

Tema de casa RCI

Protocolul DSR (Dynamic Routing Protocol) pentru rețele multi-hop wireless ad-hoc

Manea Constantin

An II IISC

Table of Contents

1	Introducere.....	3
2.	Descrierea protocolului DSP.....	3
2.1	Prezentarea proprietatilor importante ale protocolului.....	3
2.2	DSR Router Discovery de baza.....	5
2.3	DSR Route Maintenance de baza.....	7
2.4	Proprietati aditionale ale Route Discovery.....	7
2.4.1	Memorarea informatiilor de routare auzite.....	7
2.4.2	Raspunzand mesajelor Route Requests folosind rute memorate.....	8
2.5	Proprietăți aditionale ale Route Maintenance.....	9
2.5.1	Salvarea pachetelor.....	9
2.5.2	Scurtarea rutelor automata.....	10
2.5.3	Creșterea vitezei mesajelor Route Error.....	10
2.5.4	Memorarea informatiei negative.....	11
2.6	Rutarea multicast în DSR.....	11
3	Concluzii.....	12
	Bibliografie.....	13

1 Introducere

Protocolul DSR este un protocol simplu și eficient de rutare proiectat special pentru a fi utilizat în rețele wireless multi-hop ad-hoc cu noduri mobile. DSR permite rețelei să fie complet auto-organizată și auto-configurată, fără a fi nevoie de orice infrastructură de rețea existentă sau de administrare. Protocolul este compus din două mecanisme de descoperire de rute și de menținere a rutei, care lucrează împreună pentru a permite nodurilor să descopere și să mențină rute sursă către destinații arbitrare din rețeaua ad-hoc. Utilizarea rutelor sursă permite rutării pachetelor să fie lipsită de bucle, evită necesitatea informațiilor de rutare actualizate în nodurile intermediare prin care pachetele sunt transmise și permite nodurilor expedierea sau stocarea în cache a informațiilor rutelor din pachete pentru uz propriu viitor. Toate aspectele legate de acest protocol operează în întregime la cerere, ceea ce permite DSR să scaleze automat (necesitând puțină procesare).

Protocolul DSR permite nodurilor să descopere în mod dinamic o rută sursă peste multe hopuri către orice destinație din rețeaua ad-hoc. Fiecare pachet de date trimis conține în header lista de noduri completă și ordonată prin care pachetul trebuie să treacă, permițând o rutare fără bucle și evitând nevoia de informații de rutare actualizate în nodurile intermediare prin care pachetul este transmis. Prin includerea acestui traseu sursă în headerul fiecărui pachet de date, alte noduri intermediare pot, de asemenea, reține cu ușurință aceste informații de rutare pentru o utilizare viitoare.

În proiectarea DSR-ului, s-a căutat să se creeze un protocol de rutare cu un cost foarte scăzut dar capabil de a reacționa rapid la schimbările din rețea, fiind capabil să furnizeze servicii extrem de reactiv pentru a asigura livrarea cu succes a pachetelor de date, în ciuda mișcării nodurilor sau altor modificări ale rețelei.

2. Descrierea protocolului DSP

2.1 Prezentarea proprietatilor importante ale protocolului

Protocolul DSR este compus din două mecanisme care lucrează împreună pentru a permite descoperirea și întreținerea rutelor sursă din rețeaua ad-hoc:

- Route Discovery este mecanismul prin care un nod S care dorește să trimită un pachet la un nod destinație D obține o rută sursă D. Route Discovery este folosit numai atunci când S încearcă să trimită un pachet către D și nu știe deja un traseu către D.
- Route Maintenance este mecanismul prin care nodul S este capabil să detecteze, în timp ce

utilizeaza un traseu sursă catre D, daca topologia rețelei s-a schimbat astfel încât să nu mai poata folosi ruta la D, deoarece un link de-a lungul traseului nu mai funcționează. Când Route Maintenance indică o rută sursă defecta, S poate încerca să folosească orice altă ruta cunoscută catre D, sau poate invoca Route Discovery din nou pentru a găsi un nou traseu. Route Maintenance este folosit doar atunci când S trimite pachete la D.

