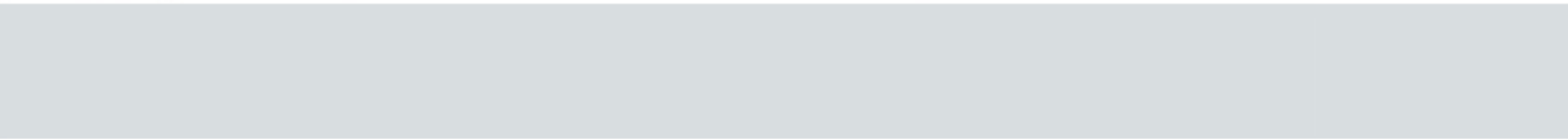




CPN-modellering og analyse af routing i mobile ad hoc-netværk

Specialeforedrag af Niels O. Jensen



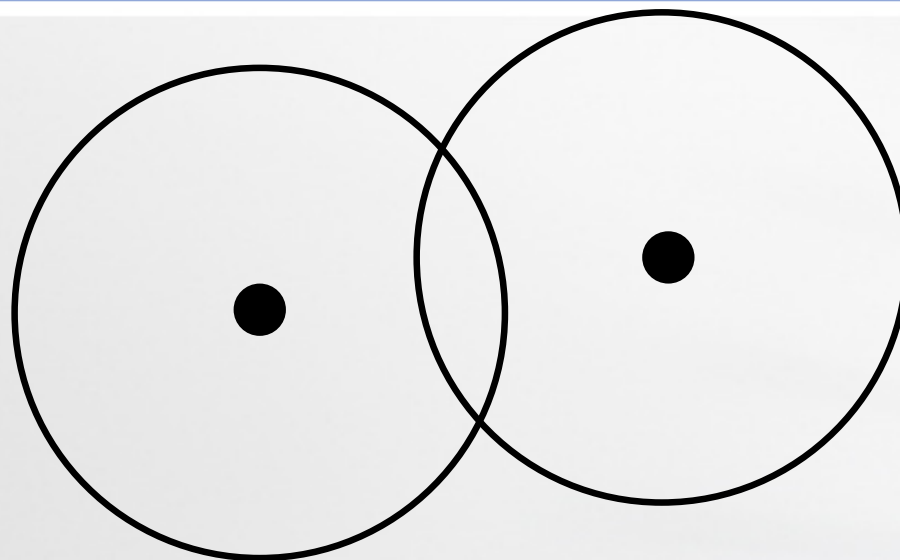
1. Præsentation af specialearbejdets hovedresultater
 - a. Kort introduktion til mobile ad hoc-netværk
 - b. Undersøgelser af **specifikationskvalitet** for "Dynamic Source Routing".
(Jeg har fundet mulige problemområder gennem en modellering af protokollen)
 - c. Undersøgelser af **valgfrie optimeringer** i "Dynamic Source Routing".
(Jeg har foretaget statistiske analyser på kørte simuleringer af modellen)

