Sluttrapport for prosjekt Situational Awareness Sensor Systems (SASS)

Vinh Pham, Erlend Larsen, Joakim Flathagen,
Terje Mjelde, Reinert Korsnes, Jostein Sander og Per Dalsjø

Forsvarets forskningsinstitutt (FFI)

8. januar 2013

FFI-rapport 2012/02490

1141

P: ISBN 978-82-464-2221-3
E: ISBN 978-82-464-2222-0

Emneord

Trådløs sensornettverk

Sensorer

Perimetersikring

Styrkebekyttelse

Detektorer

Godkjent av

Torunn Øvreås	Prosjektleder
Anders Eggen	Avdelingssjef

Sammendrag

FFI-prosjektet SASS har undersøkt potensialet for bruk av lavkost *ad hoc* trådløse sensor-nettverk innen det norske forsvaret. Intensjonen har vært å bidra til bedre trussel-oppmerksomhet innen både konsentrerte militære operasjoner og innen Heimevernet. Dialog med aktuelle brukere er da viktig. En har utredet eksisterende teknologier og utført praktiske eksperimenter for å få erfaring med anvendelse av teknologien. Denne rapporten sammenfatter dette arbeidet. Vi har sett på energihøsting, skaffet aktuell hardware for eksperimenter samt prøvd programvare for kommunikasjon og datainnsamling fra sensorer. Det er utført praktiske øvelser (CD&E) ved Rena leir der vi demonstrerte deteksjon av fiendtlig inntregning mot en vaktstyrke. Ved FFI ble det demonstrert overvåking av veistrekning for å spore biltrafikk og detektere bilstopp langs veien som kunne indikere en trussel. En hovedkonklusjon fra arbeidet er at teknologien trådløse sensornettverk er moden nok for flere militære anvendelser som oppklaring, arealovervåkning og objektsikring. Det foregår mye forskning innen fagfeltet og de teknologiske utfordringer som er identifisert i prosjektet slik som fysisk størrelse, systemlevetid, sikkerhet og radiorekkevidde, antas å kunne løses innen nær fremtid.

English summary

The FFI project SASS has explored potential applications of low cost *ad hoc* wireless sensor networks within the Norwegian Armed Forces. The intention has been to improve the capacity of threat detection within concentrated military operations as well as within the Home Forces. Interaction with actual end users of such sensor network technology within the Armed Forces has been mandatory. We have examined existing technology and performed practical experiments in order to obtain experience. We have looked into concepts for energy harvesting, developed actual hardware for experiments and tested software for communication and data retrieval from sensors. We have performed practical exercises (CD&E) at the Rena Military Camp demonstrating detection of hostile penetration towards a guard force. Another experiment demonstrated tracking of cars along a road segment at FFI with the objective to detect car stops. This may represent a threat in terms of deploying IED. A main conclusion from the present work is that the technology of wireless sensor networks is mature enough for a range of military applications within for example surveillance and object protection. The technology is an active topic for research and the technical challenges, which are identified within the project, such as physical size of hardware, system lifetime expectancy, security and radio range, is supposed to be solved within near future.

Innhold

1 Innledning

1.1 Bakgrunn

Elektroniske sensorer kan effektivt komplettere direkte menneskelig observasjon og overvåking for styrkebeskyttelse. Slike hjelpemidler vil kunne bidra til tidligere deteksjon av inntrengere, bedre reaksjonsevne og utvidet sikring. Det vil også kunne redusere personellbruk og belastningen som et tradisjonelt vakthold medfører. Denne rapporten beskriver utvikling og uttesting av et trådløst sensornettverk for styrkebeskyttelse.

Bruk av trådløse sensorer for taktiske formål er en over femti år gammel ide, som først ble realisert av US Army på slutten av sekstitallet. Siden den tid har en rekke militære trådløse sensorsystemer blitt konstruert og uttestet av mange nasjoner. Utviklingen innen mikroelektronikk og trådløs teknologi, har muliggjort stadig rimeligere og mer energieffektive sensorsystemer. Det er et markant skille mellom de første militære sensorsystemene og dagens systemer. De tidlige systemene bestod av enkle sensornoder uten særlig prosesseringskapasitet. De var gjerne store og opererte uavhengig av hverandre og sendte ufiltrerte sensormålinger direkte inn til en basestasjon. Nye systemer er derimot basert på multihopp nettverksteknologi inspirert av pakkerutingen i Internett. Dette muliggjør stor redundans og automatisk feilkorrigering. Videre vil sensornodene kunne samarbeide for å få bedre deteksjon samt å forhindre eller begrense falske alarmer. Minimal størrelse på hver enkelt node og lave kostnader muliggjør systemer med et stort antall noder, men som likevel er enkle å deployere.

1.2 Motivasjon

Det finnes en rekke sensortyper tilgjengelig for militær overvåkning. Ulike sensorsystemer gir forskjellig grad av nøyaktighet i varsling og klassifikasjon. Valg av sensortyper vil typisk avhenge av en rekke forhold som for eksempel størrelsen på området som skal overvåkes, varigheten på leiren som skal beskyttes, forventet trussel, geografi og værforhold. For eksempel er mastmonterte sensorer mest egnet til observasjon av store områder utenfor en leir eller posisjon. Gjerdemonterte perimetersensorer benyttes gjerne til overvåkning av et gjerde eller en barriere. Både mast- og gjerdemonterte sensorer er tilpasset langvarige leiroperasjoner, og er lite hensiktsmessige i små hurtigforflyttende operasjoner. Til dette trengs et mer fleksibelt og lettere system. Formålet med prosjekt SASS har vært å utvikle og utteste et trådløst sensornettverk nettopp for slikt bruk. Prosjektet har valgt å utvikle egne prototyper. Dette er gjort av to årsaker. For det første er det få sensorsystemer innen denne kategorien som er kommersielt tilgjengelige i dag. Den andre begrunnelsen var av kompetanseoppbyggende art. Vi ønsket å studere en rekke teknologier innenfor trådløse sensornettverk slik som pakkeruting, lokalisering, energiforbruk osv. for å kunne danne et bilde av fremtidige muligheter og skissere realistiske anvendelsesområder for teknologien. En slik detaljert teknologistudie er vanskelig å gjennomføre uten å bygge opp den nødvendige dybdeforståelse gjennom egen utvikling. Kommersielle sensorsystemer, i den grad de i det hele tatt er tilgjengelige, er heller ikke åpne nok til å tillate en slik studie.

1.3 Rapportens oppbygning

Resten av denne rapporten er organisert på følgende måte: Kapittel 2 gir en kort innføring i trådløse sensornettverk (WSN) teknologien, samt aktuelle scenarioer for anvendelse i forsvaret. En gjennomgang av de forskjellige teknologiområdene som FFI har jobbet med og som er relevant i konstruksjonen av et WSN system er beskrevet i Kapittel 3. Kapittel 4 omhandler den praktiske øvelsen som er gjort i prosjektet, nemlig realisering av et WSN system for uttesting av teknologien. I dette arbeidet inngår det blant annet implementering av hardware og software, testing, samt gjennomføring av forsøk med det utviklede systemet i utvalgte scenarioer. Resultatet fra forsøkene er presentert i samme kapittel. Rapporten konkluderes i Kapittel 0.

2 Trådløse sensornettverk

2.1 Grunnleggende prinsipper

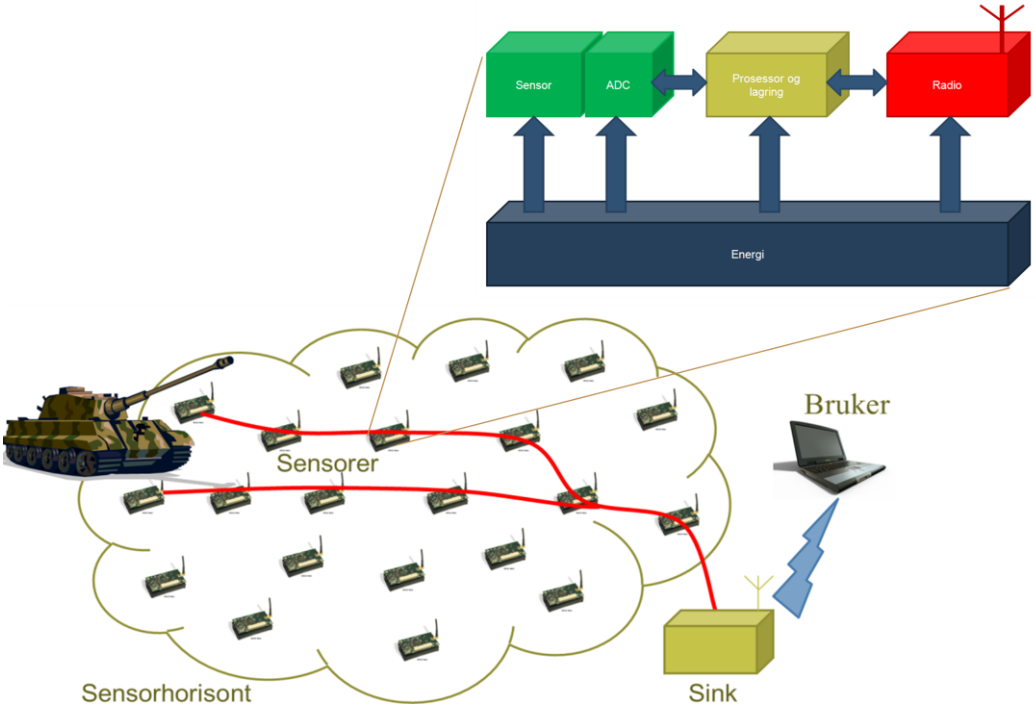
Utviklingen av WSN teknologien ble til å begynne med startet av det amerikanske forsvaret, og motivasjonen var å anvende teknologien for overvåking av slagmark, og deteksjon av angrep med masseødeleggelsesvåpen (kjemisk, biologisk, og kjernefysisk). Etter hvert som fordelene til WSN teknologien over tradisjonelle trådbaserte nettverk er blitt mer tydelig, har denne teknologien blitt mer utbredt både innen industrielle og sivile anvendelser.

Et trådløst sensornettverk (WSN) består av et sett med trådløse sensornoder, fra noen titalls til flere tusen noder. Disse nodene samarbeider med hverandre i å overvåke fysiske parametre knyttet til omgivelsene som for eksempel temperatur, trykk, lyd eller ulike former for forurensninger.

Et eksempel på et WSN er vist i Figur 2.1. Ved bruk av radiogrensesnittet kan nodene danne et trådløst nettverk som muliggjør kommunikasjon og samhandling nodene imellom. Dette nettverket er forutsetningen for at nodene kan samarbeide med hverandre i å overvåke området. Når en inntrenger beveger seg inn det overvåkede feltet, vil endringer i omgivelsesparametre resultere i trigging av alarmer som sendes gjennom nettverket, enten over et eller flere hopp, frem til *sinknoden*. Dette er en sentral node eller et datainnsamlingspunkt som har ansvar for å videresende dataene til en sluttbruker eller et BMS-system, enten via en direkte tilkoblet forbindelse, en radiolinje, eller via en satellittlink.

En sensornode er en liten elektronisk enhet som i prinsippet består av fire hovedkomponenter:

- Sensorer og AD konverter
- Prosessor og lagringskapasitet
- Radiogrensesnitt
- Energikilde (batteri)



Figur 2.1 Eksempel på trådløs sensornettverk (WSN)

Nodene kan realiseres på forskjellig måter med tanke på pris, kompleksitet og størrelse, avhengig av anvendelsesområde og kravspesifikasjoner. I sin enkleste form vil det være mulig å produsere nodene i en størrelse som er vesentlig mindre enn de eksperimentelle modellene vi har brukt i våre forsøk og til en lavere stykkpris. I anvendelser av temporær karakter, som for eksempel oppklaring eller sikring av en tidsbegrenset leir, er det fordelaktig med masseproduserte noder som er små, lette og billige. Disse er da ment å brukes for en begrenset periode og kan etterlates etter bruk. I permanente installasjoner derimot, kan det være mer hensiktsmessig å bruke større og mer sofistikerte noder med bedre sensorer og lengre levetid.

2.2 Anvendelser i Forsvaret

WSN-teknologien har utvilsomt mange potensielle nyttige anvendelser. Allerede i dag finnes det en rekke produkter eller systemer som tar i bruk denne teknologien, både innen militær og sivil sektor. Eksempler på disse er Exensors Flexnet sensornettverk , og Advanticsys overvåkningssystemer for industrien.

En viktig del av SASS prosjektet er å identifisere potensielle gode anvendelser av denne teknologien i Forsvaret. Fremgangsmåten er tett dialog og samarbeid med aktuelle brukere for å kartlegge de mest umiddelbare behov. I dette arbeidet er det viktig å forstå prosedyrer og arbeidsmåte hos bruker slik at man kan avdekke eventuelle nye anvendelser som ikke er åpenbare.

Dette arbeidet har resultert i et notat [1] som beskriver fire forskjellige scenarioer hvor WSN-teknologien kan være aktuelle i Forsvaret:

- Nærforsvar / perimetersikring
- Overvåkning av punkt, akse eller område - vakthold
- Akse, punkt eller område – «information on demand»
- Stort nettverk

2.2.1 Scenario A – nærforsvar/perimetersikring

Dette scenarioet er ment å dekke behovet for sikring av enkelte sektorer av nærområdet når et lag eller liten gruppe soldater har slått leir for natten eller har etablert en observasjonspost. De vil ofte velge en posisjon hvor de har visuell kontroll over omgivelsene. For en observasjonspost er dette innlysende. Men selv i dette tilfellet vil det være sektorer som innebærer risiko og som ikke er under observasjon av den som er på post.

Området som skal overvåkes kan antas å ha en utstrekning i hver retning på 50-100 m. I utkanten av dette området legges ut et sensorsystem som primært skal detektere fiendtlig personell som nærmer seg området. Eventuelt legges nodene i de deler av området som ikke er kontrollert eller overvåket på annen måte. Det kan også være aktuelt å legge noder i flere «lag» utover fra leiren for å få tidlig varsling eller for å spore (tracke) en eventuell inntrenger eller en hendelse. Her antar man at det er en eller flere sink-noder i leiren som abonnerer på varsling fra systemet.

2.2.2 Scenario B - overvåkning av punkt, veiakse eller område i sanntid

Dette er det mest klassiske scenarioet. Man ønsker alarm i sanntid dersom det er trafikk i et område av interesse – et veikryss, et høydedrag eller et avgrenset areal. Dette området vil typisk være i tilknytning til en base, men i noe avstand. Det at man ønsker beskjed i sanntid må ses i sammenheng med en tradisjonell vakt- og sikringsprosedyre; alarmen skal komme frem tidsnok til at skadebegrensende tiltak kan iverksettes. For eksempel kan vaktstyrken rykke ut, eller basen kan gå i alarmberedskap. Posisjoner for fiendtlige snikskyttere eller mobile bombekastere kan være av interesse, med avstand 1-3 kilometer fra «basestasjonen». Vi forutsetter i denne sammenhengen at det ikke er aktuelt med sensorer som kan ta bilder (kamera), men snarere gir en «binær melding» om at sensoren er aktivert. Ytterligere informasjon må trekkes ut av en fortolkning av opplysninger fra flere sensorer.

2.2.3 Scenario C - akse, punkt eller område - «informasjon på forespørsel»

Dette scenarioet tar utgangspunkt i scenario B, men skiller seg på et viktig punkt. Det er ingen rapportering til base i sanntid. I dette scenarioet vil sensornettverket settes opp og deretter overlates til seg selv. Det vil samle inn data kontinuerlig (eller ved en form for hendelsesstyrt oppvåkning), eventuelt analysere disse og klassifisere hendelser. Sink-noden kan nå i prinsippet være plassert i et kjøretøy. Når kjøretøyet nærmer seg eller stanser ved «enden av» nettverket, vil registrerte hendelser rapporteres. Typisk anvendelse vil være å overvåke deler av en veistrekning i forbindelse med veibombe-aktivitet, gjerne fjernt fra hovedkvarteret. Vi antar at veistrekningen

er klarert for veibomber, og at man ønsker å vite om det har vært trafikk der innen neste gang man skal passere. Hele strekningen kan være svært lang, men det vil være deler av strekningen som er mer interessant enn andre. Slike interessante områder kan være kulverter, kjente lokasjoner for tidligere veibomber samt kurver eller andre forhold som gjør at hastigheten må settes ned.