Route Discovery si Route Maintenance opereaza în întregime la cerere. În special, spre deosebire de alte protocoale, DSR nu necesita pachete periodice de orice fel, la orice nivel în cadrul rețelei. De exemplu, DSR nu folosește nici un anunț periodic de rutare sau pachete de detectare a vecinilor și nu se bazează pe aceste funcții din alte protocoale de rețea. Acest comportament în întregime la cerere si lipsit de activitate periodica permite numărului de pachete DSR sa scada până la zero, atunci când toate nodurile sunt aproximativ staționare în raport cu celălalt și toate traseele necesare de comunicare au fost deja descoperite. Cum nodurile încep să se miște mai mult sau modelele de comunicare se schimba, costul pachetelor de rutare DSR crește doar la nevoia pentru a urmări rutele utilizate în prezent.

Ca răspuns la un singur Route Discovery (precum și prin informații de rutare din alte pachete auzite), un nod poate învăța și retine mai multe rute către orice destinație. Acest lucru permite ca reacția la schimbările de rutare să fie mult mai rapide, deoarece un nod cu multiple rute spre o destinație se poate folosi o altă rută din memorie dacă cea folosita a picat. Acesta memorare de multiple rute evită, de asemenea, costul suplimentar pentru a efectua un nou Route Discovery de fiecare dată cand un traseu folosit cade.

Funcționarea Route Discovery si Route Maintenance în DSR sunt concepute pentru a permite existența legăturilor unidirecționale și rutelor asimetrice. În special în rețele fără fir, este posibil ca o ruta între două noduri să nu funcționeze la fel de bine în ambele direcții din cauza antenelor diferite sau a modelelor de propagare sau a surselor de interferență. DSR permite utilizarea de astfel de link-uri uni-direcționale atunci când este necesar, îmbunătățind performanța globala și conectivitate rețelei.

DSR sprijină, de asemenea, interconectarea a diferite tipuri de rețele fără fir, permițând ca o cale sursă să fie compusa din hopuri de orice tipuri de rețele disponibile. De exemplu, unele noduri din rețeaua ad-hoc pot avea doar radio de raze scurte, în timp ce alte noduri pot avea atât cu raze scurte și raze lungi; combinarea acestor noduri poate fi considerata de către DSR ca fiind o singură rețea ad-hoc. În plus, rutarea DSR a fost integrata în rutarea standard din Internet în cazul în care un nod gateway conectat la Internet participa de asemenea la protocoalele de rutare din rețeaua ad-hoc; și a fost integrată în rutarea IP Mobil, atunci când un astfel de nod servește, de asemenea, rolul unui agent străin.

2.2 DSR Router Discovery de baza

Când un nod S creează un nou pachet destinat unui alt nod D, se plasează în antetul pachetului o ruta sursă ce conține secvența de hopuri pe care pachetul ar trebui să o urmeze în drumul său spre D. În mod normal, S va obține o ruta sursă adecvată căutând prin rutele memorate anterior, dar dacă un traseu nu este găsit în memorie se va iniția protocolul Route Discovery pentru a găsi o nouă rută către D. În acest caz, noi numim S inițiator și D țintă din Route Discovery.

De exemplu, Figura 1 ilustrează un exemplu de Route Discovery în care un nod A încearcă să descopere o cale către nodul E. Pentru a iniția Route Discovery, A transmite un mesaj ROUTE REQUEST ca un singur pachet de broadcast local, care este primit de (aproximativ) toate nodurile aflate în prezent în raza de transmisie a lui A. Fiecare mesaj ROUTE REQUEST identifică inițiatorul și ținta Route Discovery, și conține, de asemenea, o cerere id unic determinată de inițiatorul REQUEST. Fiecare ROUTE REQUEST conține, de asemenea, o înregistrare cu adresele fiecărui nod intermediar prin care acest mesaj ROUTE REQUEST a fost transmis. Această înregistrare este inițializată ca o listă goală de către inițiatorul Route Discovery.

