niiden ei tarvitse ryhtyä etsimään uusia polkuja pakettien perille toimittamiseksi, vaan ovat etsineet ne jo valmiiksi. Säästyy paljon aikaa ja verkkoresursseja, kun vaihtoehtoiset reitit ovat valmiina reititysmuistissa

reititystaulussa. Sama koskee tietysti ruuhkatapauksia myös. Siksi monipolkureititysprotokollia pidetään luotettavimpina reititysprotokollina. Myös reittien huoltaminen ja reittien olemassa pitäminen on luotettavuuden mittareita. Luotettavuus tulisi määritellä, miten hyvin viestit menevät perille mahdollisimman nopeasti mahdollisimman lyhyintä reittiä pienimmillä mahdollisilla kustannuksilla.

## 2.1 Eheys

Eheys tarkoittaa, että tieto on oikeellista ja kun tietoa käsitellään tai säilytetään, niin tieto pysyy muuttumattomana ja virheettömänä. Kun tietoa tuotetaan, pyritään tuottamaan paikkansa pitävää tietoa, joka ei sisällä tahallista tai tahatonta virheellistä tietoa. Eheys saavutetaan yleensä ohjelmointiteknisillä keinoilla ja ratkaisuilla. (Hakala, 2006, p. 5). Eheys tarkoittaa myös, ettei mikään ulkopuolinen tekijä pääse muuttamaan tai poistamaan tietoa tai tiedon sisältöä. Virukset tai haittaohjelmat voivat rikkoa eheyttä tarttuessaan tietoon laitteesta tai siirron aikana. (Järvinen, 2002, p. 23). Eheyttä voidaan varmistaa käyttämällä laitteita ja protokollia, joissa on virheen korjaus- ja tunnistusmekanismeja. Myös salausmenetelmät ja- laitteet edesauttavat eheyden säilymistä. Varmuuskopioinnilla ja käyttöoikeuksien rajoituksilla voidaan myöskin varmistaa eheyden säilyminen. Eheys varmistetaan myös tarkistussummilla, tarkastuskoodeilla ja digitaalisella allekirjoituksella. (Hakala, 2006, p. 5).

Mistä voi tietää, onko lähettäjän lähettämä viesti saapunut vastaanottajalle muuttumattomana vai onko viestistä kadonnut matkalla osia tai kokonaisuuksia. Miten voidaan varmistaa, ettei viesti muutu tahallisesti jonkun toisen välityksellä tai voi viesti muuttua vahingossakin. Myös verkon rakenteen, ruuhkan tai jonkin inhimillisen syyn takia viesti voi olla muuttunut matkalla joko tahallisesti tai vahingossa saapuessaan vastaanottajalle. Viestin eheys varmistetaan tekemällä viestistä tiivistelmä, joka lähetään lähettäjän viestin mukana vastaanottajalle. Vastaanottaja vertaa viestistä tehtyä tiivistettä viestin sisältöön. Jos viesti ja viestistä tehty tiiviste täsmäävät, niin viesti on saapunut muuttumattomana vastaanottajalle. Jos viesti ja viestistä tehty tiiviste eivät täsmää jostain syystä, niin siinä tapauksessa viesti tai viestistä tehty tiiviste on muuttunut matkalla lähettäjältä vastaanottajalle ja näin viesti ei ole alkuperäinen viesti vaan sen eheys on muuttunut. Tällaisessa tapauksessa viesti pitää lähettää uudelleen olettaen, että viesti on muuttunut vahingossa tai paketteja on kadonnut matkalla lähettäjältä vastaanottajalle. Tiivisteiden avulla siis varmistetaan viestin sisällön alkuperäisyys ja muuttumattomuus eli eheys. Tiiviste viestiin tehdään ja puretaan tiivistefunktion (Hash function) avulla. Se on yksisuuntainen funktio, jolla tehdään viestistä tiiviste. Tiivistefunktio tekee viestistä tiivisteen, joka eräänlainen bittijono, tätä bittijonoa vastaanottaja vertaa saatuun viestiin. Tunnettuja yleisesti käytettyjä tiivistefunktioita MD5 (Message-Digest algoritm), SHA-1 (Secure Hash algoritm 1) ja SHA-2 (Secure Hash algoritm 2). (Kaarnalehto, 2011)

MD5 on yksi Ron Rivestin kehittämistä tiivistealgoritmeista vuonna 1991 ja se perustuu edeltävään MD4 tiivistealgoritmiin. MD5 algoritmi tuottaa 128 -bitin eli 16 tavun tiivisteen.

SHA-1 on kryptograafinen tiivistefunktio ja sen tekemä tiiviste on 160 bittiä pitkä. Se on kehitetty NSA (National Security Agency) toimesta ja NIST (National Institute of Standards and Technology) standardoima vuonna 1995. Sitä käytetään muiden tiivistealgoritmien tapaan eheyden varmistamisen lisäksi digitaalisissa allekirjoituksissa, todennuksessa, avainten vaihtoprosesseissa ja satunnaislukugeneraattoreissa. Sitä käytetään myös ohjelmakooditarkastuksissa ja yleensäkin tapahtumissa, jotka vaativat muuttumatonta tietoa tai ohjelmistotarkistusta. (Wan, et al., 2005)

SHA-2 on seuraava kehitys SHA-1stä. SHA-1 jälkeen on tehty neljä versiota SHA-224, SHA-256, SHA-384 ja SHA-512. Näistä käytetään yhteisnimitystä SHA-2. Nimiensä mukana oleva luku kertoo tiivistealgoritmin pituuden bitteinä. SHA-512 tuottaa siis 512 bitin pituisen tiivisteen eli 64 tavuisen. SHA-2 on standardoitu 2002 (NIST) ja silloin se syrjäytti edeltäjänsä. Käyttötarkoitus on sama kuin muissakin tiivistealgoritmeissa. (Sklavos & Koufopavlou, 2003), (McEvoy, et al., 2006)

## 2.2 Tietoturvallisuus

Tietoturvallisuus määritellään usein kolmella määritteellä, jotka ovat eheys, joka käsiteltiin jo edellä, luottamuksellisuus ja saatavuus. Luottamuksellisuudella tarkoitetaan sitä, että tietoa voi ja saa hyödyntää vain henkilö tai henkilöt, joilla on oikeus hyödyntää kyseistä tietoa. (Kangas, 2003). Luottamuksellisuutta voidaan parantaa salaustekniikoilla ja pääsynhallinnalla. Saatavuus taas tarkoittaa, että laite, järjestelmä, ohjelma tai palvelu on käytettävissä silloin kun käyttäjä tarvitsee sitä. Näitä kaikkia edellä mainittuja voi häiritä jokin virhe tai virheellinen toiminta, silloin saatavuutta heikentää ulkopuolinen tekijä tai ominaisuus. Jos saatavuudella on ominaisuus 24/7, tarkoittaa se sitä, että saatavuus on käytettävissä 24 tuntia vuorokaudessa ja 7 päivää viikossa. Saatavuuteen yhdistetään usein myös luotettavuus, ylläpidettävyys, huoltovarmuus, suorituskyky ja turvallisuus. Saatavuus liitetään usein tietotekniikassa siihen, miten joku järjestelmä, sovellus tai palvelu on käytettävissä. Saatavuutta haittaa, jos niiden käytettävyydessä on tahallinen tai tahaton katkos. Saatavuutta parantaa, jos suunnittelemattomiin saatavuuskatkoihin on varauduttu jollakin varajärjestelmällä tai pyritään toimimaan ja varautumaan niin, että mahdolliset odottamattomat saatavuuskatkokset jäisivät kestoltaan mahdollisimman lyhyiksi. Tietoturvallisuutta voidaan arvioida ja käsitellä monesta eri näkökulmasta; esimerkiksi laitteiden, käyttäjän, sovellusten, tietojen käsittelyn ja tietoliikenteen kannalta. Listaa voisi jatkaa, vaikka kuinka pitkälle. Uhkana tietoturvallisuudelle on erilaiset huijausyritykset, roskapostit, yritysvakoilu, piratismi, tietokonevirukset ja yksityisyyteen liittyvät loukkaukset. Tietoturvauhkiin kuuluu taas vähän toisenlaiset

asiat kuten tiedon katoaminen tai muuttuminen, tiedon kopioituminen, salaisen tiedon julkiseksi tai tutkituksi tuleminen, tiedon luvaton käyttö tai luvaton pääsy tietoon. (Lehtonen, 2004), (Hakala, 2006).

## 2.3 Todentaminen

Arkielämässä todentaminen tapahtuu yleensä silloin, kun joku pyytää esittämään henkilötodistuksen, jolla varmistetaan henkilön aitous todentamistilanteessa. Henkilö todennetaan yleisesti viranomaisen myöntämällä henkilötodistuksella kuten ajokortti tai passi tai jokin muu viranomaisen myöntämä virallinen henkilötunnistusasiakirja. Todentaminen on aidon erottamista epäaidosta ja väärennetyn erottamista väärentämättömästä. Käyttäjän todentaminen on tiedon jakamisen luottamuksellisuutta. Tietotekniikassa todentaminen tapahtuu eri tavalla kuin arkimaailmassa. Todentamisella tarkoitetaan toimenpidettä, jolla kirjaudutaan palveluun, johon kirjautujalla on oikeus ja mahdollisuus päästä. Todennus tapahtuu yleensä oikealla tunnuksen ja salasanan yhdistelmällä. Paremmin varmennetuissa järjestelmissä voidaan käyttää useaa yhtäaikaista menetelmää, kuten sormenjäljen tunnistusta, sähköistä avainkorttia ja muutettavaa koodia, jopa iiristunnistusta jossakin käytetään. Silmähän on yksilöllinen jokaisella ihmisellä kuten sormenjälkikin. Kun käyttäjä on tunnistettu, voidaan hänelle luovuttaa sovitut käyttöoikeudet. Hyvä esimerkki tästä on jokaiselle tuttu nettipankissa laskun maksaminen verkon yli. Tunnistus hoidetaan käyttäjätunnuksella ja henkilökohtaisella salasanalla, lisäksi pankin ulkopuolisella varmentajalla eli avainlukulistalla varmistetaan, että käyttäjä on henkilö, jolla on oikeudet pankkitilin käyttöön. Todentamiseen liittyy läheisesti pääsynvalvonta. Pääsynvalvonta tarkoittaa, että valvotaan että tietotekniikassa tiedostoihin, ohjelmistoihin, teknisiin tiloihin ja oikeastaan mihin tahansa, joihin on rajoitettu pääsy vain luvallisilla henkilöillä. Pääsy evätään ulkopuolisilta, joilla ei ole luvallista pääsyä tiettyyn palveluun tai muuhun, mikä vaatii käyttöoikeudet luvallisilta käyttäjiltä. Käyttäjä varmistaa käyttöoikeudet vaativaan toimenpiteeseen asiaan kuuluvalla ja sovitulla tavalla. Tapa riippuu kohteesta ja voi olla sähköinen avain, käyttäjätunnus/salasana-yhdistelmä tai mikä vain tässä edellä mainituista. Näin henkilö todennetaan ja tunnistetaan, minkä jälkeen sallitaan pääsy järjestelmään. Tällä valvotaan, että ulkopuoliset henkilöt eivät saa palvelua tai muuta vastaavaa, koska kaikki eivät voi ja saa käyttää luottamuksellisia toimintoja ja sillä suojataan myös väärinkäytökset. Yrityksissä voi olla paljon tietoa, joka ei saa joutua ulkopuolisten saataville tai luettaviksi. Usein tiedostoihin, joihin luvallisilla käyttäjillä on pääsy todennuksen ja tunnistuksen jälkeen, on järjestelmä, joka ylläpitää seurantaa tiedostoihin tehtävistä muutoksista ja muutoksen tekijöistä. Pääsynvalvontaan liittyy tosi usein jonkinlainen seurantajärjestelmä tai lokikirja. Valvontajärjestelmissä oletetaan, että todennettu henkilö on henkilö, joka omistaa valtuudet toimia kyseisessä palvelussa tai toiminnassa. Väärinkäytösten yhteydessä todennuksen takia toteutuu myös kiistämättömyys. Henkilö ei voi kiistää olleensa paikalla, todennuksen ja tunnistuksen takia. Tietotekniikassa viestin lähettäjä voidaan todentaa

sähköisellä allekirjoituksella. Sähköisellä allekirjoituksella salataan viestin lähettäjä. Silloin puhutaan digitaalisesta allekirjoituksesta. Henkilötodennuksen voi hoitaa myös luotettavaksi todettu ulkopuolinen todentaja samalla tavalla kuin pankkimaksamisessa, esimerkiksi RADIUS-palvelin (Remote Authentication Dial In User Service) voi toimia ulkopuolisena todentajana. Ulkopuolisena todentajana voi toimia oikeastaan kuka vaan, sen takia siihen liittyy turvallisuusriski. Todennukseen kannattakin käyttää tunnettuja sertifioituja todentajia. Salausavaintenvaihtoa ja tunnettuja salausalgoritmeja esimerkiksi MD5-algoritmiä käytetään viestin salaamiseen. Salausavainten käytöllä todennetaan viesti oikeaksi. Todennukseen kuuluu myös, että tieto on todenmukaista. Tarkoittaa, että tieto on oikeasta lähteestä ja alkuperäinen muuttumaton tieto. Lisäksi henkilön pitää olla oikea todennettu henkilö käsittelemään kyseistä tietoa. (Järvinen, 2002), (Hakala, 2006), (Stallings, 2002, pp. 395-396), (Comer, 2009, pp. 515-521)

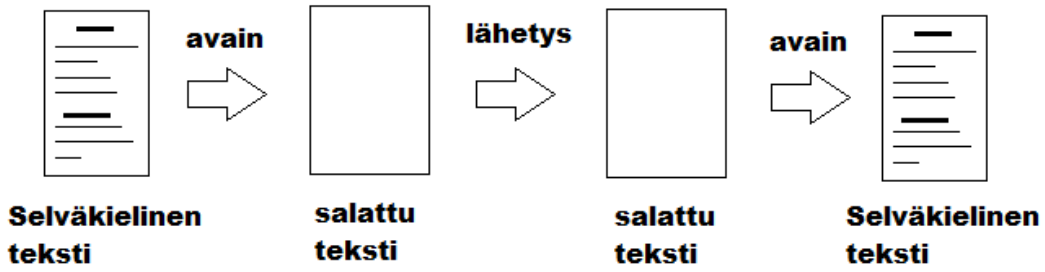
## 2.4 Salaus

Salaaminen on hyvä keino edesauttaa eheyden säilymistä. On olemassa erilaisia salaustekniikoita ja algoritmeja, joilla voidaan salata viesti tai lähetettävä tieto, jotka vain valtuutetut osapuolet voivat lukea. Salaustekniikka muuttaa tiedon tai viestin selkokielisen tekstin salatuksi tekstiksi käyttäen salausalgoritmia. Tieto muutetaan vastaanottajalla salatusta tekstistä takaisin selkokieliseksi tekstiksi käyttäen salauksenpurkualgoritmia. Salatun viestin voi kaapata, mutta sen lukeminen vaatii salatun koodin purkamisen. Viestin purkaminen on aikaa vievä toimenpide, jos purkuavainta tai algoritmia ei ole. Monissa tapauksissa tieto saattaa olla jo vanhentunutta ennen kuin se saadaan purettua. Salaustekniikoita on kaksi, jotka perustuvat avainten vaihtoon. Nämä ovat asymmetrinen(epäsymmetrinen) ja symmetrinen salaus.

### 2.4.1 Symmetrinen salaus

Symmetrinen salaus salataan ja puretaan samalla avaimella. Symmetrisestä salauksesta käytetään myös nimeä salaisen avaimen menetelmä. Osapuolet ovat etukäteen sopineet avaimien vaihdosta ja toimitustavasta. Ongelmana on avainten vaihdon hallinta siten, että avaimet eivät joudu ulkopuolisten haltuun. Jos avain tulee ulkopuolisen haltuun, hän pystyy vapaasti purkamaan ja lukemaan viestin. Tällaisessa tapauksessa osapuolten täytyy vaihtaa avain uuteen. Avainten vaihtaminen netin välityksellä ei ole turvallisin vaihtoehto. Avainten vaihtotapa pitäisi olla luotettava ja luotettavan siirtokanavan rakentaminen on hankalaa ja kallista. Symmetrisen salauksen periaate selviää kuvasta (kuva 1). Symmetrinen salaus on nopeaa ja helppokäyttöistä ja siksi se on suosittu.

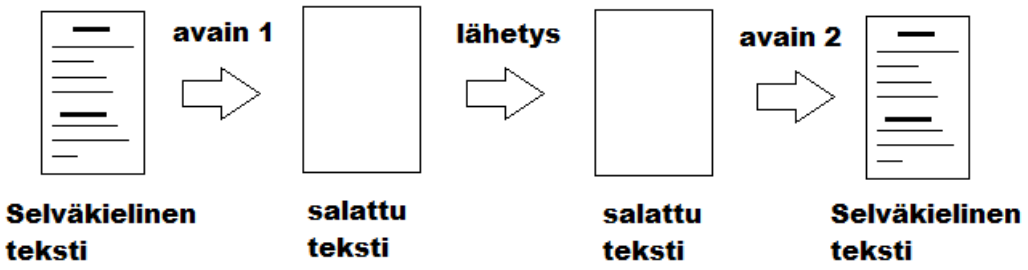
Symmetriset salausalgoritmit jaotellaan jono -ja lohkosalauksiin. Tunnettuja symmetrisiä salausjärjestelmiä ovat DES, 3DES, AES, IDEA, Blowfish ja CAST. (Lehtonen, 2004), (Käpylä, 2000).




**Kuva 1**. Symmetrinen salaus.

### 2.4.2 Epäsymmetrinen(asymmetrinen) salaus

Epäsymmetrinen salaus käyttää avainparia. Lähettäjä salaa viestin vastaanottajan julkisella avaimella ja vastaanottaja avaa viestin omalla yksityisellä avaimella. Epäsymmetristä salausta sanotaan siitä syystä julkisen avaimen menetelmäksi. Epäsymmetrinen salaus perustuu siis avainpariin, jossa toista avainta käytetään viestin salaamiseen ja toista avainta viestin sisällön purkamiseen. Julkinen ja yksityinen avain ovat toistensa vastakappaleita. Ne ovat yhteydessä toisiinsa matemaattisella tavalla ja kaavalla. Yksityistä avainta on mahdoton selvittää salaisen avaimen avulla. Julkisen avaimen voi jakaa julkisesti ja vapaasti eikä siihen liity tietoturvariskiä, koska sillä voi vain salata viestin eikä purkaminen onnistu ilman yksityistä avainta. Epäsymmetristä salausta selventää kuva (kuva 2). Epäsymmetristä salausta voidaan käyttää esim. todennukseen, digitaalisiin allekirjoituksiin ja avaintenhallintaan. Epäsymmetrinen salaus on hidas, koska se vaatii turvallisuuden varmistamiseksi pitkiä avaimia. Tunnetuimmat epäsymmetriset algoritmit ovat RSA ja Diffie-Hellman. (Salonen, 2009), (Comer, 2009, pp. 517-520)




**Kuva 2**. Epäsymmetrinen salaus. Avain 1 on julkinen avain ja avain 2 on yksityinen avain.

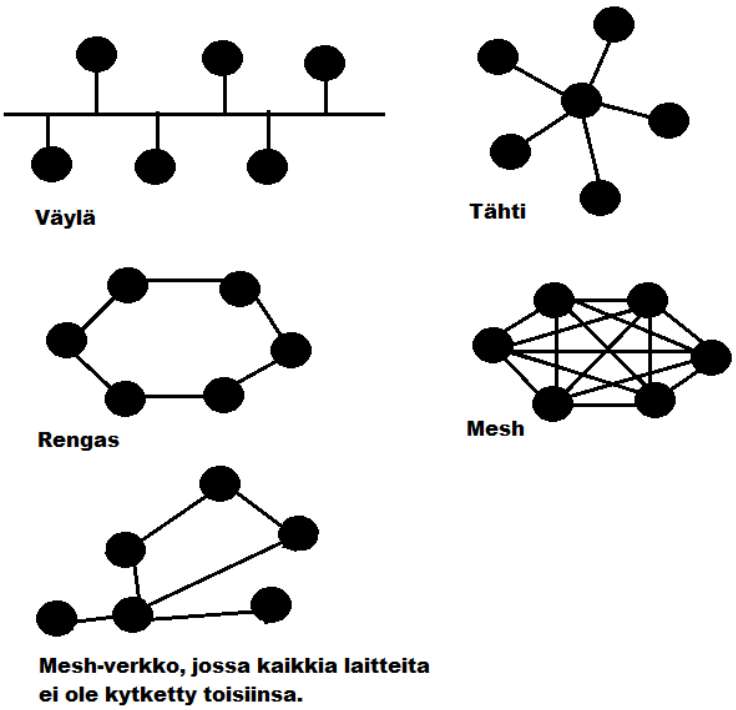
# 3. AD HOC -VERKKO

Ad hoc tulee latinankielisistä sanoista ”tähän tarkoitukseen”. Se kuvastaa hyvin ad hoc -verkoille tyypillistä tapaa, jolla verkko luodaan. Usein verkko luodaan spontaanisti ja siihen voi liittyä useita laitteita, jotka kommunikoivat keskenään ja verkko muodostuu automaattisesti. Ad hoc -verkko voidaan rakentaa nopeasti ja edullisesti, koska laitteet keskustelevat keskenään, ei tarvita reitittimiä tai muita reititykseen tarvittavia laitteita. Laitteiden pitää olla toistensa kuuluvuusalueella, jotta ne voivat kommunikoida keskenään. Koska ad hoc -verkko on usein langaton verkko, joka käyttää tiedonsiirtoon ilmatietä, on tieto muutettava siirrettävään muotoon radiotaajuuksille eli moduloitava. Langaton tietoliikenneyhteys on vaihtoehto langalliselle yhteydelle. Langaton yhteys käyttää tietoliikennekaapelin sijaan radiotietä eli radiotaajuuksia. Langaton tekniikka otti suuria askeleita eteenpäin 1990-luvulla. Langatonta verkkotekniikkaa on periaatteessa kahdenlaista; Infrastruktuuriverkko ja ad hoc -verkko. Intrastruktuuriverkossa langattoman verkon kaikki laitteet ovat yhteydessä toisiinsa langattoman reitittimen välityksellä. Ad hoc -verkossa verkon laitteet ovat yhteydessä toisiinsa suoraan ilman reititintä tai tukiasemaa. Myös ad hoc -verkon laitteilla on oltava sama IP-osoite, jolla se on liittynyt ad hoc -verkkoon sekä sama kanava ja SSID kuin muilla verkon laitteilla. Tässä tutkimuksessa keskitytään pääasiassa langattomiin ad hoc -verkkoihin. Kun käsitellään verkkoja, joissa solmuja on enemmän kuin yksi, pitää viestit reitittää vastaanottajalle. Langattomissa ad hoc -verkoissa laitteiden pitää pystyä hoitamaan reititys. Ad hoc -verkko on verkko, joka toteutetaan kahden tai useamman laitteen välille ilman erillistä tukiasemaa. Ensimmäisiä ad hoc -verkon sovelluksia oli toimistoympäristössä langallisten laitteiden korvaaminen langattomilla laitteilla. Aika usein esimerkiksi tulostin liitettiin langattomasti toimiston tietoverkkoon. Hallitseminen oli aika helppoa, koska etäisyydet olivat lyhyet ja laitteilla verkossa suhteellisen vähäinen määrä käyttäjiä. Ad hoc -verkosta on monta erilaista sovellusta. Ad hoc – voi olla langallinen tai langaton. Se voi olla myös osittain langaton ja osittain langallinen. Jos ad hoc -verkkoon kuuluva yksikin laite on yhteydessä internettiin, verkosta tulee tällöin maailmanlaajuinen. Seuraavaksi alaluvuissa esitellään tyypillisiä ad hoc -verkkoja. (Gerla, 2005).

