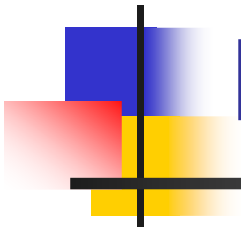


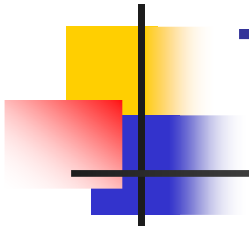
# Luotettava tiedonsiirtomenetelmä hybridiverkkoihin

25.3.2008

Eero Solarmo
Valvoja: Prof. Raimo Kantola
Suorituspaikka: Tietoverkkolaboratorio/TKK

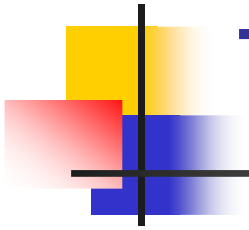
# Esityksen sisältö

- Tavoite ja tutkimusongelman esittely
- Taustatyö
  - Aiemmat menetelmät
  - Vertailu
- HBH
  - Menetelmän periaate
  - Simulointiskenaariot
- Tulokset
- Yhteenveto ja päätelmät



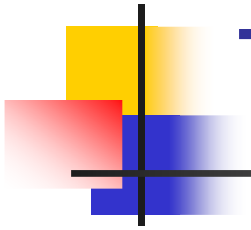
# Tavoite

- Luotettavan tiedonsiirron varmistaminen hybridiverkossa
  - Verkon infrastruktuuri vapaavalintainen, eli se voi koostua langattomista ja langallisista linkeistä mielivaltaisesti
- Riippumattomuus reitityksestä ja laitteistosta
  - Täydellinen riippumattomuus haastavaa lukemattomien eri kokoonpanojen vuoksi
- Informaation eheys lähtökohtana, QoS ei olennainen vaatimus



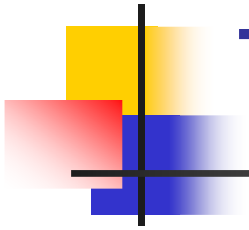
# Taustatyö

- Langattomiin anturiverkkoihin (WSN) on kehitetty useita menetelmiä
  - PSFQ, ESRT, HRS, FBcast
- Langattomat linkit ovat luotettavuuden kannalta suurin pullonkaula, joten painopiste on pääasiassa niissä
- Menetelmän tulee kuitenkin toimia myös langallisilla linkeillä
- Periaatteessa samaan ongelmaan on esitetty useita ratkaisumalleja



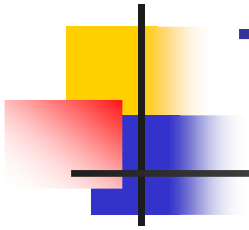
# Taustatyö

- PSFQ (Pump-slowly, fetch-quickly)
  - Kehitetty varmistamaan tiedonsiirto anturiverkon nielulta antureille (esim. ohjelmistopäivitys)
  - Tietoa lähetetään hitaasti (PS), mutta kadonneista paketeista palaudutaan nopeasti (FQ)
  - Perustuu hyppykohtaisiin negatiivisiin kuittauksiin (NACK)
  - NACK-tapahtuman eteneminen
    - Sekvenssistä kadonneen paketin NACK voi edetä huonoimmassa tapauksessa kohteelta aina lähteelle asti
    - PSFQ:ssa on toteutettu välimuistitoiminto tämän estämiseksi



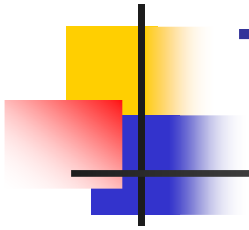
# Taustatyö

- ESRT (Event to Sink Reliable Transport)
  - Toimii anturiverkoissa tiedonsiirron varmistamiseksi antureilta nieluun
  - Perusajatus on se, että antureiden resurssit ovat rajalliset, jolloin ylimääräistä hallintaliikennettä olisi pyrittävä rajoittamaan
  - Kaikki laskenta tapahtuu nielussa resurssien säästämiseksi
  - Toiminta perustuu siihen, että nielu säätelee antureiden raportointitaajuutta siten, että tapahtumat pystytään havaitsemaan luotettavasti pienimmällä mahdollisella energiankulutuksella



