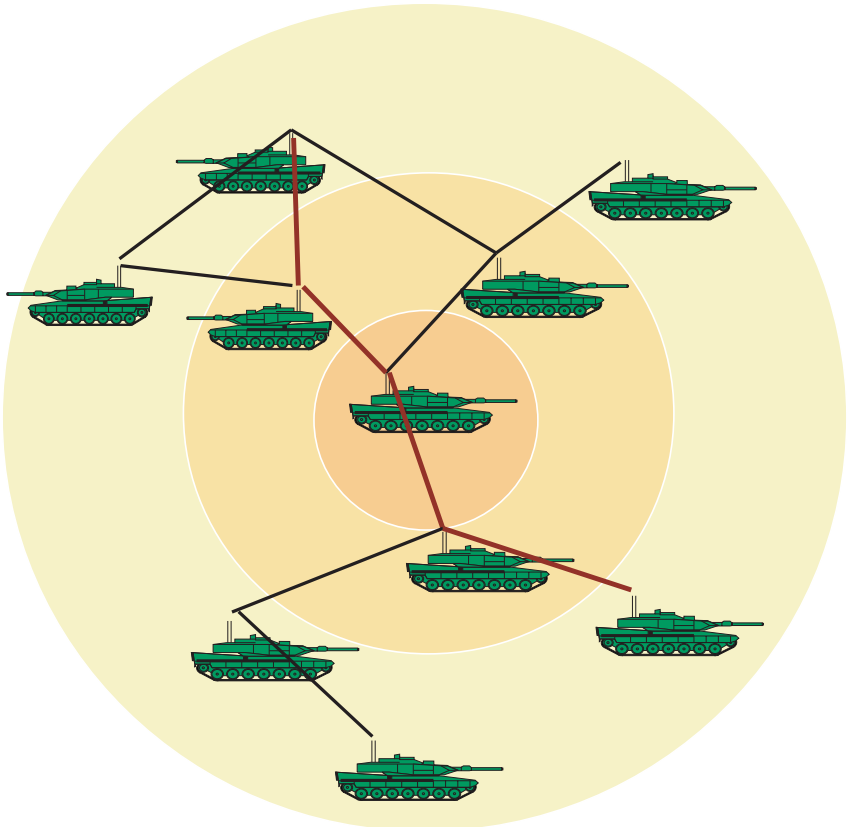
FOI
TOTALFÖRSVARETS
FORSKNINGSINSTITUT

FOI-R--0886--SE
Juni 2003
ISSN 1650-1942
Metodrapport

Katarina Persson

Routing med garanterad tjänstekvalitet i taktiska mobila ad hoc-nät



Avdelningen för Ledningssystem
Box 1165
581 11 LINKÖPING

TOTALFÖRSVARETS FORSKNINGSINSTITUT - FOI
Avdelningen för Ledningssystem
Box 1165
581 11 LINKÖPING

FOI-R- -0886- -SE
Juni 2003
ISSN 1650-1942

Metodrapport

Katarina Persson

Routing med garanterad tjänstekvalitet i taktiska mobila ad hoc-nät

Utgivare Totalförsvarets Forskningsinstitut - FOI Avdelningen för Ledningssystem Box 1165 581 11 LINKÖPING	**Rapportnummer, ISRN** FOI-R- -0886- -SE	**Klassificering** Metodrapport
	Forskningsområde 4. Spaning och ledning	
	Månad, år Juni 2003	**Projektnummer** E7035
	Verksamhetsgren 5. Uppdragsfinansierad verksamhet	
	Delområde 41. Ledning med samband och telekom och IT-system	
Författare Katarina Persson	**Projektledare** Mattias Sköld	
	Godkänd av Christian Jönsson	
	Uppdragsgivare/kundbeteckning FM - Försvarsmakten	
	Teknisk och/eller vetenskapligt ansvarig Jan Nilsson	
Rapportens titel Routing med garanterad tjänstekvalitet i taktiska mobila ad hoc-nät		

Sammanfattning

Trådlösa kommunikationssystem är en viktig del i framtidens militära operationer. Kraven som ställs på ett taktiskt radionät är många, bland annat robusthet, säkerhet, mobilitet och dessutom garanterad tjänstekvalitet, QoS. Man ska kunna lita på att informationen som sänds kommer fram inom en viss tid. Genom att använda mobila ad hoc-nät kan många av kraven uppfyllas, och man är också oberoende av fast infrastruktur.

En avgörande del i ad hoc-nätet är routingmetoden, d.v.s. metoden för att hitta vägar genom nätet och upprätthålla dessa. Routingmetoden ska vara så effektiv som möjligt för att kunna utnyttja nätets kapacitet i så hög grad som möjligt. Hög mobilitet och snabba topologiförändringar ställer höga krav på routing i taktiska mobila ad hoc-nät.

Vi undersöker i den här rapporten hur olika routingmetoder kan uppfylla våra krav på QoS. Det visar sig att routingprotokollen för ad hoc-nät i hög grad saknar möjlighet att ge QoS, men att de är under utveckling. Man kan konstatera att det är de proaktiva routingprotokollen som bäst utnyttjar övrig information som finns i nätet och kan ha de mest komplexa QoS-metoderna.

Nyckelord routing, ad hoc-nät, garanterad tjänstekvalitet	
Övriga bibliografiska uppgifter Sve	**Språk** Svenska
ISSN 1650-1942	**Antal sidor:** 59 s.
Distribution enligt missiv	**Pris: Enligt prislista**

Issuing organization FOI - Swedish Defence Research Agency Command and Control Systems P.O. Box 1165 SE-581 11 LINKÖPING SWEDEN	**Report number, ISRN** FOI-R--0886--SE	**Report type** Methodology Report
	Research area code 4. C^4ISR	
	Month year June 2003	**Project No.** E7035
	Customers code 5. Contracted Research	
	Sub area code 41. C^4I	
Author/s Katarina Persson	**Project manager** Mattias Sköld	
	Approved by Christian Jönsson	
	Sponsoring Agency FM- The Swedish Armed Forces	
	Scientifically and technically responsible Jan Nilsson	
Report title Quality of Service Routing in Tactical Mobile Ad Hoc Networks		

Abstract

Wireless communications will in the future be an important part of the tactical operations. Requirements on the network are robustness, security, mobility and guaranteed quality of service, QoS. It is important that the capacity or a maximum allowed delay of a connection can be assured. Mobile ad hoc networks are infrastructureless networks with no centralized administration which today can fulfil some of the demands.

The nodes in the network can communicate over multiple hops using a routing protocol. The method for routing is of major importance for the grade of utilization of the capacity in the network. Routing in a mobile ad hoc network is difficult because the network topology may change constantly which gives us imprecise information about the links.

QoS routing is one way to make it possible to give guarantees about capacity in an ad hoc network. A study and an evaluation of the existing routing protocols for ad hoc networks and their adaptability to guarantee QoS have been done. The conclusion is that little work has been done in this area, but a lot of methods are under development. Proactive protocols are best suited to use all information in the network and create good routes.

Keywords routing, ad hoc network, QoS	
Further bibliographic information Swe	**Language** Swedish
ISSN 1650-1942	**Pages** 59 p.
	Price acc. to pricelist

Innehåll

Kapitel 1

Inledning

1.1 Bakgrund

Framtidens militära operationer är beroende av ett robust och effektivt kommunikationssystem. Man vill kunna kommunicera och utbyta information mellan olika enheter inom försvaret, och systemet måste vara stabilt och säkert. Man ska kunna lita på att viktig information når mottagaren och att en begärd kvalitet på en tjänst kan garanteras, t.ex. eldgivningskommandon eller positionsförmedlingstjänster. I det framtida nätverksbaserade försvaret betonas även samverkan mellan informations-, lednings- och verkanssystem. Man vill utnyttja den information som finns tillgänglig i nätet. Kommunikation mellan olika enheter, vapenslag och även mellan olika nationer vid internationella insatser är viktigt, och det krävs att systemen är stabila, robusta och ger god prestanda. Då användarna är mobila är lösningen trådlösa kommunikationssystem, radionät. Dessa ska kunna kopplas samman med det fasta kärnnätet när det behövs och de ska snabbt kunna etableras.

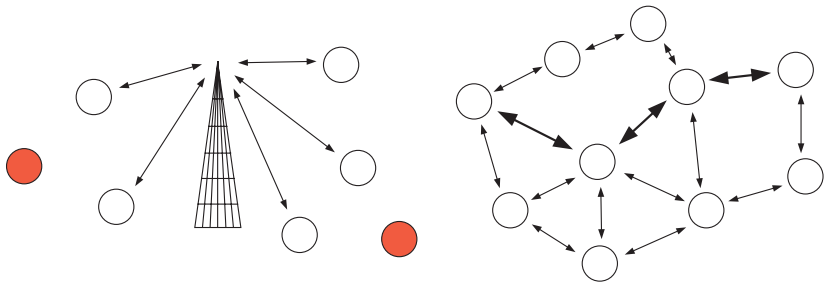
För att undvika svaga punkter i radionätet vill man inte ha någon central styrning. Ett cellbaserat system kan lätt slås ut genom att den centrala noden tas ur funktion. Man vill vara oberoende av utbyggd infrastruktur och om radiolänkar bryts eller ändras då noderna rör sig, ska nätet vara självkonfigurerande och självläkande. En viktig aspekt är också att näten ska klara av en viss nivå av störning och telekonflikt samt att det inte ska kunna upptäckas alltför lätt. Ytterligare ett krav är stöd för en positionsförmedlingstjänst, där man kan kontrollera

var alla enheter befinner sig, och även kunna rapportera in fiendens positioner. Eftersom olika nät ska kunna kommunicera med varandra finns också krav på kompatibilitet mellan dessa, t.ex. med det fasta telenätet. En typ av nät som är under utveckling för att kunna uppfylla dessa krav är *ad hoc-nät*.

1.2 Ad hoc-nät

Ad hoc är latin och betyder ”för detta ändamål”, och med det menas att nätet är anpassningsbart efter situationen och terrängen. Nätet är uppbyggt av noder som kan placeras ut slumpmässigt, t.ex. ett sensornät. Det är självkonfigurerande och ska klara av att delar av nätet slås ut. Varje nod fungerar som sändare och mottagare men fungerar också som router och kan reläa information mellan andra noder. Detta kan jämföras med mobiltelefonsystemet, ett cellbaserat nät, där man alltid måste befinna sig inom räckhåll för en basstation för att kunna kommunicera med andra noder, se figur 1.1. Två mobiltelefonanvändare som befinner sig relativt nära varandra kan ibland inte kommunicera eftersom de inte når basstationen. I ett ad hoc-nät är tanken att varje enhet ska kunna kommunicera med alla andra enheter som finns inom räckhåll. Befinner sig noderna som vill kommunicera utom räckhåll för varandra vidareförmedlas informationen via mellanliggande noder. Detta innebär att man har ett flerhoppsnät till skillnad från mobiltelefonnätet där man i princip har ett enhoppsnät. För att söka reda på vägen som informationen ska skickas i ad hoc-nätet används olika algoritmer. Detta kallas *routing*, och innebär att vägar söks upp och underhålls. Ju effektivare routingalgoritm man kan utveckla desto snabbare och bättre förbindelser går att få.

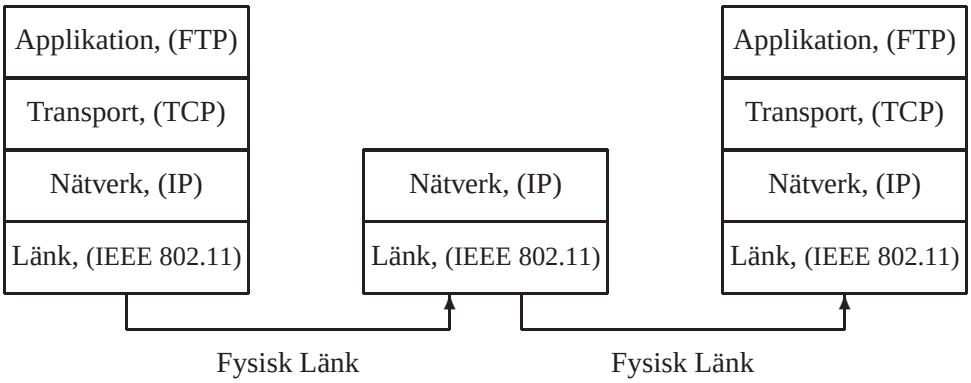
Ad hoc-nätet kan vara mobilt och kan användas i situationer då man snabbt behöver kunna kommunicera inom ett område utan utbyggd fast infrastruktur, t.ex. för att soldaterna i en bataljon ska kunna kommunicera under en operation. Ett annat område där ad hoc-nät kan användas är i civila nödsituationer, t.ex. vid skogsbränder eller jordbävningar då den fasta infrastrukturen är utslagen eller inte finns utbyggd. Ad hoc-nätstekniken kan också användas i cellulära system för att öka täckningsarean, genom att man utnyttjar andra enheter som finns i närheten för att reläa informationen via och nå basstationen.