2.2.4 Scenario D – Stort nettverk

Scenario D dreier seg om overvåking av et større område hvor det er usikkert om det er forsvarlig å sende inn personell. Nettverket inkluderer et meget stort antall noder, for eksempel mer enn 1.000 noder. Hvis det er et område hvor det er mistanke om gass eller radioaktivitet, kan ikke nettverket deployeres av personell. Enten må nodene slippes fra lufta med for eksempel en UAV, eller de må skytes inn i området. Teknologisk kan dette scenarioet sees på som en utvidelse av scenario B. Det store antallet noder og den tilfeldige nodeplasseringen stiller imidlertid andre og strengere krav til både nettverksprotokollene og lokaliseringsmekanismen. En ytterligere utfordring blir overføring av informasjon fra sensornettverket og inn til hovedkvarteret. Sensornettverket må derfor ha en, eller helst flere, langtrekkende noder (for eksempel satellittlink). Dette scenarioet har mindre betydning i dag, men vil kunne bli aktualisert med fremtidens bruk-og-kast scenarioer.

3 Teknologi

Et WSN system består gjerne av mange komponenter og teknologiske løsninger, både fysiske (maskinvare) og logiske (programvare). Disse komponentene utgjør systemets byggeklosser, og er forutsetningen for at systemet skal kunne utføre de oppgavene som det er tiltenkt. I dette kapitlet gis det en oversikt over og kortfattet beskrivelse av de teknologiene som prosjektet har jobbet med. En mer detaljert beskrivelse av disse teknologiene, vil vi underveis referere til i egne rapporter/notater eller vitenskapelige artikler.

3.1 Radio – transmisjon

Et SASS-system setter spesielle krav til radioløsningen. Rekkevidde, følsomhet for flerbaner og plassering i forhold til terreng er viktige parametre for valg av radioløsning. I de statiske WSN-løsningene som foreløpig er blitt tilgjengelige, er det brukt radioer utviklet for åpne frekvensbånd (ISM-bånd). Det må undersøkes om disse radioene er egnet for militær bruk. Det må også kartlegges hvilke krav SASS setter til radio og transmisjon, og disse kravene må veies opp mot de radioene som allerede eksisterer.

Transmisjonsproblematikk er en spesiell utfordring i SASS da en i utgangspunktet har liten innflytelse over plassering av nodene. Dette er forskjellig fra typiske radionettverk hvor en ved oppsett av systemet ofte kan plassere nodene optimalt for god transmisjon. I SASS vil nodene kunne bli plassert ganske tilfeldig i forhold til hverandre. Hvordan transmisjon over kort hold vil påvirkes av terreng og bygninger må undersøkes for å kunne lage et grunnlag eller regelsett for hvordan utplassering/dropping skal utføres.

Det er gjort omfattende forsøk med små sensornoder med integrert radioenhet, for å måle hvor lang transmisjonsrekkevidde som kan oppnås når nodene ligger på eller nært bakken. Målingene er sammenlignet med en teoretisk modell hvor man antar at det mottas en direkte og en reflektert stråle i mottakeren, for å se om man kan beregne rekkevidden for radioen med rimelig nøyaktighet. Det viser seg å være stor variasjon i mottatt signalstyrke for forskjellige målinger hvor avstand og høydeforskjell mellom nodene er den samme. Det er derfor vanskelig å nøyaktig beregne og forutsi rekkevidden på bakgrunn av de teoretiske betraktningene lagt til grunn, men de vil kunne gi en pekepinn. Spesielt ved korte avstander er variasjonene store på mottatt signalnivå. Resultatene er beskrevet i rapport [2].

3.2 Nettverk, ruting, mac

3.2.1 Rutingprotokoller

WSN er en teknologi under voldsom utvikling. Det foregår mye forskning spesielt på lag 2 og 3, altså datalink-laget (MAC) og nettverkslaget. En viktig oppgave i prosjektet er å ta utgangspunkt i det arbeidet som er gjort innen dette feltet, for deretter å bestemme de protokollene som er egnede/optimale i en SASS-anvendelse, der sensordata primært sendes til en sentral sinknode som har oppgaven å bygge et samlet bilde av situasjonen for videre prosessering. I dette arbeidet inngår litteraturstudie, simuleringer og uttesting i fysiske testoppsett.

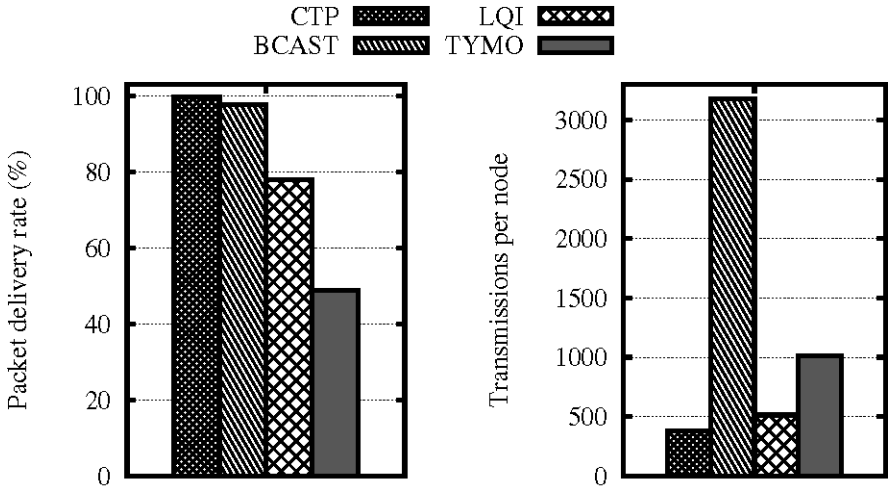
Det er gjort en studie på Directed Diffusion [3], som er en rutingprotokoll spesielt utviklet for WSN. Protokollen skiller seg ut fra tradisjonelle rutingprotokoller ved at den er datasentrisk fremfor adressesentrisk. Dette muliggjør at en sinknode kan søke etter informasjon eller dataobjekter av interesse, som befinner seg i ute i nettverket uten behov for kunnskap om topologien. Noder som er i besittelse av den forespurte informasjonen vil periodisk sende meldinger til sinknoden. Videre er protokollen spesielt designet med tanke på mulighet for datafusjon i nettverket, for å oppnå bedre utnyttelse av ressurser. Ulempen med Directed Diffusion er at andelen kontrolltrafikk kan være stor, spesielt ved bruk av TwoPhasePull modus (dvs. toveis signalering). En måte å redusere dette på er å bruke OnePhasePull modus (dvs. signalering i retning fra sinknoden til sensornoder), men med den konsekvens at den ruten som velges ikke nødvendigvis er optimal.

To artikler som beskriver en tilpasning av OLSR til WSN ved hjelp av dynamisk økende intervaller, er publisert i [4] og [5]. Proaktive linktilstand rutingprotokoller, som er populære å bruke i mobile ad-hoc nettverk, har ikke hatt så stor suksess i trådløse sensornettverk. Dette er hovedsakelig på grunn av den store energibruken som behøves for kontrolltrafikktransmisjoner og lagring av tilstandsinformasjon. Imidlertid kan denne typen rutingprotokoller i mange situasjoner være bedre å bruke enn andre rutingtyper. Fordelene er topologioversikt og tilgangen på et optimalisert spredningstre for informasjonsdistribusjon. Ved å utnytte den statiske egenskapen (dvs. noder forflyttes ikke etter deployering) til WSN kan en redusere på signaleringskostnaden betydelig.

Man kan tilpasse OLSR protokollen slik at den fungerer bedre i et WSN-miljø ved å sende kontrollmeldinger med en lav frekvens når nettverket er stabilt, og oftere hvis topologiforandringer, som for eksempel pga. linkbrudd inntreffer. Den foreslåtte løsningen blir undersøkt ved bruk av simuleringer i både tapsfritt og tapsfylt propagasjonsmiljø, og resultatene er lovende.

Videre er det satt opp en testbed bestående av 20 sensornoder, montert på en vegg i en lab, for å teste ut ulike rutingprotokoller. Ved å minimere uteffekten på radiotransmisjonen, fikk vi et realistisk multihop-nettverk på et relativt lite område. Sensornodene ble koblet sammen ved hjelp av USB. På denne måten kunne vi gjøre reprogrammering av sensornodene enklere, samt overvåke systemstatus og kjøre repeterte eksperimenter under de samme testforhold. Resultatet i Figur 3.1 viser uttesting med fire protokoller: CTP (trebasert), BCAST (en veldig enkel broadcast-protokoll), MultihopLQI (LQI, en trebasert protokoll), og TYMO, som er basert på DYMO og er en ende-til-ende protokoll.

Som figuren viser ga CTP de beste resultatene både med tanke på pakkeleveringsraten, som var nær 100 % og overhead, som var lavere enn for de tre andre protokollene. Disse resultatene er også i tråd med resultater fra tidligere forskning i litteraturen. Vi valgte dermed å benytte CTP som basis for vår testplattform. Videre har vi gjort tester av CTP i flere scenarioer. Figur 3.2 viser ytelsen til CTP i seks ulike miljøer. Første test i figuren er resultatet fra test-bed som nevnt over. Siden en slik innendørs testbed ofte underestimerer dynamikken som kan oppstå i et deployert nettverk, utsatte vi nettverket for en jamme-kilde for å skape mer dynamikk. De neste forsøkene var kontorlokale (50 noder), perimetersikring (50 noder) og veiakse. For det siste scenarioet ble det gjort forsøk med sinkplassering både på enden av nettverket og midt i nettverket. For alle forsøkene ser vi at CTP gir god leveringsrate. Leveringsraten er lavest når dynamikken er høy (interferens) og i nettverk med stor utstrekning (veiakse). At dynamikken i disse tilfellene er høy, kan ses av at ruteendringer gjøres hyppig. Ut fra veiakseeksperimentene ser vi at sinkplasseringen har stor betydning for ytelsen til nettverket. Videre ser vi ut fra testbedeksperimentene at interferens er en annen faktor som påvirker leveringsraten. Av denne grunn har vi gjort arbeider for å lage algoritmer som finner gode sinkplasseringer gitt en nettverkslayout. Vi har også forsøkt å forbedre rutingprotokollens ytelse når nettverket blir utsatt for interferens eller jamming. Disse to arbeidene blir beskrevet i de følgende underkapitler.



Figur 3.1 *Figuren til venstre viser pakkeleveringsraten (i prosent) for fire rutingprotokoller; CTP, Broadcast (BCAST), MultihopLQI (LQI) og TYMO, uttestet i en testbed med 20 sensornoder. Til høyre vises summen av antall transmisjoner i løpet av en times kjøring med 20s intervall mellom hver datapakke.*

Nettverk	Areal	Noder	Hopp		Churn	PDR
	m		Avg	Max		
Testbed	2.5x2.5	20	2.0	4.1	0.46	0.996
Testbed (m. interf.)	2.5x2.5	20	6.9	180	17.7	0.804
Kontorlokale	40x70x6	50	3.0	6.0	0.72	0.994
Perimeter	80x120	50	3.0	5.1	0.96	0.986
Veiakse (endepl. Sink)	190x4	40	6.5	11.2	3.64	0.784
Veiakse (midpl. Sink)	190x4	40	3.7	9.0	1.38	0.968

Figur 3.2 *Resultater fra uttesting av CTP i seks ulike miljøer: Testbed med og uten interferens (jamming) et kontorlokale over tre etasjer, et perimetersikringsscenarie og en veiakse med to ulike sinkplasseringer (endeplassert og midtplassert). Tabellen viser areal, antall noder, gjennomsnittelig og maksimalt antall hopp, antall ruteendringer per node per time (Churn) og pakkeleveringsraten (PDR).*

3.2.2 Sinkplassering

Ruting i trådløse sensornettverk utføres gjerne av en innsamlingsprotokoll (collection protocol) som bygger et rutingtre med rot i én sink. I store nettverk kan både nettverkets levetid og skalerbarheten økes ved å utplassere flere sinker. I tillegg til å redusere gjennomsnittlig antall hopp mellom en node og en sink, forbedres også fordelingen av energiforbruket i nettverket. Bruk av multiple sinker gir også redundans dersom en av sinkene feiler, blir vandalisert eller stjålet, eller går tom for energi. Hvilken plassering man velger for sinkene vil være avgjørende for ytelsen til rutingen. I vårt arbeid har vi fokusert på løsninger for å finne optimale plasseringer for et gitt antall sinker.

For effektivt å kunne bestemme optimal plassering for en eller flere sinker, må man enten samle inn eller estimere nettverksinformasjon. De ulike metodene for sinkposisjonering kan

kategoriseres basert på om de behøver informasjon om den geografiske posisjonen til alle noder, eller kunnskap om nettverkstopologien. Vi har undersøkt fire ulike sinkposisjoneringsalgoritmer, to i hver av disse kategoriene. Resultatene våre, som er gitt i [6], viser at den beste ytelsen oppnås med en sinkplasseringsstrategi som tar hensyn til den faktiske nettverkstopologien. Dersom man kun tar hensyn til den geografiske posisjonen til nodene, vil algoritmen kunne foreslå sinkplasseringer som ikke er brukbare i en reell deployering på grunn av hindringer eller områder med dårlig nettverksdekning. En strategi som benytter faktisk linkinformasjon og linkkvaliteter vil implisitt ta hensyn til slike hindringer. Våre resultater viser at pakkeleveringsraten og nettverkets levetid kan økes betraktelig ved å øke prosentandelen sinker i nettverket. Dersom sinkene plasseres basert på en gjennomtenkt strategi, vil man få en ytterligere kapasitetsøkning.

3.2.3 Hvilken innvirkning har jamming og interferens på ruting?

Både radioforstyrrelser fra andre radiosystemer, miljøpåvirkninger eller fiendtlige jamme-angrep kan føre til svært uforutsigbar kommunikasjon i trådløse senornettverk. Resultater fra deployerte sensornettverk viser at typisk pakkeleveringsrate er mellom 70 og 99%, men at den kan gå så lavt som 20-40%. En del av disse dårlige resultatene skyldes implementeringsfeil, dårlig planlegging eller «uflaks». En stor del vil også skyldes miljøpåvirkninger (vind, personell i bevegelse) eller interferens. Mye gjøres for å minske systemenes følsomhet for slike påvirkninger, for eksempel å forbedre radiosystemet eller å benytte en mer robust MAC-protokoll. Dette er imidlertid endringer som er vanskelige å implementere etter at sensorsystemet er konstruert, og det kan være formålstjenlig å forbedre protokoller lengre opp i protokollhierarkiet. Vi har derfor studert hvordan ulike rutingprotokoller oppfører seg når sensornettverket er påvirket av forstyrrelser og vi foreslår mekanismer for å maksimere leveringsraten i slike tilfeller.

Vi har konstruert og implementert en ny hybrid opportunistisk protokoll (O-CTP), som bruker tradisjonell ruting basert på CTP [7] når nettverket er stabilt og har lavt pakketap. Dersom nettverket blir utsatt for forstyrrelser eller jamming, bytter protokollen til opportunistisk videresending. Den opportunistiske delen av protokollen bygger på en forenklet versjon av ExOR [8]. Et sett triggere gjør ulike vurderinger av nettverkskvaliteten og lar protokollen bytte til opportunistisk ruting dersom det antas at dette gir høyere pakkeleveringsrate. Vi sammenliknet O-CTP med fem andre rutingprotokoller: CTP, Broadcast (BCAST), MultihopLQI, TYMO og en ren opportunistisk rutingprotokoll (GEOPP). Uttestingen av O-CTP ble foretatt ved å benytte samme testbed som beskrevet tidligere. En interferenskilde kontrollert fra programvare gjorde det mulig å kontrollere interferensen i fire ulike mønstre. Resultatene viste at O-CTP gir den beste balanse mellom pakkeleveringsrate og overhead av de seks protokollene. O-CTP ga for alle interferensnivåene mellom 0-15% høyere pakkeleveringsrate og hadde bare en tyvendedel av overheaden. Det henvises til [9] for utfyllende beskrivelser av protokollen og detaljerte resultater.

O-CTP er bygd for å være uavhengig av de underliggende radiokretsene. Vi ønsket ikke å bryte denne uavhengigheten og de triggerne som bytter til opportunistisk ruting er derfor basert på lag-3 funksjoner. Vi tror imidlertid at protokollen kan prestere bedre dersom den kunne utnytte målinger direkte fra det fysiske laget. Et annet minus med O-CTP er at den i realiteten er bygd på to protokoller. Dette øker både kompleksiteten og kodestørrelsen. Lagringsminnet på de TelosB-

enhetene som ble benyttet var kun 48KB, og optimalisering av kodestørrelsen er derfor overordnet viktig for alle programvarekomponenter som skal benyttes på plattformen.

