TEKST NR 437 2004

Ruteplanlægning i ad-hoc netværk

Kasper G. Christensen og Jens Chr. Larsen

TEKSTER fra

IMFUFA ROSKILDE UNIVERSITETSCENTER
INSTITUT FOR STUDIET AF MATEMATIK OG FYSIK SAMT DERES FUNKTIONER I UNDERVISNING, FORSKNING OG ANVENDELSER

IMFUFA
Roskilde Universitetscenter
Postboks 260
DK-4000 Roskilde

t +45 46 74 22 63
f +45 46 74 30 20
m imfufa@ruc.dk
w imfufa.ruc.dk

Kasper G. Christensen og Jens Chr. Larsen: Ruteplanlægning i ad-hoc netværk

IMFUFA tekst nr. 437/2004 49 sider issn 0106-6242

I dette projekt ser vi på ruteplanlægning i mobile ad-hoc netværk. Det oprindelige formål var at muliggøre analytisk sammenligning af konkrete ruteplanlægningsalgoritmer. Dette viste sig imidlertid vanskeligt, og vi har derfor arbejdet med et bredere spørgsmål, der lyder:

> Hvad kan matematiske modeller sige om ruteplanlægning i mobile ad-hoc netværk, og hvilke mangler har disse modelbygninger i forhold til at foretage en vurdering af konkrete ruteplanlægningsalgoritmer?

Vi fremdrager modeller baseret på stokastisk grafteori. Navnlig går vi i dybden med to modeller. En der beskriver hvad sandsynligheden er, for at et netværk med endeligt antal enheder er forbundet, og en der beskriver hvorledes benyttelse af flere ruter, kan nedsætte ruteplanlægningsalgoritmens overhead, ved at udskyde behovet for at søge efter nye ruter.

Konklusionen er, at det er begrænset, hvad vi på nuværende tidspunkt kan sige om ruteplanlægning vha. matematiske modeller, men at vi som en begyndelse kan give ovennævnte begrænsede delresultater. Det kan diskuteres, hvorvidt nogle af modelantagelserne er for grove.

1 Forord

Nærværende Imfufa-tekst er resultatet af vores projektarbejde i forårssemestret 2003 på matematik, og en efterfølgende lettere redigering af den oprindelige projektrapport.

Efter projekteksamen blev vi opfordret til at lave denne udgave af projektrapporten som Imfufa-tekst, specielt fordi censor og vejleder fandt formen og sproget, der bl.a. i afsnit 6.1 afspejler, hvorledes vi har famlet os frem, lærerigt. Vi håber således, at andre studerende kan finde inspiration i dette arbejde med matematikken. Desuden indeholder det netop nævnte afsnit originale tanker, som vi ikke har set andre steder.

En af bagsiderne, ved dette projekt, er, at det angriber et forholdsvist bredt problem, som vi ikke kan give noget fuldstændigt svar på. Vi er således langt fra til bunds i problemstillingen.

Forfatterne til dette projekt er RUC-studerende. Jens Christian Larsen har en humanistisk basisuddannelse og studerer filosofi og matematik. Kasper G. Christensen har en naturvidenskabelig basisuddannelse og studerer datalogi og matematik. Begge fik vi vores bachelorgrad med afleveringen af dette projekt. Vores fælles reference er således matematikken, som vi ser på fra forskellige indfaldsvinkler.

God fornøjelse med læsningen!

Indhold

2 Indledning

I ethvert datanetværk er spørgsmålet om, hvordan data kan komme fra A til B, centralt. Hvis alle computere i netværket er direkte forbundet, er det et trivielt spørgsmål, men i større netværk, som f.eks. Internettet, men også de fleste større lokalnet, ville dette kræve et helt urealistisk antal af forbindelser (dvs. f.eks. kabler, sattelitlinks osv.). Derfor må der findes en anden løsning. Her må computere i netværket sende data videre for hinanden, således at data sendes fra den ene til den anden, indtil de når den rette modtager. Spørgsmålet bliver da om, og hvordan det er muligt at finde en sådan vej fra en given computer til en anden. Grundlæggende er *ruteplanlægning* den teknik, der fastlægger den vej data transporteres fra et punkt til et andet i et mere eller mindre komplekst netværk. Disse ruter kan findes med forskellige kriterier for øje: Måske er vi tilfredse blot vi har en rute, andre gange vil vi måske helst have flere ruter, eller vi ønsker at finde den hurtigste rute. For netværk, der kun ændrer sig langsomt, dvs. netværk med faste forbindelser, som f.eks. Internettet og de fleste lokalnet, findes der algoritmer, der finder den korteste rute (med hensyn til forskellige mål for afstand). Det, der gør disse algoritmer mulige, er, at der opbygges stor viden om netværkets struktur, samt at denne information bliver forældet meget langsomt. Denne langsomme forældning, der direkte følger af, at den grundlæggende faste struktur kun sjældent og langsomt ændres, betyder ligeledes også, at algortimerne arbejder effektivt, idet en stor og hyppig udveksling af information om netværkets struktur ikke er nødvendig.

Imidlertid er interessen for trådløse netværk, uden nogen form for fast infrastruktur, vokset. De fremhæves bl.a. af militæret som værende nyttige i fjendens land, og i katastrofesituationer hvor den faste infrastruktur er ødelagt. Med udbredelsen af håndholdte computere og interessen for at bygge sensorer eller computere ind i snart sagt hvad som helst, bliver flere og flere mere dagligdags anvendelser også foreslået. I disse netværk, kaldet *MANET* (mobile ad-hoc network), er der naturligvis også brug for ruteplanlægning, men da disse netværk kan tænkes at ændre sig hurtigt og ofte, og da overførselshastigheden ved trådløse forbindelser er meget lavere end typiske faste forbindelser, fungerer de eksisterende ruteplanlægningsalgoritmer, der benyttes i netværk med faste forbindelser, dårligt i disse ad-hoc netværk (Park & Corson; 1997).

Der er udviklet en række algoritmer til ruteplanlægning i MANETs, og effektiviteten af disse er hovedsageligt undersøgt vha. simuleringer, samt i enkelte egentlige tests. Sigtet med dette projekt, er at finde nogle matematiske værktøjer, der kan benyttes til analytisk at vurdere ruteplanlægningsalgoritmer designet til MANETs. Udover den rent matematiske tilfredsstillelse ved dette, samt den følelse af bedre forståelse for tingenes sammenhæng der kan følge af en sådan modellering, er der endnu et argument for, at en analytisk tilgang, frem for en tilgang baseret på simuleringer, kan være interessant: Simuleringerne foregår vha. såkaldte netværkssimulatorer, og en undersøgelse af disse viser, at selv en meget simpel algoritme implementeret i 4 forskellige af disse simulatorer, viser ret forskellig opførsel i disse (Cavin et al.; 2002). Der

er altså grund til at stille spørgsmålstegn ved simuleringernes pålidelighed. Tilsvarende kan der sandsynligvis også stilles spørgsmålstegn ved de modeller, der kan benyttes som grundlag for at udlede analytiske resultater, men vi finder det alligevel interessant at se nærmere på disse.

Det viser sig imidlertid, at der kun i meget begrænset omfang er gjort forsøg på at lave analytiske vurderinger af ruteplanlægningsalgoritmer. Opgaven i dette projekt er derfor i første omgang, at se på nogle af de kriterier samt matematiske værktøjer, der muligvis kan føre til, at det bliver muligt at vurdere ruteplanlægningsalgoritmer på et analytisk grundlag. Første skridt er derfor at se på, hvad vi overhovedet p.t. kan sige analytisk om ruteplanlægning i ad-hoc netværk.

Ud fra disse betragtninger bliver problemformuleringen:

2.1 Problemformulering

Hvad kan matematiske modeller sige om ruteplanlægning i mobile ad-hoc netværk, og hvilke mangler har disse modelbygninger i forhold til at foretage en vurdering af konkrete ruteplanlægningsalgoritmer?

2.1.1 Uddybning af problemformuleringen

Problemformuleringen skal læses i lyset af, at det oprindelige mål med dette projekt, var at vurdere konkrete ruteplanlægningsalgoritmer ved hjælp af matematisk modellering. Dette viste sig imidlertid at være vanskeligt, da der kun er udført meget begrænset arbejde inden for området. Derfor udvider vi fokus til, som et skridt på vejen, at fremhæve og diskutere nogle af de modelbygninger, der kan sige bare en smule om ruteplanlægning i MANETs. I den forbindelse ser vi på, hvilke kriterier som faktisk er blevet benyttet under designet af eksisterende ruteplanlægningsalgoritmer til MANETs, idet en delvis besvarelse af spørgsmålet om vurdering af disse algoritmer, kunne være en vurdering af relevansen af disse design-kriterier.

2.2 Målgruppe

Nærværende rapport er tilstræbt skrevet, således at studerende på vores eget niveau – dvs. med grundlæggende matematiske færdigheder inden for matematik, men uden kendskab til grafteori og med begrænset kendskab til sandsynlighedsregning, er i stand til at læse og forstå den.

Indholdet henvender sig til alle med interesse for matematisk modelbygning inden for datalogien, og dennes anvendelse i forbindelse med ruteplanlægning i mobile ad hoc netværk.

2.3 Rapportens opbygning

Rapporten består af en række hovedafsnit, hvis formål, for overblikkets skyld, kort vil blive skitseret i det følgende.

Afsnit 3 forklarer på et overordnet niveau, hvad et ad-hoc netværk er, med det formål at forberede en matematisk modellering af disse.

Afsnit 4 omhandler ruteplanlægning i ad-hoc netværk, set fra en anvendelsesorienteret synsvinkel. Ofte anvendte kriterier for hvad der i forbindelse med design af ruteplanlægningsalgoritmer til ad-hoc netværk bliver lagt vægt på, bliver trukket frem. Ruteplanlægningsalgoritmen DSR beskrives, idet dette er et ofte studeret eksempel på en ruteplanlægningsalgoritme til ad-hoc netværk.

Afsnit 5 fremsætter nogle grafteoretiske definitioner, samt forklarer kort hvorledes disse typisk bruges til at opstille matematiske modeller for netværk. Grundideen med stokastiske grafer forklares, og et vigtigt resultat fra stokastisk grafteori gengives.

Afsnit 6 beskriver to modeller med relevans for ruteplanlægning i ad-hoc netværk, samt de analytiske resultater disse giver. Her trækkes på de foregående afsnit. Dette er rapportens egentlige hovedbestanddel.

Afsnit 7 diskuterer rapportens samlede indhold, med speciel vægt på de i afsnit 6 beskrevne modeller og resultater. Afsnittet lægger op til rapportens konklusion, der følger i afsnit 8.

Afsnit 9 giver nogle bud på, hvorledes arbejdet med modellering af ruteplanlægning i ad hoc netværk kan fortsættes.

3 Hvad er et ad-hoc netværk?

Før vi ser på matematiske modeller for *mobile ad-hoc netværk* (MANET), vil vi først se lidt nærmere på, hvad et *MANET* egentlig er.

Idéen med MANETs er, at de gør brugeren uafhængig af tilstedeværelsen af fast infrastruktur, og dermed fjerner brugerens bundethed til f.eks. stationære computere eller stik i væggen. Således kan brugerene bevæge sig frit i forhold til hinanden. MANETs adskiller sig fra for eksempel netværk til mobiltelefoner. Disse netværk har typisk det, der kaldes et kablet backbone. Det vil sige, at netværket består af en række sendemaster, der indbyrdes er forbundet med (lysleder)kabler. Dermed er der kun tale om mobilitet for brugeren af f.eks. en mobiltelefon; alle andre elementer i netværket forbliver stationære. I et MANET er alle enheder i netværket mobile. Tanken er, at fordi der ikke findes en infrastruktur, er nettet mindre sårbart over for fejl eller angreb, hvorfor bl.a. den amerikanske hær støtter forskning i MANET.

3.1 Trådløse netværk

I et *trådløst netværk* skabes alle forbindelser ved hjælp af radiobølger. Disse udbredes i koncentriske cirkler omkring enheden, men kun ud til en vis afstand. Denne afstand bliver bestemt af enhedens *sendekraft*, der er afhængig af enhedens strømforsyning. Desto mere kraft der bliver sendt med, desto større krav stilles til strømforsyningen. Vigtigere endnu er det faktum, at hvis mange enheders sendeområder overlapper, opstår der interferens, hvilket betyder, at den *tilgængelige båndbredde* bliver kraftigt reduceret. Derfor er det ikke fordelagtigt (og som oftest heller ikke muligt), at alle deltagere i netværket sender med en sådan kraft, at de kan nå alle andre enheder i netværket. Dermed bliver spørgsmålet om ruteplanlægning relevant. Faktisk opnås de bedste resultater, hvis *senderadiussen* minimeres, således at det lige netop sikres, at der findes ruter mellem alle enheder i netværket (Gupta & Kumar; 1998).

Når der er mange enheder tilstede, der sender samtidigt, kan det som sagt medføre interferens, hvorved den tilgængelige båndbredde nedsættes. Dette fysiske forhold er tidligere forsøgt modelleret, men det kræver modellering på et lavere fysisk niveau, end vi vil forsøge os med i dette projekt, hvor vi vil koncentrere os om den *grafteoretiske abstraktion* (som indføres i afsnit 5). Tilsvarende kan anden radiostøj også medvirke til at gøre den effektive båndbredde mindre. Herudover kan genstande i landskabet betyde, at en enhed ikke kan sende lige langt i alle retninger, og den grundlæggende antagelse om, at radiobølgerne udbreder sig i cirkler, er derfor ikke nødvendigvis korrekt i et naturligt bakket landskab med diverse terrængenstande m.m.

En trådløs forbindelse er ikke nødvendigvis en *tovejsforbindelse*. Forskellige fysiske forhold, som f.eks. inteferrens, forskel i to enheders sendestyrke eller antennetyper, kan betyde, at der

kun er forbindelse den ene vej (Johnson et al.; 2001). I nogle trådløse netværk dikterer netværkets arkitektur, at der kun må eksistere tovejsforbindelser, hvorfor envejsforbindelserne ikke benyttes. I denne type netværk vil ruteplanlægningsalgoritmen derfor se det som om, alle forbindelser er tovejs (Park & Corson; 1997). I andre trådløse netværk tillades envejs forbindelser. Af hensyn til simplicitet i modelleringen, vil vi i dette projekt koncentrere os om netværk, hvor alle forbindelser er tovejs.

3.2 Yderligere forhold

Der er ikke noget galt for at have mange typer af udstyr i det samme ad-hoc netværk. F.eks. kunne man forestille sig, at enheder placeret i biler har bedre strømforsyning, og derfor kan sende med større kraft uden problemer, mens håndholdte enheder har mere begrænset rækkevidde. Desuden er det muligt, at nogle enheder kan ændre deres senderadius (og dermed strømforbrug) efter forgodtbefindende. I dette projekt vil vi imidlertid antage, at udstyret er ens for alle deltagere i netværket, og at dette udstyr sender med den samme kraft eller senderadius. Dette er udelukkende af hensyn til simplicitet i modelleringen.

I dette projekt taler vi om *pakkenetværk* (packet-switched). Det betyder, at data brydes op i små afsluttede stumper, kaldet *pakker*, der sendes gennem netværket uafhængigt af hinanden. Dette vil sige, at det som vi måske ville opfatte som »en samlet forsendelse« i virkeligheden bliver brudt op, og evt. ligefrem sendt af forskellige ruter, for så at blive samlet igen ved modtagelsen. Det betyder også, at enkelte dele af »forsendelsen« kan gå tabt, hvilket typisk vil resultere i, at disse dele (i form af pakker) sendes igen. Hvorledes dette foregår, skal vi imidlertid ikke koncentrere os om i dette projekt, men blot huske, at data sendes som relativt små separate pakker. Dette adskiller forøvrigt denne type netværk fra f.eks. telefonnetværk, som er ruteorienterede (cicuit-switched). I ruteorienterede netværk reserveres en bestemt rute med tilhørende resourcer, før kommunikation mellem to enheder begyndes, og kommunikationen mellem disse to vil derefter foregå via denne reserverede forbindelse, som frigives igen ved kommunikationens afslutning (Kurose & Ross; 2001).

4 Ruteplanlægning i ad hoc netværk

Der findes en lang række algoritmer beregnet på *ruteplanlægning* i ad hoc netværk. Fælles for disse er, at de er designet specielt med henblik på de særlige problemer, der er forbundet med rutnning i MANET, herunder begrænset båndbredde og en *netværkstopologi der ofte ændrer sig*. Det er netop disse to forhold, der er med til at definere, hvad der anses for at være fornuftige egenskaber for en ruteplanlægningsalgoritme. Det følgende vil handle om, hvilke *kvaliteter* ruteplanlægningsalgoritmer skal have, samt de *designfilosofier* der benyttes, når algoritmerne konstrueres.