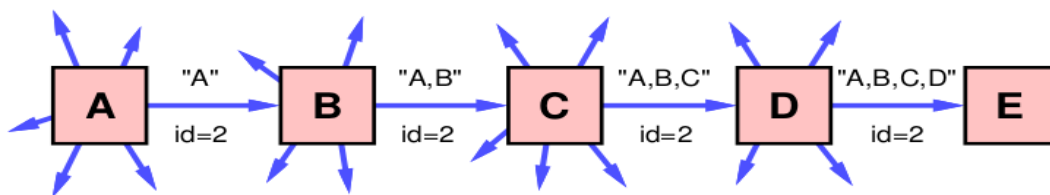


Figura 1. Exemplu de Route Discovery. Nodul A este inițiatorul și nodul E ținta

Când un alt nod primește un pachet Route Request, în cazul în care este ținta unui Route Discovery, returnează un mesaj Route Reply către inițiatorul Route Discovery, oferind o copie a înregistrărilor acumulate în solicitarea Route Request; când inițiatorul primește acest Route Reply, acesta memorează acest traseu în Route Cache pentru utilizarea ei în trimiterea de pachete ulterioare către această destinație. În caz contrar, dacă acest nod la primirea Route Request a văzut recent un alt mesaj Route Request de la același inițiator având același id de cerere, sau în cazul în care constată că propria sa adresă este deja înregistrată în traseul mesajului Route Request, elimină cererea. În caz contrar, acest nod atașează propria sa adresă la înregistrarea traseului în mesajul Route Request și îl propagă ca un pachet de broadcast local (cu același id de cerere).

În returnarea Route Reply inițiatorului Route Discovery, cum ar fi nodul E răspunzând înapoi nodului A în figura 1, nodul E va examina în mod obișnuit propriul traseu din memorie pentru o rută înapoi către A, și dacă este găsit, acesta va folosi pentru traseul sursă de livrare a pachetului Route Reply. În caz contrar E poate efectua propriul Route Discovery pentru nodul A, dar pentru a evita posibile descoperiri recursive infinite, trebuie să atașeze acest Route Reply propriului Route Request pentru A. De asemenea, este posibil să atașeze alte mici porțiuni de pachet, cum ar fi un TCP SYN, la un Route Request, folosind aceeași mecanism. Nodul E ar putea pur și simplu să inverseze ordinea hopurilor din înregistrarea traseului pe care încearcă să îl trimită în Route Reply, și să folosească acest lucru ca traseu sursă în pachetul care transportă Route Reply-urile.

Pentru protocoale MAC cum ar fi IEEE 802.11 care necesită un schimb de cadre bi-direcțional ca parte a protocolului MAC, această inversare de rută este preferată deoarece evită costul suplimentar al unui eventual al doilea Route Discovery, iar acesta testează traseul descoperit asigurându-se că este bi-direcțional înainte ca inițiatorul pachetului Route Discovery să înceapă să folosească ruta. Totuși, această tehnică va împiedica descoperirea rutelor cu ajutorul legăturilor uni-direcționale. În medii wireless în care este permisă utilizarea de legături unidirecționale, rutele pot în unele cazuri să fie mai eficiente decât cele doar cu link-uri bi-direcționale, sau pot fi singura modalitate de a atinge conectivitate la nodul țintă.

La inițierea unui Route Discovery, nodul expeditor salvează o copie a pachetului inițial într-un tampon local numit tampon de expediere. Tamponul de expediere conține o copie a fiecărui pachet care nu poate fi transmis prin acest nod, deoarece nu are încă o cale sursă către destinația pachetului. Fiecare pachet în tamponul de expediere este ștampat cu momentul în care a fost plasat în tampon și este eliminat dacă stă în tamponul de expediere pentru o anumită perioadă de timp; dacă este necesar pentru a preveni umplerea tamponului de expediere, o strategie FIFO sau o alta de înlocuire poate fi de asemenea folosită pentru a evacua pachetele înainte de expirare.

În timp ce un pachet rămâne în tamponul de expediere nodul ar trebui să inițieze un nou pachet Route Discovery pentru adresa de destinație a pachetului. Cu toate acestea, nodul trebuie să limiteze rata la care sunt inițiate astfel de noi descoperiri de traseu pentru aceeași adresă, deoarece este posibil ca nodul destinație să nu fie momentan accesibil. În special, ca urmare a razei limitate de transmisie wireless și mișcarea nodurilor din rețea, rețeaua poate deveni segmentată uneori, ceea ce înseamnă că nu există nicio secvență de noduri prin care un pachet ar putea fi transmis pentru a ajunge la destinație. În funcție de modelul de mișcare și densitatea nodurilor în rețea asemenea segmentare a rețelei poate fi rară sau poate fi comună.