3.3 Lokalisering, posisjonering

For å kunne lage et bilde av målesituasjonen, må en vite nøyaktig posisjon på den enkelte node. Dette er en meget viktig og samtidig vanskelig problemstilling. Selv om GPS er blitt veldig billig, er det ikke i utgangspunktet naturlig å utstyre hver enkelt node med en GPS mottaker. Årsaken til dette er pris, volum og effektforbruk på GPS-mottakeren, men vel så viktig er at nøyaktigheten som kreves kan være bedre enn det GPS kan gi. Et annet viktig aspekt ved nøyaktig lokalisering er konsekvensene det har for utplassering eller deployering. Med nøyaktig lokalisering stilles det mindre krav til deployeringen, og fra et brukersynspunkt vil det være en fordel om nodene kan kastes ut eller slippes fra lufta.

Ulike eksisterende posisjoneringsmetoder er undersøkt. Det er i simulator implementert en posisjoneringsmetodikk som gjør avstandsmålinger simultant med ruteopprettelse. Denne bruker evolusjonær algoritme for å finne frem til sannsynlige lokasjoner for nodene i nettverket, og metoden er beskrevet i publisert artikkel [10]. Det er i arbeidet benyttet signal/støyforhold som måledata, men andre måledata kan også anvendes på metoden. I tillegg er det også utviklet en metode for å overføre GPS posisjoneringsdata fra enhet med kjent GPS posisjon til sensornoder som selv ikke har GPS mottaker. Dette er beskrevet nærmere i avsnitt 0.

3.4 Datafusjon

Nedenfor oppsummeres erfaringer og innsikter som er vunnet innen SASS prosjektet angående datafusjon. Arbeidet med datafusjon er gjort via både teoretiske og praktiske øvelser der en bruker innsamlede data som vist i Figur 3.3. Visualisering av sanntids alarmdata fra sensornoder er også utført ved fysiske eksperimenter (CD & E) utendørs og ved innendørs testing ved FFI. Slike data kan også avspilles i ettertid.

Rapporten i [11] omhandler metoder for datafusjon for å målfølge kjøretøyer på en vei ved hjelp av et sensornettverk langs veien. Datafusjon i slike sammenhenger inkluderer gjerne komprimering av data fra målingene, slik at mengden av lagring og transport av data varierer i samsvar med informasjonen som hentes inn. Når mengden av innsamlet informasjon reduseres er det rimelig at mengden av transport og lagring i nettverket også reduseres. Å måtte samle inn alle sensordata for sentral prosessering ville i motsatt fall bety en stor begrensning i mulig design og bruk av sensornettverk.

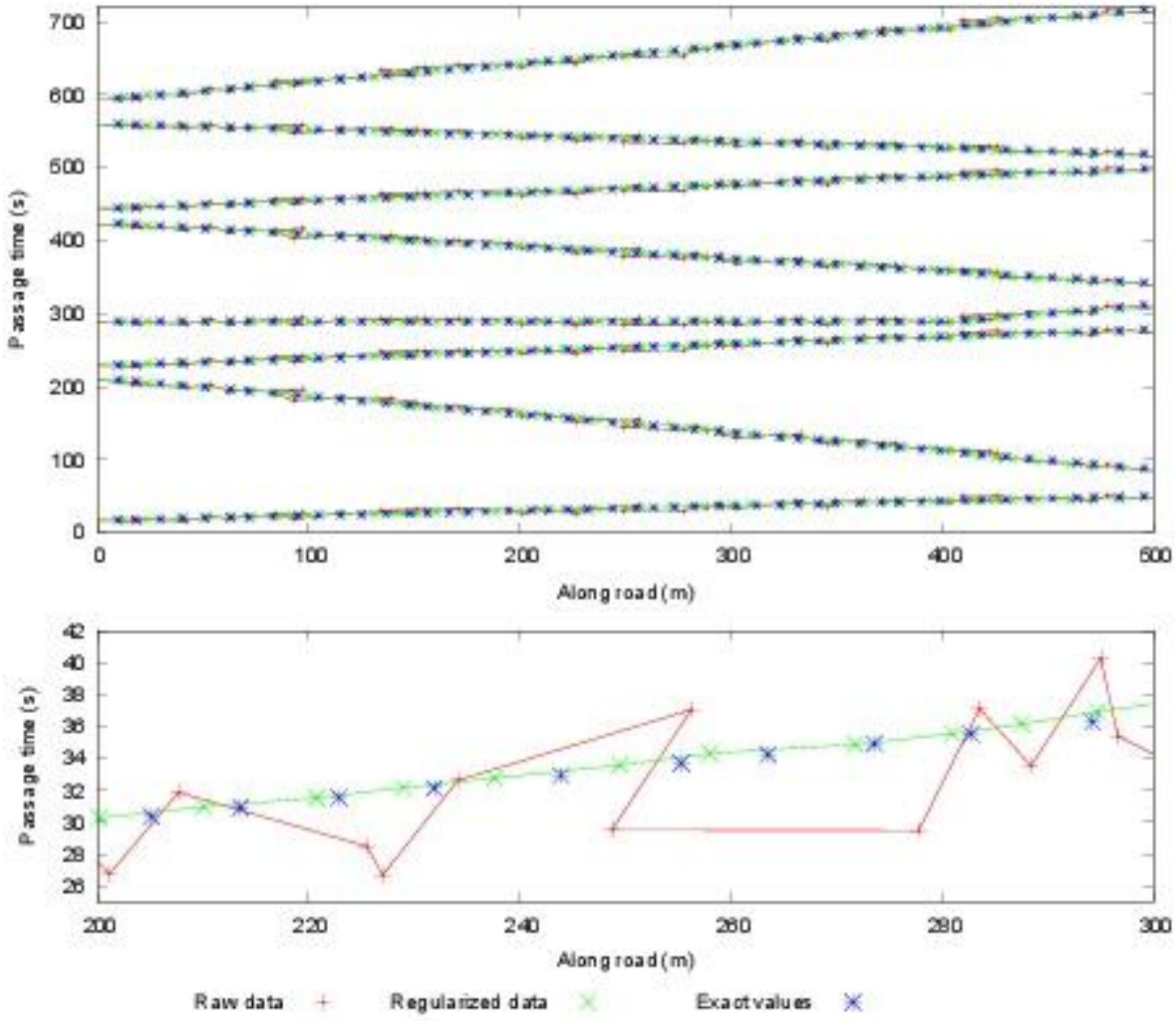
Datafusjon kan altså bidra til å utvide mulighetene for hvordan et sensornettverk kan operere uten tap av funksjonalitet. Datafusjon kan foregå i selve nettverket mens det opererer, eller det kan utføres som en behandling av gitte data i ettertid. Sanntids-datafusjon i selve nettverket gir mulighet til styring av målinger og datatransport avhengig av forutgående målinger. En kan da utnytte at måledata typisk er mer korrelerte dess nærmere de er i rom og tid. Noder kan

samarbeide om å filtrere vekk falske alarmer, eller de kan begrense seg til bare å komme med vesentlige korrigeringer til predikerte stier, som produseres i sann tid i nettverket. Dette blir, løst formulert, som å anvende ordtaket «den som tier den samtykker».

Datafusjon og komprimert datainnsamling kan generelt sees på som å anvende Occam's razor [12], som går ut på å foretrekke de enklest mulige forklaringene til gitte data. Dette kan for eksempel bety at en prøver å finne færrest mulige stier som stemmer overens med dataene (for eksempel alarmer ved gitte steder/tider). Disse stiene må også være enkle og reflektere økonomisk kjøring som kan være transport mellom to eller flere steder. Occam's razor er nå for øvrig hyppig referert til i litteraturen angående komprimert måling av korrelerte data (i motsetning til data fra utfallsrom uten struktur og sammenheng).

Anta at m_0 er et mål som angir hvor godt en mengde Z av mulige stier passer med dataene. Z er altså et forslag til forklaring på dataene. La m_1 være antall kjøretøyer i henhold til Z. La m_2 være et mål på akselerasjoner (eller økonomi i kjøringene som kan være transport mellom steder). I henhold til Occam's razor vil en da foretrekke den mengden av stier/kjøretøyer som gir minst verdier for m_0, m_1, og m_2. Dette betyr at en i praksis må utføre en multiobjektiv optimalisering. Treffer forslaget Z virkeligheten, vil alternative forklaringer gjerne skåre dårligere for alle målene hver for seg, gitt begrensninger for de andre målene. Merk at feil ved noder kan inngå som en del av forklaringene til dataene. En forklaring, dvs. mengden av mulige stier, vil gi estimater på «falske» alarmer. Dersom disse falske alarmene er lite korrelerte betyr dette gjerne «pluss-poeng» for den foreslåtte forklaringen.

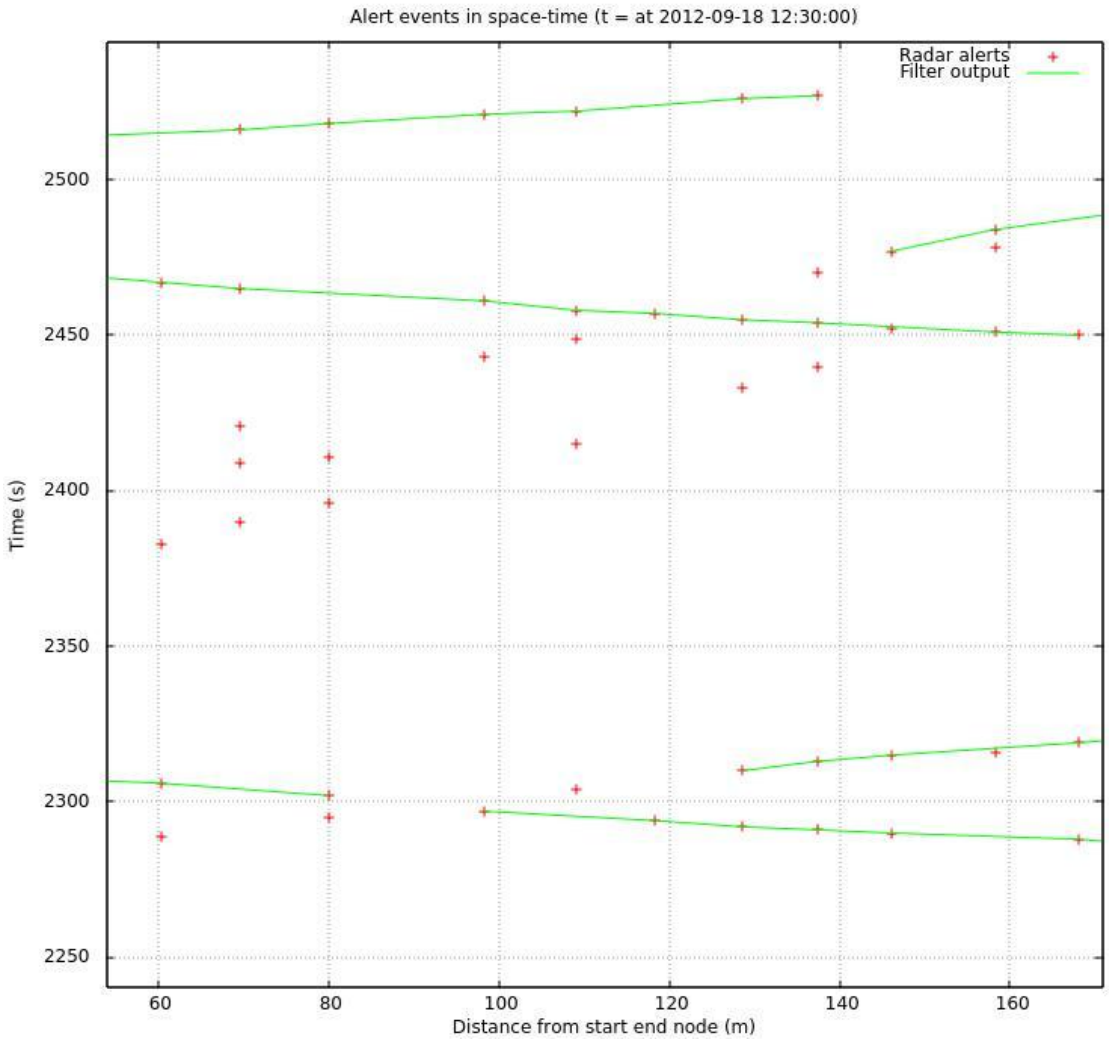
Evolusjonær programmering og mer generelle stokastiske myke metoder for dataanalyse [13], kan brukes for å utføre dataanalyse etter retningslinjene ovenfor, der en fleksibelt kan definere mål på hvor god en forklaring på data er. Dette er demonstrert i [11] med simulerte data som illustrert i Figur 3.3. Denne illustrasjonen viser tid og sted for alarmer i et nettverk langs en veistrekning. Den nederste delen av illustrasjonen viser et detaljert utdrag av den øverste delen. Den grønne linjen er et estimat av en sti som «forklarer» dataene ved regularisering. Denne forklaringen inkluderer estimater av feil i tid og sted for alarmene. Dette betyr at regulariseringen produserer korrigeringer av feil i klokker og posisjoner til nodene i nettverket og slik genererer et situasjonsbilde som enklest mulig forklarer dataene. En forutsetning her er at klokkene går tilnærmet like fort (men er gjerne initiert innbyrdes forskjellig) og at nodene ikke flyttes på i tiden dataene produseres.



Figur 3.3 Resultat fra regularisering av alarmdata som angir tid og sted for alarmer.

3.4.1 Filter for falske alarmer

Datafusjon kan brukes til å redusere forekomst av falske alarmer i et sensornettverk eller til å øke sensitiviteten til sensorer uten å skape ødeleggende falske alarmer. Figur 3.4 illustrerer dette. Figuren viser tid og sted for alarmer langs en veistrekning (fra veiakse forsøket). Alarmene utløses i en node i nettverket når et kjøretøy på veien passerer den. Hendelser som er knyttet sammen med grønn linje er alarmer som samsvarer med at et kjøretøy passerer i tilnærmet konstant fart mellom noder i nærheten (i en avstand mindre enn 35 m). En alarm som slik inngår i en serie av alarmer som samsvarer med et passerende kjøretøy med tilnærmet konstant fart, kan da klassifiseres som en sikker alarm (ikke falsk) og formidles til bruker. Legg merke til at denne måten å filtrere data/alarmer på kan brukes til å gjøre sensorene mer sensitive enn ellers uten at det produseres for mye falske alarmer.



Figur 3.4 Resultat fra identifikasjon av nære alarmer som kunne ha vært utløst av forbipasserende kjøretøy som varierer farten lite over 30 m. Grønn linje forbinder her slike alarmdata.

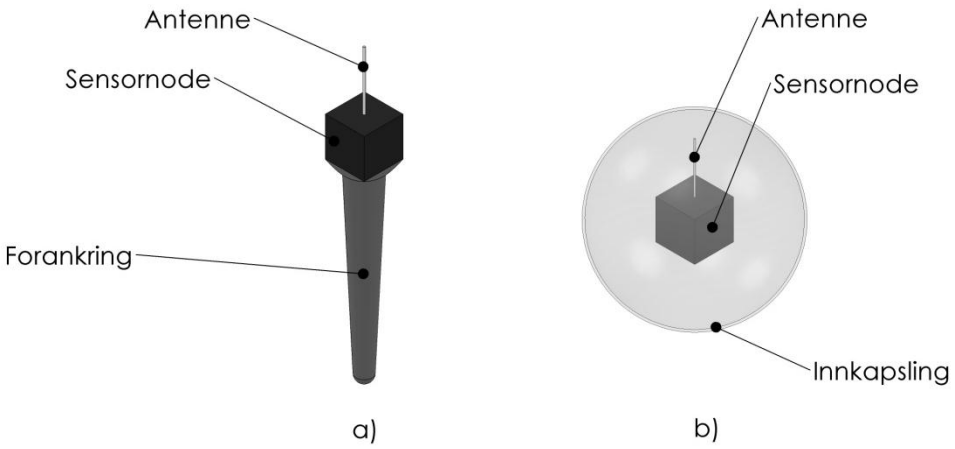
3.5 Sensorer

Sensoren(e) er nodens ”øyne og ører” og må velges ut ifra hva som sensornettverket skal detekterte. En rekke forskjellige sensorteknologier eksisterer som på hver sin måte vil ha stor innvirkning på utførelse av noden og det trådløse sensornettverket. Relevante parametere er:

- Følsomhet, rekkevidde og retning
- Størrelse
- Energiforbruk
- Kompleksitet
- Pris

Figur 3.5 illustrerer to sensornodekonsepter. Konseptet i Figur 3.5 a) har en for forankring og må utplasseres manuelt. Dette betyr for eksempel at en retningsbestemt sensor eller en sensor som måler vibrasjoner i bakken (geofon) kan benyttes. Når det gjelder konseptet i Figur 3.5 b) så er sensornoden plassert inne i en ball. Dette gjør at sensornoden kan utplasseres fra for eksempel et fly. Orientering vil imidlertid bli mer vilkårlig og sensoren som benyttes kan derfor ikke være retningsbestemt etc. Mer detaljerte betraktninger angående forskjellige sensorteknologier er gjort i [1].