Figur 1.1: En jämförelse mellan ett cellbaserat nät och ett ad hoc-nät.

1.3 Utmaningar

Det pågår mycket forskning kring ad hoc-nät, och det finns fortfarande problem som måste lösas för att kunna designa ett system som klarar militära krav och önskemål. Då information ska överföras i ett nät krävs det regler för hur data ska sändas, kodas, tas emot och tolkas. Dessa regler kallas för *protokoll*, och vid sändning/mottagning används flera protokoll, med olika ansvarsområden och uppgifter. Protokollen delas in i så kallade lager, och då data ska tas emot eller sändas passeras de olika lagren där informationen behandlas av protokollen på olika sätt, t.ex. delas upp i paket, märks, kodas och sänds. Två av de vanligaste protokollen är TCP (Transmission Control Protocol) och IP (Internet Protocol) som båda används vid dataöverföring över Internet. Dessa protokoll har gett namn åt en referensmodell för protokollsindelning i lager, den så kallade TCP/IP-modellen [1]. Denna innehåller fyra lager med olika ansvarsområden; applikations-, transport-, nätverks- och länklagret. IP används alltid på nätverkslagret och på varje lager används minst ett protokoll. På transportlagret används TCP eller UDP (User Datagram Protocol). Uppsättningen protokoll som används över Internet brukar kallas TCP/IP-stacken, se figur 1.2. I figuren visas hur data som skickas mellan en klient och en server går från applikationslagret och ner på den fysiska länken, och när servern till sist nås vandrar data genom stacken och upp till applikationslagret. Applikationen som används i exemplet är FTP (File Transfer Protocol) och som länkprotokoll används IEEE 802.11. Det finns en router mellan servern och klienten som tar emot paketet på



Figur 1.2: Exempel på TCP/IP-protokollstacken hos en klient, en router (i mitten) och en server.

nätverksnivå och tittar på IP-adressen för att sedan välja hur paketet ska skickas vidare, routas.

- Applikationslagret
  På applikationsnivån används olika protokoll beroende på tillämpning, t.ex. filöverföring eller e-post. Applikationerna ställer krav på överföringen beroende på hur mycket kapacitet de behöver använda. En realtidsapplikation ställer t.ex. hårda krav på maximal fördröjning, vilket då måste hanteras av underliggande lager.

- Transportlagret
  En användare kan samtidigt ha igång flera applikationer. För att skilja mellan dessa används olika portar för de olika applikationerna då de kopplas ut på länken. FTP (File Transfer Protocol) använder t.ex. alltid port nummer 21 [2]. Portarna kontrolleras av transportprotokollen TCP (Transmission Control Protocol) och UDP (User Datagram Protocol). Om TCP används kontrollerar man dessutom att data kommer fram genom att en bekräftelse skickas då ett paket når destinationen, och om paket tappas sänds de om. Om man istället använder UDP som transportprotokoll sker ingen kontroll av om paket kommer fram. Problemet med att använda

TCP i ad hoc-nät är att protokollet är utvecklat för fasta nät och kräver dubbelriktad kommunikation. Då paket tappas tolkas detta som överbelastning i nätet och TCP sänker datatakten, eftersom man i fasta nät har mycket låg bitfelshalt och paket i princip endast tappas vid överbelastning. I ett ad hoc-nät tappas dock många paket p.g.a. den brusiga kanalen och sänkt datatakt löser då inte problemet [3].

- Nätverkslagret
  Hur adressering ska ske i ad hoc-nät är ännu inte helt löst [4], dock används IP-adresser i det fasta nätet och adresserna i ad hoc-nätet måste vara kompatibla med dessa då näten kopplas ihop [5]. En av de största utmaningarna vid utvecklandet av ad hoc-nät är hur trafikstyrning, routing, ska fungera. Hur ska vägar genom nätet hittas för trafiken och hur ska dessa optimeras? Routing handlar om att hitta en väg mellan sändare och mottagare som kan användas för överföring. I det fasta nätet fungerar routingen bra men i ett ad hoc-nät är det ett mer komplext problem eftersom nätet rör sig hela tiden vilket orsakar förändringar i nätets topologi och det saknar dessutom central styrning.

- Länklagret
  På länklagret sker bland annat kodning, modulation och avkodning. Ibland sker även routing här. Olika protokoll används för att fördela tillgången till mediet, (i det här fallet radiokanalen), mellan noderna. Vid dynamisk tilldelning sänder den nod som behöver det, t.ex. med CSMA (Carrier Sense Multiple Access). Man kan ibland först skicka ett kontrollmeddelande så att man vet att ingen annan sänder samtidigt, men mer styrning av vem som ska sända finns inte. Vid statisk tilldelning sänder var och en i sin tur, så att inga kollisioner ska ske. Man kan t.ex. sända med olika frekvenser (FDMA, Frequency Division Multiple Access) eller i olika tidluckor (TDMA, Time Division Multiple Access). Problem i ad hoc-nät är hur man ska styra tilldelningen av kanalen samt hur initiering av schemat för detta ska ske. Man vill utnyttja kanalen så effektivt som möjligt för att öka kapaciteten, t.ex. med hjälp av STDMA (Spatial reuse TDMA) [6].

Säkerheten är alltid viktig i ett taktiskt nät. Fienden ska inte kunna ta sig in i nätet, störa ut kommunikationen eller avlyssna och komma åt viktig information. Nätet ska vara robust och störtåligt.

I det nätverksbaserade försvaret kommer man att ha trafik av olika hög prioritet i näten och det kommer att ställas krav på att trafiken kommer fram, d.v.s. *garanterad tjänstekvalitet*. Detta kallas på engelska *Quality of Service* och akronymen QoS används ofta. De stora problemen med att upprätthålla QoS kommer att finnas i de delar av näten där resurserna är knappa. Dessa delar finns troligen i de trådlösa näten, eftersom man i de fasta näten har betydligt större resurser. Naturligtvis kommer vi aldrig att kunna ge absoluta garantier för att trafiken kommer fram, eftersom oförutsedda saker alltid kan hända, även i ett fast nät. I trådlösa ad hoc-nät kommer osäkerheten att vara större eftersom nätet rör sig och det lätt kan bli avbrott på en förbindelse. Man vill dock kunna ge så bra garantier som möjligt från situation till situation, eftersom detta är nödvändigt i ett militärt kommunikationsnät.

Vi har tidigare studerat QoS i IP-nät [7] och vill gå vidare med att se hur tjänstekvalitet kan garanteras i ad hoc-nät. En viktig faktor i detta är val av rätt väg för informationen genom nätet, d.v.s. routing. Vi väljer därför här att studera routing i ad hoc-nät med inriktning på garanterad tjänstekvalitet, QoS.

1.4 Kvalitetsgaranterande routing

Kraven på att ett mobilt ad hoc-nät ska vara robust, självkonfigurerande och kunna röra sig snabbt gör routing i ett ad hoc-nät till en stor utmaning. Man har dessutom ofta begränsande resurser vad gäller t.ex. bandbredd och effekt. Routingprotokollen måste kunna anpassa sig snabbt efter situationen då topologin ändras. Mycket forskning har skett inom detta område och det finns idag många förslag till routingprotokoll framtagna för trådlös kommunikation [8]. Tyvärr är det endast ett fåtal protokoll som utvecklats för att kunna garantera kvaliteten på trafiken i de trådlösa näten.

Det finns idag inte heller några egentliga kvalitetsgarantier för trafiken som går i det fasta nätet. Olika metoder för att kunna ge garantier finns framtagna och är under utveckling men problemet är att de blir för dyra och komplicerade för att kunna använda överallt [7]. Tanken är att man ska kunna bestämma själv vilken grad av kvalitet man behöver för sin trafik över Internet och betala därefter. Realtidstrafik kan t.ex. få högre prioritet i nätet och därmed minskad fördröjningen. QoS-metoderna för det fasta nätet tar dock ingen hänsyn till routing, eftersom detta inte är ett problem där. I de fasta näten skapas de största

fördröjningarna i olika köer i t.ex. routrar och switchar, och därför är det dessa man koncentrerar sig på.

Routing är en viktig del i ett ad hoc-nät, och påverkar kvaliteten på tjänsterna som används. Tidigare har man fokuserat på att hitta den kortaste vägen till destinationen men om man ska ge några garantier kan det istället handla om att t.ex. välja den mest stabila vägen. Genom att introducera kvalitetsgaranterande routing kan man anpassa vägvalen för trafiken i nätet på ett optimalt sätt och ge resurser då det behövs.

Vissa tjänster i nätet vill man kanske ge högre tjänstekvalitet och vi vill också kunna ge garantier för att trafiken i våra ad hoc-nät kommer fram. För att garantera QoS i ett ad hoc-nät måste flera villkor uppfyllas. QoS routing löser bara en del av problemet, hänsyn måste också tas till applikationer, köer, kanaltilldelning m.m.

1.5 Rapportens struktur

Syftet med den här rapporten är att sammanfatta den utveckling som skett inom trafikstyrning för mobila ad hoc-nät. Vi vill studera om, och i så fall hur väl de olika routingprotokollen kan ge kvalitetsgarantier i näten. Kapitel 1 ger en bakgrund till studien och i kapitel 2 beskrivs grundbegreppen inom routing och QoS. Kapitel 3 ger en överblick av hur kvalitetsgaranti och routing fungerar i ett fast nät. Kapitel 4, 5 och 6 beskriver olika routingprotokoll i mobila ad hoc-nät samt deras QoS-egenskaper. I kapitel 7 kommenterar vi studien och blickar in i framtiden.

Kapitel 2

Introduktion till problemet

2.1 Proaktiv och reaktiv routing

Routingprotokoll brukar ibland delas in i två grupper, proaktiv och reaktiv (kan likställas med klasserna table-driven och on-demand, se [8]). Vid *proaktiv* routing känner vi till vilken väg vi kan skicka paketen till en destination. Informationen finns sparad i tabeller hos noderna, och uppdateras med jämna mellanrum. Det innebär att även om ingen trafik går i nätet, skickas information mellan noderna för att hålla routingprotokollen vid liv. *Reaktiv* routing innebär att noderna söker upp en väg då den behövs. Routingprotokollet reagerar då ett meddelande ska skickas genom att skicka ut en förfrågan i nätet om lämplig väg till destinationen. Det är således en viss fördröjning i början på en uppkoppling om man använder sig av reaktiv routing, men å andra sidan kan nätet vara helt tyst då ingen trafik finns att sända.

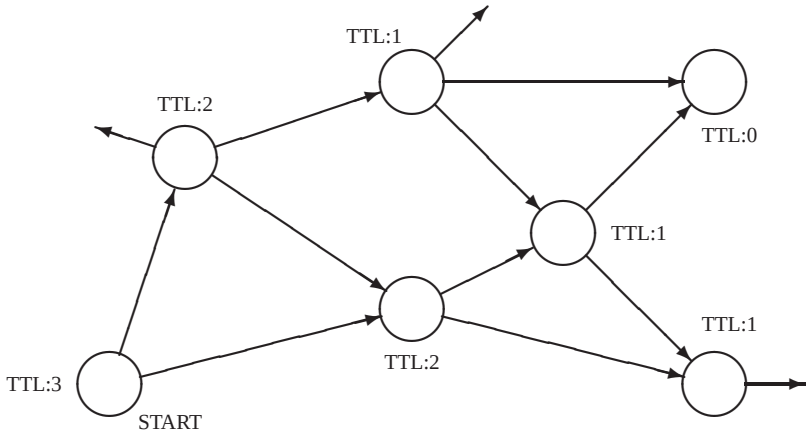
Fördelarna med ett proaktivt protokoll i ett taktiskt mobilt ad hoc-nät är att då ett viktigt meddelande ska sändas finns en väg tillgänglig omedelbart. Det negativa är att protokollet kan slösa kanalresurser då det med jämna mellanrum uppdaterar routingtabellerna, speciellt om nätet rör sig och uppdateringarna måste ske ofta. Om man vill låta radion vara helt tyst kan inte protokollet uppdateras vilket innebär att man måste börja om och uppdatera hela nätet då man vill kunna sända igen. Reaktiva protokoll kan ge en stor fördröjning vid skapandet av nya rutter men kan vara resurssnåla och passar i det avseendet ett ad hoc-nät.