2. Ramme til vurdering og sammenligning af modellerings-/analyseværktøjer til komm.protokoller

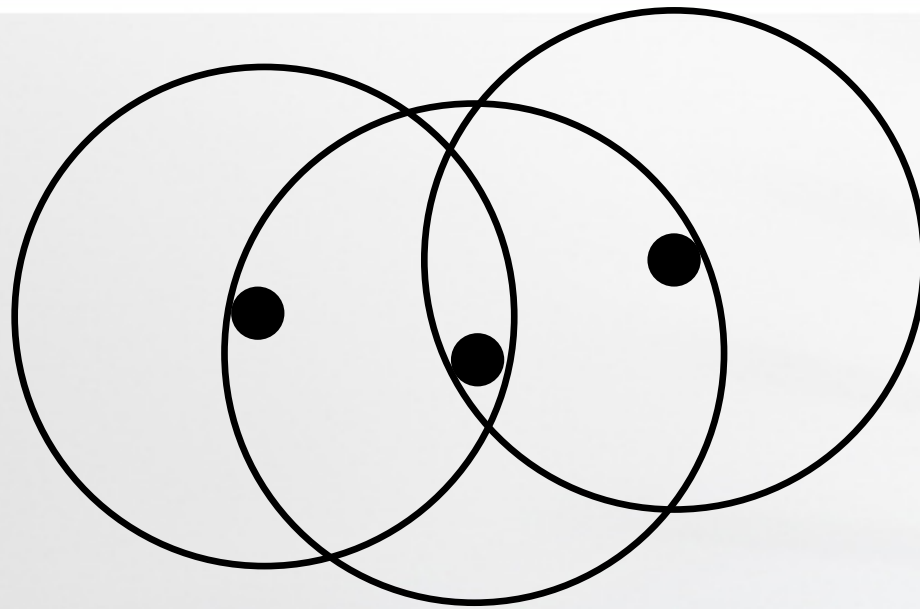
Problemdomæne

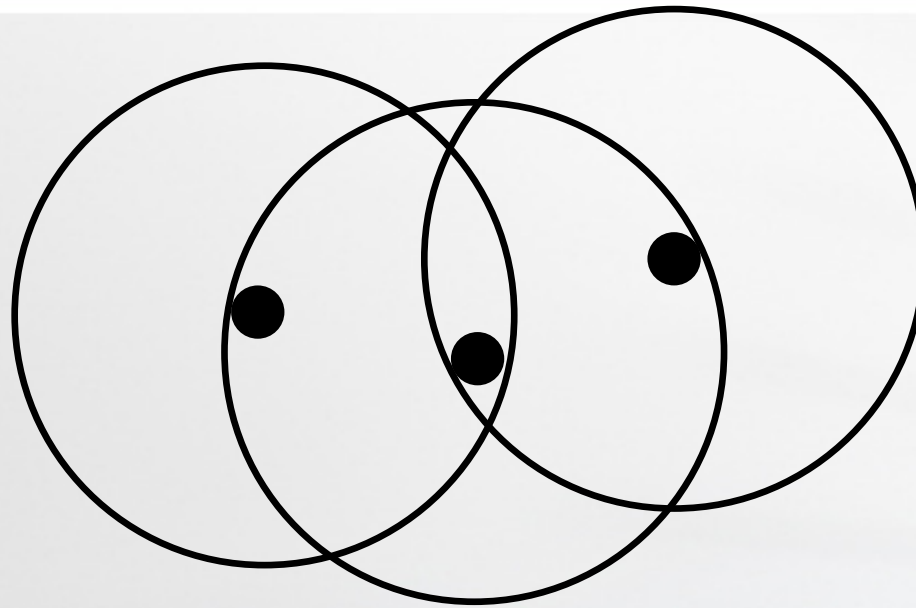


Problemdomæne



Problemdomæne





Understøttende protokol skal kunne lave et netværk, der:

- Kan benytte "multi-hop" ruter
- Understøtter mobile enheder
- Er adaptivt over for ændringer
- Er selv-organiserende

Under et: "Mobile ad hoc-netværk"-protokoller (MANET-protokoller)

Dynamic Source Routing-protokollen

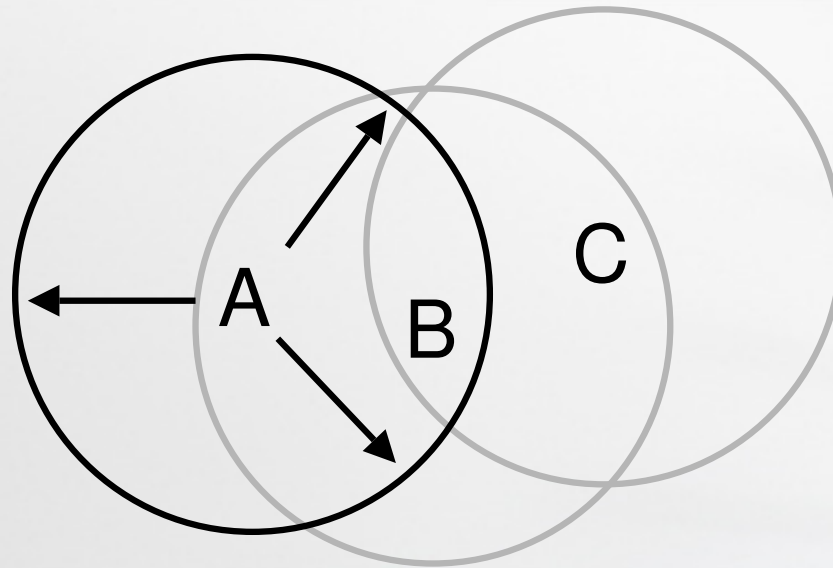
Jeg har valgt at kigge på Dynamic Source Routing (DSR).

- Topologikendskabsstrategi: "Link state".
- Informationsindhentningstidspunkt: "On demand".

DSR har tre hovedbestanddele:

1. "DSR's hovedfunktionalitet" (sende pakker via ruter)
2. "Route Discovery" (finde disse ruter)
3. "Route Maintenance" (verificere at ruter stadig virker)

”DSR’s hovedfunktionalitet” – pakke transmission 1



A udsender en pakke, der modtages af B:

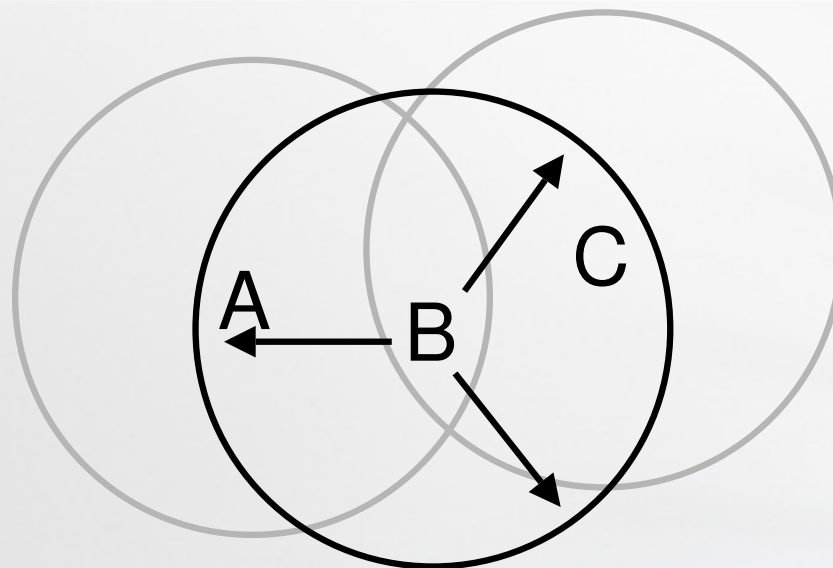
(...,

IP_fra=A, IP_til=C,

DSR_Source_Route={rute=[A,B,C], næste=B, ...},

...)