Dacă un nou Route Discovery a fost inițiat pentru fiecare pachet trimis de un nod într-o astfel de situație, un număr mare de pachete Route Request neproductive ar fi propagate în întreg subgrupul de rețea ad-hoc accesibil de la acest nod.

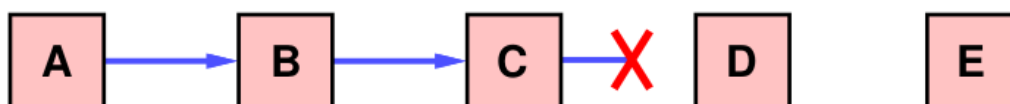


Figura 2. Exemplu de Route Maintenance: Nodul C nu poate trimite un pachet de la A la E prin link-ul său către D

Pentru a reduce costul suplimentar al pachetelor Route Discovery, vom folosi back-off exponențial pentru a limita rata la care noi descoperiri de rute pot fi inițiate de orice nod pentru aceeași țintă. Dacă nodul încearcă să trimită pachete de date suplimentare pentru același nod mai frecvent decât această limită, pachetele ulterioare trebuie să fie introduse în tamponul de expediere până când un Route Reply este primit, dar nodul nu trebuie să inițieze un nou Route Discovery până când intervalul minim admisibil între noi Route Discovery pentru acest obiectiv a fost atins. Această limitare a ratei maxime de pachete Route Discovery pentru aceeași țintă este similară cu mecanismul impus de nodurile de Internet pentru a limita rata la care cererile ARP sunt trimise pentru orice adresă IP țintă unică.

2.3 DSR Route Maintenance de baza

Când originea sau transmite un pachet folosind o ruta sursă, fiecare nod care transmite pachetul este responsabil pentru confirmarea că pachetul a fost primit de către următorul hop lungul rutei sursă; pachetul este retransmis (până la un număr maxim de reîncercări) până când confirmarea de primire este primită. De exemplu, în situația ilustrată în figura 2, nodul A a expediat un pachet către E, utilizând o ruta sursă prin nodurile intermediare B, C și D. În acest caz, nodul A este responsabil pentru primirea pachetului de către B, nodul B este responsabil pentru primirea de către nodul C, nodul C este responsabil pentru primirea de către nodul D, iar nodul D este responsabil pentru primirea, în cele din urmă, la destinația E. Aceasta confirmare de primire, în multe cazuri, poate fi furnizată fără nici un cost pentru DSR, fie ca parte existența în standardul protocolului MAC în uz, sau de o confirmare pasivă. Dacă niciuna dintre aceste mecanisme de confirmare nu sunt disponibile, nodul emitent poate seta un bit în antetul pachetului pentru a solicita o confirmare software specifică DSR-ului care returnată de hop-ul următor; această recunoaștere software va fi în mod normal transmisă direct la nodul originar, dar dacă legătura dintre aceste două noduri este unidirecțională, această confirmare software poate călători pe o altă cale multi-hop. Dacă pachetul este retransmis de către un hop de numărul maxim de ori și nici o confirmare de primire nu este primită, acest nod returnează un mesaj de Route Error expeditorului original al pachetului, identificând legătura pe care pachetul nu a putut fi transmis. De exemplu, în figura 2, în cazul în care C este în imposibilitatea de a livra pachetul la următorul hop D, apoi C returnează Route Error la A, afirmând că legătura de la C la D este în prezent "ruptă". Atunci nodul A elimină această legătură ruptă din memorie; orice retransmisie a pachetului original este o funcție de protocoale de nivel superior cum ar fi TCP. Pentru a trimite o astfel de retransmisie sau alte pachete pentru aceeași destinație E, în cazul în care A are în Route Cache altă rută pentru E se poate trimite pachetul imediat utilizând noul traseu. În caz contrar, se poate efectua o nouă Route Discovery pentru acest obiectiv.