Man kan i ett nät med reaktiv routing lösa problemet med hög fördröjning då

ett viktigt meddelande snabbt måste sändas genom att tillämpa *flooding*, se figur 2.1. Detta innebär att meddelandet skickas ut i nätet till alla noder på samma sätt som en förfrågan om väg normalt görs. För att inte meddelandet ska spridas i ett oändligt antal exemplar sätts ofta en begränsning av antalet noder som paketet får passera innan det slängs. I IP-paket sätts detta i *time-to-live*-fältet (TTL). En nod bör dessutom endast ta emot och skicka vidare samma paket en gång. Vid varje nod minskas antalet kvarvarande tillåtna antal hopp med ett och därefter skickas paketet vidare.

Det finns också kombinationer av reaktiva och proaktiva protokoll, som kallas hybridprotokoll. I dessa tillämpar man t.ex. proaktiv routing till noder inom ett visst avstånd och reaktiv routing till noder längre bort, se kapitel 6.

Hierarkisk routing används i det fasta nätet, se kapitel 3. Man känner här endast till noderna i sitt eget delnät och skickar annars meddelanden uppåt i hierarkin för att finna en väg. Adresserna måste vara hierarkiskt tilldelade, så att det finns ett system att gå efter för att hitta en mottagare, på samma sätt som postgången fungerar. Man måste t.ex. skriva gatu- och postadress på breven i Sverige eftersom det skulle ta för lång tid att hitta mottagaren om man t.ex. skulle skriva personnummer istället, och på samma sätt fungerar det i andra stora nät. I stora ad hoc-nät kan man dela in noderna i delnät med en huvudnod i varje delnät. Huvudnoden sköter all kommunikation utåt med andra delnät.



Figur 2.1: Exempel på flooding i ett ad hoc-nät.

	Proaktiv	**Reaktiv**
Uppdateringar av rutter:	Sker kontinuerligt	Endast då en väg skall användas
Uppkopplingstid:	Snabb uppkoppling	Kan ta lång tid om rutten ej använts förut
Resurskrav:	Kräver resurser även då ingen trafik går i nätet	Resurssnål vid lite trafik

Tabell 2.1: Jämförelse mellan proaktiv och reaktiv routing.

I våra ad hoc-nät vill vi dock undvika svaga punkter och centrala noder, och eftersom hierarkisk routing inte ger samma fördelar i ett litet nät har vi här valt att inte studera detta.

Det är svårt att välja mellan proaktiv och reaktiv routing i ett mobilt taktiskt ad hoc-nät, se tabell 2.1. Om nätet är hårt belastat kan det vara svårt att ge resurser till uppdateringar av det proaktiva protokollet. Å andra sidan är det i de situationer då det går mycket trafik i nätet som det proaktiva protokollet kan vara effektivast, eftersom protokollet inte behöver skicka ut förfrågningar vid varje ny uppkoppling.

Det reaktiva protokollet ger stora fördröjningar då en ny väg ska hittas. Dessutom har det visat sig att dessa vägar ofta inte är lika väl optimerade som de som tagits fram genom proaktiv routing. Reaktiv routing är dock enklare än proaktiv routing och kräver inte samma mängd information från nätet. Det kan i vissa fall vara avgörande.

2.2 Kvalitetsgaranti

Oförutsedda händelser kan alltid inträffa men i den mån det går vill man kunna garantera användare en viss nivå av tjänstekvalitet. Det ligger en del kommersiella intressen i att kunna ge trafiken olika nivåer av kvalitet i ett nät, eftersom det innebär att man kan ta olika betalt för olika god tjänstekvalitet. För att kunna ge någon en högre prioritet i ett nät ställs högre krav på autentisering och säkerhet. Det ska inte vara möjligt att störa ut ett nät genom att en inkräktare ger sig själv högsta prioritet och därmed får alla resurserna. Man måste dessutom ha kontroll

över att resurserna räcker till. I ett militärt nät kan man även tänka sig att man vid prioriteringar kastar en del annan trafik som inte är lika viktig. I kommersiella sammanhang får man prioritet efter priset man betalar, men i militära nät är prioriteringskraven ännu inte specificerade. Det kan t.ex. handla om att vissa applikationer eller särskilda noder prioriteras.

Vad behövs för att tillhandahålla kvalitetsgaranti?

För att kunna ge garanterad tjänstekvalitet (QoS) i ett radionät krävs det att flera olika delar eller funktioner uppfyller kraven på QoS. Enbart med QoS-routing löser vi inte hela problemet eftersom garantierna vi vill ge kan förstöras av andra delar, t.ex. kanaltilldelningsprotokollet. Noderna kanske måste vänta alltför länge innan de får tillgång till kanalen [7].

- Applikationen kan ställa olika krav på kapacitet, fördröjning m.m. Om applikationen är adaptiv kan den användas även om inte kraven uppfylls. En kommunikation mellan nätet och applikationen behövs då för att enas om kapaciteten. Om kraven istället är stränga kan det innebära att om dessa inte uppfylls fungerar inte applikationen alls.

- Routingprotokollet måste kunna ge garantier om att en väg hittas och att den bästa vägen väljs. Detta behöver inte alltid vara den snabbaste, utan kan också vara t.ex. den mest stabila vägen, d.v.s. som inte ändras eller får avbrott. Vi vill också att det vid eventuella avbrott på förbindelserna ska gå snabbt att hitta en ny väg. I bästa fall ska redan en alternativ väg finnas tillgänglig.

- Kanaltilldelningen måste vara sådan att garantier ska vara möjliga att ge. Valet av MAC-protokoll (*Medium Access Control*) påverkar detta. Används CSMA/CA (*Carrier Sense Medium Access/Collision Avoidance*) är det svårt att ge några garantier eftersom protokollet sänder trafik när noden har något att sända. Inträffar kollision sänder man om, vilket innebär att man inte vet hur lång tid det tar för en nod att sända ett meddelande eftersom det beror på trafikbelastningen runt noden. Använder man istället ett MAC-protokoll där resurserna delas upp mellan noderna, t.ex. med TDMA (*Time Division Multiple Access*) och FDMA (*Frequency Division Multiple Access*), kan man lättare garantera en viss kapacitet. Man

vet då hur stor kapaciteten på en länk man har tillgång till, även om denna naturligtvis kan förändras.

- Paketströmmar ska kunna hanteras så att paket med olika prioritet delas upp på olika köer. Tidsbegränsningar måste också finnas för att t.ex. kunna slänga gamla paket. Schemaläggning av paketen måste ske efter olika regler för att möta kraven på olika tjänstekvaliteter.

- Paketen som är märkta för att få en viss prioritet ska delas in i olika klasser. Klassificeringen kan också göras genom att man tittar på innehållet i paketet, vilken applikation som används m.m. Paket som tillhör samma klass får samma behandling vid schemaläggningen. Vilka klasser som finns beror på vad Internetleverantören har avtal om.

- För att man ska kunna garantera en viss QoS-nivå, t.ex. en viss datahastighet på en länk, måste man kontrollera så resursen finns tillgänglig för att sedan kunna reservera denna. En resurstillträdeskontroll måste därför göras innan en bekräftelse på en reservation kan ges.

- Man måste också kontrollera att den som begär QoS också är behörig att få QoS. I ett taktiskt nät är säkerhetsfrågan viktig och användarkontroll måste göras så att inte någon obehörig tar upp kapacitet i nätet.

QoS-protokoll i det fasta nätet

Flera metoder för att tillhandahålla QoS i det fasta nätet har under den senaste tiden utvecklats [7]. Det finns kommersiella intressen i att kunna ta betalt för hög kvalitet på vissa tjänster, och det är detta som drivit på en stor del av utvecklingen. Dock har det visat sig att efterfrågan på QoS i det fasta nätet är låg, kanske på grund av att det saknas applikationer som ställer hårda krav på kapacitet, fördröjning m.m. Kostnaderna för att införa QoS i näten är relativt höga och så länge resurserna i näten är tillräckliga för användarna och deras tjänster finns det inget behov av detta.

I det fasta nätet uppstår de flesta fördröjningarna av paket i olika köer, t.ex. i routrar, switchar eller brandväggar. Man har därför inriktat sig på att lösa detta problem genom att ge vissa paket förtur i köerna eller reservera kapacitet på en förbindelse. Routing och kanaltilldelning är inte något stort problem i det fasta

nätet och därför ger man oftast endast QoS genom köhantering av paket. Trådlösa nät behöver dock kontrollera hur algoritmer för routing och kanaltilldelning fungerar för att inte den totala tjänstekvaliteten ska bli dålig i nätet.

Vad innebär QoS Routing?

Uppgifterna för ett routingprotokoll är att finna en väg och att upprätthålla denna. Vill vi ha en kvalitetsgaranterande routing måste också krav på t.ex. på fördröjning eller kapacitet uppfyllas. QoS Routing innebär enligt [9] två saker:

- Att hitta vägar som kan tillhandahålla de resurser som uppfyller ställda krav.
- Få ett effektivt utnyttjande av hela nätet, ur ett globalt perspektiv.

Den andra punkten är viktig eftersom det innebär att man ska kunna ge QoS men inte till vilket pris som helst. Om det innebär att all annan trafik i nätet får lida oproportionerligt mycket för att en typ av meddelanden ska få högsta prioritet är det kanske inte värt det. Därför måste det globala perspektivet också tas in i beräkningarna. Ett problem här är hur kostnaden i nätet ska mätas. Konsekvenserna för resten av trafiken i nätet då QoS ges till en del av trafiken kommer att variera med omständigheterna vid tillfället, hur nättopologin ser ut, belastningen i nätet etc.

2.3 Förutsättningar

Taktiska radionät har specifika krav vilket ger näten vissa egenskaper. Ibland förutsätts att en positionsförmedlingstjänst finns tillgänglig vilket innebär att vi har mycket god information om de andra noderna. Det är också möjligt att få information om kapaciteten på länkarna och att prediktera rörelse hos enheter. I ett ad hoc-nät vill kunna utnyttja sådan information när det är möjligt, men också kunna klara av situationer då nästan ingen information ges. I vissa lägen vill man dessutom minimera trafikmängden i nätet av hänsyn till risken att upptäckas. Det viktiga är att *information bör utnyttjas då den finns tillgänglig*, för att kunna fatta bättre och säkrare beslut.

- Som MAC-protokoll är det troligt att man kommer att använda ett konfliktfritt protokoll för att möjliggöra garantier, t.ex. STDMA [6]. Används

STDMA kommer man dessutom att behöva känna till information om vilka noder som befinner sig två hopp bort för att kunna beräkna när dessa ska sända. MAC-protokollet blir då relativt komplicerat. Tanken är dock att nätet även ska ha ett dynamiskt protokoll, som är enklare, tillgängligt för vissa situationer. Detta innebär att routingprotokollet genom MAC-protokollet kommer att ha god information om de närmaste grannarna i vissa situationer, dock inte alla.

- I nätet kommer det att finnas heterogena noder vad gäller effekt, applikationer, mobilitet etc. Nodernas egenskaper förändras hela tiden och ger nätet olika utseenden. Man vill kunna variera datatakten på länkarna för att utnyttja nätet bättre. För routingprotokollet innebär detta att om det finns krav på att man ska hitta vägar med en viss kapacitet kan eventuellt noderna anpassa sig efter detta.

- Positionsförmedlingstjänsten gör att alla noders positioner kan vara kända. Genom tjänsten kan man också prediktera rörelser i nätet då positionerna uppdateras eller genom att en hastighetsvektor skickas med i positionsmeddelandena. Troligen kommer man att ha mer exakt information om noder i närheten medan information om noder långt bort i nätet inte uppdateras lika ofta. Routingprotokollet kan utnyttja den här informationen för att få en bild av nätet och därmed hitta vägar genom proaktiv routing.

- De ad hoc-nät vi studerar här har en begränsad storlek, exempelvis som en mekaniserad bataljon som består t.ex. av ungefär 200 noder. I så små nät kan vi bortse från hierarkisk routing.

- Routingprotokollen måste klara av multicast eftersom det är ett smidigt sätt att kommunicera ut till en särskild grupp. Detta innebär att vi måste välja ett routingprotokoll som stödjer detta.

När vi utvärderar de olika protokollen vill vi studera hur väl given information kan utnyttjas. Om vi känner till detaljerad information om våra närmaste grannar vill vi kunna utnyttja den för att t.ex. kunna beräkna vilka kapaciteter länkarna har. Detta behövs då man ska ge garantier för QoS. Används positionsförmedlingstjänsten är proaktiv routing lämplig, och bör kunna tillämpas.