”DSR’s hovedfunktionalitet” – pakke transmission 2



B udsender en pakke, der modtages af A og C:

(...,

IP_fra=A, IP_til=C,

DSR_Source_Route={rute=[A,B,C], næste=C, ...},

...)

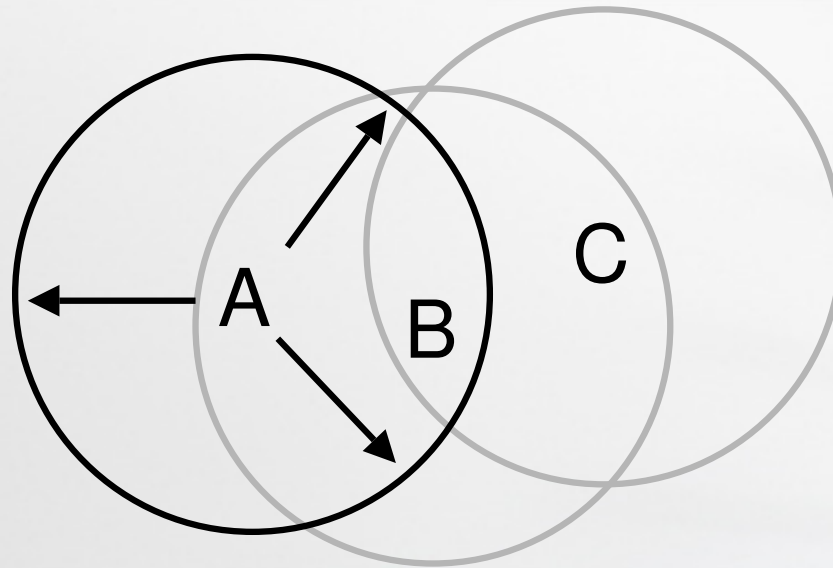
”Route Discovery”

Men hvis en knude A gerne vil sende en pakke til en knude C, og ikke har en rute dertil, hvordan finder den så en sådan rute?

I DSR sørger ”Route Discovery” for at finde denne rute.

”Route Discovery” benytter to optionsheaders:
”Route Request” og ”Route Reply”.

”Route Discovery” – requestdel



A genererer en ”Route Request”-pakke:

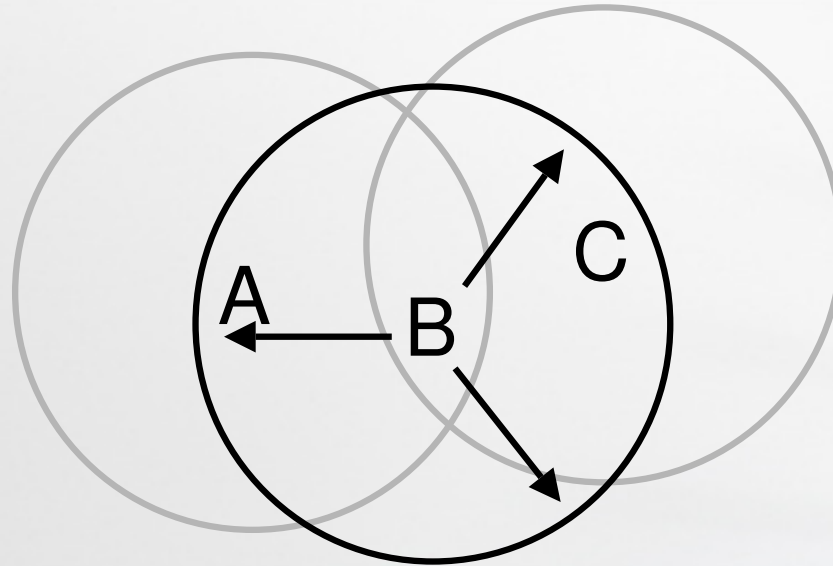
(...,

IP_fra=A, IP_til=broadcast_adresse,

Route_Request={target=C, opsamlet_rute=[A], ...},

...)

”Route Discovery” – requestdel



B sender ”Route Request”-pakken videre:

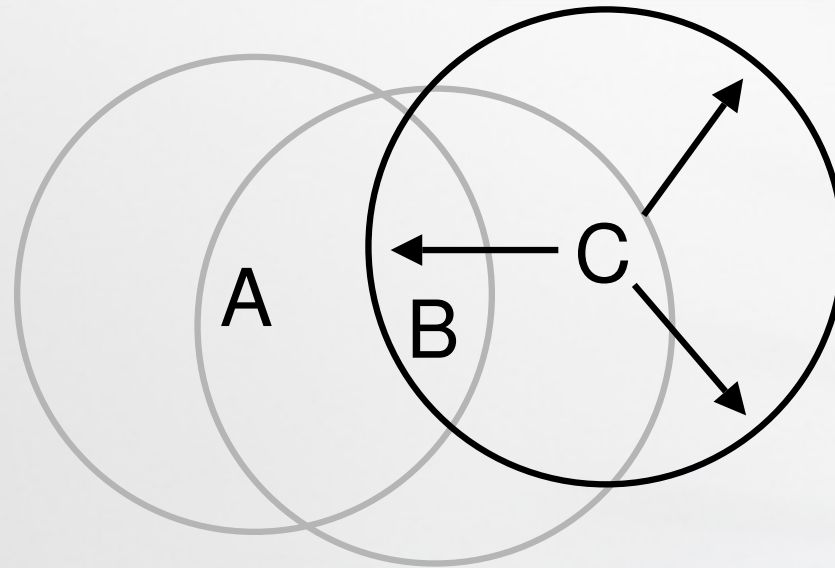
(...,

IP_fra=A, IP_til=broadcast_adresse,

Route_Request={target=C, opsamlet_rute=[A,B], ...},

...)

