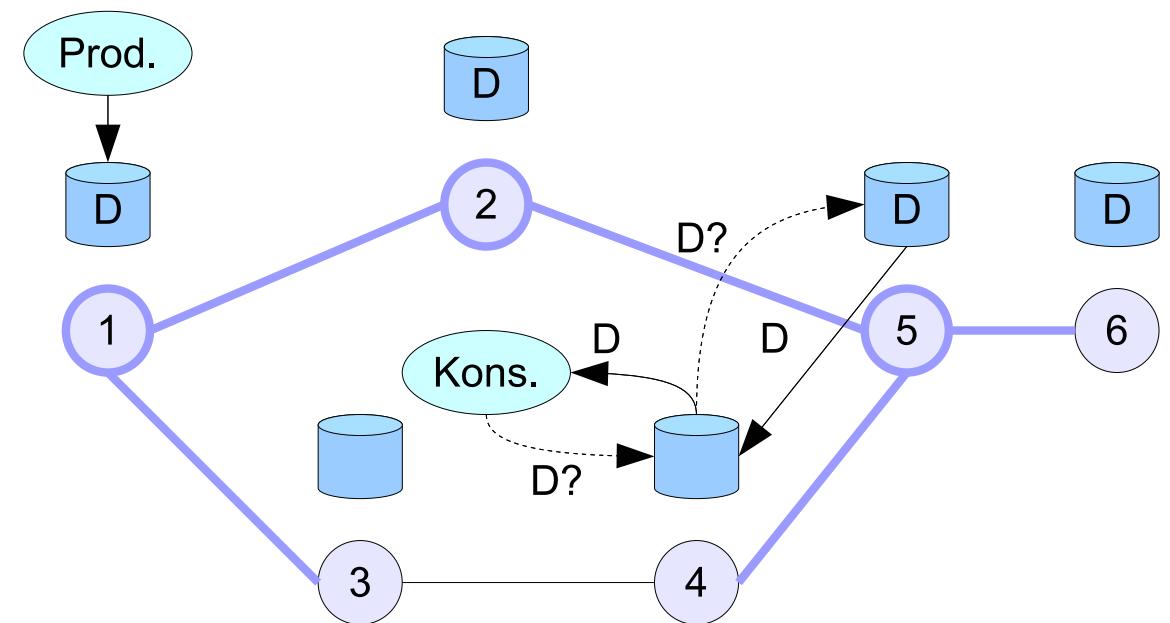


ANDERS HANSSON, ERIKA JOHANSSON,
JAN NILSSON, ULF STERNER OCH ÅSA WAERN



FOI är en huvudsakligen uppdragsfinansierad myndighet under Försvarsdepartementet. Kärnverksamheten är forskning, metod- och teknikutveckling till nytta för försvar och säkerhet. Organisationen har cirka 1000 anställda varav ungefär 800 är forskare. Detta gör organisationen till Sveriges största forskningsinstitut. FOI ger kunderna tillgång till ledande expertis inom ett stort antal tillämpningsområden såsom säkerhetspolitiska studier och analyser inom försvar och säkerhet, bedömning av olika typer av hot, system för ledning och hantering av kriser, skydd mot och hantering av farliga ämnen, IT-säkerhet och nya sensorers möjligheter.

Anders Hansson, Erika Johansson, Jan Nilsson,
Ulf Sterner och Åsa Waern

Distribuerade databaser i mobila trådlösa nät, del 2

Scenariobaserade designförslag

Titel	Distribuerade databaser i mobila trådlösa nät, del 2 Scenariobaserade designförslag
Title	Distributed databases for mobile wireless networks, part 2 Scenario based design pro- posals
Rapportnr / Report No.	FOI-R--2933--SE
Rapporttyp	Teknisk rapport
Report Type	Technical Report
Månad / Month	December / December
Utgivningsår / Year	2009
Antal sidor / Pages	38
ISSN	1650-1942
Kund / Customer	FM
Projektnr / Project No.	E53065
Godkänd av / Approved by	Anders Törne
FOI, Totalförsvarets Forskningsinstitut	FOI, Swedish Defence Research Agency
Avdelningen för Informationssystem	Information Systems
Box 1165	P.O. Box 1165
581 11 LINKÖPING	SE-581 11 LINKÖPING

Sammanfattning

Många av de tjänster som är väsentliga i ett taktiskt mobilt nät kan dra fördel av en distribuerad databasstruktur som effektivt tillgängliggör data. Genom införandet av en sådan struktur kommer kommunikationsresurserna att kunna utnyttjas effektivare och fler användare kommer att få sitt informationsbehov tillgodosett. Vi beskriver här tre tjänster som kan dra nytta av en distribuerad databas; en stridsledningstjänst, en soldatstatustjänst och en videodistributions-tjänst. I rapporten presenteras även två möjliga databasdesigner. En baserad på mellanlagring av tidigare efterfrågad data och en baserad på att data aktivt sprids till noder där den ofta är efterfrågad. Vi beskriver även två referensmetoder, en som bygger på att data alltid hämtas hos producenten och en som bygger på att all data sprids till alla. En övergripande teoretisk analys av de beskrivna metodernas lämplighet för de utvalda tjänsterna visar att det är svårt att finna en databaslösning som lämpar sig för alla tjänsterna. En lämplig designfilosofi är därför att använda en hybridalgorithm som utnyttjar olika metoder för olika tjänster.

Nyckelord: distribuerade databaser, databasstrukturer, dataspridning, trådlösa nät, stridsledningstjänst, soldatstatus, videodistribution, mellanlagring, aktiv spridning

Abstract

Many of the services presumed to be vital in a mobile, tactical network can take advantage of a distributed database system with effective data dissemination. The use of a database system will improve the utilization of the communications resources and enable meeting more users' information demands. We will here describe three services that would benefit from a distributed database system; a command and control service, a service for soldier status information, and a service for video distribution. In this report we present two possible solutions for distributed databases in mobile, wireless ad-hoc networks. One design utilizes caching of requested data along the transmission path and the other pre-emptively disseminates data to the units where it's in highest demand. Furthermore, we describe two reference designs; one where no data is transmitted unless a specific request from a unit has occurred and one with total data dissemination, i.e. all data is relayed to all units. Through analyzing these designs for the chosen services, we determine that it is difficult to find a distributed database design that is suitable for all services. The use of a hybrid solution that uses different designs for different services hence appears to be best choice.

Keywords: distributed database, database structure, data dissemination, wireless networks, ad-hoc networks, command and control service, soldier status information, video distribution, caching, replication

Innehållsförteckning

1	Inledning	7
2	Behov och tjänster	9
2.1	Scenario SLB-tjänster	10
2.1.1	Tjänstens struktur	10
2.2	Scenario soldatstatustjänster	11
2.2.1	Tjänstens struktur	14
2.3	Scenario videodistributionstjänster	14
2.3.1	Tjänstens struktur	15
2.4	Sammanfattning av tjänsternas egenskaper	16
3	Befintlig Teknik	19
3.1	Forskningsfronten	19
3.1.1	Replikering av data	19
3.1.2	Mellanlagring av data	20
3.2	Referensmetoder	22
3.2.1	Fullständig spridning	22
3.2.2	Producentförfrågning	22
3.3	Multicast-träd	24
3.3.1	MPR baserade multicast-träd	24
3.3.2	Multicast-träd och variabel datatakt	26
4	Ny Teknik	29
4.1	Reaktiv Mellanlagring	29
4.1.1	Algoritm	30
4.1.2	Lämplighet - designfördelar	30
4.2	Aktiv spridning	31
4.2.1	Algoritm	32
4.2.2	Lämplighet - designfördelar	33
4.3	Konsistens och aktualitet hos data	33
4.3.1	Synkronisering mot producent	33
4.3.2	Synkronisering mot granne	34
4.4	Hybridalgoritm	35
5	Slutsatser	37
	Referenser	37

FOI-R--2933--SE

1 Inledning

Detta arbete har utförts inom projektet *Distribuerade databaser för trådlösa mobila nät*, som finansierats av FOI:s *Strategiska forskningskärnor*. Målet med projektet är att ta fram lösningar för distribuerade databaser som fungerar i de mobila trådlösa nätverk som erbjuder taktisk kommunikation för rörliga förband. Den främsta nyttan för Försvarsmakten är att FOI bygger kompetens så att FOI kan stödja Försvarsmakten med kompetens vid kravställning av kommande ledningssystem.

Många av de tjänster som är väsentliga i ett taktiskt mobilt nät innebär att data genereras, lagras och hämtas i nätet under ett uppdrag. Sådan information kan vara av både teknisk karaktär, till exempel katalogtjänster så som Domain Name System, (DNS) och taktisk karaktär, exempelvis ledningsinformation som gemensam lägesbild och orderdistribution. För att på ett effektivt och strukturerat sätt hantera dessa tjänsters behov av att komma åt data kan olika databaser användas.

Att konstruera en effektiv databasstruktur för de mobila trådlösa nät som erbjuder taktisk kommunikation för de rörliga förbanden innebär en utmaning. I dessa nät varierar de tillgängliga kommunikationsresurserna kraftigt och saknas ibland helt, vilket kan innebära separerade delnät. Fokus vid design av en databas bör därför vara ett lågt kommunikationsresursutnyttjande och stor flexibilitet att hantera autonoma noder. Det senare kravet gör att distribuerade lösningar utan centrala noder är att föredra.

För att kunna värdera olika distribuerade databasmetoder behövs både väldefinierade taktiska kommunikationstjänster och scenarier som beskriver hur användare utnyttjar tjänsterna. I denna rapport beskriver vi tre taktiska tjänster tillsammans med ett tänkbart scenario:

- stridsledning (SLB),
- soldatstatusdistribution,
- videodistribution.