Kapitel 3

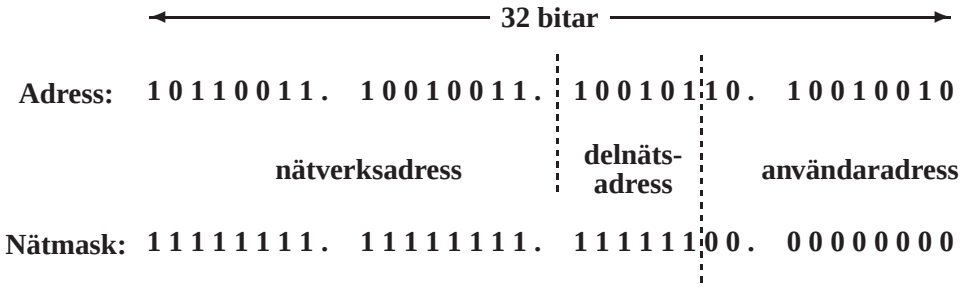
Routing i det fasta nätet

3.1 Introduktion

Routing i det fasta nätet skiljer sig en del från routing i ett mobilt ad hoc-nät. Detta beror bl.a. på nätens storlek, mobiliteten samt stabiliteten på länkarna. Vi har valt att även ta upp routing i fasta nät i den här rapporten för att kunna förklara skillnaderna jämfört med routing i ad hoc-nät och också för att ge en bakgrund till varför QoS måste ges på andra sätt i ad hoc-nät än i fasta nät. I kapitel 3.2 och 3.3. går vi igenom tekniken för routing i fasta nät. Kapitel 3.4 tar upp QoS i fasta nät (se även kapitel 2.2) och i kapitel 3.5 studerar vi skillnaderna mellan routing i det fasta nätet och i ad hoc-nät.

3.2 Adressering

Innan vi går in på vilka routingprotokoll som används i det fasta nätet ska vi studera hur adressering sker över Internet eftersom det påverkar routingen. I det fasta nätet (Internet t.ex.) baserar man sig på IP-teknologi (Internet Protocol) [10]. Detta innebär att IP används som nätverksprotokoll och trafiken sänds som IP-paket. Används IP version 4 tilldelas varje nod en IP-adress som består av 32 bitar. I den här rapporten är det IPv4 som vi använder. Adresserna är hierarkiskt ordnade och den första delen av adressen är en nätverksadress, därefter kommer delnätsadressen och till sist användaradressen. Ett delnät kan ha varierande antal användare beroende på behovet och kraven hos ägaren av nätet. Det påverkar i



Figur 3.1: Exempel på adressering i ett IP-nät

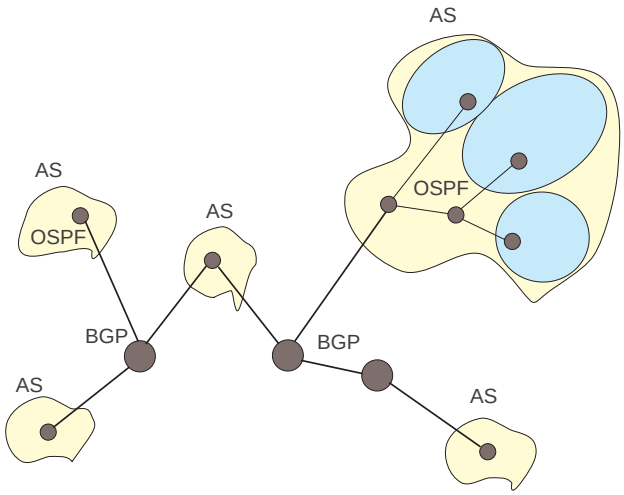
sin tur hur stor del av adressfältet som var och en av de tre adressdelarna upptar.

Nätverksadressens storlek beror på vilken typ av adress vi har, en så kallad A, B, C eller D-adress. Detta anges med de två första bitarna i adressen. Man håller dock på att överge sättet att märka adresserna på detta sätt eftersom det ändå är skiljelinjen mellan delnätsadressen och användaradressen som är viktig. För att hitta gränsen mellan användaradressen och delnätsadressen används en så kallad nätmask. I exemplet nedan vet vi att vi har en B-adress, eftersom adressen börjar med 10, se figur 3.1. B-adresser har alltid en nätverksadress som består av de 16 första bitarna, hälften av adressen. Nätmasken i det här fallet består av 22 ettor (se figur 3.1) och då vet vi att delnätets adress är de närmaste 6 bitarna efter nätverksadressen. Vi kan således ha 64 delnät, eftersom $2^6 = 64$. De sista 10 bitarna är användaradresser. I varje delnät i exemplet kan det alltså finnas 1024 användare, eftersom $2^{10} = 1024$.

IP-adresserna används i alla noderna då paket ska sändas och destinations- och mottagaradressen skickas med. Nätmasken finns ute i nätet i routrar och bestämmer hur stora delnäten är och genom att adresserna är hierarkiskt indelade kan routing ske. [10]

3.3 Routing

Internet kan ses som ett stort nätverk uppbyggt av många små nätverk, som fungerar som *autonoma system* (AS), se figur 3.2 [10]. Varje autonomt system



Figur 3.2: Routing inom autonoma system (AS) med hjälp av OSPF. Ett av de autonoma systemen är uppdelat i tre områden.

har en egen organisation och kan använda sin egen routingalgoritm inne i sina nät. Mellan de olika nätverken måste däremot vissa standarder gälla och vissa algoritmer användas. Även routing inom ett autonomt system förenklas dock om man sätter vissa regler som ska gälla i dessa system.

3.3.1 Routing inom autonoma system

Inom ett autonomt system är *Open Shortest Path First* (OSPF) [10], *Intermediate System to Intermediate System Protocol* (IS-IS) [10], och *Routing Information Protocol* (RIP) [1] vanliga routingprotokoll som används. Idag är OSPF det vanligaste på Internet. När OSPF började tas fram 1988 satte man flera krav på vad routingprotokollet skulle uppfylla:

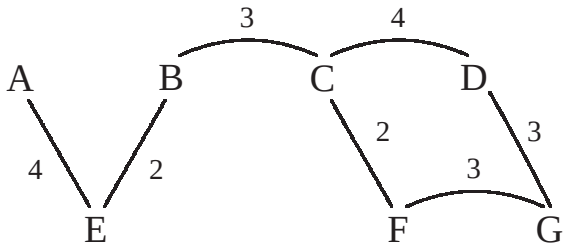
- Det skulle bygga på en öppen standard för att kunna få spridning. Alla skulle kunna få tag på koden t.ex. vid tillverkning av routrar.
- Det skulle klara av att mäta avstånd på olika sätt, t.ex. genom fördröjning

eller fysiskt avstånd.

- Algoritmen skulle vara dynamisk för att snabbt kunna anpassa sig efter förändringar i topologi.
- Routing med stöd för olika slags tjänstekvaliteter - ”type of service” - skulle finnas. Realtidstrafik skulle kunna märkas och få bättre hantering än annan trafik. Detta var möjligt redan på 80-talet, men tjänsten har hittills aldrig använts och nu försvinner fältet för märkning av kvalitetskrav och ersätts av andra lösningar i IP version 6.
- Protokollet skulle klara av att fördela trafiklasten i nätet över flera vägar. Alla paket skulle inte gå via den bästa vägen eftersom det oftast ger bättre prestanda att sprida ut trafiken på flera olika vägar som inte används annars.
- Hierarkiska system måste kunna stödjas. Internet hade redan då OSPF togs fram vuxit och blivit för stort för att någon enskild router skulle kunna ha överblick över hela nätet.
- Säkerheten skulle vara så stark att man inte skulle kunna påverka routingen och därigenom manipulera trafiken i nätet.

1990 var den första standarden för OSPF klar, och den har därefter utvecklats. Protokollet fungerar så att det använder grafer för att illustrera hur ett nät är uppbyggt, vilka routrar och länkar som finns, se figur 3.3. Varje länk kopplas till en kostnad och när en väg sedan ska tas fram viktas de olika möjliga vägarna och den med totalt lägst kostnad väljs. Kostnaden kan vara antal noder som passeras, total fördröjning m.m.

OSPF stödjer hierarkier inom ett autonomt system genom att använda sig av ett ”stamnät” som sköter informationen mellan olika områden inom systemet. Många av de autonoma systemen är så stora att routrarna inte längre kan känna till hela det autonoma systemets struktur. OSPF låter då systemen delas upp i olika områden, bestående av ett eller flera sammansatta nätverk. Varje autonomt system har ett stamnät (eng. backbone area), som alla områdena är kopplade till, se figur 3.2. Alla routrar som är kopplade till två eller fler områden tillhör stamnätet. Alla områden, inklusive stamnätet, fungerar så att topologin hos noderna endast är känd inom just det området. Alla routrar inom ett område har



Figur 3.3: Exempel hur OSPF grafiskt representerar länkar med olika vikter i varje nod. Vägen mellan A och G väljs som A-E-B-C-F-G eftersom lägst kostnad då uppnås.

samma graf av tillstånd och använder samma algoritm för att beräkna kortaste vägen till mottagaren. En router som tillhör stamnätet har flera tillståndsgrafer som används beroende på till vilket nät ett paket skall skickas. Om ett paket ska skickas inom ett område bestäms vägen hos den första routern eftersom kortaste väg kan beräknas. Ett paket som ska skickas mellan två olika områden måste däremot först routas inom sitt område för att komma ut på stamnätet, och därefter ska mottagarens område hittas och paketet måste routas även där.

För att varje router ska kunna ha en tillståndsgraf över alla områden som den tillhör måste information om vilka noder som hör ihop samt vikt på länkarna skickas ut. En router som kommer till ett nät måste säga till att han finns genom att skicka ett HELLO-meddelande till alla inom området. OSPF låter inte routrarna utbyta information med alla grannar i nätet eftersom detta kan bli ineffektivt. Istället väljs (minst) en router i området som sköter utbytet av information och som de andra routrarna med jämna mellanrum skickar information (LINK STATE UPDATE) om sitt tillstånd till. Meddelandena bekräftas för att göra dem pålitliga och märks med sekvensnummer för att man ska använda den senaste informationen. Vid förändringar i nätet skickas också meddelanden med dessa uppdateringar.

OSPF har utvecklats för att kunna ge QoS genom att sätta olika krav på vägarna i nätet [11]. En vägvalsalgoritm används som baserar sig på en del in-

formation om nätet. Man känner till tillgänglig bandbredd samt utbredningsfördröjningar på varje länk. Denna information används för att hitta vägar som uppfyller krav på en viss kapacitet och för att kunna identifiera långsamma länkar, t.ex. satellitlänkar. För varje väg anges också avståndet i antal hopp.

Alla routrar i nätet har tabeller med information om länkstatus och vägar i nätet, och dessa uppdateras med jämna mellanrum. Det är i vissa fall ineffektivt med periodiska uppdateringar och man kan istället göra så att en router skickar uppdateringsmeddelande endast om det skett en signifikant ändring i tabellerna och om det gått en viss tid sedan senaste uppdateringen. Detta minskar trafikbelastningen i nätet.

Ett förslag för att kunna ge garanterad tjänstekvalitet är att använda reservationsprotokollet *Resource Reservation Protocol* (RSVP) för att reservera en väg. En förfrågan skickas då med ett krav på vad vägen ska klara. Vägsökningsalgoritmen tittar då på möjliga vägar (givet ett maximalt antal hopp) och beräknar kapaciteten som var och en av vägarna klarar - d.v.s. det minsta värde på länkkapacitet som finns angivet längs en väg. När en väg hittas som uppfyller de ställda kraven väljs den vägen och reserveras.

3.3.2 Routing mellan autonoma system

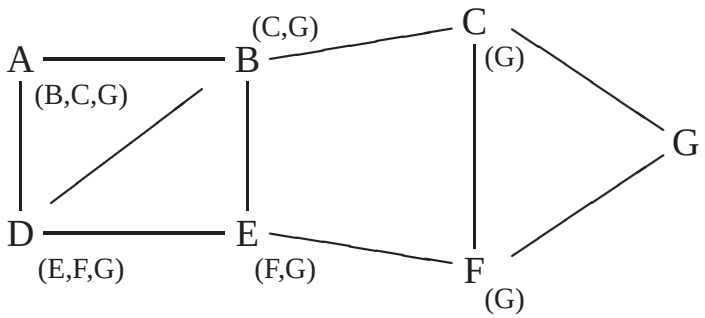
Mellan de olika autonoma systemen används protokollet *Border Gateway Protocol* (BGP) [10]. Det förekommer också andra protokoll, men BGP är i särklass det vanligaste och rekommenderas. Anledningen till att ett annat protokoll används här är att routingprotokollen mellan respektive inom autonoma system har olika mål. Inom ett autonomt system ska protokollet bara skicka paketet mellan sändare och mottagare på ett så effektivt sätt som möjligt. Det behöver inte ta hänsyn till olika Internetleverantörer och ”politik”. BGP tar däremot hänsyn till att man ibland inte kan ta den kortaste vägen mellan sändare och mottagare genom ett nät eftersom det t.ex. inte vill transportera trafik mellan andra nät, eller att man ibland kan använda ett nät enbart om man betalar för sig. Det brukar ofta handla om politik, säkerhet och ekonomiska aspekter. Restriktionerna hänger dock inte ihop med själva routingmekanismen i BGP eftersom dessa ska kunna variera och styras från varje autonomt system. Exempel på restriktioner när det gäller routing kan vara [10]:

- Ett autonomt system låter ingen trafik passera.

