

Рутирачки алгоритми за услуги во MANET во реално време

Борис Михајлов, Димитар Саздовски, М-р Митко Богданоски, Доц. Д-р Сашо Гелев

Европски Универзитет – Скопје, Р. Македонија

mihajlov.boris@live.eurm.edu.mk, dimitarsazdovski@hotmail.com,

{mitko.bogdanoski, saso.gelev@eurm.edu.mk}

Апстракт – Мобилните Ad Hoc Мрежи (MANET's) се безжични мрежи со повеќе прескокнувања, каде што сите јазли учествуваат во одржувањето на мрежната конекција. Повеќе од јасно е дека во мобилните средини, каде што јазлите континуирано се движат со големи брзини, обично е тешко да се одржуваат и обноват рутирачките патеки.

Кога големината на мрежата е голема и јазлите се мобилни, фреквенцијата на линкот е неуспешна и потрошувачката на енергија на јазлите е поголема. Неуспехот на линкот се идентификува само за време на преносот на пакетите. За да се надмине овој проблем, се користат алгоритми за отстранување на заостанатите патеки после одреден период на пауза на патеката. Целта на овој труд е да се проучат и анализираат протоколите за рутирање во MANET, со цел подобрување на испораката на пакети и намалување на контролните пораки.

Ќе бидат анализирани два типа на рутирачки протоколи, и тоа: Протоколи Управувани со Табели и Протоколи на Барање.

Клучни Зборови - Мобилни Ad Hoc Мрежи (MANET's), рутирачки протоколи, Протоколи управувани со табели, Протоколи на барање.

1. ВОВЕД

Безжичните мрежи претставуваат нова технологија која им овозможува на корисниците електронски пристап до информациите и услугите, без оглед на нивната географска положба.

Безжичните мрежи може да се класифицираат во два типа, и тоа:

1. Инфраструктурни мрежи и
2. Безинфраструктурни (Ad Hoc) мрежи.

За разлика од инфраструктурните базирани мрежи, во Ad Hoc мрежите сите јазли се мобилни и можат да бидат конектирани динамично на произволен начин. Рутирачките протоколи може да бидат поделени на рутирачки протоколи базирани на:

- Ажуриран механизам на рутирачка информација,
- Употреба на привремената рутирачка информација,
- Топологиската организација,
- Разновидни класификации врз основа на користење на одредени ресурси.

Ќе ги разгледаме рутирачките протоколи базирани на ажуриран механизам на рутирачка информација, и тоа: проактивни (Протоколи управувани со табели), реактивни (Протоколи на барање) и хибридни протоколи.

Проактивните рутирачки протоколи се однесуваат на периодично собирање и размена на тополошка информација.

Сите вакви протоколи во суштина секогаш обезбедуваат ажурирана рутирачка информација, која може да биде искористена за доставување на пакети со податоци преку мрежата до нивната дестинација. Периодичните рутирачки надградби бараат значаен придонес од мрежата за одржување на состојбата во зависност од специфичниот протокол и степенот на мобилност. Најдобри примери за рутирачки протоколи управувани со табели се Destination Sequenced Distance Vector (DSDV), Wireless Routing Protocol (WRP), Cluster-Head Gateway Switch Routing (CGSR), Optimized Link State Routing (OLSR), Fisheye State Routing (FSR), Hot Standby Router (HSR), Global State Routing (GSR).

Реактивните рутирачки протоколи се воведени за да се спречи периодична размена на рутирачка информација која може да ги потроши достапните мрежни ресурси. Идејата е да се бараат патеки низ мрежата само ако пакетите со податоци треба да бидат пренесени. Првиот протокол од оваа класа кој бил создаден е Dynamic Source Routing (DSR) протоколот.

Други моментални приоди се Ad Hoc on Demand Distance Vector (AODV), кој е најпознатиот рутирачки протокол во Ad Hoc мрежната заедница и Dynamic MANET on-Demand Routing Protocol (DYMO), Associativity-Based Routing (ABR), Simple Spectral Access (SSA), Flow-Oriented Routing Protocol (FORP), Policy Based Routing Protocol (PLBR).

Хибридните рутирачки протоколи се обидуваат да ги искористат специфичните предности на проактивните рутирачки протоколи, односно брзата испорака на пакети затоа што рутирачка информација е секогаш достапна, т.е., се овозможува намалување на трошоците во рамките на мрежата.

Типичен пример е Zone Routing Protocol (ZRP). Други примери се: Core-Extraction Distributed Ad Hoc Routing Protocol (CEDAR), Zone -Based Hierarchical Link State Protocol (ZHLS).

2. ПРОТОКОЛИ УПРАВУВАНИ СО ТАБЕЛИ

Протоколите управувани со табели постојано ги ажурираат своите патеки, т.е. секој јазол одржува една или повеќе табели со рутирачка информација

во мрежата, се со цел да обезбедат ажуриран изглед на мрежата. Овие рутирачки протоколи се разликуваат по начинот на кој топологијата ја праќа информацијата низ мрежата, по бројот на потребни рутирачки табели, по начинот на кој се ажурираат рутирачките податоци. Овие протоколи не се погодни за поголеми мрежи, затоа што е потребно да се ажурират записите за секој јазол во рутирачката табела.

2.1. Destination sequenced Distance vector routing protocol (DSDV)

Целта на ова рутирање е да се одредат “добрите” патеки (секвенци од рутери) низ мрежата, од почетната точка до крајната точка.

The Destination-Sequenced Distance-Vector (DSDV) Routing Алгоритмот е базиран на идејата на класичниот рутирачки алгоритам на Bellman-Ford, со одредени подобрувања. Овој протокол додава нов атрибут, секвенционален број, во секоја рутирачка табела за секој јазол.

Овој протокол е прилагоден за користење при размена на податоци за време на менувањето и за произволните патеки на интерконекција кои не можат да бидат блиску до секоја базна станица.