”Route Discovery” – replydel



Hvis C kender ruten $C \rightarrow B \rightarrow A$ til A:

(...,

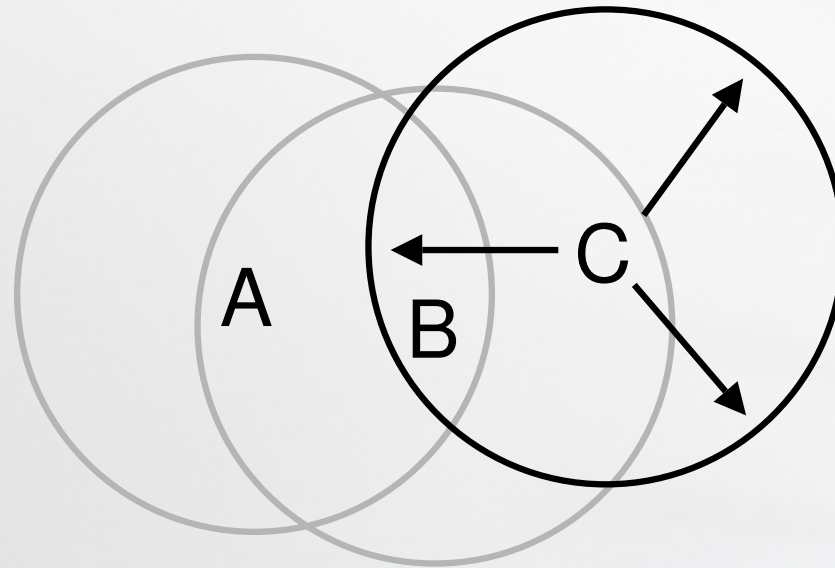
IP_fra=C, IP_til=A,

DSR_Source_Route={rute=[C,B,A], næste=B, ...},

Route_Reply={rute=[A,B,C]},

...)

”Route Discovery” – kombineret request og reply



Hvis C *ikke* kender en rute til A:

(...,

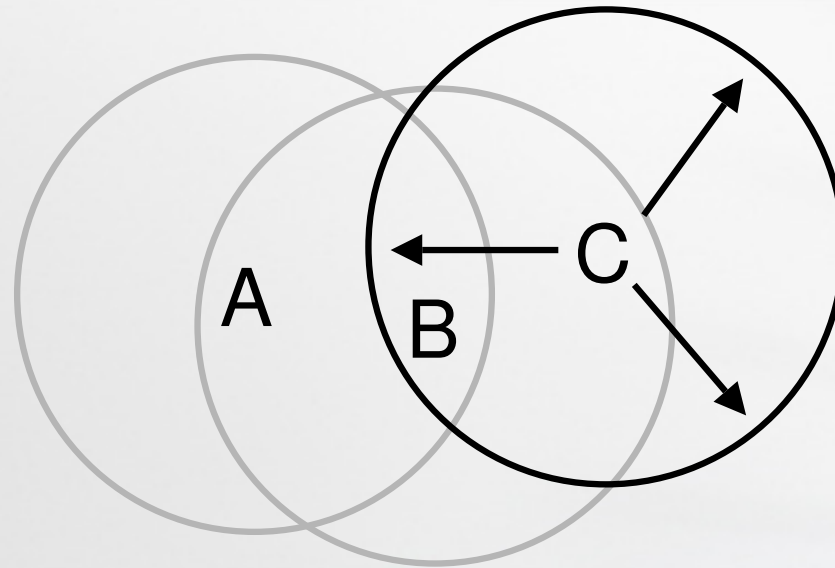
IP_fra=C, IP_til=broadcast_adresse,

Route_Request={target=A, opsamlet_rute=[C], ...},

Route_Reply={rute=[A,B,C]},

...)

"Route Discovery" – specificationspræcisering



Min specificationspræcisering: Piggybacking skal ske "omvendt":

(...,

IP_fra=C, IP_til=broadcast_adresse,

Route_Reply={rute=[A,B,C]},

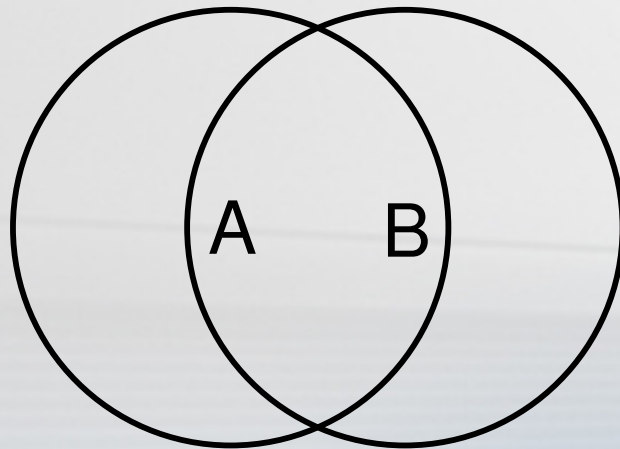
Route_Request={target=A, opsamlet_rute=[C], ...},

...)