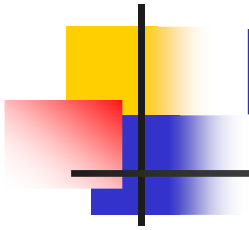
# Taustatyö

- FBcast
  - Anturiverkkojen langattomaan broadcastaukseen suunniteltu menetelmä
  - Perustuu suihkulähdekoodeihin (fountain code)
    - Alkuperäinen viesti m koodataan n:n paketin pituiseksi (n>m)
    - Vastaanotettaessa riittää, että otetaan vastaan k pakettia (k≥m), joista alkuperäinen viesti voidaan rekonstruoida
    - Analogia: Jos asetat kupin suihkulähteen alle, tärkeintä on että kuppi täyttyy, ei niinkään mitkä pisarat sen täyttävät
  - Uudelleenlähetykset vähentyvät, mikä johtaa resurssien säästymiseen (lähettäminen vie enemmän energiaa kuin koodaus)
  - Paketin koodaukseen on käytetty hajautusta, jolloin myös tietoturva paranee, mikäli kuuntelija ei tiedä käytettyä algoritmia



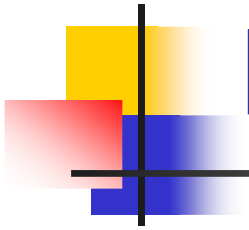
# Taustatyö

- HRS (Hop-by-Hop Reliability Support Scheme)
  - Hyppykohtaisiin kuittauksiin perustuva menetelmä
  - Ottaa huomioon reititysmuutokset ja on skaalautuva
    - Hyppykohtaisia sekvenssejä ylläpidetään ainoastaan lähimmille naapureille
  - Unicast- ja broadcast-toimintatavat erikseen
    - Unicastissa uusi solmu voi liittyä suoraan, koska sekvenssi alkaa nollasta
    - Broadcastissa first-packet-bit kertoo uusille solmuille, että paketteja ei ole jäänyt välistä
  - Perustuu NACKeihin
    - Lähetetään negatiivinen kuittaus, kun huomataan aukko sekvenssissä
  - Yksittäisiä tai sekvenssin viimeisiä paketteja varten hACK (hop-by-hop ACK) toimintatapa



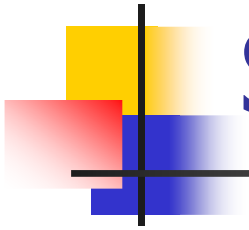
# HBH

- HBH (hop-by-hop) on hyppykohtaisiin sekvenssinumeroihin perustuva tiedonsiirron varmistusmenetelmä
- Mallinnettu ja simuloitu OPNet Modeler ohjelmalla, proof-of-concept
- Perusperiaate melko lähellä HRS:ää
- Negatiiviset kuittaukset sekvenssissä olevia aukkoja varten ja hACK-toiminta (hop-by-hop ACK) sekvenssien viimeisiä tai yksittäisiä paketteja varten



