

FTSP tidssynkronisering for trådløs sensornettverk – evaluering og tilpasning

Vinh Pham og Erlend Larsen

Forsvarets forskningsinstitutt (FFI)

1. oktober 2012

FFI-rapport 2012/02278

1141/3836

P: ISBN 978-82-464-2201-5

E: ISBN 978-82-464-2202-2

Emneord

Trådløse sensornettverk

Tidssynkronisering

Klokkesynkronisering

FTSP

Godkjent av

Torunn Øvreås

Prosjektleder

Anders Eggen

Avdelingssjef

Sammendrag

Trådløse sensornettverk (WSN) har mange potensielle nyttige anvendelser både i militær sammenheng og i det sivile samfunn. Eksempler på dette er overvåking og perimetersikring, helse relaterte sensornettverk eller sensornettverk i industrielle systemer.

I anvendelser der sensordata fra multiple sensorer kombineres (datafusjon), vil det være nødvendig at alle noder i nettverket har en felles forståelse av tiden for å kunne danne et riktig og helhetlig bilde av situasjonen og hendelsesforløpet. Et eksempel er ved målfølging av fiendtlige inntrengere, hvor bruk av multiple tidssynkroniserte sensorer er avgjørende for å presist kunne tracke og gjengi bevegelsesmønsteret.

Siden et sensornettverk er et distribuert system, vil det være nødvendig å bruke en protokoll som ivaretar synkronisering av tiden. Flooding Time Synchronization Protocol (FTSP) er et eksempel på en slik protokoll som er spesielt utviklet for bruk i et sensornettverk. Denne rapporten evaluerer ytelsen til FTSP samt utforsker muligheter for forbedring av protokollen. Motivasjonen er å vurdere om protokollen er egnet som tidssynkroniseringsprotokoll for generell bruk i militære anvendelser og mer spesifikt i to ulike konkrete scenarioer som perimetersikring og overvåking av veiakse. Disse scenarioene er benyttet i Situational Awareness Sensor Systems (SASS) prosjektet som denne rapporten er en del av.

English summary

Wireless sensor networks (WSN) have many potential useful applications both in military and civilian context. Examples include battlefield surveillance and perimeter security, health care and sensor networks in industrial systems.

In a number of applications, where data from multiple sensors are combined (datafusion), a common notion about the time among the sensor nodes is critical in order to render a correct picture of the situation and the course of events. A typical example is target tracking of hostile intruders, where the usage of multiple sensors that are synchronized is critical in order to precisely track and render the movement pattern.

Since a WSN is basically a distributed system, it is necessary to apply a synchronization protocol in order handle time synchronization in the network. Flooding Time Synchronization Protocol (FTSP) is an example of such a protocol with this capability, specifically designed for use in WSN. This report evaluates the performance of FTSP, and explores the possibility to improve the protocol. The motivation is to determine whether this protocol is a valid candidate as a time synchronization protocol for general usage in military application, as well as in two specific scenarios i.e. perimeter security and road surveillance. These scenarios are related to the Situational Awareness Sensor Systems (SASS) project in which this report is part of.

Innhold

1	Innledning	7
1.1	Bakgrunn	7
1.2	Motivasjon	7
2	Tidssynkronisering i distribuerte systemer	8
2.1	Protokoller for tidssynkronisering	8
2.2	FTSP - Flooding Time Synchronization Protocol	8
3	Evaluering av FTSP	11
3.1	Motivasjon for evalueringen	11
3.2	Testoppsett	11
3.3	Evalueringsresultater	13
3.3.1	Test 1. $P = 10$ sekunder	14
3.3.2	Test 2. $P = 1$ minutt	15
3.3.3	Test 3. $P = 5$ minutter	16
3.4	Diskusjon av måleresultater	17
4	Modifisering av FTSP	18
4.1	Beskrivelse av modifisert FTSP	18
4.2	Evaluering av modifisert FTSP, Test 4	19
5	Oppsummering og konklusjon	21
	Akronymer	23

1 Innledning

1.1 Bakgrunn

I prosjekt 1141 Situational Awareness Sensor Systems (SASS) som ble startet i 2008 arbeides det med å ta i bruk trådløs sensor nettverk (WSN) teknologi i militære anvendelser. Et trådløst sensornettverk (WSN) er en samling av sensornoder som er deployert over et geografisk område. Nodene i nettverket samarbeider med hverandre i å overvåke området ved at de dynamisk danner ett trådløst multihopp nettverk. Dette muliggjør at sensordata fra hver node kan sendes gjennom nettverket til et sentralt innsamlingspunkt i nettverket. Dette innsamlingspunktet kalles en sinknode.

Forskningen innen WSN teknologien ble startet av det amerikanske forsvarrets forskningsinstitutt Defense Advanced Research Projects Agency (DARPA) i 1978 [1-3]. Motivasjonen var overvåking i slagmark og deteksjon av snikskyttere. I dag er WSN teknologien ikke bare anvendt i militære situasjoner, men også i sivile og industrielle områder som for eksempel overvåking og kontroll av industrielle prosesser, og helse relaterte anvendelser [4-6]. I SASS prosjektet har fokuset vært å anvende WSN teknologien i perimetersikring, og overvåking av veiakse [7;8].

1.2 Motivasjon

Et WSN består av mange noder, gjerne fra titalls til flere tusen noder. Sensordata fra multiple sensorer blir som regel kombinert og prosessert for å gi en felles konklusjon eller et helhetlig bilde av det området som WSN dekker. I denne sammenheng spiller tidssynkronisering en viktig rolle med tanke på:

- riktig bilde av hendelsesforløpet
- data-logging
- målfølgning
- koordinert aktivering

Motivasjonen i arbeidet som beskrives i denne rapporten er både å få bedre teoretisk kunnskap om tidssynkronisering, samt å undersøke hvor godt tidssynkronisering fungerer i praksis. Vi tar for oss en spesifikk protokoll kalt Flooding Time Synchronization Protocol (FTSP) [9].

Protokollen finnes ferdig implementert i TinyOS [10]. Målet er å evaluere ytelsen til denne protokollen med tanke på *konvergeringstid* og *presisjon* på tidssynkroniseringen. Dette gir så grunnlag for å bestemme om dette er en egnet tidssynkroniseringsprotokoll for SASS prosjektet, og om nødvendig, hvilke endringer som må til for å tilpasse protokollen i forhold til behovene i prosjektet.