## 3.1 MESH

Mesh- verkko on langallinen verkko, jossa jokainen laite on yhteydessä verkon kaikkiin laiteisiin suoraan ilman erillistä tukiasemaa. Kaikki laitteet mesh-verkossa tekevät yhteistyötä keskenään. Riippuu verkon topologiasta, miten laitteet on liitetty toisiinsa. Päätopologioita on neljä. Väylä, tähti, rengas ja mesh- topologiat ovat päätopologioita ja

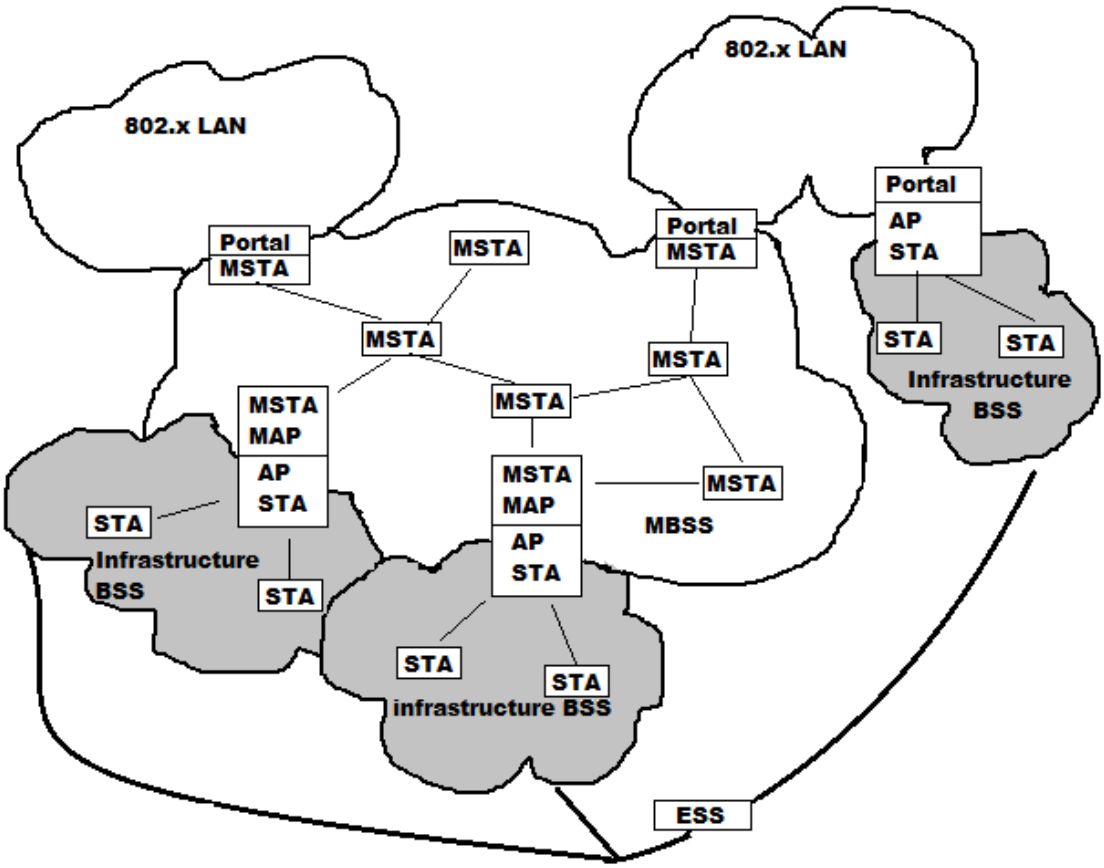
niistä kehitetty erilaisia muunnelmia, joita kutsutaan hybriditopologioiksi. Väylätopologia tarkoittaa, että siinä kaikki laitteet on kytketty toisiinsa peräkkäin liittimillä yhteen kaapeliin, jonka molemmissa päissä on terminaattoriksi kutsuttu vastus. Tämä ratkaisu on vikaherkkä ja perustuu kilpavarausmenetelmään. Kilpavarausmenetelmä tarkoittaa, että käytännössä verkkoa voi käyttää vain yksi laite kerrallaan. Jos verkkoon tulee useampi laite, liikennöinti aiheuttaa törmäyksen ja näin ruuhkauttaa verkon. Lisäksi jos laitteita yhdistävä kaapeli rikkoontuu, katkeaa yhteys kaikkiin verkon laitteisiin. Väylätopologia on vanha ratkaisu eikä juurikaan enää käytössä. Tähtitopologiassa kaikki laitteet on kytketty keskuslaitteeseen omalla kaapelilla. Keskuslaite voi olla keskitin tai kytkin. Rengastopologia on väylätopologian kaltainen, mutta on kytketty ympyrän muotoon. Mesh-topologiassa kaikki laitteet on kiinnitetty toisiinsa kaapeleilla. Jos mesh-verkko on täydellinen, se tarkoittaa, että kaikista laitteista lähtee kaapeli jokaiseen laitteeseen. Tämä on kallis ja monesti hankalakin toteuttaa kaapelien ja liittimien suuren määrän takia varsinkin, jos verkossa on paljon laitteita. Mesh-verkko onkin yleensä, jos on kokonaan langallinen verkko, niin toteutettu vain osittain hinnan ja kaapelointien takia. Lisäksi osittain toteutettu mesh-verkko on vikasietoisempi ja silti viestit voidaan toimittaa jotakin reittiä pitkin vastaanottajalle. Kuva 3 näyttää pääverkkotopologiat. (Comer, 2009)




**Kuva 3.** Verkkotopologiat.

Mesh-verkko on toteutettu nykyään joko kokonaan langattomana tai osittain. Jos mesh-verkko on kokonaan langaton, puhutaan silloin langattomasta mesh-verkosta (Wireless Mesh Network, WMN). Langaton mesh-verkko on alun perin kehitetty armeijakäyttöön ja sieltä levinnyt yleiseen käyttöön, niin kuin monet muutkin tietotekniikkaan liittyvät ratkaisut. Langattomassa mesh-verkossa ei ole selvää topologiaa langallisen verkon tapaan, johtuen laitteiden langattomuudesta ja laitteiden liikkuvuuden johdosta. Mesh-verkossa jokaisen solmun on omien viestien vastaanottamisen lisäksi kyettävä lähettämään viestit eteenpäin ja reitittämään muiden solmujen viestejä ja toimittamaan niitä kohden vastaanottajaa. Langattomuus tuo myös haastetta reititykseen, koska usein laitteet ovat mobiililaitteita, jotka ovat liikkeessä. Jos laite poistuu verkosta tai verkon kuuluvuusalueelta silloin mesh -verkot reitittyvät automaattisesti. Mesh-verkko kykenee itsenäisesti vaihtamaan tukiasemiaan ja pystyy näin korjaantumaan, kun verkkoon tulee muutoksia.

Langaton mesh-verkko käyttää IEEE 802.11s, IEEE 802.15 tai IEEE 802.16 standardia. Mesh-verkko rakentuu mesh-asemista (Mesh Station, MSTA), jotka kommunikoivat keskenään. Mesh-verkossa voi olla kaksi tai useampia mesh-asemia. Tätä verkkoa kutsutaan mesh liityntäpisteeksi (Mesh Basic Service Set, MBSS). Mesh-verkko voidaan yhdistää toisiin verkkoihin mesh-tukiaseman välityksellä (Mesh Access Point, MAP) ja portaalin (Portal) kautta. Mesh-asema voi toimia myös tukiasemana tai portaalina oman toimensa ohella. Mesh tukiasema tarjoaa kommunikoinnin mesh-verkkoon kuulumattoman laitteen kanssa. Portaalin kautta mesh-aseman on mahdollista saada yhteys internettiin. Useamman mesh-verkon kokonaisuutta kutsutaan mesh jatkettu liityntäpiste (Mesh Extended Service Set, MESS). Kuva 4 selventää mesh-verkon rakennetta. (Akyildiz, et al., 2004) (Islam, et al., 2009)




**Kuva 4**. Langattoman mesh-verkon rakenne ja kytkennät muihin verkkoihin.

Mesh-asema liittyy verkkoon lähettämällä säännöllisin väliajoin beacon-viestiä verkon muille laitteille, jolla se ilmoittaa halunsa liittyä verkkoon. Beacon-viesteillä pidetään verkkoa yllä ja saadaan tietoa, onko laite poistunut verkosta tai onko uusia asemia liittynyt verkkoon. Mesh-verkossa on yleisesti käytössä manet- verkosta tunnettuja protokollia, joita esitellään luvussa 4. Yksi vain mesh-verkkoon tarkoitettu protokolla on Hybrid Wireless Mesh Protocol (HWMP). Se on mesh-verkkoon suunniteltu reitinvalintaprotokolla. HWMP käyttää reititykseen MAC-osoitteita poiketen muista protokollista, esimerkiksi Ad Hoc On Demand distance Vector (AODV) käyttää IP-osoitteita. Koska se on risteymäreititysprotokolla (Hybrid Routing Protocol) niin siinä on ominaisuuksia sekä ennakoivista reititysprotokollista (Proactive Routing Protocol) että vastavaikutteisista reititysprotokollista (Reactive Routing Protocol). HWMP reitittää paketit kahdella tavalla vastavuorottaisella eli tarvittaessa ja ennakoivalla puurakenne moodilla. Reititystavat käydään läpi luvussa 4. Mesh-verkossa on mahdollista salata viestit, niin kuin monissa muissakin verkoissa. Avaintenvaihto ja todennus onnistuvat mesh-verkossa. Jos todennusta vaaditaan verkkoon, se tapahtuu, kun uusi laite liittyy verkkoon. (Virkkala, 2009), (Jun & Sichitiu, 2008)

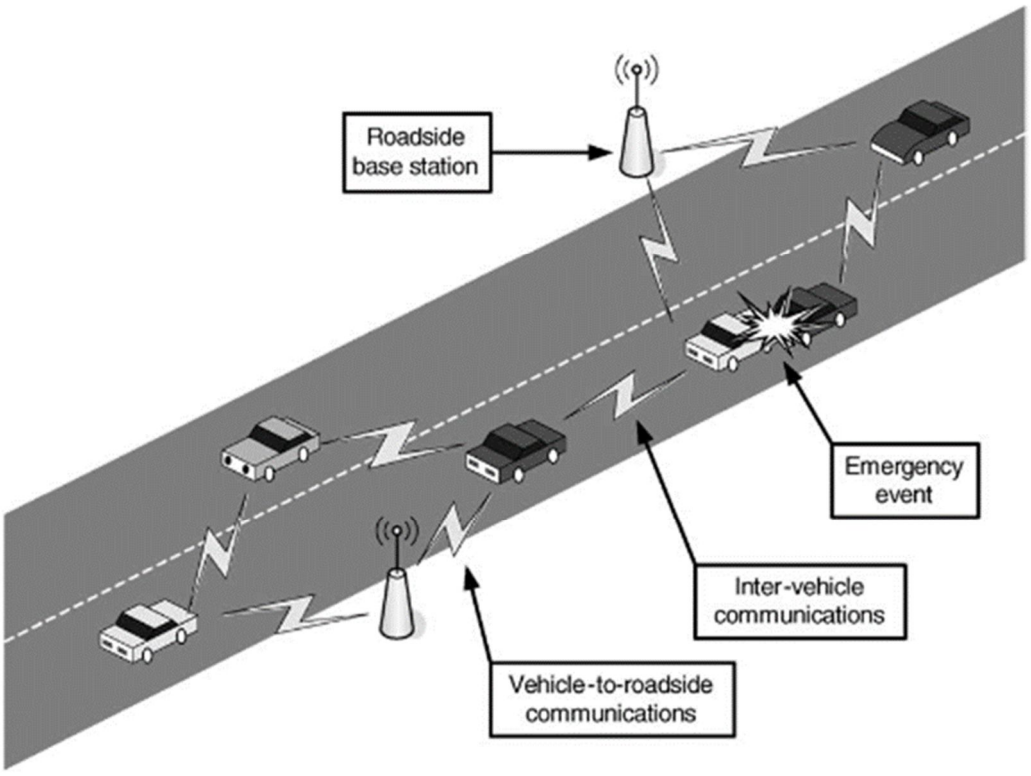
## 3.2 MANET

Manet-verkko (Mobile Ad Hoc Network) on nimensä mukaan rakennettu mobiililaitteista. Manet -verkot käyttävät IEEE 802.11 Standardin mukaisia laitteita. Manet -verkot ovat mesh-verkkojen kaltaisia, mutta manet -verkossa verkon solmut ovat usein liikkeessä. Manet -verkko voidaan muodostaa nopeassa ajassa monen langattoman laitteen välille ilman valmista infrastruktuuria. Se on myös edullinen vaihtoehto, koska ei tarvita kaapelointia eikä tukiasemia ja verkko rakentuu automaattisesti eikä sillä ole selvää topologiaa. Kun verkossa on pienempi käyttäjämäärä, silloin verkko toimii tehokkaammin. Myös manet -verkon laitteet toimivat reitittiminä muiden langattomien ad hoc -verkojen tapaan. Manet -verkko voi olla yhteydessä internettiin, jos verkon yhdelläkin laitteella on verkkoyhteys. Yhtä hyvin manet -verkko voi olla suljettu, niin ettei sillä ole yhteyttä internettiin.

Verkko muodostetaan seuraavan toimintaperiaatteen mukaan. Ensimmäinen laite rakentaa itsenäisen peruspalvelusetin (Independent Basic Service Set, IBSS). Sen jälkeen laite lähettää merkkisignaaleja, joita käytetään muiden laitteiden synkronointiin. Laitteet liittyvät verkkoon hyväksymällä merkkisignaalin parametrit. Verkon laitteet kuuntelevat toistensa merkkisignaaleja naapurilaitteilta ja naapureiden naapureilta. Näin kaikki laitteet tietävät pian toistensa sijainnit ja reitin jokaiselle laitteelle. Laitteet päivittävät omat kellonsa merkkisignaalien aikaleiman mukaan. Laite voi mennä nukkumaan manet -verkossa. Se kertoo muille laitteille nukkumaan menosta lähettämällä virranhallintabitin minkä tahansa lähettämänsä paketin kehyksessä. Paketteja ei lähetetä nukkuvalle laitteelle, paketit jäävät paikalliseen puskuriin. Laite herää säännöllisin väliajoin tarkastamaan ATIM-ikkunan (Announcement Traffic Indication Map), onko sille tullut viestejä. ATIM-kehys kertoo nukkuvalle laitteelle, että sillä olisi paketteja puskurissa. Nukkuva laite nousee lähettämään paketit eteenpäin ja lähettämisen jälkeen menee takaisin nukkumaan. Manet -verkon reititysprotokollia käydään läpi luvussa 4. Reititysprotokollan tehtävä on valita paras reitti lähettäjältä vastaanottajalle ja ylläpitää tietoja verkon muista laitteista. Riippuu reititysprotokollasta, mitä tietoja laitteet ylläpitävät muista laitteista ja verkosta yleensä. Paras reitti on jokin näistä; nopein, luotettavin, suorin tai halvin, sekin riippuu käytettävästä reititysprotokollasta. (Stallings, 2002), (Chlamtac, et al., 2003)

## 3.3 VANET

VANET (Vehicular Ad hoc Network) on ajoneuvo ad hoc -verkko. Se on uuden teknologian verkko, joka on saanut valtavasti huomiota viime vuosien aikana. Ajoneuvo ad hoc -verkot käyttävät IEEE 802.11p standardia, joka on kehitetty IEEE 802.11 standardin pohjalta. Standardi tukee älykästä siirtojärjestelmä (Intelligent Transportation System, ITS) sovellusta. ITS sovellus on kulkuneuvoille tarkoitettu kommunikointijärjestelmä, joka on verkossa, missä ajoneuvot ja ajoratayksiköt ovat keskenään kommunikoivia solmuja. Solmut vaihtavat keskenään tietoja turvavaroituksista ja liikennetiedoista. Ne voivat estää tehokkaasti liikenneruuhkia ja onnettomuuksia. Molemmat solmut ovat omistautuneet lyhyen alueen kommunikointi (Dedicated Short-Range Communications, DSRC) laitteisiin. DSRC- laitteet toimivat 5,9 GHz taajuusalueella 75 MHz kaistanleveydellä ja maksimi kantavuusalue on 1000 metrin luokkaa. Nämä standardit soveltuvat molempiin seuraavaksi esiteltäviin vanet- verkkoihin. Vanet-verkko koostuu kahdenlaisesta keskenään kommunikoivista verkoista, jossa vaihdetaan tietoa V2I (vehiche to Intrastructure) ajoneuvosta tiehen tai maastoon rakennettujen sovellusten välillä ja V2V (vehiche to vehiche) ajoneuvosta ajoneuvoon verkko. V2I-verkko kommunikoi tien sivuun tehtyjen sovellusten ajoratayksiköiden (Roadside Unit, RSU tai Roadside Base Station, RBS) kanssa. V2I-verkko voi ilmoittaa RSU välityksellä ruuhkista ja kolareista tai varoittaa lähestyvästä hälytysajoneuvosta. RSU on yhteydessä internettiin ja siitä johtuen sen ominaisuudet ja mahdollisuudet ovat rajattomat. Säätiedot ja kaupalliset tiedotteet kulkevat helposti ajoneuvoihin. V2V- verkko kommunikoi toisten ajoneuvojen kanssa vaihtamalla tietoja esimerkiksi nopeuksista, etäisyyksistä ajoneuvojen välillä ja tietysti sijaintitietoja. Muita turvallisuuteen liittyviä sovelluksia ovat muun muassa kaistan päättymisestä varoittaminen tai kulkuneuvojen yhteen törmäämisen estäminen. V2V-verkko lisää turvallisuutta huomattavasti liikenteessä. Sovelluksia kehitellään koko ajan lisää ja vain mielikuvitus on rajana. Tänä päivänä autot sisältävät paljon tietotekniikkaa, jota hyödynnetään auton hallintalaitteissa ja auton muissa toiminnoissa. Tietotekniikan lisääntyminen autoissa ja muissa kulkuneuvoissa on parantanut huomattavasti turvallisuutta liikenteessä ja myös kulkuneuvojen turvallisuutta. Autoihin on tullut viime vuosina suuri määrä lisää turvalaitteita, joita ohjaavat tietokone ja siihen liittyvät ohjelmistot. Vanet -verkko on manet -verkon kaltainen langaton mobiilitekniikkaa hyödyntävä verkko, joka kommunikoi ajoneuvojen välillä tai ajoneuvojen ja ajorataan tai maastoon tehtyjen sovellusten välillä. Seuraava kuva selventää vanet -verkon rakennetta ja toimintaa. (Kuva 5). (Gozalvez, et al., 2012), (Paul, et al., 2011), (Yang, et al., 2003), (Li & Wang, 2007).




**Kuva 5**. Vanet-verkko. (Raya & Hubaux, 2007)

Haastavaa ajoneuvojen välisessä kommunikoinnissa on erityisesti se, että ajoneuvot liikkuvat suurella nopeudella toisiinsa nähden ja tästä johtuen kommunikoivat lyhyen ajan jakson toistensa kanssa tai tukiasemien eli tien reunaan tehtyjen sovellusten kanssa. Sen lisäksi haasteita tulee vielä ympäristötekijöistä kuten rakennukset, kasvillisuus ja tietysti toiset ajoneuvot, jotka eivät kommunikoi toistensa kanssa tai keskenään. (Al-Rabayah & Malancy, 2012).

Vanet -verkko käyttää monia samoja manet -verkoista tuttuja protokollia. Esimerkkinä voidaan mainita Dynamic Source Routing Protocol (DSR), Optimized Link State Routing Protocol (OLSR) ja Ad Hoc On-demand Distance Vector (AODV). Ne ovat unicast reititysprotokollia manet-verkkoon ja sopivat hyvin myös vanet-verkkoon. Manet- verkkojen protokollia käsitellään luvussa 4. Vanet -verkon reititysprotokollat voidaan luokitella kahteen pääkatekoriaan. Topologiaan perustuvaan reititykseen ja maantieteelliseen perustuvaan (Geographic) eli paikannusperusteiseen reititykseen. Topologiaan perustuva reititysprotokolla käyttää linkin tilatietoja, jotka sen täytyy tietää voidakseen toimittaa paketit eteenpäin. Ehkä suorituskyvyltään paras topologiaan perustuva reititysprotokolla ajoneuvojen ympäristöön olisi ad hoc on-demand distance vector (AODV). Sillä on myös kaikista topologiaan perustuvista reititysprotokollista alhaisin reititys-kuormitus. Peruskriteeri topologiaan perustuvassa reititysprotokollassa on, kun verkon

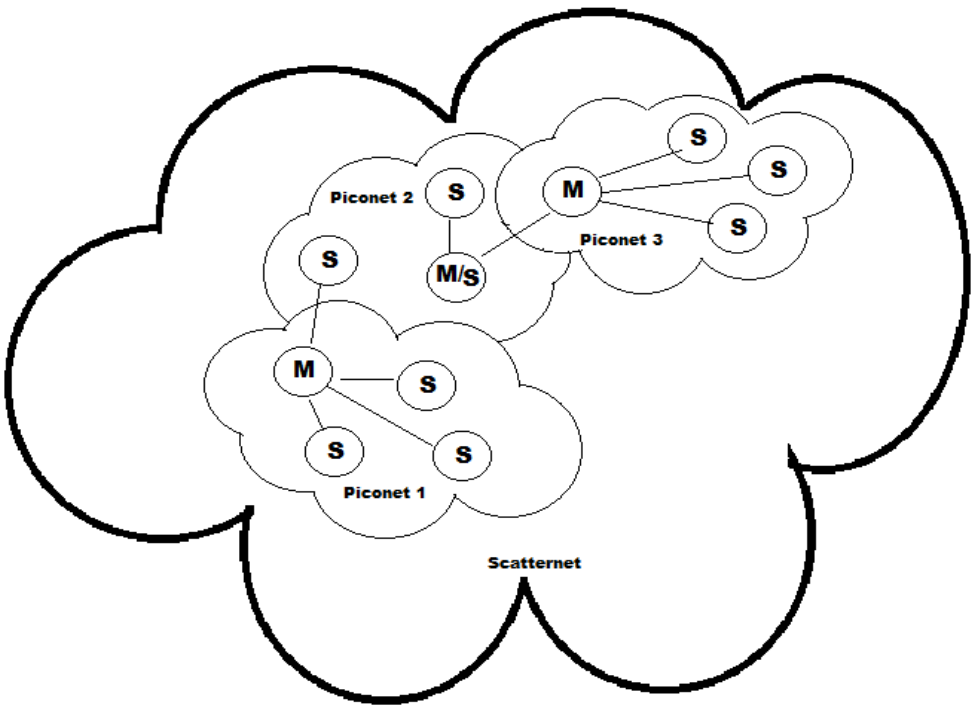
koko kasvaa niin sen suorituskyky pienenee. Se myös lisää skalaarisuusongelmia. Yleisesti uskotaan, että maantieteelliseen tai paikannukseen perustuva reititysprotokolla on ratkaisu skalaarisuusongelmaan. Tämä johtuu siitä uskomuksesta, että solmujen välillä on paljon vähemmän reitityskuormitusta, kun maantieteelliset reititysprotokollat eivät ylläpidä reititystauluja ja vaihda mitään linkkiasematietoja solmujen välillä. Maantieteellinen reititysprotokolla soveltuu hyvin dynaamiseen ympäristöön kuten vanet-verkotkin yleisesti ovat. Päätös eteenpäin lähettämiselle maantieteellisissä reititysprotokollissa kulkuneuvojen välillä perustuu pääsääntöisesti kohdekulkuneuvon ja kaikkien yhden hypyn päässä olevien kulkuneuvojen sijaintiin. Vastaanottajakulkuneuvon osoite tai oikeammin sijainti on pakattu lähettäjäkulkuneuvon lähettämän paketin osoitekenttään. Sijaintitietoja yhden hypyn alueella oleviin naapurikulkuneuvoihin pidetään yllä beacon-viesteillä, joita lähetetään säännöllisesti naapurikulkuneuvojen välillä. Salaukset voidaan hoitaa Vanet-verkossa samaan tapaan kuin muissakin verkoissa salausavainten vaihdoilla. Maantieteellinen reititys olettaa, että jokainen kulkuneuvo tietää sijaintinsa kulkuneuvossa olevan satelliittipaikannusjärjestelmän (Global Positioning System, GPS) avulla. Se olettaa myös, että lähettäjä tietää vastaanottajan sijainnin. Tämä vaatii tehokkaan paikannuspalveluohjausjärjestelmän, millä voidaan ylläpitää ajoneuvojen sijaintitietoja luotettavasti koko verkon alueella. Satelliittipaikannuksen yhtenä huonona puolena pidetään sitä, että tunneliin ajettaessa yhteys satelliittiin katkeaa ja silloin verkko on haavoittuvimmillaan. Vanet-verkon reititysprotokollista voi lukea lisää, vaikka lähteestä (Bernsen & Manivannan, 2009), sivulta 6 alkaen. (Miller, 2008).