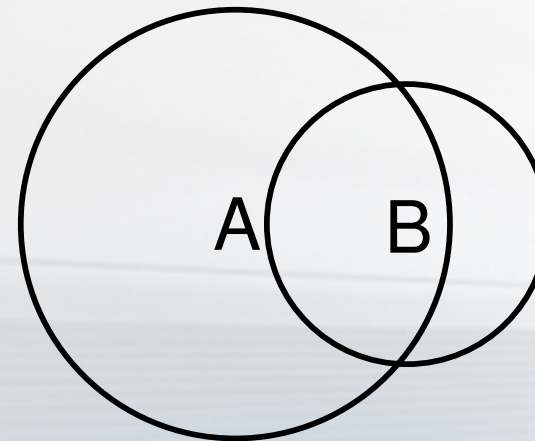
”Route Discovery” – symmetriske links

Den underliggende MAC-protokol vil i visse tilfælde kun understøtte symmetriske links:

- MAC-protokollen kan benytte sig af handshaking.
- MAC-protokollen kan benytte sig af bekræftelser.



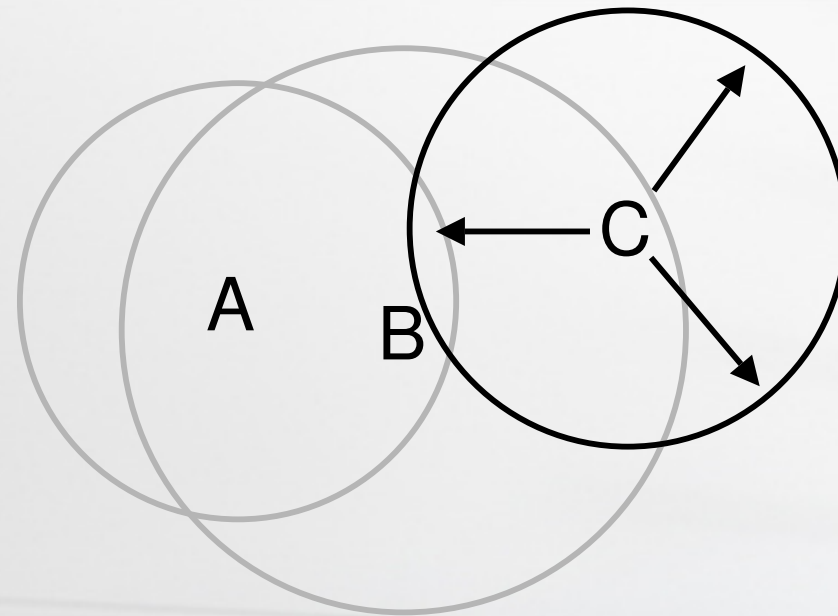
Symmetrisk link



Ikke-symmetrisk link

Hvis MAC-protokollen kun kan bruge symm. links, forsøger ”Route Discovery” at undgå ikke-symm. links.

”Route Discovery” – netværk med kun symm. links

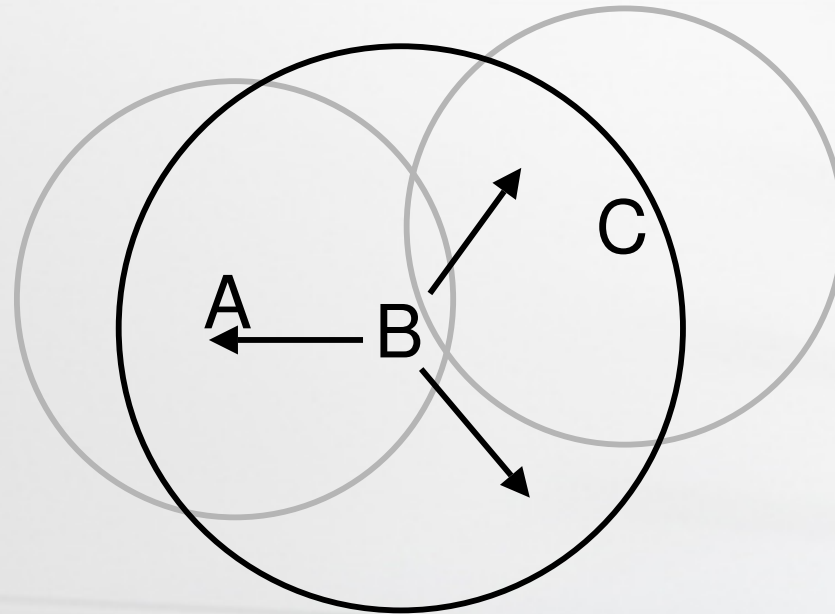


Underliggende netværk understøtter kun symmetriske links

Ved netværk med kun symmetriske links *skal* reverseret ruteliste benyttes ($C \rightarrow B \rightarrow A$) – også selvom en anden (eller ingen) haves:

```
(...,  
  IP_fra=C, IP_til=A,  
  DSR_Source_Route={rute=[C,B,A], næste=B, ...},  
  Route_Reply={rute=[A,B,C]},  
  ...)
```