2 Tidssynkronisering i distribuerte systemer

2.1 Protokoller for tidssynkronisering

Tidssynkronisering i distribuerte systemer er generelt basert på utveksling av synkroniseringsmeldinger mellom noder. Disse meldingene inneholder som regel tidsstempler både på avsender og mottaker-siden, og gir grunnlag for korreksjon av tidsavviket mellom noder. Gjennom de siste tiårene er det utviklet en rekke protokoller for tidssynkronisering, både for trådbaserte og trådløse nettverk. Network Time Protocol (NTP) er den mest utbredte protokollen i trådbaserte nettverk. Prinsippet i denne protokollen er at klienter synkroniserer sin klokke med en NTP server som igjen er synkronisert med UTC tid (Coordinated Universal Time), dvs. ”universelle tiden” basert på gjennomsnittet fra en rekke atomur. Ved tidssynkronisering mellom en klient og en NTP server, foretas det statistiske analyser av round-trip tiden på synkroniseringsmeldinger som utveksles. Basert på denne metoden er det mulig å oppnå nøyaktigheter på millisekunds-nivå. NTP er derimot ikke like egnet i trådløse nettverk, pga. usikkerheten i transmisjonstiden indusert av MAC protokollen som kan forårsake forsinkelser på noen hundre millisekund pr. hopp. Av denne grunn er det derfor utviklet en rekke tidssynkroniseringsprotokoller skreddersydd for trådløse nettverk. Eksempler på slike protokoller er:

- RBS Reference Broadcast Synchronization (RB) [11]
- Timing-Sync Protocol for Sensor Networks (TPSN) [12]
- Flooding Time Synchronization Protocol (FTSP) [9]

En gjennomgang av disse protokollene er utenfor omfanget til denne rapport. For de interesserte henvises det til de oppgitte referansene for en mer detaljert beskrivelse. I denne rapporten vil fokuset først og fremst være på FTSP der en grundig beskrivelse gis i neste avsnitt.

2.2 FTSP - Flooding Time Synchronization Protocol

FTSP er en tidssynkroniseringsprotokoll utviklet spesielt for sensornettverk. Protokollen muliggjør synkronisering av klokken mellom noder i nettverket, enten over et hopp eller over multihopp. I tillegg er protokollen designet for å være feiltolerant og ressursbesparende med hensyn på energi og båndbredde. Disse egenskapene er spesielt viktig i trådløse nettverk der det generelt er både lavere båndbredde og høyere dynamikk sammenlignet med trådbaserte nettverk. Med større feiltoleranse innebærer det at protokollen skal fortsatt kunne fungere tilfredsstillende selv når forholdene i nettverket endres som medfører til endringer i topologien. Eksempler på årsaker som fører til endringer i topologien er:

- mobilitet
- nye noder som slutter seg til eller eksisterende noder som forlater nettverket
- linkfeil

I FTSP velges det ut en node som påtar seg rollen som masternode. Klokken på denne noden vil da fungere som referanseklokke i nettverket. Dette innebærer at de øvrige nodene i nettverket, dvs. slavenodene, må synkronisere sine klokker med masternoden sin klokke for å oppnå et felles begrep om tiden i nettverket. Denne synkroniseringen skjer enten direkte eller indirekte via intermediære noder.

Utvelgelsen av masternoden foregår dynamisk, og forutsetter at alle nodene i nettverket har unik ID. I utgangspunktet er det den noden med lavest ID som blir valgt som masternoden. Dersom denne noden av en eller annen grunn faller bort, vil noden med lavest ID blant de gjenværende nodene bli valgt som ny masternode. Denne utvelgelsesmåten sikrer med andre ord at FTSP er mer feiltolerant ovenfor endringer som kan oppstå i nettverket enn om nettverket har en fast masternode.

Tidssynkroniseringen i nettverket utføres ved hjelp av periodiske utvekslinger av synkroniseringsmeldinger mellom noder. En synkroniseringsmelding kringkastes (1 hopp) over radiogrensesnittet fra en sendernode til alle naboroder, og inneholder i hovedsak to tidsstempler, ett som settes på sendersiden og ett som settes på mottakersiden. Tidsstemplet t_A som settes på en sender node A representerer den *globale* tiden, dvs. den tiden som er felles for alle noder i nettverket. Dette betyr at kun noder som først har synkronisert sin klokke vil kunne kringkaste synkroniseringsmeldinger. Tidsstemplet t_B som settes på en mottakernode B er den lokale tiden, dvs. systemklokken på mottakernoden. Med andre ord, så må hver node holde på to forskjellige ”klokker”, ett som er global og ett som er lokalt. Videre så foregår tidsstemplingen på MAC-laget for å eliminere forsinkelser knyttet til for eksempel medieaksess.

Ett tidsstempel par $\langle t_A, t_B \rangle$ utgjør et referansepunkt i tid som muliggjør tidssynkroniseringen mellom mottakernode B med sendernode A. I prinsippet vil ett referansepunkt være tilstrekkelig for å synkronisere tiden mellom to noder dersom det ikke er drifting på klokkene. Basert på dette referansepunktet så kan mottaker B estimere relasjonen mellom global tid (t_A) og lokal tid (t_B) ved å bruke likning (2.1):

$$t_A = t_B + \text{offset} \quad (2.1)$$

Variabelen *offset* representerer forskyvningen i tid mellom den lokale og den globale tiden. Merk at i dette uttrykket, så er propagasjonsforsinkelsen t_p på synkroniseringsmeldingen fra A til B neglisjert. Dette kan forsvares med at den teoretiske forsinkelsen er mindre enn 1 μ s når avstanden mellom A og B er mindre enn 300 m. I praksis vil avstanden mellom de sensornodene som vi har brukt i prosjektet være vesentlig mindre, dvs. 10-20 m, hvilket vil bety en betydelig lavere propagasjonsforsinkelse enn 1 μ s.

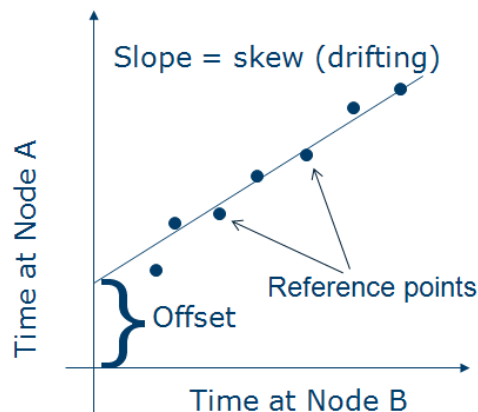
Dessverre er det slik at klokken på de forskjellige nodene alltid vil drifte i forhold til hverandre, pga. avvik i for eksempel:

- klokkekristallet
- temperaturen
- spenningen

For å kompensere for klokke-driften er det derfor nødvendig med periodiske utvekslinger av synkroniseringsmeldinger slik at multiple referansepunkter tas i bruk i synkroniseringsprosessen. Figur 2.1 illustrerer prinsippet for hvordan dette gjøres. Ved bruk av multiple referansepunkter samt lineær regresjon kan en uttrykke relasjonen mellom t_A og t_B som en lineær funksjon:

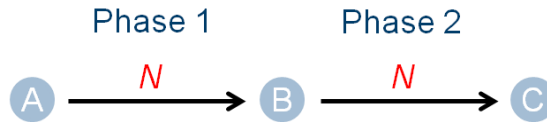
$$t_A = skew \cdot t_B + offset \quad (2.2)$$

der *skew* representerer driften på klokken, og er ekvivalent med stigningen på den lineære kurven. Dersom node A og B startes opp helt nøyaktig på samme tidspunkt slik at systemklokken er synkron ved oppstart så vil offset være lik null. Likeledes dersom det er null drift mellom klokkene hos node A og B så vil *skew* være lik 1, slik at $t_A = t_B$.



