Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií

FIIT-5220-72122

Bc. Martin Ilavský

SIMULÁCIE ÚTOČNÝCH SCENÁROV V MOBILNÝCH AD HOC SIEŤACH

Diplomová práca

Študijný program:	Softvérové inžinierstvo
Študijný odbor:	9.2.5 Softvérové inžinierstvo
Miesto vypracovania:	Ústav počítačových systémov a sietí, FIIT STU v Bratislave
Vedúci práce:	Ing. Jozef Filipek

máj 2018

Zadanie diplomovej práce

Meno študenta: **Bc. Martin Ilavský**

Študijný program: Softvérové inžinierstvo

Študijný odbor: Softvérové inžinierstvo

Názov práce: **Simulácie útočných scenárov v mobilných ad hoc sieťach**

Samostatnou výskumnou a vývojovou činnosťou v rámci predmetov Diplomový projekt I, II, III vypracujte diplomovú prácu na tému, vyjadrenú vyššie uvedeným názvom tak, aby ste dosiahli tieto ciele:

Všeobecný cieľ:

Vypracovaním diplomovej práce preukážte, ako ste si osvojili metódy a postupy riešenia relatívne rozsiahlych projektov, schopnosť samostatne a tvorivo riešiť zložité úlohy aj výskumného charakteru v súlade so súčasnými metódami a postupmi študovaného odboru využívanými v príslušnej oblasti a schopnosť samostatne, tvorivo a kriticky pristupovať k analýze možných riešení a k tvorbe modelov.

Špecifický cieľ:

Vytvorte riešenie zodpovedajúce návrhu textu zadania, ktorý je prílohou tohto zadania. Návrh bližšie opisuje tému vyjadrenú názvom. Tento opis je záväzný, má však rámcový charakter, aby vznikol dostatočný priestor pre Vašu tvorivosť.

Riaďte sa pokynmi Vášho vedúceho.

Pokiaľ v priebehu riešenia, opierajúc sa o hlbšie poznanie súčasného stavu v príslušnej oblasti, alebo o priebežné výsledky Vášho riešenia, alebo o iné závažné skutočnosti, dospejete spoločne s Vaším vedúcim k presvedčeniu, že niečo v texte zadania a/alebo v názve by sa malo zmeniť, navrhnite zmenu. Zmena je spravidla možná len pri dosiahnutí kontrolného bodu.

Miesto vypracovania: Ústav počítačového inžinierstva a aplikovanej informatiky, FIIT STU Bratislava

Vedúci práce: **Ing. Jozef Filipek**

Termíny odovzdania:

Podľa harmonogramu štúdia platného pre semester, v ktorom máte príslušný predmet (Diplomový projekt I, II, III) absolvovať podľa Vášho študijného plánu

Predmety odovzdania:

V každom predmete dokument podľa pokynov na www.fiit.stuba.sk v časti: home > Informácie o > štúdiu > organizácia štúdia > diplomový projekt.

V Bratislave dňa 13. 2. 2017

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
Fakulta informatiky a informačných technológií
Ilkovičova 2, 842 16 Bratislava 4
1

prof. Ing. Pavol Návrat, PhD.
riaditeľ Ústavu informatiky, informačných systémov a softvérového inžinierstva

STU FIIT
SLOVENSKÁ TECHNICKÁ
UNIVERZITA V BRATISLAVE
FAKULTA INFORMATIKY
A INFORMAČNÝCH TECHNOLÓGIÍ

Návrh zadania diplomovej práce

Finálna verzia do diplomovej práce [1]

Študent:

Meno, priezvisko, tituly: Martin Ilavský, Bc.
Študijný program: Softvérové inžinierstvo
Kontakt: ilavskymartin@yahoo.com

Výskumník:

Meno, priezvisko, tituly: Jozef Filipek, Ing.

Projekt:

Názov: Simulácie útočných scenárov v mobilných ad hoc sieťach
Názov v angličtine: Simulation of attack scenarios in mobile ad hoc networks
Miesto vypracovania: Ústav počítačového inžinierstva a aplikovanej informatiky, FIIT STU, Bratislava
Oblasť problematiky: bezpečnosť počítačových sietí

Text návrhu zadania[2]

MANET (MANET - Mobile Ad Hoc Network) siete sú vytvárané mobilnými zariadeniami bez predchádzajúcej sieťovej infraštruktúry a závislosti od centralizovaných zdrojov. Charakteristické sú vysokou mobilitou uzlov, častými zmenami topológie a limitovanými zdrojmi uzlov z hľadiska výpočtového výkonu a kapacity batérie. Nevýhodou MANET sietí je veľká zraniteľnosť voči útokom zvnútra aj zvonka. Z tohto dôvodu sa vyvíjajú zabezpečené verzie smerovacích protokolov, čo je motiváciou na návrh testovacích scenárov na overenie navrhovaných riešení. Analyzujte aktívne útoky na siete MANET a porovnajte ich z viacerých hľadísk ako efektivita, spoľahlivosť a stupeň poškodenia siete. Analyzujte a porovnajte smerovacie protokoly v MANET sieťach aj s ich zabezpečenými verziami, pričom sa zamerajte na reaktívne protokoly. Definujte požiadavky útočných scenárov a navrhnite útoky na overenie bezpečnostných mechanizmov. Pri návrhu sa zamerajte na modifikácie existujúcich útokov a na útoky využívajúce spoluprácu viacerých uzlov v sieti. Určte odolnosť zanalyzovaných smerovacích protokolov vzhľadom na navrhnuté scenáre. V prípade, že budú smerovacie protokoly pre vytvorené scenáre zraniteľné, uveďte dôvody týchto zraniteľností a porovnajte rozsah poškodenia sietí medzi protokolmi. Scenáre implementujte v testovacom prostredí pomocou sieťového simulátora a overte ich na zanalyzovaných smerovacích protokoloch.

---

[1] Vytlačiť obojstranne na jeden list papiera

[2] 150-200 slov (1200-1700 znakov), ktoré opisujú výskumný problém v kontexte súčasného stavu vrátane motivácie a smerov riešenia

Literatúra[3]

- M. Pateł, S. Sharma, Detection of malicious attack in MANET a behavioral approach, Advance Computing Conference (IACC), 2013
- L. Songbai, SAODV: a MANET routing protocol that can withstand black hole attack, Computational Intelligence and Security, 2009

Vyššie je uvedený návrh diplomového projektu, ktorý vypracoval(a) Bc. Martin Ilavský, konzultoval(a) a osvojil(a) si ho Ing. Jozef Filipek a súhlasí, že bude takýto projekt viesť v prípade, že bude pridelený tomuto študentovi.

V Bratislave dňa 22.1.2017

Podpis študenta

Podpis výskumníka

Vyjadrenie garanta predmetov Diplomový projekt I, II, III

Návrh zadania schválený: áno / nie[4]

Dňa: 13. 2. 2017

Podpis garanta predmetov

[3] 2 vedecké zdroje, každý v samostatnej rubrike a s údajmi zodpovedajúcimi bibliografickým odkazom podľa normy STN ISO 690, ktoré sa viažu k téme zadania a preukazujú výskumnú povahu problému a jeho aktuálnosť (uveďte všetky potrebné údaje na identifikáciu zdroja, pričom uprednostnite vedecké príspevky v časopisoch a medzinárodných konferenciách)

[4] Nehodiace sa prečiarknite

Čestne vyhlasujem, že som túto prácu vypracoval samostatne, na základe konzultácií a s použitím uvedenej literatúry.

V Bratislave, 30.4.2018

Martin Ilavský

Poďakovanie

Týmto sa chcem poďakovať vedúcemu diplomovej práce Ing. Jozefovi Filipekovi, za odbornú pomoc a cenné rady, ktoré mi poskytol pri jej vypracovaní.

Anotácia

Slovenská technická univerzita v Bratislave
FAKULTA INFORMATIKY A INFORMAČNÝCH TECHNOLÓGIÍ
Študijný program: Softvérové inžinierstvo

Autor: Bc. Martin Ilavský
Diplomová práca: Simulácie útočných scenárov v mobilných ad hoc sieťach
Vedúci diplomovej práce: Ing. Jozef Filipek
máj, 2018

Práca sa zaoberá bezpečnosťou smerovacích protokolov v MANET sieťach. Analyzuje vlastnosti MANET sietí, zabezpečené a nezabezpečené verzie reaktívnych smerovacích protokolov, ktoré využívajú zabezpečenie pomocou asymetrickej aj symetrickej kryptografie a útoky používané v MANET sieťach rozdelené podľa vrstiev modelu RM OSI. Nechýbajú ani zhodnotenia jednotlivých častí, ktoré obsahujú porovnanie zanalyzovaných smerovacích protokolov a ich odolnosť proti zanalyzovaným útokom. Praktická časť diplomovej práce sa zaoberá návrhom nových útokov s využitím spolupráce medzi viacerými útočnými uzlami v sieti. Práca obsahuje aj koncepty zabezpečení proti týmto návrhom. Navrhnuté boli štyri nové efektívne útoky, ktoré by mali dôkladne preveriť zabezpečenie smerovacích protokolov. Útoky je možné vykonať v rôznych navrhnutých režimoch, ktoré tvoria testovací scenár. Práca obsahuje ich podrobný popis spolu s diagramami a názornými obrázkami. Účinnosť a efektívnosť navrhnutých útokov boli dokázané v simulačnom prostredí na vybraných smerovacích protokoloch. Pri každom scenári sme vyhodnotili jednotlivé režimy z hľadiska sledovaných parametrov siete a dokázali sme tak lepšie vlastnosti našich útokov

Kľúčové slová: MANET, reaktívny smerovací protokol, ad hoc siete, útočné scenáre, bezpečnosť

Annotation

Slovak University of Technology Bratislava
FACULTY OF INFORMATICS AND INFORMATION TECHNOLOGIES
Degree Course: Software engineering

Author: Bc. Martin Ilavský
Master's thesis: Simulation of attack scenarios in mobile ad hoc networks
Supervisor: Ing. Jozef Filipek
2018, May

The work deals with security of routing protocols in MANET networks. The analysis deals with the characteristics of MANET networks, secured and unsecured versions of reactive routing protocols which security uses asymmetric and symmetric cryptography and attacks used in MANET networks distributed by RM OSI layers. There is also evaluation of individual parts of analysis that contains comparison of the analyzed routing protocols and their resistance against the analyzed attacks. Practical part of diploma thesis contains drafts of new attacks with focus on collaboration between attacking nodes. Concepts of security proposals against these attacks are also contained in thesis. Four new effective attacks, which should thoroughly check the security of routing protocols, were proposed. All of these attacks can be performed in multiple modes which are part of testing scenerio. Thesis describes behavior of these attacks together with diagrams and pictures. The efficiency of the proposed attacks were proven in the simulation environment on selected routing protocols. Each scenario was evaluated in terms of the selected network parameters and we proved better characteristics of our attacks.

Keywords: MANET, reactive routing protocol, ad hoc networks, attack scenarios, security

Obsah

Zoznam použitých skratiek

ABR	angl. „Associativity Based Routing“, reaktívny smerovací protokol v MANET sieťach
ARAN	angl. „A secure Routing protocol for Ad hoc Networks“, zabezpečený smerovací protokol v MANET sieťach využívajúci asymetrickú kryptografiu.
AODV	angl. „Ad Hoc On Demand Distance Vector“, reaktívny smerovací protokol v MANET sieťach
DARPA	angl. „Defense Advanced Research Projects Agency“, agentúra ministerstva obrany Spojených štátov amerických zaoberajúca sa vývojom armádnych technológií
DOS	angl. „Denial of Service“, útok zameraný na spôsobenie výpadku služieb, alebo nedostupnosti zariadenia, spôsobený jedným útočníkom
DSR	angl. „Dynamic Source Routing“, jednoduchý reaktívny smerovací protokol v MANET sieťach
DSDV	angl. „Destination Sequenced Distance Vector Routing“, proaktívny smerovací protokol v MANET sieťach
ERR	angl. „Error“, chybová správa v protokole ARAN
FSR	angl. „Fisheye State Routing“, proaktívny smerovací protokol v MANET sieťach
FTP	angl. „File Transfer Protocol“, štandardný sieťový protokol používaný pre prenos súborov.
GPS	angl. „Global Positioning System“, globálny systém určujúci polohu pomocou družíc
HMAC	typ autentizačného kódu správy využívajúci hešovacie funkcie
HTTP	angl. „Hypertext Transfer Protocol“, aplikačný protokol, ktorý je základom komunikácie pre World Wide Web.
IP	angl. „Internet Protocol“, hlavný komunikačný protokol v internete
MAC	angl. „Message Authentication Code“, krátka informácia na potvrdenie odosielateľa a „Media Access Control“, MAC adresa je unikátny identifikátor sieťového rozhrania

MANET	angl. „Mobile Ad Hoc Network“, mobilná ad hoc sieť vytváraná dynamicky bez predchádzajúcej sieťovej infraštruktúry
PAN	angl. „Personal Area Network“, počítačová sieť malého rozsahu určená na komunikáciu medzi osobnými zariadeniami
RDP	angl. „Route Discovery Packet“, správa hľadania cesty v protokole ARAN
REP	angl. „Reply“, správa odpovede na cestu v protokole ARAN
RFC	angl. „Request For Comments“, typ publikácie popisujúci technologický štandard
RM OSI	angl. „Reference Model of Open Systems Interconnection“, model zaraďujúci sieťové protokoly do siedmich vrstiev
RREP	angl. „Route Reply“, správa používaná ako odpoveď na hľadanie cesty v smerovacích protokoloch
RREQ	angl. „Route Request“, správa používaná pri hľadaní cesty v smerovacích protokoloch
RERR	angl. „Route Error“, správa používaná na notifikáciu o zmene topológie, alebo výskyte chyby v smerovacích protokoloch
SAKM	angl. „Simple Ad hoc Key Management“, systém na manažment kľúčov v ad hoc sieťach
SAODV	angl. „Secure Ad Hoc On Demand Distance Vector“, zabezpečený reaktívny smerovací protokol v MANET sieťach
SMTP	angl. „Simple Mail Transfer Protocol“, štandard pre prenos elektronickej pošty
SRP	angl. „Secure Routing Protocol“, zabezpečený reaktívny smerovací protokol pre ad hoc siete
SSL	angl. „Secure Socket Layer“, bezpečnostný štandard na vytvorenie šifrovaného spojenia medzi web serverom a prehliadačom
TLS	angl. „Transport Layer Security“, nasledovník SSL pre šifrovanú komunikáciu
VANET	angl. „Vehicle Ad Hoc Network“, špeciálny typ MANET sietí používaný vo vozidlách

Zoznam použitých obrázkov

Zoznam tabuliek

Úvod

Tento dokument vznikol na základe zadania diplomovej práce na Fakulte informatiky a informačných technológií Slovenskej Technickej Univerzity v Bratislave.

Vývoj súčasných sieťových technológií sa čoraz viac zameriava na bezdrôtové technológie. Bezdrôtové siete v budúcnosti budú pravdepodobne najbežnejšie používaným typom sietí, z dôvodu jednoduchosti pripojenia a možnosti slobodného pohybu v sieti. Medzi bezdrôtové siete zaraďujeme aj siete MANET založené na ad hoc princípe. Pre komunikáciu zariadení nie je nutná predchádzajúca infraštruktúra a na ich riadenie nie je potrebný ani centrálny sieťový uzol akým je smerovač.

Flexibilita týchto sietí prináša aj nové výzvy v bezpečnosti. Nové princípy smerovacích protokolov umožnili použitie nových aj starých útočných techník. Z tohto dôvodu vznikajú viaceré zabezpečené verzie smerovacích protokolov. Na otestovanie týchto riešení sa používajú vlastné vytvorené simulácie útokov, ktoré môžu byť na daný protokol prispôsobené a z tohto dôvodu nebude dostatočne otestovaný.

V tejto práci sme navrhli testovacie scenáre pre otestovanie navrhovaných riešení a implementovali sme ich v sieťovom simulátore OMNet++. Pomocou vytvorených scenárov boli otestované zanalyzované sieťové protokoly a porovnané ich zabezpečenia.

V prvej časti diplomovej práce sme zanalyzovali nezabezpečené a zabezpečené reaktívne smerovacie protokoly v MANET sieťach. Analýza obsahuje aj podrobný opis útokov v MANET sieťach rozdelených podľa vrstiev modelu RM OSI.

Na základe analýz sme navrhli nové útoky v MANET sieťach so zameraním na spoluprácu medzi útočnými uzlami. Tieto útoky sú súčasťou testovacích scenárov na otestovanie zabezpečenia reaktívnych smerovacích protokolov. Súčasťou scenárov sú aj bežné verzie útokov pre vzájomné porovnanie s nami navrhnutými útokmi. Pre naše navrhnuté útoky sme navrhli aj koncepty zabezpečení.

Vybrané smerovacie protokoly a navrhnuté scenáre boli implementované vo vybranom simulačnom prostredí. Výsledky simulácií boli vyhodnotené na základe dopredu definovaných sieťových parametrov. Na základe týchto simulácii sme tak dokázali efektívnosť a účinnosť našich navrhnutých útokov.

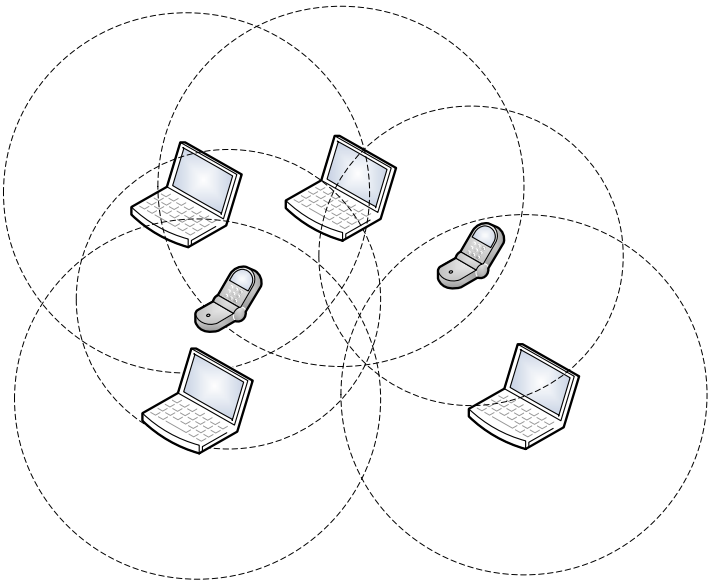
1 Analýza

Analýza práce sa zameriava na smerovacie protokoly a útoky v MANET sieťach. Analyzované sú vybrané zabezpečené a nezabezpečené smerovacie protokoly, voči ktorým budeme útočné scenáre navrhovať. Zameriame sa pritom na reaktívne smerovacie protokoly. Návrh útočných scenárov vychádza z analyzovaných útokov v MANET sieťach.