Gemensamt för de tre tjänsterna är att genererad data inte förändras men att ny, mer aktuell, information ständigt produceras i nätet. Trafikbelastning och trafikmönster skiljer sig dock väsentligt åt mellan de tre tjänsterna. För stridsledningstjänsten beskrivs hur ledningsinformation sprids under ett patrulleringsuppdrag i en stadsmiljö. Soldatstatusstjänsten behandlar insamling av soldaternas sensorinformation. Videodistribution innebär i det här scenariot att en stadsmiljö övervakas, där video och bilder från olika sensorer distribueras till olika

användare. Samtliga dessa tjänster kräver att aktuell data ständigt kan spridas och göras tillgänglig för grupper av noder.

Det finns flera olika problemområden kopplade till distribuerade databaser. Beroende på tillämpningar och förutsättningar så kan utmaningarna se väldigt olika ut. Några exempel på frågeställningar är:

- Hur kan stora datamängder fördelas över olika instanser av databasen med tillräcklig redundans?
- Hur kan man erbjuda en hög tillgänglighet till databasen över osäkra kommunikationsförbindelser?
- Hur ska databasen hantera att nät delas i olika delnät som efter en tid åter kan gå samman till ett nät?
- Hur kan data i olika instanser uppdateras så att databasen förblir konsistent, det vill säga att en fråga ger samma svar överallt?
- Hur ska data spridas så att alla användare har tillgång till aktuell data?

I den här rapporten fokuserar vi på dataspridning, eftersom det antagligen är den största utmaningen för att hantera de tjänster som vi beskrev ovan. En stor del av forskningen om dataspridning i mobila ad hoc-nät baseras på civila system och ett vanligt antagande är att datalagringskapacitet är den mest begränsade resursen. Eftersom projektets fokus ligger på taktiska mobila ad hoc-nät med lång räckvidd, så antar vi i detta arbete att det snarare är kommunikationsresurserna som är mest begränsande för tjänsternas prestanda. Detta påverkar naturligtvis valet av tekniska lösningar för distribuerade databaser.

Vi föreslår två möjliga databaslösningar för de tjänster som har beskrivits. Den ena lösningen bygger på mellanlagring av tidigare efterfrågad data medan den andra lösningen sprider data i förväg till noder där informationen ofta efterfrågas. Vi beskriver även två referensmetoder som är lämpliga att jämföra med nya lösningar. En bygger på att data alltid hämtas hos producenten och en innebär att all data sprids till alla.

I kapitel 2 ger vi tre exempel på tjänster och ett scenario där distribuerade databaser kan effektivisera trafikhanteringen. Kapitel 3 tar upp befintlig data-basteknik som är av intresse för militära applikationer, bland annat de två referensmetoderna. I kapitel 4 beskriver vi de två databasdesigner vi föreslår. Kapitel 5 slutligen, sammanfattar studiens slutsatser.

2 Behov och tjänster

För att kunna värdera olika distribuerade databasstrukturer och databaslösningar behövs scenarier. I ett första skede är syftet att kunna göra en övergripande analys och värdering av de olika lösningsförslagen. Det som behöver beskrivas i scenarierna är vilka enheter/noder som ingår samt behövda tjänster och hur dataflödena mellan noderna ser ut. Framför allt måste scenariet definiera vilka som är konsumenter och producenter, dvs. vilka som genererar data och vilka som har behov av data. När designlösningarna och deras prestanda senare ska undersökas via simuleringar, kan scenarierna behöva specificeras ytterligare eller utvidgas. Ett exempel på utvidgning kan vara att inkludera mobilitet, dvs. nodernas rörelsemönster.

Det finns många tänkbara tjänster, men vi väljer här att fokusera på några där en distribuerad databaslösning kan förbättra användarens tillgång till olika tjänster. Vi har valt att fokusera på följande tre tjänster; stridsledning (SLB), soldatstatus- och videodistributionstjänst. I det SLB-baserade scenariet beskrivs var ledningsinformation genereras och hur den sprids under ett patrulleringsuppdrag i stadsmiljö. Soldatstatus-tjänsten behandlar insamling av soldaternas sensorinformation som sedan efterbehandlas för att bedöma soldaternas och enheternas stressnivå och uthållighet. Scenariot med videodistribution är övervakning i stadsmiljö där video och bilder från olika sensorer (och soldater i vissa fall) distribueras till dem som har behov av video/bildinformationen.

De tre tjänsterna vi valt är bara några exempel där distribuerade databaser kan vara till nytta. Det finns många andra tänkbara tjänster eller scenarier där databaser kan fylla en funktion t.ex. vid distribuering av positionsinformation, hantering av logistikinformation eller vid statistikinsamling för management av kommunikationsnät. Ytterligare exempel är sensornätjänster där information från olika sensorer ska vägas samman och analyseras.

Målet med en positionstjänst är att upprätthålla en korrekt lägesbild så man vet var egna enheter befinner sig och om möjligt också andra eller motståndarens enheter. En sådan tjänst kan implementeras på väldigt många olika sätt. Huvudfrågorna är med vilken noggrannhet man behöver veta enheternas positioner och vilka som behöver informationen. Detta styr sedan hur ofta positioner behöver skickas och till vilka. Det finns ett antal metoder för att minska antalet positionsuppdateringar, exempelvis kan vi prediktera enheternas rörelser och bara göra uppdateringar vid större avvikelser från prediktionen.

I de följande delkapitlen beskrivs de tre valda tjänsterna närmare. Varje scenariobeskrivning innehåller en vinjett, dvs. ett exempel på användning. Sedan beskrivs tjänsternas struktur innehållande trafikmängder plus en producent-konsument bild. Kapitlet avslutas genom att sammanfatta scenariobeskrivning-

arna och sammanställa tjänsternas egenskaper i en tabell.

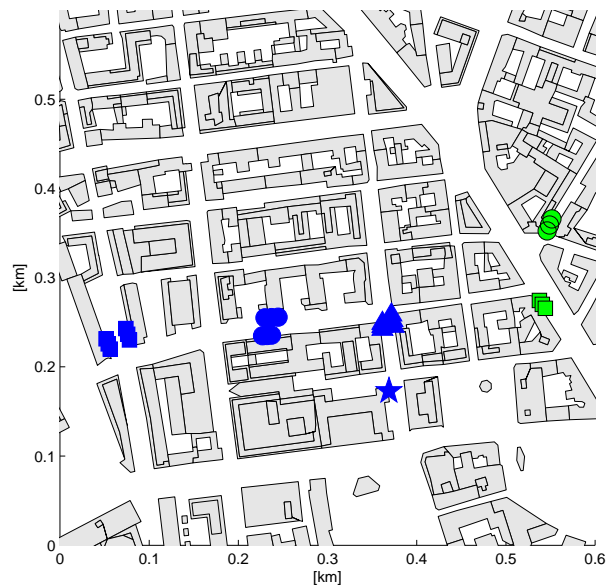
2.1 Scenario SLB-tjänster

I SLB-systemet sprids information med hjälp av olika oleat. I första hand är det som sprids planer och mer strategisk information, t.ex. vad eget förband skall uppnå, målbild, hur planerad verksamhet ska genomföras mm. Oleaten produceras företrädesvis uppifrån och ner, dvs. av bataljonschefer och kompanichefer och sprids ner till lägsta nivå utrustad med SLB-system. Oleaten kan spridas klumpvis eller i den takt de produceras och godkänns. Även om oleaten i sig är en färskvara så kan flera på varandra följande oleat skapa en historia som kan utnyttjas för att snabbt skaffa sig en lägesuppfattning och en bild över den strategi som finns, t.ex. enligt nedanstående.

En blå pluton bestående av tre fordonsutrustade grupper genomför ett patrulleringsuppdrag i en stadsmiljö, se Figur 2.1. Uppdragets strategi och mål är presenterade via ett flertal oleat som utarbetats av såväl bataljonschef som kompanichef. Eftersom hotnivån eskalerar skickas uppdaterade planer ut via den distribuerade databasen i form av nya oleat. Förutom att oleaten sprids till alla i plutonen lagras de också i den distribuerade databasen så att de är åtkomliga för nya enheter. Blå pluton sammanstötter med en uppretad folkgrupp varpå förstärkning begärs. Därefter ansluter del av grön pluton bestående av två grupper. Då de närmar sig området kopplas kommunikationsnätverken samman och grön pluton får oleaten via databasen. På så vis är de anslutande gröna grupperna väl informerade om strategin inklusive de förändringar som gjorts. Lägesbilden kan sedan kompletteras genom talkommunikation med blå plutonchef.