Figur 2.1 Prinsipp for tidssynkronisering ved bruk av multiple referansepunkter

Figur 2.2 illustrerer hvordan synkroniseringsprosessen foregår i ett 2 hopps nettverk. Anta i dette eksemplet at node A har lavest ID og vil derfor være masternoden. Node A kringkaster synkroniseringsmeldinger med perioden P , som kun node B kan motta. Etter at B har mottatt ett tilstrekkelig antall N meldinger eller referansepunkter, vil B da være i stand til å kunne synkronisere seg med A. Først når B er synkronisert med A, vil den selv begynne å kringkaste synkroniseringsmeldinger videre til C. C kan nå synkronisere sin klokke indirekte med masternode A, ved hjelp av synkroniseringsmeldinger fra B.



Figur 2.2 Tidssynkronisering over 2 hopp

Dette eksemplet illustrerer at synkroniseringsprosessen i nettverket skjer etappevis, og begynner med de nodene som er i umiddelbar nærhet til masternoden. Etter hvert vil denne prosessen gradvis forplante seg utover til de øvrige nodene i nettverket. Dette innebærer at det vil ta en viss tid før alle noder i hele nettverket er synkronisert med master noden. Denne tiden kalles for konvergeringstiden, og er blant annet en funksjon av radiusen R på nettverket, dvs. den lengste avstanden, målt i antall hopp, fra masternoden til en node på randen av nettverket. Gitt at det er ingen pakketap på pakkeoverføringen, så kan man uttrykke konvergeringstiden for et generelt tilfelle som:

$$t_{\text{Konvergenstid}} \in [C + P \cdot (N - 1) \cdot R, C + P \cdot N \cdot R] \quad (2.3)$$

der C representerer den initielle forsinkelsen ved oppstart (dvs. den tiden som alle noder må vente før de kan begynne å kringkaste synkroniseringsmeldinger). Andre ledd representerer den minste/maksimale tiden det tar for alle noder i nettverket å bli synkronisert med masternoden. Her er P lik perioden på synkroniseringsmeldinger, N er minste antall synkroniseringsmeldinger for å synkronisere, og R er radius målt i antall hopp.

3 Evaluering av FTSP

3.1 Motivasjon for evalueringen

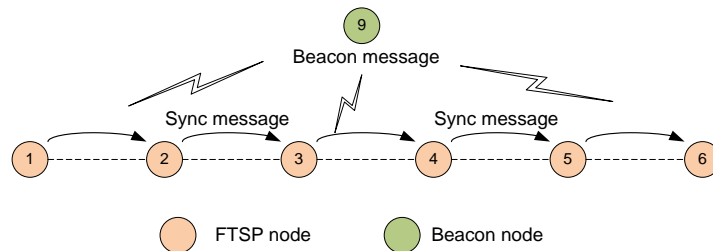
Motivasjonen med denne evalueringen er å bestemme om FTSP er egnet som en tidssynkronisering protokoll for de valgte scenarioene i SASS prosjektet, men det er sannsynlig å anta at resultatet vil også være gjeldende i andre scenarioer eller i en generell anvendelse av WSN teknologien. Evalueringen tar sikte på å måle følgende egenskaper ved protokollen:

- Konvergeringstid
- Presisjonen på tidssynkroniseringen

3.2 Testoppsett

For å evaluere ytelsen til FTSP protokollen med tanke på konvergeringstid og presisjonen er det satt opp ett enkelt testoppsett bestående av 6 noder (ID 1-6). Disse nodene kjører FTSP og er arrangert i en linjetopologi som vist i Figur 3.1. Masternoden er den noden med lavest ID, dvs. node 1. I tillegg er det brukt en beacon-node (ID 9) som trigger måling av ytelsen til tidssynkroniseringen. Dette oppnås ved at beacon-noden kringkaster periodiske beacon-

meldinger. Beacon-noden er konfigurert til å sende med høyest mulig effekt, mens de øvrige FTSP-nodene sender med laveste effekt. Hensikten er å gjøre nettverket så lite som mulig slik at en beacon-melding som kringkastes kan mottas av alle FTSP noder. Ved å logge tidspunktet (global tid) for når meldingen mottas ved hver node og sammenligne resultatet fra alle nodene, så kan en ganske enkelt bestemme presisjonen på tidssynkroniseringen. Siden en beacon-melding mottas av alle noder på nøyaktig samme tidspunkt, så vil disse rapportere samme tidspunkt dersom de er synkroniserte med hverandre. Eventuelle avvik i tid vil da representere unøyaktigheten i klokkesynkroniseringen.



Figur 3.1 Testoppsett for evaluering av FTSP protokollen



Figur 3.2 Testoppsett bestående av FTSP noder (1-6), beacon-node (9) og sniffer-node(10).

Figur 3.2 viser bildet av testoppsettet. Tallene på figuren representerer node ID på de enkelte nodene. I tillegg til de nodene som er nevnt tidligere, så er det også brukt en sniffer-node (BaseStation) med ID 10. Denne brukes for å monitorere trafikken som sendes trådløst mellom nodene. Merk videre at nodene er koblet til en PC via USB-kabler.

Dette har en rekke praktiske fordeler med tanke på:

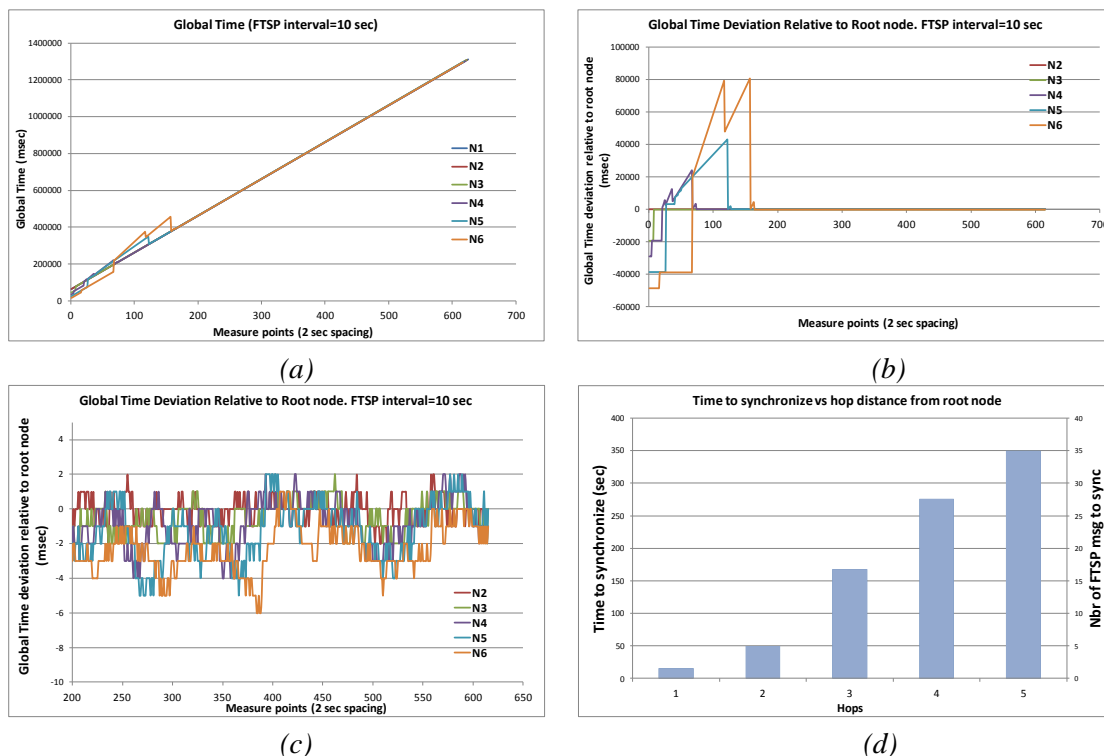
- omprogrammering av nodene
- logging av data kan gjøres direkte til PC'en
- ingen behov for batterier

Videre så kjører logg-PC'en der alle noder er koblet til, en NTP-klient som sikrer at systemklokken på PC'en regelmessig synkroniseres med UTC klokken på en NTP-server. Dette er nødvendig for å måle drifting av sensor-klokkene i forhold til "sann tid".