1.1 MANET siete

Bezdrôtové komunikačné technológie sa v súčasnosti vyvíjajú rýchlym tempom, kvôli stále zvyšujúcim sa požiadavkám z hľadiska rýchlosti, bezpečnosti a podobne. Jednou veľkou rozvíjajúcou sa skupinou týchto komunikačných technológií sú MANET siete.

Základnou charakteristikou MANET sietí je ich dynamické formovanie pomocou nezávislých mobilných uzlov prepojených pomocou rôznych bezdrôtových technológií bez predchádzajúcej sieťovej infraštruktúry a centralizovanej administrácie [1]. Nie je teda nutná žiadna infraštruktúra, alebo riadiaci sieťový prvok pre plnú funkčnosť siete. Každý uzol sa rozhoduje nezávisle na základe aktuálnej situácie. Uzly sa môžu v takejto sieti voľne pohybovať a združovať. Topológia siete sa musí vďaka tejto vlastnosti uzlov neustále a rýchlo dynamicky prispôsobovať. Takáto sieť môže byť úplne nezávislá, alebo môže byť pripojená k iným sieťam, ako napríklad k internetu. Príklad takejto siete s mobilnými zariadeniami je zobrazený na obr.1.1.



Obr. 1.1 Príklad topológie MANET siete

1.1.1 Základné charakteristiky

MANET siete majú niektoré vlastnosti spoločné so všetkými bezdrôtovými sieťami, vzhľadom na porovnanie s pevnými sieťami [1]. Konkrétne sú to:

- **Rušenie signálu a nízka spoľahlivosť:** Rôzne druhy bezdrôtových signálov sú viac či menej náchylné na rušenie. Infračervené signály môžu byť ovplyvnené svetlom a zdrojmi tepla, pohltené a tienené rôznymi materiálmi. Rádiové signály môžu byť rušené inými elektrickými zariadeniami. Rušenie môže nastať aj z dôvodu viacerých vysielajúcich uzlov.
- **Menšia šírka pásma:** V porovnaní s pevnými sieťami dosahujú bezdrôtové siete nižšie rýchlosti prenosu, dlhšie časy nadviazania spojenia, väčšie odozvy a vyšší jitter.
- **Vysoko premenlivé podmienky:** Častejšia strata dát. Sila signálu je závislá od vzdialenosti a prekážok. Pripojené zariadenia sa môžu svojvoľne pohybovať.
- **Limitované výpočtové a energetické zdroje:** Prenosné zariadenia sú menej výkonné, ako zariadenia pripojené v pevných sieťach. Napájanie mobilných zariadení je realizované pomocou batérií, ktoré dokážu zariadenia napájať len po obmedzenú dobu.
- **Znížená bezpečnosť:** Bezdrôtový signál môže byť kýmkoľvek odchytený a analyzovaný. Z tohto dôvodu je bezpečnosť v bezdrôtových sieťach nižšia a náročnejšia, ako je to v pevných sieťach.

Medzi špecifické charakteristiky MANET sietí [1][2] patria najmä:

- **Ad hoc:** Ad hoc sieť je dočasná sieť tvorená dynamicky množinou uzlov na základe potrieb.
- **Autonómne bez infraštruktúry:** Neexistuje žiadny centrálny uzol pre riadenie siete. Kontrola nad sieťou je rozdelená rovnocenne medzi všetky uzly siete. Uzly v sieti sú nezávislé a navzájom spolupracujú pri zabezpečovaní rôznych funkcionalít, ako sú smerovanie a bezpečnosť.
- **Vysoko dynamická topológia:** Zariadenia v MANET sieťach sa v porovnaní s bežnými bezdrôtovými sieťami pohybujú oveľa častejšie. Kvôli absencii centrálneho riadiaceho prvku sa tak topológia MANET sietí mení veľmi rýchlo a často.

- **Viacskokové smerovanie:** V klasických bezdrôtových sieťach sa dáta posielajú vždy na centrálny smerovač, ktorý ich odošle cieľovému zariadeniu. V MANET sieťach sa uzly správajú ako samostatné smerovače, ktoré preposielajú dáta medzi požadovanými koncovými bodmi. Výpadok jedného uzla tak neovplyvní celú sieť. Doručovanie správy je štandardne vykonávané pomocou viacerých skokov.

Celkové zhrnutie výhod a nevýhod v MANET sieťach sa nachádza v tab. 1.1.

Tab. 1.1 Porovnanie výhod a nevýhod MANET sietí [2]

Výhody	**Nevýhody**
Samokonfigurovateľné	Limitovaná šírka pásma
Nezávislé od infraštruktúry	Časté zmeny topológie
Vysoko škálovateľné	Vyššie režijné náklady
Vytvorenie kdekoľvek a kedykoľvek	Strata paketov
Robustné	Závislé od batérií
	Slabšia bezpečnosť

1.1.2 Aplikácia MANET sietí

S nárastom počtu prenosných zariadení spolu s technologickým pokrokom v bezdrôtovej komunikácií sa MANET siete začínajú čoraz viac používať vo verejnom aj privátnom sektore. Ich kľúčové vlastnosti sú v súčasnej dobe veľmi žiadané a vďaka decentralizácií sú takéto siete pružnejšie a robustnejšie [3]. Ich využitie je možné v nasledovných oblastiach:

- **Vojenský sektor:** Vojenská technika využíva najnovšie technológie na získanie výhody voči nepriateľom. Prvé použitie ad hoc sietí bolo práve vo vojenskom sektore, kde umožnili lepšiu komunikáciu na bojisku.
- **Komerčný sektor:** Ad hoc siete sú vhodné na použitie v núdzových a záchranných operáciách. Automatické vytvorenie komunikačných spojení uľahčí prácu záchranným zložkám. Medzi ďalšie komerčné využitie môžeme zaradiť napríklad komunikáciu medzi autonómnymi vozidlami. Takýto druh MANET sietí sa nazýva VANET. Tieto siete sú veľmi perspektívne ohľadom autonómneho riadenia.
- **Kolaboratívna práca:** Pre určité pracovné pozície je spolupráca a komunikácia dôležitejšia vo vonkajšom prostredí ako vnútri v kanceláriách. Ľudia, ktorí pracujú

v takýchto prostrediach môžu vďaka MANET spolupracovať a vymieňať si informácie.

- **Dátové siete:** Komerčné MANET aplikácie zahŕňajú všadeprítomné počítanie. Umožnením posielania dát ďalším zariadeniam môžu byť takéto siete rozšírené ďaleko za rozsah bežných pevných sietí, čím sa stanú viac dostupnejšími a ľahšími na použitie.
- **Senzorové siete:** Tieto siete sú zložené z veľkého počtu malých senzorov, slúžiacich na zaznamenávanie rôznych vlastností prostredia, ako sú teplota, tlak, znečistenie a pod. Schopnosti jednotlivých senzorov sú veľmi limitované a pri posielaní svojich dát na spracovanie sa musia spoliehať na ostatné senzory. Vďaka ad hoc sieťam môžu byť tieto spojenia vytvárané dynamicky bez potreby centrálneho zariadenia na manažovanie jednotlivých spojení.
- **PAN a bluetooth:** Osobné siete sú často používané len jednou osobou a majú veľmi krátky dosah. Pomocou technológií ako bluetooth je možné jednoduché vytvorenie komunikácie medzi mobilnými zariadeniami, ako sú prenosné počítače a mobilné telefóny.

1.1.3 Bezpečnosť

Používaním spoločného média sú uzly v MANET sieťach vystavené veľkému bezpečnostnému riziku. Keďže tieto siete nemajú ani pevnú infraštruktúru, tak sa bezpečnostné riešenia sústreďujú iba na zabezpečenie individuálnych uzlov z dôvodu nemožnosti vytvorenia centralizovanej kontroly [3]. Bezpečnosť tak musí byť zabezpečená pomocou spolupráce uzlov. Medzi kľúčové požiadavky na bezpečnosť môžeme zaradiť:

- **Dôveryhodnosť:** prevencia proti odpočúvaniu
- **Kontrola prístupu:** ochrana prístupu k bezdrôtovej infraštruktúre
- **Integrita dát:** prevencia manipulácie sieťovej premávky
- **Ochrana pred DoS útokmi:** prevencia pred odmietnutím služby

Podrobná analýza vybraných útokov sa nachádza v časti 1.3.

1.2 Smerovacie protokoly

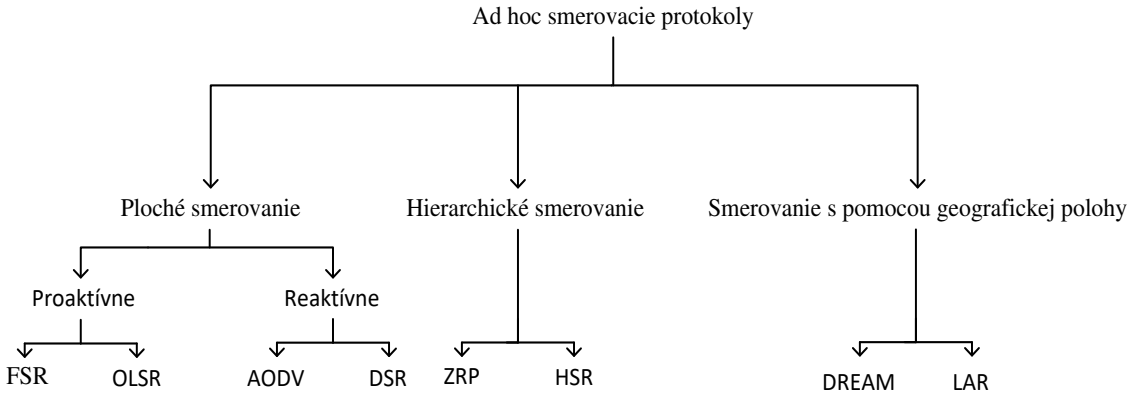
Smerovacie protokoly v MANET sieťach musia byť rýchlo adaptovateľné na časté a nepredvídateľné zmeny topológie a zároveň šetrné k výpočtovým zdrojom. Vo

všeobecnosti používajú tieto protokoly „distance vector“ (DV) alebo „link state“ (LS) algoritmy.

Vektor v DV algoritmoch obsahujúci cenu linky (počet skokov) a cestu (nasledujúci skok) je ukladaný a vymieňaný medzi uzlami. Protokoly založené na týchto algoritmoch majú pomalšiu konvergenciu a tendenciu vytvárať slučky v mobilných prostrediach.

Protokoly využívajúce LS algoritmy si uchovávajú informácie o celej sieti v každom uzle periodickým zaplavovaním siete s informáciami o svojich susedoch. Toto zaplavovanie siete kontrolnými správami je kvôli častým zmenám topológie veľmi časté. V sieťach s veľkým počtom uzlov tak môžu spotrebovať veľkú časť dostupnej šírky pásma a spomaliť tak bežnú komunikáciu medzi uzlami. Protokoly s týmto prístupom tak môžu mať problémy so škálovateľnosťou.

Z hľadiska štruktúry siete môžeme smerovacie protokoly rozdeliť na tri skupiny, a to ploché smerovanie, hierarchické smerovanie a smerovanie s pomocou geografickej polohy [4]. Rozdelenie aj s príkladmi protokolov je možné vidieť na obr 1.2.



Obr. 1.2 Rozdelenie ad hoc smerovacích protokolov

V prístupoch plochého smerovania sú všetky uzly, ktoré sa zúčastňujú na smerovaní, rovnocenné. Zaraďujeme sem proaktívne a reaktívne smerovacie protokoly.

Uzly v reaktívnych smerovacích protokoloch získavajú informácie o ceste k cieľu na požiadanie. Nie je tak nutné periodické zaplavovanie siete. Zisťovanie cesty do cieľa pomocou zaplavenia je vykonané iba vtedy, keď je to potrebné. Hlavnou výhodou reaktívnych protokolov je, že nepotrebujú veľa pamäťových prostriedkov. Na druhej strane ale generujú veľké množstvo správ pri hľadaní ciest počas zmien v topológii, ktoré sa v MANET sieťach vyskytujú veľmi často [5]. Príkladmi reaktívnych protokolov sú napríklad DSR, AODV a ABR.

V proaktívnych protokoloch si každý uzol v sieti uchováva v pamäti jednu, alebo viacero smerovacích tabuliek, ktoré sú pravidelne aktualizované. Na zistenie zmien v topológií posiela každý uzol správu všetkým zariadeniam. Na udržiavanie aktuálnych informácií tak tieto správy zaberú časť sieťových zdrojov [5]. Medzi proaktívne protokoly zaraďujeme napríklad DSDV a FSR.

Princíp hierarchického smerovania je založený na zhlukovaní uzlov do skupín a na priradení rôznych funkcionalít uzlom v skupine a mimo skupiny. Najpoužívanejším spôsobom zoskupovania uzlov je na základe ich vzájomnej polohy. Vytvorením takýchto skupín je možné znížiť počet potrebných správ na výmenu smerovacích informácií. Taktiež je možné na smerovanie v samotných skupinách a medzi skupinami využiť princípy proaktívnych a reaktívnych smerovacích protokolov [4]. Takéto kombinácie označujeme ako hybridné smerovacie protokoly. Príkladom takéhoto protokolu je napríklad ZRP.

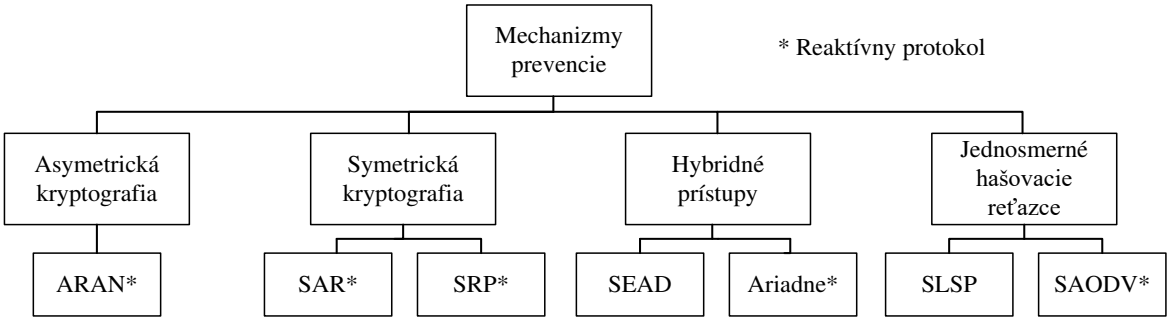
Pokrok v technológii GPS umožnil určiť polohu zariadenia s vysokou presnosťou. Táto technológia poskytuje aj spoločné hodiny. Zatiaľ čo informácie o polohe môžu byť použité na smerovanie, tak spoločné hodiny môžu slúžiť na globálnu synchronizáciu. Výskumy ukázali, že znalosť polohy výrazne zlepšuje výkon smerovania v ad hoc sieťach. Medzi protokoly využívajúce túto metódu zaraďujeme DREAM a LAR.

V našej práci sa zameriame na spomínané reaktívne smerovacie protokoly.

Pri zabezpečovaní smerovacích protokolov je možné využiť viaceré metódy. Mechanizmus prevencie zabraňuje škodlivým uzlom v ich konaní. Tento prístup využíva šifrovanie na zabezpečenie dôveryhodnosti, integrity a nepopierateľnosti smerovacích informácií.

Takisto je možné aplikovať aj detekčné mechanizmy škodlivým uzlov, ktoré identifikujú škodlivé uzly na základe ich aktivity a následne podniknú opatrenia na zabezpečenie siete.

Najčastejšie využívajú zabezpečené protokoly mechanizmus prevencie. Metódy využívané v tomto mechanizme môžeme rozdeliť na tie, ktoré využívajú asymetrickú kryptografiu, symetrickú kryptografiu, jednosmerné hašovacie reťazce, alebo hybridné prístupy. Rozdelenie zabezpečených protokolov podľa týchto metód môžeme vidieť na obr. 1.3.



Obr. 1.3 Zabezpečené smerovacie protokoly

1.2.1 DSR

DSR (Dynamic Source Routing protocol) je jednoduchý smerovací protokol navrhnutý ešte v 90.tych rokoch pre bezdrôtové ad hoc siete. Založený je na dvoch mechanizmoch, ktoré spolu umožňujú nájdenie a udržiavanie ciest [6].

Nájdenie cesty je mechanizmus, ktorý zabezpečuje nájdenie a uloženie cesty v prípade, keď chce uzol A komunikovať s uzlom B a ešte nepozná cestu k tomuto uzlu.

Udržiavanie cesty je mechanizmus zabezpečujúci aby uzol A detegoval, že zmena topológie spôsobila nemožnosť použitia naučenej cesty k uzlu B. Na základe tejto detekcie môže uzol A použiť iné naučené cesty (ak také má), alebo opäť vyvolať mechanizmus hľadania cesty.

Tieto mechanizmy v protokole DSR vytvárajú jednosmerné cesty a teda cesta pre komunikáciu z uzla A do uzla B nemusí byť rovnaká, ako pri komunikácii z uzla B do uzla A.

DSR nepoužíva žiadne pravidelné odosielanie paketov na všetkých sieťových vrstvách [6].

1.2.1.1 Smerovacie tabuľky

DSR nepoužíva klasické smerovacie tabuľky, ale namiesto nich má vyrovnávaciu pamäť s cestami (Route Cache). V tejto pamäti môže byť uložených viacero ciest, ktoré sú použité na smerovanie. Cesta vo vyrovnávacej tabuľke má formát skok – skok – skok – ... – cieľ [7].

Majme uzol D s nasledovnou vyrovnávacou pamäťou: C-F-I-K-L a A-I-J-E. Pre smerovanie do daného uzla nemusí byť uzol v ceste uvedený ako posledný. Do uzla K je teda možné smerovať pomocou prvej cesty. Môže nastať aj situácia kedy je do požadovaného uzla dostupných viacero ciest. V tomto prípade rozhoduje kratšia cesta a v našom prípade by sa do uzla I smerovalo pomocou druhej cesty.