”Route Discovery” – netværk med kun symm. links

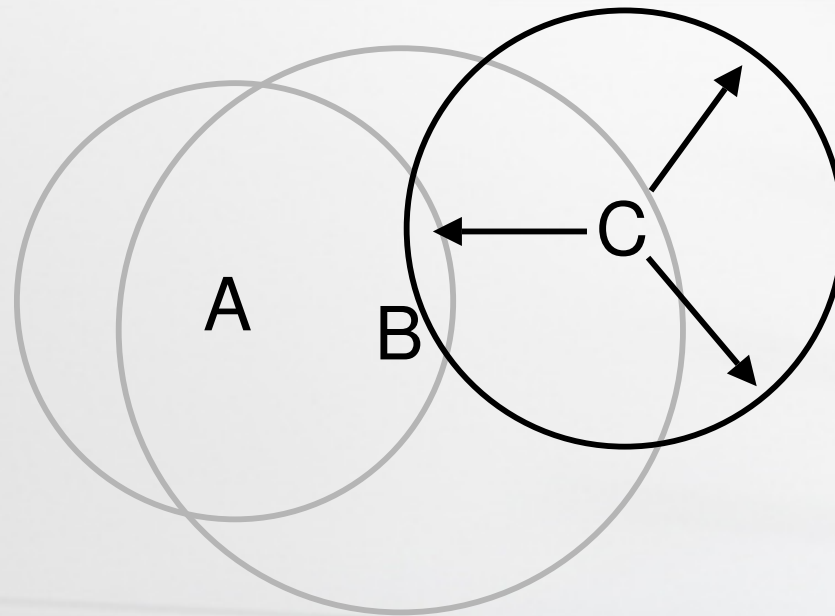


Underliggende netværk understøtter kun symmetriske links

Før C behandler request'et, kan den have benyttet forskellige teknikker til at deducere sig til, om linket er ikke-symmetrisk.

- Er linket til B ”for nyligt” fundet til at være ikke-symmetrisk?
- Er linket til B ”for et stykke tid siden” fundet til at være ikke-symm.?

”Route Discovery” – netværk med kun symm. links

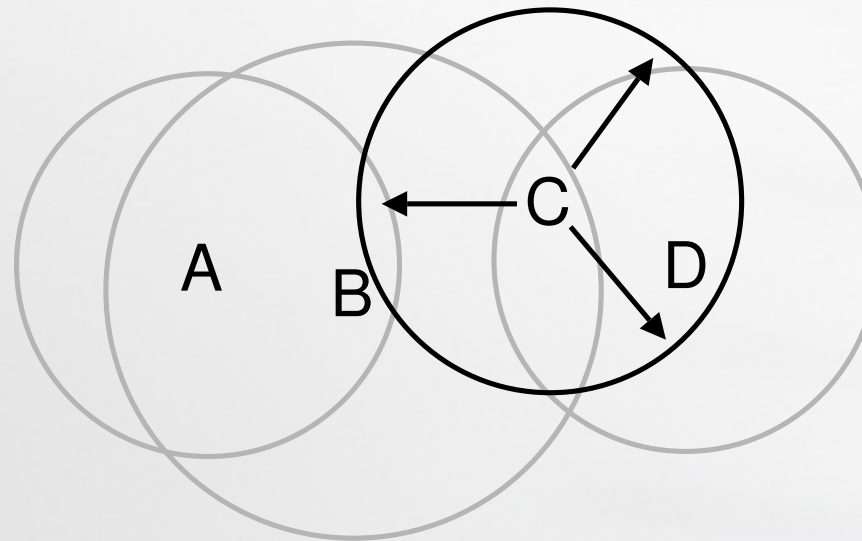


Underliggende netværk understøtter kun symmetriske links

Hvis C har vurderet linket til B ikke-symmetrisk ”for et stykke tid siden”, skal linket testes, før C må lade som om den har fået pakken fra B.

Dette gøres ved at sende et ”Route Request” til B med TTL=1. Kun B burde svare på denne.

”Route Discovery” – identificeret problemområde nr. 1

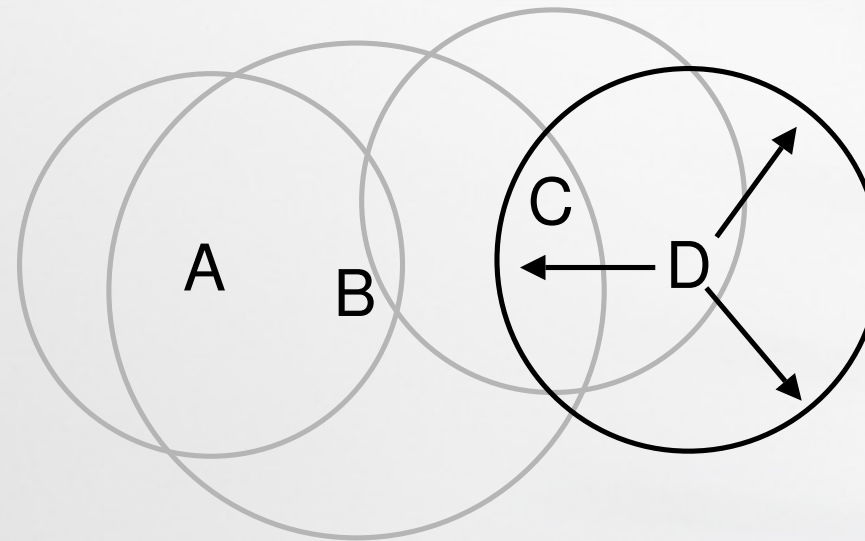


Underliggende netværk understøtter kun symmetriske links

Jeg har identificeret et problemområde her (hvor C checker linket til B):

- DSR benytter sig i høj grad af ”promiscuous mode”.
- Så selvom TTL er 1, kan enhver knude indenfor C’s antennerækkevidde modtage pakken og behandle den.
- En valgfri udvidelse til ”Route Discovery” siger, at hvis en knude får en ”Route Request” gående på en knude, den har en rute til, må den svare på det ”Route Request”.
- Dette kaldes ”Cached Route Reply”.