4.1 Kvalitetskriterier

Kriterierne er forbundet til *mobilitet* og den *begrænsede båndbredde*. Vi holder MANETs op imod statiske netværk, dvs. netværk, som f.eks. Internettet, hvor forbindelserne ændrer sig forholdsvist sjældent, for at give en idé om problemerne. Det følgende er tildels baseret på Corson et al. (1996). Først defineres, hvilke krav mobilitet stiller til algoritmen:

> - to build routes quickly so that they may be used before the topology changes, and
> - to react quickly, reestablishing routing (if possible and desired) when topological changes destroy existing routes.
>
> (Corson et al.; 1996, p. 10)

Før vi ser på, hvordan det kan gøres i MANETs, ser vi først på statiske netværk. Algoritmer til brug i statiske netværk (f.eks. OSPF (Open Shortest Path First) og RIP (Routing Information Protocol)) udveksler periodisk oplysninger om netværkstopologien, således at ruter til alle enheder i netværket, så vidt muligt, hele tiden kendes (Kurose & Ross; 2001). I et statisk netværk betyder dette, at der ikke går nævneværdig tid med at finde en rute, når den forespørges. I et ad hoc netværk er denne fremgangsmåde imidlertid ikke hensigtsmæssig, da topologien kan ændre sig så hurtigt, at det vil kræve meget stor udveksling af oplysninger, for konstant at opdatere ruteoplysningerne (Park & Corson; 1997). Derfor lægges der normalt i designet af algoritmer til mobile netværk, som f.eks. TORA (Temporally Ordered Routing Algorithm beskrevet i Park & Corson (1997)) og DSR (Dynamic Source Routing beskrevet i Johnson et al. (2001)), vægt på at algoritmerne ikke skal opbygge ruter før disse behøves. Altså forsøger algoritmerne først at finde en rute til en destination, når en enhed ønsker at sende data til denne. Dette kaldes med et engelsk udtryk for *on-demand routing*. I det følgende bliver båndbredden inddraget i betragtningerne.

> Bandwidth is at a premium in mobile applications. Protocols must be as lightweight as possible, preferably expending no more than 10 percent of total network traffic on overhead. (Corson et al.; 1996, p. 5)

Overhead er datapakker, der ikke er den data, der ønskes sendt mellem to enheder, men derimod pakker, der har til formål at få algoritmen til at fungere, f.eks. ved at finde ruter eller berette om fejl i ruter.

I både TORA og DSR findes en rute ved at udsende en *forespørgsel*, der i princippet når alle enheder i netværket. Dette foregår vha. såkaldt *flooding*, hvor en enhed når den modtager forespørgslen, undersøger om den allerede har videresendt denne. Hvis ikke dette er tilfældet, videresender den til alle sine naboer. Denne taktik har sine svagheder. Problemet opstår, hvis der er mange af denne type pakker, for så benyttes den sparsomme båndbredde på at sende forespørgselspakker, i stedet for at sende den egentlige data. Forespørgselspakker som disse, eller pakker der beretter om fejl forskellige steder i netværket, skal dermed minimeres.

Hvis algoritmerne opbygger ruter on-demand, vil det allerede være begrænset hvor meget overhead der forekommer, da der ikke konstant vedligeholdes ruter. Der findes dog andre designmuligheder der begrænser overhead.

I Internettets algoritmer er der indbygget en mekanisme, der finder den korteste rute mellem to enheder, målt f.eks. i antallet af mellemliggende enheder. Dette kan tillades, da forbindelserne i Internettet er hurtige og stabile, det medfører, at det ikke betyder noget, at der bruges tid på at finde den korteste rute, for det er meget sandsynligt, at den rute stadig er den korteste, når algoritmen er færdig med at regne.

> In a rapidly changing topology, it is likely that the topology will change again before a shortest-path computation can converge. [...] in a congested mobile network, quickly discovering multiple routing options is likely preferrable to computing a single, shortest-path route. (Corson et al.; 1996, p. 9).

I MANETs er det altså på ingen måde sikkert, at når algoritmen har beregnet den korteste rute, at ruten stadig eksisterer, eller stadig er den korteste. Derfor kan det generelt ikke betale sig søge efter den korteste rute i MANETs. Det betyder dog også, at når algoritmen finder en rute, så er den ikke nødvendigvis den mest optimale. Et argument for at flere ruter er fordelagtigt, er en betragtning om, at flere ruter giver større stabilitet. Da data sendes som pakker, kan disse pakker sendes af forskellige ruter. Derved udnyttes båndbredden bedre, og der opnås større sikkerhed for, at forbindelsen ikke forsvinder, når en rute bliver brudt. Det at have flere ruter sikrer ikke nødvendigvis en hurtig tilpasning, idet formålet med de ekstra ruter er at udskyde det tidspunkt, hvor det er nødvendigt at ruteforespørge, skal der bruges tid på at finde de ekstra ruter. Hvis der går for lang tid med at finde disse ruter, kan netværket have ændret sig meget.

Der kan opstilles to kvalitetskriterier: Hurtig tilpasning og minimalt overhead. Herunder står de tilhørende designfilosofier under dem.

Kvalitetskriterier og afledte designfilosofier

- Hurtig tilpasning
  - On-demand ruter

  - Det er ikke nødvendigt at finde den korteste rute
- Minimalt overhead
  - On-demand ruter
  - Det er ikke nødvendigt at finde den korteste rute
  - Flere ruter

Disse kriterier kan betragtes som påstande, der er baseret på logisk argumentation. Dette er tildels rigtigt, idet vi ikke har kendskab til, at disse kriterier har været afprøvet i virkeligheden, hverken af os eller Corson et al. (1996). Dette betyder ikke, at kriterierne ikke er fornuftige. De virker ganske intuitive, og de går direkte eller indirekte igen i litteraturen. Derfor kan de ikke bare ses bort fra. Disse kriterier viser den tankegang, der ligger bag ruteplanlægningsalgoritmerne til MANETs. På den måde giver de udtryk for de kvaliteter konstruktørerne af algoritmerne lægger for dagen. Derved er de relevante, for den måde algoritmerne bliver betragtet i litteraturen og i dette projekt.

I det ovenstående spiller tid ind. Tidsaspektet ved de ovenstående designfilosofier spiller en rolle, idet det giver mulighed for at kvantificere, hvorved forskellige algoritmer kan holdes op mod hinanden. Der fremkommer mindst fire tider fra det ovenstående:

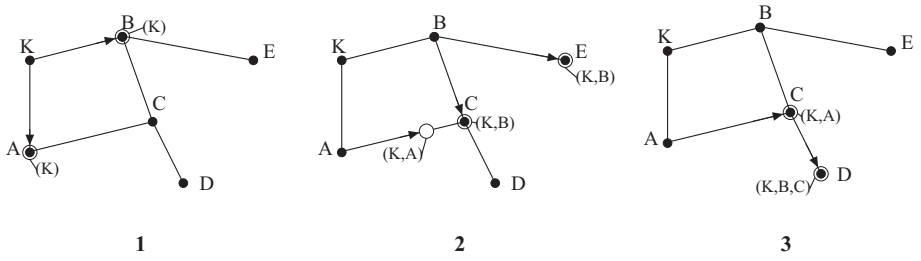
- Hastigheden hvormed netværket ændre sig
- Tid mellem ruteforespørgsler
- Søgetid (tiden det tager at finde en rute)
- Samlet ventetid (den tid en pakke i gennemsnit må vente)

Af disse tider vil der i dette projekt blive set nærmere på tiden mellem ruteforespørgsler.

For at give et eksempel på hvordan ruteplanlægning i et ad hoc netværk kan foregå, vil følgende afsnit beskrive ruteplanlægningsalgoritmen DSR, der er en ofte studeret algoritme, som ofte refereres i litteraturen. Hele nedenstående beskrivelse af DSR-algoritmen bygger på Johnson et al. (2001) – figurerne er dog fremstillet frit på baggrund af den tekstuelle beskrivelse. DSR kan håndtere netværk, hvor forbindelserne ikke nødvendigvis er tovejs, og dette vil, for fuldstændighedens skyld, blive nævnt et par gange, men vil ikke blive fremhævet specielt, da vi har valgt ikke at arbejde med denne type netværk.

4.2 Ruteplanlægningsalgoritmen DSR

DSR-algoritmen kan grundliggende opdeles i ruteopbygning og -vedligeholdelse. *Ruteopbygning* består i udsendelse af forespørgselspakker, der udbredes gennem netværket indtil de når destinationen, hvorpå destinationen besvarer forespørgslen. En forespørgselspakke indeholder en beskrivelse af den rute, den har fulgt, og når den når frem til destinationen, kan denne således sende en fuld rutebeskrivelse tilbage til forespørgeren. Alle enheder har mulighed for at gemme de ruter, de kender, således at de kan vedblive at benytte disse uden at skulle foretage nye forespørgsler. *Rutevedligeholdelse* består i at sørge for at disse gemte ruter bliver fjernet, hvis de bliver ugyldige pga. ændringer i netværkstopologien. Denne grundlæggende ruteforespørgsel og -vedligeholdelse kan forbedres på en række punkter, dels ved at give enheder mulighed for at udnytte de ruteinformationer de kan opsnappe fra andre enheders kommunikation, og dels



Figur 4.1 Illustration af ruteforespørgsel fra K til D. Cirklerne viser, hvorledes forespørgsler spredes gennem netværket frem til destinationen D. I parantes er ved hver forespørgsel (cirkel) angivet, den rute den har fulgt

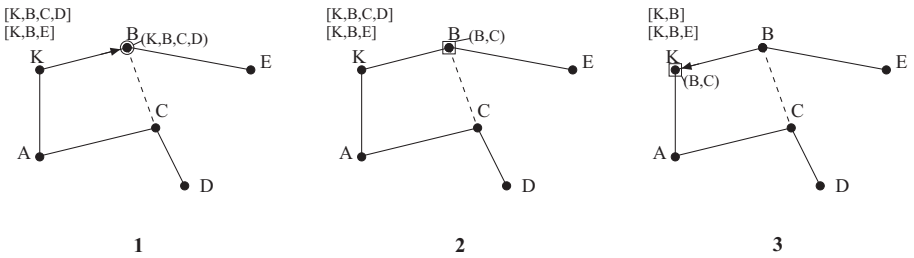
ved at udnytte de gemte ruter mere intensivt. Inden vi beskriver algoritmen nøjagtigt, vil vi først se på et par eksempler til at belyse den grundlæggende funktionalitet.

4.2.1 Et par belysende eksempler

På figur 4.1 er illustreret, hvorledes en forespørgsel af en rute fra kilden K til destinationen D spredes gennem et netværk. I forespørgslen er angivet et identifikationsnummer, navnet på destinationen, samt en liste af enheder, der angiver hvilken vej forespørgslen har fulgt. Når en enhed modtager forespørgslen, undersøger den om den selv er destinationen, er dette tilfældet aflæser den ruten, og sender denne i en svar-pakke til den oprindelige kilde til forespørgslen. Hvis ikke enheden selv er destinationen, undersøger den om den for nylig har modtaget denne forespørgsel (dvs. en forespørgsel efter samme destination, fra samme kilde og med samme identifikationsnummer), er dette tilfældet ignoreres forespørgslen. Hvis enheden hverken er destination, eller har modtaget forespørgslen før, skal enheden sende forespørgslen videre, og den tilføjer sit navn til den benyttede rute, hvorpå forspørgslen sendes ud, hvorved alle enheder inden for dens sendeafstand, modtager den. I det følgende vil vi kalde de enheder, der er inden for en given enheds sendeafstand, for enhedens *naboer*.

På illustrationen i figur 4.1 ses en sådan udbredelse af en forespørgselspakke. Af hensyn til overskueligheden er de pakker, der går tilbage til afsenderen (da pakken nødvendigvis sendes til alle naboer) ikke medtaget. F.eks. vil enhed K få forespørgslen tilbage fra B, når B videresender forespørgslen med sit eget navn tilføjet ruten, enhed K vil imidlertid se, at den har set denne forespørgsel før, og vil derfor blot ignorere den. I trin 2 kommer forespørgslen hurtigere frem via B end via A, hvorfor forespørgslen der kom via A i trin 3 vil blive ignoreret af enhed C (der jo lige har set forespørgslen en gang). Dermed er det den p.t. hurtigste rute, der findes, men den alternative rute via A findes altså ikke i dette forløb. Når forespørgslen når destinationen D i trin 3, skal D returnere den til K. Dette kan den enten gøre via en rute den allerede kender, eller hvis netværkstypen betyder, at alle forbindelser er tovejs, kan den simpelthen vende ruten om, og sende forespørgslen denne vej tilbage, endelig kan den selv starte en rute-forespørgsel til K, og sende svar på K's forespørgsel sammen med denne (for at undgå en uendelig løkke af forespørgsler, er det væsentligt at svaret til K indeholdende ruten til D, sendes sammen med en evt. forespørgsel fra D om en rute til K).

På figur 4.2 er rutevedligeholdelse illustreret. Kilden K har afsendt en datapakke ad ruten



Figur 4.2 Illustration af rutevedligeholdelse. Ved enheden K er i kantede paranteser angivet, hvilke ruter K kender. I trin 1 har B netop modtaget en datapakke fra K, i hvilken K har angivet, at den skal følge ruten K,B,C,D, imidlertid er forbindelsen fra B til C afbrudt, så i trin 2 sender B en fejlpakke (illustreret som en firkant) til K, med oplysning om den afbrudte forbindelse. I trin 3 modtages fejlpakken, og K tilretter listen over ruter

(K,B,C,D), men forbindelsen mellem B og C er afbrudt. Det er enhver enheds opgave at verificere, hvorvidt en datapakke er kommet frem til den næste enhed på ruten, er dette ikke tilfældet, skal enheden sende en fejlpakke. I fejlpakken angives, hvilken forbindelse der har fejlet. Når en enhed (her K) modtager fejlpakken, leder den sine ruter igennem, og de ruter, der benytter den brudte forbindelse, bliver afkortet ved denne. Hvis K havde kendt den alternative rute (K,A,C,D), hvilket godt kunne have været tilfældet, kunne K straks have fortsat sin kommunikation via denne rute, da K i eksemplet på figur 4.2 ikke har flere ruter til D, må K i stedet foretage en ruteforspørgsel til D, hvis kommunikationen skal fortsættes.

4.2.2 Forbedringer

Ovenstående forespørgselsproces kan forbedres på flere måder. Først kan de gemte ruter i hver enkelt enhed benyttes mere intensivt. Det kan f.eks. tillades, hvis en enhed modtager en forespørgsel til en destination, som den selv kender en rute til, kan den straks sende et svar med denne rute, i stedet for at lade forespørgslen gå videre til destinationen. Denne fremgangsmåde kan dog afføde et problem: Hvis en enhed forespørger en destination, hvortil mange af dens naboer kender en rute, vil alle disse naboer ca. samtidigt svare fra deres lager af gemte ruter, hvorfor der vil blive en stor lokal trafik, der kan forårsage forstoppelse i netværket. Derfor skal en enhed, før den svarer med en gemt rute, vente en periode der er afhængig af den fundne rutes længde (målt i antal enheder i ruten), således at naboer med kort rute svarer først (de nøjagtige detaljer er ikke relevante her, men kan ses i Johnson et al. (2001)). Mens enheden venter på at afsende, skal den lytte efter datapakker fra kilden, som viser, at kilden er begyndt at kommunikere med den forespurgte destination via en anden rute, opsnappes sådanne pakker, undlader denne nabo at fortælle om sin rute til destinationen, da kilden åbenbart har fundet en anden (og kortere) rute, som den benytter.

En yderligere udvidelse af forespørgsels-metoden, er tilføjelsen af en grænse for rutelængden. Dvs. en grænse for hvor langt ud i netværket (målt i antal passerede enheder) en forespørgsel må bevæge sig. Når forespørgslen indeholder en rute der har nået dette maksimum, men stadig ikke er nået til destinationen, bliver den droppet. Dette forhindrer, at en forespørgsel breder sig til hele netværket, men kan selvfølgelig også betyde, at destinationen ikke nås. Grænsen kan benyttes til kun at spørge naboer om en rute (som de kan kende til i forvejen, eller måske

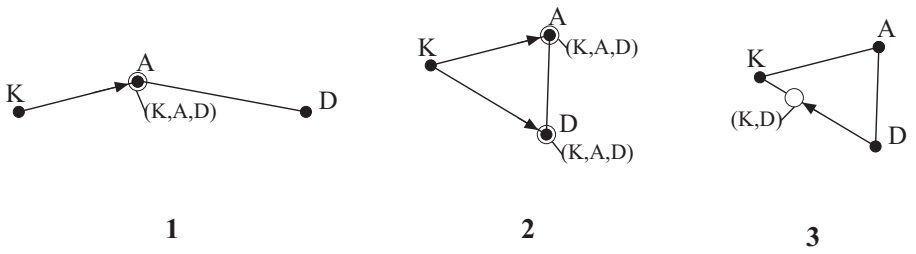
er destinationen en nabo), eller til trinvis at udvide forespørgslen til større og større dele af netværket, indtil destinationen findes (dette er en langsommere forespørgselsmetode, der til gengæld kan undgå en del unødig netværkstrafik).

Hidtil har algoritmen været beskrevet således, at enheder kun kan lære ruter at kende ved at udsende forespørgsler. En anden oplagt mulighed er, at en enhed også gemmer de ruter den lærer at kende ved at videresende datapakker, idet den gemmer den rute som pakken skal sendes af, og hvis nettet garanterer tovejsforbindelser, kan en enhed også benytte informationen i modtagne forespørgselspakker ved at vende ruterne i disse om, og derved opnå ruter til kilden, samt de enheder forespørgslen har passeret undervejs. Endelig kan en enhed selvfølgelig gemme ruteinformation fra svarpakker, når den videresender disse. En yderligere udvidelse af disse muligheder ligger i, at enheden ikke blot opsnapper ruteinformation fra datapakker, der går via den selv, men også opsnapper ruteinformation fra datapakker den modtager, fordi en af dens naboer videresender dem. F.eks kan enhed A i figur 4.2 lære en alternativ rute (A,C,B,K) hvis den hører enhed C videresende D's svar på forespørgslen, også selvom dette svar ikke går over A, men over B. På denne måde har hver enhed flere muligheder for at få opbygget et omfattende kendskab til netværket, uden at sende ekstra forespørgsler.