Секоја мобилна станица поддржува рутирачка табела која ги содржи сите списоци со достапни дестинации, бројот на скокови што ги достигнуваат дестинацијата и секвенционален број додаден од страна на крајниот јазол.

Секвенционалниот број се користи за да се направи разлика помеѓу старите и новите правци и на тој начин да се избегне создавање на циклуси.

По добивањето на рутирачката информација, приемниот јазол ја зголемува метриката и ја пренесува информацијата преку емитување.

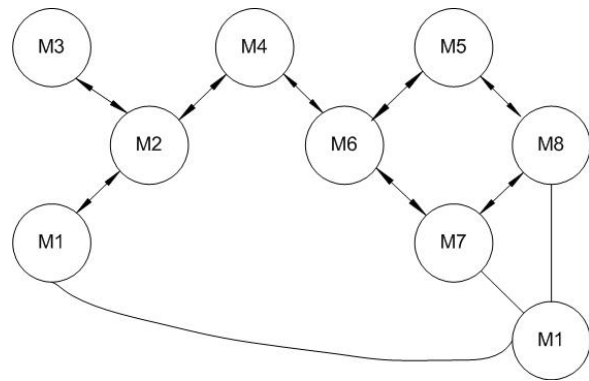
Зголемувањето на метриката се прави пред преносот, бидејќи влезните пакети ќе мора да направат уште еден скок за да достигнат до својата дестинација. Времето помеѓу емитувањето на рутирачката информација за пакетите е другиот важен фактор кој не треба да биде занемарен. Мобилниот хост е причина за прекратени врски, бидејќи се движи од едно до друго место, во рамките на мрежата. Кога патеката е прекината во мрежата, тогаш веднаш на таа метрика и се доделува бесконечна метрика со утврдување дека таму не постои скок и секвенционалниот број е ажуриран. Секвенционалните броеви од мобилните хостови се дефинираат да бидат парни броеви, а секвенционалните броеви што се генерираат за да покажат бесконечност се непарните броеви.

Ажурирањата на рутирачката табела може да бидат пратени на два начина: како "full dump" или “делумна надградба”.

"Full dump" надградбата ја испраќа целата рутирачка табела на соседите и може да опфати повеќе пакети, додека пак во “делумната надградба” се праќаат само оние записи од рутирачката табела кои имаат метричка промена

од последната надградба и мора да се вклопат во пакет. Кога сите пакети со информации се примаат од друг јазол, се споредува секвенционалниот број со сите други секвенционални броеви што се на располагање за тој запис. Додека информацијата за јазлите се ажурира, метриката се зголемува за 1, а секвенционалниот број се зголемува за 2. Решението за овој проблем е да се одложи претставувањето на овие патеки се додека не се појави некоја подобра метрика. Ако две патеки имаат ист секвенционален број, тогаш се користи патеката со најдобра метрика (најкратката патека).

Пример за DSDV работење



Слика 1. Движење на мобилните хостови во Adhoc мрежата

Како што може да се види од Слика 1, мобилната мрежа содржи 8 хоста.

Ќе ги погледнеме промените на МН4 рутирачката табела во однос на движењата на МН1. Првично, сите мобилни јазли ги претставуваат своите рутирачки информации на сите јазли во мрежата и оттука рутирачката табела (Табела 1) на МН4 првично би изгледала вака:

Табела 1. Рутирачка табела на мобилен хост МН4

| Dest. | Next Hop | Metric | Sequence number | Install | StableData |
|-------|----------|--------|-----------------|-------------|------------|
| MН1 | MН2 | 2 | S406 MН1 | T001 MН4 | Ptr1 MН1 |
| MН2 | MН2 | 1 | S128 MН2 | T001 MН4 | Ptr1 MН2 |
| MН3 | MН2 | 2 | S564 MН3 | T001 MН4 | Ptr1 MН3 |
| MН4 | MН4 | 0 | S710 MН4 | T001 MН4 | Ptr1 MН4 |
| MН5 | MН6 | 2 | S392 MН5 | T002 MН4 | Ptr1 MН5 |
| MН6 | MН6 | 1 | S076 MН6 | T001 MН4 | Ptr1 MН6 |
| MН7 | MН6 | 2 | S128 MН7 | T002 MН4 | Ptr1 MН7 |
| MН8 | MН6 | 3 | S050 MН8 | T002 MН4 | Ptr1 MН8 |

Кога хостот МН1 ја менува својата локација поблиску до МН7 и МН8, како што е прикажано на Слика 1, тогаш врската помеѓу МН2 и МН1 ќе биде прекината како резултат на бесконечната метрика на МН2 за МН1 и секвенционалниот број ќе биде променет во непарен број во рутирачката табела за хост МН2.

Сега рутирачката табела би изгледала вака:

Табела 2. Рутирачка табела после движењето на МН1

| Dest. | Next Hop | Metric | Sequence number | Install | StableData |
|-------|----------|--------|-----------------|-------------|------------|
| MН1 | MН6 | 3 | S516MН1 | T001 MН4 | Ptr1 MН1 |
| MН2 | MН2 | 1 | S238MН2 | T001 MН4 | Ptr1 MН2 |
| MН3 | MН2 | 2 | S674MН3 | T001 MН4 | Ptr1 MН3 |
| MН4 | MН4 | 0 | S820MН4 | T001 MН4 | Ptr1 MН4 |
| MН5 | MН6 | 2 | S502MН5 | T002 MН4 | Ptr1 MН5 |
| MН6 | MН6 | 1 | S186MН6 | T001 MН4 | Ptr1 MН6 |
| MН7 | MН6 | 2 | S238MН7 | T002 MН4 | Ptr1 MН7 |
| MН8 | MН6 | 3 | S160MН8 | T002 MН4 | Ptr1 MН8 |

Предности на DSDV протоколот

- DSDV гарантира патеки без да влегува во циклус.
- Проблемот со броење до бесконечност е редуциран во DSDV протоколот.
- DSDV ја поддржува само најдобрата патека, наместо повеќе патеки до секоја дестинација.