## 3.4 BLUETOOTH

Bluetooth on lyhyille etäisyyksille tarkoitettu radiotekniikkaan perustuva langaton tiedonsiirtotekniikka. Bluetooth käyttää radiotaajuustekniikkaa 2,4 GHz ISM (industrial, scientific, medical) -taajuusalueella ja WPAN (Wireless Personl Area Network) alueella. Laitteiden etäisyydet vaihtelevat muutamasta metristä sataan metriin, jos olosuhteet ovat oikein suosiolliset voivat etäisyydet olla parhaimmillaan ylikin sata metriä. Se pääsee parhaimmillaan 720 kbits siirtonopeuteen, mikä riittää videokuvan ja multimediasiirtoon. Bluetooth käyttää IEEE 802.15.1 standardia. Sen tarkoituksena on korvata kaapeli tietokoneen, tulostimen, mobiililaitteen tai oheislaitteen (esim. hiiri tai näppäimistö) välillä. Bluetooth perustettiin ruotsalaisen L. M. Ericssonin toimesta vuonna 1994, kun se alkoi tutkia erilaisia menetelmiä langattomaan tiedon siirtoon mobiililaitteiden ja muiden langattomaan tiedonsiirtoon tarkoitettujen laitteiden välillä. Nimi bluetooth otettiin tanskalaiselta kuninkaalta Harald Blåtand (Bluetooth), joka hallitsi Tanskaa 900- luvulla. Vuonna 1998 perustettiin Ericssonin toimesta Bluetooth SIG (Special Interest Group) yhdessä Nokian, IBM:n, Intelin ja Toshiban kanssa. Tästä alkoi

kehitystyö, jonka tarkoituksena oli kehittää standardi langattomien laitteiden välille lyhyillä etäisyyksillä. Vuoden kehitystyön jälkeen ryhmä julkaisi ensimmäisen bluetooth protokollan vuonna 1999. Seuraavana vuonna 3COM, Agere (Lucent Technologies), Microsoft ja Motorola liittyivät SIG- ryhmään. (Ferro Erina, 2004, p. 2).

Bluetooth -verkko koostuu isäntälaitteesta(Master) ja yhdestä seitsemään renkilaitteesta (Slave). Käytän orja nimen sijaan renki -nimitystä. Tällaista verkkoa kutsutaan piconet -verkoksi. Piconet- verkkoon voi olla liittyneenä yhteensä 255 erillistä laitetta, joista 7 voi olla aktiivisena ja kommunikoida isäntälaitteen kansa. Loput odottavat passiivitilassa (park), että isäntälaite päättää, mikä laite seuraavaksi kommunikoi isäntälaitteen kanssa ja mikä laite passivoidaan. Piconet- verkko voidaan linkittää toiseen piconet-verkkoon tai useampaan. Tällöin piconet- verkot muodostavat verkojen kokonaisuuden, jota kutsutaan scatternet- verkoksi. Renkilaite voi olla useammassa verkossa saman aikaisesti ja voi olla toisen piconet- verkon renki ja toimia toisessa piconet- verkossa isäntänä. Selventävä kuva (kuva 6) bluetooth- verkkoarkkitehtuurista. (Sairam, et al., 2002)




**Kuva 6**. Bluetooth- verkkoarkkitehtuuri. M on isäntälaite, S on renkilaite ja M/S on toisen piconet -verkon isäntä ja toisen piconet -verkon renki.

Bluetooth määritellään kerrosprotokolla -arkkitehtuurilla, joka sisältää seuraavat elementit.

- Radio. Kerros koostuu ilmatiehen tarvittavista vaatimuksista kuten taajuushyppelyn käyttö, modulaatiotekniikka ja lähetysteho. Vaatimukset, jotka liittyvät laitteiden toimintaan 2,4 GHz ISM (industrial, scientific, medical) -taajuusalueella.
- Baseband. Baseband kerros on kantataajuusosa, joka ohjaa radiorajapinnan toimintoja.
- LMP (Link Manager Protocol). Kerroksen tehtävänä on yhteyden muodostus, linkkitason toiminnot, laitteen tila, salaus ja nimipalvelut.
- L2CAP (Logical Link Control and Adaptation Protocol). Tehtävänä kantataajuusprotokollan liittäminen ylempiin protokolliin.
- SDP (Service Discover Protocol). Kerroksen tehtävänä palvelinsovellusten palvelut ja tietokantapalvelut.
- RFCOMM (Radio Frequency Communication). Se on sarjalinjan uudelleen sijoitus protokolla eli emulointiprotokolla.
- HCI. (Host Controller Interface). Se on rajapinta, joka määrittelee tavan ohjata bluetooth -laitetta.

Protokollista ja rajapinnoista voi halutessa lukea lisätietoa lähteestä (Bisdikian, 2001).

Bluetooth -verkossa tiedonsiirto tapahtuu taajuushyppelyyn perustuvalla hajaspektritekniikalla, joka tunnetaan nimellä Frequency Hopping Spread Spectrum (FHSS). Modulointitekniikka on Gaussian frequensy Shift-Keying (GFSK). Koko kaistanleveys käsittää 79 fyysistä kanavaa, joista jokainen on leveydeltään 1 MHz. Lähetty viesti vaihtaa fyysistä kanavaa 1600 kertaa sekunnissa hyppimällä kanavalta toiselle isäntälaitteen kanssa sovitun algoritmin mukaan. Aikaväli yksittäisellä kanavalla kestää ennen kanavan vaihtoa 0,625 ms. Taajuushyppely hankaloittaa viestin kaappaamista, mikä lisää tiedonsiirron luotettavuutta. Taajuushyppelyn siirtokanavia voi häiritä, mutta se oppii välttämään taajuuksia, joissa esiintyy häiriöitä. Bluetooth lähetystehot luokitellaan kolmeen luokkaan. Luokka 1 on 20 dBm lähetysteho (100 mW). luokan 2 lähetysteho on 4 dBm (2,5 mW) ja luokan 3 lähetysteho on 0 dBm (1mW). Luokan 2 ja 3 laitteet ovat pääsääntöisesti niitä laitteita, joita on käytössä enemmän. Luokan 1 laitteissa on aina virranhallintaominaisuuksia, mitä ei kaikissa luokan 2 ja 3 laiteissa välttämättä ole. (Bisdikian, 2001)

Isäntälaitteen ja renkilaitteen välillä voi kahdenlaisia fyysisiä linkkejä eli yhteystyyppiä; Synchronous Connection- Oriented (SCO) ja Asynchronous Connection -Less (ACL). SCO on piirikytkentäinen ja yhteysnopeus sama molempiin suuntiin. Yhteys varataan koko yhteistyön ajaksi. SCO käytetään pääsääntäisesti äänensiirtoon. Siinä ei ole myöskään pakettien uudelleen lähetystoimintoa. ACL on taas pakettikytkentäinen eli siirrettävä tieto kulkee paketteina. Lähetykset voi tapahtua isännän ja rengin välillä eri nopeuksilla. Isäntä valvoo kaistanleveyttä ja paketit voidaan lähettää tarvittaessa uudelleen. (Stallings, 2002)

Bluetooth -verkossa voi verkon laitteilla olla kaksi eri lopputilaa; Standby ja Connection. Standby tila tarkoittaa, että laite on valmiustilassa. Connection tarkoittaa, että laite on yhteydessä verkkoon niin sanotussa yhteystilassa. Jokaisen renkilaitteen yhteystila alkaa Poll – viestillä, jolla renkilaite synkronoituu isännän kellotaajuuteen ja sovitaan kanavataajuushyppely algoritmi. Kun renkilaite on yhteystilassa, niin se on yhdessä alla mainitussa toiminnallisessa tilassa.

- Active. Laite toimii piconet -verkossa aktiivisena laitteena, joka kuuntelemalla, lähettämällä ja vastaamalla isännän viesteihin toimimalla verkon jäsenenä.
- Sniff. Renkilaite toimii virransäästötilassa, jolloin se tarkistaa harvemmin isäntälaitteen lähetyksiä.
- Hold. Renkilaite voi olla hetken irti verkosta ilman, että yhteys ja tahdistus isäntälaitteeseen katkeaa. Isäntälaite voi laittaa kaikki renkilaitteet hold -tilaan.
- Park. Renkilaite luovuttaa aktiivisen verkko -osoitteen (active member address AM_ADDR) ja saa tilalle passiivisen verkko -osoitteen (parking member address PM_ADDR). Isäntälaite lähettää ajoittain park -tilassa olevalle renkilaitteelle synkronointidatan (beacon-viesti), jotta laite pysyy verkossa. Koska passiivitilassa on hyvin vähän verkkoliikennettä, se säästää renkilaitteen virran kulutusta.

Bluetooth -verkoissa tietoturva ja luotettavuus muodostuvat uuden laitteen todennuksesta ja tiedonsiirron salauksesta. Verkon hallintaan käytetään MAC -osoitteita, avainten vaihtoa ja todennuksessa satunnaislukua. Bluetooth -laitteet sisältävät satunnaislukugeneraattorin, jolla voidaan hallita erilaisia salaus -ja tunnistusmenetelmiä. Bluetoothlaitteet tarvitsevat tunniste -ja salausavaimia verkon hallintaan. (Sriskanthan, et al., 2002).

## 3.5 Ad hoc -verkkoon kohdistuvia uhkia

Tässä on muutamia uhkia, jotka ovat yleisiä ad hoc -verkkoon kohdistuvia hyökkäyksiä tai haittatekijöitä. Langaton ad hoc -verkko alttiimpi kaikenlaisille ulkopuolisille uhille verrattuna langalliseen verkkoon. Esimerkiksi mesh-verkossa tai Manet- verkossa voi olla kahdenlaisia hyökkääjiä; Ulkoisia ja sisäisiä. Ulkopuolinen hyökkääjä tarkkailee verkkoa ulkopuolelta eikä näin ollen ole osa verkkoa. Ulkopuolinen hyökkääjä tarkkailee verkon liikennettä, joka kulkee yhdyskäytäväreititintä pitkin ja yrittää saada ongittua tietoja. Voi ulkopuolinen hyökkääjä tarkkailla verkon solmua ja kalastaa tietyn yhteysvälin vastaanottaja- ja lähettäjätietoja, jotka ovat verkkoliikenteessä selkokielisenä tekstinä. Pakettien tyyppiä on lähes mahdoton verkon ulkopuolisentarkkailijan saada, sen

sijaan reititintä kuunteleva hyökkääjä saa selville käytetyt osoitetiedot. Langattomissa ad hoc -verkoissa yksittäinen laitehan toimii myös reitittimenä. Sen sijaan sisäinen hyökkääjä on osa verkkoa, solmu tai voi olla useampiakin. Sisäinen hyökkääjä tietää ainakin yhden hypyn päähän olevat solmut. Koska paketit kulkevat hyökkääjäsolmun kautta eteenpäin, hyökkääjäsolmu saa selville pakettien kohde- ja lähdeosoitteiden lisäksi pakettien tyypit. Verkossahan kulkee kahdenlaisia paketteja, tietopaketteja ja verkonhallintaan liittyviä paketteja. Langattomassa verkossa solmujen tarkkailu on melko helppoa, koska ilmatiellä kulkeva tieto on kaikkien ulottuvilla. Lisäksi verkossa voi olla solmuja, joiden luotettavuutta ei tunneta. Sen lisäksi sisäinen hyökkääjä voi tehdä yhteistyötä ulkoisen hyökkääjän kanssa ja saada tietoja solmujen toiminnasta. Sisäisellä hyökkääjällä on myös mahdollisuus muuttaa ja muokata viestejä tuottamalla vääränlaista liikennettä verkkoon. Solmujen kaappaamista ja identiteettivarkauksia joskus tapahtuu verkossa. Paras keino turvautua tähän on salausmenetelmien ja digitaalisten allekirjoitusten käyttö viestien lähettämisessä kahden solmun välillä. Tästä esimerkkinä mies välissä hyökkäys (man-in-the-middle attack). Tästä käytetään myös nimitystä epärehellinen välittäjä. Silloin hyökkääjä toimii kahden solmun välillä toimien välittäjänä ja halutessaan kerää tietoja tai kuuntelee viestejä, joita pystyy myös halutessaan muuttamaan. Viestin lähettäjä ja vastaanottaja luulevat kommunikoivansa keskenään, mutta todellisuudessa kaikki liikenne kiertää hyökkääjän kautta. Hyökkääjä esiintyy mies välissä hyökkäyksessä vääränä henkilönä. Hyökkääjä voi myös etsiä aktiivista solmua tai solmuja, jotta pystyisi tarkkailemaan tämän liikennettä internettiin. Solmuille pitää pystyä tarjoamaan anonymiteetti, jottei kuka tahansa tiedä, mikä solmu on yhteydessä internettiin. Olisi löydettävä tapa, millä solmut pystyisivät pitämään kaikki osapuolet anonyymeina. Tähän ratkaisu voisi löytyä niin sanotusta sipulireitityksestä, jossa tiedot salataan kerroksittain paketeiksi ja solmu saa avattua kerroksen vain omalla salaisella avaimella. Näin reititystiedot pysyvät salassa ulkopuolisilta. Osapuolet tietävät vain naapurisolmunsa, josta paketit tulevat ja minne paketit lähetetään eteenpäin. Sipulireitityksestä voi lukea lisää, jos kiinnostusta riittää vaikkapa lähteestä. (Koskinen, 2007)

Palvelunestohyökkäyksellä tarkoitetaan hyökkäystä, jolla aiheutetaan verkkoon niin paljon liikennettä, että verkko ruuhkautuu, jolloin verkossa tapahtuva palvelu tai liikenne hidastuu tai katkeaa kokonaan. Hyökkäykset vaikuttavat verkon toimintaan kuten tiedon eheyteen, verkon suorituskykyyn ja koko verkon palveluihin. Palvelunestohyökkäykset toteutetaan käyttämällä hyväksi ohjelmointivirheitä tai tekemällä ohjelmia, jotka suorittavat palvelunestohyökkäyksen. Hyökkäykset kohdistuvat verkkokapasiteetin kulutukseen, resurssien kuten levytilan, muistin tai suoritintehojen ylikuormittamiseen sekä järjestelmien ja sovellusten kaatamiseen. Tavallinen hyökkäys kohdistetaan verkkosivulle tai palvelimelle aiheuttamalla niin paljon liikennettä, että verkkosivu tai palvelin tukkeutuu viestien määrästä ja palvelu katkeaa. Näitä viestejä voidaan lähettää kaapatuista tietokoneista ja kohdistamalla niistä kaikki liikenne vain palvelunestokohteeseen. Palvelunestohyökkäykset voivat kohdistua mihin tahansa OSI-mallin kerrokseen joko fyysiseen kerrokseen, verkkokerrokseen, kuljetuskerrokseen tai sovelluskerrokseen.

Peilaushyökkäys tarkoittaa hyökkäystä, jossa käytetään hyväksi UDP-protokollan heikkouksia. Peilaushyökkäys hyödyntää UDP-protokollan käyttämiä palveluita, joita ovat DNS (Domain Name System), NTP (Network Time Protocol), SNMPv2 (Simple Network Management Protocol) ja SSDP (Simple Service Discovery Protocol). Peilaushyökkäykset perustuvat UDP-protokollan haavoittuvuuteen tai oikeastaan ominaisuuteen, koska UDP-protokolla on yhteydetön protokolla eli se ei neuvottele yhteyden muodostusta tehdessä lähettämisen parametreista. Vastaanottaja ei myöskään tarkista lähettäjän osoitetietoja. Peilattu palvelunestohyökkäys tehdään hyökkääjän ohjaamalla bottiverkolla, joka etsii verkosta tietokoneita, joissa on heikosti suojattu UDP-palvelu, esimerkiksi jokin edellä mainituista. Hyökkääjän ohjaamat bottiverkon kaapatut tietokoneet lähettävät huonosti suojatuille koneille esimerkiksi kyselyn. Kun koneet vastaavat kyselyyn, niin lähdeosoitteeksi ei ole merkitty todellinen lähettäjä vaan hyökkäyksen kohteen osoite. UDP-palvelu, joka toimii välikätenä lähettää bottikoneiden vastaukset hyökkäyksen kohteelle. Näin saadaan kuormitettua peilaushyökkäyksen uhrin laitetta.

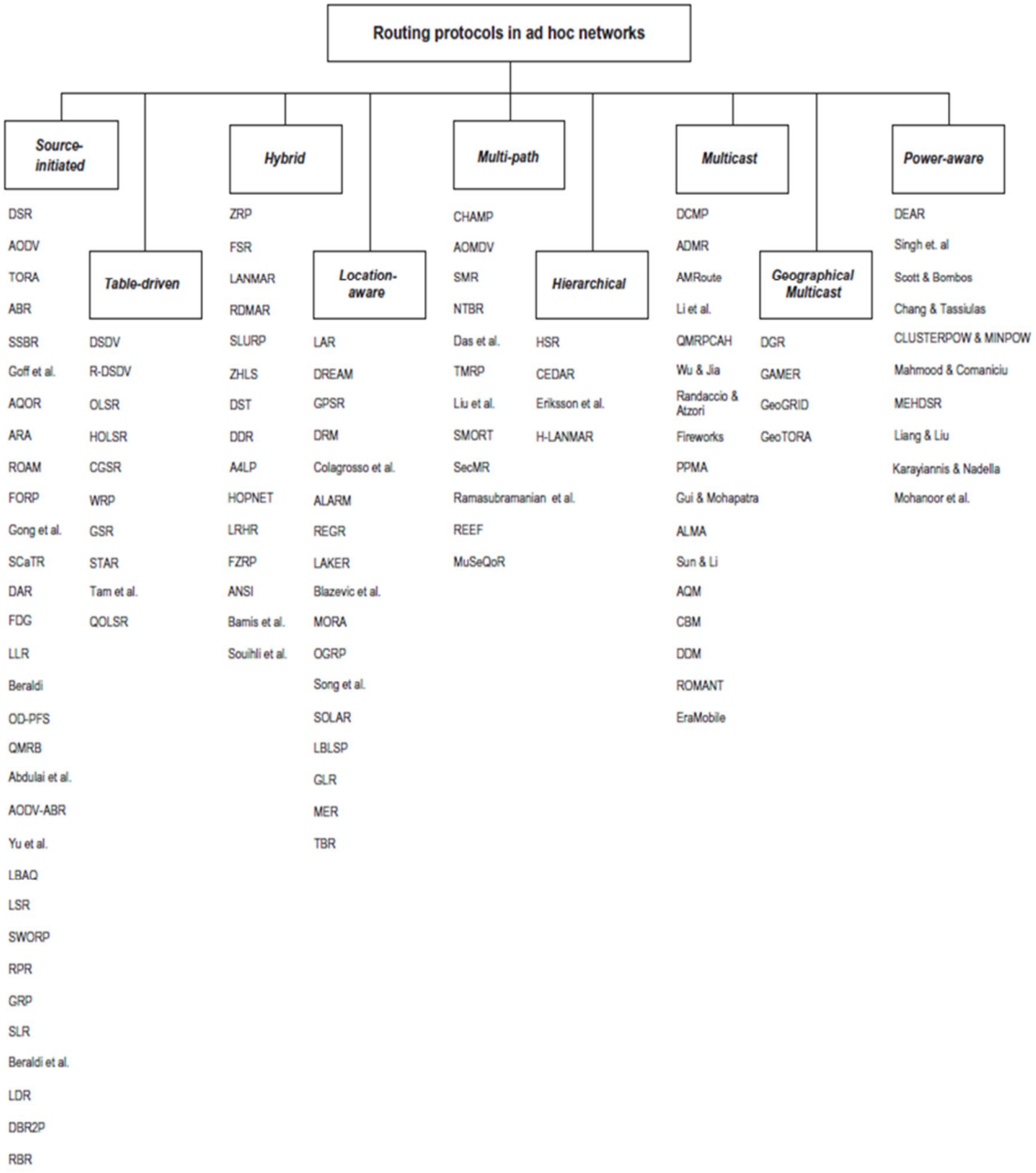
Bluejacking -hyökkäys ja bluesNarf -hyökkäys ovat bluetooth -verkkoon kohdistuvia hyökkäyksiä. Bluejacking- hyökkäys on tavallaan sosiaalinen hyökkäys, jossa hyökkääjä yrittää saada hyökkäyksen kohteen hyväksymään bluetooth- yhteyden muodostamisen pyynnön. Tällä tavalla hyökkääjä voi päästä käyttäjän laitteeseen ja sen sisältämiin tietoihin. Tähän voi varautua, ettei vastaa epämääräisiin yhteyden muodostuspyyntöihin. Bluesnarf -hyökkäys on kohdistettu joihinkin matkapuhelimiin, joissa on huonosti toteutettu bluetooth-verkon standardi. Tällainen hyökkääjä saa yhteyden laitteeseen ja pääsee laitteen tietoihin omistajan huomaamatta hyökkäystä ollenkaan. (Koskinen, 2007)

# 4. AD HOC -VERKON PROTOKOLLAT

Aikaisemmin mainittiin, että ad hoc -verkossa on kymmeniä protokollia eikä standardia, joka ohjaisi niiden toimintaa ja ominaisuuksia. Protokollien tehtävänä on laatia säännöt, miten verkossa toimitaan ja protokollan laatimien sääntöjen mukaan saadaan toimitettua viestit perille oikeille vastaanottajille lyhyimmällä, edullisimmilla ja luotettavimmalla reitillä. Se on haastavaa protokollille varsinkin langattomissa verkoissa, joissa on otettava huomioon, että solmu voi mobiililaite. Solmu voi olla epäluotettava, solmulla on usein rajoitettu virrankäyttö ja muita epävarmuustekijöitä. Aikaisemmin ad hoc -verkon protokollat oli luokiteltu tarvittaessa (On-demand) ja tauluajettaviin (Table-driven) protokolliin. Näiden kahdentyyppisen protokollan lisäksi oli kehitetty risteymäprotokolla (Hybrid). Ad hoc -verkkojen koon kasvaminen on aiheuttanut sen, että on pitänyt seurata kehitystä ja kehittää paremmin toimivia protokollia koko ajan laajeneviin verkkoihin. Tästä voisi mainita sellaiset tarpeelliset käytettävät tekniikat kuin hierakinen (Hierarchical) reititys ja maantieteellinen (Geographical) reititys. Lisäksi energiakulutusongelmat valtaavat alaa muilta suorituskykymittauksilta ja virtaa säästävät protokollat yleistyvät kovaa vauhtia tai ominaisuudet, jotka kuluttavat vähemmän laitteiden virtaa koko ajan kasvavalla mobiiliverkkojen aikakaudella. Toimintatapansa ja arkkitehtonisen jaottelun mukaan reititysprotokollat voidaan jakaa yhdeksään luokkaan.