3.3 Målekampanje og resultater

I evalueringen av FTSP er det gjort flere eksperimenter med forskjellige periode P på synkroniseringsmeldingen. Hensikten med dette er å undersøke hvordan denne parameteren påvirker ytelsen til FTSP med tanke på presisjon og konvergeringstid. Dette gir grunnlag for å fastsette optimale innstillinger for senere anvendelser. Det er blitt gjort forsøk med følgende innstillinger på perioden P :

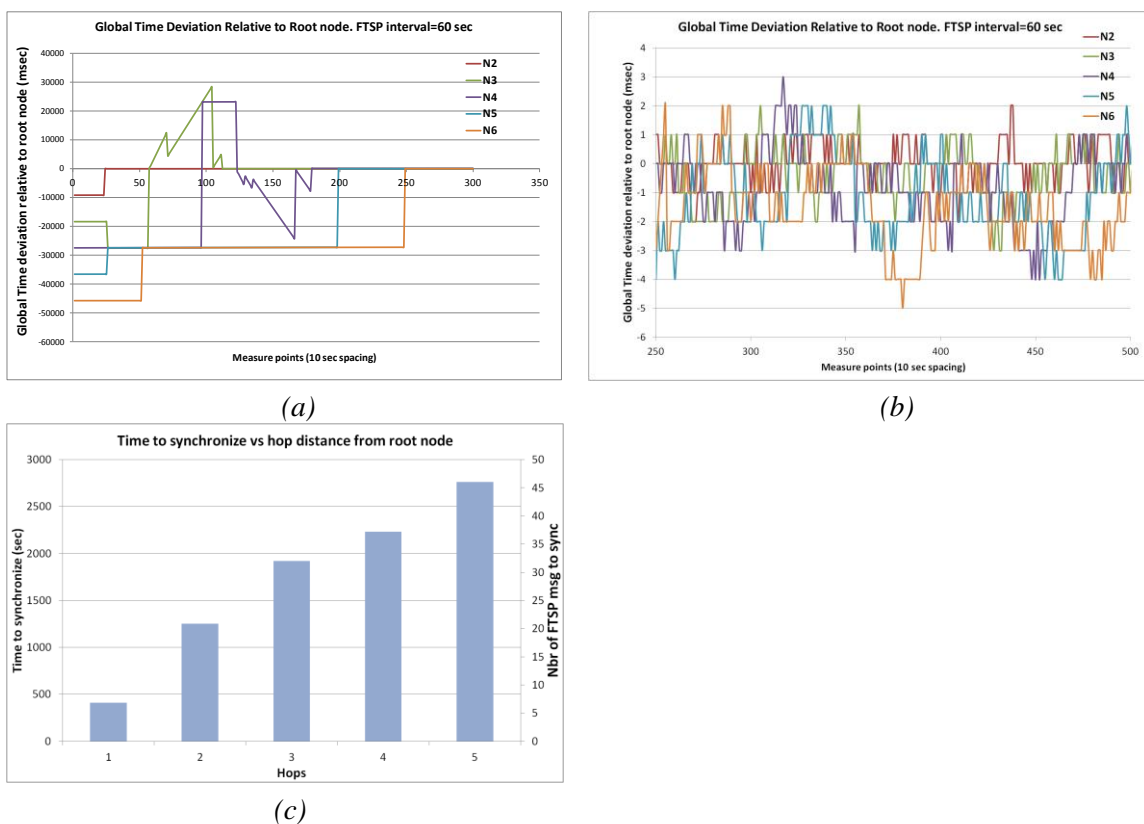
- 10 sekunder
- 1 minutt
- 5 minutter



Figur 3.3 Måleresultat der $P=10$ sek. (a) Global tid. (b) Relativt avvik i global tid. (c) Relativt avvik i global tid i den persistente fasen. (d) Konvergeringstid som funksjon av avstanden fra masternoden.

3.3.1 Test 1. $P = 10$ sekunder

Figur 3.3 viser måleresultatet for Test 1, der perioden P er konfigurert til å være 10 sekunder. Her er samplingsintervallet, dvs. intervallet på beacon-meldinger, satt til å være 2 sekunder. Figur 3.3(a) viser den globale tiden (y-aksen) som funksjon av samplingspunkter (x-aksen). Fra figuren kan vi dele synkroniseringsprosessen inn i to faser, en initiell transient fase og en persistent fase. I den transiente fasen mellom $t_s=0$ til ca. $t_s=160$ (dvs. ca. 320 sekunder) ser vi at den globale tiden på de enkelte nodene til å begynne med, har et avvik i forhold til masternodens klokke. Men i løpet av den transiente fasen vil nodene etter hvert komme i synk med masternoden. Vi ser av figuren at denne prosessen skjer gradvis og starter med de nodene som er nærmest masternoden og forplanter seg etter hvert til noder lenger ut i nettverket. Resultatet stemmer godt overens med tidligere forklaringer i avsnitt 2.2. Dette kan ses mer tydelig i Figur 3.3(b) som viser avviket i global tid relativt til masternoden. Her ser vi at node 2 er den noden som først kommer i synk med masternoden, deretter er det node 3 og så node 4 som kommer i synk, osv. Node 6 er den siste noden som synkroniserer sin klokke med masternoden ved $t_s=160$. Denne tiden representerer også konvergeringstiden, dvs. den tiden det tar for alle noder i nettverket å komme i synk med masternoden. Merk at Figur 3.3 (a) og (b) er nesten like, men ved å presentere resultatet som relativt avvik i forhold til den globale tiden som gjort i Figur 3.3(b), vil detaljene i den initielle fasen blir med tydelig. Av den grunn vil senere resultater kun bli presentert på samme form som i Figur 3.3(b) og ikke som i Figur 3.3(a).