”Route Discovery” – identificeret problemområde nr. 1



Underliggende netværk understøtter kun symmetriske links

Jeg har fundet to forskellige løsningsmodeller til dette:

- C kan checke, at ”Route Reply” en kommer fra B
- C kan checke, at ”Route Reply” en indeholder en ruteliste af længde 0 (ekskl. afsender og modtager)

”Route Discovery” – identificeret problemområde nr. 2

Jeg har fundet et andet problemområde i ”Route Disc.”:

- ”Route Request”-pakker udsendes indtil en rute findes (inden for visse grænser)
- Udsendelsen af disse *skal* dog følge en ”backoff”-algoritme.
- Hvis en ”Route Reply” skal piggybackes til en ”Route Request”, mens ”Route Discovery” er i en sådan backoff-periode, specificeres det ikke, hvad der skal ske.

I specialet påviser jeg v.h.a. en simulation, at visse løsninger kan lede til en deadlock.

Forskellige forslag gives i specialet, men en ”rigtig” løsning er ikke fundet.

”Route Maintenance”

En MANET-protokol skal være adaptiv overfor ændringer.

Dette inkluderer f.eks., at protokollen selv skal finde ud af det, hvis en rute ikke længere fungerer.

I DSR gøres dette ved, at det er hver knudes ansvar at sørge for, at en pakke når frem til den næste modtager i en ruteliste – og evt. melde en fejl tilbage.

Til dette benyttes forskellige former for bekræftelser.

”Route Maintenance”

- Jeg har fundet et muligt problemområde i ”Route Maintenance”:
En datastruktur mangler til opbevaring af fejlbeskeder til senere piggybacking.
- Jeg foreslår desuden en optimering i specialet:
Knuders evne til at cache ruteinformation forbedres i visse situationer.
(Specifikationen gør meget ud af at udtrække/udnytte al cachebar information, hvor det er muligt).

CPN-modellen af DSR

- Både DSR's hovedfunktionalitet, Route Discovery og Route Maintenance blev modelleret
- De valgfrie optimeringer "Cached Route Reply" og "Salvage Operations" er inkluderet
- Desuden simulering af omgivende OSI-lag
 - Tilfældige knudedeflytninger og pakkeforsendelser
 - Sprog til at afspille kommandolister med knudedeflytninger og pakkeforsendelser
- I alt 35 CPN-sider med over 300 transitioner og pladser samt ca. 1400 linier SML-kode

Simulering af DSR-protokollen

- CPN-modellen af DSR blev simuleret 7 gange under hver af 12 forskellige konfigurationer:
 - Knudernes hastighed: Lav, medium, høj
 - Cached Route Reply slået til eller fra
 - Salvage Operations slået til eller fra
- Nøgletal fra simuleringerne blev udtrukket:
 - Pakkefremkomst
 - Antal pakkeovermissioner
 - Antal pakkeovermissioner før pakkefremkomst

Simulering af DSR-protokollen

- Problem: Simulationskørslerne blev relativt få (pr. konfiguration). Kan man stole på resultaterne?
- Løsning: Benytte statistisk analyse.
- Hvis man udregner et konfidensinterval f.eks. for pakkefremkomstprocentdelen, får man et interval, hvori gennemsnittet (over *alle* simulationskørsler) for pakkefremkomstprocentdelen med en vis sikkerhed vil ligge.
- Intervallet vil være smallere, jo flere måledata, der ligger til grundlag for det.
- Hvis intet kan udledes af tallene, må man prøve igen med flere simulationer.

Resultater af konfidensintervalanalyse

Middelværdien for **pakkefremkomsten** ligger med 95% konfidens mellem:

- Knudernes hastighed = lav: 96-100%
- Knudernes hastighed = medium: 84-100%
- Knudernes hastighed = høj: 65-100%

Alle tal kun med få variationer alt efter benyttede udvidelser (Cached Route Reply/Salvage Operations).

Resultat af sammenligning af observationsgrupper

Når **knudernes hastighed** sættes op, hvad sker der så med middelværdierne?

- Pakkefremkomst: Falder (80%, og ikke for "SO")
- Antal pakke transm.: Stiger (90%)
- Antal pakke transm. før fremkomst: Stiger (80%, kun for "SO" og "CRR+SO")

Med andre ord: Antal pakke transmissioner stiger (som forventet), ved de øvrige parametre kan en effekt være relativt svær at se.

Resultat af sammenligning af observationsgrupper

Når man sammenligner simulationer uden og med **Cached Route Reply**, hvad sker der så med middelværdierne?

- Pakkefremkomst: *Falder* (80%, kun for "høj hastighed")
- Antal pakke transm.: *Falder* (95%, kun "lav hastighed")
- Antal pakke transm. før fremkomst: *Falder* (95%, "lav")
- Antal pakke transm. før fremkomst: *Falder* (80%, alle hastigheder)

Med andre ord: Brugen af Cached Route Reply bliver en afvejning af, om man vil have færre pakke transmissioner ved lav hastighed eller færre fremkomne pakker ved høj hastighed.

DSR's **specifikationskvalitet** er undersøgt:

- Jeg laver en præcisering af protokollen.
- Jeg identificerer en række mulige problemområder.
- Jeg foreslår en enkelt optimering af protokollen.

Identifikationerne skete gennem modellering af DSR.

DSR's **valgfrie optimeringer** er undersøgt:

- "Cached Route Reply"'s effektivitet afhænger af knudernes hastighed.