Ограничувања на DSDV протоколот

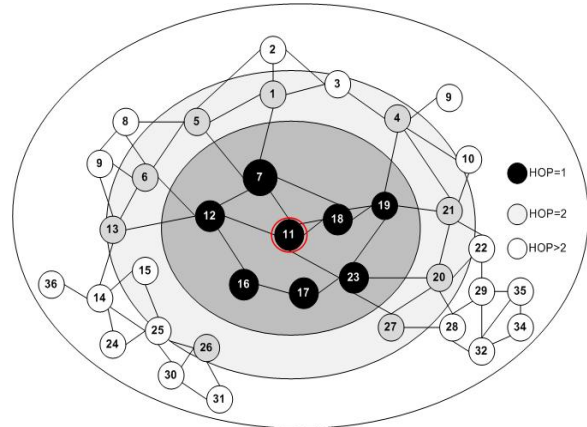
- Загуба на пропусност за време на непотребно претставување на рутирачката информација, дури и ако нема промена во мрежната топологија.
- DSDV не поддржува рутирање на повеќе патеки.
- Тешко е да се определи времето на одложување за претставувањето на патеките [1].

2.2. Fisheye State Routing Protocol (FSR)

Fisheye State Routing е протокол управуван со табели и е адаптиран за безжичните Ad Hoc средини. Обезбедува имплицитна хиерархиска рутирачка структура.

Преку линкот за ажурирање на информацијата со различни фреквенции, во зависност од fisheye растојанието, FSR се држи добро за поголеми мрежи и го одржува пречекорувањето на ниско ниво, без да се загрози пресметаната рутирачката

точност кога дестинацијата е блиску. Како резултат на тоа, FSR е поприфатлив за поголемите мобилни мрежи, каде мобилноста е висока и опсегот е помал. Со избирање на соодветен број на нивоа и големина на радиусот, FSR докажува дека е пофлексибилно решение за предизвикот на одржување на точните патеки во Ad Hoc мрежите.



Слика 2. Изглед на Fisheye механизмот

Секој јазол кој го користи FSR протоколот е должен да ги одржува следните табели:

Тополошка табела (Topology table)

Тополошката табела содржи тополошки информации добиени од пораките за состојба за линк.

Соседна листа за состојба на линкот

Кога јазолот добива порака за состојба на линкот, ја ажурира информацијата за состојба на линкот на испраќачот во неговата соседна табела.

Рутирачка табела

Рутирачката табела на FSR протоколот го покажува наредниот сет информации за препраќање на пакетите до другите дестинации во мрежата.

Предности на FSR протоколот:

- Едноставност,
- Користење на ажурирани кратки патеки,
- Робусност на хост мобилноста [2].

2.3. Рутирачки Протокол со Оптимизирана Состојба на Линкот (OLSR)

OLSR е проактивен рутирачки протокол со состојба на линкот кој ги користи пораките Hello и Topology Control (TC) за откривање, а потоа и за дистрибуирање на информацијата за состојба на линкот низ мобилната Ad Hoc мрежа.

Секој јазол селектира множество од своите соседни јазли како “повеќестепени релиња” (MPR’s). Идејата на MPR’s е да ги минимизираат дополнителните непрекинати пораки во мрежата преку редуцирање на препраќањата во истиот регион. Всушност, единственото барање за OLSR протоколот да обезбеди најкратка патека до сите дестинации е MRP јазлите да декларираат

информација за состојбата на линкот за своите селектори.

OLSR може да ја оптимизира реактивноста во тополошка промена преку намалување на максималното време за периодична контрола на пораките што се пренесуваат [3] [4].

2.4. Cluster-head gateway switch routing protocol (CGSR)

CGSR го користи DSDV протоколот како главен протокол. Мобилните јазли се поделени во кластери и главниот дел на кластерот е избран користејќи дистрибуиран алгоритам.

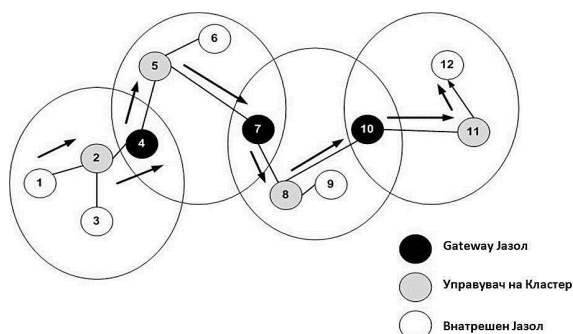
Јазол кој е во комуникацискиот опсег на два или повеќе главни делови на кластерот се нарекува gateway јазол. CGSR го користи Least Cluster Change (LCC) алгоритмот. Промена на главниот дел на кластерот се случува само кога два главни делови на кластерот доаѓаат во еден кластер или еден од јазлите се движи надвор од опсегот на сите главни делови на кластерот.

Секој јазол одржува 2 табели:

- Табела за член на кластер,
- DV рутирачка табела.

Рутирање во CGSR

Ако еден јазол треба да рутира пакет, го наоѓа најблискиот главен дел на кластерот по должината на патеката до дестинацијата, во согласност со табелата за член на кластерот и рутирачката табела. Потоа ќе ја спореди својата рутирачка табела за да ја пронајде наредната стапка, со цел да се достигнат главните делови на кластерот избрани погоре и потоа го испраќа пакетот на јазолот.



Слика 3. Рутирање во CGSR

Рутирањето во CGSR е многу поефикасно отколку во DSDV, бидејќи се прави преку главните делови на кластерите и gateway-и.

Распределени безжични канали

Методот со користење на кластери обезбедува ефикасен начин за распределување на безжични канали помеѓу различни кластери.

Со користење на CDMA можно е да се подобрат повторните употреби на просторот меѓу различни групи.

Рутирањето во CSGR е поефикасно отколку во другите DV протоколи, што значи дека големината на табелата за рутирање е намалена само преку чување на еден запис за еден краен кластер.

2.5. Безжичен Рутирачки Протокол

Безжичниот рутирачки протокол (WRP) е протокол во мобилните Ad Hoc мрежи кој што е управуван со табели.