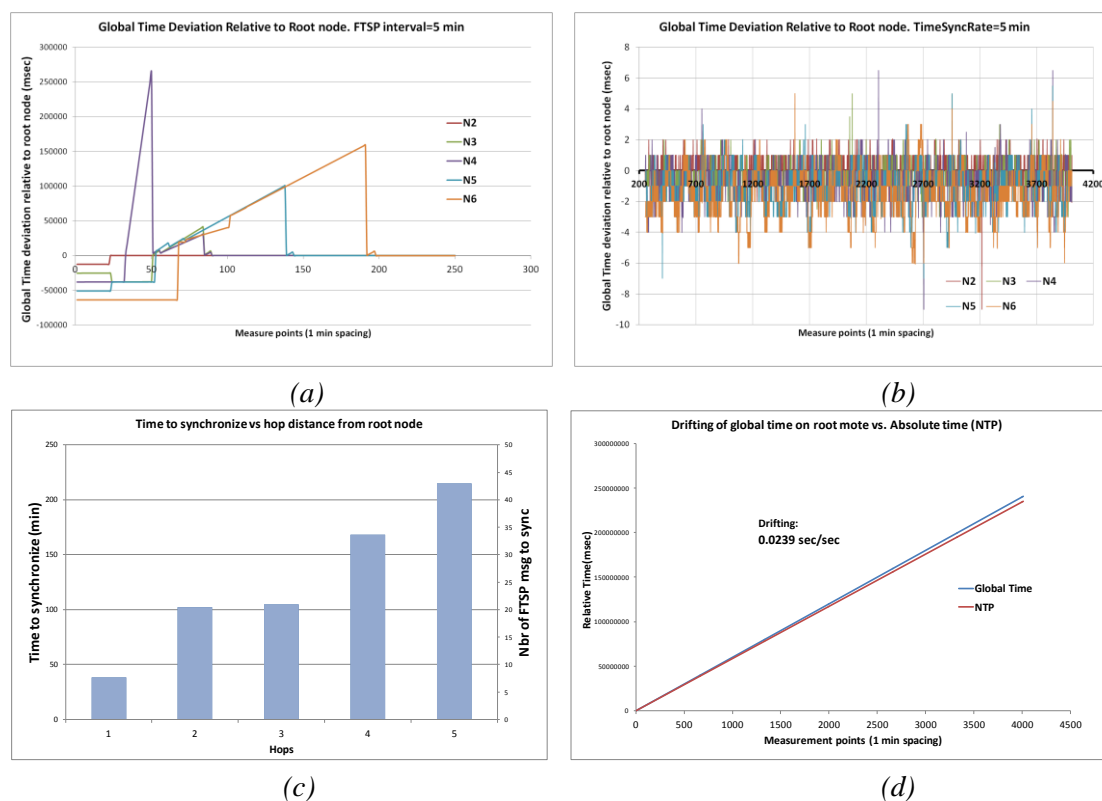
Figur 3.4 Måleresultat der $P=1\text{min}$. (a) Relativt avvik i global tid i den transiente fasen. (b) Relativt avvik i global tid i den persistente fasen. (c) Konvergeringstid som funksjon av avstanden fra masternoden.

Figur 3.3(c) viser relativ avvik i global tid i den persistente fasen. Her ser vi at avviket i global tid er mellom -6 ms til 2 ms. Av denne figuren kan vi også se at avviket varierer i noe større grad for noder som er lengst unna masternoden sammenlignet med de som er nærmere. Dette impliserer at presisjonen reduseres med økende avstand fra masternoden.

Figur 3.3(d) viser konvergeringstiden som funksjon av avstanden fra masternoden. Denne tiden er registrert ved å observere den globale tiden og interne variabler (IS_SYNC flagg og masternode ID) i FTSP algoritmen. Dette gir ett mer presist bilde for når en node faktisk anser seg selv som synkronisert med masternoden. Her ser vi som forventet at konvergeringstiden øker med antall hopp. For node 2 som er kun 1 hopp unna masternoden, er konvergeringstiden relativt liten ca. 8 samplingsperioder hvilket tilsvarer 16 sekunder. For node 6 som er 5 hopp unna masternoden er konvergeringstiden vesentlig lenger ca. 350 sekunder. Legg merke til at denne tiden er noe lengre enn hva resultatet i Figur 3.3 (a) og (b) tilsier.

3.3.2 Test 2. $P = 1$ minutt

Figur 3.4 viser måleresultatet for Test 2 der $P = 1$ minutt. Samplingsintervallet her er 10 sekunder. Figur 3.4(a) viser relativ avvik i global tid i den transiente fasen. Det mest interessante i denne figuren er konvergeringstiden som ligger på rundt 250 samplingsintervaller, hvilket tilsvarer $t=2500$ sekunder. Videre, så viser Figur 3.4(b) relativ avvik i global tid i den persistente fasen.

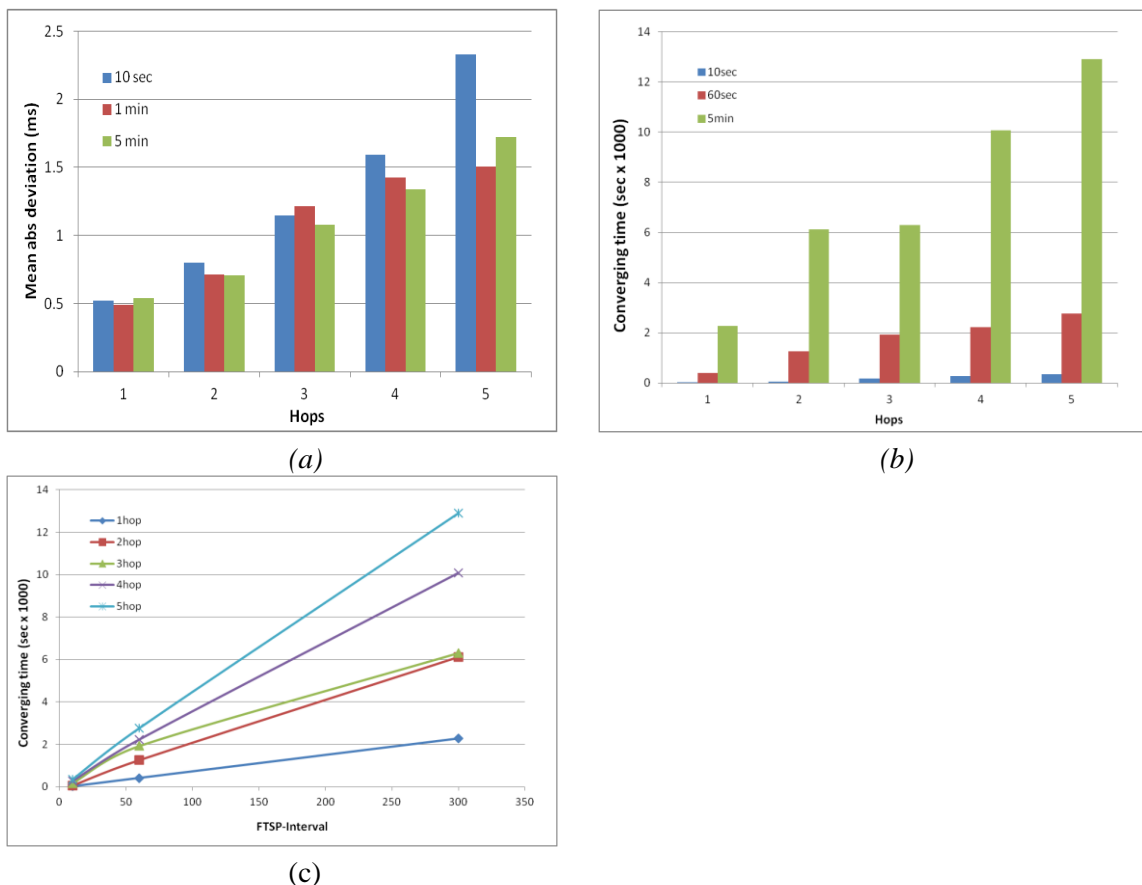


Figur 3.5 Måleresultat der $P=5$ min. (a) Relativt avvik i global tid i den transiente fasen. (b) Relativt avvik i global tid i den persistente fasen. (c) Konvergeringstid som funksjon av avstanden fra masternoden. (d) Drifting på sensorklokken relativt til NTP tid.

