

# Energiatehokkaista reititysmenetelmistä

Olli Puttonen

21. helmikuuta 2005

## 1 Johdanto

Tässä paperissa esitellään lyhyehkösti kaksi reititysmenetelmää.  $max-min zP_{min}$ -algoritmin ovat kehittäneet Li, Aslam ja Rus [1], ja min-MINMAXin Gandaham, Dawande ja Prakash [2]. Molemmat ovat suunniteltu käytettäväksi langattomissa sensoriverkoissa, joissa laitteiden käytössä oleva energia ja langattoman yhteyden kantama ovat rajoitettuja. Tavoitteena on onnistua reitittämään mahdollisimman monta pakettia ennen kuin jonkun paketin reititys on mahdotonta.

Yllä asetetun tavoitteen saavuttamiseksi energiaa tulee pyrkiä säästämään, mutta pelkkä tehonkäytän minimointi ei ole hyvä strategia, sillä se johtaa samojen (minimaalisesti energiaa kuluttavien) polkujen jatkuvaan käyttöön, josta taas seuraa eräiden laitteiden energian loppuminen varsin aikaisin. Molemmat esiteltävät algoritmit pyrkivät paitsi kuluttamaan niukasti energiaa myös jakamaan reitityksestä syntyvät kustannuksen suhteellisen tasaisesti eri laitteille.

Sensorilaitteiden muodostamaa verkkoa on luonnollista mallintaa graafina, jonka solmut vastaavat laitteita ja kaaret laitteiden välisiä suoria viestiyhteyksiä. Kaarien painoilla voidaan kuvata standardikokoisten pakettien lähettämiseen kuluvaa energiaa.

Ennen algoritmien tarkempaa esittelyä todetaan kaksi lausetta, jotka osoittavat optimaalisen ratkaisun etsimisen olevan käytännössä mahdotonta. Samalla ne perustelevat optimia huonomman tuloksen antavien algoritmien käytön. Ennen lauseita käymme esittelemme tarvittavia käsitteitä.

Usein ei voida etukäteen tietää viestisekvessejä, eli seuraavien viestien lähettäjiä ja vastaanottajia. Algoritmia, joka ei tiedä tulevia viestisekvessejä kutsutaan online-algoritmiksi. Vastaavasti viestisekvenssin tuntevaa algoritmia kutsutaan offline-algoritmiksi.

Kahta eri algoritmia vertaillen tutkimme kilpailusuhdetta (competitive ratio). Kilpailusuhteella tarkoitetaan algoritmien toimintaa kuvaavien suureiden, esim. onnistuneesti välitettyjen pakettien määrä, suhteita. Kilpailusuhde riippuu paitsi algoritmeista, myös saadusta syötteestä.

**Lause 1.** *Optimaalisen offline-algoritmin ja minkä tahansa online-algoritmin kilpailusuhde voi olla mielivaltaisen suuri, kun tarkastellaan onnistuneesti välitettyjen pakettien lukumäärää ennen kuin jokin pakettia ei pystytä välittämään tai ennen kuin joltain laitteelta loppuu energia.*

**Lause 2.** *Optimaalinen offline-reititys päätösongelmana (onko mahdollista välittää vähintään  $n$  pakettia annetuin energiarajoituksin) on NP-täydellinen.*

Molempien lauseiden todistukset on esitetty julkaisussa [1].

## 2 *Max-min* $zP_{min}$ -algoritmi

### 2.1 Algoritmin kuvaus

Tämän algoritmin oletuksen ovat varsin yleisiä. Kyseessä on online-algoritmi. Viestin lähettämiseen kuluva energia on muotoa  $kd^c$ , missä  $d$  on laitteiden välinen etäisyys ja  $k$  ja  $c$  vakioita,  $2 \leq c \leq 4$ .