En anden mulighed for udbygning af algoritmen i forbindelse med ruteopbygning, er at gemme *negativ information*. Dette skal her forstås som, at når en enhed modtager en *fejlpakke*, hvori det jo er angivet, at en given forbindelse er afbrudt, gemmer den denne information i en kort periode, og sikrer sig på denne måde mod at modtage ugyldige ruter, f.eks. hvis den udsender en forespørgsel, og en nabo kender en rute via det ugyldige link (fordi naboen ikke har modtaget fejlpakken), og derfor svarer fra sit lager med denne rute. Det er nødvendigt med en forholdsvist begrænset levetid af sådan negativ information, da den afbrudte forbindelse jo kan komme til at fungere igen, hvorfor ruter af denne forbindelse igen skal accepteres som brugbare.

Hvad angår *rutevedligeholdelse*, er der igen nogle optimeringsmuligheder til den grundlæggende funktionalitet. Først og fremmest er det muligt for alle enheder, der videresender en fejlpakke, eller som er naboer til en enhed, der videresender den, og derfor kan opfange, at den bliver sendt, at benytte informationen, om den afbrudte forbindelse, til at opdatere deres egne gemte ruter. På denne måde bliver information om brudte ruter hurtigere kendt i hele nettet. Endvidere kan yderligere udbredelse af fejl-pakkerne vedtages. F.eks. kan en fejlpakke af den oprindelige kilde sendes til alle dens naboer, eller den kan udbredes af den rute, der indeholdt den afbrudte forbindelse, for at sikre, at alle enheder langs denne rute ved, at forbindelsen »længere fremme« er afbrudt. Dette medfører hurtigere tilpasning af ruter i netværket, men kræver ekstra netværkstrafik.

En mulighed for vedligeholdelse, er *automatisk ruteforkortning*, der er illustreret på figur 4.3, hvor enhed K sender til enhed D via enhed A. Imidlertid flytter enhed D sig, således at den nu bliver nabo til enhed K. Dette kan enhed K imidlertid ikke umiddelbart vide, men enhed D kan pludselig opfange en pakke, som er til den, men hvor den kan se på ruten, at den skal over A først. Som reaktion på dette kan enhed D (i trin 3) sende en uopfordret svar-pakke til K, og fortælle om den nye og kortere rute, som K da kan begynde at benytte.



Figur 4.3 Illustration af automatisk ruteforkortning. I trin 1 sender K til D via A. I trin 2 er D rykket tættere til K, så disse kan nå hinanden direkte, men K kan ikke vide dette, og sender derfor stadig via A, men disse pakker kan nu opfanges af D. I trin 3 sender D en uopfordret svar-pakke til K, for at gøre denne opmærksom på den kortere rute som D har fået kendskab til via de opfangede pakker (der skulle via A)

4.2.3 Opsummering

Når en enhed (kilden K) ønsker at kommunikere med en anden enhed (destinationen D), undersøger den først om den allerede kender en rute til destinationen, er dette ikke tilfældet, igangsætter den en forespørgsel. En forespørgsel består af en data-pakke, der indeholder et identifikationsnummer, en liste over enheder den har besøgt, samt angivelse af hvilken enhed der forespørges. Forespørgselspakken sendes til alle naboer.

Når en enhed modtager en forespørgselspakke, skal den reagere således:

1. Undersøg om enheden selv er destinationen, er dette tilfældet sendes et svar til afsenderen. Svaret indeholder den i forespørgslen angivne rute, og sendes, afhængig af netværkstypen, enten via denne rute i omvendt rækkefølge (for netværk med tovejsforbindelser), eller via en allerede kendt rute (for netværk med envejsforbindelser). Hvis der er tale om et netværk med envejsforbindelser, og destinationen ikke kender en rute tilbage til kilden, skal den forespørge om en sådan rute, og sende svaret i denne forespørgselspakke.
2. Undersøg om denne forespørgsel (dvs. en forespørgsel med samme identifikationsnummer samt samme kilde og destination) er modtaget for nylig, er dette tilfældet ignoreres forespørgslen.
3. Undersøg om enheden selv indgår i den rute, der er angivet i forespørgslen (altså den rute forespørgslen har fulgt til denne enhed), er dette tilfældet ignoreres forespørgslen, da der kun ønskes løkkefri ruter.
4. Tilføj enhedens eget navn til ruten angivet i forespørgslen og udsend forespørgslen til alle naboer.

Det er ikke i Johnson et al. (2001) gjort klart hvad »for nylig« i denne sammehæng betyder (dvs. hvor længe enheden skal huske på, at den har set en given forspørgsel). Endvidere er det ikke klart, hvorledes det skulle kunne forekomme, at en forspørgsel ikke tidligere er set af enheden, men dog har enheden angivet i den sti den har fulgt. Måske skal forklaringen findes i, at hvis en forespørgsel sendes rundt i netværket i meget lang tid, kan en enhed have glemt, at den har set forespørgslen tidligere, men vil dog stadig undlade at sende den videre, hvis den selv forekommer i stien.

Enhver enhed der videresender en datapakke, er ansvarlig for at sikre, at denne bliver modtaget af den næste enhed på ruten. Hvis ikke denne modtagelse kan bekræftes, skal enheden returnere en fejl-pakke til kilden, så denne kan rette sit lager over gemte ruter, og forsøge at nå destinationen af en anden vej. En *fejlpakke* indeholder oplysninger om, hvilken forbindelse der fejlede.

5 Grafteoretisk modellering af netværk

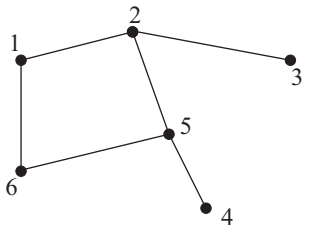
I det følgende indføres nogle grundlæggende grafteoretiske begreber, der kan benyttes til at modellere netværk. I første omgang ser vi på statiske netværk, og definerer matematiske begreber til at beskrive disse. I senere afsnit vil de dynamiske aspekter blive inddraget. Definitionerne er hentet fra Biggs (1989).

Lad os starte med at indføre det grundliggende grafbegreb. En graf består af en mængde af hjørner, samt en mængde af kanter imellem disse. I netværkssammenhæng repræsenterer hjørnerne de enkelte enheder i netværket, mens kanterne repræsenterer forbindelserne imellem disse. Hjørnerne navngives, f.eks. ved at lade mængden af hjørner være en delmængde af $\mathbb{N}$. En visuel repræsentation af en graf, hvor hjørnerne netop er navngivet på denne måde, ses på figur 5.1. Som definition vil vi her benytte:

Definition 1 (Graf)
En graf G består af en mængde V, hvis elementer kaldes hjørner, samt en mængde $E \subseteq \{\{v_1, v_2\} : v_1 \in V \wedge v_2 \in V \wedge v_1 \neq v_2\}$ hvis elementer kaldes kanter. Til tider skriver vi $G = (V, E)$

Det er muligt at ændre definitionen, således at en kant fra et hjørne til sig selv bliver mulig, men da den fysiske parrallel i modelleringen af netværk: At et hjørne har en netværksforbindelse til sig selv, ikke er hensigtsmæssig at have med, da en sådan netværksforbindelse typisk vil have andre karakteristika (f.eks. langt højere hastighed og sikkerhed for at data kommer frem), og derfor vil være årsag til en række uinteressante særtilfælde, vil vi holde os til ovennævnte definition.

Ud fra et rent matematisk synspunkt, kan det måske hævdes, at kanterne i virkeligheden er de interessante entiteter, og at det ville være enklere kun at definere en graf som en samling af



Figur 5.1 Illustration af en graf hvor hjørnerne er navngivet med naturlige tal.

kanter (der så ville implicere tilstedeværelsen af visse hjørner). Imidlertid ønsker vi også at kunne modellere isolerede enheder i netværket, dvs. enheder, der ikke har nogle forbindelser, og vi finder det som netop nævnt, unødigt besværligt at arbejde med netværksforbindelser fra et hjørne til sig selv. Desuden er ovennævnte definition den gængse i diverse lærebøger (f.eks. Biggs (1989)), og vi ser ingen reel grund til at ændre ved denne.

Definitionen kunne også ændres, således at mængden af kanter E, er en delmængde af $V \times V$, hvorved der bliver tale om, at $(v_1, v_2) \neq (v_2, v_1)$. Dermed kan vi tale om retningsorienterede grafer, idet vi kan tillægge disse ordnede par betydningen: En kant *fra* 1. koordinaten *til* 2. koordinaten. Denne model er imidlertid ikke relevant i dette projekt, da vi ikke ser på netværk med envejs forbindelser.

En helt grundlæggende egenskab ved netværk, er hvorvidt der eksisterer kanter mellem givne hjørner (altså hvorvidt enhederne er direkte forbundet). Vi benytter begrebet *nabo* til at beskrive dette:

Definition 2 (Nabo)
To hjørner $v_1, v_2 \in V$ er naboer i grafen $G = (V, E)$, hvis der eksisterer $e \in E$ således, at $v_1 \in e$ og $v_2 \in e$.

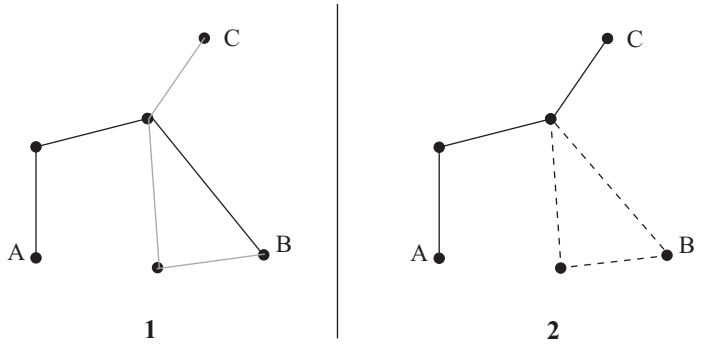
Eller med ord: To hjørner er naboer, hvis der findes en kant imellem dem, hvilket i netværkssammenhæng svarer til, at de to enheder er direkte forbundet.

Når vi taler ruteplanlægning i netværk, er vi ikke kun interesseret i, at enheder er direkte forbundne (dvs. er naboer), men også i; at de kan nå hinanden ved hjælp af mellemliggende enheder. Dermed er det relevant at indføre følgende begreb:

Definition 3 (Sti)
En sti i en graf $G = (V, E)$ er en følge af hjørner $\{v_j\}_{j \in [1;k]}$, $v_j \in V$, $\forall j \in [1; k]$, hvor v_j er nabo til v_{j+1} $\forall j \in [1; k-1]$, og intet hjørne indgår mere end én gang: Dvs. $v_i \neq v_j$ $\forall i, j \in [1; k], i \neq j$.

En sti kan opfattes som en rute gennem et netværk. I Biggs (1989) findes to beslægtede begreber: *Path* der svarer til ovenstående definition af sti, samt *walk*, hvor et hjørne godt kan indgå mere end én gang. Vi benytter her begrebet *path*, da vi normalt ikke er interesseret i ruter i netværket, der passerer flere gange gennem den samme enhed, idet sådanne ruter er længere end nødvendigt, og dermed bruger unødige ressourcer.

Lad os betegne det faktum, at der i grafen $G = (V, E)$ eksisterer en sti fra $v_1 \in V$ til $v_2 \in V$ med $v_1 \sim v_2$. Det ses, at $\sim$ er en ækvivalensrelation: Relationen er refleksiv, idet stien bestående udelukkende af hjørnet selv, er en sti fra hjørnet til det selv. Symmetrien er ligeledes klar: Hvis der er en sti fra v_1 til v_2, kan denne sti benyttes i modsat rækkefølge, hvorfor der også er en sti fra v_2 til v_1. Endelig er relationen transitiv. Hvis $v_1 \sim v_2$, og $v_2 \sim v_3$, for $v_1, v_2, v_3 \in V$, kan vi tage de to stier (fra v_1 til v_2 og fra v_2 til v_3) i forlængelse af hinanden, og dermed få en *walk*. Hvis der ikke er nogle gentagne hjørne i denne, har vi at $v_1 \sim v_3$, er der derimod gentagne hjørner, kan vi fjerne disse, ved at udelade disse 'løkker', hvorved vi igen får en sti fra v_1 til v_3 (se figur 5.2). Altså kan vi dele grafen G ind i ækvivalensklasser svarende til relationen: $\sim$. Hver af disse ækvivalensklasser vil indeholde hjørner, således at for alle v_1, v_2 i ækvivalensklassen findes en sti mellem v_1 og v_2, og for ethvert hjørne uden for ækvivalensklassen findes der ikke



Figur 5.2 Illustration af at hvis der findes en sti fra A til B og en sti fra B til C, kan der konstrueres en sti fra A til C. I 1 ses en sti fra A til B (sort) og en sti fra B til C (grå). I 2 ses hvorledes de 2 stier kan sættes i forlængelse af hinanden, hvorpå den del af den sammensatte sti der går fra det gentagne hjørne til det selv (stiplet), kan fjernes, hvorved der fremkommer en sti fra A til C

en sti til noget hjørne i klassen. Vi kalder disse ækvivalensklasser for komponenter (Biggs; 1989).

Definition 4 (komponent)
Lad $G = (V, E)$ være en graf, og lad for denne graf $\sim$ være ækvivalensrelationen givet ved, at for $v_1, v_2 \in V$ er $v_1 \sim v_2 \Leftrightarrow \exists$ sti fra v_1 til v_2. En ækvivalensklasse mht. $\sim$ kaldes da en komponent i G.

Med andre ord, så er en komponent indeholdende et givent hjørne v, en *maksimal samling af hjørner*, heriblandt v, fra grafen G, således at der findes en sti imellem ethvert vilkårligt valgt par af hjørner i komponenten. Hvis en graf består af netop én komponent, betyder det altså, at der er en sti mellem ethvert par af vilkårligt valgte hjørner. Vi siger da også, at grafen er *sammenhængende*. Når vi taler om netværk, vil vi endvidere sige at netværket er *forbundet*, når den tilsvarende graf er sammenhængende, dvs. når der er en rute mellem et vilkårligt valgt par af enheder i netværket.

Vi kan vælge at karakterisere et hjørne i en graf, ud fra hvor mange naboer hjørnet har. Dette svarer i et netværk til at se på, hvor mange direkte forbindelser enheden har, og siger dermed i en vis forstand noget om, hvor »godt« enheden er forbundet til netværket. Vi kan ligeledes benytte det gennemsnitlige antal naboer for et hjørne, til at sige noget om netværkets struktur.

Definition 5 (Valens)
I en graf $G = (V, E)$ er valensen af et hjørne $v \in V$ antallet af kanter, der har det ene endepunkt i dette hjørne. Hvis vi betegner valensen af $v \in V$ med $\delta(v)$, har vi altså: $\delta(v) = |\{e \in E : v \in e\}|$.

Det må altså gælde, at summen af valenserne for alle hjørner, bliver 2 gange antallet af kanter, dvs. for grafen $G = (V, E)$ er (Biggs; 1989, p. 163):

$$\sum_{v \in V} \delta(v) = 2|E| \tag{5.1}$$

En særligt »godt forbundet« graf, er den graf, hvor alle hjørner er naboer. Denne graf kaldes den *fuldstændige graf*. Denne graf er nyttig at kende til, da enhver anden graf kan konstrueres ved at fjerne kanter fra en fuldstændig graf. Dette benytter vi i forbindelse med beskrivelsen af stokastiske grafer, men lad os indtil videre blot indføre begrebet:

Definition 6 (Fuldstændig graf)
Den fuldstændige graf $G = (V, E)$ med n hjørner, er en graf, hvor $|V| = n$, og alle hjørner er naboer

Da alle hjørner i den fuldstændige graf er naboer, må der i den fuldstændige graf med n hjørner, være $n - 1$ kanter fra hvert hjørne, dvs. hvert hjørne har valensen $n - 1$. Dermed er summen af valenserne for alle hjørnerne (idet der er n hjørner): $\sum_{v \in V} \delta(v) = \sum_{v \in V}(n-1) = n(n-1)$, og jvf. (5.1) er antallet af kanter i den fuldstændige graf med n hjørner dermed:

$$|E| = \frac{1}{2}n(n-1) \tag{5.2}$$

5.1 Stokastiske grafer

En stokastisk graf (Random graph) er en matematisk konstruktion, der beskriver egenskaber ved en graf genereret vha. et element af tilfældighed. Stokastiske grafer benyttes til at analysere forskellige typer af netværk, herunder dynamiske computernetværk. Matematisk set er en stokastisk graf et udfaldsrum bestående af grafer. Sandsynlighedsfordelingen for disse grafer (udfald) fastlægges forskelligt i forskellige modeller. En ofte studeret model er Erdös og Rényi's $G(n, p)$-model (Newman; 2002), som vi vil se på i følgende afsnit.

Vi vil ikke beskrive teorien om stokastiske grafer på en matematisk set fuldstændig måde, dels fordi dette er meget omfattende, og er beskrevet af f.eks. Bollobas (1985), og dels fordi vi ikke får brug for denne fuldstændige teori i nærværende projekt.

5.1.1 $G(n, p)$-modellen

$G(n, p)$-modellen er en model for en stokastisk graf, og kan opfattes som en graf med n hjørner, hvor sandsynligheden for, at der findes en kant mellem to vilkårligt valgte hjørner, er den samme som for, at der findes en kant mellem to andre vilkårligt valgte hjørner. Denne sandsynlighed for, at en kant findes benævnes p, og sandsynligheden for, at en kant ikke findes bliver da $1 - p$, idet vi altid har $0 \leq p \leq 1$. Dette modelleres matematisk ved at sige, at $G(n, p)$ er udfaldsrummet bestående af alle grafer med n hjørner, og hvor sandsynligheden for en given graf med m udvalgte kanter er $p^m(1-p)^{M-m}$, hvor $M = \frac{1}{2}n(n-1)$ (Newman; 2002). Denne sandsynlighed fremkommer, idet p^m er sandsynligheden for, at en graf netop har de udvalgte m kanter, og da M er antallet af kanter i en fuldstændig graf (se (5.2)) med n hjørner er $(1-p)^{M-m}$ sandsynligheden for, at de øvrige $M - m$ kanter ikke findes.