Her ser vi at presisjonen på FTSP har et avvik som ligger mellom -5 og 3 ms. Dette er i samme størrelsesorden som for tilfellet der $P = 10$ sek, hvilket indikerer at intervallet på synkroniseringsmeldingen har liten innvirkning på presisjonen. Figur 3.4(c) viser konvergeringstiden som funksjon av hoppavstanden fra masternoden.

3.3.3 Test 3. $P = 5$ minutter

Figur 3.5 viser måleresultater for Test 3, der $P = 5$ min. Samplingsintervallet i dette tilfellet er satt til 1 min. Relativt avvik i global tid i den transiente og persistente fasen er vist i Figur 3.5 (a) og (b). Figur 3.5(c) viser konvergeringstiden, som i dette tilfellet er ca. 215 min for node 6. Node 6 er 5 hopp unna masternoden. Videre viser Figur 3.5(d) driftingen på masternodene i forhold til NTP tiden, hvilket indikerer at sensorklokken går noe raskere enn ”sann tid”. Driften er målt til å være på 0,0239 sek/sek. Dette innebærer at for å få mer presise estimater av for eksempel hastigheten til et objekt som målfølges, må en ta høyde for denne driftingen i utregningen. Merk at siden de samme sensornodene er brukt i alle de tre testene, vil derfor denne driftingen i tid i forhold til ”sann tid”, være den samme for alle de tre tilfellene. Av den grunn er dette måleresultatet kun presentert i Figur 3.5 og ikke i Figur 3.3 og Figur 3.4.



Figur 3.6 Sammenligning av måleresultater. (a) Gjennomsnittlig avvik i global tid. (b) Konvergeringstid som funksjon av avstand R fra masternoden. (c) Konvergeringstid som funksjon av perioden P (FTSP-intervall)

3.4 Diskusjon av måleresultater

Figur 3.6 viser en sammenligning av gjennomsnittlig absolutt avvik i global tid som funksjon av antall hopp for Test 1-3. Resultatet i Figur 3.6 (a) viser igjen at perioden P har ingen innvirkning på presisjonen til tidssynkroniseringen. Derimot viser resultatet at gjennomsnittlig absolutt avvik i global tid øker med økende avstand fra masternoden. I et stort nettverk med mange noder, og der radiusen R er stor, for eksempel $R > 30$, kan presisjonen relativt til masternoden være et problem, spesielt for de nodene som befinner seg i ytterkanten av nettverket. På en annen side, for noder som befinner seg i nærheten av hverandre, vil usikkerheten i presisjonen relativt til hverandre være mindre.

Ett interessant resultat vist i Figur 3.6 (a) er at i Test 1 der $P = 10$ sek., så er gjennomsnittlig absolutt avvik i flere tilfeller noe høyere sammenlignet med resultatet i Test 2 og 3 der perioden P har en høyere verdi. Dette gjelder spesielt for tilfellet med 5 hopp der gjennomsnittlig absolutt avvik er mellom 0,6-0,8 ms høyere i forhold til resultatet i henholdsvis, Test 3 og 2. En mulig forklaring på dette kan være at noder som er lengre unna masternoden er mer sensitiv for pakketap når perioden P blir mindre og at faren for pakkekollisjoner øker ved høyere trafikkpåtrykk.

Videre ser vi at perioden P har en mer markant innvirkning på konvergeringstiden. Dette er vist Figur 3.6(b) som sammenligner konvergeringstiden for de tre testene. Resultatet viser at konvergeringstiden øker med økende verdi P og at relasjonen mellom konvergeringstiden og perioden P er lineær som vist i Figur 3.6(c). Dette stemmer godt overens med likning (2.3). Basert på Test 1-3, kan vi trekke frem følgende resultater:

- FTSP gir en presisjon som i gjennomsnitt er $< 2,5$ ms. Denne presisjonen kan anses å være tilstrekkelig god for anvendelsen i våre scenarioer.
- Presisjonen reduseres med økende avstand fra masternoden.
- Konvergeringstiden er en funksjon av perioden P , der høyere verdi av P gir høyere konvergeringstid.
- I ett WSN nettverk med begrenset batterikapasitet er det ikke akseptabelt at verdien på P er for liten, siden dette vil medføre mye signaleringstrafikk og kortere levetid på nettverket. På en annen side, med for stor verdi på perioden P så vil konvergeringstiden være uakseptabelt høyt.

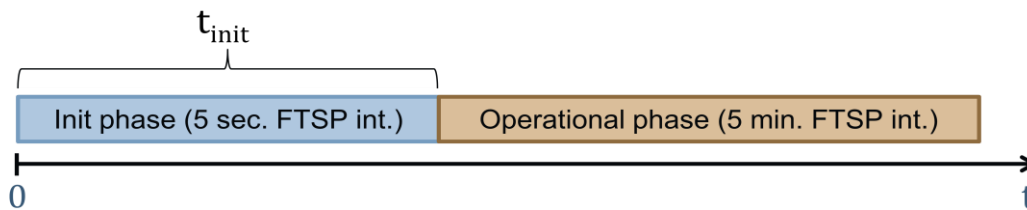
På bakgrunn av disse resultatene ser vi at det er nødvendig å tilpasse eller foreta modifikasjoner av FTSP for å minimere konvergeringstiden uten å forkorte levetiden på nettverket.

4 Modifisering av FTSP

4.1 Beskrivelse av modifisert FTSP

Konklusjoner fra testene som er utført tilsier at det er behov for modifikasjoner av FTSP protokollen for å minimere konvergeringstid uten at dette skal gå ut over bruken av båndbredde og energi. For å oppnå disse målene er FTSP modifisert slik at forløpet til protokollen deles inn i to faser som illustrert i Figur 4.1. Ved oppstart går FTSP inn i en initiell fase med en varighet t_{init} . I løpet av denne fasen er perioden $P=P_1$ satt til en lav verdi slik at synkroniseringsmeldinger utveksles hyppig for å minimere konvergeringstiden. Etter den initielle fasen når alle noder er forventet å være synkroniserte, går FTSP over i en operasjonell fase der $P=P_2$ settes til en høyere verdi for å minimere bruken av båndbredde og energi.

I den initielle fasen er standardverdien for $P_1 = 5$ sek. Videre så er varigheten t_{init} en viktig parameter og må settes til en tilstrekkelig stor verdi slik at alle noder i nettverket rekker å synkronisere seg. Her er den optimale verdien en funksjon av nettverkets radius R med utgangspunkt i masternoden som senternode. Standardverdien for denne parameteren er 360 sekunder, og vil i teorien dekke ett nettverk med radius på 16 hopp, forutsatt at det ikke er pakketap. For å ta høyde for eventuelle pakketap bør denne parameteren settes til en høyere verdi enn den teoretiske. Basert på likning (2.3), kan vi beregne teoretiske og anbefalte verdier for t_{init} , for et utvalg av nettverksradius R som vist i Tabell 4.1. I utregningen av disse verdiene er det tatt utgangspunkt i at FTSP intervallet i den initielle fasen er 5 sekunder og at konstanten $C = 25$ sekunder. For de anbefalte verdiene er det tatt høyde for 3 pakketap ved hvert hopp i utregningen.