WRP користи подобрена верзија на DSDV протоколот, кој го користи Bellman-Ford алгоритмот за пресметување на патеките.

За да се спротостави на проблемот “броење до бесконечност” и да овозможи побрза конвергенција, се воведува уникатен метод за одржување на информацијата во однос на најкраткото растојание до секој јазол во мрежата и претпоследната стапка на патеката на јазолот до секој краен јазол.

WRP се разликува од DSDV во одржувањето на табелите и во процедурите за ажурирање. Додека DSDV поддржува само еден топологичен тип на табели, WRP користи множество од табели за одржување на поголема информација.

Табели кои се поддржуваат од јазлите се:

- distance табела (DT)
- рутирачка табела (RT)
- табела за загуби на линкот (LCT)
- листа за препаќање на порака (MRL).

WRP ги има истите предности како и DSDV. Покрај ова, има побрза конвергенција и вклучува помалку ажурирани табели. WRP бара поголема меморија за складирање и одржување на своите ресурси во табелите [5] [6].

3. ПРОТОКОЛИ НА БАРИЊЕ

Овие протоколи претставуваат значаен дел од мобилните Ad Hoc мрежи. Начинот на кој овие протоколи функционираат е поразличен од начинот на кој функционираат протоколите управувани со табели.

Глобално, може да се каже дека за да се овозможи комуникација помеѓу два уреди, било кој од овие протоколи ја преплавува самата мрежа со пакети кои содржат барања за патека. Ова се случува бидејќи кај овие типови на рутирање не постои претходно одредена патека по која ќе се испраќаат информации, туку таа се креира во оној момент кога има потреба од неа. Ова е овозможено со начинот на кој функционираат самите мобилни Ad Hoc мрежи. Секое јадро ги слуша соседните јадра во мрежата и кога има потреба од комуникација јадрото кое треба да прати информации испраќа барања до сите јадра во мрежата. На спротивната страна, бидејќи сите јадра слушаат и очекуваат барања, првото слободно јадро одговара и на тој начин се одредува патеката по која ќе се пратат податоците. Самата патека постои додека не се

заврши препраќањето на податоците или се додека постои можност за комуникација помеѓу двете јадра [7].

Од ова може да се заклучи дека примарната предност на овие протоколи, па и на самите Ad Hoc безжични мрежи, е можноста да се направи вмрежување многу брзо и на многу едноставен начин. Но, исто така, преплавувањето на мрежата со барања за рутирање доведува до поголема латентност при пронаоѓањето на патеките што во краен случај може и да значи преоптоварување или “гушење” на самата мрежа. Последново се оценува како голема негативна карактеристика на Протоколите на барање [7].

Во овој дел од трудот ќе опишеме некој од овие Протоколи на барање.

3.1 Рутирање Според Стабилноста на Сигналот (SSR)

Овој протокол, како што кажува и неговото име, ја бира патеката по која ќе се праќаат податоците според јачината на сигналот и стабилноста на локацијата на самиот јазол, самото јадро во мрежата. Тоа значи дека овој протокол ќе ги прати податоците до оној јазол во мрежата каде што конективноста е најголема. SSR се состои од два протоколи кои соработуваат меѓу себе за да го овозможат рутирањето на горенаведениот начин.

Првиот од нив е динамичкиот рутирачки протокол. Негова задача е периодично да ги проверува сигналите на соседните јазли и да ја забележува нивната состојба како канал со јак или слаб сигнал. Сите трансакции на податоци најпрво ги прима овој протокол и ги процесира, со тоа што одредува кој е посакуваниот примател на податоците и каков е сигналот во неговите јадра. Од кога ќе заврши со оваа проверка тој го предава барањето на статичкиот рутирачки протокол.

Овој протокол е вториот кој го сочинува SSR. Неговата функција се состои од тоа секое барање примено од динамичкиот рутирачки протокол да го проследи до посакуваната дестинација. Ако се случи да нема податоци за патеката до дестинацијата, тогаш овој протокол прави пребарување на патекети до посакуваната дестинација и ќе ја избере онаа патека која што динамичкиот протокол ја означил како канал со силен сигнал. Ако се случи да не пронајде таква крајна точка, тогаш преминува на пребарување на канали со слаб сигнал, бидејќи тие се единствените преостанати канали преку кои може да се случи трансакција на податоци. Кога некој од јазлите во мрежата би ја променил својата локација или би станал недостапен тогаш низ мрежата се емитува “link failure” порака и динамичкиот рутирачки протокол ги променува своите претходно зачувани податоци за овој јазол и ги препушта на статичкиот, со што би се избегнала патека до линк во мрежата со кој во тој момент неможе да се комуницира или не се наоѓа на таа локација [8].

3.2 Рутирање Базирано на Асоцијативност (ABR)

Овој протокол за рутирање, на некој начин слично како и SSR протоколот, ја користи асоцијативноста или поврзаноста на јазлите во мрежата како примарен принцип за создавање на патека.

Сите јазли во мрежата периодично праќаат сигнали дека постојат, односно дека се достапни за комуникација. Со овој протокол секој јазол ги забележува овие сигнали во свои табели за асоцијативност и на основа на овие табели ја одредува патеката по која ќе се препратат информации до саканата цел. Значи, секој јазол константно ја обновува својата табела за асоцијативност во однос на сите соседни јазли. Секое примање на периодичниот сигнал се запишува во овие табели, односно ја зголемува вредноста која веќе е запишана во табелата, со цел да се одржат податоците за конкретниот јазол од овој сигнал. Ова значи дека тој јазол има помал степен на мобилност или повеќе пати го пратил истиот сигнал од приближно иста локација. Спротивно на ова, колку е помала вредноста во асоцијативната табела за конкретен јазол, значи дека тој јазол се карактеризира со поголем степен на мобилност.

ABR протоколот подразбира три фази во процесот на рутирање: Откривање на патека, реконструкција на патека и бришење на патека.