Tavoitteenamme on reitittää paketti energiatehokkaasti eli löytää kahta annettua laitetta yhdistävä polku, jota pitkin paketin välitys ei kuluttaisi kohtuuttomasti energiaa. Eräs kandidaatti on *max-min*-polku, eli polku, jossa jäljellä olevan energian osuuden (alkuperäisestä energiasta) minimi polun laitteilla on mahdollisimman suuri sen jälkeen, kun paketti on reitetetty polkua pitkin. *max-min* polkuja pitkin reitittäminen jakaa kustannukset varsin tasaisesti eri laitteille, mutta on huono valinta energiankulutuksen minimoinnin kannalta.

Olkoon  $P_{min}$  vähiten energiaa kuluttavan polun energiankulutus ja  $z$  algoritmille annettava parametri,  $z \geq 1$ . *max-min*  $zP_{min}$ -polku on *max-min*-polku sillä rajoituksella, että energiankulutus saa olla enintään  $zP_{min}$ . Tämä rajoitus korjaa *max-min*in turhan suuren energiankulutuksen.

Olkoon  $u_{ij}$  laitteen  $i$  jäljellä olevan energian osuus sen jälkeen, kun se on lähettänyt paketin laitteelle  $j$ . *max-min*  $zP_{min}$ -polku voidaan etsiä seuraavalla algoritmilla:

- 1 Etsi vähiten energiaa kuluttava polku Dijkstran algoritmilla. Olkoon  $P_{min}$  sen energiankulutus.
- 2 Etsi graafista vähiten energiaa kuluttava polku. Jos sen energiankulutus  $> zP_{min}$ , edellinen polku on ratkaisu.
- 3 Etsi minimaalinen  $u_{ij}$  polulta. Olkoon se  $u_{min}$ .
- 4 Poista verkosta kaikki kaaret, joihin liittyy  $u_{ij} \leq u_{min}$ .
- 5 Siirry kohtaan 2.

Joka kierroksella graafista poistetaan ainakin yksi kaari, joten kukin algoritmin kohta suoritetaan korkeintaan  $|E|$  kertaa. Dijkstran algoritmin suoritus-aika on  $O(|E| + |V| \log |V|)$ . Siis koko algoritmin suoritus-aika on  $O(|E| \cdot (|E| + |V| \log |V|))$ . Jos sopiva  $u_{min}$  etsittäisiin binäärihaulla, suoritus-aika olisi vain  $O(\log |E| \cdot (|E| + |V| \log |V|))$ . Pelkkä *max-min*-polku voidaan etsiä modifioidulla Bellman-Ford-algoritmilla ajassa  $O(|V| \cdot |E|)$  (tästä tarkemmin seuraavalla sivulla).

Erilaisten kokeilujen perusteella vaikuttaa sopivalta valita  $1,5 \leq z \leq 5$ .

### 2.2 Aluejakoon perustuva variantti

*max-min*  $zP_{min}$  olettaa, että kaikkien verkon laitteiden sijainnit ja jäljellä olevan energian määrät on tiedossa. Suurissa verkoissa kaiken tämän tiedon ylläpitäminen on hyvin vaikeaa. Mahdollinen ratkaisu on hierarkkinen lähestymistapa, jossa verkko jaetaan alueisiin ja summittainen reitti löydetään reitittämällä alueiden välillä.

Miten verkko sitten jaettaisiin alueisiin? Tästä huolehtisi jokin kontrolloiva taho, jolla ilmeisesti olisi runsaasti informaatiota ja resursseja käytössään. Mikä tällainen taho olisi ja millaista jakoalgoritmia kannattaisi käyttää on vielä avoin kysymys.

Jotta reititys aluetasolla onnistuisi, pitää jotenkin arvioida eri alueiden energiatasoja. Tämä onnistuu keräämällä kullakin alueella tiedot eri laitteiden energiatasoista ja simuloimalla pakettien lähetystä max-min  $zP_{min}$ -algoritmilla. Energiataso arvioidaan erikseen eri suuntaan kulkeville paketeille, esim. pohjoisessa olevalta alueelta eteläpuoliselle alueelle kulkeville paketeille. Energiataso on arvio pakettien määrästä, jotka on mahdollista lähettää ennen energian loppumista.