2.1.1 Tjänstens struktur

Oleaten skickas i första hand från bataljonschef till kompanichefen. Denne bearbetar oleaten vidare för att passa plutonen. Relevant för vårt scenario är de oleat som kompanichefen producerar till plutonen. Oleaten uppdateras med jämna mellanrum och distribueras via databasen, se Figur 2.2. Soldaterna i plutonen prenumererar på oleaten, vilket innebär att de automatiskt får de nya oleaten när de kommit till databasen. Kompanichefen befinner sig ca 2 km från insatsområdet. Oleaten antas vara jämförbara med en taktisk vy i en GIS-karta och har en storlek av max 500 kB och skickas en gång i halvtimmen till den distribuerade databasen. När den gröna plutonen ansluter sänder de en förfrågan till databasen om samtliga oleat som producerats de senaste 2 timmarna. Gör vi en uppskattning av trafiken i nätet så kommer kompanichefen skicka 2x500

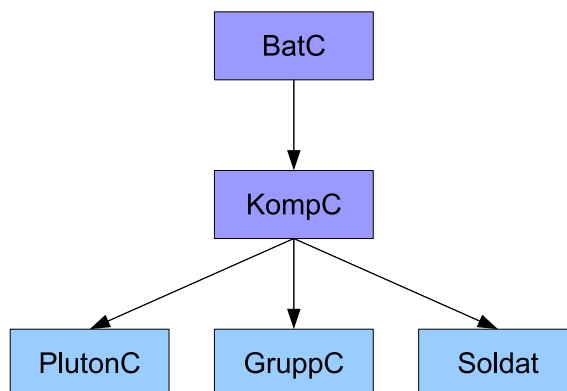


Figur 2.1: Exempel där blå pluton med tre grupper får förstärkning i form av två grupper från grön pluton. Den blå stjärnan representerar en enhet med bättre kommunikationsresurser exempelvis en ledningsvagn.

kB i timmen på ett avstånd av 2 km till databasen. Vidare kommer databasen att göra massutskick av oleaten till samtliga i plutonen, 36 soldater. I Tabell 2.1 sammanfattas trafikmängden från vårt första scenario.

2.2 Scenario soldatstatustjänster

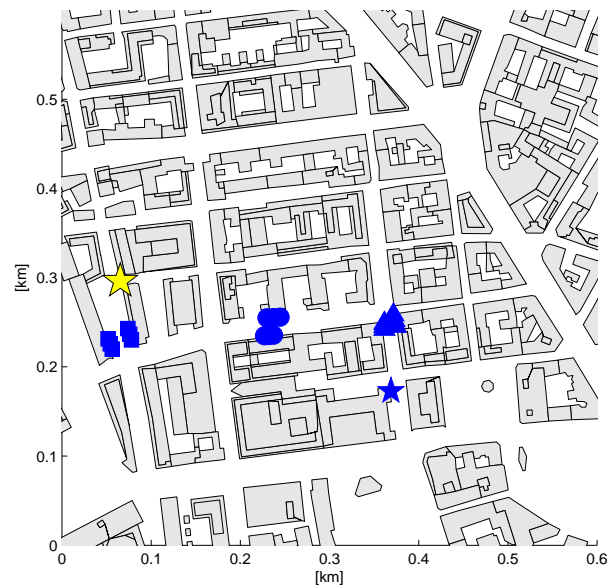
Framtida kommunikationssystem kommer att kunna tillhandahålla såväl tal- som datatrafik. I och med det så öppnas nya möjligheter där radion kan användas för att skicka bland annat sensorinformation till andra enheter. Vi kan också anta att radion har relativt gott om minne vilket gör att den kan lagra en hel del sensorinformation. Batterikraften är däremot begränsad varför förmågan att processa sensorinformationen eller hantera någon annans sensorinformation får antas vara liten. Här kan en gemensam distribuerad databas användas. Varje enskild soldats sensorinformation skulle kunna distribueras via databasen till de noder som har bättre förmåga att utföra analys av sensordata. Dessa kan sedan sammanställa all sensorinformation från gruppen och presentera en gemensam lägesbild. Denna lägesbild kan spridas till gruppen och överordnande chefer via den distribuerade databasen.



Figur 2.2: Producenter (lila) och konsumenter (ljusblå) för oleattjänsten

	Producent	Konsument	Antal	Storlek	Trafikflöde
Oleat	kompanichef	plutonen	2 per timme	500 kB	multicast, 36 och 60 noder
Oleat	databas	tillkommande pluton	1	2 MB	multicast en gång till 24 noder

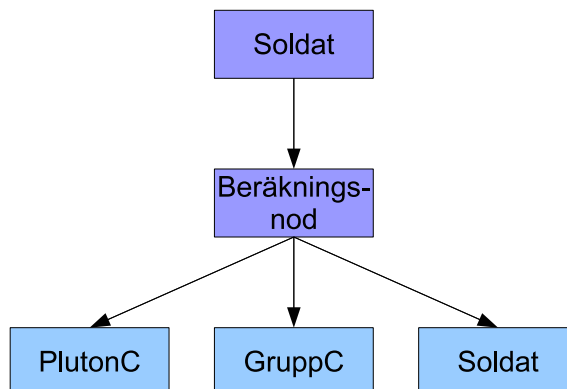
Tabell 2.1: Trafikegenskaper - storlek på data objekt och trafikflöden



Figur 2.3: Blå pluton med tre grupper patrullerar ett bebyggt område. Respektive soldats status skickas med hjälp av den distribuerade databasen till en nod med kapacitet att analysera data, markerad med blå stjärna. Statusinformationen distribueras sedan vidare till överordnande chefer.

Som exempel kan vi studera sceneriet i Figur 2.3 där blå pluton patrullerar ett stadsområde under relativt hög hotbild. Varje soldat är utrustad med sensorer som registrerar position, puls, förbrukad ammunition mm. Soldatens samlade sensorinformation skickas var 15:e sekund via radion till en nod med mera beräkningskapacitet för analys (blå stjärna). Där sker bearbetning av sensorinformationen för att bland annat göra en bedömning om soldaternas stressnivå och uthållighet. Lägesbilden sprids inom plutonen samt till distribuerade databaser i omgivningen, bl.a. till överordnade chefer vilka kan nyttja informationen för framtagande av nya planer.

Första gruppen (blå fyrkant) sammanstötter med en uppretad folkmassa, gruppchefen lyckas trots turbulensen att på ett ungefär markera folkmassans position på kartan (orange stjärna i Figur 2.3). Situationen är kaotisk. Det är svårt att bedöma folkmassans storlek och om flera grupperingar ingår. Stenkastning och skottlossning sker och bilar sätts i brand. Blå grupp (fyrkant) begär omgående hjälp från övrig i blå pluton medan elden besvaras. Soldatsensorerna registrerar ökad stress och minskad ammunition.



Figur 2.4: Producenter (lila) och konsumenter (ljusblå) för soldatstatusjänster.

	Producent	Konsument	Antal	Storlek	Trafikflöde
Soldatstatus	soldater	beräkningsnod	1 per 15 sek	200 B	unicast från 36 noder
Lägesbild soldatstatus	beräkningsnod	soldater	1 per 2 min	10 kB	multicast till 36 noder

Tabell 2.2: Trafikegenskaper - storlek på data objekt och trafikflöden

2.2.1 Tjänstens struktur

Sensordata skickas från de 36 soldaterna till den distribuerade databasen var 15:e sekund. Sensordata har en storlek av 200 B. I en av noderna finns en beräkningsdator som sammanställer sensorinformationen till en lägesbild som kontinuerligt sprids till grupperna. Eftersom lägesbilden även finns i databasen är den tillgänglig för kompanichefen. Lägesbilden har en storlek av 10 kB och vi kan anta att den sprids till alla (36) i plutonen varannan minut, Figur 2.4. I Tabell 2.2 sammanfattas trafikmängden från vårt andra scenario.

2.3 Scenario videodistributionstjänster

För övervakning och för att kunna skapa en lägesbild är transmission av video och bildsegment en efterfrågad tjänst. De noder/enheter som genererar vi-

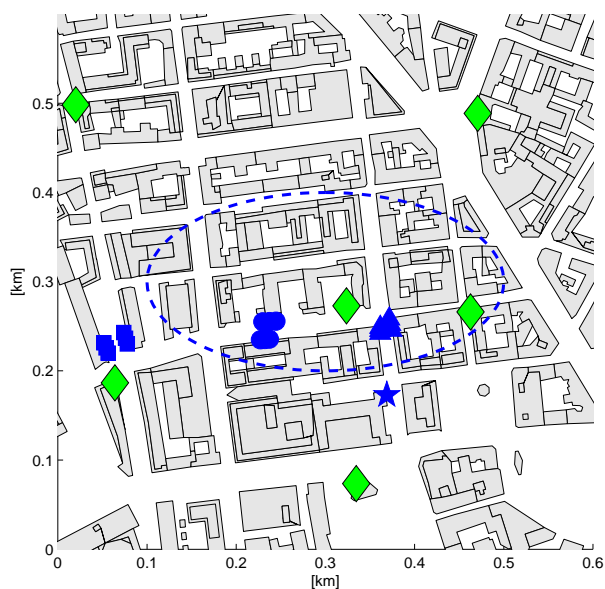
deo/bilder kan vara av olika typer, t ex, soldater, fordon, sensorer, UGVer och UAVer. Vid avsökning av ett område, eller olika rum inomhus, kan video från en UGV användas. Video från en UAV kan ge en bra översikt över ett geografiskt område och underlätta planeringen av framtida aktioner. Video kan också användas för diagnos av skada och telemedicin. Det kan också vara så att video eller bilder behövs långt i efterhand för att analysera ett händelseförlopp. Det finns alltså många situationer där video kan vara efterfrågad men samtidigt är överföring av video väldigt kapacitetskrävande. Det finns därför skäl att försöka begränsa distributionen av de olika videoströmmarna till de situationer och enheter där behovet verkligen finns. Dessutom kan man när det är möjligt passa på att sända videoströmmar och bilder när ledig kapacitet finns.

Scenariot här är övervakning i stadsmiljö med relativ hög hotnivå. En UAV finns tillgänglig och ett antal marksensorer utrustade med videokameror är utplacerade på strategiska platser. De individuella soldaterna är också utrustade med videokamera. En blå pluton bestående av 4 grupper med vardera 3 soldater genomför ett patrulleringsuppdrag i stadsmiljö. Dessutom finns en ledningsgrupp bestående av 2 personer i ett ledningsfordon som är parkerat centralt i staden.