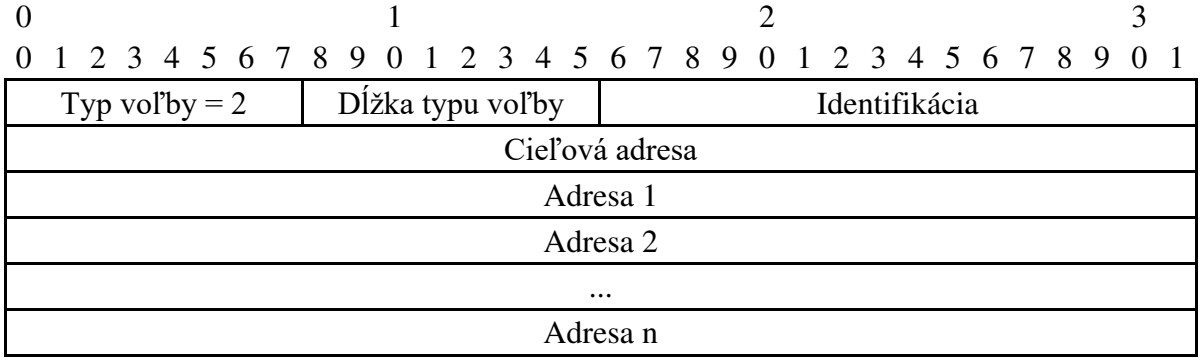
Vyrovnávacia pamäť si ukladá tieto cesty na základe prijatých správ. Každá správa obsahuje zoznam adries, ktorými správa prešla v prípade RREQ (Route Request) správy. alebo ktorými prešla a ešte má prejsť (RREP a dáta). RFC odporúča ukladanie iba dopredných ciest (ktorými má správa ešte prejsť), ale nevylučuje ukladanie aj reverzných ciest (ktorými už správa prešla) [7].

Rušenie ciest je realizované na základe zistení o zmene topológie napríklad pomocou správy RERR (Route Error), alebo po zlyhaní posielania správy.

1.2.1.2 Hľadanie cesty

Pri hľadaní cesty sa v protokole DSR používajú dve správy, a to požiadavka na cestu RREQ (Route Request) a odpoveď na cestu RREP (Route Reply). Formát týchto správ [7] je možné vidieť v tabuľkách 1.2 a 1.3. Hlavička IP obsahujúca zdrojovú, cieľovú IP adresu a hodnotu TTL, nie je zobrazená.

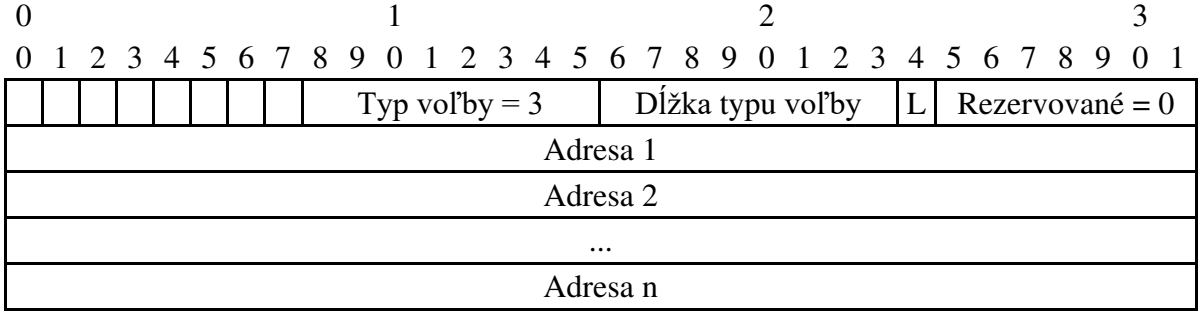
Tab. 1.2 Formát rámca požiadavky na cestu



- **Typ voľby:** typ DSR rámca, 1 - požiadavka na cestu, 2 - odpoveď na cestu, 3 - chyba cesty.
- **Dĺžka typu voľby:** dĺžka rámca v oktetoch, hodnota je (4 * n) + 6, kde n je počet adries v hlavičke.
- **Identifikácia:** obsahuje jedinečnú hodnotu vygenerovanú počiatočným uzlom. Pre každú požiadavku na cestu vygeneruje uzol jedinečnú identifikáciu. Táto hodnota slúži na prevenciu slučiek. Ak uzol príjme viacero správ od zdrojového uzla s rovnakou identifikáciou, tak spracuje iba prvú prijatú.
- **Cieľová adresa:** adresa zariadenia ku ktorému sa hľadá cesta
- **Zoznam adries:** zoznam zariadení, ktorými bol daný rámec preposlaný. Uzol, ktorý vytvoril prvú správu nesmie byť uvedený ako adresa 1. Jeho adresa sa nachádza v IP hlavičke. Počet adries v rámci je definovaný v poli dĺžka typu voľby.

Každý uzol preposielajúci danú správu pridá pred jej odoslaním do tohto zoznamu vlastnú IP adresu. IP adresa uzla ktorý hľadá cestu sa nachádza v IP hlavičke a pri preposielaní inými uzlami sa táto adresa nemení.

Tab. 1.3 Formát rámca odpovede na cestu



- **L:** príznak indikuje, že posledný skok medzi adresami n-1 a n vedie cez externú sieť vzhľadom na DSR sieť. Pri hľadaní cesty sú preferované tie, ktoré tento príznak neobsahujú.

Pri hľadaní cesty uzol pošle všetkým uzlom v dosahu správu RREQ. Táto správa obsahuje zdrojovú a cieľovú adresu uzlov spolu s jedinečným identifikátorom RREQ správy a prázdnym zoznamom adries. Keď uzol príjme takúto správu, porovná identifikátor správy s predchádzajúcimi prijatými. Ak už prijal inú RREQ správu s rovnakým identifikátorom, alebo sa jeho adresa nachádza v zozname adries (v RREQ hlavičke), tak správu zahodí [6]. Následne vykoná nasledovné rozhodovanie:

- Ak pozná cestu do cieľovej adresy, alebo je daný uzol cieľovým uzlom RREQ správy, tak vygeneruje správu RREP spolu s kópiou prijatého zoznamu adries, ktorú odošle zdrojovému uzlu naspäť v RREQ správe.
- Ak cestu nepozná, tak zoznam adries rozšíri o svoju adresu a RREQ správu pošle ďalej všetkým uzlom v dosahu.

Pomocou jedinečného identifikátora a zoznamu adries sa zabezpečí prevencia vytvárania slučiek.

Správu RREP cieľový uzol už nemusí posielať všetkým zariadeniam, keďže vďaka zoznamu adries v prijatej RREQ správe vie, akou cestou má odpoveď poslať. Druhou, nie však odporúčanou možnosťou (v prípade ak cieľový uzol nemá vo vyrovnávacej pamäti uloženú cestu do zdrojového uzla) je vygenerovanie RREQ správy na zistenie cesty k zdroju. Týmto spôsobom by ale vznikla nekonečná slučka zisťovania ciest a preto sa

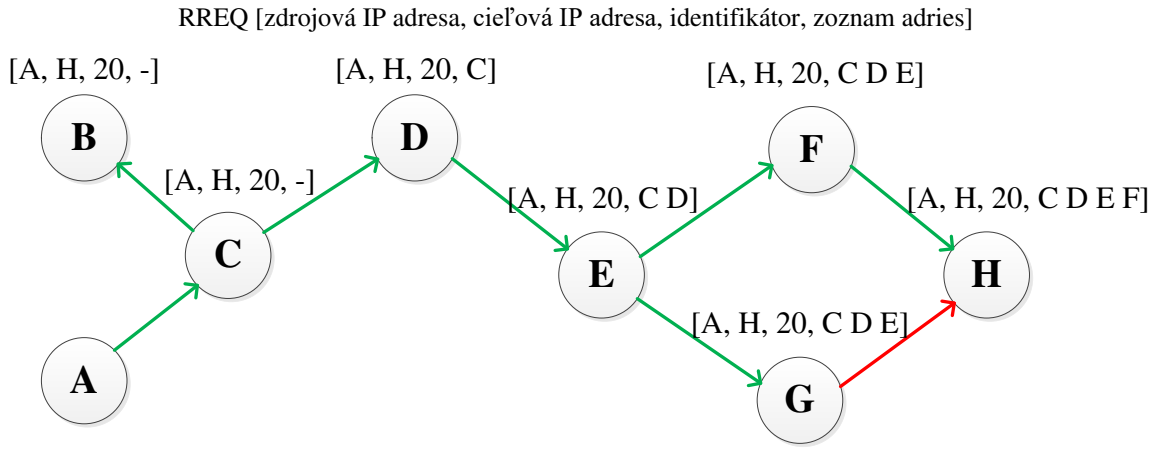
správa RREP pošle spolu so správou RREQ. Tento princíp ale nie je odporúčaný z dôvodu zbytočného vyťažovania siete. Pomocou prvej metódy RREP sa tak vytvoria obojsmerné cesty a pomocou druhej metódy jednosmerné cesty [7].

Rámce s dátami obsahujú vždy hlavičku so zoznamom adries, cez ktoré sa dáta dostanú od zdroja do cieľa.

Pri generovaní RREP správy musí mať zoznam adries nasledovný formát. Adresa 1 je prvý skok na ceste a posledná adresa n musí byť adresou cieľa. Ostatné adresy reprezentujú ďalšie skoky medzi týmito počiatočnými uzlami [7].

1.2.1.3 Príklad hľadania cesty

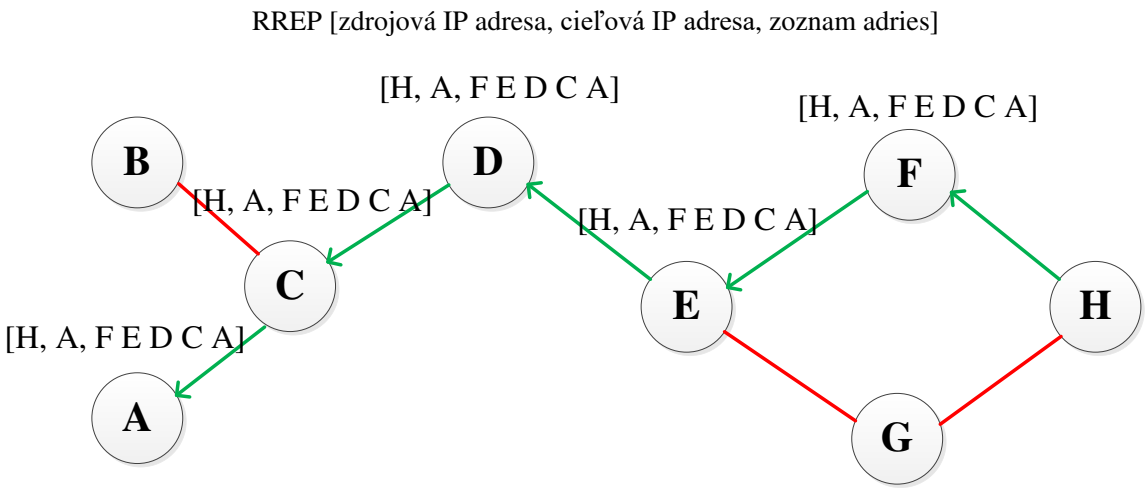
Princíp fungovania protokolu DSR môžeme vidieť na obrázkoch 1.4 a 1.5. Kruhy reprezentujú uzly a spojnice reprezentujú spojenia v dosahu uzla. Uzol môže komunikovať len s tými uzlami, ktoré má v dosahu. V našom príklade chce uzol A komunikovať s uzlom H. Žiadny z uzlov nemá uložené vo vyrovnávacej pamäti žiadne cesty. Na obrázku 3 je nad každým uzlom zobrazená prijatá RREQ správa vo formáte [zdrojová IP adresa, cieľová IP adresa, identifikátor, zoznam adries]. Obrázok 1.4 zobrazuje priebeh posielania správy RREQ až do cieľového uzla. Predpokladáme, že uzol H príjme RREQ správu ako prvú od uzla F. Správy zobrazené červenými šípkami sú zahodené na základe identifikátora. Z dôvodu prehľadnosti nie sú zobrazené spätné posielania. Uzol totiž prepošle prijatú RREQ správu všetkým uzlom v dosahu, ku ktorým patrí aj uzol od ktorého správu dostal. Tieto správy sú zahodené na základe identifikátora.



Obr. 1.4 Hľadanie cesty v protokole DSR

Na obr. 1.5 môžeme vidieť preposielanie odpovede RREP od uzla H. Nad každým uzlom je zobrazená prijatá RREP správa vo formáte [zdrojová IP adresa, cieľová IP

adrresa, zoznam adries]. Obsah vyrovnávacích pamätí uzlov je zobrazený v tab. 1.4 pri uvažovaní ukladania dopredných ciest.



Obr. 1.5 Odpoveď na hľadanie cesty v protokole DSR

Tab. 1.4 Obsah vyrovnávacích pamätí uzlov po doručení správy RREP

Uzol	Cesta vo vyrovnávacej pamäti
A	C - D - E - F
B	
C	A
D	C - A
E	D -C - A
F	E - D -C - A
G	
H	F - E - D -C - A

1.2.1.4 Zmeny topológie

Pri posielaní paketov po ceste sú jednotlivé vysielajúce uzly zodpovedné potvrdiť jeho prijatie. Pokiaľ odosielajúci uzol nedostane potvrdenie o prijatí správy, tak odosielanie zopakuje. Maximálny počet opätovných odosielaní je dopredu stanovený. Ak ani po týchto viacerých opakovaniach nedostane uzol potvrdzujúcu správu o prijatí, tak vygeneruje správu RERR, ktorú odošle zdroju pôvodnej správy informujúc ho tak o nemožnosti jej doručenia [6]. Zdroj na základe tejto správy vymaže príslušnú cestu vo vyrovnávacej pamäti. Ak nemá v tejto pamäti ďalšie cesty, cez ktoré by mohol správu doručiť do cieľového uzla, inicializuje nový proces hľadania cesty do tohto cieľa.

1.2.2 AODV

AODV (Ad-hoc On Demand Distance Vector) patrí medzi najznámejšie reaktívne smerovacie protokoly v MANET sieťach. Oproti proaktívnym smerovacím protokolom

vyžaduje menšiu dostupnú šírku pásma, pretože správy neposiela periodicky všetkým uzlom. V sieťach s veľkým počtom uzlov je AODV veľmi efektívny vďaka dynamickému vytváraniu smerovacích tabuliek. Prevencia proti vytvoreniu slučiek je zabezpečená pomocou sekvenčných čísiel, ktoré slúžia ako časové odtlačky a umožňujú tak uzlom rozlíšiť medzi novšími a staršími informáciami [8]. Protokol môže nájsť viacero ciest medzi zdrojom a cieľom, ale použije sa iba tá najlepšia, ktorá nemusí byť nutne najkratšia. Odpadá tak zložitý manažment viacerých ciest.

1.2.2.1 Smerovacie tabuľky

Každý uzol si udržiava vlastnú smerovaciu tabuľku, ktorú si aktualizuje na základe prijatých správ. Záznam v smerovacej tabuľke má nasledovný formát [9]:

Tab. 1.5 Formát záznamu v smerovacej tabuľke protokolu AODV

Cieľ	Nasledujúci uzol	Počet skokov	Sekvenčné číslo cieľa	Životnosť

Pričom cieľ je IP adresa cieľa. Nasledujúci uzol je IP adresa uzla, cez ktorý sa správa pošle ďalej. Počet skokov udáva počet uzlov na ceste k cieľu a sekvenčné číslo je posledné prijaté sekvenčné číslo od cieľa, ktoré sa v správe použije. Toto číslo je aktualizované po každej prijatej správe od príslušného cieľového uzla. Pomocou expirácie záznamov v smerovacej tabuľke sa predchádza udržiavaniu starých ciest a nie je nutné ani dodatočné spravovanie cesty.

1.2.2.2 Hľadanie cesty

Pri hľadaní cesty sa v protokole AODV používajú dve správy, a to RREQ (Route request) a RREP (Route Reply) [9]. Správa RERR (Route Error) slúži na informovanie o chybách. Formát správ RREQ a RREP je možné vidieť v tabuľkách 1.6 a 1.7.

Tab. 1.6 Formát správy RREQ

0 1 2 3 4 5 6 7	8	9	10 11 12	13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
Typ = 1	J	R	G D U	Rezervované = 0	Počet skokov
RREQ ID				
IP adresa cieľa				
Sekvenčné číslo cieľa				
IP adresa zdroja				
Sekvenčné číslo zdroja				

- **Typ:** typ AODV správy, 1 - požiadavka na cestu, 2 – odpoveď na cestu, 3 – chyba cesty.
- **Príznaky:** J – „join“, R –„repair“, G –„gratuitous“, D – „destination only“, U – neznáme sekvenčné číslo, príznaky J a R sú vyhradené pre multicast, ostatné príznaky sa používajú na kontrolné funkcie.
- **Počet skokov:** počet uzlov, ktorými už správa prešla.
- **RREQ ID:** spolu s IP adresou zdroja tvorí jedinečný identifikátor požiadavky na cestu. Inkrementovaný je vždy pri generovaní novej RREQ správy
- **IP adresa cieľa:** IP adresa uzla, ku ktorému sa hľadá cesta.
- **Sekvenčné číslo cieľa:** posledné prijaté sekvenčné číslo od cieľa v minulosti. V prípade, že nie je známe, tak sa nastaví príznak U (neznáme sekvenčné číslo cieľa).
- **IP adresa zdroja:** IP adresa uzla, ktorý hľadá cestu k cieľu.
- **Sekvenčné číslo zdroja:** aktuálne sekvenčné číslo, ktoré má byť použité v odpovedi na správu.

Tab. 1.7 Formát správy RREP

0 1 2 3 4 5 6 7	8	9	10 11 12 13 14 15 16 17 18	19 20 21 22 23 24 25 26	27 28 29 30 31
Typ = 2	R	A	Rezervované = 0	Veľkosť prefixu	Počet skokov
IP adresa cieľa					
Sekvenčné číslo cieľa					
IP adresa zdroja					
Životnosť					

- **Príznaky:** R - „repair“, A - „acknowledge required“.

- **Veľkosť prefixu:** ak má nenulovú hodnotu, tak toto číslo určuje prefix podsiete v ktorej je daný uzol dostupný.
- **Životnosť:** čas v milisekundách, pomocou ktorého môžu ostatné uzly pred jeho uplynutím považovať cestu za platnú.

Keď chce uzol komunikovať s iným uzlom, tak odošle RREQ paket všetkým susedným uzlom (broadcast). Keď uzol príjme takúto správu, tak skontroluje, či už takúto správu neprijal pomocou polí RREQ ID a zdrojovej IP adresy. Ak áno, tak správu zahodí a predchádza tak vytvoreniu slučky. Ak nie, tak navýši hodnotu v poli počet skokov o 1 a aktualizuje si záznam o cieli v smerovacej tabuľke v nasledovných prípadoch [8]:

1. Sekvenčné číslo cieľa v RREQ pakete je väčšie ako to, ktoré je uložené v smerovacej tabuľke.
2. Sekvenčné čísla sa rovnajú, ale počet skokov po navýšení je menší ako ten, ktorý je uložený.
3. Sekvenčné číslo je neznáme (príznak U).