# HBH

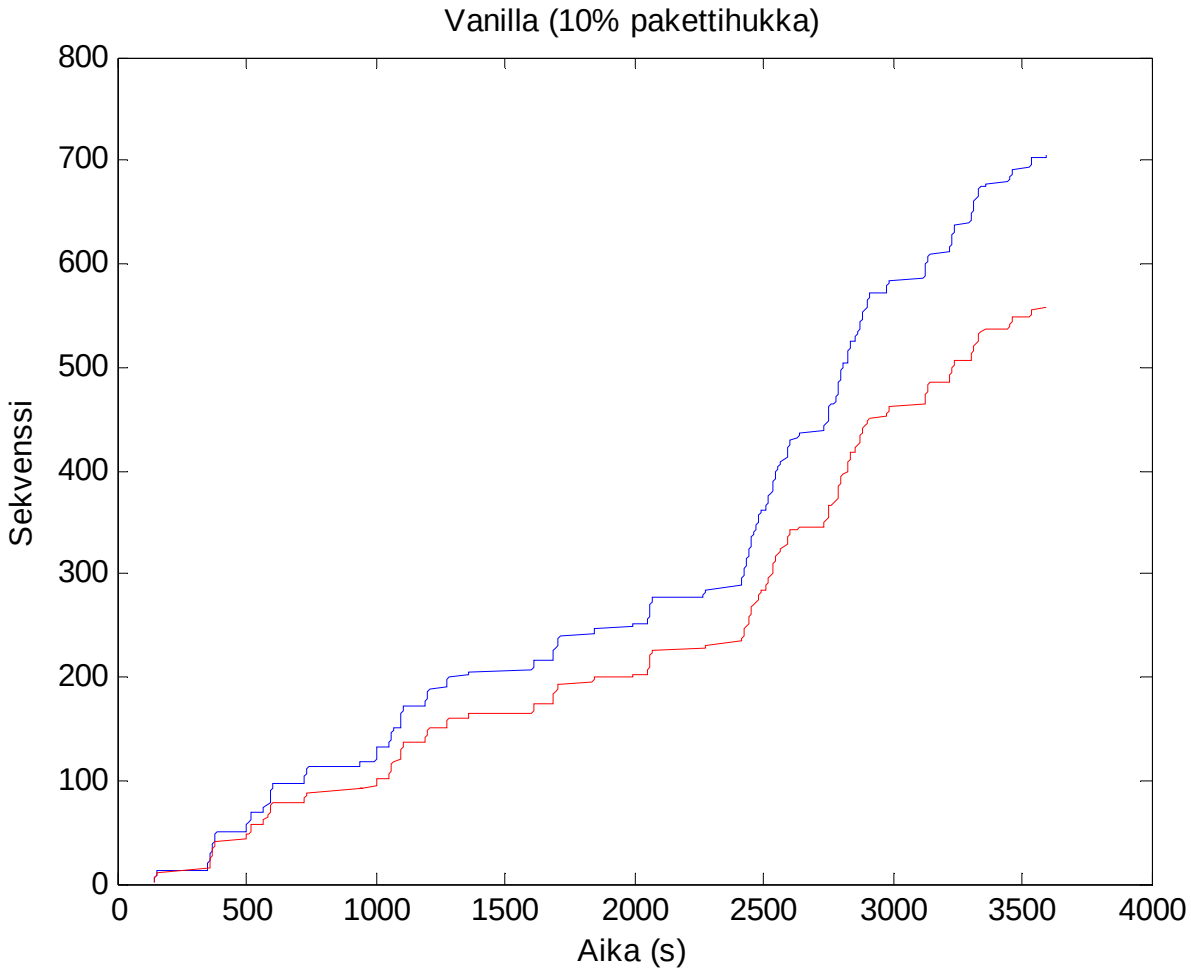
- Solmut ylläpitävät lähetys- ja vastaanotto-sekvenssejä lähimmille naapureilleen
- Sekvenssinumero leimataan pakettiin ja paketti siirretään lähetysjonoon
- Vastaanottaja tarkistaa sekvenssinumeron ja varmistaa, että sekvenssi on jatkuva
- Jos välistä puuttuu paketteja, lähetetään NACKit puuttuvista paketeista, joihin vastauksena puuttuvat paketit lähetetään uudelleen
- Viimeisen paketin jälkeen ei tule uusia paketteja, jotka voisivat laukaista NACK-mekanismin, joten lyhyen ajan kuluttua lähetetään hACK-viesti kuittauksena viimeisimmästä paketista
- Jos hACKia ei kuulu, viimeisen paketin oletetaan kadonneen ja se lähetetään uudelleen (niin monta kertaa, että hACK saadaan)