- Lähdekäynnistettävä reititysprotokolla (Sourse-initiated, Reactive tai On-demand), vastavaikutteinen reititysprotokolla tai tarvittaessa.
- Taulu-ajettava reititysprotokolla (Table-driven). Ennakoiva reititysprotokolla.
- Risteymäreititysprotokolla (Hybrid).
- Sijaintitietoinen reititysprotokolla (Location-aware, Geographical). Maantieteellinen reititysprotokolla.
- Monipolkuinen reititysprotokolla (Multipath).
- Hierakinen reititysprotokolla (Hierarchical).
- Monilähetys reititysprotokolla (Multicast).
- Maantieteellinen monilähetysreititysprotokolla (Geographical Multicast).
- Virtatietoinenreititysprotokolla (Power-aware).

Taulukko (taulukko1) selventää, mitä reititysprotokollia ad hoc -verkossa on.




**Taulukko 1**. Reititysprotokollat ad hoc -verkossa.

Lähdekäynnistettävä reititysprotokolla toimii niin, että luodaan reitti lähteen kysyessä reittiä kohteelle. Reitinetsintä alkaa lähdesolmun lähettämällä reittietsintätiedustelun kaikille naapurisolmuille ja naapurisolmut edelleen naapureilleen ja niin edelleen. Tätä jatkuu niin kauan, että viesti tavoittaa kohteen tai jonkun solmun, joka tietää reitin kohteelle. Reittihuoltotoiminto kestää ajanjakson, minkä lähdesolmu on asettanut. Esimerkiksi DSR (Dynamic Source Routing) ja AODV (Ad hoc On-demand Distance Vector) ovat lähdekäynnistettäviä tai vastavaikutteisia reititysprotokollia.

Taulu-ajettavat reititysprotokollat ylläpitävät aina reittienpäivitystietoja joka solmulta jokaiselle solmulle koko verkon alueella. Jokainen solmu tallettaa reititystiedot reititystauluunsa ja reittipäivityksillä pidetään reittitiedot niin ajan tasalla kuin mahdollista koko verkon alueelta. Eri protokollat pitävät eri reititystilatietoa poluista. Kaikilla protokollilla on tavoitteena pitää reittien huoltokuormitus niin pienenä kuin on mahdollista. Tämän tyyppiset reititysprotokollat eivät sovellu kovin hyvin laajoihin dynaamisiin verkkoihin, koska reititystaulujen päivitykset ja reittien ylläpito joka solmulle kuormittavat laajasti verkkoa. Esimerkiksi DSDV (Destination-Sequenced Distance Vector Routing Protocol) ja OLSR (Optimized Link State Protocol) ovat taulu-ajettavia tai ennakoivia reititysprotokollia.

Risteymäreititysprotokolla yhdistää parhaat ominaisuudet kahdesta edellä esitellystä reititysprotokollasta, taulu-ajettavasta ja lähdekäynnistettävästä reititysprotokollista. Ajatuksena on, että alueilla, missä verkko muuttuu suhteellisen hitaasti taulu-ajettavat reititysprotokollat soveltuvat sinne paremmin, kun taas alueet suuremmalla liikkuvuudella soveltuisivat paremmin lähdekäynnistettäville reititysprotokollille. Yhdistämällä nämä kaksi erilaista reititysjärjestelmää voidaan saavuttaa suurempi koko verkon alueen suorituskyky. Esimerkiksi ZRP (Zone Routing Protocol) ja FSR (Fisheye State Routing) ovat risteymäreititysprotokollia.

Sijaintitietoinen reititysprotokolla olettaa mobiili ad hoc -verkossa jokaisen yksittäisen solmun tietävän jokaisen solmun sijainnin koko verkon alueella. Maailmanlaajuinen satelliitteihin perustuva paikannusjärjestelmä (Global Positioning System, GPS) on paras ja helpoin tekniikka määrittää tarkat koordinaatit maantieteellisestä sijainnista näille solmuille. Reititysprotokollat hyödyntävät tätä sijaintitietoa määritelläkseen reittejä. Esimerkiksi LAR (Location-aided routing) ja DREAM (Distance Routing Effect Algorithm for Mobility) ovat sijaintitietoisia reititysprotokollia.

Monipolkuiset reititysprotokollat luovat useita reittejä lähdesolmulta kohdesolmulle, kun taas toiset reititysprotokollat luovat vain yhdet reitit. Etu tästä monipolkuetsinnästä on, että kaistanleveys linkkien välillä käytetään tehokkaammin suuremmalla toimitus luotettavuudella. Se myös auttaa verkon ruuhka-aikoina, joita voi ilmetä lähetysten räjähdysmäisen kasvun takia verkossa. Monipolut luodaan tarvittaessa tai käyttämällä ennakoivaa lähestymistapaa ja on erittäin tärkeää, että reitit saadaan yleensä irrotettua nopeasti johtuen solmun liikkuvuudesta. Esimerkiksi CHAMP (Caching and multipath routing protoco) ja AOMDV (Ad hoc On-demand Multipath Distance Vector Routing) ovat monipolkuisia reititysprotokollia.

Hierarkkinen reititysprotokolla. Kun verkko kasvaa kooltansa, nämä lähestymistavat johtavat kasvavaan reititystaulujen kokoon ja lisääntyvään kontrollipakettien kuormitukseen. Hierarkkinen reititysprotokolla rakentaa solmuhierarkian tyypillisesti ryhmätekniikoilla. Solmut korkeammalla hierarkiatasolla hankkivat erityispalvelua, parantaa skalaarisuutta ja reitityksen tehokkuutta. Esimerkiksi HSR (Hierarchical State Routing) ja CEDAR (Core-extraction Distributed Ad Hoc Routing) ovat hierarkkisia reititysprotokollia.

Monilähetysreititysprotokolla. Monilähetys tarkoittaa samanaikaista tiedon lähettämistä yhdeltä lähettäjältä monelle vastaanottajalle. Monet laajasti käytetyt sovellukset vaativat monilähetystä vähintäänkin loogisella tasolla. Esimerkkinä audio-video puhelinkokous, reaaliaikainen videosuoratoisto ja yleinen tietokannan huoltotoimenpide. Monissa tapauksissa on hyödyllistä toteuttaa monilähetys reititysalgoritmi tasolla. Esimerkiksi DCMP (Dynamic Core Based Multicast Routing) ja ADMR (Adaptive Demand-driven Multicast Routing) ovat monilähetysreititysprotokollia.

Maantieteellinen monilähetysreititysprotokolla lähettää poikkeavan monilähetyksen, missä päämäärä on, että reitille paketit tulevat lähettäjältä vastaanottajille, joiden sijainti on erityisellä maantieteellisellä alueella. Toimiakseen maantieteellinen monilähetysprotokolla vaatii, että solmuilla on luotettava paikallistamistekniikka käytössään. Esimerkiksi GPS eli satelliittipaikannusjärjestelmä. Esimerkkinä maantieteellisistä reititysprotokollista voisi mainita DGR (Direction Guided Routing) ja GAMER (Geocast Adaptive Mesh Environment for Routing).

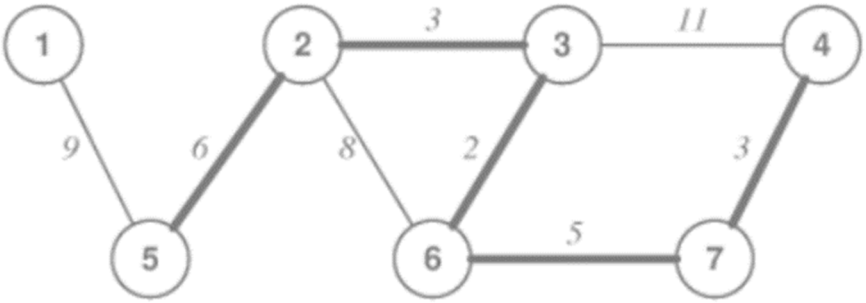
Virtatietoinenreititysprotokolla tekee reitityspäätöksen harkinnan perusteella, joka on riippuvainen solmujen saatavilla olevasta energiasta. Tämä harkinta voi olla huomattavasti vaikeampi kuin löytää reitti matalimmalla energiatasolla. Lyhyin reitti on usein myös pienimmällä energiakulutuksella, ainakin yksi pienenemmistä. Tämän luokan protokollat valitsevat molemmat sekä solmujen heterogeeniset energiavarat, että yhtä hyvin epäsäännöllisen energiasäästön johtuen verkon topologiasta ja tietovirran ympäristöstä. Monille näille protokollille lopullinen tavoite on maksimoida verkon elinaika solmuille, joilla on rajoitetut ja pysyvät energiavarat. DEAR (Device and Energy Aware Routing) ja MEHDSR (Minimum Energy Hierarchical Dynamic Source Routing) ovat virtatietoisiareititysprotokollia. (Boukerche, 2011).

Seuraavaksi käsitellään reititysprotokollia ja sen jälkeen monipolkureititysprotokollia, joita pidetään luotettavimpina, koska ne rakentavat valmiiksi vaihtoehtoisia reittejä lähdesolmulta kohdesolmulle. Luotettavien reititysprotokollien jälkeen tässä luvussa esitellään REEF (REliable and Efficent Forwarding) ja PIDIS (Protocol-Independent Packet Delivery Improvement Service), jotka ovat mekanismeja, joilla parannetaan verkon luotettavuutta ja tehokkuutta. Esitellään lyhyesti muutama edellä mainitun kaltaisia sovelluksia tai palveluita reititysprotokolliin, joilla parannetaan verkon luotettavuutta, tehokkuutta ja palvelun laadullisuutta.

## 4.1 Reititysprotokollat

Langattomien verkkojen yleistyminen ja kehittyminen on aiheuttanut suuria haasteita protokollille ja niiden toiminnalle. Langattomat verkot luodaan usein spontaanista ja niillä ei ole selvää topologiaa. Luotettavan tiedonsiirron luominen päästä päähän aiheuttaa reititysprotokollille kaikenlaisia haasteita eikä pelkästään vain langattoman verkon

solmujen liikkuvuuden takia. Reititysprotokollan tulee kasvattaa verkon luotettavuutta ja toimintavarmuutta kaikissa olosuhteissa. Manet –verkko käyttää perinteistä TCP/IP struktuuria rakentamaan päästä päähän kommunikoinnin solmujen välille. Mobiiliverkon toimiminen TCP/IP mallissa on hurjan kehitystyön tulos. Reititys on haasteellinen tehtävä manet -verkossa. Tästä johtuen mobiiliverkkoihin on suunniteltu suuri määrä erilaisia reititysprotokollia ja reititysprotokollilta puuttuu kokonaan standardi. Vaatii älykkään reititysstrategian verkkoon, että pystyy tehokkaasti selviytymään verkon topologiamuutoksien kanssa. Verkon koko ja liikennevirta voi kasvaa samalla kun solmut ovat liikkeessä. Reititysprotokolla tarvitsee toimiakseen kunnolla erilaisia laatutasoja erityyppisille sovelluksille ja käyttäjille. Pystyäkseen reitittämään ja lähettämään paketit eteenpäin seuraaville solmuille ja edelleen kohdesolmulle, pitää olla jokin tapa, jolla tiedetään mihin paketit toimitetaan. Solmujen pitää tietää mihin paketit toimitetaan ja vielä lyhyintä mahdollista reittiä. Siihen, miten solmut löytävät lyhyimmät ja luotettavimmat reitit lähdesolmulta kohdesolmulle, on kehitetty kaksi pääalgoritmia langattomaan verkkoon. Nämä algoritmit, joilla reittien arvot ja etäisyydet lasketaan ovat linkkitila- algoritmi (link -state) ja etäisyysvektori-algoritmi (distance vector). Linkkitilareitityksessä joka solmu ylläpitää päivitysnäkymää verkon tilasta lähettämällä säännöllisesti linkkitilaetäisyystiedon jokaiselle naapurisolmulle. Jokainen verkon solmu vastaanottaa tiedon koko verkon etäisyyksistä ja näin algoritmi laskee lyhyimmän mahdollisen reitin jokaiselle kohteelle. Etäisyysvektorireitityksessä jokainen solmu ylläpitää etäisyystietoa jokaisen naapurisolmun kautta kohdesolmulle. Säännöllisin väliajoin naapurisolmu lähettää päivitystiedon etäisyystiedosta naapurisolmuille. Näin solmut saavat päivitettyä tiedon jokaisesta kohdesolmusta ja reitistä sinne. Tästä johtuen jokainen solmu osaa valita lyhyimmän reitin kohdesolmulle. Linkkitila -ja etäisyysvektorialgoritmit eivät sovellu kovinkaan hyvin laajoihin manet -verkkoihin, koska säännöllinen ja tiheä reittien päivittäminen isoissa verkoissa saattaa viedä suuren osan verkon saatavilla olevasta kaistanleveydestä. Näin ollen se saattaa kasvattaa verkossa ruuhkaa ja lisäksi kuluttaa jokaisen solmun saatavilla olevaa laitteen virtakapasiteettia. Manet -verkossa reititysprotokollat jaetaan kolmeen ryhmään. Ensimmäinen on ennakoivat reititysprotokollat (proactive routing protocols), joista voidaan käyttää myös nimeä global tai taulu-ajettava (table - driven). Toinen on vastavaikutteiset reititysprotokollat (reactive routing protocols), joista voidaan käyttää myös nimeä on -demand. Kolmas on risteymäreititysprotokolla (hybrid routing protocols). Ennakoivissa reititysprotokollissa reitit kaikkiin kohdesolmuihin pidetään ja päivitetään säännöllisesti käyttämällä päivitysprosessia. Vastavaikutteisissa reititysprotokollissa reitit luodaan tarvittaessa. Risteymäreititysprotokollilla on ominaisuuksia sekä ennakoivista että vastavaikutteisista reititysprotokollista. Alla on kuva reitityksen periaatteesta (kuva 7). (Comer, 2009), (Abolhasan, et al., 2004), (Wang, et al., 2009).




**Kuva 7**. Reitityksen periaate. Paksumpi viiva kuvaa valittua reittiä solmulta 5 solmulle 4.

### 4.1.1 Ennakoivat reititysprotokollat (Proactive routing protocols)

Ennakoivissa reititysprotokollissa jokainen solmu pitää reititystietoa jokaiselle kohdesolmulle koko verkon alueella. Solmut pitävät näitä reititystietoja reititystauluissa. Nämä reititystaulut päivitetään säännöllisin väliajoin, kun verkon topologia muuttuu. Protokollien välillä on eroja sen suhteen, minkä tyyppisiä tietoja pidetään yllä reititystauluissa ja miten päivityksiä hoidetaan. Joka reititysprotokollalla voi eri määrä reititystauluja ja voivat siitä johtuen ylläpitää erilaisia tietoja.

Destination-Sequenced Distance Vector Routing Protocol (DSDV) tarjoaa yksinkertaisen polun kohdesolmulle käyttämällä etäisyysvektorireititysalgoritmia, joka laskee lyhyimmän reitin kohdesolmulle. Vähentääkseen kuormitusta verkossa se käyttää kahdenlaisia päivityspaketteja. Nämä päivityspaketit sisältävät joko koko päivitystiedon tai vain osan siitä. Täysipaketti kuljettaa kaiken päivitystiedon ja paketti, joka kuljettaa vain osan päivitystiedosta toimittaa vain tiedon, joka poikkeaa viimeisestä täydestä päivityspaketista. Osapakettipäivitys viestit lähetetään useammin kuin täydet paketit. DSDV kuljettaa suuren määrän kuormaa verkossa johtuen vaatimuksista säännöllisiin päivitysviesteihin ja näin ollen kuormittaa verkkoa kohtuuttomasti. Osapakettipäivitykset kuormittavat kerralla vähemmän verkkoa, mutta lähetykset tapahtuvat useammin

kuin täysipäivityspaketit. Tästä johtuen DSDV reititysprotokolla ei sovellu laajoihin verkkoihin, koska se vie päivitysprosesseilla suuren osan verkon kaistanleveydestä.

Wireless routing protocol (WRP) käyttää etäisyysvektori ja Bellman-Ford algoritmia polun laskutoimituksiin. Se on luuppivapaa ja välttää väliaikaiset luupit käyttämällä edeltäjän tietoa. WRP vaatii jokaisen solmun ylläpitämään neljää päivitystaulua. Tämä kasvattaa muistikapasiteetin määrää joka solmulla ja samalla verkon kuormitus lisääntyy. WRP käyttää hello- viestejä, jotka myös lisäävät verkossa liikennettä ja näin myös kuormittavat verkkoa lisää. Hello -viestit vaihdetaan naapurisolmujen välillä, jos ei ole pakettien lähettämistä. Tällä pidetään yhteys naapurisolmuun. Tämä lisää virrankulutusta yksittäisille laitteille ja vie myös verkon kaistanleveyttä. Solmujen täytyy myös olla aktiivitilassa eikä solmu voi mennä nukkumaan eli virransäästötilaan.

Global state routing (GSR) käyttää reititykseen linkkitila –algoritmia. GSR on kehittänyt tavan vähentää liikennettä verkossa lähettämällä linkkitila -algoritmilla päivitysviestit ainoastaan välittäjäsolmuille. Jokainen solmu ylläpitää linkkitilataulun päivitystietoa vain, joka on lähetetty naapurisolmuilta ja päivittää säännöllisiä linkkitilatietoja vain naapurisolmujen kanssa. Tämä vähentää kontrolliviestien määrää verkossa huomattavasti.

Fisheye state routing (FSR) kuvataan tehokkaana ja yksinkertaisena reititysprotokollana, joka ylläpitää topologista karttaa jokaiselle solmulle. FSR käyttää linkkitila -algoritmia reititystietojen päivitykseen. FSR eroaa perinteisestä linkkitilaprotokollasta tavalla, jolla se levittää reititystietoa. Se vaihtaa koko linkkitilatiedot vain naapurisolmujen kanssa. Fisheye nimi tulee tavasta, millä solmu päivittää ja ylläpitää tarkkoja etäisyys ja polun laatutietoja aivan lähipiiristä, mutta yksityiskohtien määrä kasvaa etäisyyden suhteessa muihin solmuihin. Jokainen solmu ottaa huomioon ympäristön määrän fish-eye ulottuvuudessa, koska alueet mistä FSR ylläpitää tietoa voivat olla kahden jopa useammankin hypyn päässä riippuen polun pituudesta. Sen linkkitilataulussa on vain päivitystiedot, joita se saa naapurisolmuilta. (Pei, et al., 2000).

Optimized Link State Protocol (OLSR) on pisteestä pisteeseen reititysprotokolla, joka perustuu perinteiseen linkkitila–algoritmiin. Perinteisessä linkkitilareititysprotokollassa jokainen solmu ylläpitää topologiatietoja jaksottaisilla linkkitilaviesteillä. Parannellussa versiossa OLSR minimoi kontrolliviestien koon ja uudelleen lähetysten määrää solmuille reittien päivitysten aikana käyttämällä MPR (multipoint relay) strategiaa eli monipistelähetystä. Siinä jokainen verkon solmu valitsee joukon naapurisolmuja, jotka välittävät sen paketteja. Solmu valitsee monipistelähettäjäksi (MPR) vain naapurisolmut, jotka lähettävät paketteja toiselle hypylle eivät siis kaikkia naapurisolmuja. Solmu lähettää Hello -viestin naapurisolmun kautta kahden hypyn päähän, näin solmu saa tietää naapurisolmut ja verkon topologian kahden hypyn päähän. Näin solmu laskee ja päivittää reitin monipistelähettäjän kautta kohdesolmulle. Jokainen verkon solmu päivittää reititys -ja topologiatietoja vain naapurisolmun kanssa, joka välittää paketteja eteenpäin eli valittu MBR. Tämä säästää verkon resursseja, koska kaikki solmut eivät osallistu päivityspakettien välittämiseen. (Arun, et al., 2008).

Distance routing effect algorithm for mobility (DREAM). Tässä reititysprotokollassa jokainen solmun paikka tiedetään maantieteellisten koordinaattien avulla. Nämä koordinaatit paikannetaan GPS (Global Positioning System) paikantimen avulla. DREAM käyttää GPS paikannussysteemiä, jolla se paikantaa solmut. Koordinaatit vaihdetaan solmujen välillä ja jokaisen solmun sijaintitiedot tallennetaan omiin reititystauluihin. Sijaintitietojen vaihtaminen on suuri etu DREAM reititysprotokollaa käytettäessä verrattuna perinteisiin linkkitila -ja etäisyysvektoritietojen välittämiseen verkossa, koska DREAM tarvitsee sijaintitietojen välittämiseen vähemmän kaistanleveyttä. Tämä tekee siitä skaalautuvamman. Tekemällä taajuuksilla lähetettävät päivitysviestit suhteutettuna liikkuvuuteen ja etäisyyteen, reitityskuorma vähenee entisestään. Paikallaan pysyvien solmujen ei tarvitse lähettää lainkaan päivitystietoja. (Abolhasan, et al., 2004).

Source-tree adaptive routing (STAR) perustuu linkkitila-algoritmiin. STAR on tauluajettava reititysprotokolla. Jokainen solmu etsii ja ylläpitää verkkotopologiatietoa koko verkosta ja rakentaa lyhyimmän polkupuun kohteelle. STAR reititysprotokolla hoitaa naapurin havaitsemisen ja topologiatietojen vaihdon solmujen kesken. STAR reititysprotokollalla on kaksi mekanismia löytää naapurit. Jokainen solmu lähettää Hello-viestin säännöllisesti verkkoon tiedustellakseen naapurisolmun olemassa olosta. Viestit voivat olla pieniä paketteja, jotka eivät sisällä mitään muuta. Kun solmu vastaanottaa viestin joltain, se tietää löytäneensä uuden naapurisolmun. Jos solmu ei saa vastausta viestiin, se päättelee, että naapuri on poissa verkosta tai ulkona sen kuuluvuusalueelta. Toinen tapa löytää naapuri on naapuriprotokolla, joka voi olla toteutettuna linkkikerroksella. Se ilmoittaa STAR reititysprotokollalle uuden naapurin olemassa olosta tai yhteyden menettäneen naapurin takaisin tulosta. Naapuriprotokollalla ei ole hello-viestejä. Käyttämällä kuormitusta vähentävää reitityslähestymistapaa nimeltä LORA (Least-Overhead Routing Approach), se vähentää kontrollikuormitusmäärää ad hoc -verkko ympäristössä ja käyttämällä optimaalista reitityslähestymistapaa nimeltä ORA (Optimum Routing Approach), se saavuttaa lyhyimmän reitin kohteelle. Lähdesolmun ei tarvitse ylläpitää lyhyimpien reittien tietoja kohdesolmulle. Se vähentää topologiapäivityksiä ja tekee säännölliset päivityslähetykset ehdollisiksi rajoittamalla linkkitila-algoritmin toimintaa. Topologiapäivityksiä tehdään vain, kun verkossa on varmasti tapahtunut muutos. STAR reititysprotokolla toimii hyvin laajoissa verkoissa, koska se on pystynyt vähentämään huomattavasti ruuhkaa supistamalla päivitysliikennettä ja samalla vähentämään viivettä verkossa käyttämällä valinnaisia reittejä. Ilman näitä toimenpiteitä STAR reititysprotokollalla olisi huomattavan suuri muisti- ja prosessikuormitus laajoissa ja liikkuvissa verkoissa. Jokaiselta solmultahan vaaditaan ylläpitämään osittaista verkon topologiagrafiikkaa, mihin kuuluu vaihtaa säännöllisesti tietoja kuten naapurisolmun pitämästä raportista verkon erilaisista lähdepuista. (Jiang & Garcia-Luna-Aceves, 2001).