En mikrofon er et eksempel på en sensor som ikke er retningsbestemt og kan være egnet for forskjellige sensornodekonsepter. Med dagens forbrukerelektronikk så har det blitt utviklet kraftige, men energieffektive prosesseringskretser for blant annet lyd. Disse har også gjerne ferdige utviklede algoritmer for å fjerne støy/vindstøy, detektere stemmer, filtrering, koding, etc. Deteksjon og klassifisering av endringene i en nodes omgivelsesparametre kan enten gjøres ved lokal prosessering på node eller ved global prosessering på sink. Ved lokal prosessering på node vil man kunne representere avvik med liten datapakke. Dette vil redusere datamengden som må sendes over nettverket. Det motsatte er tilfellet ved global prosessering på sink. Da vil datapakken som sendes over være en representasjon av målingen fra sensor. Dette gir økt datamengde over nettverket. Den optimale organiseringen av deteksjon og klassifisering er avhengig av scenario, batterilevetid, størrelse og pris på noder, rutingprotokoll, med mer.



Figur 3.5 Illustrasjon av to sensornodekonsepter.

For å redusere datamengden over nettet, er det fordelaktig med en viss lokalprosessering i noden. Det vil være ønskelig å skille mellom forskjellige avvik som kan trigge sensoren. Avvik som ønskes detektert kan være mennesker, store og små kjøretøy, kjøretøy med hjul eller belter. Samtidig kan det være avvik som ikke ønskes detektert, slik som vind, regn, dyr, eller annen naturlig lyd. Å ta en første avgjørelse på noden vil kunne spare nettverket for mye unødvendig trafikk.

I prosjektet så ble det derfor gjort noen forsøk med en lydprosesseringskrets. En MEMS (mikroelektromekaniske system)-mikrofon ble også benyttet. MEMS-teknologien har muliggjort

mikrofoner og andre sensorer som er små, bruker lite energi og som tilbyr høy følsomhet til en meget lav pris.

Det å implementere MEMS mikrofonen og lydprosseseringskretsen sammen med den tilgjengelige noden, ville imidlertid kreve utvikling av ny hardware. I tilegg så ble integrasjonen mot den tilgjengelige sensornoden tidkrevende. For de praktiske forsøkene så ble det derfor valgt å benytte PIR, lydsensor, og radar. En mer detaljert beskrivelse av disse sensorene er beskrevet i avsnitt 4.2.1 til 4.2.3.

3.6 Energihøsting

I utgangspunktet er det tenkt at et SASS-system skal kraftforsynes med batterier. Disse vil være begrensende for levetiden til systemet. Hvis sensornodene kunne kraftforsynes lokalt fra omgivelsene vil det øke levetiden og dermed nytteverdien av et slikt system betydelig. Det er gjort undersøkelser om det finnes tilgjengelig teknologi som kunne brukes til dette. Modenheten til teknologien skal vurderes og eventuelt inkluderes i et demonstrasjonssystem.

Fra prosjektet har det vært presentert to artikler på internasjonale konferanser [14] og [15].

Artikkelen i [14] undersøker muligheten for å bruke solcellepaneler som energikilde for WSN i Nord-Europa. Ved å bruke en energimodell for en sensornode sammen med PVGIS-databasen[1], har man regnet ut den påkrevde størrelsen av et solcellepanel som må til for å levere nok energi til sensornoden. Med en aktiv periode på 48 ms per minutt, må størrelsen på panelet være større enn 1,1 cm^2 i Roma, 1,5 cm^2 i Berlin og 1,4 cm^2 i Oslo i juli. I desember øker dette til 5,1 cm^2 i Roma, 17,4 cm^2 i Berlin og 58 cm^2 i Oslo. Lenger nord blir størrelsen for stor for alle praktiske formål.

Artikkelen i [15] ser på hvor mye energi som kan høstes for et solcellepanel på kredittkortstørrelse på seks forskjellige breddegrader i Europa. En sensornode kan bruke sovemodus for å øke levetiden til et WSN. Dette kan gjøres når datakapasiteten er mye høyere enn datatrafikken som må overføres over nettverket. Da kan radiosenderen slås av etter at en måling er sendt til en sink, og man kan se på forholdet mellom sove- og aktiv modus som et mål på om den tilførte energien kan gi et WSN nok aktiv tid. Med en typisk WSN node og beregninger fra PVGIS-databasen viser man at det er vanskelig å få akseptable resultater gjennom vintermånedene. Ved 50° og 60° nordlig breddegrad blir sove/aktiv raten omtrent 500 og 1500. Nord for polarsirkelen vil ikke systemet fungere i det hele tatt. I sommermånedene er det svært lite forskjell i innstrålt energi mellom de forskjellige breddegradene, ettersom den økte innstrålingsvinkelen ved høyere breddegrader blir kompensert med lengre dager. Disse resultatene er for standardelektronikk. Ved bruk av den mest moderne elektronikken med høyere effektivitet kan sove/aktiv-raten økes med fire.

[1] PVGIS-databasen gir data om estimert energi fra sol stråling for en gitt lokasjon i Europa og Afrika.

4 Praktiske forsøk

4.1 Innledning

En viktig del av prosjektet er å demonstrere bruk av et WSN gjennom en demonstrator. Dette er gjennomført som en CD&E aktivitet «EP 1257 Trådløst sensornettverk». Hensikten med en slik demonstrasjon er:

- Formidling av de muligheter relatert til WSN-teknologien som er realiserbar i dag og eventuelt kan forventes i nær framtid.
- Evaluering av systemet som er utviklet og den operative nytteverdien som oppnås.
- Bidra til økt forståelse av behovene i Forsvaret innen oppklaring og sikring.

Omfanget av forsøket er bestemt til å være begrenset til scenarioene A og B, dvs. perimetersikring og overvåking av veiakse. Disse forsøkene er gjennomført henholdsvis på Rena (3-6. september 2012) og Kjeller (18. september 2012).

Som et ledd i forberedelsen til disse forsøkene, har det vært stor aktivitet relatert til oppbyggingen av SASS systemet. I denne aktiviteten inngår det design, implementering av hardware og software, og kontinuerlig testing og feiloppretting. En gjennomgang av dette arbeidet er beskrevet i avsnittene 4.2 og 4.3, mens en kortfattet beskrivelse av resultatene fra forsøket er gitt i avsnitt 4.4.

4.2 Hardware

For å demonstrere konsepter i SASS-prosjektet og det etterfølgende CD&E eksperimentet, ble det bestemt å lage 50 hardware enheter som ble kalt sensornoder. Disse måtte være ferdig til oppsatt tid for test i samarbeid med Hærens taktiske treningssenter på Rena i begynnelsen på september. I praksis måtte disse selvfølgelig være ferdig en del tidligere for funksjonstest på FFI før reisen til Rena.

Funksjonelt sett består sensornodene av 4 deler: lydmottaker, IR-mottaker, radar og en CPU/mikrokontroller som styrer det hele. Hvis en ser nøyere på hva som trengs vil en se at det er nødvendig med noen flere deler, det trengs en boks som beskytter mot vær og vind og samtidig holder elektronikk-kortene i hensiktsmessig stilling. Boksene ble montert på en pinne for å stå stabilt og samtidig få en viss avstand til bakken. Det var også nødvendig med 2 batteriholdere for at alle enhetene skal få riktig spenning, samtidig gir dette noden forlenget levetid. En sensornode består av 3 kretskort. Lyd og IR-mottakeren ligger på et kretskort som er designet på FFI,

Figur 4.1 Til venstre: egenutviklet sensorkort med passiv IR (under fresnel-linse) og mikrofon. Til høyre: dopplerradar fra Samraksh (foto © The Samraksh Company).

produsert eksternt og så ble det til slutt påmontert komponenter på FFI. Figur 4.1 viser sensorkort for PIR, audio og radar.

4.2.1 Lyd

Lydmottakeren består av en kondensator-mikrofon som skal spenningsmates med 1-10 V DC og en forsterkerkobling som gir et passende nivå til A/D omformeren på kontrollerkortet samtidig med at signalet filtreres mellom 270 Hz og 3,4 kHz. Mikrofonen som ble valgt er en PWM-6052-5383-7GM fra Vansonic (Elfa 30-106-59). Denne mikrofonen har en innbygd forforsterker. Vår forsterkerkobling er realisert vha. en operasjonsforsterker LM7301 og noen diskrete komponenter.

4.2.2 IR-mottaker

IR-mottakeren er laget på tilsvarende måte ved at signalet fra en enhet går gjennom en tilpasningskrets designet ved FFI og så til A/D omformeren på kontrollerkortet. Terskling av signalet og en viss justering av følsomheten kan en da gjøre i software på kontrollerkortet. IR-mottakeren er helt passiv (PIR) og er bygget opp på tilsvarende måte som brukes i tyverialarmdetektorer for innendørs bruk. Detektoren bygger på såkalt pyro-elektrisk effekt. Enkelte keramiske materialer har egenskaper som gjør at hvis stoffene absorberer termisk energi endres polarisasjonen i overflaten og det genereres en overflate-ladning. Selve mottakerelementet RE 200B fra Nippon Ceramics (Elfa 75-224-06) er følsomt for stråling fra objekter med en temperatur innenfor et visst område. For å redusere følsomheten for temperaturvariasjoner i stillestående objekter består detektoren av 2 elementer. Det er differansen mellom disse som forsterkes opp og danner utgangssignalet. Dette formes så av en forsterkerkobling og føres til A/D på kontrollerkort.

For å realisere denne funksjonen er det nødvendig at varme objekter i bevegelse gir forskjellig innstråling på de 2 detektorelementene slik at det blir en forskjell å forsterke opp. Denne effekten kan en få til på forskjellige måter med innretninger som skaper delvis skygge, fasettspeil eller en IR-linse med fasetter. Vi valgte en fresnel-linse 9425 fra Murata (Elfa 75-224-36) med en stor åpningsvinkel (i størrelsesorden 150 grader). Dette er en fasettlinse som sørger for at stråling fra varme objekter først når fram til det ene elementet og så til det andre. Linsen består av 37 detekteringssoner, er skålformet og helt sirkulær. Den er i utgangspunktet designet for plassering i taket over objekter som skal overvåkes.

4.2.3 Radar

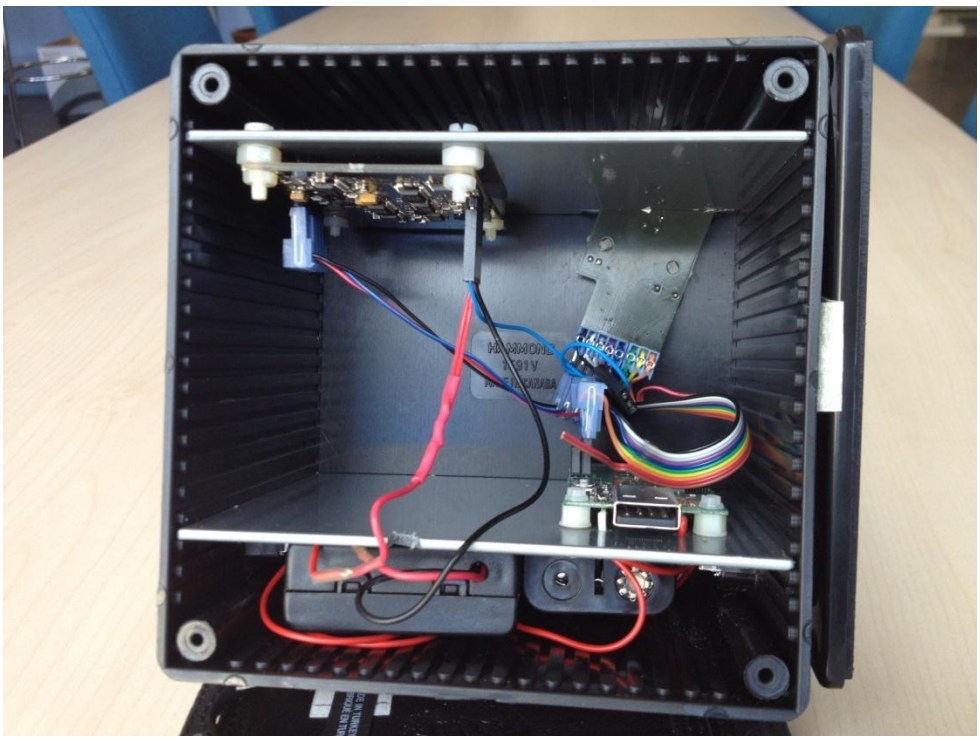
Radaren er en lav-effekt pulset doppler radar med en senter frekvens på 5,8 GHz og rekkevidde på ca 10 meter. Den er designet for deteksjon av personer f.eks i tyverialarmer. Den er utformet slik at den kan kobles direkte til A/D på kontrollerkortet. Det finnes ferdige TinyOS 2.x drivere for Telos B. Utsignalet fra radaren kan behandles forskjellig etter hva slags A/D og prosessorkraft en har til rådighet. Radaren gir ut både I og Q kanal (realdel og imaginærdel). Det er derfor mulig å bedre egenskapene til systemet hvis en har en rask A/D og stor prosessor-kapasitet. Radarkortet krever høyere drivspenning enn resten av systemet og forsynes derfor fra egne batterier i en egen batteriholder. Rekkevidden til radaren skal kunne kontrolleres vha. software mellom 1 og 10 m. Dekningen kan beskrives med et 60 graders konisk mønster. Den reagerer på radielle hastigheter mellom 2,6 cm/s og 2,6 m/s.

Radaren's senterfrekvens på 5,8 GHz gir en bølgelengde på omkring 5,2 cm. Antennen til radaren ligger på kretskortet og det har vist seg å være nødvendig at denne er minst 5 cm unna større metallobjekter som batterier o.l. Dette har gitt føringer for størrelse og utforming av innkapsling av elektronikken. For å få til tilstrekkelig stor avstand har vi valgt å bruke en stor boks med målene 120 mm x 120 mm x 94 mm og innvendige riller for montering av kretskort. Dette gjør det mulig å få en avstand på ca. 65 mm mellom radarkortet og batteriholderen. Kretskortene for mikrokontroller og radar ble skrudd fast med plastskruer i 2 mm tykke PVC-plater.

4.2.4 Innkapsling

Boksene som var levert av Hammond Manufacturing ble festet med strips til noen svært solide trepinner med tverrsnitt: 27mm x 55 mm og varierende lengde. Dette gav en viss antennehøyde over bakker. Dette har tidligere vist seg å ha stor betydning for rekkevidden på radioen.
Valget av innkapsling bør sees i lys av at hensikten med CD&E eksperimentet var å demonstrere potensialet til WSN. Det var aldri snakk om å lage en prototyp. Skulle vi lage en prototyp hadde det krevd mer ressurser mtp. tid og økonomi. Da ville vi valgt helt andre løsninger for å gjøre boksen vanntett, enklere skifte av batteri og brytere som var mer motstandsdyktige mot uttilsiktet aktivering.

Boksen har ingen pakning og vi hadde heller ikke ambisjon om å gjøre den vanntett da bryterne ikke har tilstrekklig tetting slik at boksen kan bli helt tett. Resten av boksen kunne ha vært tettet med silikon, men dette ville vanskeliggjøre batteriskift. Boksen har innvendige riller som skal kunne holde kretskort på plass hvis disse har akkurat riktig størrelse, men våre kretskort er mye mindre. Vi laget derfor 2 plater av 2 mm PVC som passer inn i rillene og skrudde fast radio og kontrollerkort til disse PVC-platene, som vist i Figur 4.2. Sensorkortet (IR og lyd) holdes på plass av en utsparring i den ene platen. Batteriholderne holdes bare på plass ved at plassen er liten. Den ferdige sensornoden sett fra utsiden er vist i Figur 4.3.



Figur 4.2 Viser innsiden av boks med sensorer.

Figur 4.3 Viser utsiden av boks med IR-linse som er montert i boksens side.

4.2.5 Mikrokontroller

Mikrokontrollerkortet er av typen TelosB Mote TPR2420 fra Memsic Inc. Dette er selve hjernen og er uten tvil den viktigste komponenten i systemet. Dette er designet rundt mikrokontrolleren MSP430 fra Texas Instruments. Denne kjører software for behandling av sensorinput og andre

viktige oppgaver som ruting i nettet. For kommunikasjon med andre enheter brukes kretsen CC2420 som følger IEEE 802.15.4 og tilbyr en datarate på 250 kbps i ISM båndet mellom 2,4 og 2,4835 GHz. RF transceiveren bruker en invertert F-antenne som er integrert på kortet. Denne skal gi en rekkevidde på 20 m til 30 m innendørs og 75 m til 100 m utendørs. Forsøk har vist at høyden over bakken har svært stor betydning for rekkevidden [2]. Det er også mulig å bygge om til ekstern antenne. Sensorer for lys i forskjellige bølgelengder, temperatur og fuktighet er integrert på kortet. TelosB programmeres via USB fra en PC. Kommunikasjonen går via en FTDI FT 232BM USB mikrokontroller.