Figur 4.1 Initiell og operasjonell fase i FTSP (modifisert)

Nettverksradius R (hopp)	Teoretisk verdi på t_{init} (sek)	Anbefalt verdi på t_{init} (sek)
5	125	200
10	225	375
15	325	550
20	425	725
25	525	900

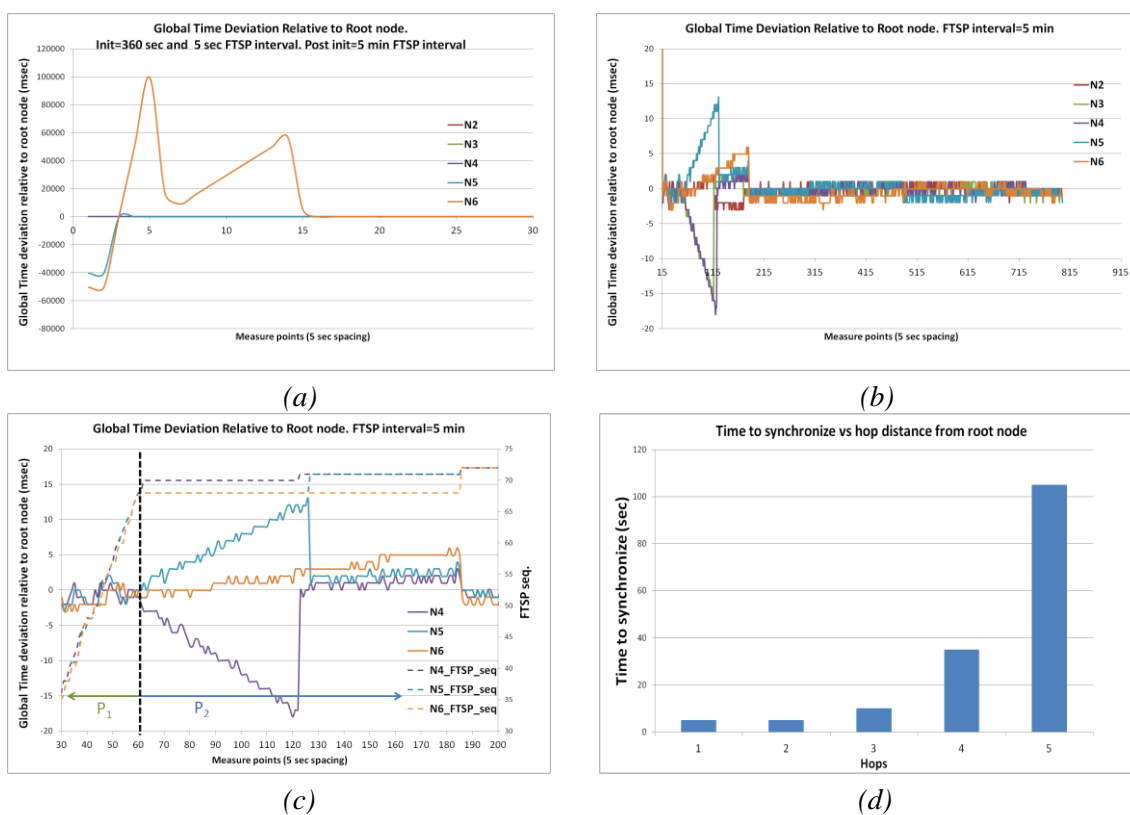
Tabell 4.1 Teoretiske og anbefalte verdier for t_{init} som funksjon av nettverksradius R .

4.2 Evaluering av modifisert FTSP, Test 4

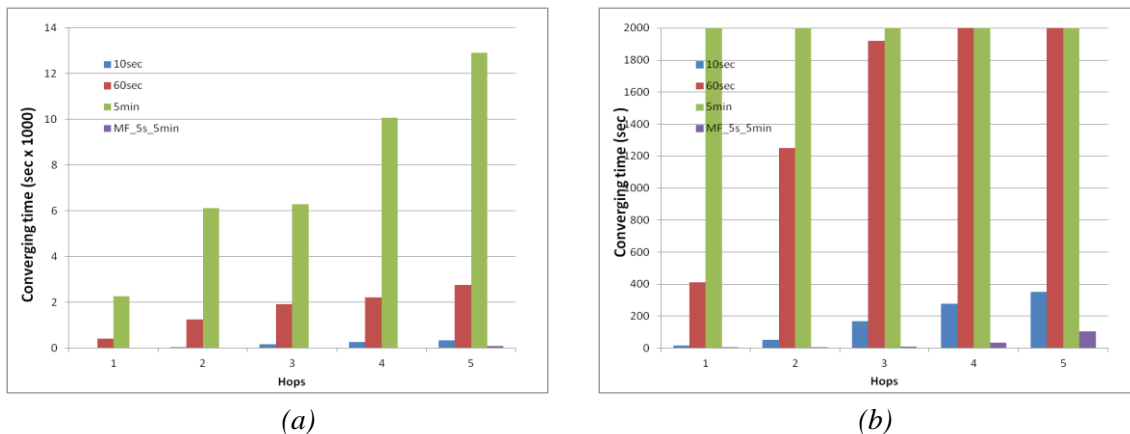
For å verifisere og evaluere den modifiserte versjonen av FTSP er en ny test utført på tilsvarende måte som tidligere tester. Følgende parametre er brukt i Test 4:

- $P_1=5$ sek.
- $P_2=5$ min.
- $t_{init}=360$ sek.

Resultater fra Test 4 er vist i Figur 4.2, der (a) og (b) viser henholdsvis relativt avvik i global tid i den transiente fasen og i den persistente fasen. Av Figur 4.2(a) ser vi at siden FTSP intervallet P_1 i den initielle fasen er satt til en lavere verdi sammenlignet med P verdien i Test 1-3, så er konvergeringstiden derfor også lavere.



Figur 4.2 Måleresultat for Test 4. (a) Relativt avvik i global tid i den transiente fasen. (b) Relativt avvik i global tid i den persistente fasen. (c) Relativt avvik i global tid i overgangen fra initiell til operasjonell fase. (d) Konvergeringstid som funksjon av antall hopp fra masternoden