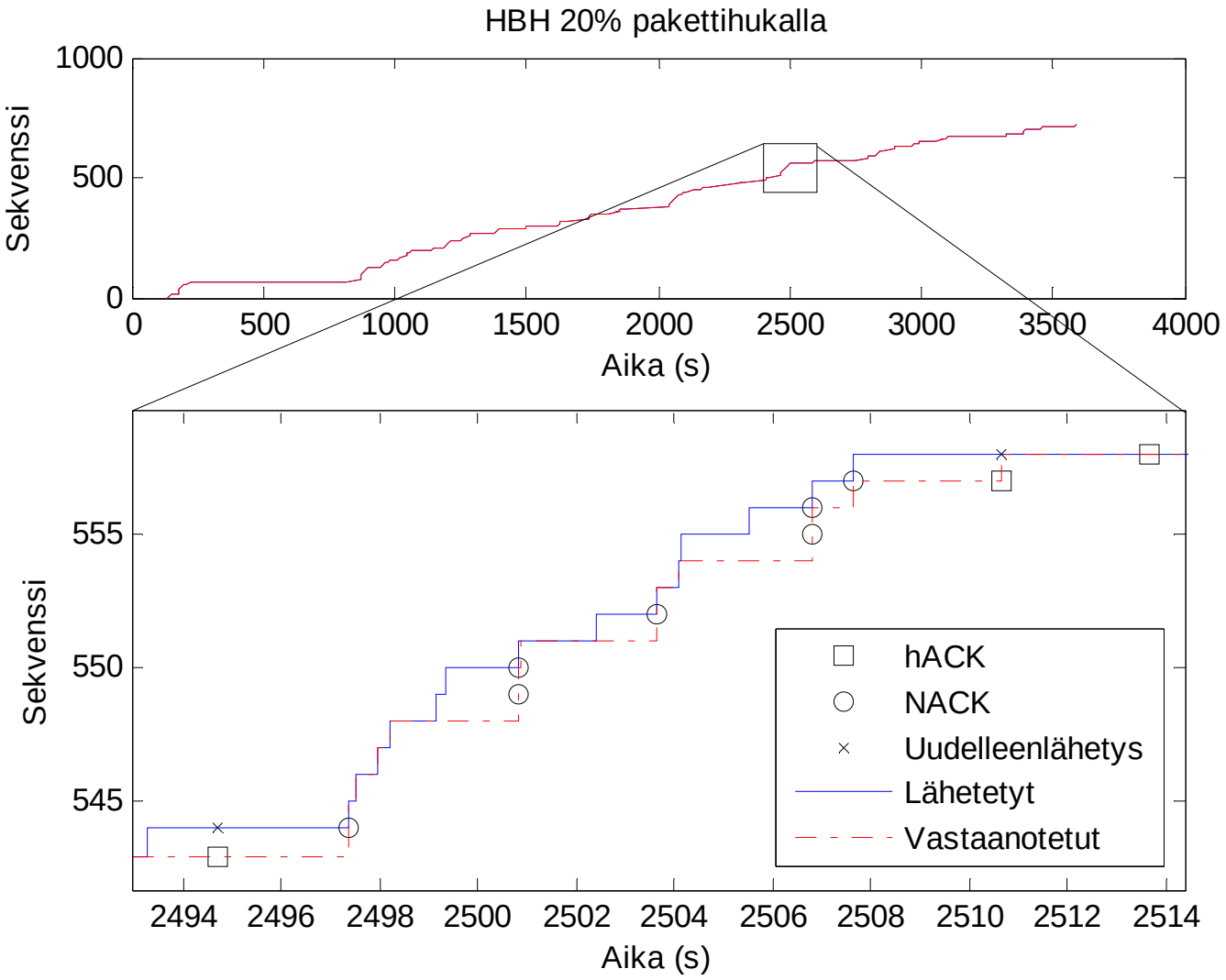
# Simulointiskenaariot

- Simuloinneissa menetelmän toimintaa tutkittiin eri kokoisilla pakettihukilla
  - 0%, 5%, 10%, 20%, 30%, 50%, 80%
  - Paketteja hukattiin satunnaisesti sieltä täältä, ei tahallisina purskeina
- Vertailun vuoksi simuloitiin myös liikennettä ilman menetelmän käyttöä
- Asetelmassa kaksi solmua kommunikoi keskenään satunnaisilla lähetyspurskeilla
  - Lähetys- ja vastaanottosekvenssien suhdetta verrattiin ajan funktiona
  - Tavoitteena selvittää, palautuuko sekvenssi pakettihukan jälkeen