Во фазата на откривање на патека јазолот од кој треба да се пратат податоци испраќа сигнал до јазлите кои се на патот до крајната цел на комуникацијата. Овој сигнал, така наречен Прашање за Пренесување (BQ), стигнува до првиот јазол кој во него ги запишува својата адреса и асоцијативност со јазолот кој го праќа сигналот и го препушта сигналот на наредниот јазол кој го прави истото. На овој начин јазолот, кој претставува крајна цел на комуникацијата, може да ја избере патеката по која ќе се врши праќање на податоци според асоцијативноста на јазлите во мрежата.

До фазата на реконструкција на патека доаѓа кога некој од јазлите во мрежата ќе ја промени својата локација. Тогаш останатите јазли за него ќе примат нови асоцијативни вредности и одново ќе треба да се повтори фазата за откривање на патека.

Фазата на бришење на патека претставува фаза во која се бришат претходно утврдени патеки кои веќе не се потребни, праќањето на податоци е завршено или доколку дошло до некаква промена, односно некој од јазлите ја променил својата локација [9,10].

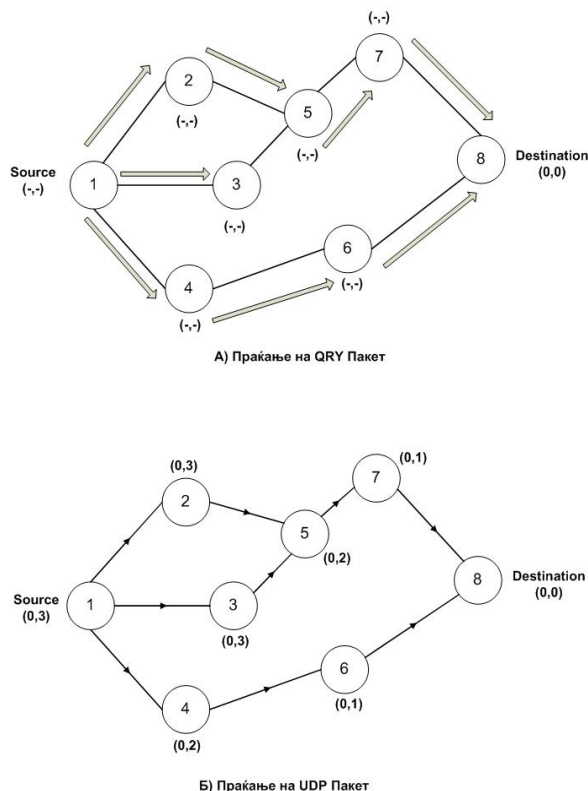
3.3 Привремено Подреден Рутирачки Алгоритам (TORA)

TORA за основа на рутирање ја користи промената на линковите во мрежата, односно промената на позициите на самите јазли. Тоа

значи дека секој јазол се карактеризира со одредени параметри кои овозможуваат или ја карактеризираат неговата состојба во мрежата. Такви параметри, на пример, се логичното време на така наречен “link failure”, уникатниот ID на јазолот и уникатниот ID кој го одредува неговото ниво на референтност, индикатор за рефлексија и параметрот за подредување. Овие параметри се потребни бидејќи овој протокол или алгоритам функционира на тој принцип што од јазолот од кој треба да се прати некаков податок се одредуваат сите можни патишта до јазолот кој претставува крајна дестинација.

Рутирањето со овој алгоритам опфаќа три елементи: Креирање на патека, нејзино одржување и нејзино бришење.

Креирањето на патеките се врши со емитување на QRY и UPD пакети. Алгоритмот ги поставува параметрите за подреденост како недефинирани на сите јазли, освен на јазолот кој треба да е дестинација. Кај него овој параметар добива вредност 0. Јазолот кој претставува извор на комуникацијата праќа QRY пакет кој може да биде примен само од јазол со дефиниран параметар за подреденост, што во овој случај претставува јазолот кој е крајна дестинација. Овој јазол одговара на примениот пакет со праќање на UPD кон сите останати јазли во мрежата. Секој јазол кој ќе го прими овој пакет го зголемува својот параметар за подреденост за 1 во однос на параметарот на оној кој го праќа пакетот. На овој начин се креираат патеките помеѓу изворот и крајната дестинација. Овој процес е илустриран на Слика 4.



Слика 4. Креирање на патеки со TORA

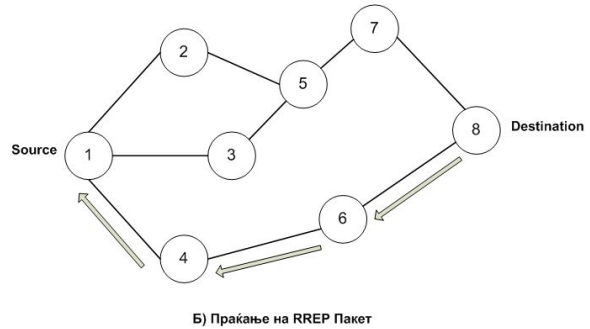
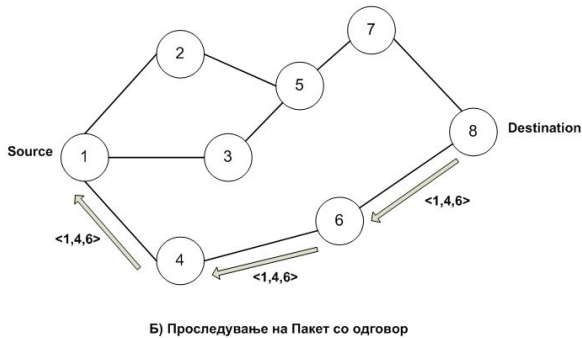
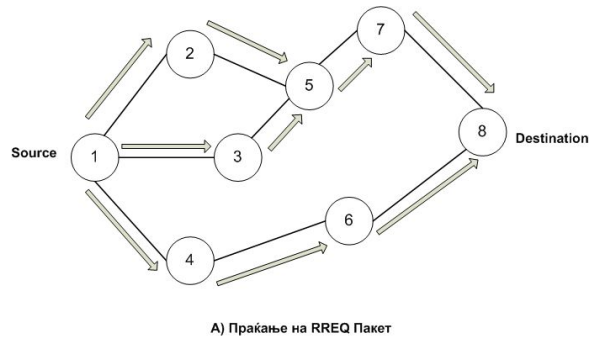
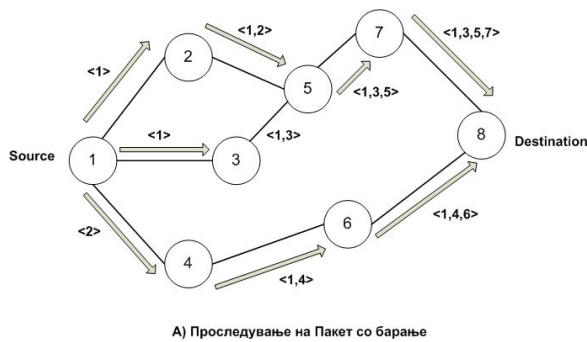
Одржувањето на патеката настанува кога некој од јазлите ќе ја промени својата локација и со тоа ќе го промени својот уникатен ID кој го одредува неговото ниво на референтност. Бришењето пак на патекети се случува на тој начин што TORA испраќа CLR порака до сите јазли во мрежата и нивните параметри на подреденост се бришат. Ова се случува кога патеката не е веќе потребна или е нефункционална поради промени на локациите на јазлите [11].