En viktig del av sensornoden er den 12 bit analog til digitalkonverteren (A/D). Alle våre tre sensorer bruker denne for å registrere det som skjer rundt noden. I forbindelse med denne A/D-en er det en 8 kanalers multiplekser, slik at enheten kan behandle inntil 8 forskjellige analoge signaler med nivå mellom 0 og 3 V. To av disse ADC 4 og ADC 5 er allerede i bruk til lys-sensorer som er montert direkte på kortet. Mikrokontrollerkortet har 2 konnektorer for tilleggskort, en 10 pin ekspansjons-konnektor U2 og en 6 pin konnektor U28. I vår demo ble følgende pinner benyttet:

- IR ble koblet til pinne 1 i 6 pin-konnektor U28 (Analog Input 6, ADC6)
- Audio ble koblet til pinne 5 i 10-pin-konnektor U2 (Analog Input 1, ADC1)
- Radar realdel ble koblet til pinne 3 i U2 (Analog Input 0, ADC0)
- Radar imaginærdel ble koblet til pinne 7 i U2 (Analog Input 2, ADC2)
- Alle sensorene ble jordet i pinne 9 i U2 (Analog Ground)
- IR og Audio fikk strøm fra pinne 1 i U2 (Analog VCC)

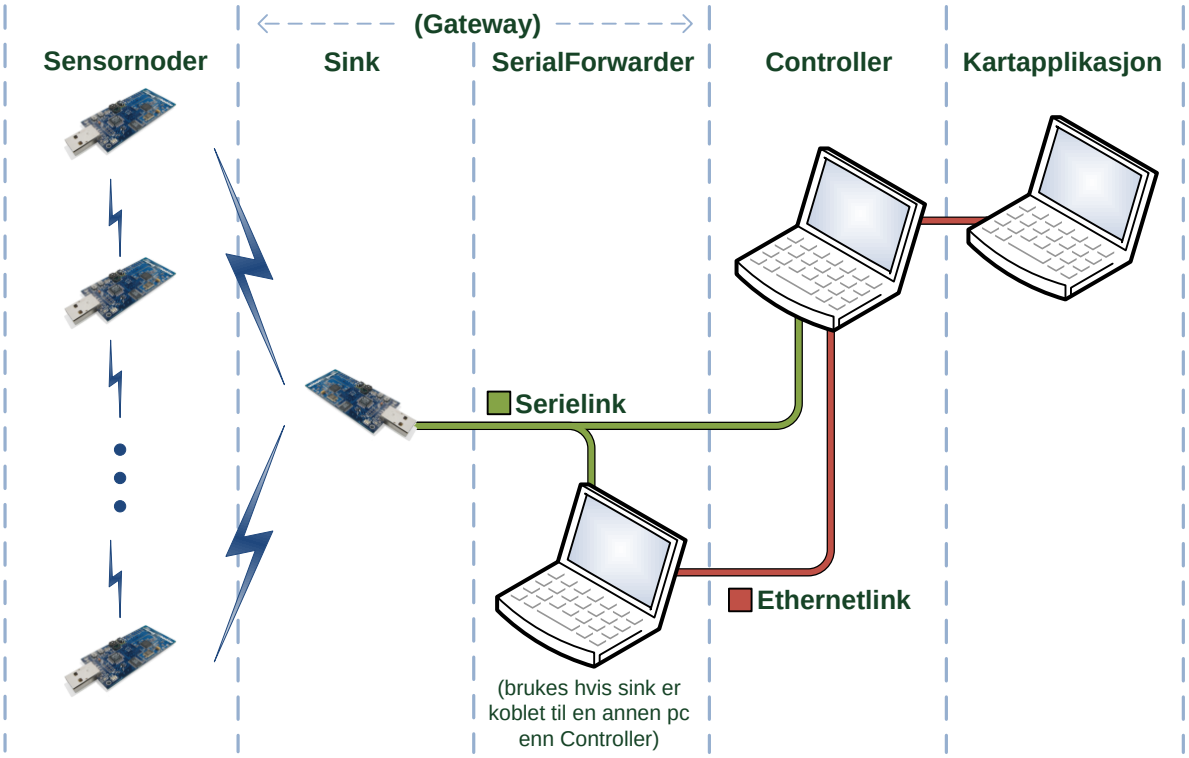
4.2.6 Strømforsyning

Strømforsyningen kommer fra USB-pluggen og/eller batterier. Sinknoden drives av tilkoblet PC gjennom USB-porten. Radarkortet trenger minst 3,65 V og drives av 3 AA batterier montert i en egen batteriholder. Denne er koblet til en egen bryter på boksen, slik at radaren kan slås av hvis når den ikke brukes. Resten av noden strømforsynes fra USB eller 2 AA batterier i annen batteriholder. Under eksperimenteringen ble det brukt 2 stk spesialbatterier for denne oppgaven. Energizer Ultimate Lithium, som er et engangs lithium (Lithium/Iron Disulfide(LiFeS2)) batteri, ble valgt for å slippe batteriskift. Også fra denne batteriholderen ble det montert en bryter på forbindelsesledningen til mikrokontrolleren.

4.3 Programvare

Vi valgte å bruke TinyOS [16] som utviklingsplattform. Denne er utviklet nettopp for trådløse sensornettverk, og da særskilt for sensornoder som bl.a. TelosB. TinyOS er opprinnelig utviklet ved UC Berkeley, og kan lastes ned (åpen kildekode) fra linken gitt i [17].

TinyOS er et statisk-linket rammeverk som tilbyr operativsystemfunksjonalitet til applikasjonen det er linket sammen med. TinyOS er skrevet i nesC [18], som er et overbygg til standard C og som tilbyr komponentbasert struktur og hendelsesbasert programflyt. Egenutviklede applikasjoner



Figur 4.4 Meldingsflyt mellom komponentene. Alle linkene er toveis.

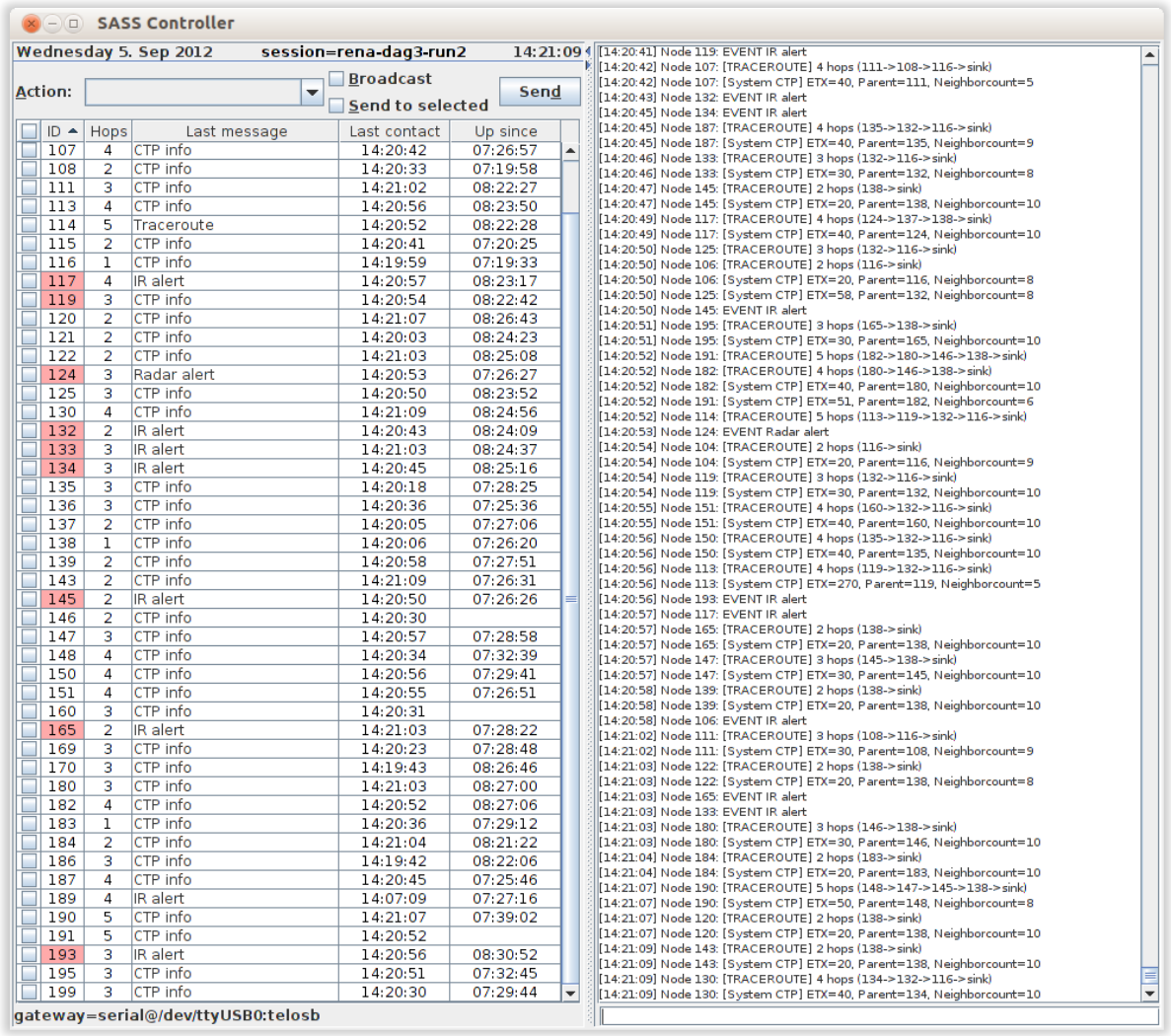
som skal kjøre på nodene skrives også i nesC, og linkes sammen med TinyOS-rammeverket for å produsere en enkeltstående binærpakke som kan lastes opp til noden.
Sammen med TinyOS følger det med SDK for et utvalg programmeringsspråk for å lage PC-programmer som kan interagere med nodene via USB/serieport. Vi valgte å bruke Java som plattform for de deler av systemet som må kjøres på PC, hovedsakelig fordi dette er det mest utbredte (og dermed best testede) SDK for TinyOS.

4.3.1 Komponenter

Systemet består i hovedsak av seks ulike komponenter dvs. sensornoder, sink SerialForwarder, Controller, kartapplikasjon samt GPS-posisjoneringskomponent. De fem første komponentene er vist i *Figur 4.4*. GPS-posisjoneringskomponent er beskrevet senere i avsnitt 0.

4.3.2 Controller

Controller-applikasjonen er ansvarlig for å styre og konfigurere sensornodene, samt å motta alarmmeldinger og sende disse videre til kartapplikasjonen (jfr.*Figur 4.4*). Figur 4.5 viser et skjermbildeeksempel. På høyre side er hendelsesloggen. Der vises alle meldinger som sendes og mottas, samt annen informasjon som Controller-applikasjonen gir. Denne loggen kan lagres fortløpende slik at man i ettertid kan gå tilbake og rekonstruere situasjonen på et gitt tidspunkt. Venstre side av skjermen lister alle sensornoder som er koblet til nettverket, samt en viss statusinformasjon for hver node.



Figur 4.5 Skjermbilde av Controller-applikasjonen. På høyre side vises bl.a. hvordan alle nodene regelmessig sender rutinginformasjon («Traceroute» og «System CTP») slik at vi i ettertid kan danne oss et bilde av kommunikasjonsstrukturen.

Grensesnittet inneholder svært få knapper og andre GUI-elementer; i stedet styres systemet ved å skrive inn kommandoer i tekstfeltet nederst til høyre. Programmeringsmessig skalerer dette veldig bra, og er enklere og raskere å få til enn å lage menyer og knapper for hvert valg som kan gjøres. Brukerterskelen blir selvsagt noe høyere, men siden dette verktøyet ikke er ment å bli brukt av eksternt personell er ikke dette vektlagt.

4.3.2.1 Logging (sesjoner)

For at feltforsøk skal kunne analyseres i ettertid er det viktig at alle hendelser og all node-informasjon blir logget. Controller-applikasjonen lar en inndele feltforsøk i *sesjoner* med beskrivende navn, som for eksempel «cdedemo-rena-day1-run1». Informasjonen som blir logget er:

- Liste over alle sensorer som er tilknyttet systemet, samt deres sist kjente status.
- All tekst i hendelsesloggen.
- Alarmer som er mottatt.
- Alle meldinger som Controller har sendt og mottatt.
- GPS-posisjon til sensorene, samt logg av denne dersom posisjonen har blitt endret

Merk at logging *kun* utføres når man har en åpen sesjon. Man kan når som helst opprette en ny sesjon eller laste inn en eksisterende. Dersom en sesjon er åpen når applikasjonen avsluttes, vil denne sesjonen automatisk åpnes igjen når applikasjonen startes på nytt.

4.3.3 Sensornoder

Nodenes primæroppgave er å analysere data fra tilkoblede sensorer og rapportere inn hendelser den oppfatter som viktige; altså når en eller flere sensoralgoritmer klassifiserer signalresultater som hhv. «bevegelse» eller «passering». Den må derfor også kunne sende og motta meldinger over radio. Den må ha en rutingfunksjonalitet som videresender meldinger ved behov, og den må kunne reagere på innkommende meldinger som for eksempel rekonfigurasjon eller statusforespørsler. I tillegg er det lagt inn støtte for klokkesynkronisering og GPS-posisjonering.

Koden for de enkelte sensorene (IR, radar og mikrofon) er modularisert slik at man må velge hvilke sensorer som skal brukes under kompileringen. Dette er hovedsakelig fordi MSP430-prosessoren som nodene bruker kun har 48K ROM tilgjengelig for kode, og det blir plassproblemer dersom all koden skal inn samtidig.

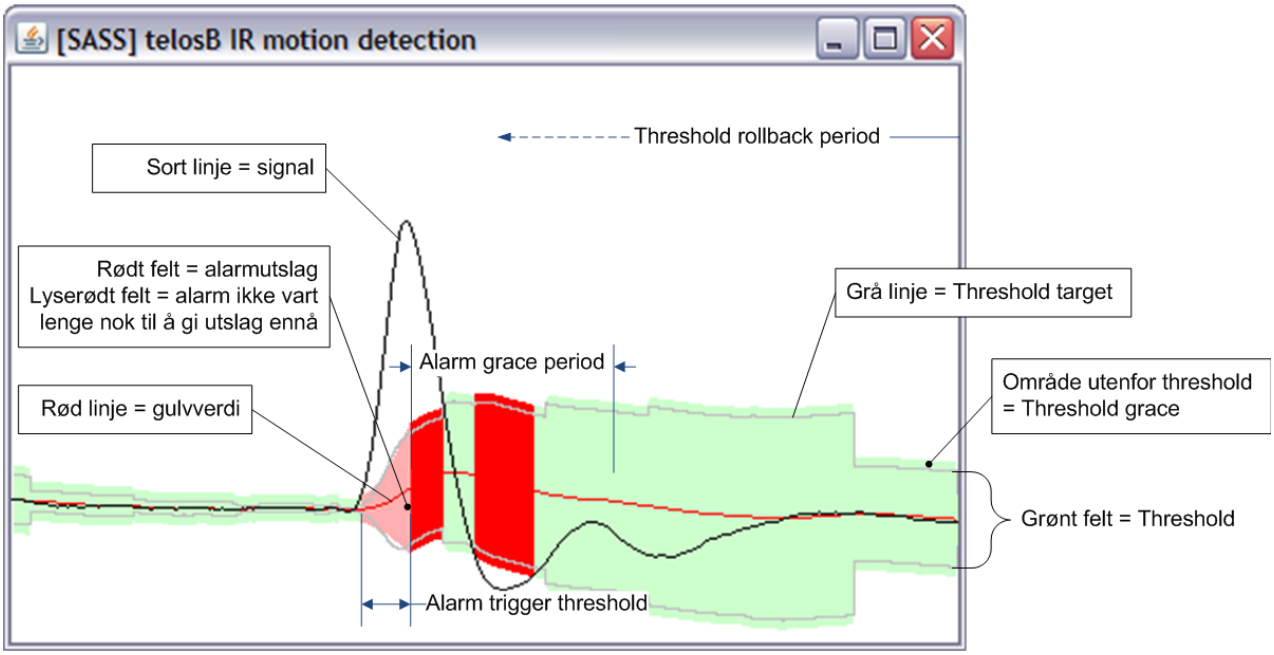
4.3.3.1 IR-sensor

Figur 4.6 er et plot fra en applikasjon laget for å hjelpe med algoritmetesting og parametersetting. Den viser innsignalet fra IR-sensoren (tykk sort linje), samt verdier som algoritmen beregner fortløpende ut fra signalverdien. Samplingsfrekvens på IR-signalet er 50Hz.