Simulointia varten luodaan kuvitteellinen lähtöaluetta kuvaava solmu. Tämä solmu yhdistetään kaikkiin reunalla oleviin laitteisiin nollapainoisilla kaarissa. Vastaaavasti kohdealueeseen luodaan samanlainen solmu. Reititys tapahtuu näiden solmujen välillä.

Tiedon keruusta ja energiataason laskemisesta vastaa jokin alueen laite. Kyseessä voi olla laite, jolla on eniten energiaa, tai eri laitteet voivat vuorotella keskenään. Toki muitakin mahdollisuuksia on. Energiatasoarviota päivitetään energiatasoon kääntäen verrannollisella taajuudella, sillä mitä pienempi energiataso on, sitä suurempia suhteellisia muutoksia samassa ajassa voi tapahtua. Toisaalta alue, jolla on vain vähän energiaa jäljellä, ei saisi kuluttaa sitä vähää energiataason tiheään uudelleenarviointiin.

Reititys aluetasolla tehdään muodostamalla graafi, jossa kutakin aluetta kuvaa solmu. Yhteyttä alueelta toiselle kuvataan solmulla, jonka "energia" on yllämainittu energiataso lähtöalueelta kohdealueelle viestiessä. Reititys alueiden välillä tehdään etsimällä *max-min*-polku modifioidulla Bellman-Ford-algoritmilla:

Annettu graafi  $G(E, V)$ , ja solmujen energiataason  $p(v)$ ,  $v \in V$ .

Etsitään polku  $s$ :stä  $t$ :hen,  $s = v_0, v_1, \dots, v_{k-1}, v_k = t$  siten, että  $\min_{i=1}^{k-1} p(v_i)$  on mahdollisimman suuri.

```

for each  $v \in V[G]$ 
    do if  $(s, v) \in E[G]$ ,
        then  $d[v] \leftarrow \infty$ ,  $\pi[v] \leftarrow s$ 
        else  $d[v] \leftarrow 0$ ,  $\pi[v] \leftarrow NIL$ 
 $d[s] \leftarrow \infty$ 

for  $i \leftarrow 1$  to  $|V[G]| - 1$ 
    do for each edge  $(u, v) \in E[G]$  and  $u \neq s$ 
        do if  $d[v] < \min(d[u], p[u])$ 
            then  $d[v] \leftarrow \min(d[u], p[u])$ 
                 $\pi[v] \leftarrow u$ 

return  $\pi[t]$ 

```

Aluejaon tulee olla sellainen, että viereikkäisten alueiden rajalla on muutamia molempiin alueisiin kuuluvia laitteita. Jokaiselle näistä laitteista lasketaan, kuinka monta pakettia se voi välittää alueen halki. Reititys alueen läpi onnistuun valitsemalla reunoilta laitteet, jolla saadut pakettien määrät ovat suurimmat, ja etsimällä *max-min*  $zP_{min}$ -polku näiden laitteiden välillä.

### 2.3 Testituloksia

*max-min*  $zP_{min}$ -algoritmia testattiin sijoittamalla 20 laitetta satunnaisesti  $10 \times 10$ -alueelle ja simuloimalla algoritmia. Muut laitteet lähettivät viestejä yh-

delle yhdyskäytävälaitteelle. Koe toistettiin 500 kertaa. Havaittiin, että *max-min*  $zP_{min}$ :n kilpailusuhde optimiin (offline-algoritmillä laskettuna) verrattuna on lähes vakio verkon elinikää verrattaessa. *max-min*  $zP_{min}$  kilpailusuhde on yli 80% 92% kokeista ja yli 90% 53% kokeista. Tulos on varsin hyvä online-algoritmillemme.