3.4 Протокол за Рутирање со Динамички Извор (DSR)

Принципот на кој функционира овој протокол се состои од две поголеми фази, односно од фазата на откривање на патека и фазата на одржување на патека.

Генерално ова се изведува на тој начин што секое јадро од мрежата ги памети можните извори на патеки во мрежата и ги променува овие податоци во истиот момент кога некое јадро ќе ја промени својата локација и со тоа ќе ја смени можната патека, односно можниот извор.

Кога некое јадро треба да прати податоци до одредена дестинација, прво проверува дали во својата кеш меморија има информации за веќе постоечка патека до таа дестинација и ако таква патека постои, тогаш тоа јадро ја користи истата за праќање на податоците. Во случај кога не постои патека, односно јадрата ги промениле своите локации, јадрото кое претставува извор праќа пакет со барање во кој се содржат адресите на изворот и дестинацијата и уникатен идентификувачки број за таа трансакција. Барањето доаѓа до наредното јадро во мрежата и тоа проверува во својата кеш меморија дали има веќе постоечка патека до дестинацијата. Ако пронајде патека, тогаш таа се користи за трансакција, а ако не, го проследува барањето понатаму со тоа што во пакетот ја додава и својата адреса. Ова додавање на адресата е задолжително за да се избегне повторно проверување за истото барање од истото јадро, ако тоа барање повторно стигне до него. Овој процес се повторува се додека пакетот со барањето не стигне до дестинацијата или до некое јадро кое има информации за истата. Бидејќи во пакетот со барањето веќе има информации за сите дотогашни јадра кои го виделе овој пакет, може да се формира патека до дестинацијата. Патеката се праќа во нов пакет кој е одговор на барањето и треба да стигне до изворот. Ако некои од локациите се променети и се изгубат податоците за изворот, тогаш овој пакет се проследува на ист начин како и пакетот со барањето, се додека не се формира патека до изворот. Ова воедно е фазата на откривање на патека, која е прикажана на Слика 5.



Слика 5. Откривање на патеки со DSR протоколот

Слика 6. Откривање на патека со AODV протоколот

Фазата за одржување на патека користи два пакети. Едниот е пакет за грешка, а другиот пакет за потврда. Кога некое јадро нема да може да прати информации до наредното јадро од патеката, бидејќи тоа јадро е недостапно или ја променило локацијата, тогаш во својата кеш меморија ги брише податоците за патеката, генерира пакет за грешката и го проследува низ мрежата за и останатите јадра да ја избришат оваа патека од нивниот кеш. Спротивно на ова, периодично се проверуваат патеките со испраќање на пакети за потврда. Овие пакети се проследуваат низ мрежата се додека не дојде до некаква грешка во патеката. Тогаш наместо пакетот за потврда се проследува пакетот за грешка [12,13].

3.5 Протокол За Рутирање Со Ад Нос Вектор на Оддалеченост На Барање (AODV)

Овој протокол ги открива патеките по потреба, односно кога има извор. Откривањето на патеката со одвива со праќање на пакети со барања или Route Request Packet (RREQ) и пакети со одговор или Route Reply Packet (RREP).

Јазолот кој е извор емитува пакет со барање. Овој пакет го примаат соседните јазли и го проследуваат понатаму се додека не се дојде до дестинацијата или додека не се дојде до јадро кое содржи некакви информации за дестинацијата. Овој пакет со барање содржи секвенционални броеви за да се избегне повторување во мрежата и на тој начин секој јазол што претходно го видел ова барање не го прегледува. Истото се користи и за добивање на временски актуелна информација кога се генерира и проследува пакетот со одговор на истото барање. Генерирањето на одговорот, како што споменавме, се случува кога барањето ќе стигне до дестинацијата или до некој соседен јазол кој содржи информации за дестинацијата. Патеката за движење на овој пакет назад до јазолот што е извор е веќе создадена, бидејќи секој јазол кој го проследува пакетот со барање во своите табели запишува од каде доаѓа ова барање. На овој начин се одвива откривањето на патеката помеѓу изворот и дестинацијата во мрежата. Истиот процес е прикажан на Слика 6.

Ако се случи некој од јазлите да ја промени својата локација, тогаш неговиот соседен јазол неможе да прати информации до него и веднаш емитува известување за погрешен линк во патеката до оние јазли кои претходеле во низата на праќање. Ова известување стигнува до изворот и тој повторно ја иницира фазата на откривање на патека. Доколку се случи самиот извор да ја промени својата локација, тогаш тој самиот ја иницира истата фаза од почеток.