En ofte studeret egenskab ved tilfældige grafer, er sandsynlighedsfordelingen af hjørnernes *valenser*. Sandsynligheden for, at et givent hjørne i $G(n,p)$-modellen netop har valensen k, er givet ved:

$$p_k = \left(\begin{array}{c} n-1 \\ k \end{array} \right) p^k (1-p)^{(n-1)-k}$$

Dette indses, idet binomialkoefficienten $\left(\begin{array}{c} n-1 \\ k \end{array} \right)$ er antallet af mulige måder at vælge k kanter fra et vilkårligt hjørne og til k forskellige af de andre $n-1$ hjørner. p^k er sandsynligheden for, at de k udvalgte kanter findes, og $(1-p)^{(n-1)-k}$ er sandsynligheden for, at de $(n-1)-k$ andre mulige kanter fra det valgte hjørne ikke findes.

Når antallet af hjørner n bliver stor, går *valensfordelingen* p_k mod en possionfordeling. Denne fordeling har Newman (2002) sammenlignet med en række virkelige netværk, herunder internettet betragtet som en række autonome systemer og World Wide Web forstået som det netværk, hvis forbindelser er links mellem hjemmesider. I disse og andre studerede eksempler, viser det sig, at valensfordelingen er væsentligt anderledes end poissionfordelingen, og bl.a. Newman (2002) har opstillet alternative modeller, for grafer med *vilkårlig valensfordeling*. Ligeledes opstiller han modeller, der tager hensyn til et andet aspekt af fysiske netværk som $G(n,p)$-modellen ikke håndterer, nemlig »sammenklumpning« (clustering), der er det fænomen, at hvis to hjørner har en fælles nabo, så er sandsynligheden for, at de også er naboer til hinanden større. Dette modellerer det faktum, at enheder i et mobilt netværk, der er fysisk tæt på hinanden, danner en form for »klynge«, hvor der er stor sandsynlighed for, at der kan kommunikeres direkte med de andre enheder i »klyngen«.

5.2 Faseovergange og forbundethed

I det følgende vil vi gengive et par generelle resultater fra stokastisk grafteori, som viser sig at være inspirerende i vores modelbygning. Vi vil ikke bevise resultaterne her, men blot henvise til Bollobas (1985).

Lad os sige, at hvis en *grafegenskab* Q er *monoton*, betyder det, at det gælder, at hvis grafen G har egenskaben Q, og grafen H er grafen G, med en eller flere kanter tilføjet, da har grafen H også egenskaben Q. Lad os skrive sandsynligheden for, at en graf har egenskaben Q som $P(Q)$. Da siger et resultat af Erdös og Rényi, at for monotone grafegenskaber, er der en *hurtig faseovergang* for $n \to \infty$, hvor n er antallet af hjørner:

Sætning 5.1 (Faseovergang)
Hvis antallet af kanter i grafen G er givet ved $M(n)$, som er en funktion af antallet af hjørner n i G, og Q er en monoton grafegenskab, da findes der $M(n)$ således, at $n \to \infty \Rightarrow P(Q) \to 0$, mens det for »lidt større« $M(n)$ gælder, at $n \to \infty \Rightarrow P(Q) \to 1$ (Bollobas; 1985, afsn. II.1).

Dette resultat siger altså, at hvis en stokastisk graf fremkommer ved en process, hvor vi i hvert trin tilføjer et hjørne samt et antal kanter, da findes der en kritisk hastighed, hvormed kanter tilføjes, som sikrer, at sandsynligheden for grafegenskaben kan gøres vilkårligt stor, ved at tilføje tilstrækkeligt mange hjørner. Samtidig siger resultatet, at hvis vi ikke tilføjer kanter hurtigt nok

i forhold til hastigheden, hvormed vi tilføjer hjørner, da vil sandsynligheden for egenskaben blive lille, for n stor. Der er altså tale om en skarp faseovergang.

Et interessant spørgsmål at stille i forbindelse med stokastiske grafer, kunne være hvad sandsynligheden er for, at en sådan graf består af netop én komponent (altså er sammenhængende). Vi bemærker, at hvis en graf G er sammenhængende, og vi tilføjer flere kanter til G, da er den resulterende graf stadig sammenhængende, og altså er sammenhæng en monoton grafegenskab. Dermed er det ikke så overraskende, at der findes et generelt resultat om faseovergangen, hvori en stokastisk graf skifter fra at være ikke-sammenhængende til at være sammenhængende: Hvis antallet af kanter $M(n) > \frac{n}{2}\log(n)$ da gælder, at sandsynligheden P_c for, at grafen er sammenhængende, går mod 1, for $n \to \infty$, mens $M(n) < \frac{n}{2}\log(n) \Rightarrow P_c \underset{n\to\infty}{\longrightarrow} 0$ (Bollobas; 1985). Da vi jvf. (5.1) har, at summen af valenserne er 2 gange antallet af kanter, da må den gennemsnitlige valens være: $\delta = \frac{2M(n)}{n}$ og vi har derfor, at for den gennemsnitlige valens $\delta > \log(n) \Rightarrow P_c \underset{n\to\infty}{\longrightarrow} 1$, mens det ligeledes gælder, at $\delta < \log(n) \Rightarrow P_c \underset{n\to\infty}{\longrightarrow} 0$. Vi har altså her endnu en »skarp grænse« for, hvornår en stokastisk graf er forbundet for n stor.

6 Modeller og resultater

I de følgende afsnit, vil vi se på forskellige modeller af ad-hoc netværk, samt de ruteplanlægningsrelaterede resultater vi kan uddrage af disse. Desværre har vi ikke kunnet finde nogle artikler, der, vha. analytisk modellering, analyserer en konkret ruteplanlægningsalgoritme, som f.eks. DSR. Det nærmeste vi kom dette, er Nasipuri et al. (2001), der modellerer udvidelser af DSR, der udnytter flere ruter på forskellig vis, og i denne kontekst vurderer nytten af at have flere ruter. Artiklen hævder at være unik i den forstand, at der ikke er nogen, der før den har undersøgt ruteplanlægning i MANET analytisk:

> »In our knowledge, this is the first attempt to analytically model on-demand routing protocols in ad hoc networks.« (Nasipuri et al.; 2001)

Hvis denne udtalelse står til troende, skal alle andre artikler, der forsøger sig med analytisk modellering af konkrete algoritmer, være fra efter 2001. Som nævnt har vi ikke fundet sådanne artikler. Vi vil i afsnit 6.2 se på modeller og resultater fra Nasipuri et al. (2001), men det er altså meget begrænset, hvad vi, vha. analytiske modeller, kan sige om konkrete ruteplanlægningsalgoritmer. Der findes dog en række resultater, der på en lidt anden måde relaterer sig til vores problemstilling.

En række artikler, f.eks. Gupta & Kumar (1998), Callaway et al. (2000), Dousse et al. (2002) og Xue & Kumar (2002), beskæftiger sig, på forskellige abstraktionsniveauer, med, hvad vi kunne kalde, »strukturelle spørgsmål«, dvs. spørgsmål om hvorledes strukturen i et ad-hoc netværk er og hvorledes dette påvirker netværkets egenskaber (eller for de mere abstraktes vedkommende: Hvorledes strukturen i en graf er). Netværket modelleres her ved stokastiske grafer. I denne kontekst er det så muligt at udtale sig om, hvad sandsynligheden er, for at et netværk vil have en bestemt egenskab.

Et typisk af disse »strukturelle spørgsmål« er, hvorvidt et ad-hoc netværk, med givne egenskaber, kan forventes at være forbundet. De egenskaber der analyseres, er f.eks. det gennemsnitlige antal af naboer en enhed har, enhedernes senderadius eller sandsynligheden for, at en enhed fungerer (dvs. ikke er midlertidigt ude af drift f.eks. slukket eller optaget). Udover forbundethed analyseres også på f.eks. kapacitet og flaskehalse. Disse resultater kan siges at være af en vis relevans for ruteplanlægning, idet netværkets egenskaber må være betydende for, hvorvidt det overhovedet kan forventes. At ruteplanlægning er mulig, samt sige noget om, hvilke forhold algoritmerne skal tilpasses.

Flere af de ovennævnte resultater, bygger, udover stokastiske grafer, på percolationsteori, der har sin oprindelse inden for fysikken. I denne teoribygning ses der på spørgsmål om, hvor stor sandsynligheden er, for at en given struktur kan »gennemtrænges« (percoleres). Det tidligste

eksempel på percolationsteori var en model for en gas eller væskes gennemtrængen af et komplekst materiale modelleret som en bestemt type stokastisk graf. Der kunne f.eks. være tale om spørgsmålet: Vil en porøs sten nedsænket i vand blive våd i midten (Kesten; 1982, kap. 1)? Denne teori er senere udvidet til stor generalitet, således er Callaway et al. (2000) en meget generel beskrivelse af, hvorledes det forholder sig med sandsynligheden for gennemtrængning af en vilkårlig stokastisk graf.

Alle de ovennævnte resultater lider imidlertid af en fælles vanskelighed: De udtaler sig om egenskaberne i grænsen, hvor antallet af noder n i netværket går mod uendelig. Resultaterne drejer sig således om, hvilke kriterier der skal være opfyldt, for at sandsynligheden, for en given egenskab, går mod 1, når antallet af noder n går mod uendelig. I praksis ville det være nyttigt, hvis vi kan give et estimat af, hvor stor n skal være, for at sandsynligheden, for den givne egenskab, er stor. I praksis må det være af mindre værdi, at vide, at netværket med stor sandsynlighed vil have en given egenskab for n stor, hvis vi ikke har nogen idé om, hvorvidt n skal blive urealistisk stor (så der f.eks. ikke er fysisk plads til enhederne på det begrænsede område), eller om sandsynligheden allerede er høj for en realistisk værdi af n.

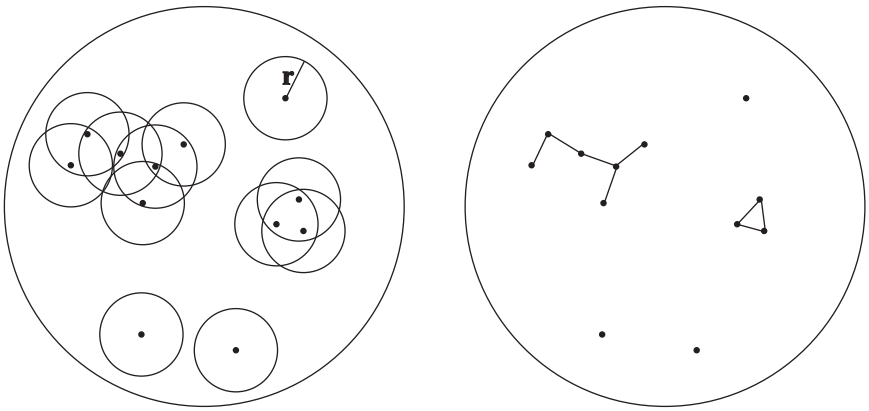
Imidlertid kan resultaterne i afsnit 5.2 lade os håbe, at resultater, der gælder for n gående mod uendelig, kan udvides til en større anvendelighed, idet der muligvis forekommer en skarp faseovergang. Resultatet om den skarpe faseovergang for forbundethed siger, at der findes en tilstrækkelig hastighed, hvormed vi tilføjer kanter, som sikrer, at grafen er forbundet i grænsen, for $n \to \infty$. Videre håber vi på, at der også er en faseovergang i n, således at for n mindre end denne, er sandsynligheden for forbundethed lille, mens for n over denne grænse, er sandsynligheden stor. Hvis vi kan vise, at denne grænse findes, og afgøre hvor overgangen sker, har vi fået resultatet transformeret til noget mere anvendeligt, nemlig at over denne grænse er der stor sandsynlighed for forbundethed, uanset om n varierer lidt.

I afsnit 6.1 vil vi opstille en model, inspireret af Gupta & Kumar (1998), hvori vi kan komme med et estimat af, hvornår et ad-hoc netværk overgår fra lille til stor sandsynlighed for at være forbundet. Forbundethed er en relevant egenskab at kigge på i forbindelse med ruteplanlægning, da forbundethed sikrer, at der faktisk eksisterer ruter mellem ethvert par af enheder, hvorfor ruteplanlægning vil være muligt.

6.1 Forbundethed af ad-hoc netværk

Et relevant spørgsmål at stille, er om det overhovedet kan forventes, at det i et ad-hoc netværk generelt er muligt at finde en rute mellem to vilkårligt valgte enheder. Hvis dette spørgsmål skal kunne besvares positivt, skal netværket have en høj sandsynlighed for at være forbundet. Vi vil derfor prøve at se på en model, hvori vi kan sige noget om sandsynligheden, for at et ad-hoc netværk er forbundet.

Gupta & Kumar (1998) foreslår en model, hvor et ad-hoc netværk modelleres som n punkter, der er jævnt fordelt i en cirkel med areal 1. Disse punkter er hjørnerne i en graf, og der er en kant mellem to punkter, hvis afstanden mellem disse er mindre end r. Dette svarer til, at et antal enheder, med senderadius r, er jævnt fordelt på et begrænset geografisk område, og er illustreret på figur 6.1. Det er intuitivt klart, at hvis r er konstant, vil vi for tilstrækkeligt stort n, kunne få en vilkårligt stor sandsynlighed P_c, for at netværket er forbundet (altså for r konstant:



Figur 6.1 Illustration af enheder med senderadius r, fordelt i et begrænset cirkulært område, og den graf som dermed opstår.

$n \to \infty \Rightarrow P_c \to 1$). Gupta & Kumar (1998) ser på spørgsmålet: Hvor hurtigt kan r, som funktion af n, aftage, samtidig med at $n \to \infty \Rightarrow P_c \to 1$ stadig holder. Det viser sig, at hvis $r^2 = \frac{\log n + c(n)}{n\pi}$, så vil sandsynligheden for forbundethed gå mod 1, for n gående mod uendelig, som ønsket, hvis og kun hvis $c(n) \underset{n\to\infty}{\longrightarrow} \infty$.

Som allerede diskuteret (i indledningen til dette kapitel), er det svært at se en praktisk anvendelse af resultater som ovenstående, hvor antallet af enheder går mod uendelig. Da forbundethed er en monoton grafegenskab, har vi imidlertid en forhåbning om en hurtig overgang fra lille til stor sandsynlighed, for at en graf er forbundet, idet vi forventer, at når tætheden af enheder er lille, så vil der kun opstå få kanter, når vi tilføjer nye enheder, og vi vil derfor befinde os under faseovergangen. Efterhånden som tætheden af enheder bliver stor, vil hastigheden hvormed kanter dannes, overstige den kritiske hastighed, hvorfor vi forventer, at grafen bliver forbundet med stor sandsynlighed. Vi bemærker, at dette ikke er et fuldstændigt argument, men blot en form for retfærdiggørelse af vores forhåbninger til modelleringens udfald. Hvis denne overgang findes, betyder det, at det er muligt at sige, at for n over en given størrelse, er sandsynligheden for forbundethed stor, og flere enheder (og dermed kanter) bidrager ikke væsentligt til at forøge denne. Vi vil i det følgende opstille en model, inspireret af Gupta & Kumar (1998), hvor vi vil prøve at finde et analytisk udtryk for P_c som funktion af n, og på denne måde udtale os om spørgsmålet om forbundethed med endeligt antal enheder.

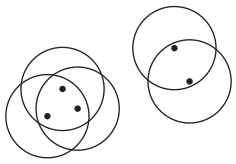
6.1.1 Forbundethed af netværk med endeligt antal enheder

Lad os se på en cirkel med radius 1 – efterfølgende er det muligt at skalere enhederne, så de passer til en konkret situtation, så radius 1 er valgt af hensyn til simplicitet i udledningen. Vi betragter nu den graf, der opstår ved at placere n punkter tilfældigt og uafhængigt i denne cirkel, og lade disse punkter være hjørner i grafen, med kanter mellem de hjørner, der har (euklidisk) afstand mindre end r. Sandsynligheden P_c, for at denne graf er forbundet, må afhænge af r og n. Vi lader r være en konstant, hvorved hver cirkel, med centrum i et af de tilfældigt placerede

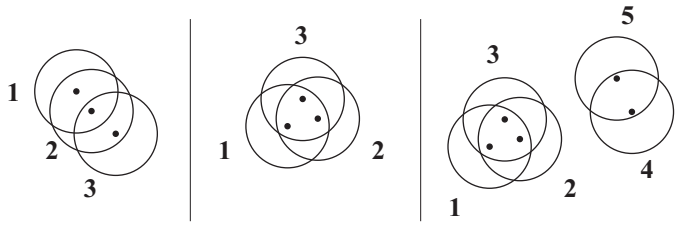
punkter og radius r, må dække arealet πr^2.