Figur 4.3 Sammenligning av konvergeringstid for Test 1-4

Ett interessant resultat som følge av den modifiserte oppførselen til FTSP er at ved overgangen fra P_1 til P_2 , så får vi en temporær og liten økning i avviket i global tid, dvs. til mellom -18 til 13 ms. For å forklare dette fenomenet, viser Figur 4.2(c) en forstørret versjon av Figur 4.2(b). Kun resultatet for noen av nodene (dvs. node 4 til 6) er vist i figuren for å begrense mengden med informasjon og for at det skal være mer oversiktlig. I tillegg er FTSP sekvensnummer også inkludert for å vise sammenhengen mellom forløpet til mottatte synkroniseringsmeldinger og hvilke konsekvens dette har for avviket i global tid. Det vi ser er at i området merket P_1 , så har sekvensnummer-kurvene en bratt stigning, hvilket betyr at synkroniseringsmeldinger mottas jevnlig og med kort intervall P_1 . I dette området er avviket i global tid relativt lite og er kun på noen få millisekunder. Når perioden P skifter så til en større verdi P_2 , ser vi at sekvensnummer-kurvene har en flatere stigning, og vi får et lengre opphold mellom hver utveksling av synkroniseringsmeldinger. På grunn av denne overgangen trenger protokollen en viss periode for å tilpasse seg til den nye P verdien. I løpet av denne tilspasningen vil unøyaktigheter i tids-estimeringen akkumuleres. Dette vil igjen resultere i en økning i avviket i global tid. Siden relasjonen mellom lokal tid og global tid, som angitt i likning (2.2), er en lineær funksjon, vil denne akkumuleringen resultere i en lineær økning i avviket. Videre ser vi fra Figur 4.2(c) at det tar ca. 2 synkroniseringsperioder, dvs. $2 \cdot P_2$, for protokollen å stabilisere seg igjen. Av Figur 4.2(b) ser vi at når protokollen først har tilpasset seg til den nye perioden P_2 , så er avviket i global tid redusert til mellom -3 og 1 ms. Dette bekrefter tidligere konklusjon om at perioden P har liten effekt på presisjonen på klokkesynkroniseringen.

Figur 4.2(d) viser konvergeringstiden som funksjon av antall hopp fra masternoden, forutsatt at vi ser bort i fra økningen i avvik i overgangsfasen fra P_1 til P_2 . Denne forutsettelsen er rimelig siden økningen i avviket er minimal sammenlignet med avviket i den transiente fasen.

Figur 4.3 viser en sammenligning av konvergeringstiden for alle de 4 testene. Av figuren ser vi tydelig at den modifiserte FTSP algoritmen (MF_5s_5min) har en betydelig lavere konvergeringstid enn de øvrige. Figur 4.3(b) er en forstørret versjon av Figur 4.3(a).

5 Oppsummering og konklusjon

Denne rapporten dokumenterer vår evaluering av ytelsen til tidssynkroniseringsprotokollen FTSP for bruk i WSN nettverk. Gjennom eksperimenter med utgangspunkt i et enkelt testoppsett bestående av 6 noder, er protokollens ytelse testet ut med tanke på presisjon og konvergeringstid. Den målte presisjonen har et avvik på under 10 ms i forhold til masternoden sin klokke, hvilket også representerer den globale tiden eller referanse tiden. Dette kan anses som en god presisjon for å oppnå korrekt gjengivelse av hendelsesforløpet i de scenarioene som SASS prosjektet har valgt ut: perimetersikring og overvåking av veiakse. Særlig i veiakse-scenarioet er høy presisjon i tidssynkroniseringen viktig, pga. en generell høyere hastighet på objekter eller kjøretøyer som overvåkes og målfølges enn i perimetersikring-scenarioet. En høy presisjon gir oss for eksempel mulighet til å estimere hastigheten på forbi passerende kjøretøy, og varigheten på eventuell opphold, mer eksakt [7;8].

Videre viser resultater fra målinger at konvergeringstiden, dvs. tiden for synkronisering av alle noder i nettverket, er en lineær funksjon av parametre som radiusen R på nettverket og perioden P på synkroniseringsmeldinger. For å oppnå kort konvergeringstid kan en enten minimere R eller P . Ved å minimere R vil en redusere frihetsgraden for deployering av sensornettverket. På en annen side, ved å minimere perioden P vil det resultere i økt signaleringstrafikk, som igjen vil redusere båndbredden for nyttelast.

For å løse dette problemet foreslår vi å dele forløpet til FTSP inn i to faser, en initiell- og en operasjonell-fase. Med denne delingen oppnår vi en vesentlig forbedring i konvergeringstiden, uten at dette vil medføre en stor økning i signaleringstrafikken. I den initielle fasen er perioden $P=P_1$, mens i den operasjonelle fasen er $P=P_2$, der $P_2 \gg P_1$. Ideen er at synkroniseringsmeldinger utveksles hyppig i den initielle fasen slik at noder raskt kan synkroniserer seg. Når synkronisering er oppnådd reduseres hyppigheten av signaleringstrafikk for å frigi verdifull båndbredde til datatrafikk.

Den foreslåtte løsningen er implementert med utgangspunkt i den originale implementasjonen av FTSP. Den modifiserte FTSP er senere testet ut i samme testoppsett som den originale FTSP. Resultater fra våre tester viser at den modifiserte FTSP protokollen gir en rimelig god presisjon og en vesentlig forbedring i konvergeringstid sammenlignet med original versjonen av FTSP.

Referanser

- [1] W.Dargie and C.Poellabauer, *Fundamentals of Wireless Sensor Networks Theory and Practice* Wiley, 2010.
- [2] N.Xu, "A survey of sensor network applications," *IEEE Communications Magazine*, Vol. 40, No. 8, pp. 102-114, 2012.
- [3] S.P.Kumar, "Sensor networks: Evolution, opportunities, and challenges," 2003.
- [4] I.Akyildiz, W.Su, Y.Sankarasubramaniam, and E.Cayirci, "A survey on sensor networks," *Communications Magazine, IEEE*, vol. 40, no. 8, pp. 102-114, 2002.
- [5] K.Khedo, R.Perseedoss, and A.Mungur, "A Wireless Sensor Network Monitoring System," *International Journal of Wireless & Mobile Networks (IJWMN)*, Vol. 2, No. 2, pp. 31-45, 2010.
- [6] M.Zorzi, "Wireless sensor networks," *IEEE Wireless Communications*, Vol. 11, No. 6, pp. 2-2, 2004.
- [7] Joakim Flathagen, Terje M.Mjelde, Reinert Korsnes, Vinh Pham, and Jostein Sander, "Eksperimenter med trådløst sensornettverk for perimetersikring og akseovervåkning," FFI-rapport 2012/02009, 2012.
- [8] Vinh Pham, Erlend Larsen, Joakim Flathagen, Terje M.Mjelde, Reinert Korsnes, and Jostein Sander, "Sluttrappport for prosjekt 1141 Situational Awareness Sensor Systems (SASS)," FFI-rapport 2012/02490, 2012.
- [9] M.Maroti, B.Kusy, G.Simon, and A.Ledeczi, "The flooding time synchronization protocol," in *Proceedings of the 2nd international conference on Embedded networked sensor systems. ACM New York, NY, USA, 2004*, pp. 39-49., 2004.
- [10] TinyOS, <http://www.tinyos.net/>
- [11] J.Elson, L.Girod, and Estrin, "Fine-Grained Network Time Synchronization using Reference Broadcasts," in *The Fifth Symposium on Operating Systems Design and Implementation (OSDI)*, p.147-163 2002.
- [12] S.Ganeriwala, .Kumar, and .Srivastava, "Timing-Sync Protocol for Sensor Networks," *The First ACM Conference on Embedded Networked Sensor Systems (SenSys)*, p. 138-149, 2003.

Akronymer

FTSP	Flooding Time Synchronization Protocol
MAC	Medium Access Control
NTP	Network Time Protocol
RBS	Reference Broadcast Synchronization
SASS	Situational Awareness Sensor Systems
TPSN	Timing-Sync Protocol for Sensor Networks
UTC	Coordinated Universal Time
WSN	Wireless Sensor Network