Се на се, може да се каже дека главна предност на овој протокол е тоа што рутирањето се одвива само по барање на изворот, но од друга страна,

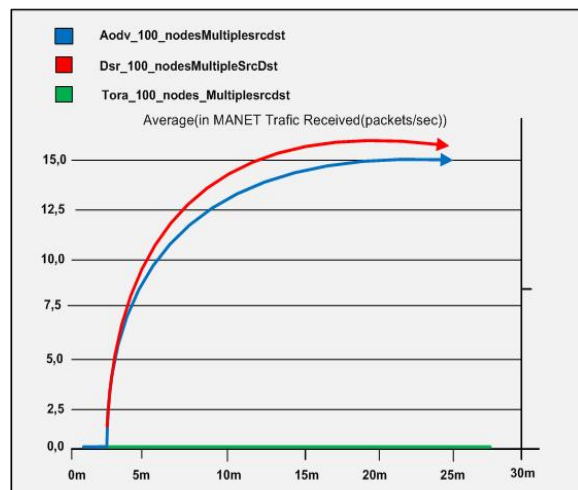
како негативна карактеристика, можат да се наведат проблемите кои се случуваат кога некои од јазлите во мрежата ќе ги променат своите позиции и секвенционални броеви. Ако се случи изворот да ги задржи старите секвенционални броеви за овие јазли, бидејќи праќањето на податоците се одвива пред периодичното обновување на истите броеви, тогаш во мрежата се јавуваат неконзистентни, односно непостоечки патеки. Исто така, возможно е да се случи и непотребно оптоварување на мрежата и искористување на нејзиниот “пропусен опсег” во ситуациите кога до изворот стигнуваат голем број на пакети со одговор за само еден испратен пакет со барање [14].

4. СИМУЛАЦИЈА НА РУТИРАЊЕ СО AODV, DSR И TORA

За нашата симулација користен софтверот OPNET Modeler, чија главна намена е симулирање на мрежи. За оваа симулација создадовме повеќе сценарија за наведените протоколи, како би можеле да ги споредиме добиените резултати. Секое сценарио кое го создадовме за оваа симулација се содржи од сто јазли, распоредени на површина од 16000000 метри квадратни (4000м *4000м). Во секое сценарио 40 јазли претставуваат парови од извор и дестинација, 30 јазли се мобилни, а останатите 30 се статични јазли во мрежата кои не се ниту извор, ниту дестинација.

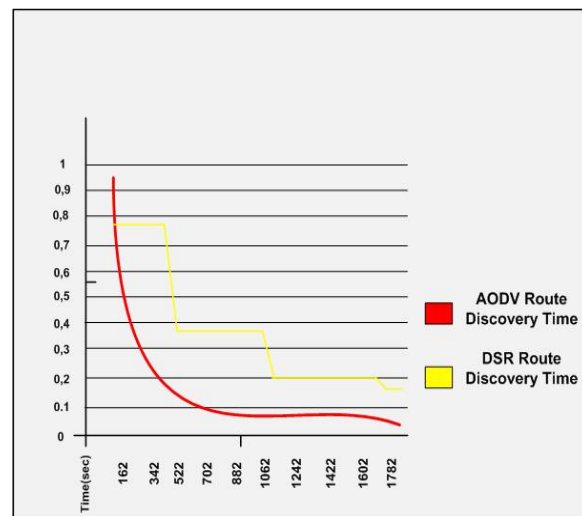
Јазлите кои во овие сценарија се мобилни започнуваат да се движат по изминати 260 секунди од почетокот на симулацијата. Тие се движат според Random Waypoint Mobility Model, кој OPNET Modeler го нуди како опција. Првиот пакет со барање изворите го испраќаат по 100 секунди од почетокот на симулацијата и продолжуваат со генерирање на сообраќај се до крајот на симулацијата. Целата симулација трае 30 минути. Она што ние сакаме да го добиеме од оваа симулација се разликите во перформансите помеѓу AODV, DSR и TORA рутирачките протоколи. За таа цел како резултати за споредба од различните сценарија се земени неколку параметри и тоа целосниот примен сообраќај, бројот на скокови по патека и времето потребно за откривање на патека.

На Сликата 7 се прикажани резултатите од целосниот примен рутирачки сообраќај во оваа симулација според трите протоколи.

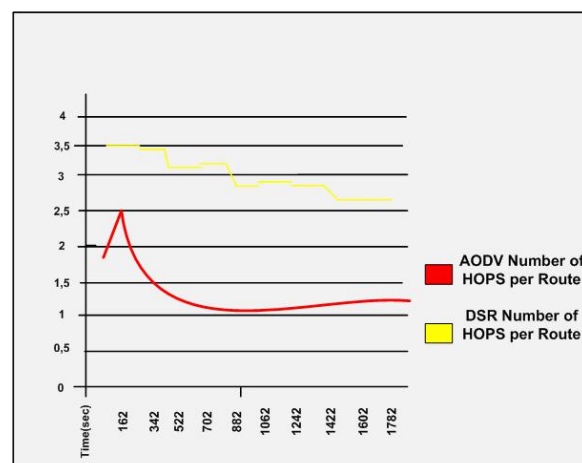


Слика 7. Споредба на целосниот примен сообраќај

Од Слика 7 може да се забележи дека AODV и DSR приближно подеднакво се справуваат со пакетите испратени низ мрежата и дека нивната ефикасност се зголемува како што изминува времето, односно тие одлично се справуваат со примањето на поголем број на пакети, додека тоа не случај и со TORA, каде што воопшто не е возможно да оствари сообраќај.



Слика 8. Споредба на времето потребно за откривање на патека



Слика 9. Споредба на бројот на скокови по патека

На Сликите 8 и 9 се прикажани резултатите од времето потребно за откривање на патека и бројот на скокови по патека за AODV и DSR рутирачките протоколи, додека протоколот TORA е изоставен поради тоа што не остварува никаков сообраќај.