Aluepohjaista reititystä testattiin erikseen. Koska aluejako on tietoa vähentävä yksinkertaistus, sen käytön voi olettaa heikentävän tuloksia. Siksi aluepohjaista algoritmia verrattiin tavalliseen *max-min*  $zP_{min}$ -algoritmiin.

Kokeissa sijoitettiin 1000 laitetta satunnaisesti säännölliseen alueeseen, joka jaettiin 24 alueeseen. Jokainen laite lähettää säännöllisesti viestejä tietylle yhdyskäytävälaitteelle. Ennen kuin jonkun paketin välitys oli mahdotonta, *max-min*  $zP_{min}$ -algoritmi välitti noin 41000 pakettia ja aluepohjainen algoritmi noin 39000. Lisäksi aluepohjainen algoritmi tarvitsi moinekertaisesti vähemmän tietoa verkosta ja prosessoriaikaa.

Toisessa kokeessa sijoitettiin 1240 laitetta satunnaisesti  $6,2 \times 6,2$ -neliöön. Neliö jaettiin  $5 \times 5$ -alueisiin. Aluepohjaista algoritmia käytettäessä aika ennen jonkun laitteen energian loppumista oli 96% vastaavasta *max-min*  $zP_{min}$  saavutetusta ajasta.

### 3 Min-MINMAX

#### 3.1 Algoritmin kuvaus

Tämä algoritmi ei ole niin yleispätevä kuin *max-min*  $zP_{min}$ . Nyt oletamme, että verkossa on sekä sensorilaitteita että tukiasemia. Toisin kuin sensoreilla, tukiasemille on runsaasti energiaa ja viestintäkapasiteettia käytössään. Säännöllisin väliajoin kukin sensori lähettää yhden datapaketin tukiasemalle. Lisäksi lähitetään myös reititystä koskevia kontrollipaketteja, mutta sensorit eivät viesti keskevään. Reititys tapahtuu kierroksissa, joiden aikana kukin sensori lähettää  $T$  pakettia.

Ilmeisesti oletetaan myös paketin lähetykseen kuluvan energian olevan vakio, joten kaavalla  $kd^c$  ei saadakaan energiankulutusta vaan maksimikantama. Tätä oletusta ei kuitenkaan ole eksplisiittisesti todettu.

Min-MINMAX-algoritmi perustuu maksimivirtausongelma muunnelmaan, minimihintavirtaukseen. Siinä jokaiseen kaareen liittyy paitsi kapasiteetti, myös hinta, jonka yksikkövirtaus maksaa. Tavoitteena on löytää hinnaltaan mahdollisimman alhainen maksimivirtaus. Kuten maksimivirtaus, myös minimihintavirtaus voidaan löytää polynomisessa ajassa. Parhaan tällä hetkellä tunnetun algoritmi aikavaatimus on  $O((|E| \log |V|)(|E| + |V| \log |V|))$ .

Tällä kertaa energiankulutus pyritään jakamaan tasaisesti vaatimalla, että kunkin laitteen energiankulutus kierroksen aikana on enintään  $\alpha \times RE_i$ , missä  $RE_i$  on laitteen  $i$  jäljellä olevan energian määrä ja  $\alpha$  on parametri ( $0 < \alpha \leq 1$ ).

Edelläolevan lisäksi vaadimme, että laitteiden energiankulutusten maksimi on mahdollisimman pieni. Jos tämäkään ehto ei anna yksikäsitteistä ratkaisua, minimoimme laitteiden energiankulutusten summan. Näin saadaan Min-MINMAX-menetelmä.

Jos  $\alpha \times RE_i$ -energiankulutusehdon lisäksi vain minimoimme kokonaisenergiankulutuksen, saadaan menetelmä jotka kutsutaan minimihintareititykseksi (Minimum Cost Routing, MCR).