I Figur 2.5 flyger UAV:en (grön flagga) över stadsdelen i en bana (grönstreckad ellips) och filmar de olika områdena. Videon från UAV:en sänds direkt till ledningsvagnen (blå stjärna) på en dedicerad förbindelse. Väl i ledningsvagnen publiceras videon i den distribuerade databasen för vidare distribution till andra konsumenter. Övriga videoproducenter antas ha en egen instans av databasen så att de kan publicera data direkt i den. Marksensornerna (gröna halvmånar) är passiva så länge inget händer men startar att filma när något rör sig inom deras bevakningssfär. Soldaterna (blå symboler) filmar också när de bedömer att något är av intresse. Istället för att hela tiden distribuera video från producenterna till alla noder/soldater i nätet distribueras den bara efter behov. De soldater som har behov av att undersöka ett visst område kan göra en förfrågan via databasen efter relevanta videoklipp, som visar det aktuella området. På samma sätt kan information och video från marksensorer och andra soldater efterfrågas vid behov.

2.3.1 Tjänstens struktur

Kapacitetsbehovet för en videoöverföring är kraftigt varierande beroende på vilken kvalitet man behöver, t. ex från 128 kbps upp till 3.5 Mbps. Vi definierar därför tre videokvaliteter; en på 200 kbps av låg kvalitet, en på 800 kbps av medelkvalitet och på 3.2 Mbps av hög kvalitet. Storleken på en bild kan också

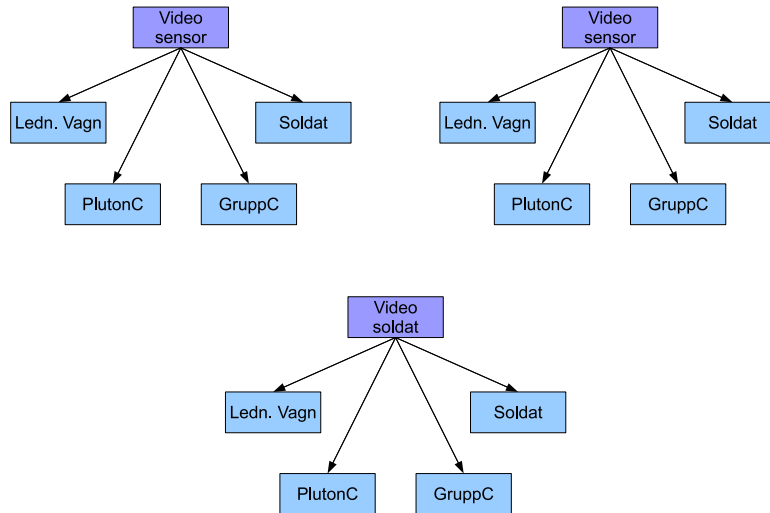


Figur 2.5: Videodistributionstjänst vid övervakning i stadsmiljö. De gröna romberna är marksensorer och den gröna linjen representerar UAV:ens bana. Den blå stjärnan är ett ledningsfordon.

variera kraftigt, en enklare bild kan t. ex vara 20 kB medan en högupplöst bild kan vara 2 MB eller ännu större. I Figur 2.6 beskrivs konsumenter och producer center schematiskt för videodistributionstjänsten och i Tabell 2.3 sammanfattas tjänstens egenskaper.

2.4 Sammanfattning av tjänsternas egenskaper

Det finns ett antal principiella egenskaper hos tjänsterna som är viktiga att ta hänsyn till vid val av databaslösning. Vi har gjort en grov indelning av egenskaperna i Tabell 2.4. Antal konsumenter och antal medelanden är grovt klassificerade som få, mellan och många. Storleken av dataobjekt grupperas i små, mellan och stora.



Figur 2.6: Producenter (lila) och konsumenter (ljublå) för videodistributionstjänsten

	Producent	Konsument	Datatakt	Trafikflöde
Video	UAV	ledn. vagn/ gruppchef/ soldat	200 kbps - 3.2 Mbps	multicast till dem som efterfrågar
Video	video sensor	ledn. vagn/ gruppchef/ soldat	200 kbps - 3.2 Mbps	multicast till dem som efterfrågar
Video	soldat	ledn. vagn/ gruppchef/ soldat	200 kbps - 3.2 Mbps	multicast till dem som efterfrågar

Tabell 2.3: Trafikegenskaper - storlek på data objekt och trafikflöden

	Producent	Konsument	Antal konsumenter	Antal medelanden	Meddelandestorlek
Ledningstjänst	ledningsfordon	underställda	många	mellan	mellan
Video	UAV/senor	gruppchef/soldater	få	få	stora
Soldatstatus	alla/stridande	stab/ledningsfordon	mellan	många	små

Tabell 2.4: Principiella egenskaper hos tjänsterna

3 Befintlig Teknik

Även om forskningen inom området har delvis andra grundantaganden än i detta arbete finns det en hel del tidigare arbeten som är relevant för vårt problem. Detta kapitel sammanfattar några av de mest relevanta arbetena. I den första delen beskrivs några befintliga designlösningar som är av intresse för de applikationer vi fokuserar på i denna rapport. I den andra delen tar vi upp två grundläggande metoder som främst är tänkta som referensmetoder. Slutligen beskriver vi hur data effektivt kan spridas till alla noder i nätet med hjälp av multicast-träd.

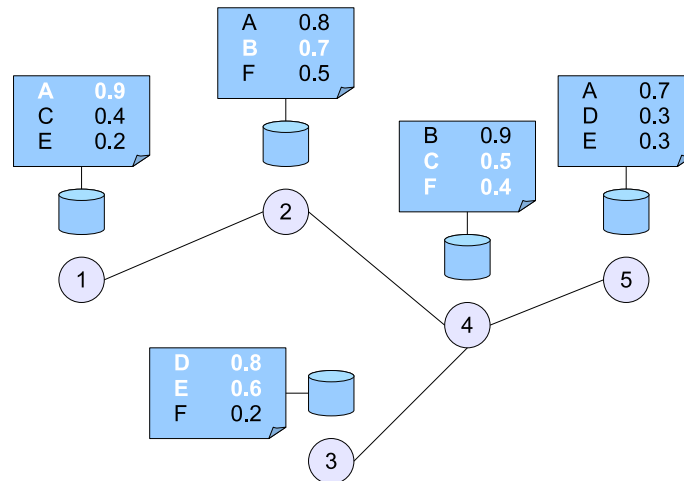
3.1 Forskningsfronten

Kravbilden hos de tjänster som har beskrivits i kapitel 2 har gjort att fokus för projektet har förändrats sedan fjolårets rapport [1]. I den tidigare rapporten tog vi upp en del databastyper för tjänster där det finns ett behov av att ha gemensamma data som grupper av noder kan ändra, som t.ex. ett saldo på ett bankkonto. De tjänster som projektet nu har valt att fokusera på har dock främst behov av att kunna tillgängliggöra ny data för grupper av noder, så kallad replikering av data. Förändringen innebär att en del nya designer har tillkommit inom sfären av intressanta koncept.

3.1.1 Replikering av data

Inom området replikering av data i mobila ad hoc-nät finns ett antal olika arbeten [2]. Gemensamt för de flesta är antagandena att minne är en begränsad resurs medan kommunikationsresursen inte utgör någon större begränsning. Grundantagandena är således i princip motsatta till vad vi antar i denna studie. Att grundantagandena är så olika beror sannolikt på att många arbeten inom området baserar sig på civila korthållskommunikationsystem, typ WLAN. Dessa system har betydligt högre kapacitet jämfört med det mera långräckviddiga militära system vi här betraktar. Vidare är de flesta arbetena några år gamla vilket möjligen kan förklara synen på minnestillgången.

Fokus för många papper är följaktligen att optimalt utnyttja det tillgängliga minnet hos en nod så att tillgängligheten hos data är så hög som möjligt om noden skulle tappa kontakten med resten av nätet, se Figur 3.1 för ett exempel hur detta kan se ut. För att ytterligare öka tillgängligheten finns också flera metoder där noder som verkar ha bra förbundenhet bildar grupper varefter data



Figur 3.1: Exempel på nät med 5 noder och 6 dataobjekt A,B,C,D,E,F. De olika noderna kan maximalt lagra 3 objekt sina respektive databaser. Till varje objekt associeras en användningsfrekvens. Vilka objekt som lagras i en nod avgörs av dess användningsfrekvens för de olika dataobjekten. Noden som publicerar ett visst objekt har det alltid i sin databas (fet vit stil).

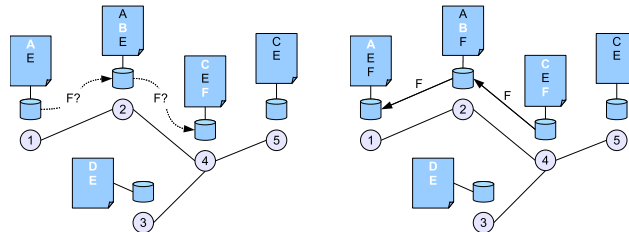
allokeras inom gruppen [3, 4]. På detta sätt fås sammanlagt mera minne vilket leder till högre tillgänglighet, givet att noder inte lämnar gruppen.

Det finns även arbeten där man i likhet med vårt arbete betraktar kommunikationsresursen som en begränsande faktor. I exempelvis [5] studeras olika metoder för att sprida data till alla noder i ett satellitsystem. För att avgöra vilken data som ska spridas i första hand beaktas både kostnaden och nyttan att sprida ett visst objekt. Metoden kräver dock mera sidoinformation än metoderna i [6, 7, 3, 8, 4] då nyttan av att sända ett visst dataobjekt måste estimeras.