Undersøgelser skete gennem statistiske analyser af måledata fra simulationer af modellen.

(Skete ellers kun i 14 ud af 112 artikler jvf. [KCC05]).

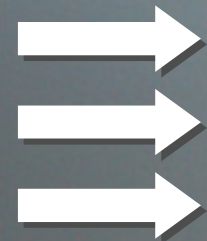
Del 2 af specialeforedraget:

Ramme til vurdering og sammenligning af modellerings-/analyseværktøjer til komm.protokoller

Hvad er vigtigt, når man skal vurdere et modellerings-/analyseværktøj til kommunikationsprotokoller fremfor et andet?

Der er en række ting, man kan vurdere hvert værktøj ud fra. De kan vurderes på en skala fra "vigtigt" til "mindre vigtigt".

Forslag til prioriteret liste af vurderingspunkter



		CPN	ns-2
2.	Verifikation gennem simulation	√	√
3.	Verifikation gennem matematisk analyse	(÷)	÷?
4.	Automatisk kodegenerering – trinvis forfining indtil komplet...	÷	√
6.	Inklusion af (model af) omkringliggende OSI-lag	÷	√

Forslag til prioriteret liste af vurderingspunkter

		CPN	ns-2
2.	Verifikation gennem simulation	√	√
3.	Verifikation gennem matematisk analyse	(÷)	÷?
4.	Automatisk kodegenerering – trinvis forfining indtil komplet...	÷	√
6.	Inklusion af (model af) omkringliggende OSI-lag	÷	√
8.	Grafisk simulering	√	(√)
9.	Grafisk modellering	√	÷

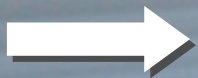


Forslag til prioriteret liste af vurderingspunkter

		CPN	ns-2
2.	Verifikation gennem simulation	√	√
3.	Verifikation gennem matematisk analyse	(÷)	÷?
4.	Automatisk kodegenerering – trinvis forfining indtil komplet...	÷	√
5.	Pris (i arbejdstid): Værktøjets stabilitet	(÷)	√?
6.	Inklusion af (model af) omkringliggende OSI-lag	÷	√
7.	Pris (i arbejdstid): Inkrementel kompilering hurtigere end fuld	√	√
8.	Grafisk simulering	√	(√)
9.	Grafisk modellering	√	÷
10.	Pris (i arbejdstid): Simulationernes hastighed	(÷)	√
11.	Pris (i arbejdstid): Kompileringshastighed v/modelindlæsning	(÷)	√

Forslag til prioriteret liste af vurderingspunkter

		CPN	ns-2
2.	Verifikation gennem simulation	√	√
3.	Verifikation gennem matematisk analyse	(÷)	÷?
4.	Automatisk kodegenerering – trinvis forfining indtil komplet...	÷	√
5.	Pris (i arbejdstid): Værktøjets stabilitet	(÷)	√?
6.	Inklusion af (model af) omkringliggende OSI-lag	÷	√
7.	Pris (i arbejdstid): Inkrementel kompilering hurtigere end fuld	√	√
8.	Grafisk simulering	√	(√)
9.	Grafisk modellering	√	÷
10.	Pris (i arbejdstid): Simulationernes hastighed	(÷)	√
11.	Pris (i arbejdstid): Kompileringshastighed v/modelindlæsning	(÷)	√
12.	Statistisk modul til behandling af måledata	÷	÷?



Forslag til prioriteret liste af vurderingspunkter

		CPN	ns-2
1.	Support: Nu, om 2 år og om 5 år	?	(√)
2.	Verifikation gennem simulation	√	√
3.	Verifikation gennem matematisk analyse	(÷)	÷?
4.	Automatisk kodegenerering – trinvis forfining indtil komplet...	÷	√
5.	Pris (i arbejdstid): Værktøjets stabilitet	(÷)	√?
6.	Inklusion af (model af) omkringliggende OSI-lag	÷	√
7.	Pris (i arbejdstid): Inkrementel kompilering hurtigere end fuld	√	√
8.	Grafisk simulering	√	(√)
9.	Grafisk modellering	√	÷
10.	Pris (i arbejdstid): Simulationernes hastighed	(÷)	√
11.	Pris (i arbejdstid): Kompileringshastighed v/modelindlæsning	(÷)	√
12.	Statistisk modul til behandling af måledata	÷	÷?
13.	Pris (for produktet selv)	√	√

Forslag til prioriteret liste af vurderingspunkter

		CPN	ns-2
1.	Support: Nu, om 2 år og om 5 år	?	(√)
2.	Verifikation gennem simulation	√	√
3.	Verifikation gennem matematisk analyse	(÷)	÷?
4.	Automatisk kodegenerering – trinvis forfining indtil komplet...	÷	√
5.	Pris (i arbejdstid): Værktøjets stabilitet	(÷)	√?
6.	Inklusion af (model af) omkringliggende OSI-lag	÷	√
7.	Pris (i arbejdstid): Inkrementel kompilering hurtigere end fuld	√	√
8.	Grafisk simulering	√	(√)
9.	Grafisk modellering	√	÷
10.	Pris (i arbejdstid): Simulationernes hastighed	(÷)	√
11.	Pris (i arbejdstid): Kompileringshastighed v/modelindlæsning	(÷)	√
12.	Statistisk modul til behandling af måledata	÷	÷?
13.	Pris (for produktet selv)	√	√

Tak

Spørgsmål?