Lad os nu forsøge at finde et udtryk for sandsynligheden, for at grafen er forbundet. Et første naivt forslag, kunne være, at grafen må være forbundet, hvis punkterne er placeret i en given rækkefølge, og det gælder, at hvert punkt er placeret inden for afstand r til et af de foregående punkter, altså inden for det areal som de foregående $n-1$ cirkler dækker. Hvis $A(n,r)$ er en funktion, der angiver, hvor stort et areal n cirkler med radius r dækker, er sandsynligheden, for at en graf, bestående af n punkter, er forbundet, altså: $P_c(n,r) = \prod_{k=1}^{n-1} \frac{A(k,r)}{\pi}$. Dette betyder altså, at netværket er forbundet, hvis centrum for cirkel 2 lander i cirkel 1, centrum for cirkel 3 lander i cirkel 1 eller 2, centrum for cirkel 4 lander i cirkel 1,2 eller 3 osv. Dette er imidlertid ikke den eneste måde, grafen kan blive forbundet på! Hvis f.eks. enhed 1 og enhed 2 ikke er forbundne, men ligger, så deres sendearealer overlapper, da kan cirkel 3 lande i dette overlap, hvorved den resulterende graf bliver forbundet. Altså angiver udtrykket ikke det ønskede. I stedet forestiller vi os så, at hvis blot hver enhed er forbundet til de andre, så er netværket forbundet, dvs. hver enhed skal lande i det areal de $n-1$ andre enheder dækker med deres senderadius. Sandsynligheden for at en vilkårligt valgt enhed i netværket er forbundet til de andre, må være $\frac{A(n-1,r)}{\pi}$, og hvis vi antager, at det, at en enhed er forbundet til netværket, er uafhængigt af om de andre enheder er det (dette er ikke korrekt – hvilket vi straks vil se på), får vi at: $P_c(n,r) = \prod_{k=1}^{n} \frac{A(n-1,r)}{\pi} = \left(\frac{A(n-1,r)}{\pi}\right)^n$. Imidlertid er det klart, at uafhængighedsantagelsen ikke er korrekt, idet at hvis en enhed har en nabo, så er der nødvendigvis også en af de andre, der har det (nemlig den førstes nabo)! Desuden er der endnu et par ting, der taler mod, at dette er en god model: Det kan sagtens forholde sig sådan, at alle enheder falder inden for det areal de øvrige dækker, men at netværket ikke er forbundet. Dette betyder blot, at netværket er inddelt i flere komponenter, og er illustreret på figur 6.2. Samtidig ser vi, at $P_c(2,r) = \left(\frac{A(1,r)}{\pi}\right)^2$, men det er klart, at dette burde være $\frac{A(1,r)}{\pi}$, da sandsynligheden, for at et netværk med 2 enheder er forbundet, netop må være sandsynligheden, for at den ene enhed befinder sig inden for senderadius af den anden, hvoraf det direkte følger, at den anden enhed også er inden for senderadius af den første. Vi kan se, at i dette tilfælde gælder: $P_c(n,r) = \left(\frac{A(n-1,r)}{\pi}\right)^{n-1}$. Det gælder altså, at hver gang en enhed har en nabo, så er det givet, at endnu en enhed har en nabo, hvorfor vi ikke skal regne sandsynligheden, for at denne har en nabo ud. Vi kan altså forsøge at snakke om enheder, der »trivielt har en nabo« og så om »ikke-trivielle naboer«. Med denne sprogbrug kan vi se, at i et netværk med tre enheder, må den første enhed have en nabo, den anden enhed har da en triviel nabo, mens den sidste enhed skal have en af de to andre som nabo, for at netværket er forbundet. Altså holder formlen $P_c(n,r) = \left(\frac{A(n-1,r)}{\pi}\right)^{n-1}$ også i dette tilfælde (idet der er $3-1=2$ enheder der skal have en ikke-triviel nabo). På figur 6.3 er dette illustreret for tre enheder, samt for 5 enheder. Generelt ser det altså ud til, at før et netværk med n enheder er forbundet, kræver det, at $n-1$ af enhederne har en ikke-triviel nabo. En anden måde at udtrykke dette på, er at sige, at enhederne skal kunne tilføjes i en rækkefølge, således at med tilføjelsen af hver enhed undtagen den første, tilføjes ligeledes mindst 1 ny kant, der forbinder enheden til netværket. Lad os antage, at dette er en god model for sandsynligheden for forbundethed, og at det derfor gælder at:

$$P_c(n,r) = \left(\frac{A(n-1,r)}{\pi}\right)^{n-1} \tag{6.1}$$



Figur 6.2 Illustration af netværk delt i to komponenter, hvorved enhver enhed ligger i det areal, som de andre enheder dækker.



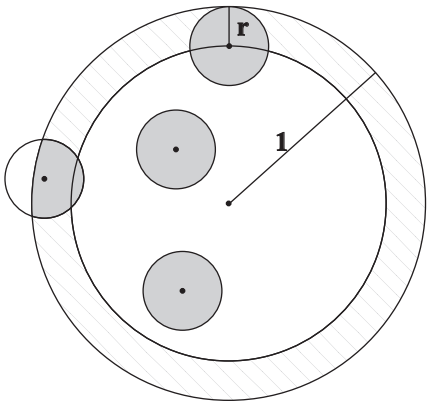
Figur 6.3 Illustration af kriterier for forbundethed. Ved tre enheder, ses det i det første illustrerede tilfælde, at enhed 1 har enhed 2 som ikke-triviel nabo, enhed 2 har enhed 3 som ikke triviel nabo, og dermed er netværket forbundet. I det andet tilfælde gælder det samme, idet vi ikke vil sige, at enhed 3 har enhed 1 som ikke-triviel nabo, idet enhed 1 og 3 allerede har en sti imellem sig, hvorfor den ekstra forbindelse mellem 1 og 3 ikke er nødvendig for at netværket er forbundet. Altså er det tilstrækkeligt, at to enheder har en ikke-triviel nabo. Til højre ses et ikke forbundet netværk med 5 enheder. Her har enhed 1, 2 og 4 ikke trivielle naboer, hvis endnu en af enhederne havde en ikke-triviel nabo, ville netværket være forbundet, og vi ville have $5 - 1 = 4$ ikke trivielle naboer ialt. Det ser altså ud til, at (6.1) også holder for 5 enheder.

Lad os nu finde et udtryk for $A(n, r)$, som altså er det areal, der gennemsnitligt vil være dækket af n enheder med senderadius r.

6.1.1.1 Arealformlen

Det forekommer umiddelbart naturligt, at $A(1, r) = \pi r^2$, hvilket imidlertid ikke er helt korrekt, da det kan forekomme, at et punkt placeres i en afstand mindre end r til periferien af den enhedscirkel, punkterne fordeles i, og derfor ikke dækker sit fulde areal inden for denne. Hvis r er meget mindre end 1, vil der være ringe sandsynlighed, for at et punkt havner i »kantområdet« (altså området nærmere end r til cirkelperiferien – se figur 6.4), og $A(1, r) = \pi r^2$ vil derfor være en god tilnærmelse. Vi har altså, at $A(1, r) \simeq \pi r^2$ idet vi husker, at for lille r gælder der næsten lighedstegn.

Hvad sker der med det overdækkede areal, når vi tilføjer flere enheder (altså for større n)? Det er klart, at jo større område af enhedscirklen der er dækket, des større er sandsynligheden, for at den n'te tilføjede cirkel, overlapper med dette, og derfor ikke bidrager med en arealtilvækst på $A(1, r)$. Det må forholde sig således, at hvis f.eks. 90% af enhedscirklen er dækket, da må den gennemsnitlige arealtilvækst, forbundet med at tilføje endnu en cirkel, være 10% af denne



Figur 6.4 Illustration af »kanten« hvori en cirkel med radius r ikke dækker med hele sit areal.

cirkels areal. Vi kan altså udtrykke arealtilvæksten for den n'te cirkel $\Delta A(n,r)$ rekursivt som:

$$\Delta A(n,r) = \left(1 - \frac{A(n-1,r)}{\pi}\right) A(1,r) \simeq \pi r^2 - r^2 A(n-1,r)$$

Idet $\frac{A(n-1,r)}{\pi}$ netop er den brøkdel af arealet (der er π), som de $n-1$ foregående cirkler dækker. Vi kan dermed opskrive $A(n,r)$ ved følgende rekursive formel:

$$A(n,r) = \Delta A(n,r) + A(n-1,r) \simeq \pi r^2 - r^2 A(n-1,r) + A(n-1,r) \qquad (6.2)$$

Ved at benytte $A(1,r) \simeq \pi r^2$ og udregne de første værdier af $A(n,r)$, fås en idé til et iterativt udtryk:

$$A(n,r) \simeq \pi \sum_{k=1}^{n} (-1)^{k+1} \binom{n}{k} r^{2k}$$

der reduceres idet vi jvf. definition af binomialkoefficienten har at $\binom{n}{n-k} = \binom{n}{k}$, hvis vi så lader $m = n - k$ får vi vha. binomialformlen følgende udregning:

$$\begin{aligned}
\sum_{k=1}^{n} (-1)^{k+1} \binom{n}{k} r^{2k} &= -\sum_{k=1}^{n} (-1)^{k} \binom{n}{n-k} r^{2k} \\
&= -\sum_{m=0}^{n-1} (-1)^{n-m} \binom{n}{m} r^{2(n-m)} \\
&= 1 - 1 - \sum_{m=0}^{n-1} (-1)^{n-m} \binom{n}{m} r^{2(n-m)} \\
&= 1 - \sum_{m=0}^{n} (-1)^{n-m} \binom{n}{m} (r^2)^{n-m}
\end{aligned}$$

$$\begin{aligned} &= 1 - \sum_{m=0}^{n} \binom{n}{m} 1^m (-r^2)^{n-m} \\ &= 1 - (1-r^2)^n \end{aligned}$$

Lad os nu bevise, at denne formel for $A(n,r)$ følger af den rekursive formel (6.2)

Lemma 1 (Arealformel)
Hvis den rekursive formel (6.2) gælder samt $A(1,r) \simeq \pi r^2$, har vi at:

$$A(n,r) \simeq \pi(1-(1-r^2)^n) \tag{6.3}$$

for alle $n \geq 1$

Bevis
Beviset forløber ved induktion. Induktionsbasis tages i $n = 1$, hvor det ved direkte beregning ses, at $A(n,r) \simeq \pi r^2$ som antaget.

Induktionsskridtet tages, idet vi antager, at (6.3) holder for $n-1$. Da vi har antaget (6.2), har vi at:

$$\begin{aligned} A(n,r) &\simeq \pi r^2 - r^2 A(n-1,r) + A(n-1,r) \\ &\simeq \pi r^2 - r^2 \pi(1-(1-r^2)^{n-1}) + \pi(1-(1-r^2)^{n-1}) \\ &= \pi(r^2 - r^2(1-(1-r^2)^{n-1}) + 1 - (1-r^2)^{n-1}) \\ &= \pi(r^2 - r^2 + r^2(1-r^2)^{n-1} + 1 - (1-r^2)^{n-1}) \\ &= \pi(1 + r^2(1-r^2)^{n-1} - (1-r^2)^{n-1}) \\ &= \pi(1 + (r^2-1)(1-r^2)^{n-1}) \\ &= \pi(1-(1-r^2)(1-r^2)^{n-1}) \\ &= \pi(1-(1-r^2)^n) \end{aligned}$$

hvoraf vi ser, at (6.3) holder for n, hvilket afslutter induktionsbeviset. □

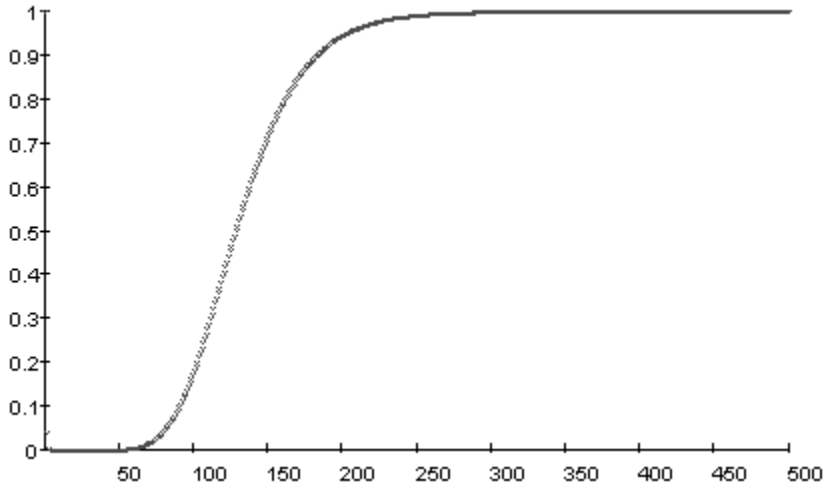
Hermed er vi klar til at se på, hvilke resultater vi kan få ud af vores modellering.

6.1.1.2 Resultater

Vi kan jvf. lemma 1, samt (6.1), give følgende udtryk for $P_c(n,r)$:

$$P_c(n,r) = \left(\frac{A(n-1,r)}{\pi}\right)^{n-1} \simeq \left(\frac{\pi(1-(1-r^2)^{n-1})}{\pi}\right)^{n-1} = (1-(1-r^2)^{n-1})^{n-1}$$

Grafer af denne funktion ses i figur 6.5 og 6.6, hvor det for fast r ses, at der er et område, hvor sandsynligheden vokser hurtigt, og efter denne overgang er sandsynligheden nær 1. Altså findes et kritisk antal enheder, således at der for dette antal og større, er stor sandsynlighed, for at netværket er forbundet. Tilsvarende ses, at for fast n findes en kritisk radius, over hvilken sandsynligheden, for at netværket er forbundet, er stor, og hvor en yderligere forøgelse af senderadius ikke forøger sandsynligheden for forbundethed væsentligt. Deraf slutter vi, at for at

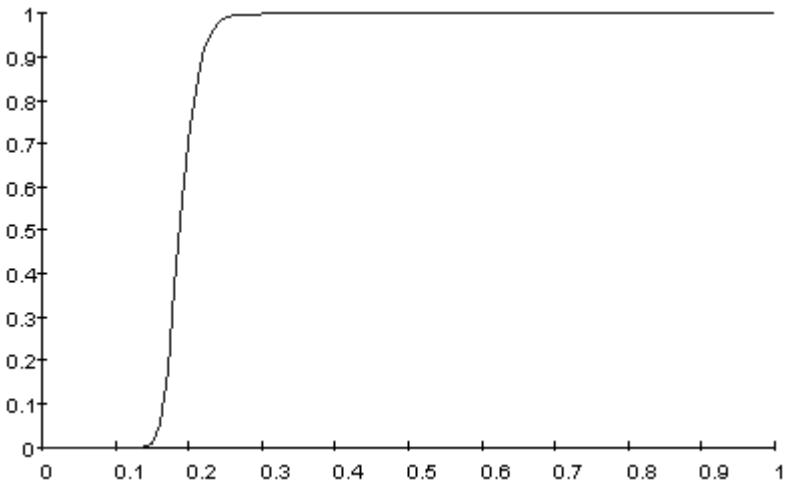


Figur 6.5 Plot af vores analytiske udtryk for $P_c(n, r)$ med fast $r = 0.2$. Førsteaksen angiver antal enheder n, og andenaksen angiver sandsynligheden for forbundethed. Det ses, at der er en relativt hurtig overgang (i n) fra lille til stor sandsynlighed for forbundethed.

sandsynliggøre, at ruteplanlægning mellem to vilkårlige enheder er mulig, kan antallet af enheder og deres senderadius afpasses, således at det sandsynliggøres, at netværket er forbundet. Det fænomen, at overgangen fra lav til høj sandsynlighed for forbundethed indtræder inden for et lille interval, betyder at det i praksis burde være muligt at foretage en realistisk afpasning, selvom estimaterne af n og r ikke er nøjagtige. Det betyder endvidere, at hvis vi i praksis observerer, at et netvæk gentagne gange ikke er forbundet, da befinder vi os sandsynligvis under faseovergangen, og kan dermed ikke håbe på, at netværket i den givne konfiguration vil være forbundet særligt ofte i fremtiden. Omvendt gælder det også, at hvis et konkret netværk i en given konfiguration er forbundet i et stort tidsrum, da kan vi forvente at være over faseovergangen, hvorfor netværket i fremtiden også vil have stor sandsynlighed for at være forbundet.

I Sun (2002) foretages computersimuleringer af en model, hvor 36 punkter er uniformt fordelt i et enhedskvadrat. Modellen svarer til den her opstillede, idet to enheder siges at være naboer, hvis afstanden mellem dem er mindre end r. Det undersøges blandt andet, hvornår den resulterende graf er forbundet. Da der er tale om en simulering, vil »kanteffekter« naturligvis gøre sig gældende. I figur 6.7 ses resultatet af simuleringen (der er ikke en analytisk modellering af det to-dimensionelle tilfælde i denne artikel, kun af et lignende en-dimensionalt tilfælde). Da simuleringen er foretaget, med enhederne fordelt i et område med areal 1, må vi forvente, at der skal π gange så mange enheder til i et område, som det vi har analyseret, dvs. med areal π. På denne måde må enhederne have samme gennemsnitlige tæthed. Vi har derfor (da $36\pi \simeq 113$) til sammenligning plottet vores analytiske udtryk for $P_c(n, r)$ for $n = 113$ i figur 6.8.

Som det ses, er der stor kvalitativ lighed mellem vores analytisk fundne sandsynlighed, og den i Sun (2002) simulerede. Det ser dog ud til, at faseovergangen indtræder for lidt større r i simuleringen, end i vores analytiske model. Dette kan muligvis forklares med kanteffekter, der



Figur 6.6 Plot af vores analytiske udtryk for $P_c(n, r)$ med fast $n = 150$. Førsteaksen viser r, og andenaksen viser sandsynlighed for forbundethed. Det ses, at der er en relativt hurtig overgang (i r) fra lille til stor sandsynlighed for forbundethed.

betyder, at vores model overvurderer sandsynligheden for forbundethed.