I ett flertal metoder antas att användningsfrekvensen för olika typer av data är känd [6, 7, 8], dvs. att alla noder känner till hur ofta alla noder behöver tillgång till vart och ett av datatyperna. Ett nyckelproblem för oss om vi ska utnyttja dessa metoder är därmed att skatta användningsfrekvensen för olika datatyper. Normalt ignoreras detta problem genom antagandet att frekvenserna antas fixa och givna i förväg.

3.1.2 Mellanlagring av data

Att mellanlagra tidigare efterfrågad data (s k caching) och låta nya förfrågningar använda sådan data är idag en vanlig metod att minska trafiken i fasta IP-nät. Vanligen sker lagringen på flera olika platser i nätet, dels lokalt i klientens applikation, dels i strategiska nätverkspunkter hos olika företag och Internet-



Figur 3.2: Exempel på system där noderna mellanlagrar all data som passerar i mån av plats. En nod kan maximalt lagra 3 objekt i sin databas, egenproducerade dataobjekt är markerade med fet vit stil. Nod 1 behöver dataobjektet F och skickar en förfrågan till producenten (nod 4). När svaret reläas i nod 2 ersätts E i databasen av F.

leverantörer. För att mellanlagring av data ska vara effektivt krävs att många användare vill komma åt samma data. Sannolikheten för detta ökar givetvis om många användares förfrågningar passerar genom samma mellanlagringsplats. En metod att åstadkomma det är att låta fysiskt separerade mellanlagringsplatser samarbeta och utbyta information om sitt innehåll [9].

Möjligheten att använda mellanlagring i mobila ad hoc-nät har studerats i [10] och [11]. Normalt antas alla noder ha förmåga att lagra data som passerar dem. Minnet antas dock vara begränsat så frågan vad som ska lagras är central. För att avgöra vilka data som ska mellanlagras i en specifik nod utnyttjas användningsfrekvenserna som antas vara kända. Se Figur 3.2 för ett exempel.

Det finns även förslag på strukturer där olika noder samarbetar för att spara resurser, även här antas minnet begränsat. I [10] presenteras en metod som försöker minimera kommunikationskostnaden genom att beakta både användningsfrekvensen och vart i nätet tidigare mellanlagrade objekt finns när noden avgör om ett visst objekt ska mellanlagras. Metoden kräver således information om både användningsfrekvenser och vilka noder som tidigare har lagrat ett visst objekt.

Metoderna som beskrivs i [10] och [11] kan ses som två extremfall sett till graden av samarbete. I [11] klarar sig noderna på lokal information medan man i den [10] känner till i princip allt om alla. I [10] jämförs också den förslagna metoden med de i [11]. Resultaten tyder på att det, i alla fall om mobiliteten inte är alltför hög, kan finnas vinster med att utbyta data om olika databasers innehåll.

3.2 Referensmetoder

Vi kommer här att presentera två enkla metoder för dataspridning. Dessa representerar två extremfall för en distribuerad databas, alla noder har all data respektive en nod har bara egengenererat data.

3.2.1 Fullständig spridning

Fullständig spridning av data kan vara en lämplig design om många noder har behov av samma data och data är i samma storleksordning som ett meddelande för att fråga efter data, t ex övriga enheters positioner. Om vi däremot har ett nät där stora mängder data, t ex video, som genereras i varje nod men bara ett fåtal noder har behov av data så innebär fullständig spridning ett stort resursslöseri.

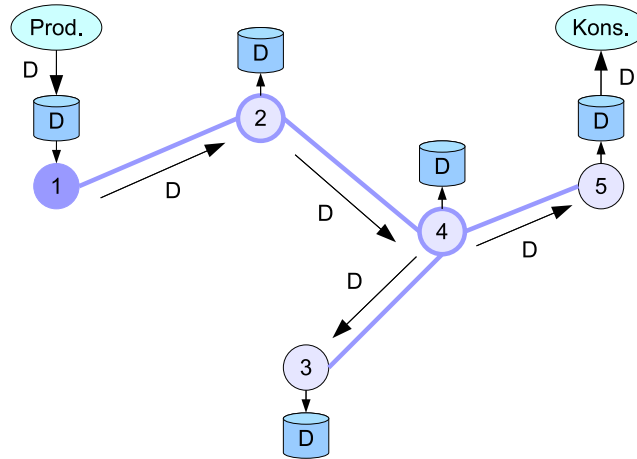
Att alla noder får all data, oavsett om de har behov av den eller inte, innebär att det genereras mycket multicast-trafik i nätet. Vi behöver därför en effektiv metod för multicast till alla noder. En enkel men relativt ineffektiv multicast-metod är flooding där alla noder sänder ut ett meddelande en gång. Metoden kräver visserligen ingen information om nätets struktur utan bara att noderna har koll på vilka meddelanden de har sänt ut men leder i de flesta fall till onödigt många återutsändningar. Den höga kommunikationskostnaden måste således ställas mot metodens enkelhet.

Effektivare metoder bygger på det faktum att det ofta räcker med att en mindre grupp noder sänder ut ett meddelande för att avsedda noder i nätet ska få meddelandet. För att bestämma vilka noder som ska reläa ett meddelande bygger man ofta ett så kallat multicast-träd. Trädet innehåller alla noder som ska ha meddelandet samt eventuellt några extra som behövs för att meddelandet ska nå hela gruppen. I Figur 3.3 visas ett exempel på ett multicast träd med nod 1 som startnod.

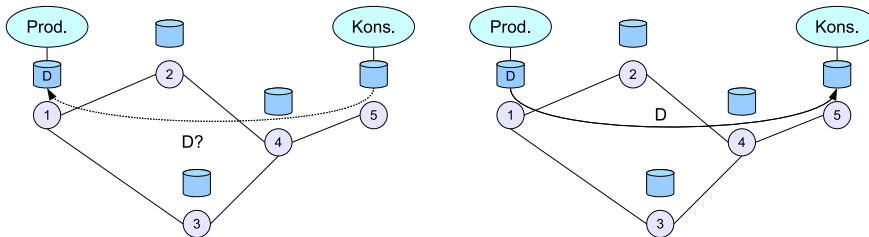
Metoden med fullständig spridning resulterar i hög redundans. En delning av nätet behöver således inte påverka tillgängligheten för data. Metoden bygger dock på aktuell data antas finns lokalt i noders databaser. Tillkommer nya noder till ett nät kan det därför finnas ett behov att synkronisera databaserna så att aktuell data verkligen finns tillgänglig lokalt hos alla noder.

3.2.2 Producentförfrågning

I ett nät där producentförfrågning används skickas inget data förrän det efterfrågas. Metoden är enkel. Ingen sidokunskap om användningsfrekvenser eller motsvarande behövs. Om en nod behöver ett visst dataobjekt och inte själv har



Figur 3.3: Nät där all data sprids till alla noder i nätet med hjälp av ett multicast-träd.



Figur 3.4: Nät utan distribuerad databas.

det skickas en förfrågan till dataobjektets producent varpå denna skickar dataobjektet till konsumenten, se Figur 3.4. Här görs antagandet att noderna har tillgång till ett tjänsteregister som anger vilka typ data som finns i respektive nod.

Detta kan vara en lämplig design när kostnaden för en förfrågan är liten jämfört med kostnaden för att reläa dataobjektet samtidigt som få noder inom samma geografiska område efterfrågar samma data. En fördel med denna design är att det inte blir några problem med konsistens då data hämtas när den behövs från datakällan. Däremot är redundans ett problem om kontakten med den dataproducerade noden är bruten kan man inte få tag på data.

3.3 Multicast-träd

För att på ett effektivt sätt sprida information till en grupp av noder i ett ad hoc-nät utnyttjas normalt någon form multicast-metod. Dessa baseras ofta på att ett multicast-träd skapas. Trädet innehåller alla noder som ska ha meddelandet samt eventuellt några extra som behövs för att meddelandet ska nå hela gruppen.

Den stora utmaningen ligger i att på ett kostnadseffektivt sätt skapa och underhålla multicast-träd för olika grupper av noder. Att generera optimala träd i nät som förändras över tiden är knappast realistiskt. I stället gäller det att hitta algoritmer som kan generera och underhålla tillräckligt bra träd till en rimlig kostnad. Vi väljer här att fokusera på multicast-träd som når alla noder i nätet.

3.3.1 MPR baserade multicast-träd

En metod för att skapa multicast-träd är att skapa lokala träd som är bra och sedan kombinera dessa till globala träd. Det globala trädet blir visserligen inte optimalt, men ofta blir det tillräckligt bra. Denna teknik används av exempelvis routingprotokollet Optimized Link State Routing Protocol (OLSR) [12] för att sprida information.

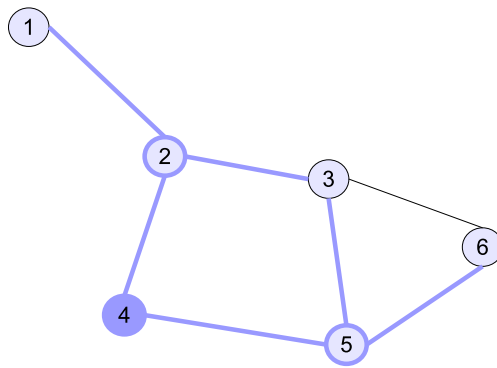
Alla noder antas i OLSR känna till alla sina enhopps- och tvåhoppsgrannar samt alla länkar mellan dessa noder. Noder utser ett antal av sina enhoppsgrannar till så kallade MultiPoint Relays (MPR). MPR-noderna väljs så att om alla MPR-noder återutsänder ett meddelande från noden som valde dem så skall alla tvåhoppsgrannar till ursprungsnoden få meddelandet. Se Figur 3.1 och Tabell 3.1 för ett exempel.