- Enligt vissa politiska beslut får inte trafik gå överallt, trafik från Pentagon får t.ex. inte passera Irak på sin väg någon annanstans.

- Trafik får passera om ni betalar för tjänsten.

För BGP består nätet av autonoma system med gränsroutrar samt länkar mellan dessa, inget annat. BGP delar in nät i tre kategorier: nät som endast har en ingång och därför inte kan användas för transport (*eng. stub networks*), nät som har flera kopplingar utåt men som inte får användas (*eng. multiconnected networks*) och nät som kan användas för genomfartstrafik (*eng. transit networks*). Restriktioner längs en länk ger en hög kostnad i routingprotokollet, för att man i möjligaste mån ska välja en annan väg. BGP är i grunden ett distansvektorprotokoll, men använder inte bara en kostnad baserad på avståndet till målet, utan väger också in hur restriktioner påverkar. BGP håller också koll på exakt vilka noder som passeras längs vägen, något som distansvektorprotokoll normalt inte gör. Det gör att BGP lättare fattar ”kloka beslut” om nya vägar, se exempel 3.4. I figuren håller alla noderna ordning på vilka noder som passeras på vägen till mottagaren G. Anta att nod B tar vägen B-C-G för att komma till nod G. Vid utbyte av information med de andra noderna framgår exakt vilken väg de också använder till nod G. Om nod C försvinner och nod B måste byta väg kan B se att vägen förbi A är ointressant eftersom den passerar genom B själv (vilket inte alla routingprotokoll klarar av). Valet står då mellan att ta vägen via D eller E och genom att beräkna distansen till mottagaren väljs vägen B-E-F-G.



Figur 3.4: Exempel på hur BGP fungerar.

3.4 QoS i fasta nät

I det fasta nätet har det sedan länge funnits möjligheter att ge QoS för trafiken. Genom att i IPv4-paketens huvuden sätta fältet ”Type of Service” skulle man ge olika typer att trafik olika hantering i routrarna. Men eftersom inte fältet används försvinner det bort i IPv6 och ersätts istället av en mängd nya förslag på utvecklingar av klassificering, märkning och hantering. Man har i det fasta nätet i huvudsak två inriktningar på metoderna för att ge QoS.

- Man kan reservera plats längs en väg för ett visst flöde. Man är då garanterad en viss kapacitet.

- Man låter paketen märkas med en kod som delar in paketen i olika klasser. Beroende på vilken klass man tillhör går trafiken olika snabbt. Detta kan jämföras med den vanliga posten där man kan skicka brev märkta med första klass, ekonomi eller express.

De tre mest stabila och omdiskuterade protokollen idag är *Integrated Services* (IntServ), *Differentiated Services* (DiffServ) och *Multi-Protocol Label Switching* (MPLS) [7]. IntServ reserverar kapacitet längs vägen med hjälp av protokollet *Resource Reservation Protocol* (RSVP). DiffServ och MPLS tillhör istället den andra inriktningen och skiljer mellan paketen och ger dem olika hög prioritet.

IntServ reserverar kapacitet längs vägen för varje enskilt flöde vilket ger goda garantier men inte fungerar så bra i ett hårt belastat nät eftersom det inte är skalbart. Om något förändras i nätet måste reservationerna uppdateras och det kan bli en i sig resurskrävande process.

DiffServ skiljer istället mellan olika klasser av paket och ger dem olika hög prioritet i nätet. Det bygger på att nätet först delats in i domäner, där man på gränsen till varje domän klassificerar och märker paketen med en s.k. DiffServ-kod och ger paketen olika prioritet i routrarna. Klassificeringen kan t.ex. baseras på innehåll, adresser eller tidigare märkning. Inga större problem uppstår här om trafiklasten ökar i nätet eftersom DiffServ är skalbart och inte reserverar kapacitet per flöde utan för en grupp av flöden. MPLS liknar DiffServ genom att inom en domän snabbt kunna märka, klassificera och skicka paketen vidare.

3.5 Skillnader mellan routing i fasta nät och ad hoc-nät

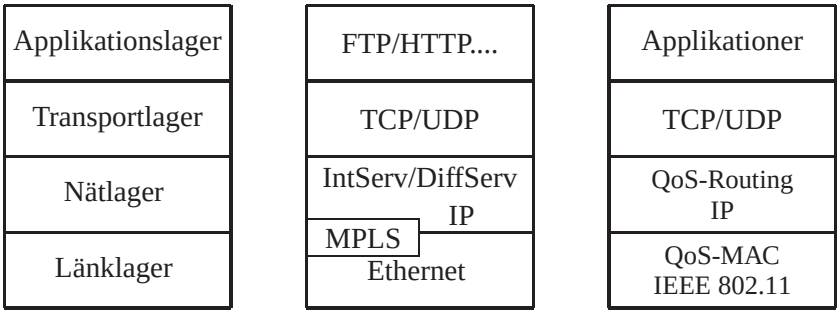
Medan man i det fasta nätet använt proaktiv routing under en lång tid har man i ad hoc-nät gett reaktiv routing stort utrymme. Det kan förklaras med att man i det fasta nätet har en topologi som inte förändras särskilt snabbt och det är därför effektivt att spara information om andra noder. I ad hoc-nät däremot kan det vara svårt att ha kontroll över nätets topologi och reaktiv routing har därför fått mycket uppmärksamhet.

Eftersom vägen för ett meddelande i ett ad hoc-nät ofta väljs av hur snabbt en förfrågan kommer tillbaka kan det i värsta fall innebära att den snabbaste vägen också är den mest lågkapacitiva vägen. Man har alltså många gånger inte kontroll över hur hög kapacitet olika vägar klarar vilket ibland kan vara till problem. I det fasta nätet drabbas man sällan av detta, men i radionät är resurserna mindre och variationerna mellan olika länkar stora och det kan snabbt ske stora förändringar. Valet av vägar blir därför mer avgörande och det är viktigt att vi tar fram rutter som kan ge begärd kapacitet.

Det finns protokoll som ger stöd för QoS i det fasta nätet, se kapitel 3.4. **Problemen** med att använda dessa protokoll i ad hoc-nät är flera, eftersom de har utvecklats för fasta nät. Problemet med IntServ är att reservationerna av resurser måste uppdateras om något sker i nätet, vilket i ett mobilt ad hoc-nät kan ge mycket extra trafik och avbrott i informationsöverföringen, eftersom nätet kan ändra topologi och egenskaper hela tiden. DiffServ och MPLS kräver att man har definierade domäner i nätet och känner till ytternoderna till dessa. Problemet i ett mobilt ad hoc-nät är hur en domän ska kunna bestämmas och avgränsas samt hur gränserna ska kunna ändras. Ytternoderna ska klassificera och märka paketen samt ta bort märkningen då paketet lämnar domänen, och måste därför vara medvetna om sin roll. Man måste dessutom inom en domän kunna enas om ett system för klassificering, vilket kan ge problem eftersom detta måste kunna ske självkonfigurerande.

Inget av protokollen ställer några krav på det underliggande routingprotokollet. Det innebär att man totalt sett kan få en dålig tjänstekvalitet även om något av protokollen ovan används, även om problemen med detta är små i fasta nät. I figur 3.5 kan man se var dessa protokoll verkar relativt TCP/IP-stacken och var man framförallt i trådlösa nät också vill ge QoS.

Det finns dock förslag på hur routingprotokoll i fasta nät kan anpassas för att uppfylla ställda krav på QoS. Detta gäller OSPF, som man redan 1999 i en



Figur 3.5: Till vänster kan man se TCP/IP-stackens uppbyggnad. I mitten visas var metoder för att kunna garantera QoS för fasta nät placeras in och till höger visas var QoS kan ges i trådlösa nät

RFC [11] föreslog en utvidgning av för att kunna stödja QoS.

Adressering är också ett problem i ad hoc-nät. Nätet ska kunna vara helt autonomt men också kunna kopplas samman med ett fast nät [5]. IP-adresser kommer troligen att användas även i ad hoc-nät men man kommer inte att kunna ha dem hierarkiskt indelade. Det gör att routingalgoritmerna skiljer sig åt även på detta sätt.

Kapitel 4

Proaktiva routingprotokoll för mobila ad hoc-nät

4.1 Inledning

Med ett proaktivt protokoll känner vi alltid till vilken väg ett paket kan skickas. Detta innebär att fördröjningarna med denna typ av protokoll är små vid starten på en ny uppkoppling, till skillnad från med de reaktiva protokollen. Vi har i det här kapitlet valt att studera länkstatusprotokollet *Optimized Link State Routing* (OLSR) [12] samt det vektorbaserade protokollet *Destination-Sequenced Distance-Vector* (DSDV) [8] för att ge en bild av hur de proaktiva protokollen fungerar. Sedan gör vi också en något djupare studie av protokollet *Fisheye State Routing* (FSR) [13]. Till sist sammanfattar vi hur protokollen kan uppfylla QoS-krav och hur väl övrig information i nätet kan utnyttjas.

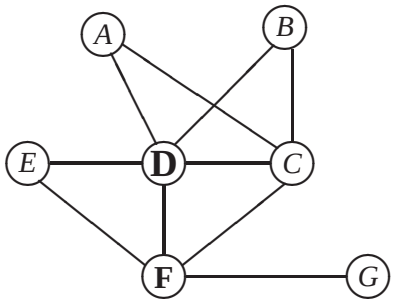
4.2 Optimized Link State Routing

OLSR [12, 14]- *Optimized Link State Routing* - är ett proaktivt routingprotokoll utvecklat för ad hoc-nät. Med jämna mellanrum utbyter noderna i nätet information med varandra och uppdaterar därmed sina routingprotokoll. Det speciella med OLSR är att man väljer ut vissa noder som ska reläa trafik, s.k. *Multi-Point Relays*, MPR. Genom att introducera dessa noder kan man minska den totala trafiken för uppdateringar, kontroller av status på länkar m.m. MPR-noderna

väljs av sina grannar, så att varje nod har minst en MPR inom räckhåll. En vald nod skickar med jämna mellanrum ut ett kontrollmeddelande till sina grannar och påminner om att den är MPR. För att inte onödig informationen ska sändas i nätet är det endast de noder som tillhör MPR-noden som skickar vidare informationen från denna. Det är viktigt att den valda noden har symmetriska länkar så att informationen kan skickas samma väg till och från noden.

För att välja en MPR utgår varje nod från sin position i nätverket, se figur 4.1. MPR väljs av en nod som den granne som täcker in flest tvåhoppsgrannar. Det innebär att man alltid försöker välja en nod med många länkar som MPR.

Grunden för vägsökning i OLSR är att varje enskild nod använder sin egen tillgängliga information för att välja kortaste väg till destinationen. Då en vanlig nod vill sända ett paket behöver man inte söka efter vägar utan den skickar sina meddelanden till sin MPR som i sin tur har en routingtabell med vägbeskrivning. En MPR kan välja att vid uppdatering av tabellen endast rapportera om noder i sin närhet till andra MPR, för att ytterligare minska den överflödiga informationen i nätet. OLSR är särskilt lämpligt i stora nät med hård belastning eftersom det är då de största förtjänsterna kan göras.



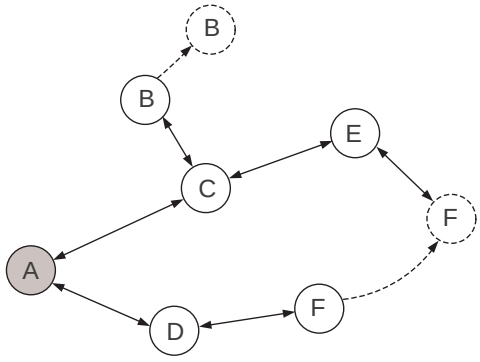
Figur 4.1: Val av MPR-nod i OLSR: nod F ska välja MPR, och vill välja en nod som kan maximera antal grannar två hopp bort. Nod D och C täcker båda in lika många noder för nod F på avstånd två hopp (nod A och nod B), och nod F väljer då nod D som MPR eftersom denna nod har flest grannar totalt.

4.3 Destination-Sequenced Distance Vector Routing

Används DSDV-routing bygger det på att alla noder har en routingtabell som innehåller möjliga destinationer, antalet hopp för att nå dit, första noden att sända via samt ett sekvensnummer som genereras av destinationsnoden [8, 12]. DSDV är en utveckling av protokollet *Routing Information Protocol* (RIP) [1] som tidigare användes mycket i det fasta nätet men har vissa problem med att loopar lätt uppstår. Loopar undviks i DSDV-algoritmen genom att sekvensnummer används för att märka informationen så att man alltid vet att den som används är den senaste.