2.4 Proprietati aditionale ale Route Discovery

2.4.1 Memorarea informatiilor de routare auzite

Un nod care trimite mai departe sau aude un pachet poate adăuga informațiile de rutare de la pachet în propria Route Cache. În particular, ruta sursă utilizată într-un pachet de date, înregistrarea rutei acumulată într-o Route Request sau ruta fiind returnată într-o Route Response pot fi memorate în orice nod. Informațiile de rutare de la oricare dintre aceste pachete recepționate pot fi memorate, fie ca pachetul a fost adresat acestui nod, trimis la o adresă MAC broadcast (sau multicast) sau a primit în timp ce interfața de rețea a nodului este în modul promiscuu.

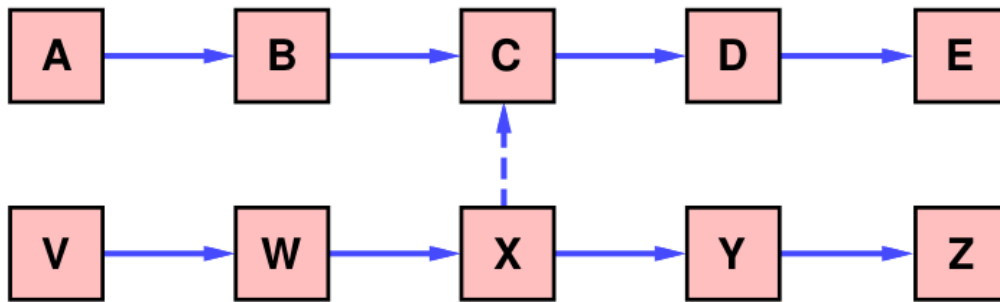


Figura 3. Limitele memorării informației de rutare auzită. Nodul C trimite pachete către E și aude pachete de la X

O limitare, însă, a memorării de astfel de informații de rutare auzite constă în posibila prezența a legăturilor unidirecționale din rețeaua ad-hoc. De exemplu, Figura 3 ilustrează o situație în care nodul A utilizează o rută sursă și comunică cu nodul E. În timp ce nodul C transmite un pachet de date de-a lungul traseului de la A la E, se poate adăuga întotdeauna la memoria cache prezența legăturii învățată din antetele acestor pachete, de la el la D și de la D la E. Cu toate acestea, direcția legăturilor inverse identificate în header-ul pachetelor, de la el înapoi la B și de la B la A, nu pot funcționa deoarece aceste link-uri ar putea fi uni-direcționale. Dacă nodul C știe că legăturile sunt de fapt bi-direcționale, de exemplu ca urmare a protocolului MAC utilizat, ar putea să le memoreze, dar altfel nu ar trebui.

De asemenea, nodul V din figura 3 folosește o rută sursă diferită pentru a comunica cu Z. Dacă nodul C aude nodul X transmitând un pachet de date pentru a transmite către Y (de la V), nodul C ar trebui să ia în considerare dacă link-urile implicate pot fi cunoscute ca fiind bi-direcționale sau nu înainte de a le memora. În cazul în care link-ul de la X la C (peste care acest pachet de date a fost primit) este cunoscut ca fiind bi-direcțional, atunci C ar putea memora legătura de la el la X, link-ul de la X la Y, și legătura de la Y la Z. Dacă toate link-uri pot fi presupuse a fiind bi-direcționale C ar putea, de asemenea, memora legăturile de la X la W și de la W la V. Considerații similare se aplică la informațiile de rutare, care ar putea fi învățate de la pachete Route Request sau Route Reply înaintate sau auzit.

2.4.2 Raspunzand mesajelor Route Requests folosind rute memorate

Un nod care primește un Route Request pentru care nu este ținta, caută traseul în memorie pentru ruta ținta a cererii. Dacă este găsită, nodul returnează un Route Response inițiatorului, mai degrabă decât transmiterea unui Route Request. În Route Response se setează înregistrarea rutei pentru a defini secvența de hopuri pe care această Route Request a făcut-o înainte, concatenată cu propria idee a traseului de la el la ținta din Route Cache-ul său.