Následne je RREQ paket porovnaný so záznamom v smerovacej tabuľke o zdroji. Ak záznam neexistoval, tak sa vytvorí a ak existoval, tak sa aktualizuje. Nastaví sa aktuálne sekvenčné číslo, ďalším uzlom pre preposielanie správ sa stane ten, od ktorého bol paket prijatý a hodnota počtu skokov sa nastaví na rovnakú, ako bola prijatá v pakete.

Po tomto procese uzol odošle správu pomocou broadcastu ďalej a celý proces sa zopakuje, až kým správa nepríde na cieľový uzol, alebo ju neprijme uzol, ktorý má v smerovacej tabuľke aktívnu cestu do cieľa so sekvenčným číslom väčším, alebo rovným ako v prijatej RREQ správe [8]. Po odpovedaní takýmto uzlom ale môžu nastať rôzne nepríjemnosti. Uzol, ktorý odpovie, môže byť jediným uzlom, cez ktorý sa dá do cieľa dostať. Tým pádom ale nevznikne obojsmerná cesta a cieľový uzol bude musieť hľadať cestu pre odpoveď. Ak aj tento uzol nebude jediným, tak sa síce obojsmerná cesta vytvorí, ale nemusí byť rovnaká smerom tam aj späť a navyše cieľ navýši svoje sekvenčné číslo a vygeneruje správu RREP. Tým pádom príjme žiadateľ 2 RREP správy a podľa špecifikácie si uloží len prvú z nich (čo pravdepodobne nebude správa pochádzajúca od cieľa) a tým pádom bude ignorovať prichádzajúcu RREP správu od cieľa s vyšším sekvenčným číslom.

Uzol ktorý je cieľovým uzlom RREQ správy odošle zdrojovému uzlu RREP správu už pomocou unicastu, pretože z prijatej RREQ správy zistí, cez ktorý uzol má správu odoslať.

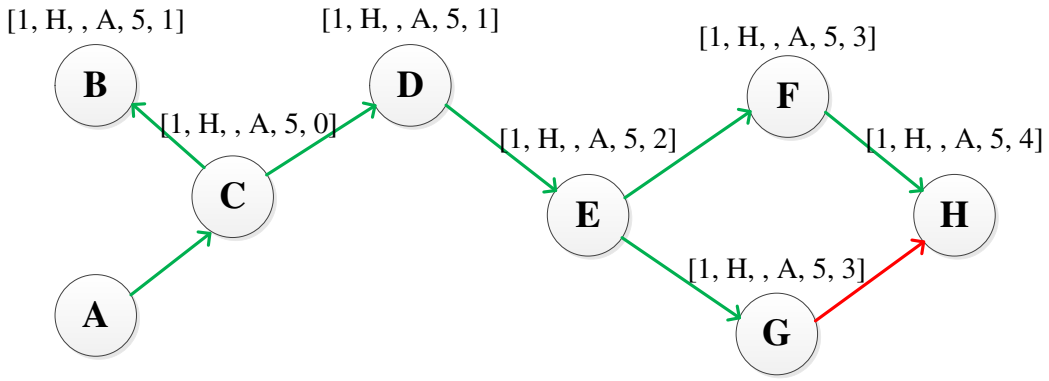
Sekvenčné číslo cieľa nastaví podľa hodnoty v prijatej RREQ správe. Táto správa sa posiela naspäť podobným spôsobom. Uzly cez ktoré správa prechádza si aktualizujú svoje smerovacie tabuľky, navyšujú hodnotu poľa počet skokov a preposielajú správu ďalej. Po dosiahnutí zdroja pôvodnej RREQ správy je vytvorená aktívna cesta po ktorej si môžu tieto dva uzly vymieňať dáta. Ak uzol príjme viacero RREP správ, použije iba prvú z nich a týmto spôsobom si uloží tú najlepšiu (nie nutne najkratšiu) cestu [9].

Správy RREP slúžia aj na ohlasovanie uzlov. Ak je uzol súčasťou nejakej aktívnej cesty, tak by mal v definovaných intervaloch odosielať správu RREP všetkým svojim susedom ako HELLO správu.

1.2.2.3 Príklad hľadania cesty

Princíp fungovania protokolu AODV môžeme vidieť na obrázkoch 1.6 a 1.7. Predpokladajme, že na začiatku sú smerovacie tabuľky prázdne a uzol A chce zistiť cestu k uzlu H. Na obr. 1.6 je nad každým uzlom prijatá RREQ správa vo formáte [RREQ ID, IP adresa cieľa, sekvenčné číslo cieľa, IP adresa zdroja , sekvenčné číslo zdroja, počet skokov]. Obr. 1.6 zobrazuje stav po doručení správy RREQ až do cieľového uzla. Predpokladáme, že uzol H príjme RREQ správu ako prvú od uzla F. Tým pádom prijatú správu z uzla G na základe sekvenčného čísla zahodí. Na základe sekvenčného čísla sú zahodené aj správy posielané naspäť ako to bolo aj pri protokole DSR.

RREQ: [RREQ ID, IP adresa cieľa, sekvenčné číslo cieľa, IP adresa zdroja, sekvenčné číslo zdroja, počet skokov]



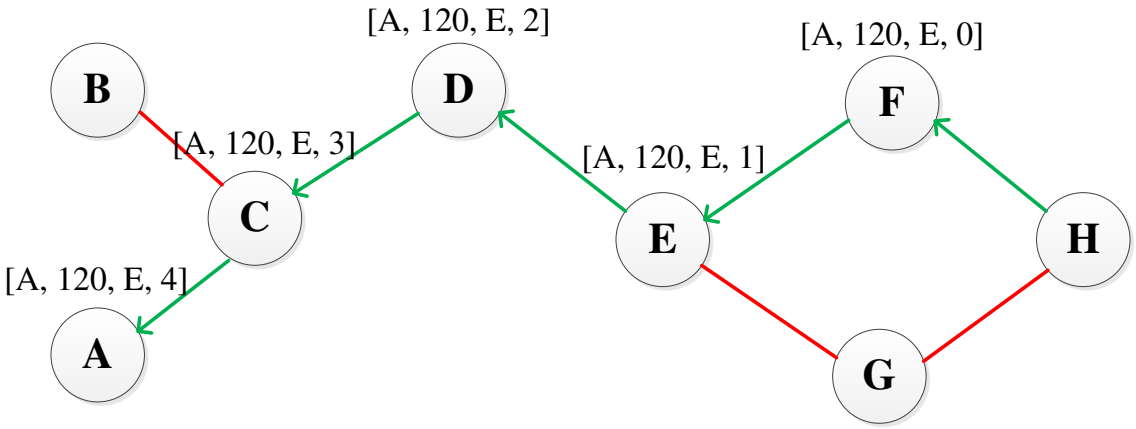
Obr. 1.6 Posielanie správy RREQ v protokole AODV

Tab. 1.8 Obsah smerovacích tabuliek uzlov po prijatí správy RREQ

Uzol	Smerovacia tabuľka			
	Cieľová IP adresa	**Nasledujúci uzol**	**Počet skokov**	**Sekvenčné číslo cieľa**
A	-	-	-	-
B	A	C	2	5
C	A	A	1	5
D	A	C	2	5
E	A	D	3	5
F	A	E	4	5
G	A	E	4	5
H	A	F	5	5

Na obr. 1.7 môžeme vidieť preposielanie odpovede RREP od uzla E. Nad každým uzlom je zobrazená prijatá RREP správa vo formáte [IP adresa cieľa, sekvenčné číslo cieľa, IP adresa zdroja, počet skokov]. Takisto je v nasledovnej tabuľke zobrazený obsah smerovacích tabuliek jednotlivých uzlov. Uzly B a G správu RREP nedostanú, pretože sa posiela unicastom.

RREP: [IP adresa cieľa, sekvenčné číslo cieľa, IP adresa zdroja, počet skokov]



Obr. 1.7 Posielanie správy RREP v protokole AODV

Tab. 1.9 Obsah smerovacích tabuliek uzlov po prijatí správy RREP

Uzol	Smerovacia tabuľka			
	Cieľová IP adresa	**Nasledujúci uzol**	**Počet skokov**	**Sekvenčné číslo cieľa**
A	H	C	5	120
B	A	C	2	5
C	A	A	1	5
	H	D	4	120
D	A	C	2	5
	H	E	3	120
E	A	D	3	5
	H	F	2	120
F	A	E	4	5
	H	H	1	120
G	A	E	4	5
H	A	F	5	5

1.2.2.4 Zmeny topológie

Uzol, ktorý prestane prijímať pravidelné RREP správy, môže okamžite vygenerovať správu RERR [9]. Uzol môže taktiež vygenerovať správu RERR signalizujúc chybu v nasledovných situáciách:

1. Ak počas vysielania dát zistí, že uzol cez ktorý dáta posiela je nedostupný.
2. Ak príjme paket, ktorého cieľovú adresu nemá v smerovacej tabuľke.
3. Ak príjme RERR správu od suseda pre jednu, alebo viacero aktívnych ciest.

Takéto situácie vznikajú z dôvodov pohybu uzlov v MANET sieti a vďaka správe RERR môže protokol prispôsobiť cesty k uzlom. Správa RERR obsahuje zoznam uzlov ovplyvnených pádom linky. Keď uzol na vytvorenej ceste príjme túto správu označí si konkrétnu cestu v smerovacej tabuľke ako neplatnú nastavením nekonečnej vzdialenosti. Správa je následne preposlaná ďalej, až k zdroju cesty, ktorý inicializuje hľadanie novej cesty [8].

Mechanizmus hľadania cesty pomocou zaplavenia siete môže mierne zahltiť sieť. Čím menej budú nastávať zmeny v topológii, tým menej bude sieť zaplavovaná. Na druhej strane poskytuje tento spôsob potencionálne vyššiu spoľahlivosť, pretože pakety môžu byť do cieľa doručené viacerými cestami.

1.2.3 SAODV

Pri zabezpečovaní bezpečnosti MANET sietí je možné zabezpečovať samotnú výmenu dát, alebo smerovacích informácií, na ktoré sa protokol SAODV zameriava. Tento protokol je rozšírením protokolu AODV, popísaného v časti 1.2.2. Na autentizáciu nemenných polí používa digitálne podpisy a na zabezpečenie jediného premenlivého poľa, ktorým je počet skokov, používa reťazce hašov. Zabezpečená by tak mala byť integrita a autentizácia. Dôveryhodnosť zabezpečená nie je z dôvodu povahy ad hoc sietí, do ktorých sa môže pripojiť ktokoľvek a kedykoľvek.

Podľa návrhu autora v IETF [10] existujú dve možnosti úpravy pôvodného AODV protokolu. Prvá je založená na využití jedného podpisu. Druhá možnosť pridáva druhý podpis podpisujúci reverznú cestu k uzlu, ktorý správu vytvoril. Pri popise protokolu sa zameriame najmä na prvý variant s jedným podpisom.

1.2.3.1 Rozšírenie AODV správ

V tab. 1.10 môžeme vidieť, ako sú v protokole SAODV rozšírené správy RREQ a RREP. Rozšírenie je pre obe správy rovnaké, líšia sa iba hodnotou poľa typ.

Tab. 1.10 Rozšírenie správy RREQ/RREP s jedným podpisom v protokole SAODV

<table>
<tr><td colspan="32">0 1 2 3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1</td></tr>
<tr><td colspan="8">Typ = 64/65</td><td colspan="8">Dĺžka</td><td colspan="8">Hašovacia funkcia</td><td colspan="8">Maximálny počet skokov</td></tr>
<tr><td colspan="32">Vrchný haš
...</td></tr>
<tr><td colspan="8">Metóda podpisu</td><td>H</td><td colspan="23">Rezervované</td><td colspan="8">Dĺžka paddingu</td></tr>
<tr><td colspan="32">Verejný kľúč
...</td></tr>
<tr><td colspan="32">Padding (voliteľné)
...</td></tr>
<tr><td colspan="32">Podpis
....</td></tr>
<tr><td colspan="32">Haš
...</td></tr>
</table>

- **Dĺžka:** udáva dĺžku dát, bez polí typ a dĺžka.
- **Hašovacia funkcia:** funkcia použitá na výpočet hodnôt polí vrchný haš a haš.
- **Maximálny počet skokov:** maximálny počet skokov podporovaný autentizáciou
- **Vrchný haš:** Haš pre autentizáciu počtu skokov. Hodnota má variabilnú dĺžku v násobkoch 32-bitoch.

- **Metóda podpisu:** metóda použitá na výpočet podpisov.
- **H:** príznak polovičného identifikátora. Používaný pri metóde HMAC.
- **Rezervované:** nastavené na hodnotu 0
- **Dĺžka paddingu:** udáva dĺžku paddingu v 32-bitových jednotkách.
- **Verejný kľúč:** Verejný kľúč uzla, ktorý správu vytvoril. Hodnota má variabilnú dĺžku v násobkoch 32-bitoch.
- **Padding:** náhodný padding. Jeho dĺžka je reprezentovaná poľom dĺžka paddingu.
- **Podpis:** podpis všetkých polí paketu, ktoré sa nachádzajú pred týmto poľom a poľom počtu skokov. Hodnota má variabilnú dĺžku v násobkoch 32-bitoch.
- **Haš:** hašová hodnota aktuálneho počtu skokov. Hodnota má variabilnú dĺžku v násobkoch 32-bitoch.

1.2.3.2 Princíp podpisov

Pri počítaní podpisov je vždy vynulované pole počtu skokov, pretože je jediným premenlivým poľom. Takisto sú vynulované príznaky R a A v prípade správy RREP. Vždy keď uzol vygeneruje RREQ správu, tak sa musí rozhodnúť či využije jeden, alebo dva podpisy. Všetky implementácie SAODV protokolu musia podporovať možnosť jedného podpisu. Možnosť dvoch podpisov je voliteľná.

Po prijatí RREQ správy je podpis overený pomocou verejného kľúča. Po úspešnom overení si uzol uloží, alebo zaktualizuje cestu. V prípade použitia dvoch podpisov je spolu s cestou uložený podpis, životnosť, a IP adresa cieľa. Tento záznam sa použije na generovanie RREP správ pri dopytoch na túto cestu pomocou RREQ správ. Správy bez podpisu by mali byť zahadzované.

Správa RREP je vygenerovaná cieľovým uzlom po prijatí správy RREQ. Tak ako v prípade RREQ, musia všetky implementácie SAODV protokolu podporovať RREP správy s jedným podpisom. Dva podpisy sú voliteľné.

Uzly, ktorými správa RREP prechádza overia podpis a v prípade jeho správnosti si s cestou uložia aj informácie o podpise, životnosti a IP adrese odosielateľa. RREP správy bez podpisu by mali byť zahodené.

Podpisované a overované sú aj správy RERR používané pri chybách a zmenách topológie.

1.2.3.3 Princíp hašov

Hašovacie reťazce sú v protokole AODV použité na autentizáciu počtu skokov v smerovacích správach.

Vždy keď chce uzol odoslať správu RREQ, alebo RREP, tak vygeneruje náhodné číslo, ktoré vloží do poľa haš a nastaví hodnotu poľa maximálneho počtu skokov na rovnakú, ako je hodnota TTL v IP hlavičke. Hodnota poľa vrchného hašu sa vypočíta viacnásobných hašovaním vygenerovaného náhodného čísla. Počet hašovaní reprezentuje hodnota maximálneho počtu skokov.

Pri prijatí správy tak uzol overí počet skokov zahašovaním hodnoty poľa heš toľkokrát, koľko je hodnota v poli maximálny počet skokov a následne túto hodnotu porovná s hodnotou poľa vrchný heš. Ak sa hodnoty nerovnajú, tak je paket zahodený.

Pole heš je jeden krát hašované pred preposielaním správ RREQ a RREP.

Funkcia na výpočet hašu je určená v poli paketu. Toto pole je chránené proti zmene pomocou podpisu uzla, ktorý pôvodnú správu vygeneroval. Ak uzol funkciu v tomto poli nepozná, tak paket zahodí.

1.2.4 SRP

SRP (Secure Routing Protocol) [11] je podobne ako SAODV rozširujúcim protokolom. Najjednoduchšie je aplikovateľný pre protokoly ZRP, DSR a ABR. Protokol zabezpečuje obojsmernú komunikáciu medzi koncovými uzlami. Medzi týmito uzlami musí byť vytvorená bezpečnostná asociácia (SA) pri ktorej si tieto uzly vytvoria zdieľaný kľúč. SA je nutná iba medzi koncovými uzlami, nie je potrebné ju vytvárať s uzlami cez ktoré komunikácia prechádza.

Protokol chráni proces hľadania cesty. Efektívnosť a škálovateľnosť riešenia je zabezpečená nevyžadovaním SA na prechodných uzloch. Pri procese hľadania cesty vygeneruje zdrojový uzol pár identifikátorov. Sekvenčné číslo dopytu a náhodný identifikátor dopytu. Zdroj, cieľ a jedinečné identifikátory spolu so zdieľaným kľúčom slúžia ako vstup pre výpočet autentizačného kódu správy (MAC).

1.2.4.1 Protokol hľadania susedov

Protokol NLP (Neighbor Lookup Protocol) je časťou protokolu SRP a je zodpovedný za udržiavanie a mapovanie MAC adries na IP adresy susedov, identifikáciu potencionálnych nezrovnalostí, ako je viacero IP adries používaných jedným rozhraním, a meranie frekvencie prijímania kontrolných paketov od každého suseda, najmä na základe MAC adries. Týmto spôsobom môže byť prichádzajúca premávka z uzlov, ktoré vykazujú sebecké správanie zahodená.

NLP ukladá 48 bitové MAC adresy a IP adresy prijatých rámcov do tabuľky susedov. Každý záznam v tabuľke susedov má časovač straty suseda. Po jeho vypršaní je záznam

z tabuľky odstránený. Pomocou tohto mapovania môže NLP informovať SRP o viacerých udalostiach. Susedný uzol môže použiť inú IP adresu, akú doteraz používal. Dva susedné uzly môžu používať rovnakú IP, alebo MAC adresu, čo môže byť znakom „spoofingu“.

Monitoruje taktiež frekvencie prijímania správ od susedných uzlov. Zabezpečuje tak ďalšiu ochranu proti sebeckým a škodlivým uzlom.

1.2.4.2 Hľadanie cesty

Ako už bolo spomínané, protokol SRP je navrhnutý ako rozšírenie iných MANET protokolov. Jeho hlavička sa v rámci nachádza pred hlavičkou použitého základného protokolu. Túto rozširujúcu hlavičku môžeme vidieť v tabuľke 1.11.