Vid valet av MPR-noder försöker noden normalt att minimera antalet återutsändningar, dvs. välja så få MPR-noder som möjligt. Vill vi sprida ett meddelande längre än två hopp så låter vi först ursprungsnodens MPR:er sprida meddelandet till alla tvåhoppsgrannar. De tvåhoppsgrannar som är valda till MPR-noder till ursprungsnodens MPR-noder sänder sedan meddelandet till alla trehoppsgrannar och så vidare. För att undvika återutsändning av redan sända meddelanden antas noderna hålla koll på vilka paket de har sänt vidare.

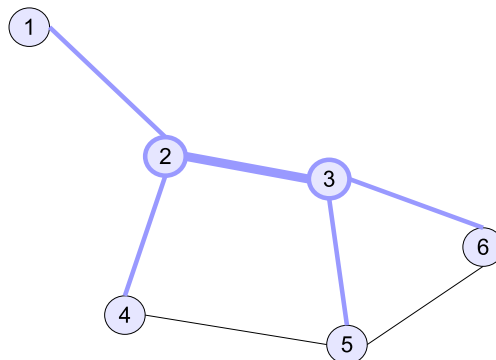
Metoden ovan resulterar ofta i olika multicast-träd för olika startnoder. Vill vi istället ha ett gemensamt multicast-träd för alla noder kan ett sådant konstrueras utgående från samma grundinformation. En metod är att skapa ett så kallat CDS (Connected Dominating Sets), se [13, 14]. Ett CDS är en grupp av noder som tillsammans har alla andra noder i ett nät som grannar samtidigt som det finns länkar mellan noderna i CDS:et, dvs. vårt nät är inte delat. En nackdel med denna typ av träd är dock att de vanligen kräver lite flera återutsändningar

Nod	MPR
1	2
2	3
3	2
4	2 & 5
5	3 (alt. 4)
6	3 & 5

Tabell 3.1: Olika noders val av MPR noder för nätet i Figur 3.1.



Figur 3.5: Exempel på lokalt multicast-träd med nod 4 som ursprungsnod och nod 2 och 5 som MPR:er. Tabell 3.1 ger övriga noders val av MPR:er. 1.3.2 CDS baserade multicast-träd.



Figur 3.6: Exempel på CDS-träd. CDS:et utgörs här av nod 2 och 3.

än metoder där olika träd används för olika startnoder.

Ett CDS baserat multicast-träd kan skapas på flera olika sätt, en metod är att skapa ett träd där alla noder som är MPR noder för någon nod ingår och sedan reducera detta. För att minska reduceringsfasen kan MPR-noderna väljas så att de i möjligaste mån sammanfaller med grannodernas val, se [13, 14] för en mera utförlig beskrivning av algoritmen. Denna metod kräver inte mer information än vad som behövs för att välja MPR noderna vilket gör den till ett intressant alternativ.

Om vi tillämpar den i [13, 14] beskrivna algoritmen på exemplet i Figur 3.1 så får vi CDS-trädet i Figur 3.6, dvs. vi får en nod mindre i trädet än om vi inkluderar alla MPR noder i CDS trädet.

Vilken typ av multicast-metod som är lämpligast beror bland annat på typ av access- och routingprotokoll som används och vilka applikationer som används i noderna. För ett TDMA baserat accessprotokoll kan exempelvis ett entydigt globalt träd ge vinster, trots att det ökar sändningskostnaden något, eftersom det underlättar trafikutjämnningen i nätet.

3.3.2 Multicast-träd och variabel datatakt

Problemet att konstruera multicast-träd blir mera komplicerat om nätet erbjuder variabel datatakt på länkarna [15]. Att avgöra vilken datatakt som ska användas är nämligen ett komplext problem. En enkel metod är givetvis alltid använda lägsta möjliga datatakt så att så många noder som möjligt nås vis en sändning. Nyttan med variabel datatakt blir dock mycket låg med denna strategi. En alternativ metod är att om möjligt använda en högre datatakt och låta några noder

reläa pakten så att vi når all noder som kan nås på den lägsta datatacken. Hur en sådan metod praktiskt skall se ut är dock ett komplext problem.

FOI-R--2933--SE

4 Ny Teknik

I denna del av rapporten presenterar vi vårt designbidrag i form av två algoritmer. Den ena bygger på mellanlagring av tidigare efterfrågad data medan den andra bygger på att data sprids till noder där den ofta är efterfrågad iförväg. Båda algoritmerna har klara likheter till de tidigare beskrivna algoritmerna men vissa anpassningar har gjorts till våra antaganden rörande kommunikationsresurser och minnestillgång.

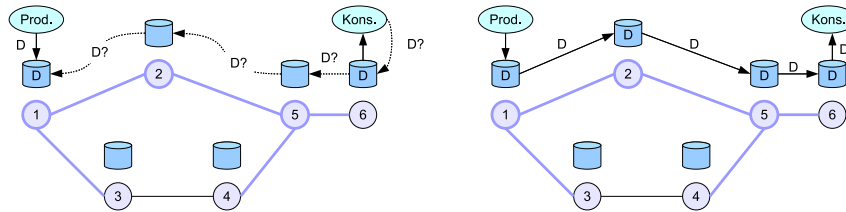
Vilken metod som är lämpligast beror på vilken typ av data som ska distribueras. Den första metoden är främst avsedd för applikationer som producerar stora dataobjekt som relativt få har behov av. Den andra metoden är främst avsedd för applikationer som producerar små data objekt som många har behov av. Det är dock viktigt att observera att den senare typen av algoritmer i många scenarion även behöver ha möjlighet att efterfråga saknad data, eftersom vi inte alltid kan prediktera vilken data som kommer att efterfrågas hos en viss nod. Behovet finns därför av hybridalgoritmer som använder olika metoder beroende på vad det är för data som efterfrågas.

Vi börjar med att beskriva metoden som bygger på att data hämtas då den behövs, vi går sedan vidare och beskriver de metoder där data sprids, innan den behövs, i förebyggande syfte. En viktig fråga är hur konsistens och aktualitet hos data hanteras. Dessa frågor diskuterar vi i det efterföljande avsnittet. Vi avslutar sedan med en diskussion om hybridalgoritmer.

4.1 Reaktiv Mellanlagring

Mellanlagring av data är en vanlig metod att för minska trafiken i dagens fasta IP-nät. Att införa tekniken i ad hoc-nät introducerar dock ett antal nya utmaningar. För att reaktiv mellanlagring av data ska vara effektiv så måste mellanlagrad data återanvänds, dvs. förfrågningar måste passera noder med lagrad data för den aktuella frågan.

I fasta nät är det ofta möjligt att placera ut servrar med stöd för mellanlagring i punkter där mycket trafik passar, och därigenom uppnå en hög återutnyttjande grad av lagrad data. Att använda samma metod i mobila ad hoc-nät är dock inte helt enkelt då topologin och trafikflöden ständigt förändras. I stället för att utse vissa noder som ansvariga för mellanlagringen av data antar vi här att alla noder är ansvariga, givet att de har förmågan. Ad hoc-nätets inneboende brist på struktur gör dock att risken är stor att förfrågningar efter data inte passerar noder med mellanlagrad data utan i stället hamnar hos källan.



Figur 4.1: Nät med reaktiv mellanlagring. Nod 6 söker efter dataobjektet D publicerat i nod 1. Eftersom ingen nod längs vägen har objektet svarar nod 1.

4.1.1 Algoritm

En metod som skulle kunna förbättra återanvändningsgraden av mellanlagrad data är att styra trafiken i nätet så att vissa noder belastas mera. En metod att göra detta är att låta trafiken gå via ett gemensamt CDS träd. Genom att göra detta riskerar vi visserligen en högre kommunikationskostnad. Detta ska vägas mot möjligheten till en bättre fungerande mellanlagringsmekanism.

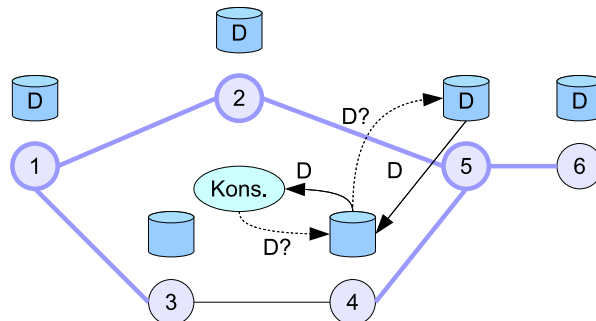
I Figur 4.1 och Figur 4.2 visas ett exempel på hur en reaktiv mellanlagring fungerar. Vi antar här att routrarna i nätet väljer att styra trafik via stomträdet även om kortare rutter existerar. Detta gör att sannolikheten att mellanlagrad data återanvänds ökar. I Figur 4.1 vill nod 6 ha tillgång till data objektet D som produceras i nod 1. Nod 6 sänder då en fråga till nod 1 via nod 5 och 2. Dessa har dock inte det eftersöka objektet utan frågan hamnar hos nod 1 som svarar nod 5 med objektet D. De noder som svaret passerar mellanlagrar svaret i sina databaser.