Noderna uppdaterar med jämna mellanrum sina routingtabeller genom att utbyta information med sina grannar. En nod skickar också ut information om en signifikant ändring i routingtabellen skett sedan senaste uppdateringen. I exemplet i figur 4.2 rör sig nod F så att länken med nod D bryts och en ny länk bildas till nod E. Nod E skickar då ut ett meddelande om den nya länken.

Hela routingtabellen skickas periodiskt ut till grannarna men man kan också välja att inte uppdatera hela routingtabellen utan bara sända ut information om de delar där en ändring skett. Detta görs för att inte onödigt mycket trafik ska behöva sändas, t.ex. i ett relativt stabilt nät. Om nätet istället ändras mycket



Figur 4.2: Exempel på rörelse mellan noder. Nod B rör sig bort ur nätet och förlorar kontakten med nod C samtidigt som nod F byter granne från nod D till nod E.

kan det vara effektivare att skicka ut hela routingtabellerna. Om man tar emot flera paket med routinginformation om samma nod (från flera sändare), väljs den nyaste informationen, baserat på sekvensnumret, för att spara i den egna routingtabell. Om även sekvensnumren är samma, väljs den informationen som ger kortast väg istället.

Meddelandet som nod E skickar ut i exempel 4.2 innehåller då ett nyare sekvensnummer för länken än tidigare. Ett uppdateringsmeddelande går först ut till grannarna ett hopp bort. De tar emot paketet och uppdaterar sina routingtabeller och skickar sedan vidare uppdateringsmeddelandet. Nod B rör sig dessutom bort från nätet och tappar kontakten med nod C. Detta märker nod C genom att meddelanden från nod B uteblir och sätter därför antalet hopp till oändligheten för att nå nod B, tills ny väg hittas. Routingtabellen för nod A innan uppdateringen sker visas i tabell 4.1. Denna visar möjliga destinationer, nästa nod som ska passeras för att nå dit, antalet hopp till destinationen samt sekvensnummer för varje uppgift om en nod. Varje nod har en egen sekvensnummerserie. Nod C har nu uppdaterats om förändringarna och skickar vidare dessa i nätet. Ett routingmeddelande, se tabell 4.2, når nod A och meddelandet jämförs då med den befintliga routingtabellen och förändringar införs. Den slutliga tabellen hos nod A visas i tabell 4.3 för exemplet.

destination	**nästa hopp**	**antal hopp**	**sekvensnummer**
A	A	0	121
B	C	2	232
C	C	1	123
D	D	1	456
E	C	2	347
F	D	2	400

Tabell 4.1: Routingtabellen hos nod A innan uppdatering skett.

DSDV fungerar bra i ad hoc-nät där inte för mycket trafik genereras, d.v.s. inte för stora nät eller nät med alltför mycket topologiförändringar. Det kan då uppstå för mycket routingtrafik i nätet som blir alltför belastat. Protokollet utvecklades relativt tidigt och det finns många andra routingprotokoll som utvecklats från DSDV. Försök att utveckla protokollet för att ge QoS har också gjorts [15].

destination	nästa hopp	antal hopp	sekvensnummer
B	B	∞	**233**
C	C	0	123
D	A	2	456
F	**E**	**2**	**401**

Tabell 4.2: Del av routingtabellen hos nod C efter förändringarna i bild 4.2, som nod C skickar ut till sina grannar för att utbyta information. Hela routingtabellen skickas således inte ut denna gång. Notera att vägen till nod B och nod F är uppdaterade, och sekvensnumren ändrade.

destination	nästa hopp	antal hopp	sekvensnummer
A	A	0	121
B	**C**	∞	**233**
C	C	1	123
D	D	1	456
E	C	2	347
F	**C**	**3**	**401**

Tabell 4.3: A uppdaterar sitt routingprotokoll efter informationen som mottagits från nod C. Eftersom sekvensnumret är högre för vägen till destinationen B inför A denna ändring i sin egen routingtabell, och på samma sätt för destination F.

4.4 Fisheye State Routing

Fisheye State Routing (FSR) [13, 16] används i ad hoc-nät för att reducera kontrolltrafiken i stora nät. FSR är ett proaktivt routingprotokoll som uppdateras kontinuerligt. Det är ett tabellstyrt protokoll, där varje nod har information om alla andra noder och känner till en väg dit.

I stora nät går det åt mycket trafik i nätet för att uppdatera protokollen då förändringar sker. Det innebär att annan ”nyttotrafik” blir lidande eftersom mindre resurser då finns kvar till denna. Samtidigt är kontrolltrafiken i nätet är nödvändig för att routingen ska fungera och det innebär att den måste sändas och ges plats.

För att minska kontrolltrafiken utnyttjar FSR det faktum att man oftast be-

höver säkrare information om noder nära sig själv än noder långt borta. Uppdateringar av routingtabeller skickas därför oftare ut med information om noderna nära sig själv än om dem som befinner sig mycket långt borta. Detta gör att man oftare tar emot information om sina närmaste grannar och informationen om noder långt borta sprids långsammare. Man har således en bra uppfattning om noder nära men noder långt bort får man information om så sällan att de kan ha förflyttat sig eller fått ändrade förutsättningar sedan senast. När ett meddelande ska skickas till en nod långt borta väljer sändaren mest ”rätt riktning”, ju närmare destinationen man kommer, desto mer precis lägesinformation finns att tillgå i reläande noder. Fisheye State Routing har visat sig effektivt i stora nät genom att kontrolltrafiken kan minskas.

Algoritmen

FSR är ett länkstatus protokoll. Noderna i FSR har information om status på alla länkar i nätet, men det som är speciellt för FSR är sättet som informationen och uppdateringarna sprids i nätet.

”Fiskögetekniken” (eng. *fisheye*) innebär att en nod har god kännedom om länkarna närmast men noggrannheten minskar med avståndet. Detta liknas vid ett fisköga som är skarpt nära fokus, men minskar i skärpa ju längre bort från fokus man går (”Fisköga” är för övrigt även en populär kameralins).

Varje nod måste ha information om nätet sparat i tabeller och listor. Man har en lista över de närmaste grannarna med en notering om när man senast tog emot ett meddelande från varje granne. Dessutom har varje nod två tabeller, en tabell som sparar information om topologin i nätet, vilka noder som finns och deras respektive grannar, samt en tabell med routinginformation, d.v.s. vägar till noderna i nätet.

Gruppen av länkar närmast den egna noden får man den säkraste informationen om eftersom den uppdateras kontinuerligt. Nästa grupp får man mer sällan statusinformation om och när man väljer väg så är denna information inte fullt lika tillförlitlig. En nod som ska sända ut statusinformation tittar igenom sin topologitabell och skickar med information om näraliggande noder ofta och noder längre bort mer sällan. Informationen i topologitabellen består av möjliga noder och deras grannar. Utifrån denna information räknar sedan den enskilde noden ut sin routingtabell.

I exemplet i tabell 4.4 visas hur man väljer vilka noder som man ska skic-

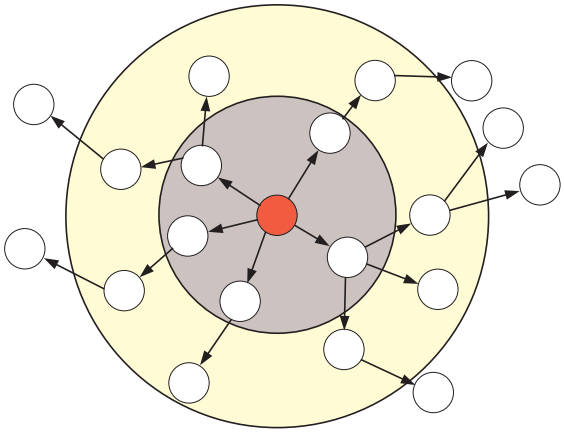
destination	nästa hopp	avstånd	sändningsintervall
B	B	1	1 s
C	B	2	2 s
D	D	1	1 s
E	B	3	4 s
F	B	4	8 s

Tabell 4.4: Exempel på en del av routingtabellen hos en nod som använder FSR. Hur ofta informationen om en nod ska sändas följer i det här exemplet formeln $2^{(\text{avstånd}-1)}$.

ka med i routingmeddelandet. Information om nod B och D skickas ut ofta, varje sekund eftersom det är de närmaste noderna, medan t.ex. information om nod F endast skickas ut var 8:e sekund. Informationen skickas endast till grannoderna som i sin tur uppdaterar sin tabell och med jämna mellanrum skickar ut information på samma sätt. På så vis propagerar informationen genom nätet och information om noder i närheten skickas oftare vidare. Hur snabbt informationen distribueras i nätet är svårt att uppskatta eftersom det beror på hur lång tid det är tills grannnoden i sin tur skickar vidare informationen. Den skickas alltså inte direkt vidare utan följer schemat, se t.ex. tabell 4.4.

Antalet nivåer som man delar in noderna i kan variera och storleken på grupperna likaså. En grupp - eller ett område med samma uppdateringsfrekvens har en radie som uttrycks i antal hopp. I exemplet i figur 4.3 är den närmaste gruppen inom två hopp, och där uppdateras noden om lägesförändringarna ofta. I nästa område sker uppdateringar mer sällan. Intervallen, antal nivåer och storleken på grupperna väljs beroende på stabiliteten hos nätet och de krav som ställs. Ett nät som är mycket mobilt och där stora förändringar sker måste uppdatera routingprotokollet tillräckligt ofta för att informationen ska nå fram. En väg till en destination många hopp bort kan således vara osäker och kanske inte den närmaste, men ju närmare målet man kommer desto bättre information får man och man kan därmed förbättra vägvalet.

Sekvensnummer på vägarna används för att man vid uppdateringar alltid ska spara den senaste versionen, på samma sätt som t.ex. DSDV (kap 4.3) fungerar. Hos varje nod sparas till sist en topologikarta över hela nätet och med hjälp av denna kan man beräkna den kortaste vägen till alla destinationer.



Figur 4.3: FSR fungerar så att man har god kännedom om noderna närmast, och denna minskar sedan med avståndet.

FSR minskar routingtrafiken i nätet genom att inte skicka med hela topologitabellen vid varje uppdatering utan bara information om noderna i den ”grupp” som är aktuell. För att ytterligare minska trafiklasten i nätet skickar FSR endast ut information som nätet behöver. Information om förändringar i nätet skickas alltid ut och om en nod märker att andra delar av nätet inte har rätt uppdaterad information sänds även denna.

När FSR tar emot information om förändring sätts en flagga vid de poster i topologitabellen som ändras. Flaggan sätts också om noden märker att nätet behöver uppdaterad information om någon nod. Dessutom sätts flaggan vid nodens egen post periodiskt för att t.ex. nya noder ska kunna få information om nätets topologi utan att förändringar sker. En räknare ser till att man vid vissa tider tittar igenom topologitabellen och sänder ut information om de noder som har flaggan satt **samt** ingår i gruppen som är aktuell [13, 16].

Protokollet fungerar oberoende av paketens IP-format och kan implementeras på applikations- eller nätverkslagernivån [17].

4.5 QoS-garantier och kommentarer

De proaktiva protokollen har redan mycket information om nätet, och genom att utöka denna information kan man lägga till krav på QoS. Metoder som studerats är att låta noderna ha kännedom om effekt, stabilitet, fördröjning, positioner m.m. Genom att sedan prediktera hur noderna kommer att röra sig och därmed hur förbindelserna kommer att påverkas kan en bra väg hittas [18]. Detta kan göras genom att noderna skickar med en hastighetsvektor tillsammans med sin position till de andra noderna som då kan beräkna nodens nästa position förutsatt att riktning och hastighet följs. Man kan därigenom förutse när en förbindelse kommer att brytas och man kan välja en väg som kan ge den kapacitet som krävs. I [18] görs dock inga reservationer av kapaciteten men en kontroll av om resurserna finns kan göras. Genom att använda sig av källrouting - då varje nod som ska passeras finns specificerad i varje paket - kan man dessutom försäkra sig om vilken väg paketen tar.

Ett problem med detta är naturligtvis all information som måste utbytas i nätet. I stora nät eller nät med hög mobilitet försvåras den här metoden. Å andra sidan har vi i vissa fall redan tillgång till positioner och god information om grannar (se kapitel 2.3), och informationen bör då utnyttjas. Det finns förslag på hur DSDV ska kunna ge QoS-garantier [15, 19], och även hur man ska kunna prediktera hur länkar förändras. Man sparar då ner information om kapaciteten i nätet och använder den som viktfunktion för att ta fram vägar istället för att använda minst antal hopp. För OLSR finns liknande förslag för att stödja QoS, där man t.ex. känner till kapacitet och fördröjningar i nätet och väljer lämplig väg [12, 20].