Tab. 1.11 Formát hlavičky SRP protokolu

0								1								2								3		
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Typ								Rezervované																							
Identifikátor dopytu																															
Sekvenčné číslo dopytu																															
SRP MAC																															

- **Typ:** identifikuje typ SRP paketu. Aktuálna verzia SRP definuje dva typy paketov: požiadavka na cestu a odpoveď na cestu.
- **Identifikátor dopytu:** náhodné číslo vygenerované zdrojom, ktoré je spolu so zdrojovou adresou dopytu použité na jedinečnú identifikáciu požiadavky na cestu.
- **Sekvenčné číslo dopytu:** rastúce sekvenčné číslo spojené s každým cieľom, s ktorým zdrojový uzol komunikuje.
- **SRP MAC:** autentizačný kód správy, ktorý je výstupom z hašovacej funkcie s použitím zdieľaného tajného kľúča.

Zdrojový uzol si udržiava sekvenčné čísla dopytov pre každý z cieľových uzlov s ktorými komunikuje. Každá požiadavka zdrojového uzla smerom k cieľu navýši toto číslo o 1, čím umožní cieľovému uzlu detegovať staré požiadavky. Počiatočné sekvenčné číslo je dohodnuté počas bezpečnostnej asociácie.

Pre každú požiadavku na cestu je zdrojovým uzlom vygenerovaný identifikátor dopytu. Toto číslo je výstupom pseudonáhodného generátora čísel, ktorého výstup je štatisticky nerozlíšiteľný od pravých náhodných čísel a je teda nepredvídateľné vzhľadom

na výpočtový výkon mobilných zariadení. Toto číslo využívajú preposielajúce uzly na jednoznačnú identifikáciu správy.

Spolu s týmito číslami je pri generovaní požiadavky na cestu vypočítaná 96 bitová hašovacia hodnota vypočítaná pomocou zdieľaného kľúča. Pri výpočte sa neuvažujú premenlivé polia ako zdrojové IP adresy, počet skokov a v prípade protokolu DSR zoznam IP adries uzlov, cez ktoré požiadavka prešla.

Uzly, ktoré príjmu požiadavku na cestu skontrolujú či sa v rámci nachádza hlavička protokolu SRP. Ak sa táto hlavička v rámci nenachádza, tak je zahodený. Na základe identifikátora dopytu zistí uzol, či už rovnakú správu spracoval. Ak nie, porovná adresu z ktorej správu prijal so svojou tabuľkou susedov. Správa pochádzajúca od uzla nenachádzajúceho sa v tabuľke susedov je zahodená. Po týchto kontrolách si aktualizuje záznam vo svojej tabuľke dopytov, ktorej formát je zobrazený v tab. 1.12. Následne je dopyt na základe tabuľky dopytov poslaný ďalej definovaným susedom. Ak je zoznam dopredných adries prázdny, požiadavka je poslaná všetkým susedom. Pole je aktualizované na základe zoznamu adries zo zapuzdreného protokolu.

Tab. 1.12 Formát tabuľky dopytov protokolu SRP

Zdrojová adresa	Cieľová adresa	Identifikátor dopytu	Dopredný zoznam adries

Pokiaľ je uzol, ktorý správu požiadavky na cestu prijal, cieľovým uzlom, porovná sekvenčné číslo dopytu s posledným prijatým od zdrojového uzla. Ak je prijaté číslo nižšie alebo rovnaké, tak je paket je zahodený. Následne vypočíta a porovná hašovaciu hodnotu rámca. Ak sa zhodujú, zaktualizuje si uloženú hodnotu sekvenčného čísla pre zdrojový uzol a vygeneruje správu odpoveď na cestu.

Táto odpoveď obsahuje rovnaké hodnoty identifikátora dopytu a sekvenčného čísla dopytu ako v prijatej požiadavke. Vypočítaná je nová hašovacia hodnota a správa je poslaná rovnakou cestou akou bola prijatá (pomocou zoznamu adries v protokole DSR).

Uzly preposielajúce túto správu na základe zoznamu adries skontrolujú či odpoveď prijali od správneho suseda. V opačnom prípade je správa zahodená. Zabezpečí sa tak poslanie správy po rovnakej ceste, akou bola posielaná požiadavka na cestu.

Uzol ktorému je správa určená skontroluje, či odpoveď na cestu čaká, overí hašovaciu hodnotu a uloží si cestu z odpovede, ktorá už je obojsmerná.

1.2.4.3 Zmeny topológie

Na notifikácie o zmenách topológie a vyskytnutých chybách sú použité príslušné správy zapuzdreného protokolu. Tieto správy neobsahujú kompletnú SRP hlavičku, keďže pole typ definuje iba správy hľadania a odpovede na cestu. V prípade, že má uzol generujúci túto správu nadviazanú bezpečnostnú asociáciu s cieľovým uzlom správy, môže použiť na jej zabezpečenie iba pole SRP MAC s vypočítanou hašovacou hodnotou.

1.2.5 ARAN

ARAN je samostatným smerovacím protokolom, ktorý nie je rozšírením klasických smerovacích protokolov v MANET sieťach ako to je pri protokoloch SAODV a SRP. Autori opísali princíp protokolu v [21]. Tento návrh neobsahuje konkrétne formáty rámcov, ale len opis ich obsahu.

Zabezpečenie smerovacieho protokolu je realizované pomocou kryptografických certifikátov. ARAN pozostáva z počiatočného certifikačného procesu, ktorý je nasledovaný procesom hľadania ciest zabezpečujúci garanciu end-to-end autentizácie. Je jednoduchý v porovnaní s väčšinou nezabezpečených smerovacích protokolov v MANET sieťach. Hľadanie cesty je realizované pomocou posielania broadcastových správ, na ktoré odpovedá cieľový uzol unicastovými správami. Všetky smerovacie správy sú autentizované na každom skoku medzi zdrojom a cieľom.

1.2.5.1 Certifikácia

ARAN vyžaduje použitie dôveryhodného certifikačného servera T, ktorého verejný kľúč je známy všetkým uzlom. Generované kľúče sú vymieňané pomocou existujúceho spojenia medzi uzlom a serverom T. Toto spojenie by malo byť realizované iným kanálom, aký sa používa na komunikáciu medzi uzlami.

Pred tým ako sa uzol stane súčasťou ad hoc siete, musí od T získať certifikát, ktorý sa používa pri autentizácii smerovacích správ. Každý uzol získa po bezpečnej autentizácii práve jeden certifikát. Metóda autentizácie s certifikačným serverom nie je pevne definovaná. Uzol A získa certifikát od T *v* nasledovnom tvare:

$$T \rightarrow A : cert_A = [IP_A, K_{A+}, t, e]\ K_{T-}$$

Certifikát obsahuje IP adresu uzla A, verejný kľúč K_{A+} uzla A, časovú značku t kedy bol certifikát vytvorený a čas exspirácie certifikátu e. Hodnoty týchto premenných sú podpísané T. Všetky uzly musia udržiavať aktuálne certifikáty s dôveryhodným serverom.

V prípade, že je nutné zrušiť určitý certifikát, server T odošle broadcastovú správu oznamujúcu jeho zrušenie. Zrušenie vydaného certifikátu $cert_A$ je nasledovné:

$$T \rightarrow \text{broadcast} : [\text{revoke}, \text{cert}_A]\ K_{T-}$$

Každý uzol, ktorú príjme túto správu ju automaticky odošle všetkým svojím susedom. Informácie o zrušení certifikátu musia byť uložené do doby expirácie pôvodného certifikátu.

1.2.5.2 Hľadanie cesty

Pomocou end-to-end autentizácie môže zdrojový uzol overiť, či správa dorazila až na cieľový uzol, ktorý rozhodne o výslednej ceste.

Proces hľadania cesty inicializuje zdrojový uzol A do cieľového uzla X pomocou odoslania správy RDP (Route Discovery Packet) všetkým svojím susedom:

$$A \rightarrow \text{broadcast} : [\text{RDP}, IP_X, \text{cert}_A, N_A, t]\ K_{A-}$$

Správa RDP obsahuje identifikátor typu paketu RDP, IP adresu IP_X cieľového uzla X, certifikát cert_A uzla A, číslo N_A a aktuálny čas t. Správa je podpísaná súkromným kľúčom uzla A.

Číslo N_A má podobnú funkciu ako sekvenčné čísla v už predstavených protokoloch. Pri každom hľadaní cesty je toto číslo inkrementované. Spolu s časovou značkou slúži na detekciu starých správ. Po prijatí RDP správy si uzol uloží číslo N_A a časovú značku spolu s adresou uzla od ktorého správa prišla. Akceptované sú iba správy, ktorých číslo N_A je väčšie a časová značka je rovnaká, Alebo je číslo N_A menšie s väčšou časovou značkou, ako uložené hodnoty. Pole počet skokov sa v protokole ARAN nenachádza.

Po prijatí správy RDP, uzol použije verejný kľúč uzla A, nachádzajúceho sa v certifikáte cert_A na overenie platnosti certifikátu a podpisu K_{A-}. Pomocou dvojice N_A, časová značka sa zistí či už táto správa nebola niekedy spracovaná. Následne si uzol uloží reverznú cestu k zdrojovému uzlu cez suseda, od ktorého správu prijal. Odpoveď na túto správu bude tak môcť byť doručená zdrojovému uzlu. Pred ďalším odoslaním správy všetkým susedom uzol podpíše celú správu a pripojí k nej svoj vlastný certifikát. Ak uvažujeme uzol B, ktorý príjme RDP správu od uzla A, tak vytvorená správa bude vyzerať nasledovne:

$$B \rightarrow \text{broadcast} : [[\text{RDP}, IP_X, \text{cert}_A, N_A, t]K_{A-}]\ K_{B-}, \text{cert}_B$$

Sused C príjme túto správu, overí podpis uzla B, odstráni podpis a certifikát uzla B, uloží si cestu k uzlu A cez uzol B, znova podpíše pôvodnú RDP správu, pripojí k nej svoj certifikát a odošle ju všetkým svojím susedom:

$$C \rightarrow \text{broadcast} : [[\text{RDP}, IP_X, \text{cert}_A, N_A, t]\ K_{A-}]\ K_{C-}, \text{cert}_C$$

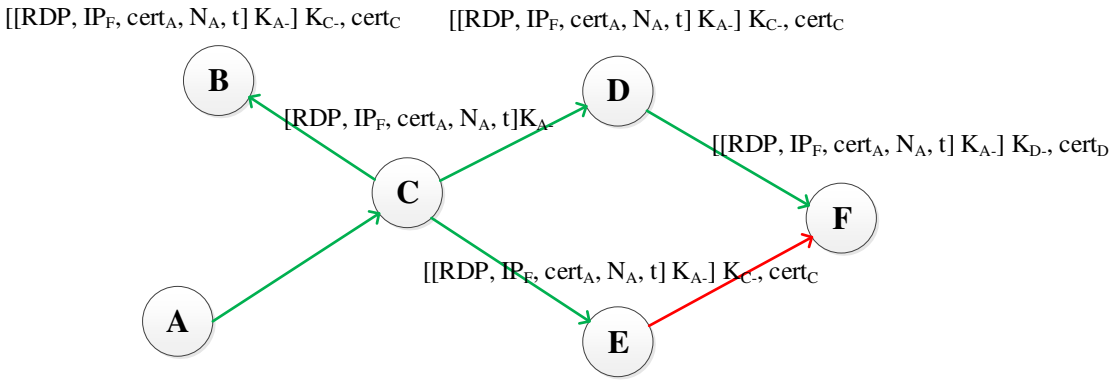
Tento proces sa opakuje až kým správa nie je doručená cieľovému uzlu X. Uzol X odpovie iba na prvú prijatú RDP správu. Ostatné správy od rovnakého zdroja s rovnakou dvojicou N_A, časová značka sú zahodené. Protokol nezaručuje doručenie správy najkratšou, ale najrýchlejšou cestou. Útočník nemôže ovplyvniť voľbu cesty z dôvodu chýbajúceho poľa počtu skokov a podpisovaní správy na každom uzle. Ako odpoveď je vygenerovaná správa REP (Reply) poslaná unicastom reverznou cestou, ktorou bola doručená správa RDP. Predpokladajme, že uzol X dostal RDP správu od uzla D. Odpoveď od uzla X bude potom nasledovná:

$$X \rightarrow D : [REP, IP_A, cert_X, N_A, t] K_{X-}$$

Obsah správy je podobný ako obsah správy RDP, zmenil sa iba typ správy (REP) a spôsob jej posielania. Dvojica N_A, časová značka t je rovnaká aká bola v prijatej RDP správe. Uzly, ktoré príjmu túto správu ju overia a podpíšu rovnakým spôsobom ako pri správe RDP. REP správa je ale vďaka ukladaniu ciest posielaná vždy unicastom reverznou cestou. Pomocou dvojice N_A, časová značka môže uzol A po prijatí správy jednoznačne priradiť odpoveď REP k odoslanej RDP správe a uložiť si zistený nasledujúci uzol pre komunikáciu s uzlom X.

1.2.5.3 Príklad hľadania cesty

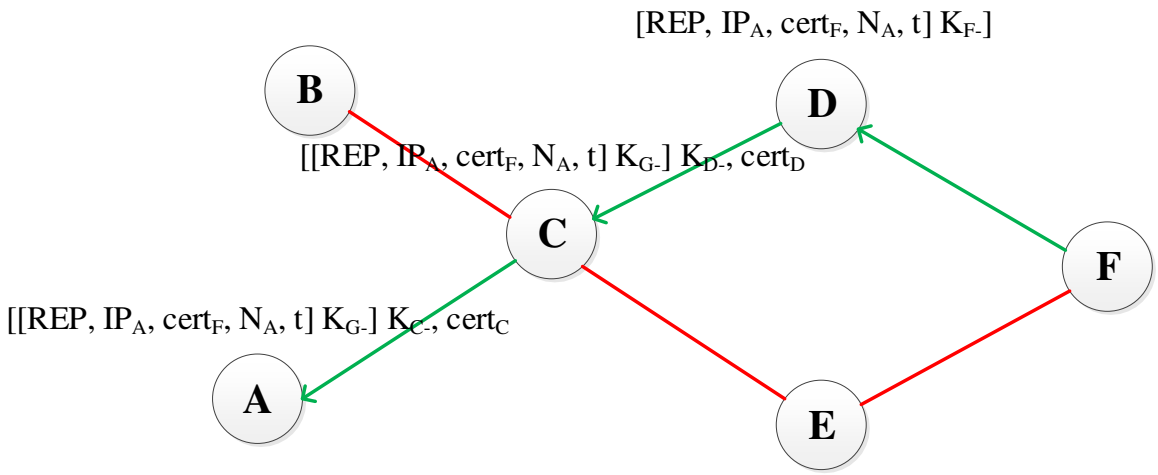
Princíp fungovania protokolu ARAN môžeme vidieť na obrázkoch 1.8 a 1.9. Predpokladajme, že na začiatku sú smerovacie tabuľky prázdne a uzol A chce zistiť cestu k uzlu F. Nad každým uzlom je zobrazený formát prijatej správy. Predpokladáme, že uzol F príjme RDP správu ako prvú od uzla D. Tým pádom prijatú správu z uzla F na základe dvojice N_A, časová značka zahodí. V tab. 1.13 je zobrazený aj obsah smerovacích tabuliek po doručení správy RDP do uzla F. Po doručení odpovede REP pribudne uzlu A cesta do F cez uzol C.



Obr. 1.8 Posielanie správy RDP v protokole ARAN

Tab. 1.13 Obsah smerovacích tabuliek po doručení správy RDP

Uzol	Cieľový uzol	Nasledujúci uzol	Životnosť (s)
A	-	-	-
B	A	C	59
C	A	A	59
D	A	C	60
E	A	C	60
F	A	D	60



Obr. 1.9 Posielanie správy REP v protokole ARAN

1.2.5.4 Zmeny topológie

ARAN je reaktívny protokol. Uzly si udržiavajú informáciu o tom, či sú uložené cesty aktívne. Ak sa neobjaví žiadna sieťová premávka na uloženej ceste po určitý čas, tak je cesta v smerovacej tabuľke deaktivovaná. Prijaté dáta na deaktivovanej ceste spôsobia vygenerovanie ERR (Error) správy a jej odoslanie zdrojovému uzlu cesty. ERR správy sa používajú aj na pád liniek medzi uzlami spôsobených pohybom uzlov. Všetky ERR správy musia byť podpísané. Pre cestu medzi zdrojovým uzlom A a cieľovým uzlom X pošle uzol B susedovi C správu v nasledovnom formáte:

$$B \rightarrow C : [\ ERR, IP_A, IP_X, cert_B, N_B, t]\ K_{B-}$$

Správa je doručená až do zdrojového uzla bez modifikácie. Ďalšie uzly ktoré ju preposielajú ju teda už nepodpisujú a je tak možné jednoznačne určiť ktorý uzol správu vytvoril. Vďaka podpisu nemôžu škodlivé uzly generovať ERR správy za iné uzly. Akcia po prijatí tejto správy zdrojovým uzlom cesty nie je definovaná. Odporúča sa opätovná inicializácia procesu hľadania cesty.

Ošetrenie chýb neplatných certifikátov, zlých podpisov a zneužitia ERR správ nie je taktiež definované a ponecháva sa na implementujúceho.

1.2.6 Zhodnotenie smerovacích protokolov

V tejto časti sme zanalyzovali rôzne reaktívne smerovacie protokoly MANET sietí. AODV a DSR sú nezabezpečené protokoly s možnými rozšíreniami SAODV a SRP, pomocou ktorých ich je možné zabezbečiť. Úplne samotných nerozširujúcim zabezbečeným protokolom je ARAN.

Všetky spomínané protokoly okrem DSR využívajú na ukladanie ciest smerovacie tabuľky. Pri správe ciest síce využívajú rozdielne metódy, ale založené sú na podobnom princípe ako sekvenčné čísla, časové značky, ich kombinácia a podobne. Pomocou špeciálnych identifikátorov tak dokážu rozlíšiť medzi starými a novšími informáciami.

Každý z týchto protokolov má určité výhody a nevýhody. Napríklad SRP na rozdiel od SAODV je možné použiť s rôznymi smerovacími protokolmi ako ZRP, DSR a ABR. Neviaže sa tak len na jeden protokol. Výhodou SAODV je možnosť zvolenia počtu podpisov, keď pri požiadavke na vyššiu bezpečnosť môžeme použiť verziu s dvomi podpismi a pri požiadavke na nižšie zaťaženie uzlov zase verziu s jedným podpisom.

Nevýhodami DSR a SAODV sú veľké hlavičky. DSR hlavička obsahuje zoznam všetkých uzlov, ktorými správa prešla a pri veľkej sieti tak môže dosiahnuť veľmi veľkú veľkosť. Väčšia hlavička v prípade SAODV je výrazná najmä pri použití dvoch podpisov.

Celkovo sme pri výbere zabezpečených protokolov dbali na to, aby používali rôzne spôsoby zabezpečenia. SRP je založený na symetrickej kryptografii, ARAN na asymetrickej kryptografii a SAODV využíva hybridný princíp.