Algoritmen fungerer på følgende måte:

- En gulvverdi (rød linje på figuren) peiler seg inn mot signalverdien med en faktor definert i variabelen `floor_adjustment_factor`.
- Det defineres en terskelverdi («threshold target»; grå linje) rundt gulvverdien som er gjennomsnittet av signalets maksverdier de siste `threshold_rollback_period` sekunder.

- Terskelområdet (grønt område på figuren) utvider seg mot terskelverdien med en faktor definert i variabelen `threshold_adjustment_factor`. Innsnevring skjer 50 ganger raskere.
- Det legges til et `threshold_grace_value` område utenfor terskelområde.
- Hver gang signalet er utenfor terskelområdet økes en alarmteller. Dersom man i en periode på `alarm_threshold_period` samples får flere enn `alarm_trigger_threshold` alarmer regnes det som alarm. Alarmene behøver ikke være sammenhengende.
- Etter at alarm er trigget starter en `alarm_grace_period` der det ikke vil bli gitt nye alarmer. I eksempelfiguren er det indikert at det oppdages en ny alarm i denne perioden, men algoritmen vil ikke melde denne inn til Controller-applikasjonen.



Figur 4.6 Eksempel på plot fra IR-sensoralgoritme

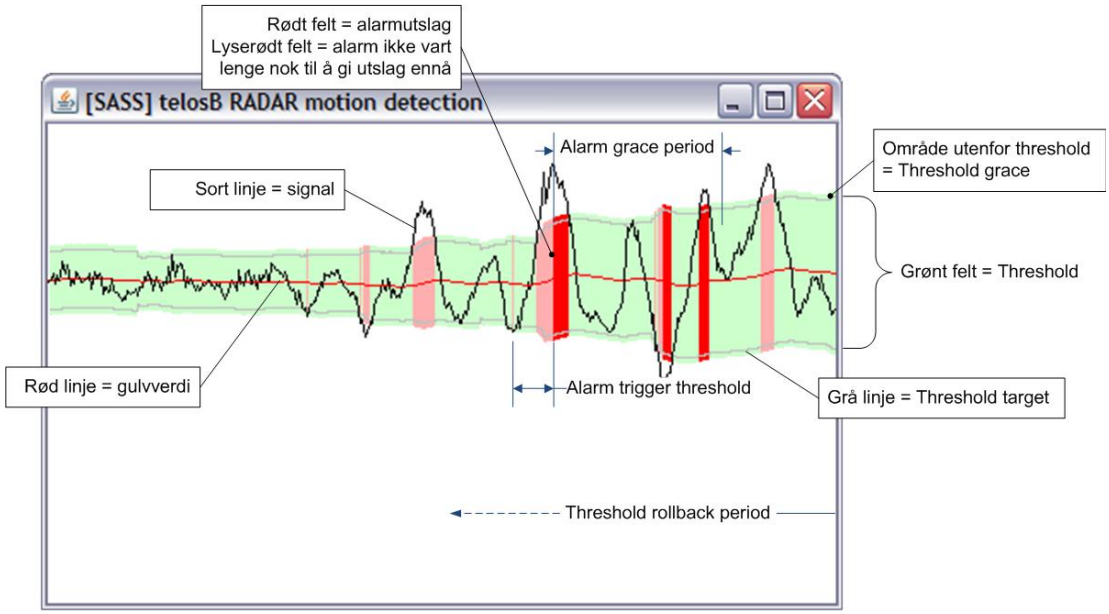
Parameter	Enhet
Alarm trigger threshold	samples
Alarm threshold period	samples
Alarm grace period	sekunder
Floor adjustment factor	0.1 prosent
Threshold grace value	bits
Threshold rollback period	sekunder
Threshold adjustment factor	0.1 prosent
Saturation discovery enabled	0 (=false) eller 1 (=true)
Saturation threshold period	samples

Tabell 4.1 Parametre som brukes av IR-sensoralgoritmen og radaralgoritmen

IR-sensoren blir veldig påvirket av værforandringer; spesielt i overgangen mellom sol og skyer. Det som skjer da er at signalet gir maks utslag over en lengre periode (typisk noen sekunder). Normalt vil dette gi alarm, så det er lagt inn en ekstra sjekk som hindrer algoritmen i å gi alarm dersom utslaget holder seg i metning over en gitt tid. Dette medfører dog at algoritmen må vente til denne perioden er over før den kan sende alarm. Denne funksjonaliteten kan slås av (`saturation_discovery_enabled`) og perioden kan justeres med parameteret `saturation_threshold_period`. Tabell 4.1 oppsummerer de IR/radar parametrene som kan endres fra Controller under kjøring.

4.3.3.2 Radaralgoritme

Radaralgoritmen er identisk med IR-algoritmen, med unntak av metningssjekk som ikke er relevant for denne sensoren. Figur 4.7 viser at radarsignalet har mer støy og hyppigere svingninger enn IR-signalet, og parameterverdiene må justeres deretter.



Figur 4.7 Eksempel på plot fra dopplerradaralgoritme

4.3.3.3 Audiosensor (mikrofon)

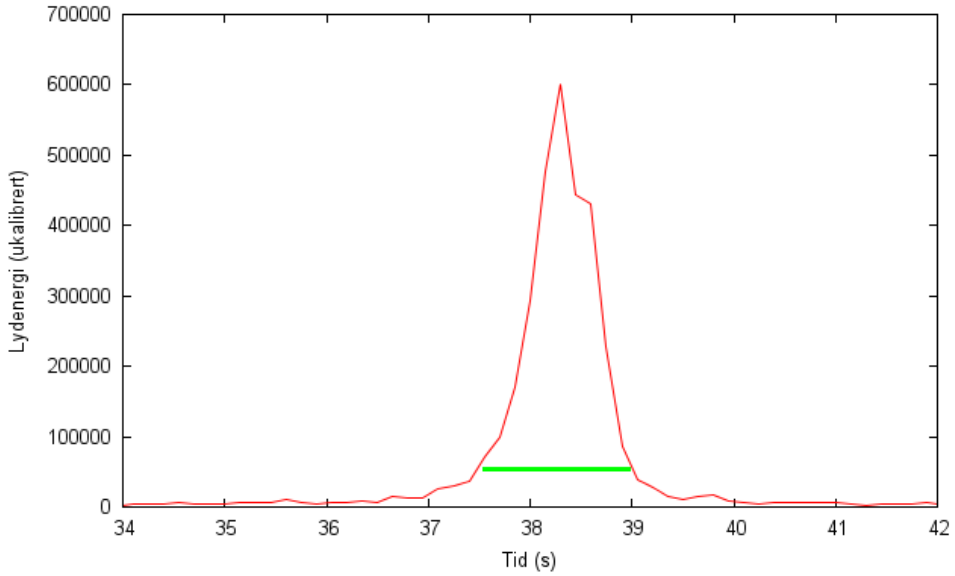
Denne sensoren benytter en helt annen algoritme enn de to foregående. I stedet for å bare registrere bevegelse, skal vi her detektere passering av motoriserte kjøretøy. Figur 4.8 viser et eksempel på estimert variabel lydintensitet ved en node langs en vei under passering av bil. Dersom det er en sammenhengende periode over en gitt tid (variabelen `Period`) med lyd-intensitet over et gitt nivå (variabelen `threshold`) vil dette bli identifisert som en hendelse (sannsynligvis en bilpassering). Lyden fra denne hendelsen rundt tidspunktet for maksimum lyd blir så analysert for geometriske egenskaper (inkludert symmetriegenskaper). Tabell 4.2 viser de parametrene for audio som kan endres fra Controller under kjøring.

Prosessoren i noden har ikke kapasitet til å lese og behandle sensordata samtidig ved en samplingsfrekvens på 4 kHz. Den har for eksempel ikke kapasitet til å kjøre et enkelt rekursivt filter på dataene for å estimere glidende middel og minimums-nivå («gulv») i dataene. Dette

problemet løses ved at algoritmen leser inn en sekvens på 200 trykk-målinger i et buffer, for så å ta en pause på 0.1 s som brukes til å behandle dataene. Denne raske målesekvensen varer altså 200/ 4000 = 0.05 s (50 millisekunder) etterfulgt av en pause på 0.1 sekund (dvs algoritmen utfører måleserier hvert 0.15 sekund).

Parameter	Enhet
Threshold	amplitude
Period	sekunder

Tabell 4.2 Parametre som brukes av audiosensoralgoritmen



Figur 4.8 Eksempel på estimert lyd-intensitet ved node på siden av vei når det passerer en bil forbi stedet (her en VW Golf diesel modell 2009). Grønn strek illustrerer tid (ca 1.5s) for sammenhengende lyd-nivå over 50000 (ukalibrert verdi).

4.3.4 Rutingfunksjonalitet

Når nodene er plassert ut må de etablere et stabilt nettverk slik at alle noder kan sende og motta meldinger. Det er mulig for nodene å sende meldinger til hverandre, men i dette systemet foregår meldingsflyten kun mellom node og sink (på vegne av Controller-applikasjonen). All radiokommunikasjon baserer seg på en av to rutingprotokoller, *CTP* [7] og *Dissemination*, avhengig av hvilken retning meldingen går (til eller fra sink). Begge disse protokollene er en del av standardbiblioteket til TinyOS.

4.3.4.1 Collection Tree Protocol (CTP)

CTP-ruting sørger for at alle noder har en vei (rute) inn til sinknoden (og dermed videre til Controller-applikasjonen). Denne protokollen baserer seg på at det er lite mobilitet i nettet, og fungerer best når det er mange redundante/overlappende radiolinker.

TinyOS-komponenten er laget slik at rutingen foregår i bakgrunnen, og vår applikasjon skal normalt ikke behøve å bry seg mer om den straks den er satt opp. Den har allikevel støtte for å la vår applikasjon se på og eventuelt gjøre endringer i meldinger før de sendes videre til neste hopp, noe vi benyttet oss av for å kunne implementere en traceroute-funksjonalitet for å logge (i Controller-applikasjonen) hvilke noder meldingen har vært innom på tur tilbake til sink.

4.3.4.2 Dissemination Protocol

Dette er en flooding-protokoll som sender meldinger fra sink til alle noder i nettet. Den er i utgangspunktet ment brukt for å sende konfigurasjonssettinger til noder, og har mekanismer som sørger for at nye noder som kommer inn i nettet, alltid vil motta sist sendte melding[2], og dermed alltid ha en oppdatert konfigurasjon. Algoritmeparametrene til de forskjellige sensorene er definert som separate konfigurasjoner, og denne protokollen passer dermed godt til å distribuere eventuelle endrede parametre ut i nettet.

Siden dette er en flooding-protokoll egner den seg dårlig til å sende meldinger til kun en spesifikk node. Dette er en funksjonalitet vi likevel krever, men i stedet for å lage vår egen protokoll[3] valgte vi å benytte Disseminationprotokollen selv med de ulemper det medfører. Vi definerte et felt i meldingspakken til å være destinasjons-id, slik at selv om alle nodene mottar meldingen kan de som ikke har samme id-nummer ignorere den. Dette fungerer tilsynelatende bra, men gir følgende ulemper:

- Det skaper mer radiotrafikk siden meldingen vil bli distribuert via alle radiolinker i stedet for noen få hopp som normalt vil være tilstrekkelig.
- Meldingen blir «liggende i nettet» og sendt på nytt til nye noder eller noder som er restartet. Dette kan skape problemer dersom meldingen f.eks. var en kommando til en node om å restarte seg selv. I det øyeblikket noden kommer opp igjen vil den da motta kommandoen på nytt, så det måtte legges inn mekanismer for å unngå dette.
- Hvis samme melding skal sendes til flere noder vil hver melding bli «overskrevet» av den som kommer etterpå. En melding som bruker for lang tid på å komme frem til destinasjonen risikerer derfor å bli droppet ut av nettet hvis det kommer en ny melding kort tid etterpå. Husk at Dissemination-meldinger er egentlig et øyeblikksbilde av en ønsket status, og det kan til enhver tid kun eksistere *en* slik melding av hver type i nettet.

4.3.5 Tidssynkronisering

Tidssynkronisering er en viktig komponent i et distribuert system som et sensornettverk. Dette muliggjør at noder i nettverket har en fellesforståelse av tiden, og dermed kunne danne et riktig bilde av hendelsesforløpet. I veiaksescenarioet, der hastigheten på et mål er normalt høyere enn i perimetersikringscenarioet, er dette spesielt viktig for å presist bestemme hendelsesforløpet og eventuelt hastigheten.

[2] Det kan defineres flere *typer* meldinger (kalt AM-type) som identifiseres med nummer. Dissemination-protokollen sørger for at nye noder mottar siste kopi av *hver* meldingstype som er definert.
[3] TinyOS har ingen standard rutingprotokoll som tillater dette.

I TinyOS-rammeverket tilbys det en ferdig implementasjon av FTSP, som er en tidssynkronisering protokoll utviklet spesielt for trådløse sensornettverk. Ved hjelp av denne protokollen er det mulig å oppnå en presisjon på millisekundersnivå. Det er imidlertid en svakhet i den originale versjon av protokollen at konvergeringstiden, dvs. den tiden det tar for alle noder i nettverket å bli synkronisert, kan være lang. Dette er særlig et problem hvis størrelsen og radiusen på nettverket er stor, hvilket er analysert og påvist i en intern rapport [19]. For å løse dette problemet er FTSP modifisert til å gå gjennom to faser ved oppstart: en *initiell fase* og en *operasjonell fase*. I den initielle fasen utveksles synkroniseringsmeldinger hyppig for at nettverket raskt skal bli synkronisert. Når nettverket så er synkronisert vil FTSP gå over til operasjonell fase der intervallet på synkroniseringsmeldinger reduseres for å minimere bruken av ressurser som båndbredde, prosesserings- og batteri-kapasitet. Forsøk viser at denne løsningen reduserer konvergeringstiden betraktelig samtidig som presisjonen på noen millisekunder ivaretas.

4.3.6 Gateway

Gateway er kommunikasjons-bindeleddet mellom Controller og sensornodene, som vist på *Figur 4.4*. Den består av to komponenter: *Sinknode* og *SerialForwarder*.

4.3.6.1 Sinknode

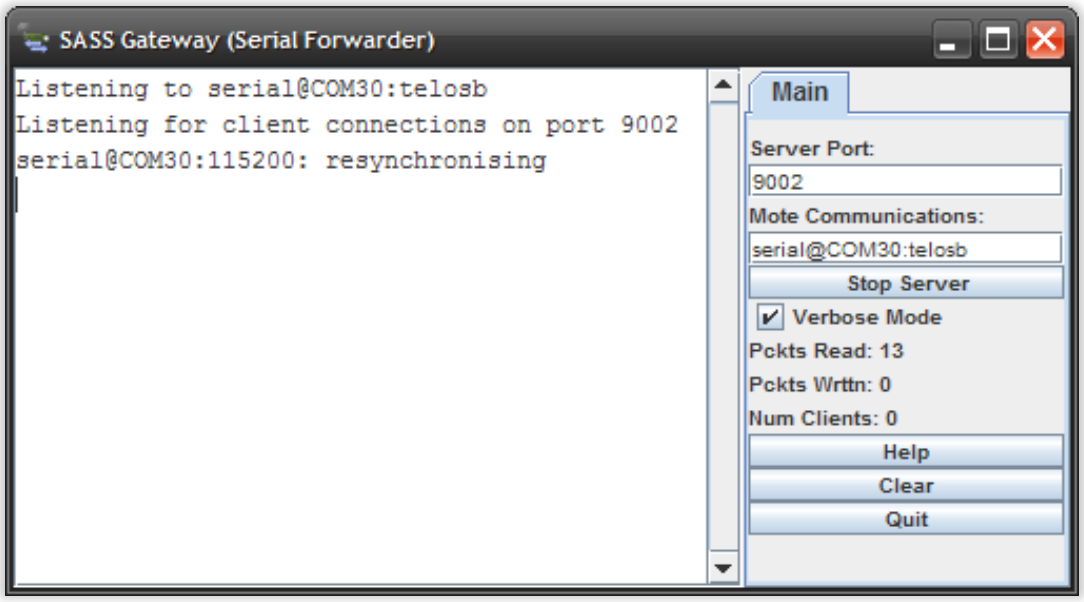
Dette er en TelosB-node med egen programvare som videreformidler meldinger som kommer fra PC (serieport) ut på radio, og motsatt vei fra radio til PC. Normalt er sinknoden koblet til en serie-port på samme PC som kjører Controller-applikasjonen.