Cu toate acestea, înainte de a transmite un pachet Route Response care a fost generat folosind informații din Route Cache în acest fel, un nod trebuie să verifice dacă traseul rezultat și returnat în Route Response, după această concatenare, nu conține noduri duplicat enumerate în

înregistrarea traseului. De exemplu, Figura 4 ilustrează un caz în care un Route Request pentru nodul E a primit de nodul F și nodul F are deja în Route Cache un traseu de la sine pentru nodul E. Concatenarea traseului acumulat din Route Request și Route Cache-ul lui F ar include un nod duplicat și anume trecerea de la C la F și înapoi la C.

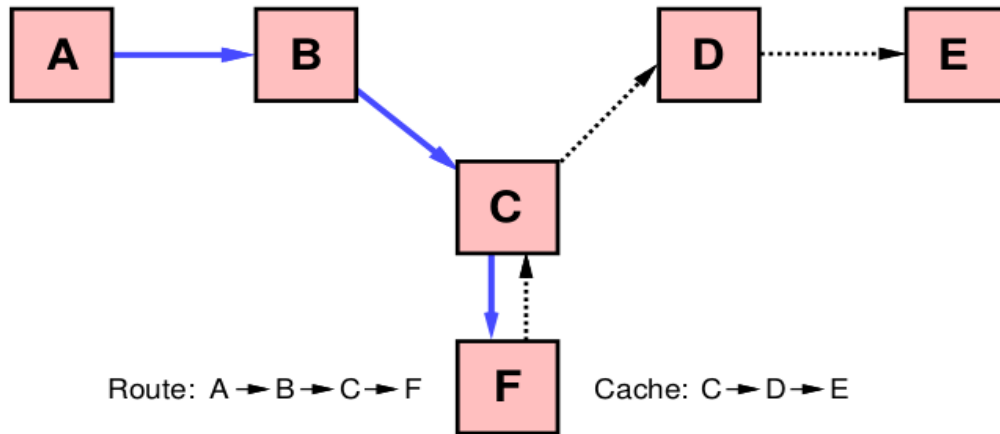


Figura 4. O posibilă dublare a hopurilor rutei evitată prin limitarea Route Discovery

Nodul F în acest caz, ar putea încerca să editeze traseul pentru a elimina dublarea, rezultând într-o cale de la A la B la C la D și la E, dar în acest caz, nodul F nu ar fi pe traseul care a fost returnat în propriul Route Response. DSR Route Discovery îi interzice nodului F să returneze un astfel de Route Response din două motive. În primul rând, această limitare crește probabilitatea ca traseul rezultat este valabil, deoarece F în acest caz ar fi primit un Route Error dacă traseul s-ar fi defectat. În al doilea rând, această limitare înseamnă că un Route Error care traversează ruta este foarte probabil să treacă prin orice nod care a trimis Route Response pentru ruta (inclusiv F), care ajută la asigurarea că datele vechi sunt eliminate din cache (cum ar fi la F) în timp util. În caz contrar, urmatorul Route Discovery inițiat de o ar putea fi, de asemenea, contaminat cu un Route Response din F ce conține același traseu vechi. Dacă Route Request nu îndeplinește aceste restricții, (nodul F în acest exemplu) nodul elimină Route Requestul.

2.5 Proprietăți aditionale ale Route Maintenance

2.5.1 Salvarea pachetelor

După trimiterea unui mesaj Route Error ca parte a Route Maintenance așa cum este descris în secțiunea 3.3, un nod poate încerca să salveze pachetul de date care a cauzat mesajul Route Error în loc să îl arunce. Pentru a încerca să salveze un pachet, nodul care trimite Route Error caută propriul Route Cache pentru o rută de sine la destinația pachetului care a provocat eroarea. Dacă

este găsit un astfel de traseu, nodul poate salva pachetul după returnarea Route Error prin înlocuirea traseului sursă original în pachet cu traseul din Route Cache-ul său. Nodul transmite apoi pachetul la următorul nod indicat pe această rută sursă. De exemplu, în figura 2, în cazul în care nodul C are o altă rută cache către nodul E, acesta poate salva pachetul aplicând acest traseu pachetului în loc să îl arunce.