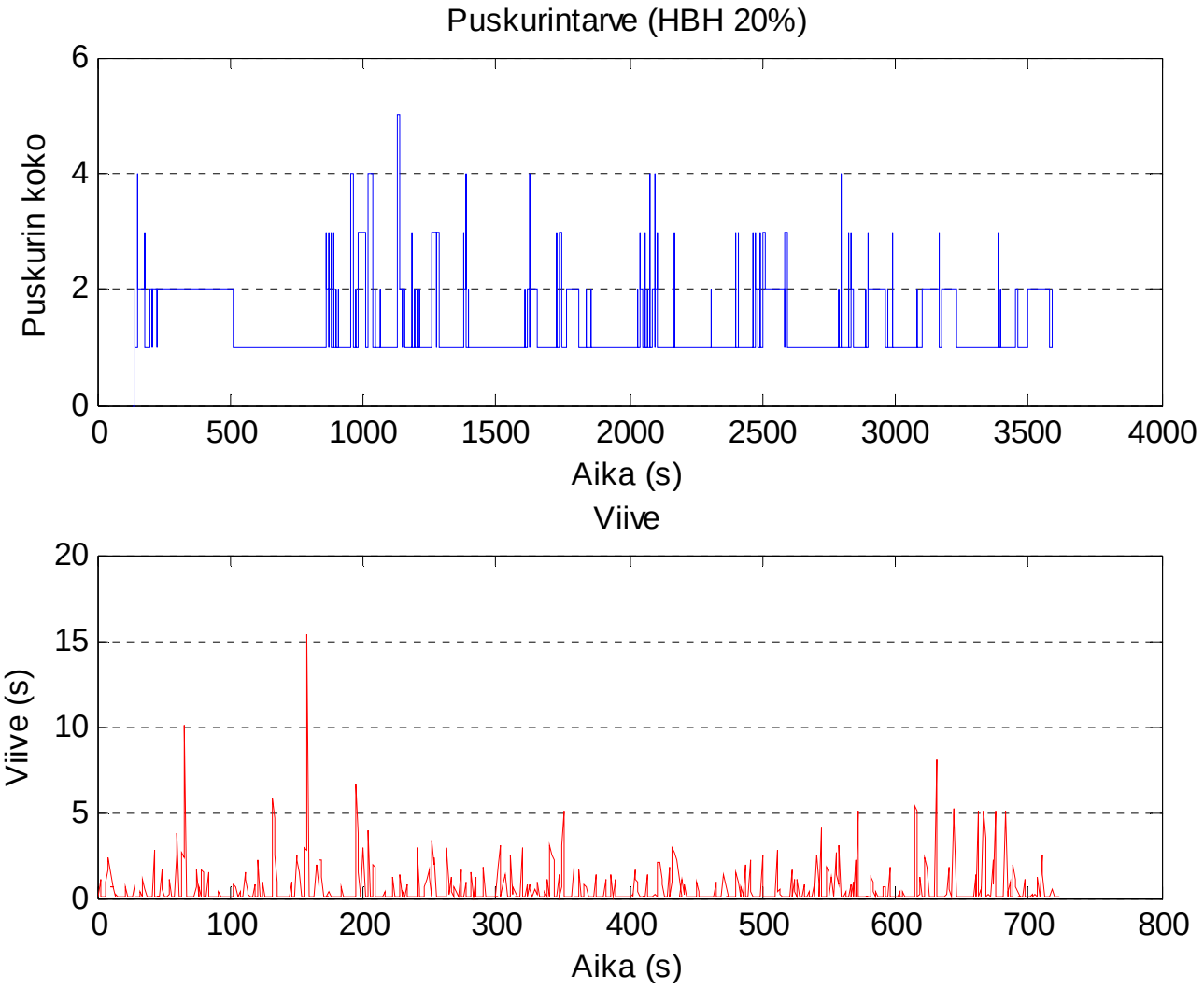
# Tulokset


Vanilla (10% pakettihukka)
Sekvenssi
Aika (s)
0
100
200
300
400
500
600
700
800
0
500
1000
1500
2000
2500
3000
3500
4000

# Tulokset


HBH 20% pakettihukalla
Sekvenssi
1000
500
0
0
500
1000
1500
2000
2500
3000
3500
4000
Aika (s)
Sekvenssi
555
550
545
2494
2496
2498
2500
2502
2504
2506
2508
2510
2512
2514
Aika (s)
hACK
NACK
Uudelleenlähetys
Lähetetyt
Vastaanotetut