När nod 4 i Figur 4.2 efterfrågar samma data sänds frågan via stomträdet till nod 1. Eftersom nod 5 har objektet kan nod 5 svara nod 4 direkt. Hade inte frågan ruttats via stomträdet hade meddelandet troligen gått via nod 3 vilket hade gjort att nod 1 hade fått svara på frågan.

4.1.2 Lämplighet - designfördelar

Strukturen med distribuerad mellanlagring är lämpligast om flera noder inom samma geografiska område efterfrågar samma data, dvs. de använder huvudsakligen samma noder för att reläa trafik från producenten. Vidare bör kostnaden för att sända en fråga efter viss data vara liten jämfört med kostanden att överföra efterfrågad data.

En uppenbar fördel med denna typ av struktur är att den inte behöver innebära någon extra kommunikationskostnad relativt ett system utan distribuerad databas, givet att inte någon trafikkoncentrerande metod används. Risken är



Figur 4.2: Nät med reaktiv mellanlagring av data. Nod 4 söker efter dataobjektet D som nod 1 har producerat och sänder en förfrågan via stomträdet till nod 1. När förfrågan når nod 5 har denna nod efterfrågad data i sin databas och svarar nod 4 direkt utan att sända vidare förfrågan.

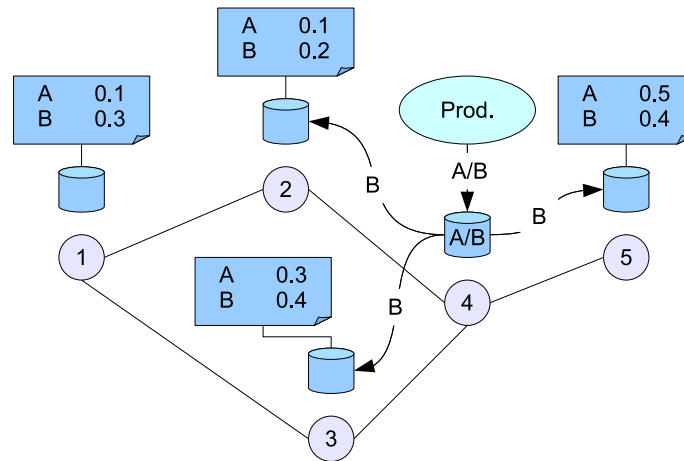
dock stor att utan koncentration av trafiken så blir träffsannolikheten låg och effekten av databasstrukturen ringa. Frågan om aktualitet och konsistens för data behöver också hanteras explicit för denna metod. Att data finns i ett mellanlager betyder ju inte att den är aktuell.

Om kommunikationssystemet erbjuder aktiva val av datatakt tillkommer ytterligare en problemställning. I system med fix datatakt innebär det ingen extra kommunikationskostnad att sprida data till alla sina grannar när man ändå sänder till en. I ett system med flera datatakt kan dock kostnaden vara avsevärd då vi i många fall kan använda högre datatakt på enskilda länkar än vad vi kan om vi vill nå alla våra grannar. Vinsten av att sprida data i nätet måste såldes här nogra vägas mot den extra kommunikationskostnaden.

4.2 Aktiv spridning

Det finns idag gott om algoritmer som aktivt sprider data till de noder som bäst behöver den. De allra flesta bygger dock på grundantagandena att lagringskapaciteten utgör den största begränsningen medan kommunikationsresursen är tillräcklig. Våra grundantaganden att minnet är tillräckligt medan kommunikationsresursen begränsad gör därför att algoritmerna behöver anpassas för att fungera väl.

Målsättningen med algoritmen är att aktivt sprida det data som utnyttjas mest först. Finns det sedan ytterligare tillgängliga kommunikationsresurser sprids det data som utnyttjas näst mest osv. Om ett dataobjekt har en överlevnadstid associerad med sig kan algoritmen även ta hänsyn till denna så att objekt som snart blir ogiltiga inte sänds i onödan.



Figur 4.3: Nät med aktiv spridning av data. Nod 4 har producerat dataobjekt av typen A och B. För att avgöra vilken data som ska spridas först så beräknas den totala accessfrekvensen för övriga noder i nätet.

4.2.1 Algoritm

För spridning av data över flera hopp antar vi att någon typ av multicastträd baserat på MPR-noder används. Vid spridning över flera hopp försvåras problemet att avgöra nyttan av att sända ut ett visst objekt då inte enbart nodens enhoppsgrannar behöver beaktas utan även noder på längre avstånd. Flera olika metoder är här tänkbara, en möjlighet är att ignorera problemet och enbart beakta de grannar noden har ansvar för i sin roll som MPR-nod. Även mera avancerade alternativ där noder på längre distans beaktas är tänkbara.

En viktig frågeställning när nyttan av att sända ett visst dataobjekt skattas är om nyttan för flerhoppgrannar ska vägas relativt nyttan för enhoppsgrannar. Flerhoppgrannarna kräver ju att dataobjektet reläas minst en gång extra. Vi väljer dock här att inte vikta nyttan då vi inte vill missgynna noder på långt avstånd.

I Figur 4.3 kan vi se ett exempel på hur nod 4 resonerar. Noden har producerat två dataobjekt av typerna A och B och måste nu bestämma vilket objekt som ska sändas ut först. Ovanför de olika nodernas databaser visas accessfrekvensen för de två datatyperna hos de olika noderna. Den totala accessfrekvensen för typen A blir således 1.0 (0.1 + 0.1 + 0.3 + 0.5) respektive 1.3 (0.3 + 0.2 + 0.4 + 0.4) för B typen. Nod fyra ska således sända B typen i första hand. Observera att nodens egna accessfrekvenser inte tas med.

Om vi har en modell för hur länge ett visst dataobjekt är aktuellt innan det ersätts av ett annat så kan accessfrekvenserna justeras så att även detta beaktas.

4.2.2 Lämplighet - designfördelar

Algoritmen är främst avsedd för situationer där de tillgängliga kommunikationsresurserna periodvis understiger behovet. Den inneboende förmågan att prioritera vad som sänds kommer då bäst till sin rätt. Då algoritmen aktivt sprider data till konsumenter istället för att de frågar efter data är den lämplig om dataobjekten är små, dvs. kostnaden att fråga efter ett dataobjekt är i samma storleksordning som att överföra själva dataobjektet. Vidare krävs att användarmönstren är kända samt att relativt många noder vill ha viss data. Är antalet noder som vill ha viss data lågt bör metoder där data sänds direkt istället övervägas, då aktiv spridning riskerar att sprida data till onödigt många noder efter vägen. Överspridningen kan dock vara positiv sett ur redundanssynpunkt om nätet delar sig samtidigt som nya konsumenter tillkommer.

Att enbart använda aktiv spridning som metod är troligen inte möjligt i många fall. Metoden måste kompletteras så att de som vill få tillgång till data, som de normalt inte använder, kan få det. I detta fall bör stöd för att efterfråga data erbjudas. Vidare måste metoden kunna hantera att nät delar sig och åter går samman. I detta fall behöver databaserna kunna synkroniseras.

4.3 Konsistens och aktualitet hos data

Om användarnas tillit till sina applikationer ska vara hög krävs att det data databasen tillhandahåller är tillräckligt aktuellt och konsistent. Med aktualitet hos data avser vi här att de data som presenteras ska innefatta den som senast producerats. Med konsistens hos data avser vi här att samma data ska presenteras oavsett vilken instans av databasen vi använder.

I praktiken är det givetvis inte möjligt att nyproducerad data presenteras direkt hos alla konsumenter utan en viss fördröjning måste alltid accepteras. På samma sätt går det inte att garantera fullständig konsistens. Vissa instanser kan ju sakna förbindelse med instanser med ny data vilket omöjliggör att dessa instanser blir konsistenta tills kommunikationen återupprättas.

4.3.1 Synkronisering mot producent

Aktualitet hos data kan hanteras på olika sätt. Det kanske enklaste, och idag vanligaste sett till dagens Internet, är tidsstämplig av data. En överlevnadstid binds här till producerad data. När överlevnadstiden har passerat betraktas data som inaktuellt och slängs. Ett alternativ till att slänga data när överlevnadstiden har passerats är att fråga producenten om data behöver uppdateras. Om produ-

centen inte anser att det är nödvändigt att uppdatera data returnerar den bara en ny överlevnadstid för det aktuella objektet.

Svårigheten med metoden är att avgöra vad som är en rimlig överlevnadstid. Produceras data med konstanta tidsintervall är det detta relativt enkelt. Många applikationer har dock mera slumpmässiga produktionsmönster vilket gör det betydligt svårare att sätta överlevnadstiden. En möjlig metod är att sätta överlevnadstiden relativt lågt och sedan låta konsumenten genomföra ovanstående synkronisering mot producenten. Detta innebär dock en del extra trafik. Vidare uppstår problem om producenten inte är tillgänglig för konsumenten på grund av exempelvis fragmentering hos nätet. En möjlig kompromiss i sådana situationer är att undersöka vilken data som är mest aktuellt i det egna segmentet och sedan använda den. Några garantier för att detta är det aktuella data finns givetvis inte men data är i alla fall konsistent i nätsegmenten. Då nätets struktur åter kan förändras bör även i detta fall en överlevnadstid sättas så att processen återstartas efter en viss tid. Möjligen skulle upptäckten av rutter till nya noder kunna användas istället för överlevnadstiden i fall där kontakt med producent-noden saknas.

4.3.2 Synkronisering mot granne

Om den aktuella metoden för att sprida data baseras på att konsumenten aktivt efterfrågar data, som på dagens Internet, passar metoden med överlevnadstider relativt väl. Används metoder som bygger på att producenten aktivt sprider data till konsumenter matchar metoden dock inte så väl. Noden förväntas ju inte här efterfråga data vilket gör att synkronisering mot producenten inte passar in i mönstret. För dessa metoder väljer vi istället att använda parvis synkronisering mellan noder då nya enhopsgrannar bildas.