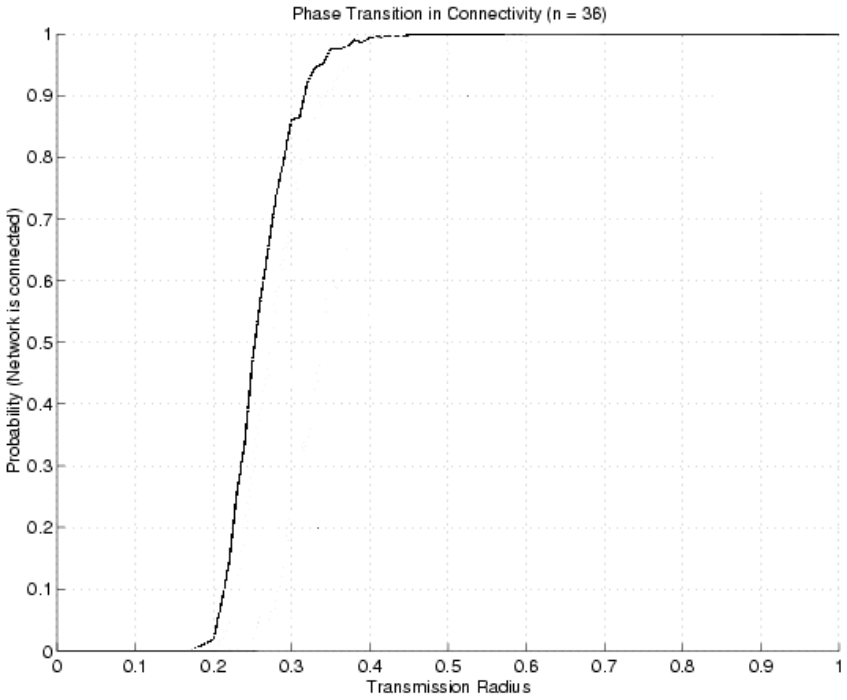
Desai & Manjunath (2002) modellerer en en-dimensional udgave af den her behandlede model. I denne en-dimensionale model, ligger alle enhederne på den reelle akse, og hvis afstanden mellem to af disse er mindre end r, anses de for at være naboer. Resultatet af denne modellering udvides til at give en øvre grænse for sandsynligheden, for at netværket, i et enhedskvadrat, er forbundet. Sandsynligheden, for at det en-dimensionelle netværk er forbundet, er jvf. Desai & Manjunath (2002) givet ved:

$$p(n, r) := \sum_{k=0}^{n-1} \binom{n-1}{k} (-1)^k (1 - kr)^n u(1 - kr)$$

Hvor $u(x)$ er step-funktionen:

$$u(x) = \begin{cases} 0 & x < 0 \\ 1/2 & x = 0 \\ 1 & x > 0 \end{cases}$$

Vi observerer, at hvis vi projicerer den to-dimensionelle graf ned på hver af de to akser, da er sandsynligheden, for at netværket er forbundet, mindre end sandsynligheden, for at begge projektioner er forbundet, idet en forbundet graf altid vil være forbundet på begge akser, mens det godt kan forekomme, at to enheder er forbundet på begge akser, men ikke er forbundet i to dimensioner (se figur 6.9). Da forbundethed på den ene akse er uafhængig af forbundethed på den anden, ses at sandsynligheden, for at grafen er forbundet, er mindre end $p(n, r)^2$ (Desai & Manjunath; 2002).



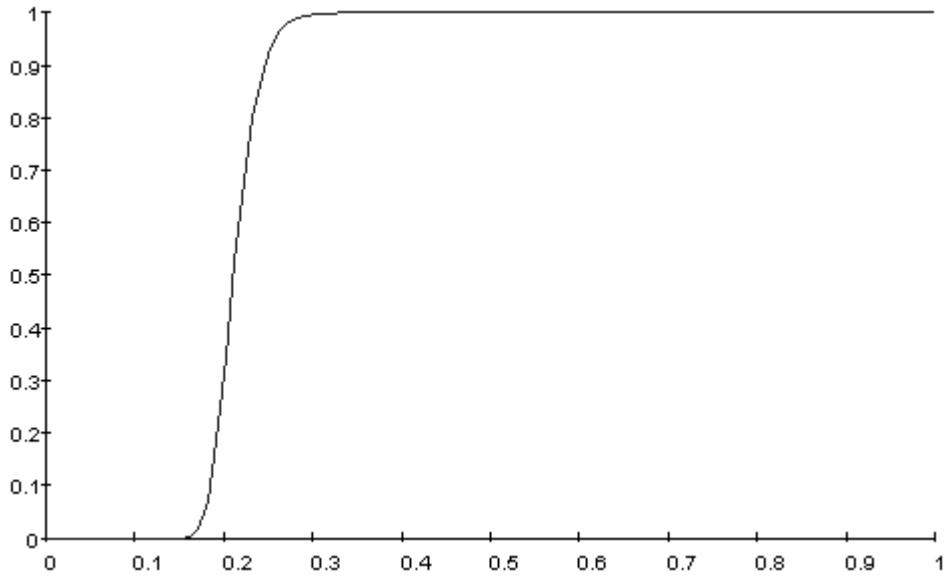
Figur 6.7 Graf fra Sun (2002) der viser simulering med 36 enheder, og angiver den simulerede sandsynlighed, for at den resulterende graf er sammenhængende. (Grafen er bearbejdet, idet den oprindelige graf også viste kurver for andre relaterede simuleringer).

På figur 6.10 ses et plot af $p(n,r)^2$ samt af vores udtryk for $P_c(n,r)$. Her ses det, at vores udtryk giver en senere faseovergang, hvorfor vores udtryk må være en bedre øvre grænse for sandsynligheden for forbundethed, end den i Desai & Manjunath (2002) givne. Samtidig ses, at graferne kvalitativt ligner hinanden, hvilket giver endnu et fingerpeg om, at vores model er fornuftig.

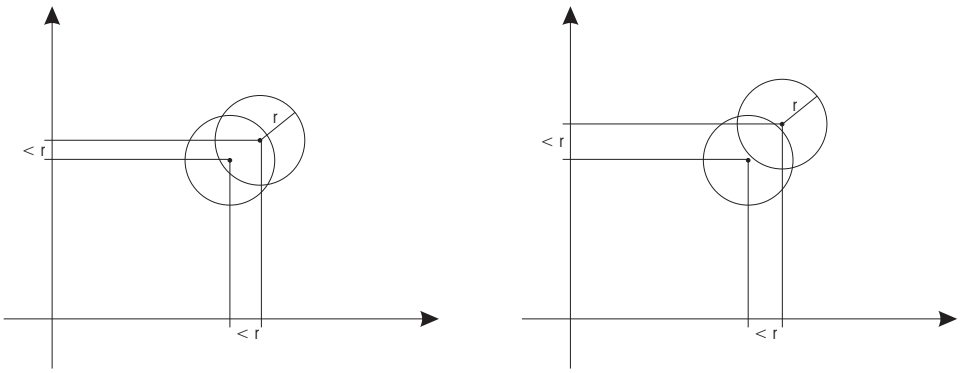
6.2 Flere ruter

Artiklen Nasipuri et al. (2001) forsøger at belyse, hvorvidt det, som ofte påstået, er fordelagtigt at have flere ruter i et ad hoc netværk. Artiklen foreslår to udgaver af DSR-algoritmen, der på forskellig vis benytter flere ruter mellem kilde og destination. At der er flere ruter, betyder i denne sammenhæng ikke, at de alle bliver benyttet samtidig, snarere virker disse ekstra ruter som et sikkerhedsnet.

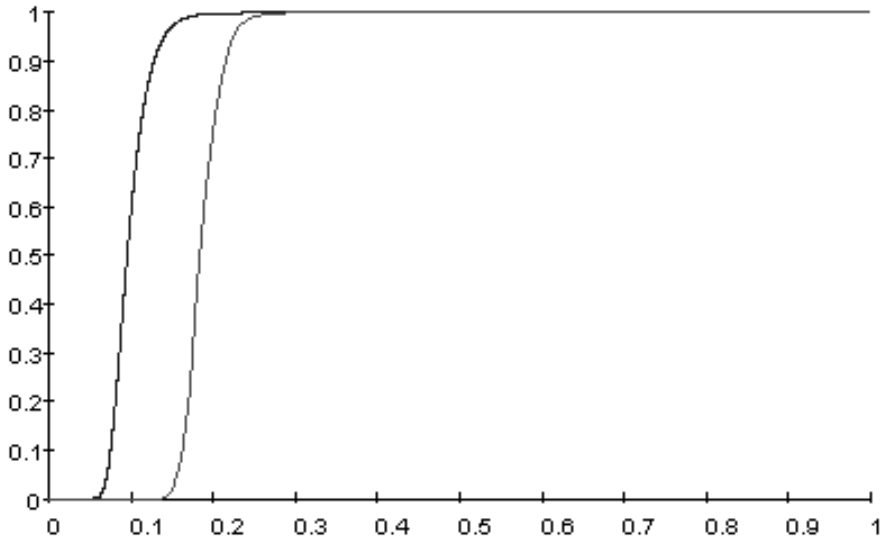
Som det fremgik af fremstillingen af DSR-algoritmen tidligere i projektrapporten, sendes forespørgselspakker til alle i netværket. Hvis netværket er stort og meget mobilt, vil der være mange af denne type pakker i netværket, hvilket betyder, at en stor del af den begrænsede båndbredde



Figur 6.8 Plot af vores analytiske udtryk for $P_c(n, r)$ med fast $n = 113$ – hvilket svarer til omtrent samme enhedstæthed som i simuleringen vist i figur 6.7. Førsteaksen angiver radius, og Andenaksen angiver sandsynlighed for forbundethed.



Figur 6.9 Illustration af, at hvis to hjørner i en graf, fremkommet ved tilfældig placering af enheder med ens senderadiusser, er naboer, da er de forbundet på begge akser, men dette kan også være tilfældet for to ikke-forbundne enheder.



Figur 6.10 Grafer af $p(n,r)^2$ som udledt i Desai & Manjunath (2002) (denne graf har faseovergang for mindst r), samt vores udtryk for $P_c(n,r)$ med fast $n = 50$. Førsteaksen angiver radius, og andenaksen angiver sandsynligheden.

går til forespørgsler.

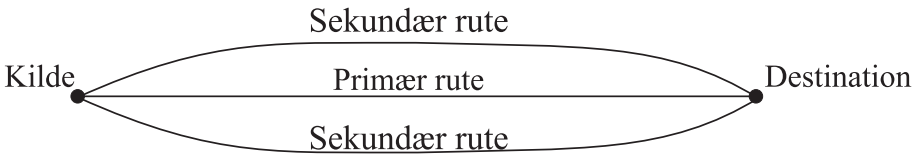
Idéen, med at have flere ruter til rådighed, er, at begrænse antallet af forespørgsler. Idet den første rute fejler kan en anden benyttes, hvorved det undgås, at der udsendes forespørgselspakker til hele netværket.

I artiklen opstilles der to forslag til, hvordan benyttelse af flere ruter kan udmøntes. Disse bliver behandlet herunder, først overordnet derefter mere i dybden.

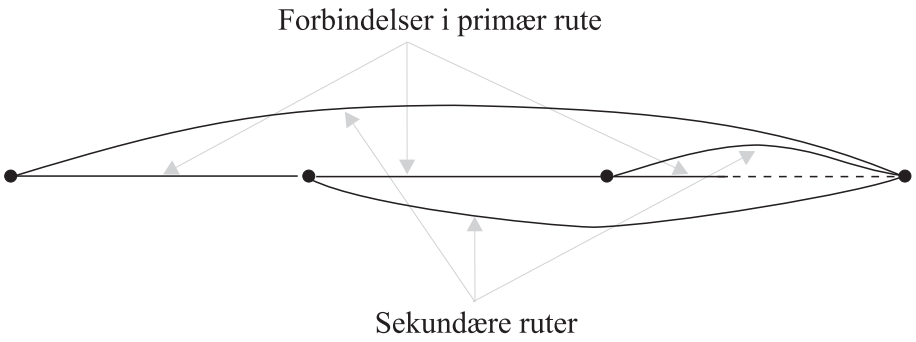
6.2.1 Algoritmer

I det første forslag er taktikken at sørge for at finde »nok« ruter mellem kilden og destinationen. Desto flere ruter desto længere tid går der, før alle ruterne forsvinder. Det er vigtigt, at disse ruter er forskellige, dvs. at der ikke findes forbindelser, der indgår i mere end én rute. Hvis mange af de mellemliggende forbindelser indgår i flere ruter, vil tab af en forbindelse betyde, at der er mange ruter der fejler. Sådanne situationer undgås, idet pakkerne indeholder ruterne, hvorved gentagelser kan opdages.

Når forespørgselspakkerne ankommer til destinationen, bliver ruterne ordnet efter den rækkefølge, de ankommer. Den første rute bliver først benyttet til datatransmission, da det antages, at den er »kortest«, hvilket vil sige hurtigst. Når denne rute bliver brudt, benyttes den næstkorteste af de tilbageværende ruter. Disse ruter er gemt i kildens hukommelse. Det er først, når alle ruter forsvinder, at der igangsættes en ny ruteforespørgsel.



Figur 6.11 Figuren viser konceptet med at have flere ruter til det samme sted. Ud over den primære rute findes der også to sekundære.



Figur 6.12 Den primære rute går gennem flere enheder, der alle har en sekundær rute til destinationen.

Det centrale ved den ovenstående protokoludvidelse er, at det kun er kilden, der har overblikket over alle ruter. Men når ruterne fejler, skal der stadig sendes en fejlpakke tilbage gennem systemet. Det medfører en vis trafik af fejlpakker, og datatab som konsekvens af at dataen, der er sendt før fejlpakken modtages, er tabt.

Det andet forslag til en protokoludvidelse tager højde for denne problematik. Tanken er: Hvis fejlpakken ikke skal rejse hele vejen tilbage, kan der spares ressourcer. Når destinationen har modtaget en række ruter (ikke nødvendigvis forskellige), sender den en pakke tilbage til hver mellemliggende enhed i den primære rute, hvor der er angivet én alternativ rute fra enheden til destinationen, forskellig fra de andre ruter[1]. Dette kan lade sig gøre, fordi alle mellemliggende enheders ruter er beskrevet i forespørgselspakkerne. Hvis en forbindelse i den primære rute fejler, bliver den sekundære rute benyttet først, hvis denne også fejler, bliver der sendt en fejlpakke, men den bliver kun sendt til den enhed, der er umiddelbart før. Således rejser fejlpakker aldrig særligt langt i netværket. Det er værd at bemærke, at hver enhed kun har to ruter til destinationen.

6.2.2 Modellering

Det der skal minimeres ved de to udvidelser, er den hyppighed, hvormed der skal forespørges efter en ny sti. Det findes der en beskrivelse af nedenunder. Her skal redegøres for nogle forhold, der gælder for begge modeller. En sti mellem kilde og destination har længden k. Længden er antallet af kanter stien består af, hvor en kant har længden 1. Formålet med at

1 Selvfølgelig er det ikke altid muligt for hver mellemliggende enhed, at der findes en ekstra forskellig rute. Dette bliver der ikke taget højde for i den analytiske model.

indføre denne betegnelse er, at forskellige stier kan have forskellige længder. I den virkelige verden er det højst usandsynligt, at alle ruter går over lige mange enheder. Det følgende baserer sig i høj grad på Nasipuri et al. (2001). Der indgår i disse beregninger, begreber og sætninger fra sandsynlighedsregningen. De vil herunder blive fremført uden egentlig argumentation.

6.2.2.1 Algoritme 1

Det antages, at der mellem kilde og destination findes en sti med k kanter, hver kant benævnes $L_i, i \in \mathbb{N}$. Levetiden for hver kant benævnes X_{L_i}. Det antages, at kanternes levetid ikke er afhængig af hinanden, og er eksponentielt fordelt. Endvidere har hver kant en gennemsnitlig levetid ℓ. Da en sti P bryder sammen, hvis kun én kant forsvinder, er stiens levetid $X_P = \min(X_{L_1}, X_{L_2}, \ldots, X_{L_k})$. Det vil sige, at det er den kant, der har den korteste levetid, der afgør, hvornår stien bliver brudt. Hvis kilden har N stier vil tiden, hvor der ikke er flere stier, være:

$$T = \max(X_{P_1}, X_{P_2}, \ldots, X_{P_N})$$

Da ingen af stierne deler kanter, er det fornuftigt at antage, at de ikke afhænger af hinanden. Når alle ruterne bliver oprettet første gang, udvælges der en primær rute, men når den bryder sammen, er det muligt, at nogle af de andre ruter også er brudt sammen. Hvis der er nogen tilbage, benyttes de efter tur. Så det, der søges, er et tidspunkt, hvorefter der ikke er flere ruter til rådighed. Det svarer til en værdi af T.

For at kunne beregne den forventede tid for T: $E[T]$, benytter vi, at forventet værdi er defineret således:

$$E[X] = \int_{-\infty}^{\infty} x f(x) dx \tag{6.4}$$

Her er $f(x)$ tæthedsfunktionen. For at kunne beregne tæthedsfunktionen er det nødvendigt at beregne sandsynligheden for T. Det vi ønsker at beregne i denne sammenhæng, er sandsynligheden for, at tidspunktet $t \geq T$, som er tidspunktet for fornyet ruteforespørgsel. Det kan skrives således $P(T \leq t)$. Da det gælder, at hvis A og B er uafhængige, så $P(A \cap B) = P(A)P(B)$. Hermed kan der beregnes: $P(T \leq t) = P((X_{P_1} \leq t) \cap (X_{P_2} \leq t) \cap \ldots) = P(X_{P_1} \leq t) \cdot P(X_{P_2} \leq t) \cdot \ldots = \prod_{i=1}^{N} P(X_{P_i} \leq t)$.