### 4.1.2 Vastavaikutteiset reititysprotokollat (Reactive routing protocols)

On -demand reititysprotokollat on kehitetty vähentämään kuormitusta proaktiivisissa protokollissa ylläpitämällä tietoa vain aktiivisille reiteille. Vastavaikutteisissa reititysprotokollissa reitti kohteeseen selvitetään vasta lähetyksen yhteydessä. Reittien etsiminen tapahtuu tavallisesti lähettämällä reitinpyyntöpaketti verkkoon. Vastavaikutteiset reititysprotokollat voidaan luokitella kahteen katekoriaan: lähdereititys ja hyppy hypyltä reititys. Lähdereititys on -demand protokollissa joka datapaketti kantaa täydellistä osoitetta lähteeltä kohteelle. Tämän takia jokainen välittäjäsolmu, joka lähettää paketit eteenpäin, sallii osoitetietojen pysyä jokaisen paketin otsikossa ja lisää omat tietonsa. Välittäjäsolmujen ei tarvitse ylläpitää päivitysreititystietoja eteenpäin lähetettäessä paketteja kohti kohdesolmua. Solmujen ei myöskään tarvitse lähettää naapurisolmujen kanssa beacon-viestejä. Lähdereititysprotokollien haitta laajoissa verkoissa on, etteivät ne toimi kovinkaan hyvin. Välittäjäsolmujen määrän kasvaminen kohtuuttoman suureksi voi aiheuttaa verkossa reittivikoja. Jos välittäjäsolmujen määrä jokaisella reitillä kasvaa, kannettavan kuorman määrä jokaisen datapaketin otsikossa voi kasvaa reilusti. Nämä protokollat eivät skaalaudu hyvin laajoihin verkkoihin, joissa on paljon hyppyjä ja liikkuvuutta. Hyppy hypyltä reititys tunnetaan myös pisteestä pisteeseen reitykseksi. Näissä verkoissa datapaketti kantaa vain kohdesolmun osoitteen ja seuraavan hypyn osoitteen. Jokainen välittäjäsolmu reitillä kohti kohdesolmua käyttää reititystaulua lähettämään datapaketit eteenpäin kohden kohdesolmua. Reitit ovat mukautuvia dynaamisiin muutoksiin manet-verkkoympäristössä ja jokainen solmu voi päivittää reititystaulunsa, kun ne vastaanottavat uutta topologiatietoa verkosta ja lähettävät datapaketteja uusille ja paremmille reiteille. Vähemmän reittien uudelleenlaskemista tarvitaan datapakettien lähettämisen aikana, kun käytetään tuoreempia reittejä. Jokaisen välittäjäsolmun pitää ylläpitää reititystietoa aktiivisista reiteistä ja olla hereillä vastaamaan beacon- viesteihin. Erilaisten vastavaikutteisten protokollien määrän kasvaminen on lisännyt vastavaikutusreitityksen tehoa.

Ad hoc on-demand distance vector (AODV). Tunnettu ja hyvin määritelty reaktiivinen reititysprotokolla, joka on määritelty Internet-standardissa RFC 3561. Protokolla perustuu DSDV ja DSR algoritmeihin. AODV reititysprotokollalla on kolme kontrolliviestiä joita se lähettää verkkoon reitin ylläpitoon.

- RREQ. A route request -viesti. Lähetetään, kun solmu haluaa reitin kohdesolmulle. Jokainen RREQ kantaa time to live (TTL) arvoa, kuinka monta hyppyä tämä viesti voidaan lähettää eteenpäin.
- RREP. A route reply -viesti. Reittivastaus -viesti. Kontrolliviesti eli vastaus reitin löytymisestä.
- RERR. A route error -viesti. Reittivirhe -viesti. Kontrolliviesti reitin huoltamisen tapauksessa.

Solmujen kommunikointi toistensa välillä on erinomaista. Lähdesolmulla, välittäjäsolmulla ja kohdesolmulla kaikilla pakettien lähettämisessä erilainen tehtävä. Kun lähdesolmu haluaa yhteyden kohdesolmulle, se tarkistaa reititystaulusta onko tuore reitti kohdesolmulle saatavilla. Jos tuore reitti on saatavilla, se käyttää sitä. Jos reittiä ei ole reititystaulussa, se käynnistää reitin etsinnän lähettämällä RREQ kontrolliviestin kaikille naapurisolmuille. Välittäjäsolmut lähettävät saman RREQ kontrolliviestin omille naapurisolmuille. Tätä jatketaan, kunnes RREQ kontrolliviesti saavuttaa kohdesolmun tai kun välittäjäsolmulla on tuore reitti selvillä kohdesolmulle, joka vastaanottaa RREQ viestin. Tämän jälkeen luodaan RREP kontrolliviesti, joka lähetään lähdesolmulle. Lähdesolmu lähetettyään RREQ viestin jää odottamaan vastausta viestiin eli odottaa RREP kontrolliviestiä polun löytymisestä kohdesolmulle. Kun polku lähdesolmulta kohdesolmulle on löytynyt, sen jälkeen lähetetään viestipaketit kohdesolmulle. Samalla solmut päivittävät reititystaulunsa lisäämällä uuden ja tuoreen reitin. Jos reittilinkki on pois toiminnasta tai reitissä on vikaa reitin huoltotoimenpiteiden takia, siinä tapauksessa lähetetään RERR kontrolliviesti kaikille solmuille, jotka käyttävät tätä reittiä lähetyksiin. (Mistry & Jinwala, 2010).

Dynamic source routing (DSR) on on -demand reititysprotokolla. Se ei ole kovin tehokas laajoissa verkoissa ja yleensäkin verkoissa, joissa on paljon hyppyjä. Tämä johtuu, siitä, että se kantaa koko reitin osoitetietoja mukanaan lähdesolmusta kohdesolmuun. Tästä johtuen se kuormittaa verkkoa kohtuuttomasti ja kaistanleveyttä koko reitin varrelta. DSR on toiminnaltaan saman kaltainen kuin AODV. Se muodostaa reitin samalla tavalla kontrolliviesteillä. Reititysprotokolla koostuu kahdesta mekanismista, reitinetsintä ja reitinhuolto. Solmut voivat pitää muistissaan monikertaisen määrän reittejä. Lähetettäessä solmut voivat tarkistaa reittitaulusta, onko reitti muistissa. Jos reitti on reititystaulun muistissa, solmun ei tarvitse lähettää reitinkyselyviestiä verkon yli. Tästä on se hyöty, että se vähentää verkossa tapahtuva liikennettä. DSR ei vaadi beacon -viestien ja hello -viestien vaihtamista solmujen välillä. Siksi solmu voi käyttää hyväkseen tämän ja mennä sleep -tilaan eli nukkumaan. Tämä säästää solmujen laitteiden virrankäyttöä ja samalla myös vähentää liikennettä verkossa ja näin se ei myöskään kuormita niin paljon verkon kaistanleveyttä. Reitin huoltoon kuuluu reitin RERR kontrollipaketit, joilla ilmoitetaan reittien ja linkkien välisistä ongelmista. Myös kuittauspaketit kuuluvat reitin huoltoon. (Usop, et al., 2009).

Temporally- ordered routing algorithm (TORA). Park ja Corson on kehittänyt TORA reititysprotokollan. TORA on luuppivapaa, mukautuvainen ja yleinen reititysalgoritmi, joka perustuu linkkipurkukäsitteeseen. Se käyttää ohjaavaa asyklistä grafiikkaa (directed asyclic graphs DAG) luomaan reitit ylä -tai alavirtaan. Tämä grafiikka antaa mahdollisuuden TORA reititysprotokollan luoda parempia reittejä toimien apuvälineenä laajoissa verkoissa, joissa on paljon solmuja. TORA reititysprotokollan pitää synkronoida solmut protokollasovelluksen rajapintaan saadakseen tämän graafisen ominaisuuden. Kun muissa reititysprotokollissa tapahtuu linkkikatkos, silloin reititysprotokollan täytyy käynnistää uudelleen reittietsintä. TORA reititysprotokolla hoitaa reittikatkoksen kiertämällä katkospisteen. Tämä ominaisuus aiheuttaa pienemmissä verkoissa suurempaa

kuormaa, mutta suuremmissa verkoissa se parantaa skaalaavuutta. TORA reititysprotokollalla on neljä toimintoa: luominen, ylläpito, pois pyyhkiminen ja reittien tehostaminen. TORA reititysprotokollan etuna on, että se on vähentänyt kaukaa saapuvien kontrolliviestejä naapurisolmuryhmien kanssa, missä topologiavaihto on tapahtunut. TORA reititysprotokolla tukee myös monilähetystä, vaikka se ei ole sen perustoimintoja. (Gupta, et al., 2011), (Abolhasan, et al., 2004).

### 4.1.3 Risteymä reititysprotokollat (Hybrid routing protocols)

Risteymäreititysprotokollat ovat uuden sukupolven reititysprotokollia, joilla ominaisuuksia sekä ennakoivista reititysprotokollista että vastavaikutteisista reititysprotokollista. Yleensä vielä parhaat ominaisuudet molemmista edellä mainituista reititysprotokollista. Reititysprotokollalla pyritään lisäämään verkon skalaarisuutta tavalla joka samalla vähentää reitin etsinnän kuormitusta. Tämä useimmiten saavutetaan ominaisuuksilla ennakoivista reititysprotokollista ylläpitämällä reittejä lähisolmujen välillä ja kaukaisempiin solmuihin käytetään vastavaikutteisista protokollista tuttua reittien etsintästrategiaa. Näin risteymäreititysprotokollat voivat käyttää hyviä ominaisuuksia molemmista reititysprotokollista sekä ennakoivista että vastavaikutusprotokollista. Monet risteymäprotokollat päivittävät tietoja alueittain. Tämä tarkoittaa, että verkko on jaettu alueisiin ja jokainen solmu näkee verkon joukkona alueita. Toisten ryhmien solut nähdään puuna tai ryhmänä. Manet-verkossa on suuri määrä risteymäreititysprotokollia. (Beijar, 2002), (Nikaen, et al., 2001)

Zone routing protocol (ZRP) on tyypillinen risteymäreititysprotokolla, joka käyttää ominaisuuksia ennakoivista reititysprotokollista ja vastavaikutteisista reititysprotokollista. Nimensä mukaan ZRP reititysprotokolla on jaettu alueisiin, jotka muodostuvat ryhmistä solmuja. ZRP on sopiva erityisesti jatkuvasti muuttuville ja koko ajan liikkeessä oleville mobiiliverkolle. Jokainen solmu päivittää ennakoivan reititysprotokollan tapaan reititystaulun paikallisella alueella, joka on luotu reititysprotokollan toimesta tietyn kokoisen ympyrän säteen päähän solmusta. Alue käsittää nämä solmut, jotka ovat tällä ympyrän alueella hypyn tai hyppyjen päässä solmusta. Solmut, jotka ovat reititysalueen ulkopuolella, niille ZRP reititysprotokolla etsii vastavaikutteisten protokollien tapaan on-demand menetelmällä reitit. Tämä kuormittaa verkkoa vähemmän, koska silloin verkossa ei ole niin paljon päivitysliikennettä. ZRP reititysprotokollalla on kaksi aliprotokollaa, jotka ovat IntrAzone Routing Protocol (IARP) ja IntErzone Routing Protocol (IERP). IARP ja IERP eivät ole erityisreititysprotokollia. IARP on paikallisesti toimiva komponentti, joka nimensä mukaan hoitaa ZRP reititysprotokollan määrittelemän alueen sisäistä reititystä ennakoivien protokollien tapaan. IERP taas hoitaa reitityksen alueen ulkopuolella vastavaikutteisten protokollien tapaan. Koska ZRP koostuu monista komponenteista, jotka hoitavat yhdessä koko reitityksen reitin alusta reitin loppuun. Jokainen komponentti toimii itsenäisesti riippumatta toisista komponenteista ja

usein vielä käyttää eri tekniikkaa. Tämä parantaa tehokkuutta koko verkon toiminta-alueella. Tästä esimerkkinä, että ennakoiva reititysprotokolla OLSR ja sitä voidaan käyttää IERP protokollan sijaan. AODV on vastavaikutteinenprotokolla ja se voi korvata IARP reititysprotokollan. Voidaan käyttää muitakin ennakoivia protokollia ja vuorovaikutteisia protokollia. (Beijar, 2002), (Lemus & Mendez, 2004), (Subramaniam, 2003).

Distributed dynamic routing (DDR) on puutopologiaan perustuva reititysprotokolla. DDR reititysprotokollassa puilla ei ole juurisolmua. Puurakennelmassa jokainen solmu kommunikoi lähettämällä jaksottaisia beacon-viestejä ja vaihtamalla niitä vain naapurisolmujen kanssa. Puuhun kuuluu keskenään kommunikoivat solmut. Puut, jotka on yhdistetty toisiinsa yhdyskäytävällä, muodostavat verkossa metsän. Jokainen puu muodostaa alueen. Näin verkko on rakentunut dynaamisista alueista, joissa jokainen solmu laskee jaksollisesti alueensa id-osoitteen itsenäisesti. Jokainen alue on yhdistetty toisiinsa solmujen välityksellä. Solmut, jotka yhdistävät alueita eivät kuulu samaan puuhun, mutta ovat yhteydessä lähetysalueen toisiin solmuihin. Näin koko verkko näyttää joukolta yhdistettyjä alueita. Tämän johdosta jokainen solmu voi kommunikoida minkä tahansa alueen minkä tahansa solmun kanssa. DDR -algoritmi vertaa, laskee ja asettaa kuutta seuraavassa esiteltävää tietoa verkossa. (i) paremman naapurisolmun valintaa, (ii) puun sisäistä ryhmää, (iii) puun ulkoista ryhmää, (iv) metsän rakennetta, (v) alueen nimeämistä ja (vi) alueen sijaintitietoja. (Bakht, 2005), (Gupta, et al., 2011).

Zone-based hierarchical link state (ZHLS) perustuu hierarkkiseen struktuuriin, missä verkko on jaettu ei-ylikerroksisiin alueisiin. Jokaisella solmulla on uniikki solmuosoite (node ID) ja alueosoite (zone ID), mitkä on laskettu käyttämällä maantieteellistä tietoa. Siksi verkko seuraa kaksitasoista topologiastruktuuria. Tasot ovat solmutaso ja aluetaso. ZHLS reititysprotokollalla on kahdenlaisia linkkitilapäivityksiä. Solmutason linkkitilapaketti (LSP Link State Packet) sisältää solmun naapurisolmujen osoitteet solmun alueella ja muiden alueiden alueosoitteet. Solmu lähettää säännöllisesti solmutilansa linkkitilapaketit jokaiselle solmulle omalla alueella. Kaikki alueen solmut pitävät samoja solmutason linkkitilatietoja koko solmutilan linkkitilapakettien vaihdon ajan. Lähdesolmu tarkistaa ennen lähettämistä sisäisen alueen reititystaulusta onko kohdesolmun osoite sen omalla alueella, jos on, lähdesolmulla on osoitetieto selvillä ja se voi lähettää viestin. Toisessa tapauksessa lähdesolmu lähettää osoitekyselyn muille alueille yhdyskäytävän läpi kaikille solmuille ja saa vastauksena alueosoitteen ja solmuosoitteen kysytystä vastaanottajasta. ZHLS reititysprotokollalla lähetyt paketit vievät vähän kaistanleveyttä, koska sillä on pieni reitityskuorma. Reitityspolku on sopeutuvainen dynaamiselle topologialle johtuen lyhyistä ositetiedoista. Vaadittavat ositetiedot ovat vain solmuosoite ja alueosoite.

Distributed spanning trees based routing protocol (DST). DST reititysprotokollalla on puumainen topologia. Solmut verkossa jaetaan ryhmiin puiden lukumäärän mukaan. Jokaisella puulla on kaksi erilaista solmua, jotka ovat reittisolmu ja sisäinen solmu. Juuri kontrolloi puun rakennetta ja puu voi yhdistyä toisiin puihin. Loput solmuista näissä puissa ovat vapaita solmuja. Jokainen solmu voi olla yhdessä näistä kolmesta eri tilassa.

Se voi olla reititin, yhdistäjä ja konfiguroija riippuen tehtävätyypistä, mitä se on suorittamassa. DST reititysprotokollalla on kaksi toimintatapaa määritellä reitti lähdesolmun ja kohdesolmun välille. Hybridi puu tulviminen (HTF Hybrid Tree Flooding), jossa lähdesolmu lähettää kontrollipaketin kaikille naapurisolmuille ja viereisille silloille syklissä, missä jokainen paketti odottaa ajan, jota kutsutaan odotusajaksi. Kun se saa vastauksen, niin kohde on löydetty. Toinen on yleinen syklisukkulointi (DST Distributed Spanning Tree shuttling), jossa lähdesolmu lähettää kontrollipaketit puun reunalle, kunnes ne ulottuvat lehtisolmuun. Kun paketti löytää lehtisolmun, se lähetetään eteenpäin sukkulointitasolla. Näin se löytää reitin kohteelle. Jos sukkulointitaso on jo löydetty, kontrollipaketit voidaan lähettää puuta pitkin tai viereisille silloille ja näin vastaanottajaa kohden. DST algoritmin haittana on, että se luottaa juurisolmun konfigurointiin. Lisäksi odotusaika (holding time) aiheuttaa viivettä verkkoon. (Abolhasan, et al., 2004)

## 4.2 Luotettavat monipolkureititysprotokollat

Monipolkureititysprotokollat ovat reititysprotokollia, jotka ruuhkan, linkin katoamisen tai muun yllättävän syyn takia ovat varautunet useammilla reittivaihtoehdoilla lähdesolmulta kohdesolmulle, selviytyäkseen yllättävistä reitillä tapahtuvista ongelmista. Monipolkureititysprotokollien päätavoite on rakentaa luotettava kommunikointi ja myös varmistaa kuormitustasapaino parantaakseen palvelun laatua mobiili ad hoc -verkoissa. Nämä monipolkureititysprotokollat ovat luokitelleet viisi kategoriaa, joilla ne saavuttaa päämääränsä. Nämä kategoriat ovat pienentää verkossa tapahtuvia viiveitä, parantaa verkon ja reitityksen luotettavuutta, vähentää verkossa tapahtuvia kustannuksia, maksimoida verkon ja solmujen elinaika ja suosia risteymäreititysprotokollien (Hybrid Routing) käyttöä. Monipolkureititysprotokollien tehtäviin kuuluu useiden polkujen etsintä ja ylläpitää löydettyjä polkuja voidakseen rakentaa vaihtoehtoisia reittejä tarvittaessa. Manet-verkko on itseasentuva, itsejärjestäytyvä ja itsehuoltava. Manet-verkolla voi olla dynaaminen topologia. Manet-verkon solmuilla on rajalliset varat kuten akku, rajallinen muisti ja virran kulutusvara prosesseihin, jotka kuluttavat paljon virtaa, on usein liian pieni.

### 4.2.1 Caching and multipath routing protocol (CHAMP)

CHAMP reititysprotokolla käyttää tiedonvälimuistia ja lyhyintä monipolkuista reititystä. Se myös vähentää pakettien pudotusta tiheiden reittikatkosten yhteydessä. Jokainen solmu ylläpitää pientä puskurimuistia eteenpäin lähetettävistä paketeista. Tekniikka on hyödyllinen silloin, kun solmu joutuu katkaisemaan eteenpäin lähetyksen kohteelle

lähetyskatkoksen tai reittikatkoksen takia ja eikä voi toimittaa paketteja perille. Tällaisessa tapauksessa lähdesolmu ei joudu lähettämään paketteja uudelleen vaan välittäjäsolmu voi lähettää ne puskurimuistista eteenpäin. Tämä vähentää päästä päähän pakettiviivettä. Ja tällaisissa tapauksessa, että ne toteutuisivat pitäisi monipolkuja kohteelle olla saatavilla. Jokainen solmu ylläpitää kahta välimuistia; reittivälimuisti, joka koostuu eteenpäin lähettämistiedoista ja reittitiedusteluvälimuisti, joka sisältää viimeksi saadut ja toteutetut reittitiedustelut. Ne reittimerkinnät, joita ei ole käytetty erityisen reitin elinaikana, poistetaan reittivälimuistista. Solmu ylläpitää lähetyspuskuria lähetystä odottaville paketeille ja tietovälimuistia varastoimaan viimeksi eteenpäin lähetetyille tietopaketeille. Reittienetsintä aloitetaan, jos ei ole saatavilla valmista reittiä. Kohdesolmu vastaa reittitiedusteluun reittivastauspaketilla. Voi olla monia tasa-arvoisia reittejä, joilla on sama vakiintunut pituus, koska jokainen eteenpäin lähetyskerta-arvo, mikä aloitetaan nollasta lähteeltä ja jokainen uudelleen lähetys kasvattaa arvoa yhdellä. (Valera, et al., 2002), (Boukerche, 2011)

### 4.2.2 Ad hoc on-demand multipath distance vector routing (AOMDV)

AOMDV on monipolkureititysprotokolla, joka kehitetty AODV (Ad Hoc Distance Vector) pohjalta ja laajennettu monipolkuiseksi reititysprotokollaksi. AOMDV käyttää reittien etsintään AODV reititysprotokollasta tuttua menetelmää, joka esiteltiin AODV reititysprotokollan esittelyn yhteydessä. AOMDV reititysprotokolla laskee polkujen etsinnän aikana lyhyimmät polut kohteelle. Tehtävä koostuu kahdesta osasta. Ensinnä käytäntö reittien päivitykseen löytääkseen luuppivapaat monipolkuiset reitit jokaiselle solmulle ja toiseksi yleinen protokolla laskee ja etsii luotettavat polut, joilla ei ole epäjatkuvia solmuja tai linkkejä. AOMDV reititysprotokolla löytää epäjatkuvat solmut tai linkit lähteeltä kohteelle. Linkki katkeaa yleensä solmun liikkuvuuden, solmun katoamisen, liikennöintiruuhkan, pakettien törmäyksen tai muun vastaavan tyyppisen syyn takia. Löytääkseen epäjatkuvat reitit jokainen solmu ei välittömästi hylkää kopioita reittikyselyviestistä (RREQ). Jokainen reittikyselyviesti, joka saavuttaa epäjatkuvan solmun polun, saapuu eri naapurilta lähdesolmulle, koska solmut eivät voi lähettää kopioita reittikyselyviesteistä. Mitkä tahansa kaksi RREQ- viestiä, jotka saapuvat välittäjäsolmun läpi lähdesolmun eri naapurisolmulle eivät ole voineet kulkea saman solmun kautta. Saadakseen epäjatkuvat linkit, vastaanottajasolmu lähettää reittilöytynytvastauksen (RREP) riippumatta kopio RREQ ensimmäisistä hypystä. Varmistaakseen jatkuvan linkkiyhteyden ensimmäisellä hypyllä, kohdesolmu vain palauttaa vastauksen yksikäsitteiselle naapurisolmulle, mistä reittikysely tuli. RREP seuraa käänteisiä polkuja, mitkä

ovat epäjatkuvat linkit ja solmut ensimmäisen hypyn jälkeen. Jokainen vastausviesti läpäisee välittäjäsolmun ja myös ottaa eri käänteisen polun lähdesolmulle varmistaakseen jatkuvan linkkiyhteyden. Vastausviestit palautetaan aina samaa polkua kuin reittikyselyt tulivat. Luupit löydetään reittikyselyiden yhteydessä, jos RREQ palaa välittäjäsolmulle eteenpäin lähettämisen jälkeen eri solmulta mihin välittäjäsolmu oli lähettänyt paketit, näin luuppi on löydetty suurella varmuudella. Solmut tunnistaa paketit sekvenssinumeroista eli pakettien järjestysnumeroista. (Marina & Das, 2001), (Periyasamy & Karthikeyan, 2011), (Kute & Kharat, 2013).