Spomínané vlastnosti protokolov a ich ďalšie výhody, nevýhody a vlastnosti môžeme vidieť v tabuľke 1.14.

Tab. 1.14 Porovnanie smerovacích protokolov

Parameter	DSR	AODV	SAODV	SRP	ARAN
Bezslučkový	Áno	Áno	Áno	X	Áno
Cesty	Jednosmerné aj Obojsmerné	Jednosmerné	Jednosmerné	Obojsmerné	Obojsmerné
Multicast	Áno	Nie	Nie	X	Nie
Podpora QoS	Nie	Nie	Nie	Nie	Nie
Uloženie ciest	Vyrovnávacia pamäť celá cesta	Smerovacia tabuľka nasledujúci skok	Smerovacia tabuľka nasledujúci skok	X	Smerovacia tabuľka nasledujúci skok
Správa ciest	Zoznam adries	Sekvenčné čísla	Sekvenčné čísla	Identifikátor dopytu Sekvenčné číslo dopytu	Číslo N_A, časová značka, životnosť
Hello správy	Nie	Áno	Áno	Nie	Nie
Výhody	viaccestný	Malá hlavička	1 – 2 podpisy	Použitie s rôznymi protokolmi	Jednoduchosť
Nevýhody	Veľká hlavička v rozsiahlych sieťach	Periodické Hello správy	Veľká hlavička	Žiadna validácia správ RRER	Nutnosť certifikačného servera
Zabezpečenie	X	X	Hybridný	Symetrická kryptografia	Asymetrická kryptografia

1.3 Útoky v MANET sieťach

Zraniteľnosť ad hoc sietí je oveľa väčšia ako pri bežných pevných sieťach. Rozlišujeme pritom dva základné typy útokov, a to pasívne a aktívne útoky [2].

Pri pasívnych útokoch útočník vôbec nezasahuje do sieťovej premávky a snaží sa získať čo najviac informácií jej odpočúvaním. Aktívne útoky zasahujú do komunikácie a pomocou odosielaných správ sa snažia využiť rôzne zraniteľnosti napríklad s cieľom limitovania dostupnosti uzlov a získania prístupových práv. Rozdeliť ich môžeme na štyri nasledujúce skupiny.

Zahadzujúce útoky zahadzujú všetky správy, ktoré im nie sú určené. Môžu tak zabrániť komunikácii medzi uzlami v sieti. Modifikačné útoky upravujú správy s cieľom

narušenia celkovej komunikácie medzi uzlami v sieti. Príkladom takéhoto útoku sú potápajúce diery. Fabrikačné útoky posielajú susedným uzlom falošné správy bez toho, aby od nich ako odpoveď dostali akékoľvek relevantné správy. Pri poslednej skupine napodobňujúcich útokov skrýva uzol svoju identitu predstieraním iného uzla.

Jednotlivé hrozby môžeme taktiež rozdeliť podľa vrstiev modelu RM OSI [3], viď tabuľka 1.15.

Tab. 1.15 Útoky v MANET sieťach rozdelené podľa modelu RM OSI

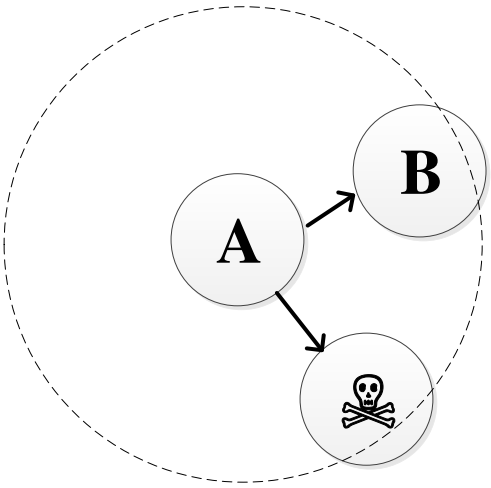
Vrstva modelu RM OSI	**Útoky**
Aplikačná vrstva	Škodlivý kód, vírusy a červy
Transportná vrstva	Unesenie relácie, SYN záplava
Sieťová vrstva	Záplavy, čierne diery, sivé diery, červie diery, falšovanie liniek
Linková vrstva	Analýza a monitorovanie premávky, sebecké správanie
Fyzická vrstva	Odpočúvanie, rušenie

1.3.1 Útoky na fyzickej vrstve

Bezpečnosť fyzickej vrstvy je v MANET sieťach veľmi dôležitá. Táto vrstva sa musí veľmi rýchlo adaptovať kvôli rapídnym zmenám v charakteristike spojení. Bezdrôtové signály je možné na rozdiel od signálov v pevných sieťach veľmi jednoduché odchytiť. Medzi najčastejšie útoky na tejto vrstve patria odpočúvanie a rušenie, ktoré sú bližšie v tejto časti popísané.

1.3.1.1 Odpočúvanie (Eavesdropping)

Odpočúvanie je typ pasívneho útoku, pri ktorom útočník nezasahuje do posielaných dát. V mobilných ad hoc sieťach zdieľajú uzly spoločné bezdrôtové médium, ktoré pre komunikáciu používa broadcast. Takýto signál môže byť jednoducho odchytený a analyzovaný. Pomocou odpočúvania môže útočník získať rôzne informácie, ako verejné a privátne kľúče, heslá, polohu zariadení a podobne. Tieto informácie môže následne využiť pri plánovaní útokov na sieť. Útok znižuje dôveryhodnosť siete a keďže sieť nie je týmto útokom ovplyvnená je takmer nemožné ho detegovať. Zabrániť takýmto únikom informácií je možné pomocou šifrovania.



Obr. 1.10 Odpočúvanie správ

1.3.1.2 Rušenie (Interference)

Rušenie je aktívny útok patriaci pod DoS útoky. Jeho cieľom je rušenie bezdrôtového signálu spôsobujúc tak stratu alebo poškodenie prenášaných správ. Existuje viacero rôznych metód rušenia bezdrôtovej komunikácie [12]. Niektoré z možností sú:

- **Konštantné rušenie:** nepretržite vysielaný signál s náhodnými bitmi, ktoré nemajú formát žiadneho MAC protokolu.
- **Klamlivé rušenie:** vysielajú sa pakety, ktorých hlavičky protokolov sú platné, ale dáta sú náhodné. Cieľom tohto rušenia je spôsobiť preťaženie malej dostupnej šírky pásma siete.
- **Náhodné rušenie:** prepínanie medzi režimami spánku a rušenia kanálov. Režimy sú prepínané po uplynutí náhodných časových intervalov. Rušenie je realizované jednou z vyššie spomenutých možností a v režime spánku je vysielanie zastavené.
- **Reaktívne rušenie:** Tento typ rušenia upravuje odchytené pakety len v niekoľkých bitoch využívajúc čo najmenej výpočtových zdrojov. Takýto paket donúti príjemcu plytvať výpočtovými zdrojmi na overenie kontrolného súčtu po ktorom je paket zahodený.

1.3.2 Útoky na linkovej vrstve

Algoritmy používané na linkovej vrstve sú častým terčom DoS útokov. Útoky na tejto vrstve môžu spôsobiť chyby pri procese hľadania cesty v smerovacom protokole, zvýšenie spotreby energie, prerušenie liniek a podobne. Najčastejšie pritom ide o sebecké správanie, analýzu a monitorovanie sieťovej premávky.

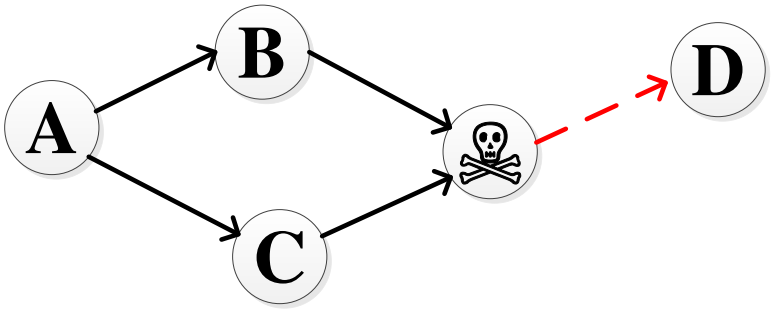
1.3.2.1 Analýza a monitorovanie premávky (Traffic analysis and monitoring)

Analýza a monitorovanie sieťovej premávky nie je vlastne útokom, ale môže viesť k zneužitiu zraniteľností siete. Útočník sa snaží zistiť podrobné informácie o sieti ako je to pri odpočúvaní, no analýzu a monitorovanie premávky je možné vykonať aj pri šifrovaní. Všímať si môže napríklad opakujúce sa vzory v šifrovaných správach. Na základe rôznych metód môže následne zo získaných dát odvodiť určité informácie o sieti, ktoré použije pri ďalších útokoch.

1.3.2.2 Sebecké správanie (Selfish behavior)

Hlavným predpokladom smerovacích protokolov v MANET sieťach je spolupráca všetkých uzlov v sieti. Uzol, ktorý sa odmieta zúčastniť procesov posielania správ a hľadania ciest, či už cielene alebo kvôli šetreniu výpočtových zdrojov a batérie, môže spôsobiť nedostupnosť určitých uzlov v sieti. V prípade cieleného správania útočník jednoducho zahadzuje prichádzajúce správy.

Detekcia takéhoto správania je náročná pretože smerovacie protokoly často nemajú mechanizmy na kontrolu smerovania správ. Výnimkou je napríklad protokol DSR, analyzovaný v časti 1.2.1, ktorý je schopný sebecký uzol detegovať pomocou zoznamu uzlov nachádzajúceho sa v rámci.



Obr. 1.11 Sebecké správanie

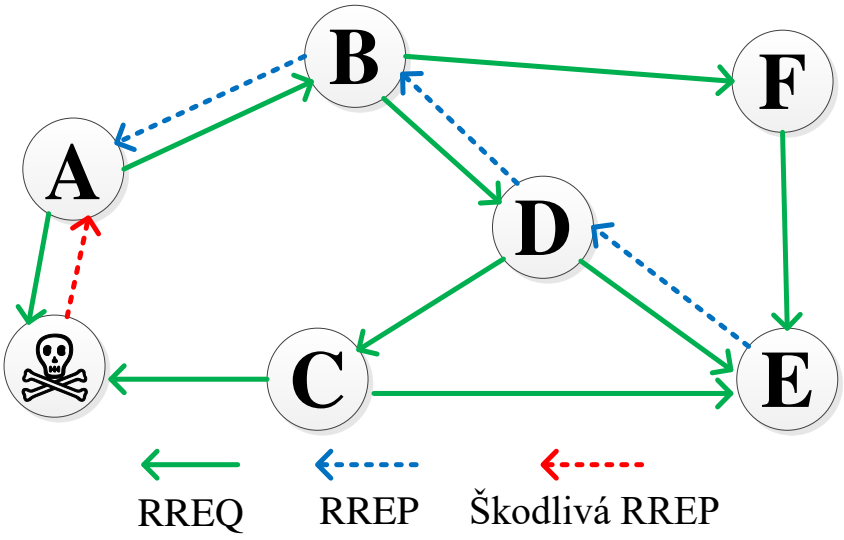
1.3.3 Útoky na sieťovej vrstve

V MANET sieťach sa uzly správajú aj ako smerovače, ktoré hľadajú a spravujú cesty medzi inými uzlami v sieti. Hlavným zámerom je vytvorenie optimálnych a efektívnych ciest. Útoky zameriavajúce sa na procesy hľadania ciest môžu narušiť komunikáciu medzi uzlami a paralyzovať tak celú sieť. Bezpečnosť na sieťovej vrstve je preto v MANET sieťach veľmi dôležitá. V tejto časti bližšie opíšeme niektoré zo známych útokov na sieťovej vrstve.

1.3.3.1 Čierna a sivá diera (Blackhole and Greyhole)

Čierna diera je spôsobená škodlivým uzlom označujúcim sám seba ako najkratšiu cestu do hľadaného cieľa. Celá komunikácia medzi uzlami tak bude smerovaná práve cez tento škodlivý uzol, ktorý môže túto situáciu využiť na ďalšie útoky. Príkladom je napríklad analýza a monitorovanie správ a zahadzovanie správ. Vytvorenie takejto čiernej diery je v MANET sieťach veľmi jednoduché [13].

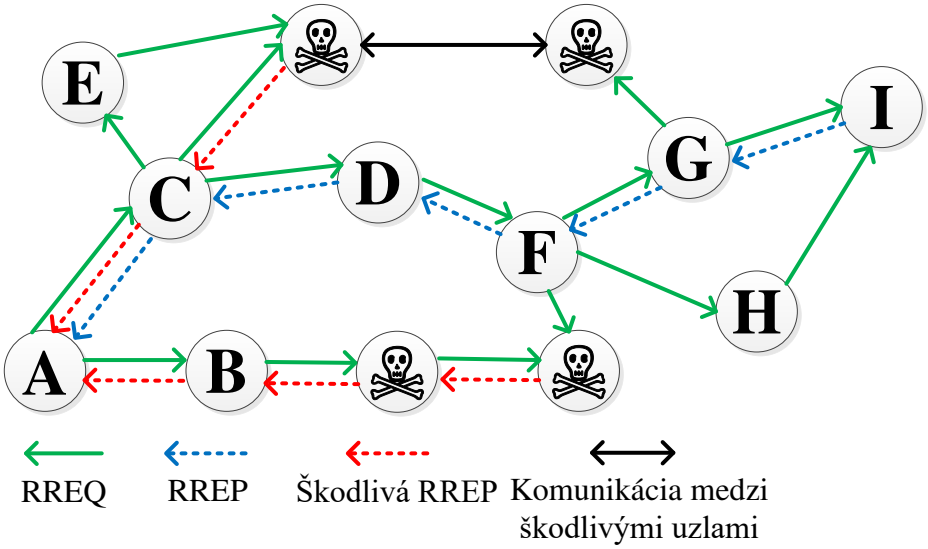
Príklad útoku je zobrazený na obr. 1.12. Uzol A hľadá cestu do uzla E. Pošle preto správu RREQ, ktorá sa sieťou prešíri tak, ako je zobrazené na obrázku. Škodlivý uzol reaguje na túto správu vygenerovaním falošnej RREP správy, v ktorej sa tvári, že má najkratšiu cestu do cieľa. Uzol A po prijatí tejto falošnej odpovede považuje proces hľadania cesty za ukončený a všetky správy pre uzol D začne posielať škodlivému uzlu. Ten môže tieto správy zahadzovať a spôsobiť tak nedostupnosť uzla D pre uzol A, alebo sám inicializuje proces hľadania cesty k uzlu D, aby mohol modifikovať komunikáciu medzi týmito uzlami.



Obr. 1.12 Útok čiernej diery

Pre zefektívnenie a vyhýbanie sa detekcii tohto útoku je možné namiesto jedného škodlivého uzla použiť dva alebo viac škodlivých uzlov. V takomto prípade hovoríme o kolaboratívnej čiernej diere. Pomocou viacerých uzlov dokáže útočník pokryť väčšiu časť siete a minimalizuje prípady v ktorých dostane zdrojový uzol odpoveď na hľadanie cesty ako prvú od cieľového uzla. Poskytujú takisto aj zálohu v prípade detekcie niektorého zo škodlivých uzlov. Príklad takéhoto útoku je na obr. 1.13. Škodlivé uzly môžu preposielať odchytené RREQ správy inému škodlivému uzlu, ktorý na ne odpovie,

alebo sa môžu správať ako klasické čierne diery, s tým, že medzi sebou komunikujú o vyskytnutých udalostiach. Ostatné čierne diery tak nemusia odpovedať na správy RREQ na ktoré už odpovedala iná čierna diera.



Obr. 1.13 Kolaboratívna čierna diera

Modifikáciou čiernej diery je sivá diera. Tá na rozdiel od čiernej diery zahadzuje prijaté pakety s určitou pravdepodobnosťou. Tento útok je tak v porovnaní s čiernou dierou, pri ktorej sú pakety vždy zahodené, náročnejší na detekciu.

Ďalším režimom sivej diery je zahadzovanie paketov určených len pre konkrétne uzly, pričom ostatné pakety sú normálne preposielané.

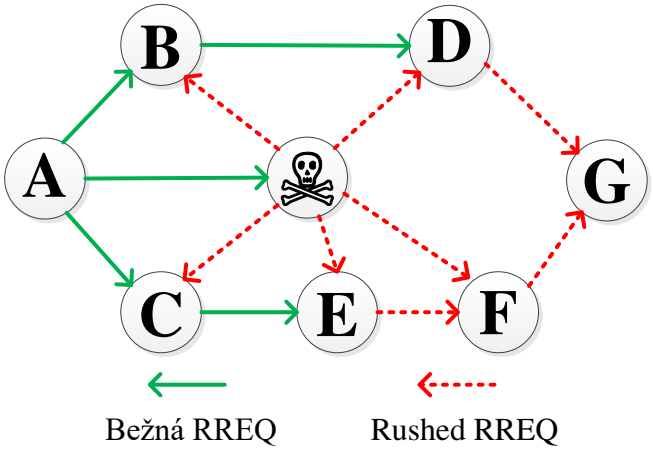
Takisto môže pakety náhodne zahadzovať len po určitý čas, po ktorom sa bude správať ako normálny uzol a neskôr sa znova prepne do režimu sivej diery, alebo môže skombinovať oba predchádzajúce režimy a ešte viacej sťažiť jej detekciu.

1.3.3.2 Rushing útok (Rushing attack)

Tento útok využíva mechanizmus potláčania duplikátov v reaktívnych protokoloch. Uzol spracuje iba prvú RREQ správu, ktorú príjme a ďalšie prijaté duplikáty tejto správy nespracuje kvôli prevencii vytvárania slučiek.

Útočník sa pri tomto útoku snaží poslať broadcastom prijatú RREQ správu čo najrýchlejšie a tým ovplyvniť proces hľadania cesty. Cesta tak nemusí byť najoptimálnejšia a bude viesť cez útočníka, čo môže zneužiť na ďalšie útoky [15].

Na obr 1.14 hľadá uzol A cestu do uzla G. Útočník pomocou rushed RREQ správ docieli zahodenie duplikátnych RREQ správ na uzloch D a E. Výsledná najkratšia cesta bude teda viesť cez útočníka.