Controller-applikasjonen har ingen kjennskap til ruting; den forstår seg kun på rene meldinger. Det er derfor opp til sinknoden å «oversette» meldinger fra Controller til CTP-meldinger som sendes ut på radionettverket. Likeledes må innkommende meldinger fra nettet strippes for Dissemination-informasjon før de viderformidles til Controller.

4.3.6.2 SerialForwarder

Dette er en valgfri komponent som gjør at sinknoden kan være koblet til en annen PC enn den som kjører Controller-applikasjonen. Controller kan kommunisere med SerialForwarder via TCP/IP, som igjen kommuniserer med sinknoden over serieport.

SerialForwarder-applikasjonen, som vist i Figur 4.9, er opprinnelig utviklet av UC Berkeley, og er en del av Java-biblioteket som følger med TinyOS-rammeverket. Det er blitt gjort noen små tilpasninger, men koden er i hovedsak original.



Figur 4.9 Skjermbilde av SerialForwarder-applikasjonen

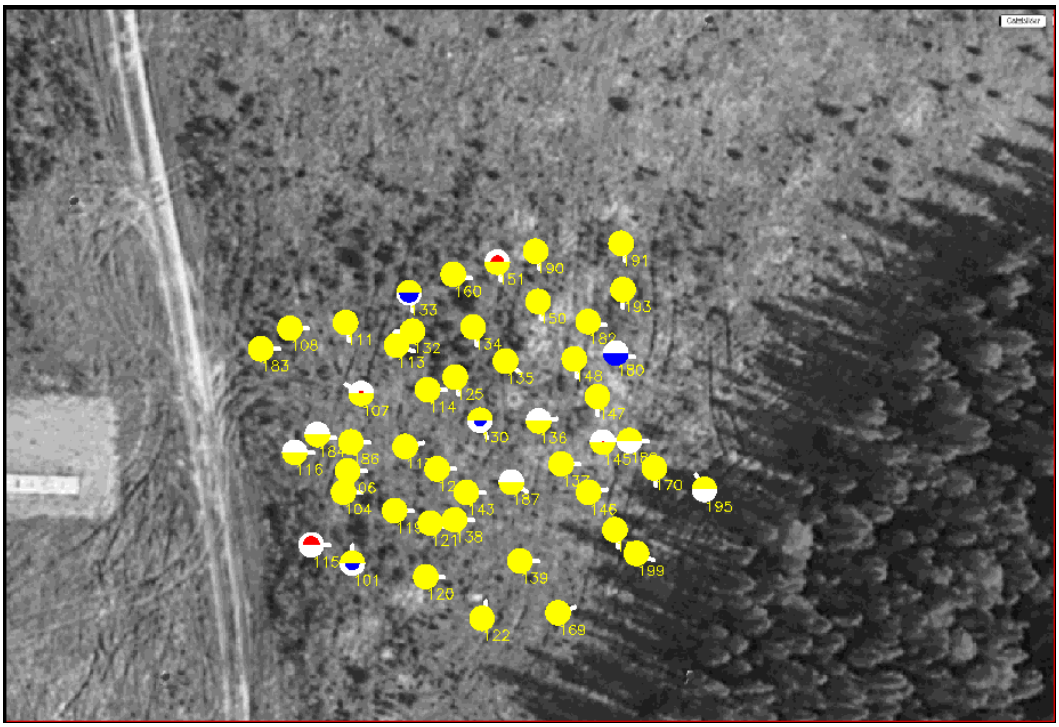
4.3.7 Kartapplikasjon

Controller-applikasjonen lister alle sensornodene fortløpende, som regel etter id-nummer. Når man har mange noder utplassert er det vanskelig å holde rede på hvilke noder som er plassert hvor, og det ble derfor utviklet en applikasjon som kan plotte alle nodene i et kart/flyfoto, samt gi visuelle indikatorer for noder som gir alarm. Figur 4.10 viser et skjermbilde fra kartapplikasjonen under feltforsøk på Rena. Kartapplikasjonen kommuniserer med Controller-applikasjonen via TCP/IP, og får informasjon derfra om hvilke noder som er utplassert, hvilke sensorer som er aktive på noden, samt GPS-posisjon. Det ble etter hvert klart for oss at det ville være praktisk å også få tegnet inn hvilken vei hver node står, men siden nodene ikke har kompass må denne informasjonen legges inn manuelt.

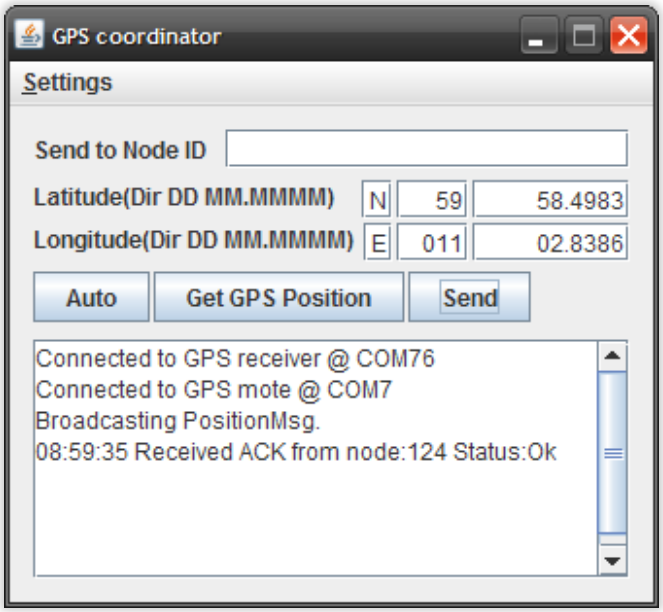
Når systemet kjører vil Controller sende melding til kartapplikasjonen for hver alarm som mottas slik at alarmen kan vises på kartet.

I utplasseringsfasen kan kartapplikasjonen brukes til å korrigere unøyaktige GPS-posisjoner ved at operatøren bruker musen til å flytte nodeindikatoren til noenlunde korrekt plass. Det blir da sendt en melding tilbake til Controller med ny (kalkulert) GPS-posisjon slik at denne blir oppdatert og lagret.

Kartet (eller flyfotoet) må hentes inn manuelt på forhånd. Det må også georefereres ved at man oppgir fire kjente GPS-posisjoner som ligger innenfor kartets grenser, helst så nær ytterkantene som mulig.



Figur 4.10 Skjermbilde av kartapplikasjonen fra feltforsøk på Rena. Sensorer med blå farge indikerer radaralarm; rødt betyr IR-alarm. Mikrofonen ble ikke brukt i dette forsøket. Hver node har en strek som viser hvilken vei sensorene ser.



Figur 4.11 Skjermbilde av GPS-applikasjonen

4.3.8 GPS-posisjonering

Nodene har ikke innebygd GPS-mottaker, og selv om det er teknisk mulig å kjøpe inn tilstrekkelig mange mottakere for å koble på nodene ville dette vært dyrt, energikrevende og upraktisk. I stedet valgte vi å ha en egen «GPS-PC» som ble brukt under utplassering av nodene. Denne PC-en hadde en tilkoblet GPS samt en egenutviklet applikasjon (Figur 4.11) som kunne overføre posisjonsinformasjon til nodene via en tilkoblet node med tilpasset programvare.

Selv om GPS-applikasjonen *kan* sende posisjonsinformasjonen til *en* spesifikt adressert node, vil den ved normal bruk kringkaste informasjonen slik at alle noder innen radiorekkevidde vil oppfatte den. Dette er for å gjøre posisjoneringsjobben enklest mulig; man setter applikasjonen i «auto»-modus slik at den kringkaster posisjonen med jevne mellomrom (vi brukte fem sekunds intervaller). For at dette skal fungere er det to kriterier som må være oppfylt:

1. En node som allerede har mottatt posisjonsinformasjon må ignorere eventuelle senere posisjonsmeldinger. Unntaket er hvis posisjonsmeldingen er spesifikt adressert til den noden, slik at vi har en mulighet til å reposisjonere manuelt.
2. Nodene må slås på en etter en, og posisjonering må utføres for hver enkelt node *før* den neste slås på.

Når nodene har mottatt en posisjon blir den sendt til Controller-applikasjonen så snart som mulig slik for lagring i sentral database. Posisjonsinformasjonen brukes blant annet av kartapplikasjon for visualisering av sensornodenes lokasjoner.

4.4 Resultater fra forsøk

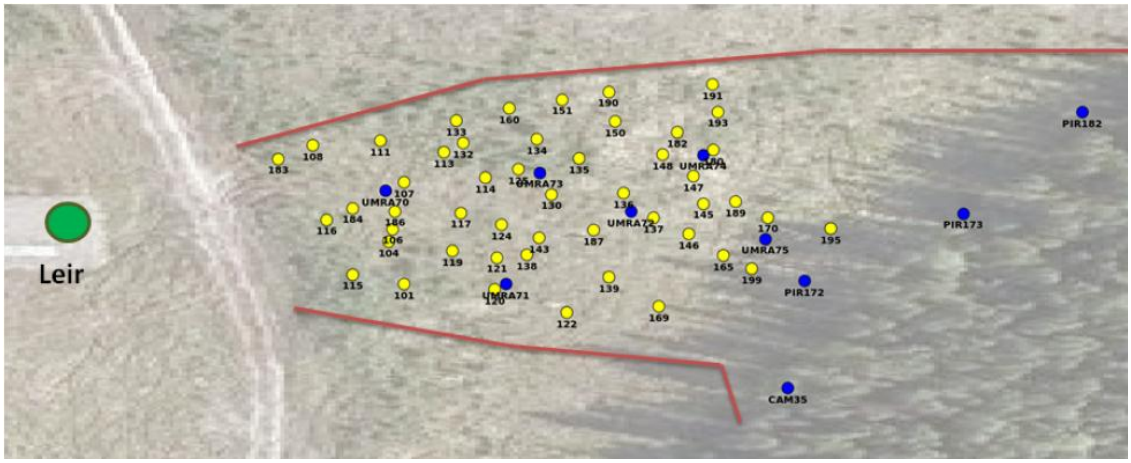
4.4.1 Perimetersikringscenarioet

Gjennomføring av perimetersikring scenarioet ble utført fra 3.-6. september 2012 ved BT-banen på Rena leir. Det utvalgte området for eksperimentet består av skog og kratt med tett vegetasjon, fordypninger og grøfter. Dette danner et utgangspunkt der det kan være vanskelig å detektere potensielle fiendtlig inntrengere til fots med tradisjonelt vakthold, eller at de kan komme så nær leiren at det utgjør en trussel i forhold til sikkerheten.

Leiren besto av to telt som skulle sikres både med tradisjonelt vakthold og sensorer. Figur 4.12 viser en oversikt over leiren og sensorfeltet som består av 50 SASS-sensorer (gule) og ti sensorer (blå) fra det kommersielle Flexnet sensorsystemet.

Det ble gjort totalt fire forsøk, to på dagtid og to på natten som vist i Tabell 4.3. I hvert forsøk utgjorde en gruppe på 4 soldater vaktstyrken ved leiren, hvorav det til enhver tid var minst to soldater utplassert ved vaktpostene. I tillegg var det en tilsvarende gruppe på fire soldater som spilte rollen som inntrengere. Oppgaven til inntrengerne var å oppklare aktiviteten i leiren uten å bli oppdaget, mens oppgaven til vaktstyrken var å identifisere fiendtlig aktivitet i leirens perimeter så tidlig som mulig.

SASS er konfigurert slik at både PIR og radar må ha vært aktive på samme sensornode innenfor et tidsintervall på 5 s for at det skal registreres som reell alarm. Likevel ble det under forsøk A og B registrert så mange falske alarmer at sikker deteksjon av inntrengere ble ytterst vanskelig. De falske alarmene skyldes i all hovedsak vind. Under forsøk A og B ble det registrert en gjennomsnittlig vindstyrke på hhv. 3,3 og 8,1 m/s på målestasjon Rena flyplass. Maks vindstyrke var trolig en del høyere enn dette. Ved kraftig vind var det mye bevegelse i vegetasjonen rundt



Figur 4.12 Oversiktsbilde over leiren og sensorfeltet. Gule og blåe sirkler representerer henholdsvis SASS- og Flexnet-sensorer. Røde streker markerer avgrensningen til sensorfeltet.

Forsøk	Dag	Starttid	Forhold	Vaktstyrke	Inntrengere
A	05.09	10:00	Dag	Gruppe 1	Gruppe 2
B	05.09	14:00	Dag	Gruppe 2	Gruppe 1
C	05.09	22:00	Natt	Gruppe 3	Gruppe 4
D	05.09	01:00	Natt	Gruppe 4	Gruppe 3

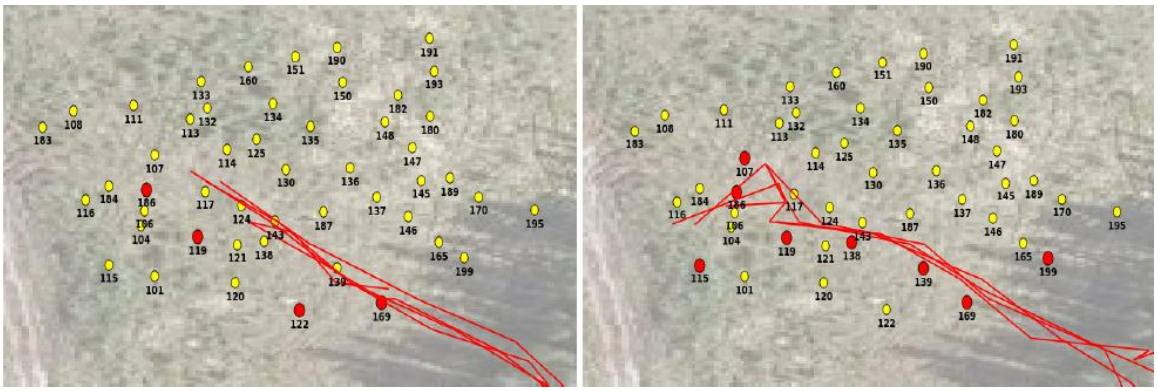
Tabell 4.3 Oversikt over forsøkene av perimetersikring scenarioet.

sensorenes deteksjonsradius, hvilket skapte problemer for både PIR og radar. Selv med finjustering av sensorparametrene var ikke resultatene fra forsøk B entydige. Under forsøk A var det mulig for operatøren å bruke alarmene til å følge inntrengerne på den visuelle kartapplikasjonen, men også her var andelen falske alarmer for høy til å gi entydige resultater.

Under forsøk C og D, var værforholdene bedre, og resultatet ble tilsvarende forbedret. Figur 4.13 viser de sensorer som ga alarm (røde sirkler) relatert til inntrengernes bevegelser (røde linjer). Disse forsøkene ga et godt innblikk i potensialet til et slikt system. Det var mulig å følge inntrengernes bevegelser gjennom sensornettverket med høy nøyaktighet. Det var stor sikkerhet i målingene og minimalt med falske alarmer.

Tabell 4.4 oppsummerer det operative resultatet til forsøkene. Tiden for inntrenging er tidspunktet da første soldat i inntrengergruppen kom inn i østlige perimeter. Siden Flexnet-sensorene var

plassert lengre øst i sensorsektoren sammenlignet med SASS-sensorer, var det som forventet at deteksjon av inntrengere ble oppdaget tidligere med Flexnet-systemet enn med SASS-systemet. Generelt viser resultatene at begge sensorsystemene var i stand til å detektere inntrengere vesentlig tidligere enn hva vaktsoldatene klarte. Det er verdt å merke seg at eksperimentets natur gjorde at vaktsoldatene tildels var klar over tidspunktet inntrengerne ville ankomme området. Videre var vaktperiodene såpass korte at vaktsoldatene ikke ble slitne eller trette i noen særlig grad. Det er derfor naturlig å anta at forskjellene mellom vaktstyrken og sensorsystemene når det gjelder deteksjonstid og presisjon vil være langt større operativt.



Figur 4.13 Oversikt over inntrengernes bevegelser og alarmer for forsøk C og D.

Forsøk	Inntrenging	Flexnet deteksjon	SASS deteksjon	Vaktsoldaters deteksjon
A	10:21	10:22	10:22/usikker	10:31
B	14:12	14:12	Usikker	14:25
C	22:05	22:10	22:22	Ikke detektert
D	01:07	01:07	01:12	01:29

Tabell 4.4 Sammenlikning av deteksjonstidspunkt. Totalt antall alarmgivere som ble aktivert av inntrengere er gitt i parentes.