Când salvează un pachet în acest fel, pachetul este, de asemenea, marcat ca fiind recuperat pentru a preveni ca un singur pachet să fie salvat de mai multe ori. În caz contrar, ar putea fi posibil ca pachetul să introducă o buclă de rutare, diferite noduri salvând în mod repetat pachetul și înlocuind traseul sursă al pachetului cu trasee de la ele. Un mecanism alternativ de colectare ar consta doar în înlocuirea sufixului neutilizat de la ruta inițială cu noul traseu de la acest nod din Route Cache, formând un traseu nou al cărui prefix este traseul original și al cărui sufix este traseul din cache. În acest caz, regulile obișnuite de evitare a nodurilor duplicate să fie listate într-un traseu sursă sunt suficiente pentru a evita buclele de rutare.

2.5.2 Scurtarea rutelor automata

Trasee sursă în utilizare poate fi scurtate automat dacă una sau mai multe hopuri intermediare de pe traseu nu mai sunt necesare. Acest mecanism de scurtare automată a rute de utilizare este oarecum similar cu utilizarea confirmărilor pasive. În special în cazul în care un nod este capabil să auză un pachet când transportă un traseu sursă, atunci acest nod examinează partea neutilizată din acest traseu sursă. Dacă acest nod nu este următorul hop destinatarii pachetului, dar este numit în porțiunea neutilizată ulterioară a traseului sursă a pachetului, atunci se poate deduce că nodurile intermediare înainte de el în calea sursă nu mai sunt necesare în traseu. De exemplu, Figura 6 ilustrează un exemplu în care nodul C a auzit un pachet de date fiind transmis de la A la B, pentru ca mai târziu să ajungă la C; săgeata îndreptată spre un nod în traseul sursă din fiecare pachet indică destinatarul următor al pachetului.

În acest caz, acest nod (nodul C) returnează un mesaj Route Reply expeditorului original al pachetului (nodul A). Route Reply oferă traseul mai scurt concatenând porțiunea de traseu sursă originală până la nodul care a transmis pachetul auzit, plus sufixul traseului sursă original începând cu nodul care a trimis Route Response. În acest exemplu, traseul returnat în Route Response trimis de la C la A da noua rută ca secvența de hopuri la A la C la D.

2.5.3 Creșterea vitezei mesajelor Route Error

Când un nod sursă primește o eroare în calea pentru un pachet de date care provine în acest nod sursă propagă acest Route Error vecinilor săi prin atașându-l pe următorul traseu cerere. În acest fel, informații vechi din cache-ul nodurilor în jurul acestui nod sursă nu vor genera Route Reply care conțin aceeași legătură invalidă pentru care acest nod sursă a primit Route Error. De exemplu, în situația prezentată în figura 2, nodul A învață din mesajul de eroare traseul din C, care leagă C la D. Se elimină astfel această legătură de la propriul traseu din Cache și inițiază o

nouă Route Discovery. Pe pachetul Route Error inițierea acestui Route Discovery, nodul A atașează o copie a acestui mesaj de Route Error, asigurându-se că mesajul de Route Error răspandeste bine la alte noduri, precum și garantarea faptului că orice Route Response pe care le primește (inclusiv cele de la alt nod de traseu cache), în răspuns la această Route Request nu conține un traseu care presupune existența acestei legături rupte.

Am considerat, de asemenea, o îmbunătățire a Route Maintenance, în care un nod, cum ar fi A în figura 4, care primește un Route Error va transmite eroarea de-a lungul același traseu sursă care a dus la eroare. Acest lucru va garanta aproape că Route Error ajunge la nodul care a generat Route Response conține link-ul spart, ceea ce va împiedica acel nod să contamineze un viitor Route Discovery cu același link-ul spart.

2.5.4 Memorarea informației negative

În unele cazuri, DSR ar putea beneficia de noduri cu caching de informații "negative", în Route Cache-urile lor. De exemplu, în figura 2, în cazul în care nodul A cachează faptul că legătura de la C la D este în prezent ruptă (mai degrabă decât dacă pur și simplu ar scoate acest hop din Route Cache-ul acestuia), se poate garanta că Route Response primește ca răspuns la noua Route Discovery, vor fi acceptate care utilizează acest link rupt. O perioadă de expirare scurtă trebuie să fie plasată pe aceste informații cache negative, deoarece în timp ce aceasta ruta este în Route Cache, un altfel de nod va refuza să permită acest link în cache, chiar dacă această legătură începe să lucreze din nou.