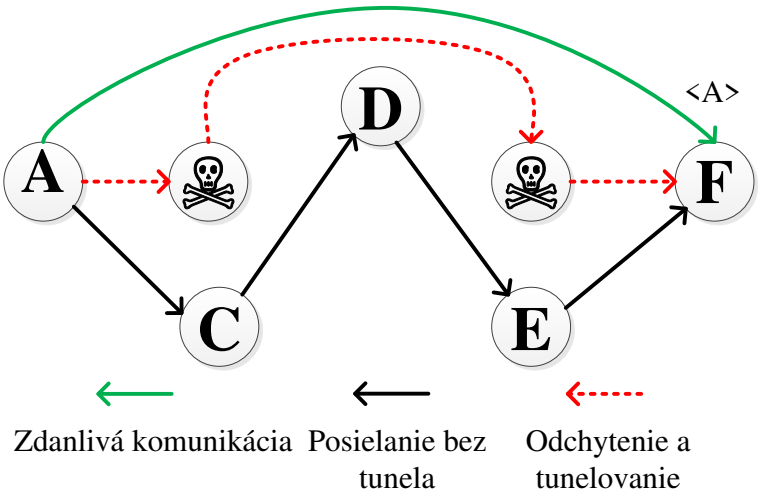


Obr. 1.14 Rushing útok

1.3.3.3 Červia diera (Wormhole)

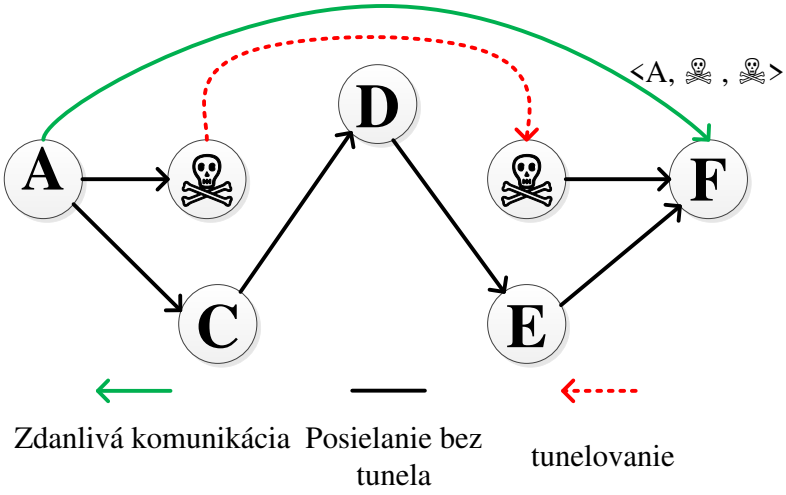
Červia diera je závažný útok realizovaný pomocou dvoch spolupracujúcich uzlov, ktoré sú prepojené priamym spojením s nízkou latenciou. Pakety prijímané jedným uzlom sú tunelované do druhého uzla. Rýchle spojenie medzi uzlami bude často najkratšou cestou pri hľadaní ciest v sieti. Pakety smerované touto cestou môžu spolupracujúce uzly odchytávať, modifikovať a zahadzovať vytvárajúc tak DoS útok. Existujú pritom tri rôzne režimy červej diery, skrytý, vystavený a kombinovaný [14].

V skrytom režime uzly skrývajú svoju prítomnosť v sieti. Ideálne je ich umiestnenie pri zdroji a cieli odosielania dát. Keďže dáta prijaté od zdroja, uzly tvoriace červiu dieru neupravujú, zdrojový a cieľový uzol komunikácie si budú myslieť, že sú susediacimi uzlami.



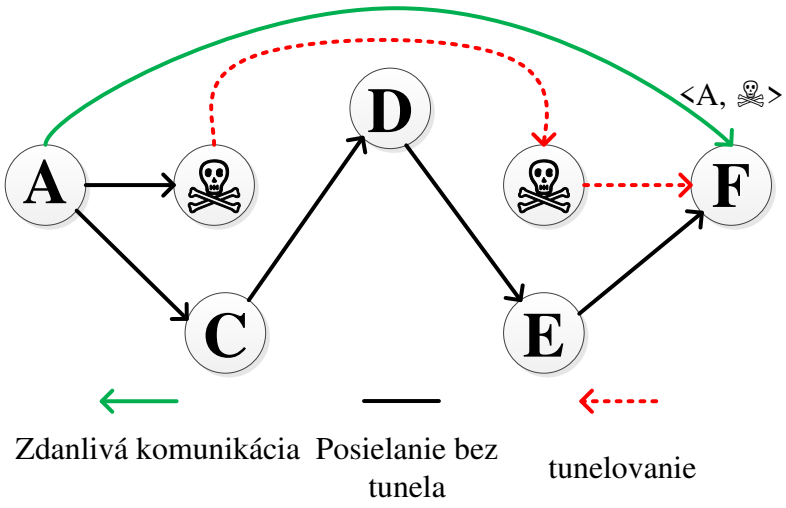
Obr. 1.15 Skrytý režim červej diery

Vo vystavenom režime uzly neskrývajú prítomnosť v sieti a aktualizujú obsah správ na základe protokolu. Rýchle spojenie medzi uzlami ale vytvorí najrýchlejšiu možnú dostupnú cestu medzi cieľovým a zdrojovým uzlom, takže si budú myslieť, že sa nachádzajú len niekoľko skokov od seba.



Obr. 1.16 Vystavený režim červej diery

Možná je aj kombinácia oboch prístupov pri ktorom jeden uzol pracuje v skrytom režime a druhý uzol vo vystavenom režime.



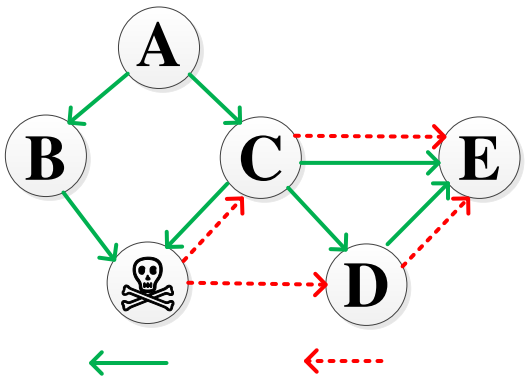
Obr. 1.17 Hybridný režim červej diery

1.3.3.4 Potápajúca diera (Sinkhole)

Jeden z najnebezpečnejších útokov v MANET sieťach je potápajúca diera. Škodlivý uzol sa snaží zo seba pomocou klamlivých smerovacích informácií spraviť špeciálny uzol, cez ktorý je smerovaná premávka celej siete. Následne môže tieto dáta modifikovať alebo zahadzovať.

Tento útok je možné realizovať pomocou zväčšovania sekvenčných čísel, alebo zmenšovania počtu skokov v kontrolných správach smerovacích protokolov. Všetky cesty sa tak budú zdať najkratšie práve cez škodlivý uzol [16].

Napríklad v protokole AODV je možné upraviť sekvenčné čísla aj počet skokov. Na obr. 1.18 útočník odošle prijatú správu s vyšším sekvenčným číslom a s nižším počtom skokov, ako v prijatej RREQ správe. Táto požiadavka bude teda požadovaná za novšiu s najkratšou cestou cez škodlivý uzol. Rovnaký spôsob je možné použiť aj pri správach RREP.



Obr. 1.18 Potápajúca diera

1.3.3.5 Opakujúce útoky (Replay attacks)

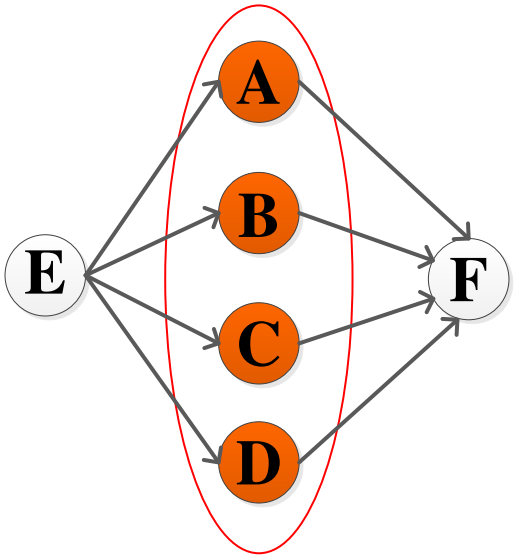
Útočník môže odchytávať a ukladať správy posielané v sieti. Posielaním starých kontrolných správ prislúchajúcich starej topológii môže dôjsť k zmenám v smerovacích tabuľkách a uzly nebudú schopné smerovania. Využiť ho je možné aj pri šifrovaných správach, pretože útočník nepotrebuje poznať ich obsah. Tento útok je neúčinný v protokoloch, ktoré obsahujú časovú značku, alebo iný časový identifikátor.

1.3.3.6 Sybil útok (Sybil attack)

V MANET sieťach majú uzly jednu jedinečnú identitu. Sybil útok vytvára viacero identít na jednom uzle. Cieľom takéhoto správania je narušenie sieťových komunikácií, získanie väčších zdrojov, informácií alebo prístupov, ako by získal jeden uzol v sieti.

Takýto škodlivý uzol je možné využiť rôznymi spôsobmi. V senzorovej sieti môže útočník vysielať dáta zdanlivo z viacerých uzlov. Takisto je možné vytvoriť viacero typov DoS útokov [17].

Na obr. 1.19 uzol E môže nadviazať spojenie s uzlom F cez 4 rôzne uzly. Tieto uzly sú ale len rôzne identity jedného škodlivého uzla.



Obr. 1.19 Sybil útok

1.3.3.7 Záplavy (Floods)

Záplavy je možné realizovať na rôznych vrstvách modelu RM OSI. Zaraďujeme ich medzi DoS útoky s cieľom zabrániť bežným používateľom prístup k rôznym službám, alebo spôsobiť chyby v sieťovej infraštruktúre.

Najčastejšou záplavou na sieťovej vrstve je záplava RREQ, ktorú je možné vykonať v reaktívnych protokoloch s týmto mechanizmom hľadania cesty. Útočník generuje veľké množstvo RREQ správ pre neexistujúce uzly.

Tieto správy tak môžu spôsobiť pretečenie smerovacích tabuliek uzlov v sieti, kvôli ktorému nebude možné uchovávať cesty k legitímnym uzlom. Záplava takisto vo vysokej miere spotrebováva dostupnú šírku pásma, limitované výpočtové a napájacie zdroje zariadení. Všetky tieto okolnosti vedú k DoS útoku [18].

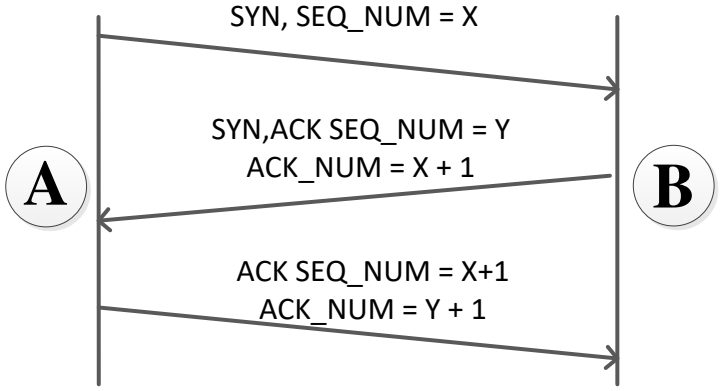
1.3.4 Útoky na transportnej vrstve

Medzi bezpečnostné hrozby na transportnej vrstve patria autentizácia, end-to-end komunikácia, ošetrenie oneskorení, straty paketov a podobne [19]. Takisto ako v pevných sieťach je možné aj v MANET sieťach uskutočniť útoky ako sú SYN záplava a unesenie relácie.

1.3.4.1 SYN záplava (SYN flood)

Na transportnej vrstve je možné na vytvorenie záplavy využiť príznak SYN v protokole TCP.

Cieľom útočníka je vytvoriť veľké množstvo napoly otvorených TCP spojení [18]. TCP spojenia sa vytvárajú pomocou trojnásobného podania rúk ako je zobrazené na obr. 1.18. Uzol A pošle uzlu B TCP segment s príznakom SYN (synchronizácia) a sekvenčným číslom X. Ten mu odpovie správou s príznakmi SYN, ACK (potvrdenie), sekvenčným číslom Y a potvrdzujúcim číslom X+1. Ako potvrdenie prijatia tejto správy odošle na záver uzol A správu s príznakom ACK, sekvenčným číslom X+1 a potvrdzujúcim číslom Y+1. Po výmene týchto troch správ je spojenie nadviazané.



Obr. 1.20 Trojnásobné podanie rúk v protokole TCP

Po prijatí prvej správy a odoslaní vygenerovanej odpovede uzol B čaká na ACK odpoveď. Pamätá si pritom adresu uzla A a stav spojenia, ktoré je napoly otvorené (nebolo ešte uzlom A potvrdené). Zdroje uzla B sú ale obmedzené a útočník tak môže generovať veľký počet správ s príznakom SYN, so zmenenými zdrojovými adresami, pomocou ktorých preťaží zdroje uzla B, ktorý nebude schopný odpovedať na ďalšie požiadavky. Uzol sa tak stane nedostupným.

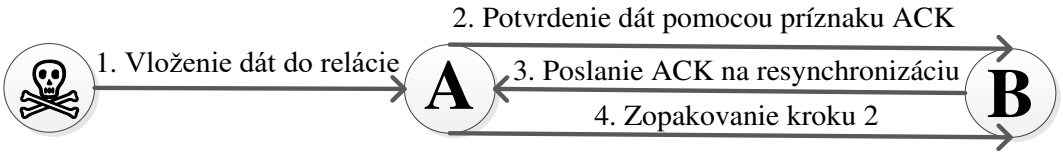
1.3.4.2 Unesenie relácie (Session hijacking)

Cieľom unesenia relácie je ukradnutie identity uzla obete a nadviazanie relácie s cieľovým uzlom. Útok sa skladá z dvoch krokov [18].

Útočník ukradne identitu obete pomocou fiktívnej zmeny vlastnej IP adresy a vypočíta sekvenčné číslo očakávané cieľovým uzlom.

Následne spustí DoS útoky na obeť, kvôli ľahšiemu nadviazaniu relácie s cieľovým uzlom.

Po spustení útoku unesenia relácie nasleduje často útok TCP ACK búrka, pomocou ktorého preruší reláciu medzi uzlom obete a cieľovým uzlom. Príklad tohto útoku môžeme vidieť na obr. 1.21. Škodlivý uzol vloží dáta do komunikácie uzlov A a B. Uzol A pošle uzlu A správu ACK, ktorou potvrdzuje prijatie týchto dát. Uzol B je zmätený keďže táto správa obsahuje neočakávané sekvenčné číslo a pošle uzlu A správu s príznakom ACK ktorá obsahuje zamýšlané sekvenčné číslo, pomocou ktorého sa snaží opäť zosynchronizovať. Kroky sa ale začnú opakovať čo vyústi do „búrky“ ACK správ [19].



Obr. 1.21 TCP ACK búrka

Uzly v MANET sieťach bývajú často náchylné na unesenia relácií aj SYN záplavy kvôli slabému zabezpečeniu transportnej vrstvy. Na ochranu proti týmto útokom je možné použiť protokoly SSL a TLS, ktoré využívajú asymetrickú kryptografiu.

1.3.5 Útoky na aplikačnej vrstve

Aplikácie sa v MANET sieťach musia vyrovnať s častým odpájaním a opätovným pripájaním zariadení, ako aj s meniacou sa dobou odozvy a stratami paketov. Táto vrstva obsahuje protokoly ako SMTP, HTTP, Telnet a FTP, ktorých zraniteľnosti sú využívané aj

v pevných sieťach. Hlavnými útokmi na aplikačnej vrstve sú realizované pomocou škodlivého kódu a útokmi nepopierateľnosti [19].

1.3.5.1 Škodlivý kód (Malicious code)

Rôzny škodlivý kód ako sú vírusy, červy a trójske kone napádajú operačné systémy aj používateľské aplikácie a môžu tak spôsobiť spomalenie systému a siete, alebo aj ich poškodenie. Ich cieľom môže byť aj získanie rôznych žiadaných informácií [19].

1.3.5.2 Nepopierateľný útok (Repudiation attack)

Nepopierateľné útoky súvisia s odmietnutím účasti na komunikácii. Príkladom môže byť komerčný systém v ktorom môže takýto uzol odmietnuť vykonanie operácie nákupu pomocou kreditnej karty [19].

1.3.6 Zhodnotenie útokov

Nasledujúca tabuľka popisuje predpokladané zraniteľnosti analyzovaných protokolov na analyzované útoky. Zraniteľnosť symbolizuje × a odolnosť ✓.

Tab. 1.16 Zraniteľnosť protokolov na útoky

Útok	DSR	AODV	SAODV	SRP	ARAN
Odpočúvanie	×	×	×	×	×
Rušenie	×	×	×	×	×
Analýza a monitorovanie	×	×	×	×	×
Sebecké správanie	×	✓	✓	✓	✓
Čierna diera	×	×	✓	✓	✓
Rushing útok	×	×	✓	✓	×
Červia diera	×	×	×	×	×
Potápajúca diera	×	×	✓	✓	✓
Opakujúce útoky	×	✓	✓	✓	✓
Sybil útok	×	×	×	×	✓
Záplavy	×	×	×	×	×
Unesenie relácie	×	×	✓	✓	✓

1.4 Zhodnotenie analýzy

Analýza práce sa zameriava na smerovacie protokoly a útoky v MANET sieťach. Začína analýzou MANET sietí a popisuje ich vlastnosti, výhody a nevýhody. Nasleduje rozdelenie

smerovacích protokolov a ich popis. Na záver analýzy sú zhrnuté rôzne útoky vyskytujúce sa MANET sieťach.

Obsah analýzy je rozdelený do troch častí. Prvá časť sa zaoberá základnými charakteristikami MANET sietí, ich aplikáciou a zabezpečením.

Druhá časť podrobne popisuje princípy činnosti vybraných nezabezpečených a zabezpečených smerovacích protokolov. Sústredili sme sa pritom iba na reaktívne protokoly. Z nezabezpečených protokolov boli analyzované protokoly DSR a AODV. Zo zabezpečených protokolov sme vybrali SAODV, SRP a ARAN. Každý z týchto protokolov zaraďujeme do rôznych skupín typu zabezpečenia.

Posledná tretia časť sa zameriava na útoky v MANET sieťach rozdelených podľa vrstiev modelu RM OSI. Najväčšia pozornosť je venovaná útokom na sieťovej vrstve, ktoré sú priamo cielené na smerovacie protokoly. V závere tretej časti sú zhodnotené zraniteľnosti zanalyzovaných protokolov na vybrané útoky.

2 Špecifikácia

2.1 Všeobecné požiadavky

Základnou požiadavkou vyplývajúcou zo zadania práce je vytvoriť testovacie scenáre pre testovanie bezpečnosti reaktívnych smerovacích protokolov mobilných ad hoc sietí. Útoky môžu byť založené na už existujúcich útokoch, alebo môžu byť aj úplne nové. Pri našich navrhnutých útokoch by sme sa mali zamerať hlavne na tie, ktoré využívajú spoluprácu viacerých uzlov v sieti.