### 4.2.3 Neighbor table based multipath routing (NTBMR)

NTBMR reititysprotokollassa jokainen solmu ylläpitää naapuritaulua, mikä rekisteröi reititystietoja pitäen sisällään asetetun hyppymäärän naapurisolmun kautta. Hyppymäärä voi olla mikä tahansa arvo, mutta hyppymäärä kasvattaa kontrollikuormitusta, mitä pitempää solmu ylläpitää tietoa muista solmuista sen suurempi tietomäärä kulkee verkossa ja myös vie solmun muistia. NTBMR reititysprotokollalla on reittietsintä ja reittihuoltomekanismi. Se ylläpitää reittimuistia jokaiselle solmulle koko verkon alueella. Reittimuistit päivitetään naapuritaulujen tiedoilla, missä on myös arvioitu langattoman linkin elinaika. Tämä tieto auttaa pitämään reittijonojen elinaikaa yllä. Jokainen solmu lähettää säännöllisesti beacon-viestejä kahden hypyn päähän naapurisolmuille. NTBMR reititysprotokolla käyttää kahta mekanismia, jotka ovat aika-ajo (Time driven) mekanismi ja tieto-ajo (Data driven) mekanismi. Näillä mekanismeilla kerätään tietoa naapuritauluun. Aika-ajo mekanismissa solmu lähettää beacon-viestin eritystä reittiä pitkin ja päättelee, onko reitti aktiivinen. Jos reitti on aktiivinen, solmu lisää tiedot kaikista naapurisolmuista kahden hypyn päästä, jotka olivat aktiivisia ja vastasivat beacon-viestiin. Haittana aika-ajo mekanismissa on aikaviive johtuen ajasta, joka kuluu topologiaetsinnän ja topologiamuutoksen välillä. Tätä haittaa voidaan pienentää toisella mekanismeista eli tieto-ajo mekanismilla. Tieto-ajo mekanismissa solmu huomioi, että naapurisolmu on saavuttamattomissa, solmu täydentää beacon-viestiin kadonneen solmun ositteen ja tiedottaa topologiamuutoksesta naapurisolmuille ja päivittää omat taulutietonsa. Naapurisolmut päivittävät myös omat taulunsa saatuaan beacon-viestin. Saavuttamattomissa oleva solmu voidaan löytää käyttämällä MAC (Access Layer Control) protokollaa tai käyttämällä tietoa aikakatkaisun jälkeen kuittaamatta jääneistä beacon-viesteistä. Solmujen reittimuisti päivitetään tietojen muutosten perusteella naapuritauluista. Reittimuisti sisältää kaikkien solmujen reititystiedot ja tiedot päivitetään monitoroimalla pakettilähetykset läpi koko verkon. Määrittääkseen yksityiset reitit, reitin määrittelysyy (route extraction reason) mekanismia käytetään priorisoimaan erilaiset paketit, joita verkossa liikkuu. Esimerkiksi reitit, joilla lähetetään reittien löytymispaketit, priorisoidaan korkeammalla arvolla kuin reittikyselypaketit tai naapuritaulupäivitykset. Reitit, joilla liikkuu datapaketit, on matalin prioriteetti. Näitä prioriteetteja voidaan käyttää apuna

reittejä valittaessa. Reittietsintä ja reittien huolto toimivat samoin kuin muillakin algoritmeilla. (Yao, et al., 2003), (Boukerche, 2011), (Tarique, et al., 2009).

Lisää luotettavista reititysprotokollista voi lukea, vaikka lähteistä (Boukerche, 2011) ja (Tarique, et al., 2009).

## 4.3 REEF -mekanismi

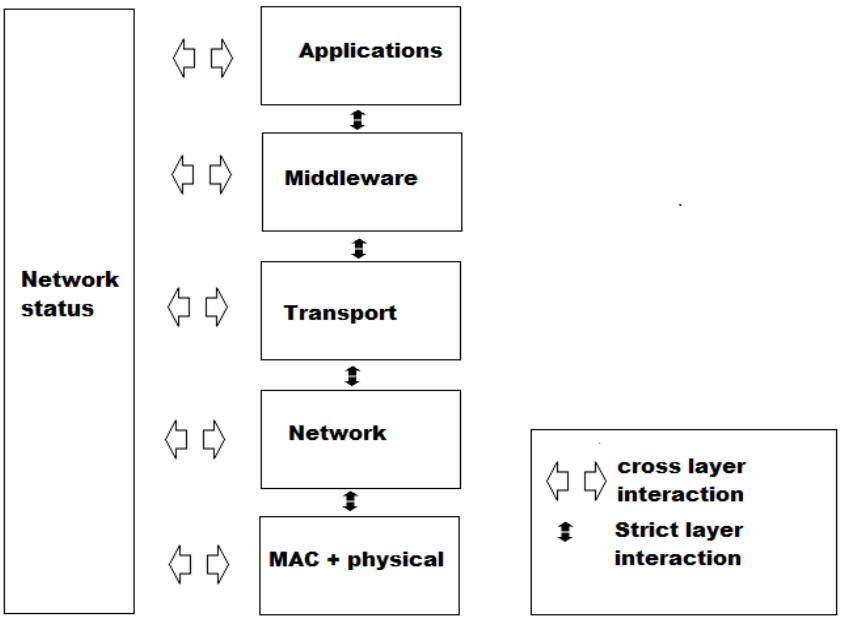
Marco Conti kumppaneineen (Conti, et al., 2006) esittelee artikkelissaan pakettikytkentäisen ad hoc -verkkoon uuden lähestymistavan parantaa solmujen kommunikointia keskenään. Tämä on REEF (REliable and Efficient Forwarding) mekanismi, jolla pyritään vähentämään haitallisia tilanteita verkossa solmujen välillä, kuten haitallisia solmuja, jotka eivät pysty yhteistyöhön toisten solmujen kanssa tai muita verkkoon liittyviä virhetilanteita. Tämä käsittelee solmujen paikallista tietämystä reitin luotettavuudesta ja pakettien reitittämisestä eteenpäin luotettavinta mahdollista reittiä pitkin. REEF on kehitetty monipolkuisille protokollille. Se on saatavilla sekä proaktiivisille eli ennakoiville protokollille esimerkiksi OLSR (Optimized Link State Protocol), että reaktiivisille eli vastavaikutteisille protokollille esimerkiksi AODV (Ad hoc on-demand distance vector) ja DSR (Dynamic source routing). REEF luo myös salausmekanismin kahden osapuolen välille, silloin kun on tarve lähettää tieto salattuna. Tämä lisätty ominaisuus REEF-mekanismiin takaa, että lähetys on vakaa, luotettava ja tarvittaessa myös salattu. Metodi kieltää palvelun epäluotettavilta solmuilta, esimerkiksi kieltämällä eteenpäin kyselyt. Ongelmaa käsitellään siten, että tarkkaillaan solmujen palvelun laatua seuraamalla solmujen käytöstä. Toisin sanoen epäluotettavat solmut hoitavat verkkoliikennettä hitaammin kuin luotettavat solmut. Vakiinnuttaminen, huolto ja erilaiset verkon tehtävät pitää hoitaa yhteistyössä verkon solmujen välillä. Solmujen pitää kommunikoimalla toistensa kanssa, ylläpitää ja löytää reittejä. Solmun itsekäs tai epäsopiva käytös aiheuttaa verkon toimintaan ongelmia. Tällaisia ongelmia ovat esimerkiksi linkin tai solmun ruuhkautuminen, joka johtuu usein protokollasta tai verkon topologiasta. Toiset solmut voivat joutua välittämään paketteja enemmän kuin toiset solmut. Toiset verkon alueet ovat taipuvaisia ruuhkaan. Toinen ongelma on yhteyden katkeaminen, syynä verkon dynaaminen topologia, solmujen liikkuvuus tai yhtä hyvin verkon vaihteleva yhdistettävyys. Kolmantena voidaan mainita linkin katoaminen, kapasiteetin vaihtelun seurauksena linkin yhteys katkeaa.

Ongelmat polulla johtuvat usein solmun ilkeästä tai itsekkäästä käytöksestä sekä kontrolloimattomasta tilanteesta, kuten ruuhka tai ruuhkautunut linkki. Ilkeä tai itsekäs käyttäytyminen tarkoittaa, solmu ei toimi yhteistyössä muiden solmujen kanssa toisten solmujen haluamalla tavalla eli jonkin asteista kommunikointiongelmaa. Tämä voidaan ratkaista vain solmujen yhteistyöllä ja jokainen solmu luottaa vain itseensä. Jokainen

solmu toimii itsenäisesti, jakamatta tietoa muille solmuille ja luottaa vain tietoon, mitä tulee kommunikoinnista muilta solmuilta. Jokainen solmu päivittää dynaamisesti luotettavuustaulua sisältäen arvot jokaiselle linkille naapurustoon. Jokainen arvo on uniikki jokaiselle naapurisolmulle ja sisältää luotettavuusindeksin polulle, joka kulkee naapurisolmun kautta. Kun solmu lähettää paketin polulle, se odottaa kuittausta(ACK) paketin perille menosta kohdesolmulta ja kuittauksen jälkeen päivittää naapurin luotettavuusindeksin. Mikäli paketti toimitetaan onnistuneesti perille, päivitys on positiivinen, muuten negatiivinen. REEF luottaa TCP ACK kuittauksiin. Tämä sallii REEF-mekanismin päätellä toisten solmujen käytöksestä viittaamalla vain paikalliseen tietoon ja mekanismi on yksinkertainen ja kevyt. Tällä ei pystytä ratkaisemaan turvallisuusongelmaa liittyen ilkeisiin käyttäjiin, mutta niiden läsnäoloa voidaan hankaloittaa. Viestit voidaan salata lähettäjän ja vastaanottajan välillä salausavaimilla, joka käytiin läpi aiemmin. REEF ei vaadi kryptografiaoperaatioita välittäjäsolmuilla polulla. Paketit lähetetään sellaisina kuin ovat ilman kryptausta. Sen sijaan ne kantavat message authentication code (MAC), mikä sallii kohteen varmistaa viestin yhtenäisyys ja aitous. Salausjärjestelmän etu on kaksijakoinen. Ensinnä se tarjoaa luotettavuuden TCP pakettikuittauksiin. Kuten luotettavuus arvioidaan pakettitoimituksien onnistumisen tuloksena, mekanismi on perusta sen toimimiseen kunnolla. Salausjärjestelmän käytön lisäetu on, että viestit salataan viestiväärennöksiä vastaan. Tämä todistaa, että sen suorituskyvyn ja luotettavuuden lisäksi sillä on myös hyvä turvallisuuden taso myös pakettien lähettämisessä. (Conti, et al., 2006).

### 4.3.1 REEF -arkkitehtuuri

REEF -mekanismi sijoitetaan samalle verkon kerrokselle ja hyödynnetään reititystietoa sekä tuotetaan samalla kerroksella kuin kuljetuskerroksen pakettikuittaukset (TCP/ACK). Tehokkaaseen kuljetuskerroksen kuittauksien parantamiseen on kehitetty protokollapinoarkkitehtuuri ad hoc -verkkoon, niin kutsuttu cross-layer (poikittaiskerros). Cross- layer on kehittynyt protokollan vuorovaikutusmuoto, joka on sijoitettu muiden kerrosten viereen ja jolla voidaan mahdollistaa useiden eri kerroksien väliset kommunikoinnit. Ideana on, että voidaan tehokkaasti käyttää hyväksi jokaisen protokollan keräämää tietoa eri tehtävistä eri kerroksien välillä. REEF arkkitehtuurissa on moduuli, jota kutsutaan verkon statukseksi (Network Status, NeSt), joka kontrolloi kaikkia cross-layerin tehtäviä yhdistämällä ja tiivistämällä vertikaalista kommunikointia. Kuvassa alapuolella (Kuva 8) selvennetään cross-layerin tehtävää ja sen arkkitehtuuria ad hoc -verkossa. (Srivastava & Motani, 2005).




**Kuva 8**. Cross-layer arkkitehtuuri ad hoc -verkon solmulle.

Network status tukee cross- layeria toteuttamalla kahdenlaista vuorovaikutusta kahden protokollan välillä. Synkronista (synchronous), missä protokollat jakavat sisäistä tietoa ja epäsynkronista (asynchronous), missä protokollat allekirjoittavat ja kirjaavat tapahtumia. Protokollat ovat vuorovaikutuksessa synkronisesti, kun ne jakavat yksityistä tietoa. Yksityisellä tiedolla tarkoitetaan pyyntöä, joka tapahtuu tilauksesta pyytää toisilta protokollilta tiedusteluja verkkostatuksen välityksellä toisilla kerroksilla tuotetusta tiedosta ja tuloksen odottamista. Epäsynkroninen vuorovaikutus liittyy erityisolosuhteiden tapahtumiin, mihin protokollat voivat reagoida. Tällaiset tilanteet ovat satunnaisia, missä protokollia vaaditaan allekirjoittamaan tapahtumat. Eli protokollat allekirjoittavat tapahtumat, sen jälkeen palauttavat tuloksen muille solmuille. Verkon status on puolestaan vastuussa toimittamaan lopulliset tapahtumat oikeille allekirjoittajille. Poikittaiskerros on vuorovaikutuksessa kuljetus ja -verkkokerroksen välillä. Erityisesti TCP voi ilmoittaa pakettikuittaukset verkkokerrokselle verkko -statuksen kautta. Eteenpäin lähettävä protokolla voi päivittää suhteellisen luotettavuusindeksin. Tässä tapauksessa verkkokerros voi helposti ymmärtää pakettitoimituksen tilanteen ilman ylimääräistä kustannusta. (Conti, et al., 2004), (Srivastava & Motani, 2005).

### 4.3.2 REEF reittien luotettavuuden arviointi

Solmujen maineeseen perustuva mekanismi arvioi solmujen luotettavuuden käyttämällä paikallista tietoa. Joka solmu toimii itsenäisesti jakamatta minkäänlaista tietoa toisille solmuille ja luottaa vain tietoon, joka tulee toisilta kommunikointilähteiltä ACK pakettien mukana. Joka solmulla on luotettavuusindeksi jokaiselle naapurisolmulle, jonka kautta se lähettää paketteja. Tämä indeksi määrittää yhteistyön, suorituskyvyn ja luotettavuuden niin kauan kuin solmulla on polku selvillä tarkkailtavaan naapuriin. Kuten aiemmin kerrottiin, joka kerta kun paketti toimitetaan eteenpäin polulle, niin tuloksen saatuaan paketin perille menosta, solmu päivittää luotettavuusindeksin arvon naapurisolmulle. Poikittaiskerroksen arkkitehtuuri sallii verkkokerroksen pääsyn TCP -kyselyiden tietoihin saadakseen tietoa pakettitoimituksien tuloksista. Jos solmu ei saa päivitystietoja, se voi johtua solmujen epäluotettavuudesta yhtä hyvin kuin ruuhkasta tai kadotetusta linkistä. Jos lähdesolmu havaitsee, että alipuun luotettavuusarvo laskee, vähentää se välittömästi liikennettä naapurin alipuun juuren kautta ja valitsee uuden naapurin, jolla on parempi reitti korkeammalla luotettavuusindeksillä. Tämä tapahtuma toistetaan koko polun matkalla lähdesolmun ja kohdesolmun välillä. Jokainen välittäjäsolmu valitsee seuraavan hypyn käyttäen paikallisesti arvioitua luotettavuusarvoa. Yksinkertaisin tapa arvioidakseen solmujen luotettavuutta vaaditaan joka solmua pitämään todennäköisyysindeksi jokaiselle naapurille, kuvaamaan linkin luotettavuustaso itsensä ja lähimmän naapurin välillä. (Conti, et al., 2006).

### 4.3.3 Eteenpäin lähettämisen käytäntö

Jos saatavilla on useampia reittivaihtoehtoja kohdesolmulle, lähdesolmu voi valita niistä yhden varmimman vaihtoehdon. Tyypillinen ad hoc käytäntö valita lyhyin tai tuorein polku, joka on enemmän käytetty kuin toiset polut. Sellaisille kriteereille haitta on kaksinkertainen. Verkon jotkut alueet ovat alttiimpia raskaan liikenteen kuormille, toiseksi reitin luotettavuus kärsii. Toisaalta käytäntö ottaa huomioon luotettavuutta parantaa verkon suorituskykyä. Monipolkuisella reititysprotokollalla on vaihtoehtoisia reittejä kohteelle. Seuraavaksi selvitetään, miten reitit valitaan.

Ensiksi lähdesolmu valitsee aina luotettavimman reitin. Lähdesolmu vertaa luotettavuusarvoja saatavilla olevista reiteistä ja lähettää paketit linkkiin, jolla on korkeimmat luotettavuusarvot. Jos korkein luotettavuusarvo on useammalla kuin yhdellä reitillä, valinta tehdään yksinkertaisella laskutoimituksella, jonka reititysprotokollan algoritmi tekee. Tämä käytäntö varmistaa, että lähdesolmu ottaa aina luotettavimman reitin. Päähaitta tällaisesta valinnasta on kaiken liikenteen poikkeama luotettavimmalla linkillä, mikä voi aiheuttaa liikennevirrassa ruuhkaa.

Toinen käytäntö vertaa saatavilla olevien reittien luotettavuusarvoja rakentaakseen todennäköisyyskaavan. Käytäntö kertoo luotettavuusarvot niin, että tuloksena on todennäköisyysarvo, joka kuvastaa linkin luotettavuustason. Reitit valitaan myöntämällä todennäköisyysarvo liittyvällä ensimmäiseen solmuun polulla. Suurin todennäköisyys, korkein reittivalinta taajuus. Tämä todennäköisyyteen perustuva valinta sallii solmujen ottavan jopa vähemmän luotettavia reittejä. Liikenteen eteenpäin lähettävä funktio on parempi kaikilla yleisillä saatavilla reiteillä ja linkkien ruuhkat tulevat harvinaisemmiksi. (Conti, et al., 2005).

### 4.3.4 Yhteistyön valvonta

Yhteistyöllä voidaan parantaa huonon käytöksen vaikutuksia saavuttamaan paremman suorituskyvyn. Luotettavia reittejä valitaan useimmin kuin vähemmän luotettavia. Siten REEF ei vapauta huonosti käyttäytyviä solmuja pakettien lähettämisestä. Sen sijaan se yrittää tasapainottaa verkon käyttöä lähettämällä paketteja myös vähemmän luotettavia reittejä pitkin. REEF voidaan yhdistää helposti yhteistyönvalvontamekanismiin. (Conti, et al., 2006).

### 4.3.5 Itsekkyysmalli

Itsekkyys on uusi ongelma ad hoc -verkkojen ympäristössä ja pidetään epäluotettavana käytöksenä. Itsekkyysmalli on määritelty reititys ja eteenpäin lähetysfunktioille. Tässä mallissa solmujen käytös riippuu energiatasoista. Solmu, jonka energiataso laskee alas, uudelleen määritelty alaraja voi lopettaa lähettämisen eteenpäin toisten paketteja ja silti osallistua reititysprotokollan tehtäviin. Kun energiataso laskee ja putoaa alas matalimmalle alarajalle, se voi myös vaurioittaa reititysfunktiota niin, ettei näyttäydy toisten solmujen reititystauluilla ja siitä johtuen sitä ei vaadita lähettämään paketteja eteenpäin. Se käyttää energiansa vain omaan kommunikointiin. Muut solmut pitävät sitä jonkin sortin ilkeytenä. Solmujen pitää toteuttaa protokollamuutoksia tai muuttaa joitakin toimeenpanoparametreja, muutoin paketit täytyy hylätä. Tulos ei ole protokollan määrittelylle johdonmukainen, koska paketteja ei lähetetä eteenpäin.

Solmun käyttäytymiseen liittyy kaksi mahdollista tilaa: Aktiivinen ja lepotila. Niin kauan kuin solmun tarvitsee kommunikoida toisten solmujen kanssa, se on aktiivinen tekemällä yhteistyötä verkon kanssa. Kun taas se ei tarvitse verkon resursseja, se menee lepotilaan ja sen jälkeen ei ole käytettävissä muille solmuille. Solmun itsekäs käytös on enemmän yleistä, koska se huomioi kaikki verkkotoiminnot samaan aikaan ja se on käytettävissä jokaiselle käyttäjälle, kunnes katkaisee yhteyden verkkoon. Kun solmu menee

lepotilaan, se katkaisee yhteyden verkkoon ja automaattisesti kytkee pois päältä verkkotehtävät kuten reititys ja eteenpäin lähettäminen. Voidaan sanoa, että täysi itsekkyys voi raskaasti heikentää verkon suorituskykyä ja vahtikoirapohjaiset mekanismit ovat kyvyttömiä havaitsemaan tällaista käytöstä. Yksityinen solmu voi omaksua yhden seuraavista tiloista:

- Aktiviinen (active). Solmu tekee yhteistyötä koko verkon kanssa.
- Eteenpäin lähetys pois päältä (forwarding off). Solmu kytkee pois päältä eteenpäin lähetystoiminnot säästääkseen resursseja omiin toimintoihin.
- Reititys pois päältä (routing off). Solmu kytkee pois päältä reititystoiminnon.
- Pois päältä (off). Solmu valitsee täysin itsekkään käytöksen. Se kytkee pois päältä kaikki verkon toiminnot katkaisemalla yhteyden verkkoon.

Solmu voi halutessaan muuttaa tilasta toiseen. Tämä siksi, että solmu voi aina päättää omasta käyttäytymisestä. Solmun käytökseen voi olla syynä itsekkyystason kasvaminen saatavilla olevan energian vähetessä. Kuten aikaisemmin korostettiin mahdollisuus solmulle toimia itsekkäästi reitityksessä ja eteenpäin lähettämisessä merkitsee jonkin sortin ilkeyttä käyttäjältä, kun hänen täytyy tehdä protokollaan muutoksia tai muuttaa joitakin toteutuksen parametreja, jotta paketteja ei katoaisi. Tätä pidetään itsekkäänä käytöksenä. Raja itsekkyyden ja ilkeyden välillä ei ole kovin suuri. (Conti, et al., 2006).

### 4.3.6 Priorisointi pakettien eteenpäin lähettämiselle

Solmujen yhteistyötä parannetaan priorisoimalla solmulle tulevat pakettilähetykset. Ideana on erotella palvelun laatu kohti toisia solmuja, tarjota etuoikeutettu tai epäsuotuisa palvelu, myöntämällä tapa millä solmut kohtelevat toisiaan. Tämä tehdään nostamalla tai rajoittamalla liikennevirran prioriteettia, jonoilla ja palvelujonoilla erilaisilla tavoilla. Solmut voivat käyttää luotettavuusindeksiä luokittelemaan tuleva liikenne ja hidastamaan pakettien tuleminen vähemmän luotettavilta solmuilta. Paketit, jotka tulevat luotettavalta naapurilta lähetään eteenpäin korkeammalla prioriteetilla kuin paketit, jotka tulevat matalamman prioriteetin omaavalta naapurilta.