Principen i FSR - att inte varje nod i hela nätet ska ha lika god kännedom om alla noder är en intressant tanke i ett ad hoc-nät. Man inser snart då man studerar vanliga proaktiva protokoll för ad hoc-nät att de inte kan fungera i alltför stora nät eftersom kontrolltrafiken då skulle ta upp för mycket kapacitet i nätet. Man kan jämföra detta med Internet. Ingen router känner till varje nod i hela nätet eftersom det är ohanterbart. I små nät kan däremot alla känna till varandra. FSR kan således vara bra i stora ad hoc-nät genom att kontrolltrafiken där kan minskas.

Idag kan inte Fisheye State Routing hantera QoS, vi kan inte be om en viss kapacitet eller veta säkert att paketen kommer fram. Det finns dock förslag på att man skulle kunna lägga till detta i algoritmen [13]. En tanke är att länkarna

viktas och istället för att ange avståndet i antal hopp kan en godtycklig vikt istället anges, t.ex. högsta datatakt. Man skulle också kunna använda flera vägar till varje destination, och ange vilken kapacitet som dessa kan ge. Krävs en viss nivå av QoS för en förbindelse kan den väg väljas ur routingtabellen som är mest lämplig.

Kapitel 5

Reaktiva Routingprotokoll för mobila ad hoc-nät

5.1 Inledning

Reaktiv routing kallas också ofta *on-demand* routing. Vägarna skapas endast då de behövs, på begäran från den sändande noden. En förfrågan skickas ut bland de andra noderna för att hitta rätt väg. Då en väg hittas skickas en bekräftelse tillbaka till sändaren som även innehåller en vägbeskrivning. Vägen behålls därefter uppe tills den inte behövs mer eller tills avbrott sker.

De reaktiva routingprotokollen har fått mycket uppmärksamhet i ad hoc-nätssammanhang, och det är framförallt de två protokoll som presenteras nedan som är mest omdiskuterade idag, *Ad Hoc On-Demand Distance Vector* (AODV) [21] och *Dynamic Source Routing* (DSR) [22]. Andra reaktiva routingprotokoll för mobila ad hoc-nät är t.ex. *Temporally Ordered Routing Algorithm* (TORA) [8] och *Associativity Based Routing* (ABR) [8].

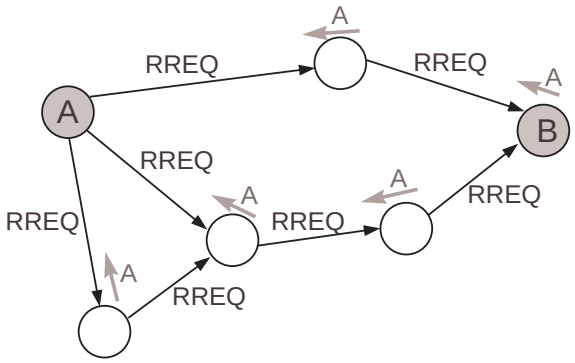
5.2 Ad Hoc On-Demand Distance Vector Routing

Ad Hoc On-Demand Distance Vector - AODV är en relativt enkel algoritm som inte ställer så stora krav på nätet. Den är utvecklad från DSDV (se kapitel 4.3) och de har därför en del likheter. Algoritmen kan delas upp i två delar, vägsökning samt underhåll av existerande väg.

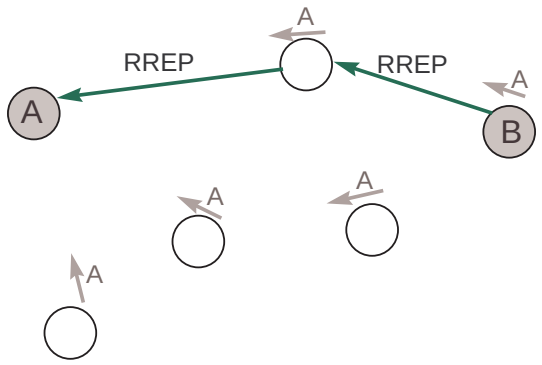
Då en sändare A ska skicka trafik till en mottagare B börjar man i AODV med att söka efter en väg som paketet kan sändas längs. Finns det ingen sedan tidigare använd väg skickar sändarnoden ut en förfrågan (*route request* - RREQ) om närmsta väg till sina grannar, se figur 5.1.

Genom att broadcasta, d.v.s. skicka förfrågan till alla grannar samtidigt sprids förfrågningen snabbt ut i nätet. Nästa nod tar emot förfrågan och skickar ut den till alla sina grannar. Noder som tar emot en förfrågan uppdaterar också sin information om sändaren A och sätter upp en *pekare* tillbaka till sändaren i sin routingtabell. Det kan liknas med hur ett system av vägar byggs samman och vägskyltar sätts upp i varje korsning, se figur 5.1. När sedan förfrågan når fram till mottagaren B eller en nod som känner till vägen till nod B skickas det en bekräftelse (*route reply* - RREP) på att en väg hittats. Denna bekräftelse tar då samma väg tillbaka som förfrågan som nådde fram först kom, se figur 5.2, och man kan därmed utöka tabellen av pekare med information om åt vilket håll mottagaren B finns. När sändare A till sist tar emot bekräftelsen kan trafik börja sändas.

En väg upprätthålls så länge den används. För att inte för gammal och därmed felaktig information om vägar ska finnas tas information bort efter en viss tid om länken inte använts. Om det skulle uppstå avbrott på en länk mellan nod



Figur 5.1: Sändaren A känner inte till någon väg till mottagare B, och sänder då en förfrågan, RREQ, genom nätet. I varje nod som passeras sätts en ”pekare” mot nod A.



Figur 5.2: Den närmaste vägen till nod B har hittats och en bekräftelse, RREP, sänds tillbaka till nod A. Pekare mot nod B sätts i noderna som passeras på vägen tillbaka och därmed kan trafiken börja flyta.

A och nod B skickar noden närmast avbrottet på sändarens sida ett felmeddelande till sändare A. Om sändaren fortfarande vill kunna nå nod B kan sändaren då skicka ut en ny förfrågan om väg.

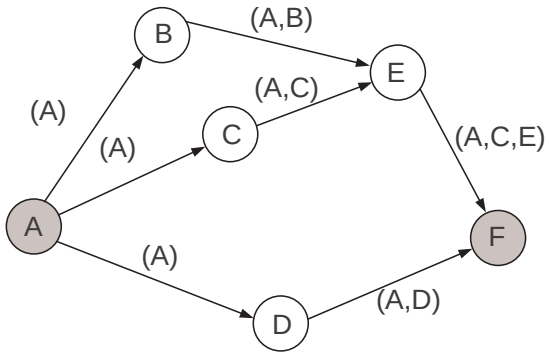
För att inte loopar ska uppstå då man söker efter vägar slänger en nod alltid en förfrågan den redan tidigare fått [10, 21, 23]. Det innebär att det endast är den första förfrågan en nod får som vidarebefordras.

5.3 Dynamic Source Routing

Dynamic Source Routing (DSR) [8, 22] är också ett rent reaktivt protokoll som endast söker upp en väg då det behövs. DSR bygger på källrouting (eng. *source routing*), vilket innebär att man redan i sändaren definierar exakt vilken väg ett meddelande skall ta genom att skicka med adresserna på noderna som ska passeras i paketet.

När en nod A ska sända ett paket till en mottagare F kontrollerar sändaren först om vägen finns sparad sedan förut. Om det finns en väg används den men om ingen väg tagits fram eller om vägen som finns är inaktuell måste sändaren leta efter en ny väg. Sändaren broadcastar då en förfrågan till alla grannar, se

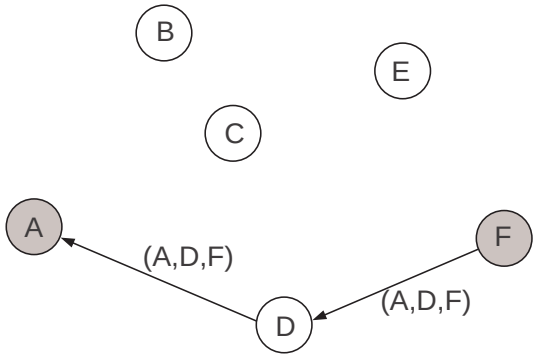
figur 5.3. Paketet innehåller adressen för sändaren A och mottagaren F och även ett identitetsnummer. Förfrågan skickas vidare på samma sätt som i AODV tills mottagaren nås. I de mellanliggande noderna kontrollerar man om man känner till vägen till mottagare F, eftersom man i så fall snabbare kan hitta en väg. Om så är fallet skickar den mellanliggande noden ett svar till mottagaren A innehållandes hela vägbeskrivningen och informationsöverföringen kan påbörjas.



Figur 5.3: Vägsökning med DSR. En förfrågan om väg skickas, och varje nod som passeras lägger till sin adress i paketet.

Om den mellanliggande noden inte känner till en väg till destinationen lägger man till sin egen adress i paketet och skickar det vidare. Det gör att då paketet når mottagare F finns alla adresser på noderna som passerats sparade i paketet med förfrågan. Därefter skickar mottagarnoden F ett svar som innehåller vägbeskrivningen till sändaren A, se figur 5.4. Paketet följer då vägbeskrivningen och skickas den snabbaste vägen till nod A som därefter kan börja sända.

För att inte onödig trafik ska gå i nätet eller loopar ska uppstå skickar noderna endast vidare förfrågningar som inte förut mottagits och paket där nodens adress inte redan finns med i paketets vägbeskrivning. För att kontrollera detta sparas identitetsnumret för en förfrågan hos de noder som passeras, och om en förfrågan med samma nummer senare tas emot slängs denna.



Figur 5.4: En väg har hittats, och för att paketet ska ta denna väg finns adresserna på de mellanliggande noderna med i paketen som skickas.

5.4 QoS-garantier

Några av de reaktiva protokollen har utvecklats för att ge stöd för garanterad tjänstekvalitet. Eftersom de reaktiva protokollen skickar ut en förfrågan om en väg kan QoS-garantier fås genom att man ställer vissa krav på vägen man vill ha. Det kan t.ex. handla om att man vill få en viss bandbredd eller att förbindelsen är stabil på något sätt.

Ett exempel är en vidareutveckling av algoritmen AODV [24] där man kan ställa krav på vägarna som tas fram. Man skickar då ställda krav med meddelandet om förfrågan efter väg. För att en nod längs vägen ska kunna skicka vidare en förfrågan krävs det inte bara att länken existera - den måste också uppfylla kraven. Dessutom, om en nod längs vägen märker att de begärda resurserna inte längre kan ges, måste noden skicka ett meddelande till sändaren och tala om att garantin inte längre kan ges.

För att kunna lägga till denna tjänst för QoS i AODV måste man utveckla routingtabellerna som används, samt meddelandena för förfrågan efter ny väg och bekräftelser. För varje destination i routingtabellen ska även maximal fördröjning, minsta tillgängliga bandbredd samt listor på andra källor som kräver garantier avseende fördröjning och bandbredd till destinationen finnas. En nod som kräver en garanti på till exempel bandbredd skickar en förfrågan innehål-

lande detta krav. För att en mellanliggande nod ska skicka paketet med förfrågan vidare krävs det att nästa länk uppfyller kravet. Annars slängs paketet. I bekräftelsen som skickas om vägen uppfyller kraven, finns också information om den kapacitet som den svagaste länken längs vägen maximalt klarar av att ge. Detta sker genom att ett fält i paketet sätts till största möjliga bandbredd. Detta fält är vid sändning från mottagaren satt till ett så stort nummer som möjligt, men vid varje nod som skickar paketet vidare jämförs fältet med länkkapaciteten och det minsta värdet väljs. Då bekräftelsen når sändaren vet man således vad minsta tillgängliga bandbredd är på sträckan. Andra källor som också bett om garantier får man också information om via bekräftelsen. Metoden som föreslås här kan jämföras med IntServ [7] som tagits fram för det fasta nätet och också reserverar resurser längs vägen. Man kan också tänka sig ungefär samma tillämpning och utvidgning av DSR för att även där kunna ge QoS.

Kapitel 6

Hybrider av Routingprotokoll för mobila ad hoc-nät

6.1 Hybrider

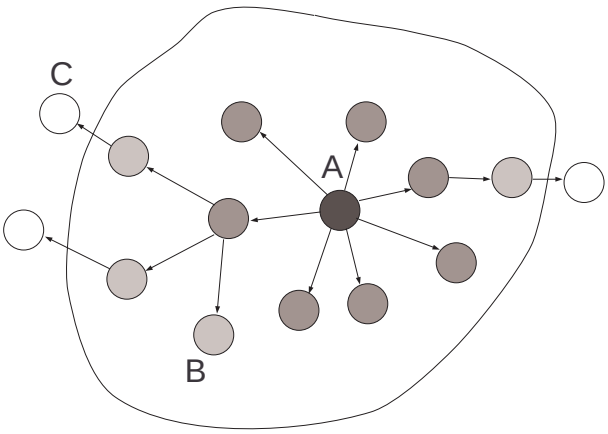
Det finns förutom reaktiva och proaktiva protokoll även hybrider mellan dessa. Det betyder att man har uppdaterad information om vägar till vissa destinationer medan man till andra måste söka efter vägen då den behövs. Ofta känner man till vägen till de närmaste noderna men ska ett paket skickas längre måste man söka efter vägen.