I ett mobilt ad hoc-nät, där ständigt nya nätstrukturer bildas, finns en stor risk att olika databasers innehåll kommer att skilja sig åt relativt mycket. Att fullständigt synkronisera två sådana databaser är en komplex process som riskerar att kräva en hel del kommunikationskapacitet. Vi väljer här att inte vidare studera frågan hur en fullständig synkronisering effektivast ska ske utan fokuserar på en partiell synkronisering av de aktuella dataobjekten. Vi antar här att noderna har en uppfattning om med vilka noder de senast har synkroniserat så att en instabil länk som fluktuerar mellan att existera och inte existera inte upprepade synkroniseringar.

En frågeställning som uppkommer här är också vilka datatyper vi ska synkronisera. Två olika databaser kan ju ha olika kravbilder på vilken data de ska innehålla. Samtidigt kan det ju finnas andra noder i grannskapet som vill ha

dataobjekten i fråga. En möjlighet här kan vara att utnyttja användningsfrekvenserna, som t.ex. den proaktiva mellanlagringen använder för att bestämma vad som ska synkroniseras.

Flera möjliga metoder finns för att partiellt synkronisera två databaser. En enkel metod är att sända alla objekt utan att undersöka vad som skiljer databaserna åt. Detta riskerar dock att bli ineffektivt om skillnaderna är små eller obefintliga. En alternativ metod är att först jämföra datamängderna och sedan enbart kommunicera data som faktiskt skiljer sig åt. Detta kräver dock att kostnaden för att jämföra datamängderna är lägre än kostnaden att jämföra dataobjekten. Detta kan exempelvis göras genom att jämföra nycklarna som tillhör dataobjekten. Metoden fungera troligen bra om dataobjekten är stora relativt sina nycklar samtidigt som andelen dataobjekt som behöver synkroniseras är låg. Är objekten små relativt sina nycklar samtidigt som huvuddelen behöver synkroniseras är dock metoden mindre lämplig. Ett tredje alternativ kan vara att först jämföra hashnycklar för grupper av data. Skiljer sig nycklarna åt synkroniserar vi all data i gruppen. Är skillnaderna stora mellan datamängderna riskerar vi dock att huvuddelen av grupperna måste synkroniseras vilket gör metoden relativt ineffektiv.

4.4 Hybridalgorithm

Den mångfacetterade kravbilden hos de applikationer som kan tänkas utnyttja en distribuerad databas gör att databasen troligen måste tillhandahålla olika spridningsmetoder för olika applikationer. De här beskrivna metoderna bör i detta sammanhang därför inte ses som konkurrerande tekniker utan mer som delar av en hybridalgorithm.

En viktig fråga som måste studeras vidare är vilka algoritmer som olika datatyper ska använda för att distribuera data. Givet algoritmernas konstruktion kan vissa grundläggande principer ses, se Tabell 4.1. För mera precisa regler för vilka algoritmer som ska användas för olika datatyper och producent-/konsumentmönster krävs dock sannolikt simuleringar.

Vid design av hybridalgoritmer som erbjuder flera olika metoder för att distribuera data är det fördelaktigt om så stor andel som möjligt av eventuella fasta kostnader kan delas mellan de olika delalgoritmerna som ingår i hybridalgoritmen. Exempel på sådana kostnader är trafik för att skapa ett multicastrod. Det kan därför vara värt att överväga tekniskt sett icke-optimala lösningar för algoritmerna om det leder till mindre extra trafik. Ytterligare utvärderingar krävs dock här för att kunna göra en korrekt avvägning.

	Algorithm			
	Producent förfrågan	Reaktiv mellanlagring	Aktiv spridning	Fullständig spridning
Storlek data	stora	stora	små	små
Antal konsumenter av visst data	få	många	många	många
Producent mönster	stokastiskt	stokastiskt	periodiskt	periodiskt
Kommunikationsresurs-hantering	extern	extern	intern	extern

Tabell 4.1: Sammanfattning av egenskaper hos fyra olika metoder för att distribuera producerad data för konsumenter.

5 Slutsatser

Många av de tjänster som kan antas vara väsentliga i ett taktiskt mobilt nät kan dra fördel av en distribuerad databasstruktur som effektivt tillgängliggör data. Genom införandet av en sådan struktur kommer kommunikationsresurserna att kunna utnyttjas effektivare och fler användare få sitt informationsbehov tillgodosett. I denna rapport har tre sådana tjänster, en stridsledningstjänst, en soldatstatusstjänst och en videodistributionstjänst beskrivits och analyserats. Gemensamt för de tre tjänsterna är att förändringar i databasen främst görs genom att nya dataobjekt läggs till. Storleken på publicerade dataobjekt samt producent och konsumentmönstren skiljer sig dock väsentligt åt mellan de tre tjänsterna.

I rapporten presenteras även två möjliga databasdesigner. Den ena bygger på mellanlagring av tidigare efterfrågad data medan den andra bygger på att data sprids till noder där den ofta är efterfrågad. Vi beskriver även två referensmetoder, en som bygger på att data alltid hämtas hos producenten och en som bygger på att all data sprids till alla. Ett genomgående drag hos de beskrivna metoderna är att de har behov av övergripande routingstrukturer för att koordinera sökning och spridning av data på ett effektivt sätt. Problemet att finna lämpliga punkt-till-punkt router och multicast-träd som uppfyller algoritmernas strukturellakrav är således centralt.

En övergripande teoretisk analys av de beskrivna metodernas lämplighet för de utvalda tjänsterna visar att det är svårt att finna en databaslösning som lämpar sig för alla tjänsterna. En mera lämplig designfilosofi är antagligen att använda en hybridalgorithm som använder olika metoder för olika tjänster baserat på bland annat dataobjektens storlekt och hur många som har behov av objektet. För att kunna avgöra mera exakt vilken typ av algoritm som är lämplig för en viss tjänsts användarmönster och datastorlekar krävs vidare analys och simuleringar.

FOI-R--2933--SE

Referenser

- [1] Ulf Sterner, Anders Hansson och Jan Nilsson. Distribuerade databaser i mobil trådlösa nät - problemställningar. Teknisk rapport FOI-R-2566-SE, FOI Ledningssystem, Linköping, september 2008.
- [2] Prasanna Padmanabhan, Le Gruenwald, Anita Vallur, and Mohammed Atiquzzaman. A survey of data replication techniques for mobile ad hoc network databases. *The VLDB Journal*, 17(5):1143–1164, 2008.
- [3] Takahiro Hara, Yin-Huei Loh, and Shojiro Nishio. Data replication methods based on the stability of radio links in ad hoc networks. In *DEXA '03: Proceedings of the 14th International Workshop on Database and Expert Systems Applications*, page 969, Washington, DC, USA, 2003. IEEE Computer Society.
- [4] Jiun-Long Huang, Ming-Syan Chen, and Wen-Chih Peng. Exploring group mobility for replica data allocation in a mobile environment. In *CIKM '03: Proceedings of the twelfth international conference on Information and knowledge management*, pages 161–168, New York, NY, USA, 2003. ACM.
- [5] Bo Xu, Ouri Wolfson, and Sam Chamberlain. Cost based data dissemination in broadcast networks with disconnection. In *ICDT '01: Proceedings of the 8th International Conference on Database Theory*, pages 114–128, London, UK, 2001. Springer-Verlag.
- [6] Takahiro Hara. Effective allocation in ad hoc networks for improving data accessibility. In *IEEE Infocom*, pages 1468–1576, 2001.
- [7] Takahiro Hara. Replica allocation methods in ad hoc networks with data update. *Mob. Netw. Appl.*, 8(4):343–354, 2003.
- [8] Prasanna Padmanabhan and Le Gruenwald. Managing data replication in mobile ad-hoc network databases. *International Conference on Collaborative Computing: Networking, Applications and Worksharing*, 0:69, 2006.
- [9] Guohong Cao, Liangzhong Yin, and Chita R. Das. Cooperative cache-based data access in ad hoc networks. *Computer*, 37(2):32–39, 2004.
- [10] Bin Tang, Himanshu Gupta, and Samir Das. Benefit-based data caching in ad hoc networks. In *ICNP '06: Proceedings of the Proceedings of the 2006 IEEE International Conference on Network Protocols*, pages 208–217, Washington, DC, USA, 2006. IEEE Computer Society.
- [11] Liangzhong Yin and Guohong Cao. Supporting cooperative caching in ad hoc networks. *IEEE Transactions on Mobile Computing*, 5(1):77–89, 2006.

- [12] Thomas Clausen, Christopher Dearlove, and Philippe Jacquet. The optimized link state routing protocol version 2-ietf-manet-olsrv2-10. <http://ietfreport.isoc.org/all-ids/draft-ietf-manet-olsrv2-10.txt>, September 2009.
- [13] Jimmi Grönkvist. The use of dominating sets in ad hoc networks. Technical Report FOI Memo 2797, FOI Ledningsystem, Linköping, april 2009.
- [14] Cedric Adjih, Philippe Jacquet, and Laurent Viennot. Computing connected dominated sets with multipoint relays. *Ad Hoc & Sensor Wireless Networks*, 1:27–39, mars 2005.
- [15] Anders Hansson och Jimmi Grönkvist. Taktiska ad hoc-nät broadcast med adaptiv datatakt. Användarrapport FOI-R–2820–SE, FOI Ledningsystem, Linköping, september 2009.