Den sandsynlighedsfordeling der typisk benyttes, i forbindelse med spørgsmål af denne type, er den ekspontielle fordeling. Den har formen $P(Y \leq t) = 1 - e^{-\lambda t}$, her er det klart, at $t > 0$. λ er her et positivt tal, der siger noget om intensiteten af hændelserne. I dette tilfælde vil det sige, at λ udtrykker stiernes længder i forhold til, hvor længe en kant i gennemsnit »lever«. Altså vil vi for en fast gennemsnitlig levetid af kanterne have, at når stierne bliver længere, vil hyppigheden, hvormed en sti bryder sammen (λ), øges. Da det tillades, at stierne har forskellige længder, må der skelnes mellem hvilke λ, der passer til de forskellige stier. Derfor er formlen for P i dette tilfælde $1 - e^{-\lambda_i t}$, hvor $\lambda_i = {}^{k_i}/_{\ell}$. Her er k_i længden af den i'te sti, og ℓ er den gennemsnitlige levetid for kanterne. Dermed bliver sandsynlighedsfordelingen for stiernes levetid:

$$P(T \leq t) = \prod_{i=1}^{N} 1 - e^{-\lambda_i t} \tag{6.5}$$

Tæthedsfunktionen står i den relation til sandsynlighedsfordelingen, at den er sandsynlighedsfordelingens første afledede. Derfor differentieres udtrykket (6.5) med hensyn til t, for at få tæthedsfunktionen, og der fås:

$$f_T(t) = \sum_{j=1}^{N} \left(\lambda_j e^{-\lambda_j t} \prod_{i=1, i\neq j}^{N} 1 - e^{-\lambda_i t} \right)$$

Nasipuri et al. (2001) ser på det tilfælde, hvor der er to stier mellem kilden og destinationen. Artiklen giver følgende formel:

$$E[T] = \frac{\lambda_1^2 + \lambda_2^2 + \lambda_1\lambda_2}{\lambda_1\lambda_2(\lambda_1 + \lambda_2)} \tag{6.6}$$

Vi vil i korte træk vise, hvordan denne formel fremkommer ved (6.4). Først bemærkes det, at $t > 0$, hvorved den nedre grænse på integralet kan ændres til 0. Der beregnes:

$$E[T] = \int_0^\infty t\left(\lambda_1 e^{-\lambda_1 t}\left(1 - e^{-\lambda_2 t}\right) + \lambda_2 e^{-\lambda_2 t}\left(1 - e^{-\lambda_1 t}\right)\right) dt$$

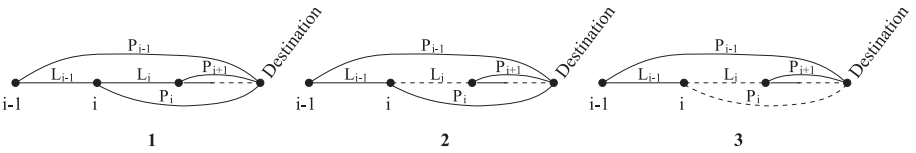
$$= \lim_{t\to\infty} \left(\frac{1}{\lambda_1 + \lambda_2} e^{(-\lambda_1-\lambda_2)t} + te^{(-\lambda_1-\lambda_2)t} - \frac{1}{\lambda_2} e^{-\lambda_2 t} - t(e^{-\lambda_2 t} + e^{-\lambda_1 t}) - \right.$$
$$\left. \frac{1}{\lambda_1} e^{-\lambda_1 t} + \frac{\lambda_1^2 + \lambda_2^2 + \lambda_1\lambda_2}{\lambda_1\lambda_2(\lambda_1 + \lambda_2)} \right)$$

Dette udtryk skal evalueres for $t \to \infty$, det er i denne sammenhæng værd at bemærke, at leddene (med undtagelse af det sidste) har formerne: e^{-at} eller te^{-at}, hvor $a > 0$. Dette skyldes, at $\lambda_i > 0$. Det er klart, at $\lim_{t\to\infty} e^{-at} = 0$, hvorfor denne type led udgår. Tilgengæld er det ikke klart, hvad te^{-at} går mod, når $t \to \infty$. Med følgende omskrivning og L'Hôpitals regel fås: $\lim_{t\to\infty} te^{-at} = \lim_{t\to\infty} \frac{t}{e^{at}} = \lim_{t\to\infty} \frac{1}{ae^{at}} = \lim_{t\to\infty} {}^1\!/_a e^{-at} = 0$. Hermed er det klart, at (6.6) er den forventede værdi ved to stier. Hvilke resultater $E[T]$ giver behandles i afsnit 6.2.3.

6.2.2.2 Algoritme 2

Denne model ser anderledes ud, da hvert hjørne har en kant og en sti tilknyttet. Kanten er et led i den primære sti, hvorimod stien er den alternative rute til destinationen. Betegnelserne L_i, for den primære kant, og P_i, for den sekundære sti, benyttes. Det der er interessant, er at kunne sige, hvornår en ny ruteforespørgsel skal påbegyndes. Umiddelbart er det klart, at hvis kildens primære kant og sekundære rute bryder sammen, så skal der ledes efter en ny sti. Lad os se på hvad der sker, hvis bruddet finder sted imellem kilden og destinationen. Vi betegner den i'te primære kant L_i. Hvis den forsvinder, så benyttes P_i i stedet, hvis den stadig eksisterer. Hvis den så også forsvinder, er der kun P_{i-1} tilbage, da den manglende efterfølgende kant medfører, at L_{i-1} er ubrugelig (se figur 6.13). Vi lader igen X_{L_i} være levetiden for kanten L_i, og X_{P_i} betegner levetiden for stien P_i. Tiden, hvor kilden igen skal søge efter en ny sti, er:

$$T = \min(\max(X_{L_1}, X_{P_1}), \max(X_{L_2}, X_{P_2}, X_{P_1}), \ldots, \max(X_{L_k}, X_{P_k}, X_{P_{k-1}}, \ldots, X_{P_1}))$$



Figur 6.13 Der vises et tilfældigt sted i et netværk. Den primære sti består af kanterne L og de sekundære ruter benævnes P. I det første tilfælde er alle kanter intakte, hvilket betyder, at den primære sti bliver benyttet. I det andet tilfælde er den primære kant L_i brudt sammen (den stiplede linie), hvilket medfører, at den tilhørende sekundære kant P_i benyttes. I det sidste tilfælde er både den primære og den sekundære kant brudt sammen, det betyder, at kanten L_{i-1} ikke kan benyttes, da den er forbundet til en blindgyde. Så nu må enheden i sende en fejlpakke til enheden $i-1$, som så benytter den sekundære rute P_{i-1}.

Grunden til, at det skal være det mindste er, at $\max(\ldots)$ leddene betegner brud, der begynder et sted ude i grafen, og som fører til et nedbrud af al kommunikation mellem kilde og destination. Dvs. at en vilkårlig af $\max(\ldots)$ leddene er nok til at fremkalde en ny søgning efter destinationen.

Ses der på leddene af formen: $\max(X_{L_q}, X_{P_q}, X_{P_{q-1}}, \ldots, X_{P_1})$ (hvor $q \in [1;k]$), ses det, at disse led har samme form for tæthedsfordeling som for algoritme 1, med den undtagelse, at der optræder et X_{L_q}, og denne har længden 1, da det er en enkelt kant. Herefter foretager Nasipuri et al. (2001) nogle udregninger af tæthedsfordelingen, men der er intet analytisk resultat for $E[T]$, der bestemmes nummerisk.

6.2.3 Resultater

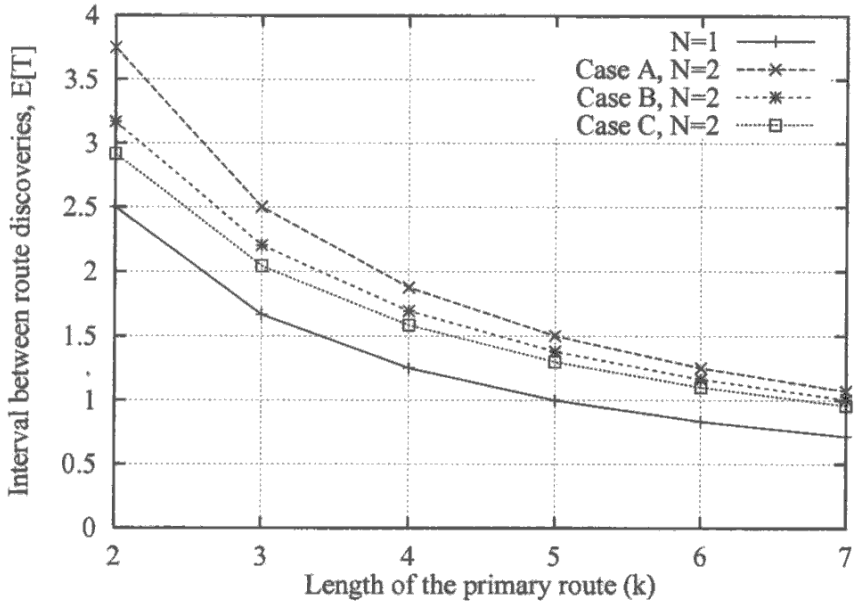
Der undersøges tre tilfælde (A,B og C), hvor det eneste der ændres er længden af hver sti. I tilfældet A er alle stier k lange. For tilfældet B er den første sti k lang, og den i'te sti er 1 længere end den $i-1$'ste sti. I tilfældet C er den første sti ligeledes k lang, mens den i'te sti er 2 længere end den $i-1$'ste sti. I det følgende vil udtrykket ydelse blive benyttet til at beskrive algoritmernes evne til at udskyde en fornyet ruteforespørgsel, desto længere der er mellem ruteforespørgslerne desto bedre ydelse.

For algoritme 1 vises det, at der er en forbedring i den tid, der går, før algoritme 1 skal lede efter nye stier. For tilfældet A med to ruter, dvs. med $\lambda_1 = \lambda_2 = {}^{k}/_{\ell}$, ser vi af (6.6) at:

$$E[T] = \frac{{}^{1}/_{\ell^2}(k^2+k^2+k^2)}{{}^{1}/_{\ell^3}(k^3+k^3)} = \frac{\ell(3k^2)}{2k^3} = \frac{3\ell}{2k}$$

For en rute P har vi: $E[X_P] = \frac{1}{\lambda} = \frac{\ell}{k}$. Heraf ser vi, at forholdet mellem den forventede levetid for to og en rute er $\frac{3}{2}$, altså får vi 50% forbedring i levetiden, når vi går fra en til to ruter.

Figur 6.14 viser, lige som ovenstående beregning, at desto kortere den primære rute desto længere tid går der før en ny forespørgsel. Det benyttede afstandsmål er stadig antallet af kanter mellem kilde og destination. Dette er på sin vis fornuftigt at benytte, da alle enheder har en senderadius, de andre enheder skal være indenfor. Derved bliver afstandsmålet også en tilnærmelse til den fysiske afstand. Der er dog det forhold, at hvis kilde og destination er placeret »ved siden af« hinanden, kan de stadig benytte en tredie part til datatransmissionen. Desuden forventer vi, at jo længere ruten er efter dette afstandsmål, des hurtigere bryder den sammen.



Figur 6.14 Graferne viser forholdet mellem den primære rute og den forventede værdi for fornyet ruteforespørgsel for algoritme 1. N er antallet af ruter. Den gennemsnitlige levetid for hver forbindelse er $\ell = 5$. Beregningerne er foretaget bl.a. på bagrund af (6.6). Grafikken er lånt fra Nasipuri et al. (2001).

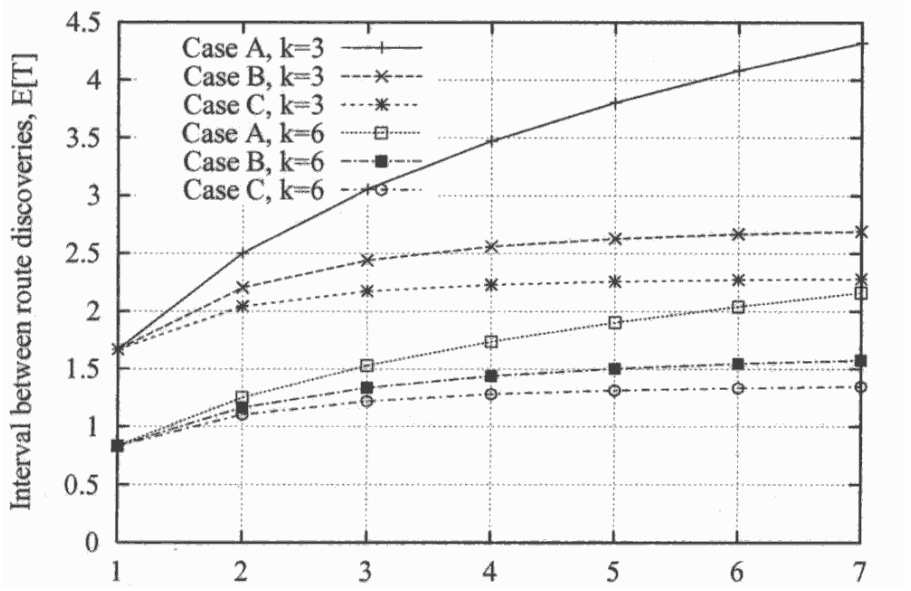
Det bør bemærkes, at undersøgelsen er lavet for $N = 2$, dvs. to ruter. Artiklen giver udtryk for, at der er ringe forbedring ved tre ruter.

Figur 6.15 viser antallet af ruter i forhold til den forventede værdi for ruteforespørgsel for algoritme 1. Der ses en kraftig effekt fra 1 til 3 ruter, hvorefter graferne flader ud. Den eneste undtagelse er tilfældet A, hvor der vedbliver med at være en effekt. Tilfældet A er dog ikke det mest realistiske tilfælde. Det er en fornuftig antagelse, at den primære rute er en mere direkte vej fra kilde til destination, hvorimod de andre ruter bliver længere, da de er mere indirekte, dvs. går en omvej. Hvad angår forskellige rutelængder, er der i figur 6.15 plottet to tilfælde, et hvor længden af den primære rute er 3 og et andet, hvor længden af den primære rute er 6.

Hvad angår algoritme 2 (se figur 6.16), er ydelsen afhængig af hvor mange hjørner stien består af. Samtidig er det relevant, hvor lange de sekundære ruter er. I dette tilfælde er længden af de sekundære ruter opdelt i tre tilfælde: A, B og C. Som ved algoritme 1 er den sekundære forbindelse henholdsvis $0, 1$ eller 2 længere end den primære forbindelse fra den pågældende enhed. Algoritme 2 viser bedre ydelse end algoritme 1, men også den samme faldende tendens, jo længere den primære rute er.

6.2.3.1 Simulering

Der bliver i artiklen lavet nogle simuleringer for at underbygge resultaterne. Det der er genstand for simuleringerne er algoritme 2, hvor det bliver forsøgt at give hver mellemliggende enhed



Figur 6.15 Graferne viser forholdet mellem antallet af ruter N og den forventede værdi for fornyet ruteforespørgsel for algoritme 1. Der ses to længder af den primære rute, nemlig $k = 3$ og $k = 6$. Igen er $\ell = 5$. Beregningerne er bl.a. lavet på bagrund af (6.6). Grafikken er lånt fra Nasipuri et al. (2001)

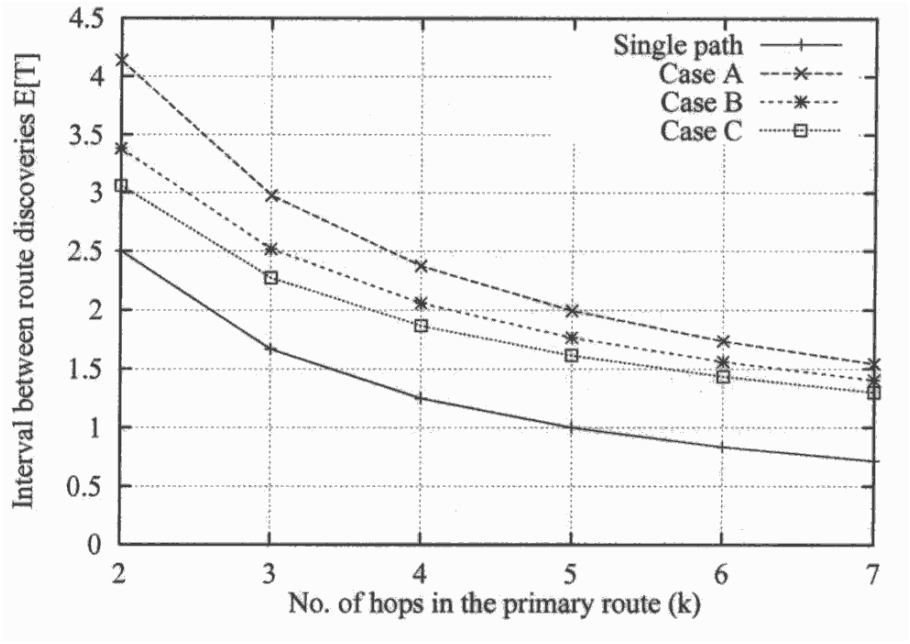
.

en alternativ rute. Der er to forhold af interesse, nemlig mængden af pakker og mobiliteten af enhederne. Der valgt to grafer ud af ialt 14, der skal sige noget om dette – henholdsvis figur 6.17 og 6.18.