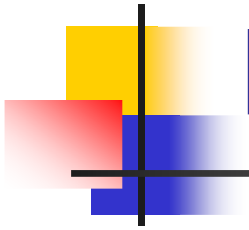
# Tulokset


Puskurintarve (HBH 20%)
Puskurin koko
0
2
4
6
0
500
1000
1500
2000
2500
3000
3500
4000
Aika (s)
Viive
Viive (s)
0
5
10
15
20
0
100
200
300
400
500
600
700
800
Aika (s)

# Yhteenveto

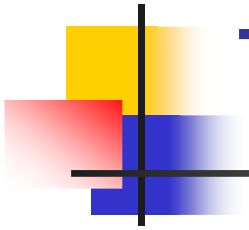
- Pakettihukan kasvaessa kommunikointi ilman varmistusta käy hankalaksi
- Ohessa eri pakettihukkien aiheuttaman overheadin määrä ja vaadittava puskurointi ja aiheutunut viive
- Kohtalaisilla pakettihukilla menetelmä toimii hyvin, mutta suuren pakettihukan tapauksessa ylimääräisen hallintatiedon määrä kasvaa sietämättömäksi
- Yli 50% pakettihukka on todella karu toimintaympäristö mille tahansa menetelmälle

| Pakettihukka | Overhead | Viive(max) | Puskuri |
|---|---|---|---|
| 0% | 30% | 2-5s | 3 |
| 5% | 50% | 2-8s | 4 |
| 10% | 60% | 3-8s | 4 |
| 20% | 90% | 5-15s | 5 |
| 30% | 140% | 5-15s | 7 |
| 50% | 280% | 20-50s | 15 |
| 80% | 1500% | N/A | N/A |



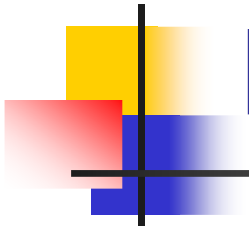
# Päätelmät

- Pienillä ja kohtuullisilla pakettihukilla toteuttaa luotettavan tiedonsiirron hyvin
- Suurella pakettihukalla menetelmä ei pysy enää kovin stabiilina
  - Ratkaisuna NACKeja voisi aggregoida yhteen, jolloin niitä ei lähetettäisi suuria määriä
  - Toisaalta erittäin huonoissa olosuhteissa paketteja voitaisiin lähettää ”yksitellen”, jolloin siirtokapasiteetti rajoittuisi tosin murto-osaan entisestä
  - Kysymys on kompromissista siirtonopeuden ja luotettavuuden välillä
- Välisolmuilla oltava store-and-forward –tyyppinen puskurointi suuren pakettihukan verkkoihin, jotta paikkaukset voidaan tehdä edelliseltä hypyltä



# Tulevaisuus

- IP-otsikon id-kenttä sisältää tunnisteen sirpalointia varten
  - Jos sirpalointi ei ole käytössä, id-kenttää voi käyttää esimerkiksi hyppykohtaisten sekvenssinumeroiden leimaamiseen
  - Iptablesissa on jo valmiit funktiot IP-otsikon kenttien käsittelyyn
- Mainitut puutteet tulisi myös korjata
  - NACKien aggregointi, kapasiteetin mittaus, usean hypyn toiminta, yms.



# Lähteitä


- HRS: H.Lee, Y.Ko, D.Lee: A Hop-by-hop Reliability support Scheme for Wireless Sensor Networks, IEEE Pervasive Computing and Communications Workshops, 2006 (PERCOMW'06).
- FBcast: R.Kumar, A.Paul, U.Ramachandran, D.Kotz: On Improving Wireless Broadcast Reliability of Sensor Networks Using Erasure Codes, INFOCOM 2005, Georgia Institute of Technology.
- PSFQ: C.Wan, A.T.Campbell, L.Krishnamurthy: Pump-Slowly, Fetch-Quickly (PSFQ): A Reliable Transport Protocol for Sensor Networks, IEEE Journal on Selected Areas in Communications, April 2005.
- ESRT: Ö.B.Akan, I.F.Akyildiz: Event-to-Sink Reliable Transport in Wireless Sensor Networks, IEEE/ACM Transactions on Networking, Oct. 2005.