Un alt caz în care informațiile negative din cache într-un nod de Route Cache ar putea fi utile și în cazul în care un link furnizează servicii foarte variabile, uneori funcționează corect, dar de multe ori nu. Această situație poate apărea, de exemplu, în cazul în care legătura este aproape la limita razei de transmisie fără fir și există surse importante de interferență în apropierea nodului de primire pe acest link. În acest caz, informațiile negative din cache pe care această legătură este ruptă, un nod poate evita adăugarea acest link problematic înapoi la Route Cache în scurte perioade în care acesta este în lucru.

2.6 Rutarea multicast în DSR

DSR nu suportă în prezent cu adevărat rutare multicast, dar are suport pentru o aproximare a acestuia, care este suficientă în multe contexte de rețea. Printr-o extindere a mecanismului Route Discovery, DSR sprijină inundarea controlată a unui pachet de date pentru toate nodurile din rețeaua ad-hoc, care sunt într-un număr specificat de hopuri de origine; aceste noduri pot aplica apoi filtrarea adresei destinație (de exemplu, în software), pentru a limita pachetul la acele noduri subscribe la adresa de destinație multicast indicată pachetului. Deși acest mecanism nu are suport

pentru trunchierea arborelui de difuzare pentru a conserva resursele de rețea, acesta poate fi folosit pentru a distribui informații pentru toate nodurile din rețeaua ad-hoc abonat la adresa multicast destinație. Acest mecanism poate fi de asemenea util pentru a trimite pachete, nivelul de aplicare la toate nodurile într-o serie limitată în jurul expeditorului. Pentru a utiliza această formă de multicasting, atunci când o aplicație pe un nod DSR trimite un pachet la o adresă destinație multicast, DSR atașează datelor din pachetul interior o cerere de traseu orientată la adresa multicast. Route Request normală de propagare va duce la acest pachet fiind distribuit în mod eficient la toate nodurile din rețea în numărul de hopuri specificat (TTL) de origine. După transmiterea pachetului așa cum este definit de Route Discovery, fiecare nod ce primește apoi analizează individual adresa de destinație a pachetului și elimină pachetul dacă este destinat unei adrese multicast pentru care acest nod nu este abonat.

3 Concluzii

Protocolul Dynamic Source Routing (DSR) oferă performanțe excelente pentru rutare multi-hop în rețele wireless ad-hoc. Într-o adevărată rețea ad-hoc DSR are costuri suplimentare de rutare foarte scăzute și este capabil să ofere în mod corect aproape toate pachete de date, chiar și cu mișcări continue și rapide ale nodurilor din rețea.

Un motiv cheie pentru această performanță bună este faptul că DSR funcționează în întregime la cerere, fără nici o activitate periodică de orice fel la orice nivel în cadrul rețelei. De exemplu, DSR nu folosește nici un anunț periodic de rutare, de detectare de status a legăturii, sau pachete de detectare de vecini și nu se bazează pe aceste funcții din orice alte protocoale care stau în rețea. Acest comportament în întregime la cerere și lipsa de activitate periodică permite costului de rutare cauzat de DSR la scară până la zero, atunci când toate nodurile sunt aproximativ staționare în raport cu celelalte și toate traseele necesare în comunicarea curentă au fost deja descoperite. Cum nodurile încep să se miște mai mult sau modelele de comunicare schimbă, costul pachetelor de rutare ale DSR crește automat numai cu cât este necesar pentru a urmări căile utilizate în prezent.

Bibliografie

<http://www.cs.jhu.edu/~dholmer/600.647/papers/dsr.pdf> - de unde am luat

http://www.researchgate.net/profile/Dr_Rakesh_Jha/publication/260984717_Analysis_of_AODV_and_DSR_Routing_Protocols_for_a_Static_Ad_Hoc_Network_Using_Qualnet_Network_Simulator/links/00b7d532dd3e382b35000000.pdf

http://en.wikipedia.org/wiki/List_of_ad_hoc_routing_protocols

<http://tools.ietf.org/html/rfc4728>

http://en.wikipedia.org/wiki/Dynamic_Source_Routing