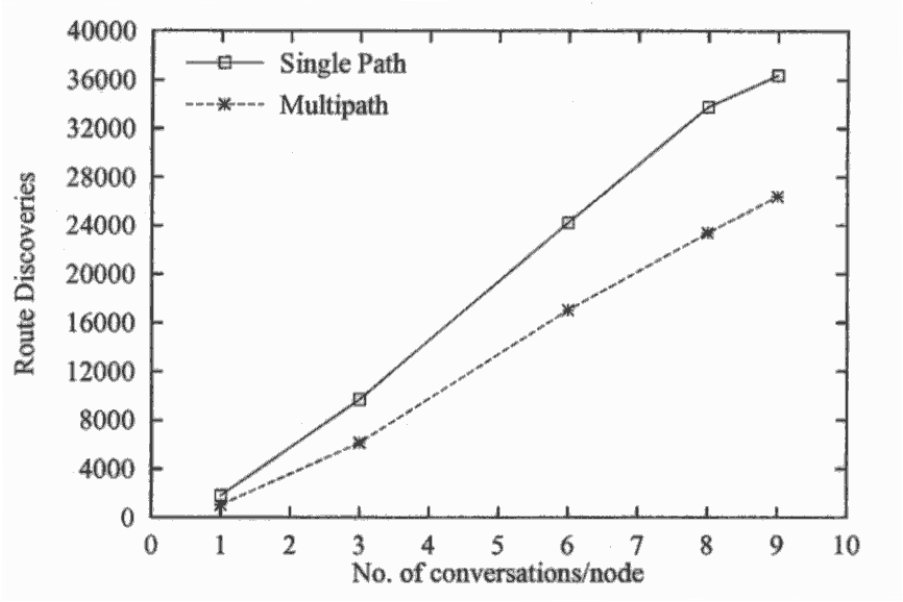
Figur 6.17 siger noget om mængden af trafik i forhold til ruteforespørgsler. Her ses der en klar forbedring i forhold til den almindelige DSR-algoritme, især ved megen trafik. Hvilket den analytiske model ikke kunne sige noget om. Det er interessant i denne sammenhæng, at en stor mængde af data kommer via sekundære ruter, idet: »Detailed instrumentation of the simulator reveals that 30-40% of delivered data packets use alternate routes.« (Nasipuri et al.; 2001, p. 346).

Figur 6.18 behandler mobilitet. Dette koncept er simuleret således, at hver enhed har en retning, som den bevæger sig i et stykke tid. Derefter stopper den op, og finder en ny retning. Det parameter der ses på her, er hastigheden, hvormed enhederne bevæger sig angivet i meter pr. sekund. Figur 6.18 viser, hvad der kan forventes: jo hurtigere enhederne bevæger sig jo flere brud på ruterne og dermed flere rute forespørgsler. Det ses tydeligt, at algoritme 2 klarer sig bedre end DSR.

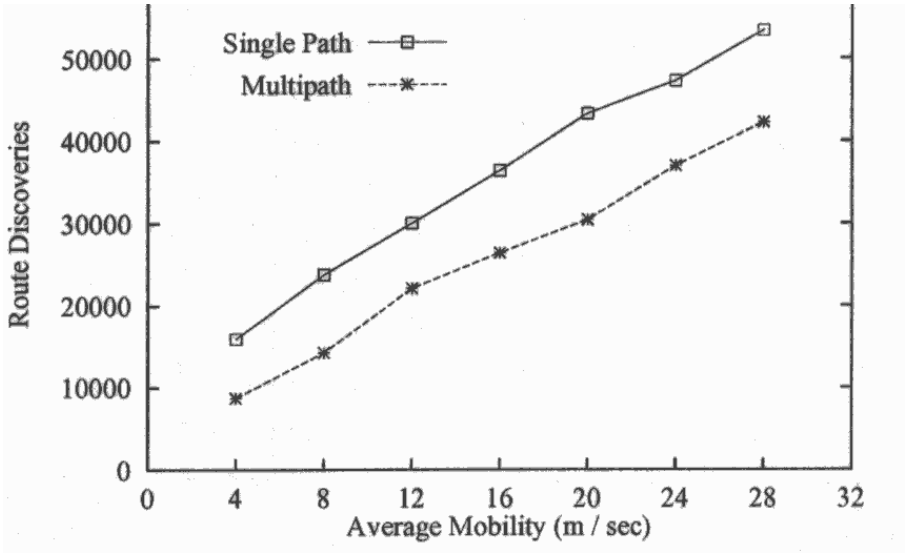
Der er dog et par forbehold der er af interesse. Det første er, at der i analysen af algoritme 2 antages, at alle enheder har én anden rute. Hvis netværket er »tyndt« på nogle punkter mellem kilden og destinationen kan det betvivles. Derfor giver den analytiske model nogle bedre



Figur 6.16 Graferne viser antallet af mellemliggende enheder i forhold til den forventede værdi for fornyet ruteforespørgsel for algortime 2. Resultaterne stammer fra nummeriske vurderinger. Grafikken er lånt fra Nasipuri et al. (2001).



Figur 6.17 Graferne viser ud fra simuleringer forholdet mellem megen trafik og antallet af ruteforespørgsler. Grafikken er lånt fra Nasipuri et al. (2001)



Figur 6.18 Graferne viser, ud fra simuleringer, forholdet mellem enhedernes bevægelseshastighed, målt i meter pr. sekund, og antallet af ruteforespørgsler. Grafikken er lånt fra Nasipuri et al. (2001).

resultater, end simulationer eller virkeligheden kan præstere. For det andet tager den analytiske model ikke højde for sendehastighed og størrelsen af datapakker. Det betyder igen, at den analytiske model viser bedre resultater.

De to algoritmer er begge tænkt som forbedringer af DSR, men det er allerede indeholdt i de forbedringer der er beskrevet i 4.2.2, at DSR kan finde flere ruter. Så det Nasipuri et al. (2001) siger noget om, er hvor stor forbedring, der fås ved flere ruter. Simuleringerne viser, hvad der kan forventes: at antallet af forespørgselspakker falder, når der benyttes flere ruter. Det samme siger de analytiske undersøgelser. Derfor vises det, der var forventet. Simuleringerne siger blot, at det lader til, at flere ruter er en god løsning.

7 Diskussion

Vi har i dette projekt set på resultater om forbundethed i MANETs, modelleret som stokastiske grafer. Dette er af relevans for ruteplanlægning, idet et forbundet netværk sikrer, at det faktisk er muligt at finde en rute mellem to vilkårligt valgte enheder. I det følgende vil vi diskutere nogle af de antagelser, der benyttes i den model for netværk med endeligt antal noder, der blev analyseret på i afsnit 6.1.

En af antagelserne var, at enhederne i netværket er jævnt fordelt, inden for et begrænset område. Denne jævne fordeling er måske ikke altid en god tilnærmelse til virkelige situationer. Man kan således sagtens forestille sig, at de personer der transporterer enhederne, f.eks. alle bevæger sig mod et bestemt mål, eller måske har en formodning om, hvor de andre enheder befinder sig, og derfor bevæger sig i retning af dem, hvis de taber forbindelsen til netværket. Man kan også forestille sig, at enhederne transporteres af personer, der af sociale eller andre årsager samler sig i grupper. Hvilken betydning en sådan gruppering af enhederne i netværket har for dets sandsynlighed, for at være forbundet, er ikke blevet undersøgt i forbindelse med dette projekt.

En anden antagelse var, at enhederne sender med en bestemt radius, og er naboer netop til de enheder der er tættere på end denne. I den forstand er der tale om en »skarp« radius, mens det i virkeligheden vil forholde sig således, at enheder kan befinde sig lige omkring den kritiske afstand til hinanden, hvorved de stadig vil kunne kommunikere, men med lav båndbredde grundet datatab som følge af den dårlige forbindelse. Der kunne så tages det forbehold at sige, at den radius der benyttes i modellen, er en »sikker« radius, og at der muligvis kan opnås længere forbindelser, men dette betyder så blot, at vi i virkeligheden har lidt større sandsynlighed, for at have et forbundet netværk, end anslået. En hårdere kritik af modellen for signaludbredelse, er at den ikke tager højde for evt. terrængenstande og højdeforskelle. Hvor stor betydning dette har, afhænger naturligvis af området netværket benyttes i.

Vi har i afsnit 4.1 fremhævet nogle designfilosofier, som benyttes for at afgøre, hvorvidt en ruteplanlægningsalgoritme er god med hensyn til de valgte kvalitetskriterier. Det ville være en del af svaret på vores problemformulering, hvis vi kunne afgøre om disse designfilosofier er fornuftige. Unødvendigheden af at finde den korteste rute og princippet om on-demand opbygning af ruter, har vi ikke set nærmere på i dette projekt. Modelbygninger der kunne udtale sig om disse designfilosofiers betydning ville være interessante.

Designfilosofien om flere ruter, har vi i afsnit 6.2 behandlet vha. en matematisk model. Her er antagelsen, at formålet med at have flere ruter, er at nedbringe antallet af nye ruteforespørgsler, for på den måde at nedbringe overhead. Dermed har vi også berørt spørgsmålet om overhead, om end dette kunne undersøges langt mere i dybden, ved også at se på algoritmernes øvrige overhead i forbindelse med fejlpakker, samt se på om en mere begrænset udbredelse af forespørgsler, f.eks. sådan som det er foreslået i forbindelse med DSR, hvor forespørgslen højst

bevæger sig gennem et begrænset antal enheder. En anden indgang til spørgsmålet, om hvorvidt det er fornuftigt at satse på flere ruter, er at undersøge hvorvidt det at benytte flere ruter, hvoraf nogle med stor sandsynlighed er langsommere end den korteste, kan betale sig i forhold til at finde og benytte den korteste. Et af argumenterne for ikke at benytte korteste-vej algoritmer, er at de simpelthen er for langsomme, hvis netværket ændrer sig hurtigt, men hvor hurtigt er det? Det kunne være endnu et interessant spørgsmål. Der er altså en del interessante spørgsmål, vi ikke har afdækket, men som nævnt har vi set på kriteriet om flere ruter, ud fra én synsvinkel.

Den matematiske model vi benytter, når vi ser på muligheden for at begrænse antallet af forespørgsler ved at benytte flere ruter, kan naturligvis også diskuteres. I modellen antages, at der altid er flere ruter tilgængelige. For algoritme 1, hvor der blot er tale om to adskilte ruter fra kilde til destination, er dette muligvis ikke en meget grov antagelse, men dog en antagelse der ikke altid er rigtig. For algoritme 2, hvor det antages, at der fra hver enhed på den primære rute findes en alternativ rute til destinationen, må antagelsen betyde, at vi regner med, at netværket er forholdsvist tæt knyttet sammen. Dette er naturligvis ikke altid tilfældet, men hvor alvorlig en indskrænkelse denne antagelse er, skal vi her lade være usagt. En yderligere – men dog mindre – anke, er at modelleringen ikke tager højde for, at der faktisk følger lidt overhead med, når man udvider til at benytte flere ruter. I algoritme 1 består overheaded af, at svar-pakken bliver større, idet den indeholder mere end 1 rute, dette må dog antages at være af begrænset betydning. I algoritme 2 er der yderligere overhead forbundet med, at kilden skal sende en pakke til hver node på den primære rute, hvori det er angivet, hvilken alternativ rute denne enhed skal benytte.

Med de ovenstående kommentarer, er det rimeligt klart, at vi ikke kan levere et fuldstændigt svar på vores problemformulering, hvilket vi da heller aldrig har turdet håbe på. Vi må derimod stille os tilfreds med, at vi kan sige noget analytisk om sandsynligheden for netværkets forbundethed, og dermed sætte et første kriterie for, hvornår ruteplanlægning er muligt. Derudover kan vi analytisk forklare, at kriteriet om, at en ruteplanlægningsalgoritme til MANETs bør benytte flere ruter, er fornuftigt – i hvert fald med de forbehold for modellens anvendelighed, som vi har diskuteret.

Dermed mangler vi stadig at komme med værktøjer til at vurdere konkrete ruteplanlægningsalgoritmer. Dette mål har vi ikke nået i dette projekt.

8 Konklusion

Vi kan konkludere, at vi i dette projekt ikke har fremfundet eller opstillet matematiske modeller, der kan benyttes til direkte vurdering af konkrete ruteplanlægningsalgoritmer. I stedet har vi set på et resultat, der angiver sandsynligheden, for at et netværk er forbundet, hvorved ruteplanlægning i netværket er muligt (afsnit 6.1). Samt på et resultat der giver et argument for, at en ruteplanlægningsalgoritme til MANETs bør finde flere ruter (afsnit 6.2).

9 Perspektivering

Der er mange muligheder for at fortsætte, det arbejde vi har påbegyndt med dette projekt. Således har vi taget et første skridt i retning af at kunne sige noget analytisk om ruteplanlægning ved hjælp af matematisk modellering, idet vi har fremfundet modelleringsværktøjer. Der er imidlertid mange områder tilbage at undersøge med disse modelleringer. I det følgende vil vi overordnet opridse nogle naturlige spørgsmål, der er opstået ud fra arbejdet med dette projekt.

I dette projekt har vi arbejdet med algoritmer specielt designet til de specielle forhold i ad hoc netværk. Imidlertid kan man sagtens forestille sig et trådløst netværk, hvor de typiske anvendelser gør, at strukturen minder meget om et statisk netværk. Tænk f.eks. på studerende eller ansatte i en virksomhed, der mødes om morgnen med hver deres bærbare computer, som de så sætter på deres bord, hvor den bliver stående det meste af dagen. Et naturligt spørgsmål er således, om vi kan opstille kriterier for, hvornår det bliver fordelagtigt at benytte de særlige adaptive algoritmer som f.eks. DSR, og hvornår de mere traditionelle algoritmer er mere effektive. En modellering der besvarer dette, f.eks. ved at udtale sig om graden af dynamik i et netværk, kunne være interessant.

Vi har set på designfilosofien flere ruter, men har også opstillet andre designfilosofier (f.eks. on-demand ruter), der kunne fortjene en tilsvarende undersøgelse ved hjælp af matematisk modellering. Derudover kunne modelleringerne behandlet i dette projekt udbygges: Modellen for forbundethed kunne udbygges med en bedre model af signaludbredelsen, hvor der ikke blot gås ud fra at signalet udbreder sig cirkulært med en fast radius, men hvor der indbygges hensyntagen til terræn m.m. Modelleringen af flere ruter kunne udbygges med overvejelser om, hvor stor betydning det ekstra overhead, i forbindelse med at finde flere ruter, egentlig har. I denne forbindelse kunne en analyse af, hvor stor sandsynligheden, for at der findes tilstrækkeligt mange ekstra ruter, egentlig er (under bestemte forudsætninger), også være interessant. Hvilke af disse modelleringer der har størst praktisk relevans, burde afgøres eksperimentielt.

Dette projekt adskiller sig fra andre modelleringsarbejder, ved i høj grad at mangle egentlige eksperimentielle data at sammenligne modellerne med. Fremskaffelsen af sådanne data er vanskelig, da det kræver opstilling og afprøvning af realistiske anvendelsesscenarier. Dette vil kræve en specificering af, hvad netværket ønskes brugt til. Det kunne være særdeles interessant, faktisk at indsamle og analysere anvendelsesdata, f.eks. ved at lade en gruppe mennesker anvende et ad hoc netværk i en periode, og indsamle trafikdata m.m. fra deres enheder.

Litteratur

Biggs, N. L. (1989). *Discrete Mahtematics*, revised edn, Oxford University Press Inc.

Bollobas, B. (1985). *Random Graphs*, Academic Press.

Callaway, D. S., Newman, M., Strogatz, S. H. & Watts, D. J. (2000). Network robustness and fragility: Percolation on random graphs, *Physical review letters* **85**(25): 5468–5471.

Cavin, D., Sasson, Y. & Schiper, A. (2002). On the accuracy of manet simulators, *Proceedings of the Workshop on Principles of Mobile Computing (POMC'02)*, ACM, pp. 38–43.
URL: *http://lsewww.epfl.ch/Publications/ById/319.html*

Corson, M., Batsell, S. & Macker, J. (1996). Architectural considerations for mobile mesh networking. Udkast til RFC.
URL: *http://tonnant.itd.nrl.navy.mil/mmnet/mmnetRFC.txt*

Desai, M. & Manjunath, D. (2002). On the connectivity in finite ad hoc networks, *IEEE Communications Letters* **10**(6): 437–490.

Dousse, O., Thiran, P. & Hasler, M. (2002). Connectivity in ad-hoc and hybrid networks, *Proc. IEEE Infocom*, New York.

Gupta, P. & Kumar, P. R. (1998). Critical power for asymptotic connectivity in wireless networks, *Stochastic Analysis, Control, Optimization and Applications* pp. 547–566.

Johnson, D. B., Maltz, D. A. & Broch, J. (2001). DSR: The dynamic source routing protocol for multi-hop wireless ad hoc networks, in *Ad Hoc Networks* pp. 139–172. Addison-Wesley, kapitel 5.
URL: *http://monarch.cs.rice.edu/monarch-papers/dsr-chapter00.ps*

Kesten, H. (1982). *Percolation Theory for Mathematicians*, Birkhauser, Boston, Basel, Stuttgart.

Kurose, J. F. & Ross, K. W. (2001). *Computer Networking: A Top-Down Approach Featuring the Internet*, Addison Wesley Longman.

Nasipuri, A., Castañeda, R. & Das, S. R. (2001). Performance of multipath routing for on-demand protocols in mobile ad hoc networks, *Mobile Networks and Applications* **6**: 339–349.

Newman, M. E. J. (2002). Random graphs as models of networks, *cond-mat* pp. 1–35.

Park, V. D. & Corson, M. S. (1997). A highly adaptive distributed routing algorithm for mobile wireless networks, *INFOCOM (3)*, pp. 1405–1413.
URL: *http://www.ics.uci.edu/~atm/adhoc/paper-collection/corson-adaptive-routing-infocom97.ps.gz*

Sun, Y. (2002). Wireless transmission characters - june report, Progress report. University of Notre Dame.
URL: *http://www.nd.edu/~nest/progress_reports/June%202002/yashan-6-16-2002.ps*

Xue, F. & Kumar, P. R. (2002). The number of neighbors needed for connectivity of wireless networks.
URL: *http://decision.csl.uiuc.edu/~prkumar/ps_files/connect.pdf*