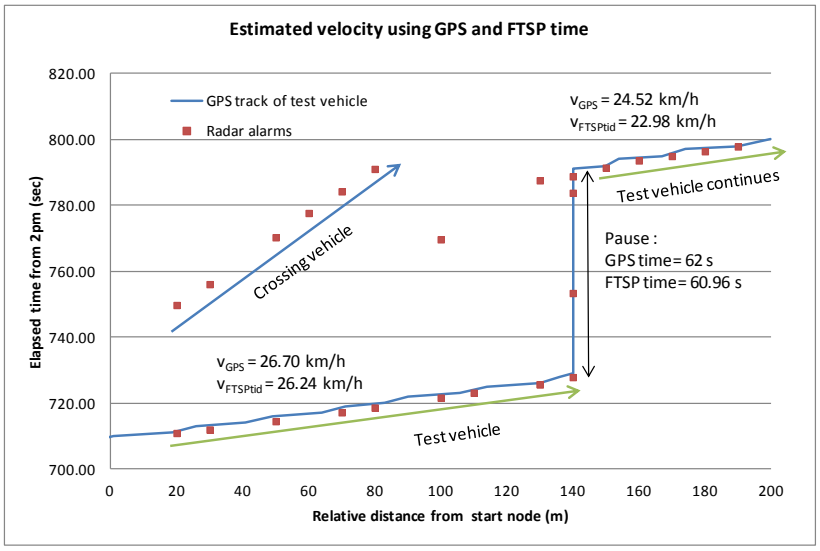
4.4.2 Veiaksescenarioet

Forsøk med veiaksescenarioet ble utført på Riisveien på Kjeller i september 2012. En veistrekning på 190 m ble overvåket med et SASS sensornett bestående av 40 sensorer, 20 på hver side av veien med 10 m innbyrdes avstand. Sensorene på den ene siden av veien ble aktivert med PIR og Radar, mens de 20 sensorene på den andre siden ble aktivert med akustisk sensor (mikrofon). Det ble gjort en rekke forsøk med passering av testkjøretøy både med og uten stopp.

Et viktig moment i dette forsøket er behovet for tidssynkronisering mellom noder i nettverket. Dette er avgjørende med tanke på korrekt gjengivelse av hendelsesforløpet og estimat av hastighet, og varighet på eventuelle stopp underveis. I motsetning til perimetersikringscenarioet, er tidssynkronisering mer kritisk i veiaksescenarioet, siden målet som spores har en vesentlig høyere hastighet. Av denne grunn har vi i dette forsøket brukt FTSP-protokollen som sørger for at tidssynkronisering ivaretas i nettverket.

Figur 4.14 viser et eksempel på kjøretøypassering med stopp. I dette forsøket kjørte vi inn i det overvåkede området og stoppet ved avstanden 140 m i ca. 1 minutt, før vi kjørte videre. Figuren viser GPS-sporet til testkjøretøyet samt radarutslag langs x-aksen og tid på y-aksen. Det er mulig å spore et objekt gjennom nettverket med radar og også i stor grad med mikrofon og PIR (ikke vist i figuren). Feilmålinger som ikke er korrelert med målinger fra andre sensornoder er enkle å filtrere bort. Spesielt for radar ga dette veldig gode resultater der falskalarmraten er liten. For mikrofon er andelen falske alarmer noe høyere, siden den er mer sensitiv for faktorer som vind og andre støykilder. PIR fungerte også bra til kjøretøydeteksjon, men siden variasjoner i solforhold og skydekke påvirker sensoren, kan flere sensornoder gi falske alarmer samtidig.

Videre ser vi at selv om et annet kjøretøy var i området samtidig, er det mulig å skille ut denne hendelsen. Figuren viser også at det er mulig å estimere gjennomsnittshastigheten og varigheten på oppholdet med rimelig god nøyaktighet. Estimatene er basert på alarmenes tidsstempel og ved



Figur 4.14 Eksempel på deteksjon av passerende kjøretøy og estimat av hastighet og oppholdets varighet.

bruk av lineær regresjon. Avviket mellom estimatene og kalkulert hastighet basert på GPS-data er relativt lite, hvilket indikerer at tidssynkroniseringen fungerer godt.

5 Konklusjoner og oppsummering

Hensikten med FFI-prosjekt 1141 har vært å undersøke hvorvidt teknologiområdet «trådløst sensornettverk» er modent nok til å kunne benyttes for militære anvendelser. Teknologiområdet betraktes i akademiske kretser å være relativt ungt og kan sies å ha sitt hovedutspring i forskningen rundt «Smart Dust»[4], MEMS og trådløs teknologi på slutten av 90-tallet. Det siste tiåret er det blitt gjennomført en formidabel forskningsinnsats innen fagfeltene elektronikk, nettverk og sensorer for å realisere 90-tallets visjon om allestedsnærværende trådløse sensornoder. Sett fra et militært ståsted er denne ideen imidlertid ikke ny og allerede under Vietnam krigen deployerte US Army mer enn 1000 trådløse sensorer for oppklaring og overvåkning i Laos. Disse første systemene var kostbare, store og ga unøyaktige resultater. Men det er også disse tre områdene, nemlig pris, fysisk størrelse og presisjon som hovedsakelig er forbedret gjennom de siste årenes forskning. Konseptideen, det å kunne overvåke et område med høy presisjon og begrenset personellinnsats er den samme.

Forskningen de siste 10-15 årene har i stor grad vært utført av informatikere og nettverksforskere, som har tilstrebet generiske løsninger. De generiske løsningene har utvilsomt åpnet for et bredere spekter av mulige bruksområder for trådløse sensornettverk. Man ser nå for seg å benytte trådløse sensornettverk innen en rekke ikke-militære områder, slik som miljøovervåkning, strukturmålinger, bilindustrien, landbruk, geologi og mye mer. Det er åpenbart at alle disse ulike variantene av trådløse sensornettverk ikke har identiske krav til datakapasitet, rekkevidder og sensormoduler, og heller ikke kan benytte de samme protokollene og algoritmene. Trådløst sensornettverk er i høyeste grad et anvendt og applikasjonsrettet forskningsfelt, der løsningene må drives frem «bottom-up» med en klar formening om applikasjonsdomenets særegne krav. En generisk tilnærming, hvor man forsøker å løse et kjerneproblem uavhengig av sluttbruker eller applikasjon, var avgjørende ved for eksempel utviklingen av Internett. Dette muliggjør for at de samme protokollene benyttes over alt, uavhengig av de overliggende applikasjonene og den fysiske infrastrukturen. Innen trådløse sensornettverk ser vi derimot en dreining bort fra denne tankegangen. Dette gjenspeiles ved at et økende antall publikasjoner er mer applikasjonsrettede og presenterer dedikerte løsninger på krevende fysiske problemer.

Vi mener at sensornettverkkonseptet har en rekke anvendelser innen det militære domenet. I våre arbeider har vi undersøkt scenarioer innen arealovervåkning, perimetersikring og akseovervåkning. Med dagens teknologi er det er fullt mulig å konstruere en fleksibel plattform, som dekker flere ulike operative behov. Likevel mener vi at dedikerte systemer gir de beste løsningene, innen både deteksjonsnøyaktighet, brukervennlighet, pris og systemlevetid. I oppklaring og akseovervåkning, vil det for eksempel være ønskelig med miniatyriserte systemer for engangsbruk, som verken vedlikeholdes, eller samles inn etter endt oppdrag. Sensornettverk for perimetersikring vil ha en annen karakter, der sensornodene gjerne kan være noe større og kan ha mer kapasitet, avhengig av leirens størrelse og varighet. For objektsikring behøves gjerne færre, men mer avanserte sensorer, men til gjengjeld kan de utstyres med bedre batterikapasitet. Med god systemkunnskap og operativ forståelse, mener vi at teknologien er moden nok til at det lar seg gjøre å konstruere trådløse sensorsystemer i dag, som gir vesentlig forbedring og

[4] Smart Dust var et forskningsprogram hos DARPA som startet i 1997

nøyaktighet, for overvåknings og etterretningsformål. De områdene som krever spesiell oppmerksomhet ved innføring av trådløse sensornettverk er: radiorekkevidde, sikkerhet, pålitelighet og brukervennlighet:

- God radiorekkevidde gir større fleksibilitet ved deployering av sensornettverket og mer robust transmisjon. De miniatyriserte nodenes natur, og radiokretsenes høye modulasjonsfrekvens, samt begrensede batterikapasitet, fører til at radiorekkevidden er naturlig begrenset. I utvikling av sivile sensornettverksplattformer har lang radiorekkevidde ikke vært spesielt prioritert. Videre forskning på FFI bør derfor undersøke tilgjengelige teknologier, slik at de operative kravene til rekkevidde blir dekket. Men her er det verdt å merke seg at kort radiorekkevidde kan kompenseres med å deployere et høyere antall noder. Dette har sine fordeler ved at systemet har høyere grad av redundans og dermed bedre robusthet, samt høyere oppløsning med tanke på målfølging.

- Sikring av sensornoder, både fysisk og på protokollnivå, er avgjørende for å oppnå et velfungerende og robust system. På fysisk nivå er det stor potensial for forbedring av kamuflasje og ytterligere miniatyrisering. Dette vil gjøre systemet mindre synlig og vanskeligere å detektere. Risikoen for nøytralisering blir dermed også mindre. I tillegg er det trådløse medium i sin natur ikke avlytningssikkert. Sensornodene må derfor sikres også på protokollnivå slik at fienden ikke får tilgang på sensitiv informasjon. FFI har innen dette feltet bred kunnskap, som vil kunne benyttes i utvikling og i kravstilling til innkjøp av ferdige systemer.

- Det er nødvendig å begrense antall falske alarmer for å sikre et pålitelig og anvendelig system. I SASS sensorsystemet har vi stort sett benyttet billige hyllevaresensorer. Resultater fra eksperimenter viser at disse sensorene er relativt følsomme for påvirkning fra omgivelsene, som sol og vind. Her er det et stort potensiale for å anvende alternative og mer pålitelige sensorer. Videre er det mulig å forbedre deteksjons- og filtrerings-algoritmene, og lokal utveksling og prosessering av sensordata (datafusjon). Dette kan redusere antall falske alarmer og dermed muliggjøre økt sensitivitet.

- Et trådløst sensornettverk bør ha enkle grensesnitt slik at operatørbyrden blir lavest mulig. Sensorsystemet må også ses i sammenheng med andre sensorsystemer og aktuatorer. Det vil være aktuelt å knytte sensorsystemet opp mot mobile overvåkningssensorer, eksterne kartsystem som BMS, kamerasystemer og mer avanserte sensorplattformer. Tilfredsstillende brukervennlighet kan kun oppnås dersom alle systemene som skal benyttes i en gitt operativ setting ses i sammenheng. Det er ingen kommersielle sensorprodusenter som i dag vil kunne fylle rollen som totalintegrator, og det er helt nødvendig at FFI tar en slik rolle.

Selv om utviklingen innen dette feltet stadig foregår i høy hastighet, mener vi at teknologien er moden nok til at den kan benyttes med hell innen flere militære anvendelser. For perimetersikring og objektsikring, vil man allerede med dagens teknologi kunne produsere systemer som er enkle og raske å deployere, og som gir meget god sikring. For oppklaring, er begrensningen i første rekke kostnad (siden sensorene bør være av typen bruk-og-kast), sikkerhet og rekkevidde.

Referanser

[1] E.Svinsås, L.Hanssen, V.Arneson, E.Larsen, V.Pham, J.Flathagen, P.G.Dalsjø, J.Gakkestad, and R.Korsnes, "Situational Awareness Sensor Systems (SASS) - grunnleggende scenarier og krav," FFI-notat 2009/01905, (Unntatt offentlighet), 2009.

[2] V.Arneson, "Propagasjon i trådløse sensornett,"FFI-rapport 2012/00820, 2012.

[3] C.Intanagonwiwat, R.Govindan, and D.Estrin, "Directed diffusion: a scalable and robust communication paradigm for sensor networks," in *Proceedings of the 6th annual international conference on Mobile computing and networking, MobiCom '00* Boston, Massachusetts, USA: ACM, 2010, pp. 56-67.

[4] E.Larsen, J.Flathagen, V.Pham, and L.Landmark, "Adapting OLSR for WSNs (iOLSR) Using Locally Increasing Intervals," *Sensors & Transducers journal (ISSN 1726-5479)*, vol. 14-2, Special Issue, pp. 254-268, Mar.2012.

[5] E.Larsen, J.Flathagen, V.Pham, and L.Landmark, "iOLSR: OLSR for WSNs using Dynamically Adaptive Intervals," in *proceedings of SENSORCOMM 2011, The Fifth International Conference on Sensor Technologies and Applications.* Xpert Publishing Services 2011 ISBN 978-1-61208-144-1, 2011, pp. 18-23.

[6] J.Flathagen, Ø.Kure, and P.E.Engelstad, "Constrained-based Multiple Sink Placement for Wireless Sensor Networks," in *Mobile Adhoc and Sensor Systems (MASS), 2011 IEEE 8th International Conference on* 2011, pp. 783-788.

[7] O.Gnawali, R.Fonseca, K.Jamieson, D.Moss, and P.Levis, "Collection tree protocol," *SenSys '09*, pp. 1-14, 2009.

[8] S.Biswas and R.Morris, "Opportunistic routing in multi-hop wireless networks," *ACM SIGCOMM'04*, vol. 34, pp. 69-74, 2004.

[9] J.Flathagen, P.E.Engelstad, and Ø.Kure, "O-CTP: Hybrid Opportunistic Collection Tree Protocol forWireless Sensor Networks," in *IEEE Conference on Local Computer Networks* LCN, 2012, pp. 947-955.

[10] J.Flathagen and R.Korsnes, "Localization in wireless sensor networks based on Ad hoc routing and evolutionary computation," in *Military Communications Conference - MILCOM 2010, Oct. 31-Nov. 3 2010* 2010, pp. 1062-1067.

[11] R.Korsnes, "Distributed data fusion in sensor networks,"FFI-rapport 2013/00116, 2013.

[12] S.C.Tornay, *Ockham: Studies and Selections*. La Salle, IL: Open Court Publishers, 1938.

[13] L.A.Zadeh, "Fuzzy Logic, Neural Netwokrs, and Soft Computing," *Communication of the ACM*, vol. 37, pp. 77-84, Mar.1994.

[14] L.Hanssen and J.Gakkestad, "Solar Cell Size Requirement for Powering of Wireless Sensor Network Used in Northern Europe," in *Proceedings of the International Workshops on PowerMEMS*, 2010, pp. 17-20.

[15] J.Gakkestad and L.Hanssen, "Powering Wireless Sensor Networks Nodes in Northern Europe Using Solar Cell Panel for Energy Harvesting," in *New Technologies, Mobility and Security (NTMS), 4th IFIP International Conference on, 7-10 Feb. 2011* 2011.

[16] TinyOS, http://www.tinyos.net

[17] TinyOS Code, http://code.google.com/p/tinyos-main

[18] Network embedded systems C, http://nescc.sourceforge.net

[19] V.Pham and E.Larsen, "FTSP tidssynkronisering for trådløs sensornettverk: Evaluering og tilpasning,"FFI-rapport 2012/02278, 2012.

Akronymer

A/D	Analog/Digital
BCAST	Broadcast Routing Protocol
BMS	Battlefield Management System
CD&E	Concept Development and Experimentation
CHURN	Changes of Routes
CPU	Central Processing Unit
CTP	Collection Tree Protocol
DARPA	The Defence Advanced Research Projects Agency
DYMO	Dynamic MANET On-demand
FOST/FSA	Forsvarets sikkerhetstjeneste/Forsvarets sikkerhetsavdeling
FSK	Forsvarets spesialkommando
FTSP	Flooding Time Synchronization Protocol
GEOPPS	Geographical Opportunistic Routing for Vehicular Networks
GPS	Global Positioning System
HV	Heimevernet
IED	Improvised Explosive Device
IR	Infrared
ISM	Instrumentation, Scientific and Medication
LPI	Low Probability of Intercept
LQI	Link Quality Indicator
MAC	Medium Access Control
MEMS	Micro Electro Mechanical Systems
O-CTP	Opportunistic Collection Tree Protocol
OLSR	Optimized Link State Routing Protocol
PDR	Packet Delivery Rate
PIR	Passive Infrared
PVC	Poly Vinyl Chloride
PVGIS	Solar Radiation Database for Europe
ROM	Read Only Memory
SASS	Situational Awareness Sensor System
SDK	Software Development Kit
TCP/IP	Transmission Control Protocol/Internet Protocol
TYMO	Dymo implementation on TinyOS
UGS	Unattended Ground Sensors
USB	Universal Serial Bus
WSN	Wireless Sensor Networks