Од овие слики може да се забележи дека AODV функционира многу подобро отколку DSR. Потребно му е помалку време за откривање на патека и помалку скокови по патека.

Од оваа симулација, односно според бројот на примени пакети, времето потребно за откривање на патека и бројот на скокови по патека, може да се заклучи дека во пообемни Ad Hoc мрежи подобар избор за рутирачки протоколи би биле AODV или DSR, отколку TORA.

Сепак, споредувајќи ги резултатите на AODV и DSR според зададените параметри, лесно е воочливо дека рутирачкиот протокол AODV нуди подобри карактеристики во ситуацијата кога го споредуваме нивното однесување во случајот на одредување на времето потребно за откривање на патека и бројот на скокови по патека, што покажува дека AODV би бил најдобар избор.

5. ЗАКЛУЧОК

Во овој труд беа опишани неколку постоечки рутирачки протоколи за Ad Hoc мрежи.

Беа разгледани следните категории на протоколи: *Протоколи управувани со табели* и *Протоколи на барање*.

Кај *Протоколите управувани со табели*, секој уред одржува ажурирана рутирачка информација до сите јазли во мрежата, додека пак, кај *Протоколите на барање*, јазолот ја пронаоѓа патеката до дестинацијата тогаш кога сака да прати пакети до неа.

DSDV протоколот ги користи крајните секвенционални броеви, за да го сочува циклусот со патеките слободен и навремено ажуриран.

FSR протоколот ја намалува големината на табелите која се менува преку одржување на помалку прецизни информации за јазлите.

CGSR е рутирачки протокол кој се заснова на кластери, каде што јазлите се групирани во групи. CBRP е рутирачки алгоритам кој, исто така, се базира на кластери и за разлика од CGSR протоколот, кој е протокол управуван со табели, овој протокол е протокол на барање.

AODV е протокол на барање - верзија на DSDV протоколот.

DSRP протоколот е рутирачки механизам на извор, каде што патеката е во секој пакет.

ABR протоколот користи степен на асоцијативност за селектирање на патеките. Слично на него, SSR протоколот избира патеки врз основа на јачината на сигналот.

Она што може да се заклучи од симулацијата е дека кога станува збор за AODV, DSR и TORA, секој од овие рутирачки протоколи би бил оптимален во одредена ситуација. Сепак, кога станува збор за мрежа со голем број на мобилни јазли, може да се заклучи дека користењето на AODV би било препорачливо додека користењето на TORA би требало да се избегнува.

6. ЛИТЕРАТУРА

- [1] C.E.Perkins and P.Bhagwat, "Highly Dynamic Destination-Sequenced Distance Vector(DSDV) for Mobile Computers." Comm.Commun Rev.Oct.1994.
- [2] G.Pei, M.Gerla, and T.-W. Chen, "Fisheye State Routing: A routing scheme for Ad Hoc Wireless Networks", Proceedings of ICC 2000, New Orleans, LA, Jun.
- [3] Extensions to OSPF to Support Mobile Ad Hoc Networking, Madhavi Chandra, Abhay Roy, 21-Sep-08.
- [4] P.Jacquet, P.Minet, P.Muhlethaler, N.Rivierre. Increasing reliability in calbe free radio LANs: Low level forwarding in HIPERLAN. Wireless Personal Communications, 1996.
- [5] Murthy, Shree; Garcia-Luna-Aceves, J. J. (1996-10-01), "An efficient routing protocol for wireless networks", *Mobile Networks and Applications* (Hingham, MA: Kluwer Academic Publishers) 1 (2): 183–197, doi:10.1007/BF01193336.
- [6] S.Murthy and J.J.Garcia-Luna –Aceves, "An efficient Routing Protocol for Wireless Networks," ACM Mobile Networks and App. J., Special Issue on Routing in Mobile Communication Networks, Oct. 1996.
- [7] <http://www.wikipedia.org/>
- [8] R. Dube et al., "Signal Stability based adaptive routing for Ad Hoc Mobile networks", IEEE Pers. Comm., Feb. 1997.
- [9] C.-K. Toh, "Long-lived Ad-Hoc Routing based on the concept of Associativity" March 1999.
- [10] Chai-Keong Toh, "A novel distributed routing protocol to support Ad Hoc mobile computing" Proc. 1996 IEEE 15th Annual Int'l. Phoenix Conf. Comp. and Commun., Mar. 1996.
- [11] VD Park and MS Corson "A highly adaptive distributed routing algorithm for mobile wireless networks", Proc. INFOCOM'97, Apr. 1997.
- [12] David B. Johnson, Davis A. Maltz, "The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks" October 1999.
- [13] David B. Johnson, Davis A. Maltz, "Dynamic Source Routing in Ad Hoc Networks", Mobile Computing, T. Imielinski and H. Korth, Eds., Kulwer, 1996.

- [14] Charles E. Perkins, Elizabeth M. Royer, Samir R. Das, "Ad Hoc On-demand Distance Vector Routing", October 99.

Routing Algoritams for real-time services in MANET

Boris Mihajlov, Dimitar Sazdovski, M-r Mitko Bogdanovski, Doc. D-r Saso Gelev

Европски Универзитет – Скопје, Р. Македонија

mihajlov.boris@live.eurm.edu.mk, dimitarsazdovski@hotmail.com, {mitko.bogdanoski,
saso.gelev}@eurm.edu.mk

***Abstract:** In this paper we talk about the routing algoritams and protocols in the mobile Ad Hoc networks. The paper is devided into two major sections. The first one discusses several table-driven protocols and the second one several on-demand. In general the paper offers a suficient introduction in the ways that both, the table-driven protocols and the on-demand protocols, rout informations in the mobile Ad Hoc networks. Informations and illustrations are given on how these protocols discover, maintain routes and how they delete non-exisitng ones. The conclusion of this paper is in a way a colision between the two major sections and offers a brief comparison of the some how similar protocols.*

***Key Words:** Mobile Ad Hoc Networks (MANET's), routing protocols, multi-hop, Route timeout*