Tämä mekanismi perustuu prioriteettijonotekniikkaan. Sitä käytetään luotettavuusarvon arviointiin lähteeltä tuleviin paketteihin. Järjestelijä valitsee ulosmenevistä pakettien seulonnasta jonot korkeimmasta matalimpaan prioriteettiin poimimalla, ensin saatavilla olevat paketit. Tämä mekanismi voidaan jakaa esimerkiksi neljään jonoon asettamalla prioriteettitasot korkea, keski, normaali ja matala. Tulevat paketit luokitellaan myöntämällä prioriteettitaso arvioituna naapurin luotettavuusindeksistä, joka lähettää paketin ja ohjataan se luotettavuusarvon kuuluvaan jonoon. Välityksen aikana algoritmi antaa korkeamman prioriteetin jonolle ehdottoman etuoikeutetun kohtelun ohi matalamman prioriteetin jonojen. Tämä järjestelijämekanismi tulee liikennevirtaus perustuvasta järjestel-

mästä, jota kutsutaan Generalized Processor Sharing (GPS) eli yleinen suoritin jakaminen, mikä on hyvin vakiintunut tekniikka toteuttamaan palvelumäärittelyä. Siinä liikennevirta jaetaan luokkiin, ja jokaisella luokalle on annettu positiivinen paino, joka määrittää taatun minimikapasiteetin luokalle. Jos tietty luokka ei voi täysin käyttää kapasiteettiaan, sitten liika kapasiteetti tulee saataville toisille luokille yllättävästi tämän solmun kautta minimiluokan yläpuolelle. Koska jonot ovat rajallisen kokoisia, ne voivat täyttyä ja ylikuormittua. Sen johdosta mekanismi on tarpeellinen välttääkseen korkea prioriteetin jonojen täyttymistä, koska pienempi prioriteetin jonoissa on tilaa. Lopuksi paketit voivat olla sijoittuneina prioriteettijonoissa, mitkä sisältävät vaihtelevan pituisia alijonoja ja sallivat prioriteettiin perustuvan lisäämisen pitämällä FIFO (First In First Out) järjestyksen. Tässä tapauksessa korkeamman prioriteetin alijonot voivat kasvaa yli maksimikoon ja lopulta pudottamalla matalamman prioriteetin paketteja. Eteenpäin lähetettäville tuleville paketeille myönnettävä priorisoitu jonotustekniikka tarkoittaa, että lähettäjä antaa palvelulaadun, mikä on verrannollinen sen luotettavuustasoon. Itse asiassa solmun käytöksen ja luotettavuusarvon välillä on vahva suhde. Jos arvioitu luotettavuus naapurille on matala, se tarkoittaa melko varmasti, että käytös on huonoa. Tästä johtuen solmun matalampi luotettavuus johtaa matalampaan liikennevirtaan koko verkon alueella. Uudet solmut, jotka liittyvät verkkoon, saavat prioriteettiarvon, joka on arvolta keskitaso (medium). Tämä antaa etuoikeutetun kohtelun solmuille, jotka kuuluvat jo verkkoon ja tekevät yhteistyötä sen toiminnoilla. On järkevää käynnistää uuden solmun luotettavuusarvo keskitasolta. Jos uusi solmu tekee yhteistyötä toisten solmujen kanssa, niin sen luotettavuus ja siitä johtuen luotettavuusarvo kasvaa hyvän käytöksen johdosta. Tämä valinta motivoidaan tarpeella välttää tosi itsekkään solmun kadota koko verkosta ja sitten liittyäkseen verkkoon uutena solmuna. Tämä käyttöönottokäytäntö näyttää rankaisevan solmuja, jotka ovat verkon rajalla ja joilta ei vaadita eteenpäin lähettämisliikennettä toisten solmujen eduksi. On mahdollista hankkia uudelleenyhdistymismekanismi, joka säännöllisesti tarkistaa reititystaulun ja päivittää solmujen luotettavuuden, jotka ovat naapureita, mutta eivät näyttäydy seuraavana hyppynä kohti toisia solmuja. Tämä mekanismi perustuu topologiainformaatioon, joka päätellään reititystaulusta. Tapauksessa, jossa energiataso on niukka, eteenpäin lähettävä solmu voi päättää pudottaa matalamman prioriteetin paketit ja lähettää eteenpäin vain liikenneluokan esimerkiksi korkea- ja keskitason prioriteetin omaavia paketteja. Tässä tapauksessa solmu on kykenemätön täydelliseen yhteistyöhön välttääkseen olematta pois verkosta, koska se varmistaa palvelun hyvin käyttäytyville solmuille, mikä puolestaan haluaa jatkaa ja palvella sen pyyntöjä. (Conti, et al., 2006).

### 4.3.7 Epäluotettavien solmujen paikantaminen

REEF -mekanismi pyrkii löytämään solmut, jotka eivät toimita paketteja eteenpäin jostain syystä. Syynä voi olla ruuhka, solmun itsenäinen käytös tai vaikkapa solmun virran vähenemisestä johtuva toimintatilan vaihto. REEF voi olla laajennettu niin, että se voi

arvioida etäisyyden huonosti käyttäytyvän solmun ja lähdesolmun välillä. Jos epäluotettava solmu havaitaan tai alue, jolla solmu sijaitsee, voi hyvin käyttäytyvä naapurisolmu ensin hidastaa pakettien lähettämistä ja katsoa onko ongelma väliaikainen. Jokainen solmu pitää jokaiselle naapurisolmulle määrättyä luotettavuusindeksiä. Jos eteenpäin lähettäminen onnistuu, luotettavuusindeksi kasvaa. Jos paketteja häviää, luotettavuusindeksi pienenee. Luotettavuusindeksi saa arvon onnistunut tai epäonnistunut. Jokainen indeksi arvioi polun luotettavuuden määritellyllä pituudella. Jokaisella solmulla on reititystaulussa reittejä seuraaville solmuille, naapurisolmujen luotettavuusindeksi sekä naapurisolmujen seuraavien hyppyjen luotettavuusindeksit ja pituudet. Näin lähdesolmu saa joka reitille luotettavuusindeksin lähteeltä kohteelle. Pitääkseen matalan läpimenoajan, luotettavuusindeksille lasketaan maksimietäisyys. Jokainen solmu valitsee luotettavimman ja lyhyimmän reitin kohdesolmulle. Jatkuva taulujen päivittäminen pitää tiedot solmujen tiloista ja käyttäytymisestä ajan tasalla. Naapurisolmu kuittaa saapuneet paketit ja toimittaa ne eteenpäin. Jos naapurisolmu ei pysty toimittaman paketteja eteenpäin, paketteja katoaa tai ei tule kuittauksia pakettien saapumisesta ja uudelleen lähettämisestä huolimatta, niin lähettäjäsolmu hylkää reitin ja valitsee seuraavaksi luotettavimman reitin. Metodi verrata luotettavuusindeksejä tulee tarpeelliseksi ja käytännölliseksi jokaiselle solmulle. Mikäli solmu ei kommunikoi toisten solmujen kanssa ja sen luotettavuus alenee. Solmulla kestää pienen aikaa saada luotettavuus takaisin. Voit lukea tarkemmin, miten huonosti käyttäytyvä solmu paikallistetaan lähteestä (Conti, et al., 2006, pp. 411 - 413).

## 4.4 PIDIS -mekanismi

PIDIS (Protocol-Independent Packet Delivery Improvement Service) on mekanismi, joka on tarkoitettu parantamaan monilähetyspakettien toimitusta mobiili ad hoc -verkoissa ja palauttamaan kadonneita monilähetyspaketteja. PIDIS tarjoaa mille tahansa monilähetysreititysprotokollalle pakettientoimitukseen parantuneita palveluita manet-verkossa hyödyntämällä parviälymekanismia tekemään älykkäitä päätöksiä saadakseen takaisin kadotettuja monilähetyspaketteja. PIDIS on juoruprotokolla ja solmut käyttävät sitä, kun juoruavat naapurisolmujen kanssa saadakseen takaisin eniten kadonneita paketteja sen sijaan, että juoruisivat jäsensolmujen kanssa. Se ei ole riippuvainen jäsenyystiedosta monilähetys toiminnoissa. PIDIS työllistää todennäköisyysperäisiä reitityksiä ja mukautuu hyvin liikkuvuuteen. PIDIS saavuttaa todennäköisyyspohjaista parannusta monipakettitoimituksiin ja toisin kuin muut juoruperusteiset järjestelmät, sen ei tarvitse ylläpitää tietoa ryhmän jäsenistä, joille monilähetyspaketit on palautettu. Lisäksi PIDIS-toiminto ei ole riippuvainen millekään reititysprotokollalle ja se voi olla liitettynä mihin tahansa ad hoc monilähetysreititysprotokollaan.

Ad hoc -verkko koostuu mobiilisolmuista, jotka itsenäisesti muodostavat yhteyden langattomalla monihyppykommunikoinnilla. Monissa sovelluksissa solmujen pitää tehdä

yhteistyötä saavuttaakseen yhteiset tavoitteet ja toimia ryhmänä pikemminkin kuin yksittäisenä parina. Monilähetyskommunikointi tukee pisteestä pisteeseen (point to point), pisteestä monipisteeseen (point to multipoint) ja monipisteestä monipisteeseen (multipoint to multipoint) sovelluksia. Useat toiminnalliset rajoitukset kuten solmujen liikkuvuus, rajallinen virransaanti, vaihteleva laitteen muistikoko, langattoman verkon kaistanleveys, häirintä ja ympäristötekijöiden häiriöt rajoittavat luotettavaa pakettien toimitusta perille. Tästä tuloksena erilaisien jäsensolmujen vaihteleva pakettien määrä. Vaikka useat monilähetysreititysprotokollat on tarkoitettu ja suunniteltu mobiili ad hoc -verkkoon, silti pakettitoimitus on niille erittäin haasteellista. Ad hoc -verkkoon on luotu luotettava monilähetysjärjestelmä ja se perustuu joko positiiviseen tai negatiiviseen kuittaukseen (ACK/NACK) tai sujuvaan tulvimiseen. Kuljetuskerros on lähestymistapa luotettavuusongelmaan monilähetys mobiili ad hoc -verkoissa. Kuljetuskerrosprotokolla luo korkean luotettavuuden käyttämällä kuljetuskerrosmekanismeja päästä päähän lähetyksissä. Äskettäin on kaksi juoruiluun perustuvaa sovellusta kehitetty helpottamaan luotettavan monilähetyksen käsitettä ad hoc -verkoille. Anonymous Gossip (AG) luo luotettavan kehittyneen palvelun, joka toimii epäluotettavan monilähetysprotokollan päällä ja Route-Driven Gossip (RDG) on luotettava protokolla. Kun toiset sovellukset tasapainoilevat luotettavuuden ja skaalautuvuuden välillä, juorupohjaiset sovellukset hyödyntävät manet-verkon luonnetta tarjoamaan todennäköisyyspohjaisen luotettavuuden skaalautuvalla tavalla. PIDIS ei ole luotettava monilähetysreititysprotokolla vaan palvelu, joka käyttää parviälymekanismia päättämään palautettavat kadotetut paketit. Parviäly viittaa monimutkaisiin käyttäytymisiin, jotka syntyvät yksinkertaisista ja yksittäisistä käyttäytymisistä ja vuorovaikutuksista, joita on usein havaittu hyönteisten keskuudessa, esimerkiksi muurahaisten keskuudessa. Vaikka jokainen yksittäinen yksilö ei ole kovin älykäs, seuraamalla perussääntöjä ja käyttämällä paikallista tietoa tarkkailemalla ympäristöä ja optimoimalla käytöstä syntyy tulosta, kun ne toimivat yhdessä ryhmänä. Parviäly (Swarm intelligence, SI) koostuu kolmesta komponentista.

- Positiivinen ja negatiivinen palaute, joka etsii hyvät ratkaisut ja vakauttaa tulokset.
- Vaihtelun vahvistaminen, joka etsii uusia ratkaisuja ja sopeutuu ympäristön muutoksiin.
- Monivuorovaikutus, joka sallii yleiset kokonaisuudet koordinoimaan ja järjestäytymään itse.

Yhdessä nämä komponentit muodostavat joustavan etsintämekanismin, jolla PIDIS lähestyy nopeasti hyviä ehdokasreittejä, minkä läpi voitiin ottaa talteen kadotettuja monilähetyspaketteja suurimmalla todennäköisyydellä. Tämän jälkeen vaihtoehtoiset reitit pakettien palautumisesta yhdistetään muuttuvaan pakettitoimituskaavioon ja verkon topologiaan. PIDIS on juoruprotokolla ja se on mukautuvainen verkon toimintaan ja voi juoruta monta kertaa kadonneista paketeista. PIDIS mittaa jatkuvasti verkon tilaa kontrolloidakseen juoruilun laajuutta ja lähetettyjen juorujen määrää kadonneista paketeista. PIDIS ei ole riippuvainen jäsenien näkökohdista. PIDIS oppii, mikä seuraavan hypyn naapurisolmu antaa paremman pakettien palautuksen suhdeluvun, jonka kanssa se on

juoruillut, pikemmin kuin oppimalla, mitkä jäsensolmut auttavat palauttamaan eniten paketteja. PIDIS rajoittaa juorupakettien määrää valitsemalla keskitetyn joukon seuraavan hypyn solmuja juorukumppaneiksi. Juoruviestit luodaan käyttämällä arvokasta tietoa juorujen keräilyn aikana juorukyselystä. Kun juorut on käsitelty, juoru palaa takaisin juoruvastauksena.

### 4.4.1 PIDIS katsaus

PIDIS on tehokas pakettipalauttajaprotokolla. PIDIS voi tehdä pakettien palauttamista useammin kuin vain kerran ja pakettien palauttamisen määrä voi olla rajoitettu. PIDIS monilähetysreititys mekanismi toimii näin:

1. (Epäluotettava) Monilähetysreititysprotokolla toimittaa paketteja solmulle.
2. PIDIS palvelu ”kehottaa” solmua hakemaan paketit, joita lähde ei pystynyt lähettämään.

PIDIS tarjoaa palvelun monilähetysprotokollalle välittömästi verkkokerroksella.

Kun monilähetyslähde lähettää tietoa ensimmäisen kerran, se lähettää naapureilleen liittymiskyselypaketin. Se on tietopaketti, jossa on kysely ”lippu asetettu”. Jokainen solmu kopio paketin ja tallentaa ylävirran solmun tunnuksen reititystauluun ja lähettää paketin uudelleen eteenpäin. Kun monilähetysryhmän jäsen saa liittymiskyselyn, se lähettää jokaiselle naapurisolmulle liittymisvastauspaketin, joka sisältää lähteen tunnuksen ja ylävirran solmun tunnuksen. Saatuaan liittymisvastauksen solmu, jonka tunnus vastaa ylävirran tunnusta liittymisvastauspaketissa ymmärtää, että se on polulla lähteen ja jäsenen välillä. Solmusta tulee välittäjäsolmu ryhmään asettamalla ryhmälippu merkin. Tämä solmu muodostaa ja lähettää oman liittymisvastauksen käyttämällä vastaavan ylävirran solmun tunnusta. Liittymisvastauspaketit, jotka sisältävät kaikkien jäsenien tiedot lähetetään takaisin lähteelle palautuspolulla. Kun lähde on lähettänyt liittymiskyselyn yhden kerran, seuraavilla kerroilla se ei enää lähetä ”lippu asetettu” kyselyä. Tämä mahdollistaa uudelleen lähetykset vain ryhmän solmuille, mikä vähentää eteenpäin lähetyksen kuormitusta verkossa. Jos jäsensolmu ei päivitä tietoja kyselyillä sovitulla aikaväleillä se putoaa ryhmästä pois. PIDIS toimii seuraavalla tavalla. Kun paketit kuuluvat lähde/ryhmä pariin ja kadotetaan jäsensolmulla.

1. Solmu lähettää juorukyselypaketin (Gossip Request Packet, GREQ) saadakseen takaisin kadotetut paketit. Kysely lähetetään valittuun yhden hypyn kohteelle, juoru seuraava hyppy toteutetaan algoritmilla.
2. Jokainen välittäjäsolmu, joka vastaanottaa kyselyn (GREQ) toimii seuraavasti.
   (i) Välittäjäsolmun tunnus kirjataan kyselyyn.
   (ii) Jos välittäjäsolmu ei ole jäsen, lähde tai jokin eteenpäin lähettäjäryhmän solmu, kysely hylätään.

(iii) Jos välittäjäsolmu on eteenpäin lähettäjäryhmän solmu, mutta ei jäsensolmu eikä lähdesolmu, välittäjäsolmu lähettää eteenpäin kyselyn äsken kohdassa 1 valitulle kohteelle.

(iv) Jos välittäjäsolmu on lähdesolmu tai jäsensolmu, välittäjäsolmu tarkistaa onko sillä kadonneita paketteja. Jos välittäjäsolmulla ei ole kadonneita paketteja, se lähettää kyselyn eteenpäin.

(v) Jos välittäjäsolmu on lähdesolmu tai jäsensolmu, jos sillä paketteja, jotka on kadonneet lähdesolmulta, niin se palauttaa paketit muististansa. Sen jälkeen se lähettää juoruvastauspaketin (Gossip Reply Packet, GREP) jokaisesta kadotetusta paketista. Jokainen juoruvastaus palautetaan juorukyselyn tullutta reittiä takaisin lähdesolmulle. Sen jälkeen kysely hävitetään.

3. Jokainen solmu, jonka kautta juoruvastaus kulkee reitillä takaisin lähdesolmulle, kirjaa ylös mistä kadonneet paketit tulivat ja mihin lähde/ryhmä pariin kuuluivat. Näillä tiedolla päivitetään juorutaulua. Tätä tietoa käytetään myös siten, että tiedetään ensimmäisten hyppyjen hyödylliset solmut, jotka palauttivat juoruvastauksia onnistuneesti.
4. Jos juoruvastauspaketit eivät saavu perille lähdesolmulle ennen aikakatkaisua, lähdesolmu voi aloittaa uudelleen juorukyselyn.

PIDIS- parviälymekanismi hyödyntää juorupakettilähetyksiä kerätäkseen tietoja verkon solmuista. GREQ ja GREP keräävät tieto solmuista, joiden kautta ne kulkevat koko verkon alueella. Ne myös valitsevat reitit, mistä kadonneet paketit voidaan palauttaa. Juoruvastauspaketti kerää tietoja niistä solmuista, joiden kautta se palaa lähdesolmulle. Se palaa samaa reittiä, millä juorukyselyt lähetettiin. Näin se hallitsee verkon kuormitusta.

### 4.4.2 PIDIS päivitystaulut

Juorutaulu sisältää tiedot, joita kerätään aktiivisesti ja päivitetään joko jäsensolmuilta tai eteenpäin lähettäjäsolmuilta. Juorutaulun tiedonkäyttö ja muoto tulee päätöstaulusta ja algoritmeista. Solmut saavat päivitystiedot juoruvastauspaketeista. Juorutaulun tietoa käytetään valitsemaan seuraava hyppy, kun lähetetään juorukyselypaketti etsimään kadonneita paketteja. Tauluun tallennetaan mahdolliset seuraavat hypyt jokaiselle monilähetys lähde/ryhmä parille. Se laskee myös algoritmilla parhaan mahdollisen seuraavan hypyn käyttäen apuna juorutauluun kerättyä tietoa.

Naapuritaulua päivittää jokainen jäsensolmu ja eteenpäin lähettäjäsolmu. Naapuritaulu sisältää listan solmuista kuten jäsensolmut, lähdesolmut ja eteenpäin lähetys ryhmän solmut. Naapuritaulua käytetään valitsemaan seuraava hyppy, jos ei voida käyttää seuraavan hypyn valitsijamekanismia. Naapuritaulu päivitetään käyttämällä varsinaisen reititysprotokollan keräämää tietoa liittymispakettien vastauksista. Tämä tieto ei lisää varsinaisen protokollan kuormitusta.

### 4.4.3 PIDIS juorukysymys (GREQ) ja juoruvastaus (GREP)

Lähdesolmun lähettämä juorukysymyspaketti sisältää järjestysnumeron lisäksi seuraavaa tietoa; Juorun lähettäjän osoitteen, juorukysymyksen järjestysnumeron, monilähetys lähde/ryhmä parin tiedot, odotettavissa oleva järjestysnumero, pakettien lukumäärä ja mistä mihin kadonneiden pakettien numerot ja vielä lista solmuista, joiden kautta juorukyselypaketti saapui.

Kun solmu vastaanottaa juorukysymyspaketin, jos kysely sisältää kadonneen paketin tai paketteja vastaanottajasolmulla. Silloin solmu luo juoruvastauksen jokaisesta kadotetusta viestistä, mitkä mainitaan kyselyssä. Edellyttäen, että viestit tai paketit ovat saatavilla vastaanottajasolmulla. Jokainen juoruvastaus on reititetty lähdesolmulle ja siksi sen täytyy sisältää lista solmuista, joiden kautta juorukyselypaketti tuli. Muuten juoruvastaus ei mene alkuperäiselle juorukyselyn lähettäjälle. Juorukyselypaketti sisältää seuraavat tiedot; Vastanottajasolmun eli juoruvastauksen lähettäjän osoitteen, listan solmuista, joiden kautta lähdesolmun kysely tuli, viestin järjestysnumeron, mikä oli kadonnut lähdesolmulta ja vielä lähde/ryhmä pari viestille. Vain jäsensolmut ovat huolissaan kadonneista paketeista ja voivat luoda juorukysymyspaketin. Vain jäsensolmut ja lähdesolmut voivat lähettää juoruvastausviestin, koska ne ovat tallentaneet mahdolliset kadonneet paketit välimuistiinsa. Juoruvastausviesteillähän päivitetään juorutaulua jokaisella välittäjäsolmulla kohden lähdesolmua.

### 4.4.4 PIDIS juoru seuraavan hypyn valinta

Lähdesolmu lähettää juorukyselyviestin, joka sisältää järjestysnumeron joko kaikille, yksittäisenä viestinä tai ei lähetä olenkaan riippuen kahdesta parametrista; Lähettämisen kaikille todennäköisyysarvosta ja naapuritiedoista. Juorukyselyviesti lähetetään kaikille algoritmin laskemalla todennäköisyydellä. Todennäköisyysarvo määritellään ja sitä valvotaan niin, ettei lähetyspaketit ylikuormita koko verkkoa. Jos juorukyselyviestiä ei lähetetä kaikille, naapuritaulua tai juorutaulua käytetään hyväksi haettaessa seuraavan hypyn tietoja yksittäiselle vastaanottajalle. Käytetään hyväksi kaikkien aikojen juorulähetyksiä, ettei lähetä juorukyselyä solmulle, jolle on jo lähetetty juorukysely. Tämä tehdään vertailemalla solmutunnuksia solmujen, joihin on jo lähetty kysely ja seuraavan hypyn solmun välillä. Tällä estetään se, ettei jo lähettyjä juorukyselyitä lähetetä turhaan ja kuormiteta verkkoa lisää. (Shen & Rajagopalan, 2005).

## 4.5 Esimerkkejä muista laatuun perustuvista mekanismeista

Esittelen muutaman laatupalveluun perustuvan reititysprotokollan tai REEF- ja PIDIS–mekanismeihin verrattavan sovelluksen lyhyesti. Niistä voi lukea lisää esittelyn jälkeen ilmoitetuista lähteistä.

Mottola, Cugola ja Picco esittelivät uuden sisältöperustusteisen reititysprotokollan (Content-Based Routing, CBR). Se järjestää Manet-verkon solmut puumuotoon. Tämä verkkojärjestely sietää tiheitä topologian uudelleen kokoonpanoja ja vähentää muutoksia, jotka kohdistuvat CBR kerrosta hyödyntäviin puihin. CBR on itsekorjautuva reititysprotokolla. (Mottola, et al., 2008).