6.2 Zone Routing Protocol

Zone Routing Protocol (ZRP) [25] är ett hybridroutingprotokoll som är anpassat för ad hoc-nät, framförallt för de nät som täcker stora ytor och har stora skillnader i mobilitet inom området. ZRP bygger som namnet antyder på att nätet delas in i zoner. Varje nod har en egen definition av sin zon, och radien på zonen uttrycks i antal hopp. I figur 6.1 är till exempel antalet hopp två. Inom nodens zon sker proaktiv routing, och vägbeskrivningar uppdateras med jämna mellanrum. Utanför nodens zon sker däremot reaktiv routing. Det innebär att om noden vill skicka ett meddelande till någon utanför zonen måste man först leta efter en väg.

I figur 6.1 kan vi se hur nod A använder proaktiv routing inom sin zon med

radie två hopp. Då nod A vill sända till nod B har nod A redan uppgifter om vilken väg informationen kan gå och vi kan sända omedelbart. Detta kräver att vi sedan tidigare fått information om våra grannar, vilket görs genom att alla noderna med jämna mellanrum sänder ut signaler. Genom att lyssna på styrkan på signalen samt riktning kan man avgöra statusen på kontakten. Då nod A istället ska sända till nod C som ligger utanför zonen finns ingen uppdaterad väg. Reaktiv routing används då för att söka efter en väg, och man använder sig av en speciell routingalgoritm för att minska onödig information i nätet. Istället för att broadcasta (sända till alla) använder ZRP *bordercasting*. Man använder informationen som finns i varje zon om de andra noderna, och låter gränsnoderna vidareförmedla förfrågningar om väg. Genom att ha kontroll på hur meddelandet sprids i nätet kan man undvika onödig trafik i områden vi redan har kontroll över. Sändaren meddelas om vägen genom en bekräftelse då vägen hittats. Antingen skickas alla mellanliggande noders adresser med i paketet som med källrouting i DSR (se kapitel 5.3) eller så sparas adresser för nästa hopp i noderna längs vägen, som i AODV (se kapitel 5.2). För att inte behöva söka efter en ny om en länk bryts kan man inom en zon leda om förbindelsen utan att sändaren eller mottagaren märker något.



Figur 6.1: Exempel på ZRP, från nod A:s synvinkel, där proaktiv routing sker inom zonen och reaktiv routing utanför.

Antalet noder inne i zonen kan regleras genom att man ändrar diametern, antalet hopp, som definierar zonen. I ett relativt stabilt och statiskt nät kan man ha stora zoner men i mycket föränderliga nät kan det vara bättre med mindre zoner. I specialfallen kan man få ett helt reaktivt respektive helt proaktivt nät [25]. Radien kan också ändras då förhållandena i ett nät ändras, och detta gör ZRP till ett anpassningsbart protokoll. ZRP har visat sig effektivt i framförallt stora nät.

6.3 QoS-garantier

Idag finns inga färdiga metoder för hur QoS kan åstadkommas i ZRP, eller vilken effekt det skulle kunna få. Man kan dock konstatera att QoS för ett reaktivt routingprotokoll i kombination med QoS för ett proaktiv routingprotokoll skulle kunna användas och provas: Om man ska använda proaktiv routing känner man till kapaciteten hos länkarna och kan därigenom välja en väg som uppfyller ställda krav. Om man istället ska använda reaktiv routing skickas kraven med i förfrågan om väg. Kan kraven uppfyllas reserveras t.ex. en väg, och om de inte kan uppfyllas får kraven eventuellt sänkas och man får försöka igen.

Kapitel 7

Slutsatser

Det finns idag få routingprotokoll för mobila taktiska ad hoc-nät som ger stöd för QoS. Det mesta som finns är endast idéer och fortfarande behövs det forskning och utveckling inom detta område. Information som finns tillgänglig i nätet bör i den mån det är möjligt utnyttjas vid routing. Detta är t.ex. intressant för protokoll som när information finns arbetar proaktivt men i andra situationer är reaktivt. Zone Routing Protocol, se kapitel 6.2, är ett hybridprotokoll som dels kan utnyttja information om grannarna och använda proaktiv routing inom den närmsta omgivningen, och dels kan använda reaktiv routing i resten av nätet. Detta är framförallt intressant i stora ad hoc-nät och bör kunna användas ihop med t.ex. MAC-protokollet STDMA, där man har god information om grannarna upp till två hopp.

De reaktiva protokollen som hittills använts mycket i simuleringar av ad hoc-nät fungerar bra med tanke på att nätet ska vara decentraliserat och självkonfigurerande. QoS-garantier föreslås och kan fås genom att sätta krav på kapaciteten hos den väg som ska hittas. Nackdelen med de reaktiva protokollen är att de oftast inte utnyttjar sidoinformation som kan finnas i nätet, vilket leder till mindre effektiva vägar.

Vill man utnyttja tillgänglig information om positioner på alla noderna i nätet kan detta göras i de proaktiva protokollen. Dessa kan göras mer avancerade och därmed få med information som finns i nätet om t.ex. länkkapacitet, fördröjningar, positioner, rörelsemönster etc. Denna information kan sedan användas för att ge QoS-garantier, t.ex. genom att utnyttjas för att välja en väg som klarar de hos en användare ställda kraven. Det kan dock göra att nätet och

noderna blir komplexa och kräver information om nätet för att proaktiv routing ska kunna användas. En förutsättning är att vi även ska klara oss utan information om nätet, och därför behövs det alltid en möjlighet att använda reaktiva protokoll. En kombination av reaktiva och proaktiva protokoll kan lösa detta problem.

7.1 Fortsatt arbete

Då det ibland kommer att förutsättas att positionsförmedlingtjänster används i nätet bör denna information även utnyttjas i routingalgoritmen. De ställda kraven kan t.ex. vara att man ska ha god kännedom om positionerna hos andra noder som befinner sig inom 3 km omkrets. På längre avstånd minskar kraven på noggrannhet i positionen. Positionsmeddelanden skickas således oftare till närmsta grannarna och sprids mer sällan till noder längre bort. Eftersom Fisheye State Routing fungerar på ett liknande sätt, se kapitel 4.4, vill vi undersöka möjligheten att skicka routinginformationen tillsammans med positionsinformationen i nätet. Detta bör kunna leda till ett effektivare utnyttjande av kanalresurserna och en god uppfattning om nätet.

Vi vill också se vad som sker då vi inför krav på QoS i nätet och utnyttjar informationen som ges för att välja vägar som uppfyller dessa krav. Vi kommer att inrikta oss på routingalgoritmen FSR för att se om algoritmen är effektiv i de taktiska mobila ad hoc-näten och undersöka om QoS effektivt kan ges.

Ytterligare ett område som är av intresse är hybridprotokoll. Då information om nätet saknas vill vi fortfarande kunna använda våra routingalgoritmer och då är hybridprotokollen funktionella. Finns information så används den, i annat fall fungerar routingprotokollet ändå. Vi vill studera eventuella hybridprotokoll som använder FSR och om det inte finns undersöka om vi kan ta fram ett sådant protokoll.

Litteraturförteckning

[1] W. R. Stevens, *TCP/IP Illustrated, Volume 1*. Addison-Wesley, 1989.

[2] J. Reynolds och J. Postel. "Assigned Numbers,". RFC 1340, juni 1992. http://www.ietf.org/rfc/rfc1340.txt.

[3] K. Persson, "TCP/IP i taktiska ad hoc-nät," Rapp. FOI-R–0527–SE, Command and Control Systems, FOI, Swedish Defence Research Agency, juni 2002.

[4] J. Boleng, "Efficient Network Layer Addressing for Mobile Ad Hoc Networks," i *Proceeding of International Conference on Wireless Networks (ICWN´02*, s. 271–277, 2002.

[5] P. Thorén, "Internetworking of tactical ad hoc and static IP-network," Rapp. FOI-R–0822–SE, Command and Control Systems, FOI, Swedish Defence Research Agency, mars 2003.

[6] J. Grönkvist, "Assignment Strategies for Spatial Reuse TDMA," Lic. tech. thesis, Radio Communications Systems, Department of Signals, Sensors and Systems, Royal institute of Technology, SE-100 44 Stockholm, 2002.

[7] K. Persson, J. Grönkvist och A. Hansson, "Garanterad tjänstekvalitet i taktiska IP-nät," Rapp. FOI-R–0641–SE, Command and Control Systems, FOI, Swedish Defence Research Agency, november 2002.

[8] E. Royer och C. Toh, "A review of Current Routing Protocols for Ad-Hoc Mobile Wireless Networks," *IEE Personal Communications*, April 1999.

[9] S. Chen och K. Nahrstedt, “Distributed Quality-of-Service Routing in Ad-Hoc Networks,” *IEEE Journal on Selected Areas of Communication*, vol. 17, aug 1999.

[10] A. S. Tanenbaum, *Computer Networks*. Prentice Hall, 4 uppl., 2002.

[11] G. Apostolopoulos, D. Williams, S. Kamat, R. Guerin, A. Orda och T. Przygienda. “QoS Routing Mechanisms and OSPF Extensions,”. RFC 2676, augusti 1999. http://www.ietf.org/rfc/rfc2676.txt.

[12] G. Ying. “Quality-of-Service Routing in Ad-Hoc Networks Using OLSR,”. Master’s thesis, Ottawa-Carleton Institute of Computer Science, Ottawa, Canada, december 2002.

[13] G. Pei, M. Gerla och T.-W. Chen, “Fisheye State Routing in Mobile Ad Hoc Networks,” *Workshop on Wireless Networks and Mobile Computing*, s. D71–D78, 2000.

[14] T. Claussen, P. Jacquet, A. Laouiti, P. M. andP. Muhlethaler, A. Qayyum och L. Viennot. “Optimized Link State Routing Protocol,”. IETF Internet Draft (work in progress), 2002. draft-ietf-manet-olsr-07.txt.

[15] G. He. “Destination-Sequenced Distance Vector (DSDV) Protocol,”. http://citeseer.nj.nec.com/531710.html, 2002. Networking Laboratory, Helsinki University of Technology.

[16] A. C. Sun. “Design and Implementation of Fisheye Routing Protocol for Mobile Ad Hoc Networks,”. Master’s thesis, Massachusetts Institute of Technology, maj 2000.

[17] M. Gerla, X. Hong och G. Pei. “Fisheye State Routing Protocol (FSR) for Ad Hoc Networks,”. IETF Internet Draft (work in progress), 2002. draft-ietf-manet-fsr-03.txt.

[18] S. H. Shah och K. Nahrstedt, “Predictive Location-Based QoS Routing in Mobile Ad Hoc Networks,” Rapp., Department of Computer Science, University of Illinois at Urbana-Champaign, U.S.A., September 1992.

[19] C. R. Lin och J.-S. Liu, “QoS Routing in Ad Hoc Wireless networks,” *IEEE Journal on Selected Areas of Communication*, vol. 17, s. 1426–1438, aug 1999.

[20] A. Munaretto, H. Badis, K. A. Agha och G. Pujolle, “A Link-state QoS Routing Protocol for Ad Hoc Networks,” i *IEEE MWCN’02: International Workshop On Mobile and Wireless Communications Networks*, 2002.

[21] C. Perkins, E. M. Royer och S. R. Das. “Ad hoc On-Demand Distance Vector (AODV) Routing,”. IETF Internet Draft (work in progress), februari 2003. draft-ietf-manet-aodv-13.txt.

[22] D. B. Johnson, D. A. Maltz och Y.-C. Hu. “The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR),”. IETF Internet Draft (work in progress), 2003. draft-ietf-manet-dsr-09.txt.

[23] C. E. Perkins och E. M. Royer, “Ad -hoc On-demand Distance Vector Routing,” *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications*, s. 90–100, feb 1999.

[24] C. Perkins, E. M. Royer och S. R. Das. “Quality of Service for Ad hoc On-Demand Distance Vector Routing,”. IETF Internet Draft (work in progress), 2000. draft-ietf-manet-qos-00.txt.

[25] N. Beijar. “Zone Routing Protocol (ZRP),”. http://citeseer.nj.nec.com/538611.html, April 2002. Networking Laboratory, Helsinki University of Technology.