Sekä Min-MINMAX että MCR-menetelmiä toteutettaessa ongelmaa kannattaa mallintaa maksimivirtausongelmana. Tätä varten luomme lähde- ja nielusolmut. Lähdesolmusta vedetään jokaiseen laitteeseen suunnattu kaari, jonka kapasiteetti on  $T$ . Jokaisesta tukiasemasta vedetään suunnattu nieluun kaari, jonka kapasiteetti on  $\infty$ .

Kaikkien kommunikaatioetäisyydellä olevien laitteiden välillä asetetaan kaaret molempiin suuntiin. Kaarien kapasiteetti on  $\infty$ . Lisäksi jokainen laite jaetaan kahtia vastaanotto- ja lähetyssolmuihin. Näiden välille luodaan suunnattu kaari, jonka kapasiteettia rajoittamalla varmistetaan, että solmun energiankulutus on enintään  $\alpha \times RE_i$ .

Eri solmujen väliin kaariin liitetään energiankulutusta vastaava hinta. Nyt MCR saadaan ratkaisemalla minimihintavirtaus yllä muodostetussa graafissa.

Min-MINMAX saadaan aikaan etsimällä binäärihaulla pienin mahdollinen energia  $E$  siten, että kaikki paketit saadaan reititettyä, kun jokaisen solmun energiankulutus  $< E$ . Binäärihaussa pitää ratkaista maksimivirtausongelma (ei minimihintavirtausta)  $\log U$  kertaa, missä  $U$  on suurin pakettien määrä, joka voi kulkea yhden laitteen läpi. Kukin maksimivirtaus voidaan ratkaista ajassa  $O(|V||E| + |V|^2 \log U)$ .  $U$  on suoraan verrannollinen  $RE_{max}$ :iin, joten binäärihaun aikavaativuus on  $O(|V||E| \log RE_{max} + (|V| \log RE_{max})^2)$ . Lopuksi, kun optimaalinen  $E$ :n arvo tiedetään, pitää vielä ratkaista minimihirtavirtaus.

Koska kyseessä ei ole online-algoritmi, jokin tukiasema voi ennen kierroksen alkua ratkaista reititysongelman ja lähettää ratkaisun laitteille. Näin laitteiden ei tarvitse kerätä mitään tietoa muusta verkosta. Tukiasemilta tiedon keruu onnistuu helposti.

Reitityksen aikana parametrin  $\alpha$  arvoa täytyy kasvattaa, sillä muuten reititys kävisi pian mahdottomaksi.

## 3.2 Testituloksia

Eri menetelmiä testaamalla havaittiin, että Min-MINMAX on selvästi parempi kuin MCR. Selvimmin ero näkyy kierrosten määrässä ennen kuin joltain laitteelta loppuu energia: ero oli 3-5-kertainen. Min-MINMAXin toiminnasta tehtiin muutama mielenkiintoinen havainto. Sen energiankulutus pysyy lähes tulkoon vakiona ajan kuluessa. Lisäksi kaikilla laitteilla riittää energiaa kunnes se samalla kierroksella loppuu ainakin puolelta laitteesta.

Testaamalla havaittiin myös, että  $T$ :n arvoilla välillä  $10^2$ - $10^5$  saatiin lähes samat tulokset. Siis  $T$ :llä ei ole juurikaan vaikutusta reitityksen sujuvuuteen.

## References

- [1] Qun Li, Javed Aslam, Daniela Rus, *Online Power-aware Routing in Wireless Ad-hoc Networks*, Proc. 7th Ann. Intl. Conf. on Mobile Computing and Networking (ACM Mobicom'01), 97-107. ACM, New York NY, 2001.
- [2] S. Gandham, M. Dawande, R. Prakash, *An integral flow-based energy-efficient routing algorithm for wireless sensor networks*, Proc. IEEE Wireless Communications and Networking Conference 2004 Vol. 4, 2341-2346.