Chakrabarti ja Kulkarni esittelivät uudenlaisen tavan laatutakuuseen DSR (dynamic source routing) reititysprotokollaan. Se laskee uudelleen vaihtoehtoiset reitit kohdesolmulle ja käyttämällä näitä reittejä, kun nykyiset reitit vikaantuvat tai ruuhkautuvat. Metodi varmistaa, että liikennevirta on tasapainossa koko valinnaisen reitin läpi, mutta myös sopiva määrä kaistanleveyttä saatavilla liikennevirralle, vaikka solmut olisivat liikkeessä ja vaihtaisivat paikkaa koko ajan. (Chakrabarti & Kulkarni, 2006).

Giruka ja Singhal kehittivät itsekorjautuvan tarvittaessa maantieteellisen polun reititysprotokollan (Self-healing On-demand Geographic Path Routing Protocol, OGRP) mobiili ad hoc -verkkoon. OGRP reititysprotokolla on tehokas, asematon ja skaalautuvainen, jolla on kolme erittäin hyvää ad hoc -verkkoon sopivaa ominaisuutta. Nämä ominaisuudet ovat ahne eteenpäin lähetys, vastavaikutteinen reittienetsintä ja lähdereititys. OGRP reititysprotokollassa lähdesolmut käyttävät hyväksi maantieteellistä topologiatietoa selvittäessään sijaintitietoa kohteesta ja maantieteellistä polkua pitkin yhteiselle kohdesolmulle. Maantieteelliset polut on kytketty irti solmu ID-osoitteesta. Lisäksi maantieteelliset polut ovat immuuneja muuttamaan verkon topologiaa. Hyödyntääkseen maantieteellisiä polkuja harvemmissakin verkoissa käyttää OGRP reititysprotokolla polun korjausmekanismia verkon topologian muutoksien hallitsemiseen. (Giruka & Singhal, 2007), (Ivascu, et al., 2009).

# 5. TYÖN TULOKSET JA ARVIOINTI

Työssä oli tarkoitus kartoittaa luotettavuutta ad hoc -verkoissa, varsinkin langattomassa mobiili-verkossa. Työssä käytiin luotettavuutta läpi määritteenä ja mitä kaikkea luotettavuus pitää sisällään. Määriteltiin mitä minä tutkimustyöntekijänä käsitän luotettavuudella ja miten luotettavuutta yleensä arvioidaan. Työssä käytiin läpi, miten luotettavuus rakennetaan tietotekniikassa. Vaikka käsiteltiin luotettavuutta ad hoc -verkoissa, niin samaan tapaan luotettavuus tehdään muihinkin verkkoihin. Luotettavuuteen verkossa vaikuttavat tekijät käytiin läpi kohtalaisen laajasti. Selvitettiin, miten eheys, tietoturvallisuus, todennus ja salaus hoidetaan ad hoc -verkoissa. Erityisesti keskityttiin langattomaan mobiiliverkkoon sen laajuuden ja tulevaisuusnäkymien takia. Langattomuus verkoissa kasvaa koko ajan. Työssä esiteltiin mielestäni tärkeimpiä manet-verkon reititysprotokollia. Käytiin läpi erityyppisiä reititysprotokollia ja niiden ominaisuuksia. Luotettavimpina reititysprotokollina pidetään monipolkureititysprotokollia, koska ne varautuvat topologiamuutoksiin, ruuhkiin ja linkin katkoksiin etsimällä valmiiksi vaihtoehtoisia reittejä. Lopuksi esiteltiin kaksi laatupalveluun perustuvaa mekanismia, joilla parannetaan reititysprotokollien toimintaa ja luotettavuutta. PIDIS- ja REEF-mekanismeja vertaillaan seuraavassa alaluvussa.

## 5.1 PIDIS vastaan REEF

PIDIS (Protocol-Independent Packet Delivery Improvement Service) ja REEF (REliable and Efficient Forwarding) ovat palvelun laatua parantavia mekanismeja, jotka toimivat reititysprotokollissa. PIDIS on kadonneiden pakettien palautuspalvelu, joka etsii kadonneet paketit ja sen jälkeen palauttaa ne uudelleen lähetettäviksi. REEF on luotettava ja tehokas eteenpäin lähettämiseen perustuva mekanismi, joka hakee luotettavimmat reitit pitämällä naapurisolmuille luotettavuusindeksejä pakettien luotettavasta toimittamisesta. Molemmat ovat hyviä mekanismeja, vaikkakin toimivat erityppisissä reititysprotokollissa. Käydään arviontina läpi molempien mekanismien perustoimintoja ja ominaisuuksia.

PIDIS lähettää juoruviestejä, jos ei saa kuittauksia pakettien läpimenoista päästä päähän. Se kyselee pakettien perään saadakseen selville, missä ne ovat kadonneet voidakseen lähettää ne uudelleen. PIDIS-mekanismi pitää välimuistia välittäjäsolmujen puskurissa, josta palautetaan kadonneita paketteja lähdesolmulle. Kun PIDIS-mekanismin välityksellä lähdesolmu lähettää juoruviestin niin solmu, joka on joko jäsenvälittäjäsolmu tai ryhmän jäsensolmu, jos sillä on kadonneita paketteja, jotka mainitaan juorupaketissa, niin se luo juoruvastauksen. Sen jälkeen se lähettää juoruvastauksen lähdesolmulle samaa reittiä, mitä juorukysymys tuli ja lisäksi lähettää muististansa kadonneet paketit

juoruvastauksen kanssa samaa reittiä. Kaikki välittäjäsolmut, jotka ovat tällä juoruvastauksen kulkemalla matkalla, päivittävät tietoja juoruvastauksesta. Näin kaikki solmut tietävät reitit, millä katoaa eniten paketteja ja myös reitit, joilla katoaa vähiten paketteja. Näitä tietoja käytetään valitessa reittejä pakettien lähettämiseen. Näin PIDIS-mekanismi auttaa reititysprotokollaa löytämään luotettavimmat ja nopeimmat reitit lähdesolmulta kohdesolmulle. PIDIS-mekanismi päivittää juoru- ja naapuritaulua. PIDIS-mekanismi toimii hyvin myös laajoissa verkoissa ja soveltuu myös pienempiinkin verkkoihin. Se käyttää parviälymekanismia kadonneiden pakettien etsimiseen. Sopeutuu kohtalaisen hyvin verkon muutoksiin ja on skaalautuvainen. Parantaa verkon luotettavuutta löytämällä parempia reittejä, joilla katoaa vähemmän paketteja.

REEF-mekanismi on monipolkuisiin reititysprotokolliin tarkoitettu palvelun laatua parantava toiminto. Kun REEF-mekanismia käyttävä lähdesolmu lähettää paketin eteenpäin, niin se tekee reittivalinnan ensimmäisen hypyn sen mukaan, miten luotettava naapurisolmu on. REEF-mekanismi pitää kirjaa naapurisolmun kautta kulkevien pakettien luotettavuudesta, sen mukaan miten hyvin paketit ovat menneet kohteelle. Kun lähdesolmu lähettää viestin naapurisolmulle, joka toimittaa viestin eteenpäin ja jos viesti menee perille ja se saa kuittauksen pakettien perille menosta. Jos paketti menee perille kohteelle, naapurisolmun luotettavuus kasvaa, jos ei mene perille luotettavuus vähenee. Näin se pitää naapurisolmuille luotettavuusindeksiä. Jokainen solmu verkon alueella toimii samoin. Jokaisella solmulla on luotettavuustaulu naapurisolmuista. Paketteja lähettäessä jokainen solmu valitsee luotettavimman naapurisolmun pakettien eteenpäin lähettämistä varten. Näin REEF-mekanismi toimii ja käyttää aina luotettavinta reittiä, koska kaikki välittäjäsolmut tekevät samoin, että valitsevat luotettavimman naapurisolmun pakettien eteenpäin lähettämiseen. Näin se REEF-mekanismi pyrkii löytämään myös solmut, jotka eivät tee yhteistyötä verkon kanssa. Jos välittäjäsolmu ei pysty jostain syystä lähettämään paketteja eteenpäin, REEF-mekanismi toimii niin, että se vähentää ensin lähetysnopeutta ja tarkkailee, onko ongelmaa väliaikainen vai jatkuva. Jos ongelma jatkuu, REEF-mekanismi vaihtaa seuraavaksi luotettavimpaan naapurisolmuun, joka saa toimittaa paketit eteenpäin. REEF-mekanismi käyttää eteenpäin lähettämiseen priorisointia. Se luokittelee tulevat paketit prioriteettiarvon mukaan. Korkeamman prioriteetin saavat paketit, jotka tulevat korkeamman luotettavuusarvon solmulta.

Molemmat sekä PIDIS että REEF ovat hyviä mekanismeja parantamaan laatua verkon toiminnoissa. Molemmat parantavat läpimenoaikoja koko verkon alueella ja toimivat hyvin laajoissakin verkoissa. Molemmat auttavat reititysprotokollia löytämään lyhyimmät ja luotettavimmat reitit lähdesolmulta kohdesolmulle. Molemmilla on vaihtoehtoisia reittejä, jos jokin reitti tai linkki katoaa. Molemmat soveltuvat hyvin verkon topologiamuutoksiin. PIDIS-mekanismi käyttää lähetyksissä hyväksi päivitystaulujen tietoja, jottei kuormittaisi verkkoa turhilla kyselyillä. REEF-mekanismi ylläpitää luotettavuusindeksiä joka naapurisolmulle päivitystaulussa ja käyttää sen tietoja hyväksi lähetyksissä. Taulukossa (Taulukko 2) on vertailtu joitain molempien mekanismien ominaisuuksia.

| PIDIS | REEF |
|---|---|
| Soveltuu lähes kaikkiin monilähetysprotokolliin | Soveltuu monipolkuprotokolliin |
| | Perustuu luottamuksen rakentamiseen ja maineeseen |
| Juoruihin perustuva palvelu, joka palauttaa kadonneita paketteja. Ylläpitää tietoa reiteistä, joilla katoaa eniten paketteja | Palvelu, joka pyrkii löytämään luotettavimmat reitit ja pyrkii löytämään huonosti käyttäytyvät solmut tarkkailemalla solmujen käytöstä |
| Käyttää lähetykseen reittejä, joilla katoaa vähiten paketteja | Käyttää parasta/luotettavinta reittiä tai todennäköisyysjärjestelmällä valittua reittiä |
| Ylläpitää ja päivittää juoru- ja naapuritaulussa solmujen tietoja | Ylläpitää luotettavuustaulua naapurisolmuille |
| Luotettavuus korkea | Luotettavuus korkea |
| Verkon kuormitus monilähetyksissä voi olla aika korkea | Verkon kuormitus päivitystietojen takia aika korkea |
| Skaalautuvuus hyvä | Skaalautuvuus hyvä |

**Taulukko 2.** PIDIS - ja REEF -mekanismien ominaisuuksien vertailu.

## 5.2 Yhteenveto

Työssä tutkittiin, miten luotettavuus rakennetaan ja toteutetaan ad hoc -verkkoihin. Mielestäni tutkimuksen aihe ei ollut helpoin mahdollinen ja vaikka aiheen valitsinkin itse. Työ oli haastava ja mielenkiintoinen. Työn tuloksena opin valtavasti ad hoc -verkoista

ja varsinkin niissä käytettävistä reititysprotokollista. Tutkimustyö vastasi mielestäni valittua aihetta hyvin ja käytiin luotettavuuteen liittyviä asioita läpi useilla lähestymistavoilla. Tutkimuksen ei ollut tarkoituskaan olla kaiken kattava ja joitain asioita olisi voinut käydä lähemmin, tarkemmin ja yksityiskohtaisemmin läpi. Rajasin tarkoituksella, että tekniikoita ei käyty kovin tarkasti ja yksityiskohtaisesti läpi vaan paremminkin vain pintapuolisesti. Tutkimukset, joita tässä työssä tutkin eivät käsitelleet niitä sen tarkemmin. Yhteenvetona voi todeta aiheen ollen hyvä ja haastava. Ad hoc -verkko ei ole luotettavin verkko, mutta nopean ja edullisen rakentamisen ja purkamisen takia se sopii moneen tilanteeseen ja tarkoitukseen tosi hyvin. Langattoman tekniikan yleistymisen seurauksena ad hoc -verkotkin kehittyvät koko ajan. Protokollatkin kehittyvät koko ajan ja niihin tulee lisää luotettavuutta ja turvallisuutta lisääviä ominaisuuksia. Voi olla hyväkin asia, ettei ad hoc -verkoissa ole standardia protokollille. Standardi ei omalla osuudella pysty suuntaamaan kehitystä vain yhteen suuntaan.

# LÄHTEET

Abolhasan, M., Wysocki, T. & Dutkiewicz, E., 2004. A review of routing protocols for mobile ad hoc networks. Teoksessa: *Ad Hoc Networks 2.* Wollongong: Elsevier, pp. 1 - 22.

Akyildiz, I., Wang, X. & Wang, W., 2004. Wireless mesh networks: a survey. *Computer Networks 47,* Marraskuu.pp. 445 - 487.

Al-Rabayah, M. & Malancy, R., 2012. A New Scalable Hybrid Routing Protocol for VANETs. *IEEE Transactions on Vehicular Technology, Vol. 61, No 6,* Heinäkuu.pp. 2625 - 2635.

Arun, K. B. R., Lokanatha, C. R. & Prakash, S. H., 2008. Performance Comparison of Wireless Mobile Ad-Hoc Network Routing Protocols. *International Journal of Computer Scie,* pp. 337 - 343.

Bakht, H., 2005. Routing protocols for mobile ad hoc networks. *1st International Computer Engineering,* pp. 1 - 8.

Beijar, N., 2002. *Zone Routing Protocol(ZRP),* Helsinki: Networking Laboratory, Helsinki University of Technology.

Bernsen, J. & Manivannan, D., 2009. Unicast routing protocols for vehicular ad hoc network: A critical comparison and classification. *Pervasive and Mobile Computing 5,* pp. 1 - 18.

Bisdikian, C., 2001. An Overview of the bluetooth wireless technology. *IEEE Communications Magazine,* Joulukuu.pp. 86-94.

Boukerche, A., 2011. Routing protocols in ad hoc networks. Teoksessa: *Computer Networks 55.* s.l.:Elsevier, pp. 3032 - 3080.

Chakrabarti, G. & Kulkarni, S., 2006. Load Balancing and Resource Reservation in Mobile Ad-Hoc Networks. *Ad Hoc Networks 4,* pp. !86 - 203.

Chlamtac, I., Conti, M. & Liu, J., 2003. Mobile ad hoc networking: imperatives and challenges. Teoksessa: *Ad hoc networks 1.* s.l.:Elsevier, pp. 13 - 64.

Comer, D. E., 2009. *Computer networks and internets.* San Jose: Pearson International Edition.

Conti, M., Gregori, E. & Maselli, G., 2005. *Improving the performability of data transfer in mobile ad hoc networks,* Pisa: ITT Institute, CNR.

Conti, M., Gregori, E. & Maselli, G., 2006. Reliable and efficient forwarding in ad hoc networks. Teoksessa: *Ad Hoc Networks 4 (2006).* Pisa, Italy: IIT Institute, CNR Via G. Moruzzi 1, pp. 398-415.

Conti, M., Maselli, G. & Turi, G., 2004. Cross-layering in Mobile Ad Hoc Network Design. *IEEE Computer Society,* Helmikuu.pp. 48 - 51.

Ferro Erina, P. F., 2004. Bluetooth and wi-fi wireless protocols. *the IEEE Wireless communication magazine,* pp. 1-24.

Gerla, M., 2005. *Ad Hoc networks.* s.l.:Ucla Computer Science Department.

Giruka, V. & Singhal, M., 2007. A self-healing On-demand Geographic Path Routing Protocol for mobile ad-hoc networks. *Ad Hoc Networks 5,* pp. 1113 - 1128.

Gozalvez, J., Sepulcre, M. & Bauza, R., 2012. IEEE 802.11p Vehicle to Infrastructure Communications in Urban Environments. *IEEE Communications Magazine, vol,* Toukokuu.pp. 1 - 8.

Gupta, A., Sadawarti, H. & Verma, A., 2011. Review of Various Routing Protocols for MANETs. *International Journal of Information and Electronics Engineering,* Marraskuu.pp. 251 - 259.

Hakala, M., 2006. *Tietoturvallisuuden käsikirja.* Jyväskylä: Docento, 2006 WS Bookwell.

Islam, M., Hamid, M. & Hong, C., 2009. *SHWMP: A Secure Hybrid Wireless Mesh Protocol for IEEE 802.11s Wireless Mesh Networks,* s.l.: Department of Computer Engineering, Kyung Hee University, Republic of Korea.

Ivascu, G., Pierre, S. & Quintero, A., 2009. QoS routing with traffic distribution in mobile ad hoc networks. *Computer Communications 32,* pp. 305 - 316.

Jiang, H. & Garcia-Luna-Aceves, 2001. *Performance Comparison of Three Routing Protocols for Ad Hoc Networks.* s.l., IEEE, pp. 1 - 9.

Jun, J. & Sichitiu, M., 2008. MRP: Wireless mesh networks routing protocol. *Computer Communications,* pp. 1 - 23.

Järvinen, P., 2002. *Tietoturva & yksityisyys /Petteri Järvinen.* Jyväskylä: Docento Finland, 2002.

Kaarnalehto, M., 2011. *opinnäytetyö.* Espoo: Laurea-ammattikorkeakoulu.

Kangas, A., 2003. Tietoturvallisuus ja tietojärjestelmät. *Sytyke ry, systeemityö4/03,* pp. 1 - 28.

Koskinen, J., 2007. *Liikkuvan tietoliikenteen tietoturvallisuus,* Tampere: s.n.

Kute, V. & Kharat, M., 2013. Analysis of Quality of Service for the AOMDV Routing Protocol. *ETASR - Engineering, Technology & Applied Science Research,* pp. 359 - 362.

Käpylä, T., 2000. Salaustekniikoilla lisää turvallisuutta. *Sytyke ry Systeemityö 4/00,* pp. 26-32.

Lehtonen, S., 2004. *Diplomityö; Turvallisuuden hallinta yrityksen langattomissa lähiverkoissa.* Helsinki(Suomi): Helsingin teknillinen korkeakoulu.

Lemus, A. & Mendez, A., 2004. *Performance Improvement of Ad-Hoc Networks with ZRP,* Monterrey, Nuevo León: s.n.

Li, F. & Wang, Y., 2007. Routing in Vehicular Ad Hoc Networks: A Survey. *IEEE Vehicular Technology Magazine,* pp. 12 - 22.

Marina, M. & Das, S., 2001. *On-demand Multipath Vector Routing in Ad Hoc Networks.* s.l., IEEE Explore, pp. 14 - 23.

McEvoy, R., Crowe, F., Murphy, C. & Marnane, W., 2006. *Optimisation of the SHA-2 Family of Hash Functions on FPGAs,* s.l.: the Irish Research Council for Science, Engineering and Technology (IRCSET).

Miller, J., 2008. *Vehicle-to-Vehicle-to-Infrastructure (V2V2I) Intelligent Transportation System Architecture.* s.l., s.n., pp. 1 - 6.

Mistry, N. & Jinwala, D. C., 2010. *Improving AODV Protocol against Blackhole Attacks.* Hong Kong, IMECS 2010, pp. 1 - 6.

Mottola, L., Cugola, G. & Picco, G., 2008. A Self-Repairing Tree Topology Enabling Content-Based Routing in Mobile Ad Hoc Networks. *IEEE Transactions on Mobile Computing,* pp. 1 - 15.

Nikaen, N., Bonnet, C. & Nikaen, N., 2001. *HARP - Hybrid Ad Hoc Routing Protocol,* Sophia Antipolis: Institut Eurecom.

Paul, B., Ibrahim, M. & Bikas, M., 2011. VANET Routing Protocols: Pros and Cons. *International Journal of Computer Applications Volume 20 - no 3,* Huhtikuu, Issue 3, pp. 28 - 34.

Pei, G., Gerla, M. & Chen, T., 2000. Fisheye state routing: A routing scheme for ad hoc wireless networks. *Cummunications,* pp. 70 -74.

Periyasamy, P. & Karthikeyan, E., 2011. Performance Evalution of AOMDV Protocol Based on Various Scenario and Traffic Patterns. *International Journal of Computer Science, Engineer and Applications(IJCSEA),* pp. 33 - 48.

Raya, M. & Hubaux, J.-P., 2007. Securing vehicular ad hoc networks. *Journal of Computer Security 15,* pp. 39 - 68.

Sairam, K., Gunasekaran, N. & Reddy, S., 2002. Bluetooth In Wireless Communication. *IEEE Communications Magazine,* pp. 90-97.

Salonen, M., 2009. *OPC UA- tietoturvatoteutus java- ohjelmointikielellä.* Tampere: Tampereen teknillinen yliopisto.

Shen, C.-C. & Rajagopalan, 2005. Protocol-independent multicast packet delivery improvement service for mobile Ad hoc networks. *Ad Hoc Networks ,* pp. 1 - 18.

Sklavos, N. & Koufopavlou, O., 2003. ON THE HARDWARE IMPLEMENTATIONS OF THE SHA-2 (256, 384, 512) HASH FUNCTIONS. *Proceedings of IEEE International Symposium on Circuits & Systems,* pp. 153 - 156.

Sriskanthan, N., Tan, F. & Karande, A., 2002. Bluetooth based home automation system. *Microprocessors and Microsystems 26,* 10 Toukokuu, Issue 26, pp. 281 - 289.

Srivastava, V. & Motani, M., 2005. *Cross- layer Design.* Singapore: s.n.

Stallings, W., 2002. *Wireless communications and networks.* Upper saddle River(New Jersey): Prentice- Hall Inc..

Subramaniam, A., 2003. *Power Management In Zone Routing Protocol (ZRP),* Birmingham: Univeercity Of Central England, Birmingham.

Tarique, M., Tepe, K. E., Adibi, S. & Erfani, S., 2009. Survey of multipath routing protocols for mobile ad hoc networks. Teoksessa: *Journal of Network and Computer Applications 32.* s.l.:Elsevier, pp. 1125 - 1143.

Usop, N., Abdullah, A. & Abidin, A., 2009. Performance Evaluation of AODV, DSDV & DSR Routing Protocol in Grid Environment. *International Journal of Computer Science and Network Security*, Heinäkuu, pp. 261 - 268.

Valera, A., Seah, W. & Rao, S., 2002. *CHAMP: A Highly-Resilient and Energy-Efficient Routing Protocol for Mobile Ad Hoc Networks.* s.l., IEEE XPLORE, pp. 1 - 5.

Wang, J., Osagie, E., Thulasiraman, P. & Thulasiram, R., 2009. HOPNET: A hybrid ant colony optimization routing algorithm for mobile ad hoc network. Teoksessa: *Ad Hoc Networks 7.* Winnipeg: Department of Computer Science, University of Manitoba, pp. 690 - 705.

Wan, X., Yin, Y. & Yu, H., 2005. *Finding Collisions in the Full SHA-1.* s.l., s.n., pp. 1-20.

Virkkala, P., 2009. *Insinöörityö.* Tampere: Tampereen ammattikorkeakoulu.

Yang, X., Liu, J., Zhao, F. & Vaidya, N., 2003. *A Vehicle-to-Vehicle Communication Protocol for Cooperative Collision Warning,* Illinois: Electrical and Computering Engineering Department at University of Illinois.

Yao, Z. ym., 2003. *A neighbor-table-based multipath routing in ad hoc networks.* Hong Kong, IEEE, pp. 1739 - 1743.