Dôležité bude pri jednotlivých útokoch sledovanie ovplyvnených parametrov siete. Konkrétne parametre budú závisieť od typu útoku.

Testovacím scenárom budeme rozumieť viacero útokov, ktoré sú založené na podobnom princípe s podobným cieľom útoku.

2.2 Základný režim

Základný režim bez útokov bude súčasťou každého testovacieho scenáru. Bude založený na komunikácii medzi uzlami v sieti pomocou princípu ping. Definované budú nasledovné parametre:

- Počet uzlov – Definuje koľko uzlov s náhodnou pozíciou má byť vygenerovaných
- Prenosová vzdialenosť – Maximálna vzdialenosť do akej dokáže uzol úspešne poslať dáta
- Plocha – Obmedzenie priestoru v ktorom sa môžu mobilné uzly pohybovať
- Pohyblivosť – Definuje ako často budú uzly meniť svoju polohu a akou rýchlosťou sa budú pohybovať
- Počet ping správ – Počet ping správ, ktoré si každý uzol vymení s každým uzlom v sieti
- Čas čakania na odpoveď – Čas, po ktorého uplynutí sa považuje ping za neúspešný
- Opakovanie ping správ – Počet pokusov opakovania pri neprijatí odpovede do definovaného času na odoslanú ping správu

Všetky parametre, okrem počtu uzlov, budú konštantné vďaka čomu bude možné porovnať jednotlivé protokoly v rovnakých podmienkach. Parameter počtu uzlov bude slúžiť na vyhodnotenie úspešnosti útokov v závislosti od tohto parametra. Úspešnosť útokov sa bude posudzovať vzhľadom na parametre siete základného scenára pri rôznom počte uzlov.

2.3 Testovací scenár

Testovací scenár obsahuje viacero útokov nazývaných režimy so spoločnými vlastnosťami. Jednotlivé režimy scenára by tak mali mať spoločný cieľ, ako napríklad zabránenie doručeniu bežných správ. Režimy scenára budú tvorené našimi navrhnutými útokmi ako aj ich bežnými implementáciami. Vďaka tomu budeme môcť jednoducho porovnať vzniknuté rozdiely.

2.4 Sledované parametre

Pri jednotlivých testovacích scenároch budú v sieti sledované rôzne parametre sieťovej premávky. V úvahu sa budú brať parametre ktoré budú, alebo by mali byť ovplyvnené konkrétnymi typmi útokov. Príkladom sledovaných parametrov sú napríklad počet úspešných výmen ping správ, počet neúspešných výmen ping správ, štatistiky počtu skokov (minimum, maximum, priemer), počet expirovaných záznamov v smerovacej tabuľke, počet vygenerovaných požiadaviek na cestu.

2.5 Implementácia protokolov

Implementované protokoly by mali byť plne funkčné z hľadiska sieťovej vrstvy a spĺňať požiadavky jednotlivých RFC, ktoré ich opisujú. Miera potreby implementácie ostatných vrstiev je limitovaná na ich základnú funkcionalitu, keďže útoky sú cielené na smerovacie protokoly, ktoré sú súčasťou sieťovej vrstvy.

2.6 Vyhodnotenie scenárov

Scenáre budú vyhodnotené na základe viacerých simulácií. Simulácie s určitým počtom uzlov by mali byť zopakované niekoľkokrát pre získanie priemerných výsledkov. Výsledky jednotlivých scenároch budú zobrazené v prehľadných grafoch s príslušným komentárom a ohodnotením jednotlivých útokov v danom scenári.

3 Návrh

V tejto časti práce sa nachádza návrh testovacích scenárov pre MANET siete. Obsahuje nami navrhnuté útoky, spolu s ich podrobným opisom a scenárom, ktorý sa bude v rámci daného útoku testovať, ako aj sledované parametre.

3.1 Základný scenár

Základný scenár pozostáva z bežnej komunikácie medzi uzlami. Uzly sa pohybujú a navzájom medzi sebou komunikujú podľa vopred definovaných parametrov. Základná komunikácia je realizovaná na princípe ping, pri ktorom sa uzly snažia o výmenu správ medzi sebou. Pri testovaní bude každý z definovaných scenárov, vrátane základného scenára odsimulovaný niekoľkokrát. Týmto spôsobom dostaneme relevantnejšie výsledky pri porovnávaní jednotlivých sieťových parametrov.

Dôležité je upozorniť, že výsledky každého z nižšie opísaných scenárov budú porovnané vzhľadom na výsledky základného scenára, vďaka čomu bude možné lepšie porovnať účinnosť rôznych režimov útokov.

3.1.1 Sledované parametre základného scenára

Všetky sledované parametre uzlov základného scenára budú sledované aj pri neútočných uzloch navrhnutých útočných scenárov. Medzi tieto parametre patria:

- Počet úspešných výmen správ
- Celkový počet odoslaných správ
- Počty odoslaných jednotlivých typov správ (REQ, REP, RER, ping ...)
- Počty vygenerovaných jednotlivých typov správ (REQ, REP, RER, ping ...)
- Počet expirovaných záznamov v smerovacích tabuľkách
- Počet vymazaných záznamov zo smerovacích tabuliek
- Štatistiky počtu skokov (priemer, maximum, minimum, celkový súčet)
- Štatistiky oneskorenia výmen správ (priemer, maximum, minimum, celkový súčet)
- Štatistiky veľkosti správ (priemer, maximum, minimum, celkový súčet)

3.2 Pridanie oneskorenia (Žonglér/Juggler)

Cieľom útoku žonglér je spôsobiť oneskorenia v komunikácii medzi uzlami, spotrebovanie časti dostupnej šírky pásma a zvýšenie zaťaženia uzlov v sieti. Správy teda budú doručené

do cieľa, ale veľké pridané oneskorenie môže spôsobiť problémy rôznym aplikáciám, alebo frustráciu používateľa. Veľkosť pridaného oneskorenia je možné kontrolovať.

Predpoklady na úspešné vykonanie útoku sú:

- Existencia cesty od zdrojového do koncového uzla cez škodlivý uzol.
- Existencia minimálne dvoch škodlivých spolupracujúcich uzlov.

Útok sa skladá z troch fáz:

- Vykonanie útoku na zabezpečenie smerovania do cieľových uzlov cez škodlivý uzol. (Nepovinné)
- Pridanie oneskorenia vytvorenia dočasnej komunikačnej slučky medzi škodlivými uzlami.
- Doručenie správy s oneskorením.

Prvú fázu útoku je možné vykonať napríklad pomocou mechanizmov čiernych, alebo červích dier. Táto fáza je voliteľná, ale čím viac ciest bude v sieti viesť cez škodlivý uzol, tým bude dopad útoku na sieť väčší. Útok je možné vykonať aj v sieťach, ktoré sú zabezpečené proti ovplyvneniu mechanizmu hľadania cesty zo zdrojového do cieľového uzla. V takomto prípade bude ale počet ciest vedúcich cez škodlivé uzly menší, čím klesne aj efektívnosť útoku.

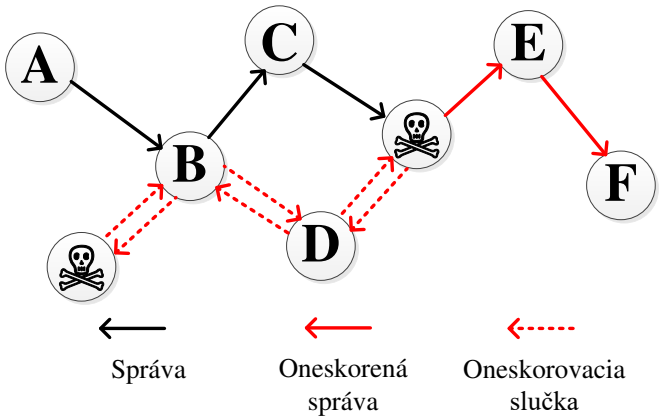
Druhá fáza pridá voliteľné oneskorenie do komunikácie medzi zdrojovým a cieľovým uzlom cesty. Jednoduchším riešením je podržanie správy pred odoslaním na definovaný interval. Naše riešenie ale využije existenciu ďalších škodlivých uzlov v sieti. Správa nie je po prijatí odoslaná ďalšiemu uzlu v ceste. Namiesto toho je správa zabalená do ďalšej správy a poslaná inému náhodnému známemu škodlivému uzlu v sieti. Škodlivý uzol po prijatí správy od spolupracujúceho škodlivého uzla pošle správu naspäť (v prípade existencie dvoch spolupracujúcich uzlov), alebo ďalšiemu inému náhodnému škodlivému uzlu (v prípade existencie troch a viacerých spolupracujúcich uzlov). Jediná zmena je vykonaná v cieľovej adrese. Vytvorí sa tak slučka, ktorá pridáva pôvodnej správe oneskorenie a vyťažuje limitované zdroje MANET siete pričom pri veľkom počte správ v slučke môže spôsobiť aj jej zahltenie.

Počiatočný škodlivý uzol, ktorý vytvorí oneskorovaciu slučku kontroluje požadované pridané oneskorenie pri každom opätovnom prijatí správy. V tretej fáze tak po dosiahnutí požadovaného oneskorenia je originálna správa odoslaná ďalej po ceste bez akejkoľvek

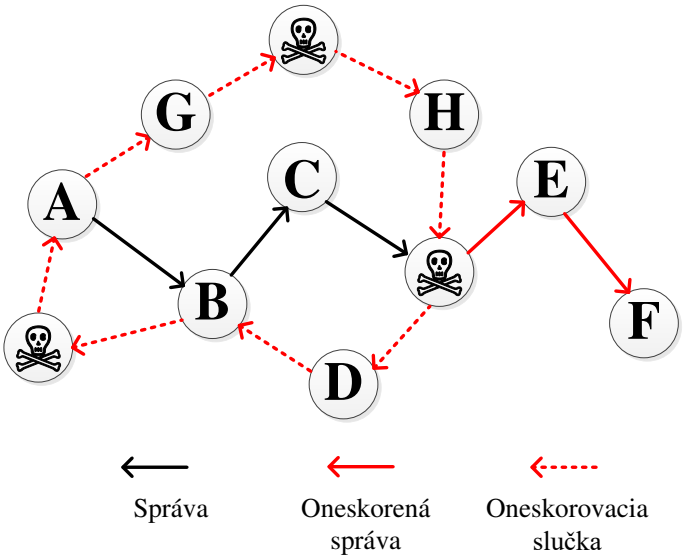
modifikácie. Dosiahne sa tak stabilné pridanie oneskorenia do kontrolovaných komunikácií medzi uzlami.

Kľúčové je udržiavanie aktívnych ciest medzi spolupracujúcimi škodlivými uzlami. Z tohto dôvodu je možné periodicky inicializovať proces hľadania cesty medzi týmito uzlami.

Na obrázkoch 3.1 a 3.2 sú zobrazené príklady útoku žonglér. Zobrazené je použitie útoku s dvoma a s tromi spolupracujúcimi škodlivými uzlami.



Obr. 3.1 Žonglér s dvoma škodlivými uzlami



Obr. 3.2 Žonglér s tromi škodlivými uzlami

3.2.1 Testovací scenár útoku žonglér

V prípade útoku žonglér bude scenár pozostávať z troch režimov, ktorých parametre budú navzájom porovnané.

Prvý režim bude pridávať staticky zadané oneskorenie bez vytvárania slučiek. Uzol správu pred odoslaním pozdrží definovaný čas a následne odošle ďalej. Režim teda nebude mať dopad na šírku pásma.

Druhý režim vynecháva voliteľnú prvú časť útoku. Škodlivý uzol nebude ovplyvňovať výber cesty zo zdroja do cieľa pomocou modifikácie správ, alebo ich tunelovaním. Tento spôsob tak bude aplikovateľný aj v prípadoch keď ovplyvnenie výberu cesty nebude možné. Navyše by mal byť bez pridanej prvej časti náročnejší na detekciu.

Tretí režim aktívne zasahuje do procesov vyhľadávania ciest a zabezpečuje aby tieto cesty viedli cez škodlivý uzol. Zvýši sa tak množstvo aktívnych ciest do ktorých môže byť pridaná oneskorovacia slučka. Celkové priemerné oneskorenie správ a aj spotrebovaná šírka pásma v sieti by mali byť vyššie ako v prípade prvých dvoch režimov.

3.2.2 Sledované parametre útoku žonglér

Pomocou troch rôznych režimov bude možné porovnať rozdiely medzi štandarným pridaním oneskorenia a naším návrhom. Porovnanie bude založené na nasledovných parametroch siete.

- Priemerný čas oneskorenia správ
- Celková veľkosť prenesených dát
- Počet správ, ktoré boli oneskorené
- Počet správ, ktoré neboli oneskorené
- Počet správ poslaných medzi škodlivými uzlami

3.3 REQ záplava

Reaktívne protokoly využívajú pri hľadaní cesty broadcastové správy, ktoré nazvime REQ (Request). Tieto správy sa prešíria celou sieťou až k cieľovému uzlu, ktorý na požiadavku hľadania cesty odpovie. Na zabránenie nekonečného broadcastu týchto správ používajú protokoly rôzne metódy. V našej analýze sa na tento účel najčastejšie používali sekvenčné čísla, alebo špeciálne broadcastové čísla. Uzol, ktorý prijme požiadavku na cestu pred jej preposlaním na broadcast skontroluje, či už v minulosti takúto správu neprijal.

Ignorované sú teda iba duplicitné žiadosti. Pri ďalšom hľadaní cesty je sekvenčné číslo navýšené a nie je tak ignorované. Navrhujeme preto po prijatí takejto správy zmeniť pole zabezpečujúce ochranu proti slučkám a správu odoslať do siete. Správa tak zaplaví celú sieť. Pre dosiahnutie konštantných záplav môžeme z uzla vysielať správy s modifikovaným poľom v pravidelných intervaloch, alebo použijeme dva škodlivé uzly,

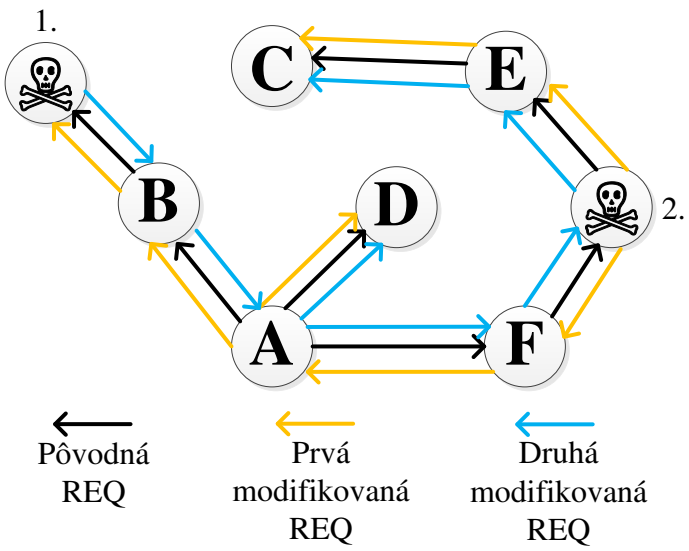
ktoré vždy budú modifikovať prijímané požiadavky na cestu a vytvoria tak nekonečné vytváranie záplav. V ďalšom variante útoku môže uzol odoslať na broadcast aj pôvodnú správu.

V prípade dvoch škodlivých uzlov ale môže dôjsť k tomu, že tieto modifikované správy nebudú druhým škodlivým uzlom prijaté a následne upravené. Oba uzly by totiž vygenerovali rovnakú správu, ktorá by sa nedostala až k druhému škodlivému uzlu, kvôli mechanizmom prevencie slučiek. Na obídenie tejto skutočnosti musia uzly generovať modifikované správy iba za určitých vzájomne sa vylučujúcich podmienok. Napríklad uzol 1 vygeneruje modifikovanú správu iba v prípade párneho sekvenčného čísla a uzol 2 iba v prípade nepárneho sekvenčného čísla. Všeobecne pri n uzloch pre i-ty uzol platí, že reaguje iba na správy ktorých sekvenčné čísla s vyhovujú nasledujúcej podmienke.

$$s \bmod n = i - 1 \tag{1}$$

Uzly tak generujú správy s modifikáciou, ktorá spôsobí vygenerovanie správy na iných škodlivých uzloch. Keďže najčastejšou obranou proti tomuto typu útoku je obmedzenie počtu vygenerovaných REQ správ za určitý čas, tak pomocou viacerých uzlov síce budeme reagovať na rovnaký počet správ, ako keby bol uzol len jeden, ale vzhľadom na časté obmedzenia budeme môcť vygenerovať n-1 krát viac správ za daný časový úsek.

Na obrázku 3.3 môžeme vidieť príklad tohto útoku. Použité sú dva škodlivé uzly pričom škodlivý uzol 1 replikuje párne sekvenčné čísla a škodlivý uzol 2 nepárne sekvenčné čísla. Pôvodná REQ má nepárne sekvenčné číslo. Taktiež predpokladáme mód s posielaním aj pôvodnej správy. Zobrazené sú iba spracované správy, ktoré neboli ignorované mechanizmom prevencie slučiek. Správy ignorované týmto mechanizmom zobrazené nie sú, ale ich spracovanie taktiež spotrebovalo časť výpočtových zdrojov uzlov. Obrázok zobrazuje iba dve modifikácie, no útok by ďalej pokračoval.



Obr. 3.3 REQ záplava

3.3.1 Testovací scenár útoku REQ záplava

Scenár testovania útoku REQ záplavy je zložený z dvoch režimov.

Prvý základný režim využíva len jeden škodlivý uzol, ktorý ale reaguje na všetky REQ správy. Obmedzený by sme mali byť najmä maximálnym povoleným vygenerovaným počtom REQ správ za daný čas.

Druhý režim využíva dva spolupracujúce uzly, ktoré zaplavujú sieť vyššie popísaným spôsobom. Celkovo by mali spolu reagovať na rovnaký počet správ ako jeden samostatný uzol v prvom režime, ale ako už bolo spomenuté vďaka častým obmedzeniam na počet vygenerovaných REQ správ, budeme môcť teoreticky vygenerovať dva krát viac správ za daný časový úsek ako v prvom režime. Uzly by sa mali od seba nachádzať vo vzdialenosti, ktorá neumožňuje ich priamu komunikáciu. Replikácia druhým uzlom tak nastane, len ak bude existovať cesta medzi týmito uzlami, ale celkové pokrytie siete bude väčšie ako v prípade prvého režimu.

3.3.2 Sledované parametre REQ záplavy

Dva definované režimy budú porovnané z hľadiska hodnôt nasledovných parametrov.

- Celková veľkosť prenesených dát
- Počet vytvorených REQ správ škodlivými uzlami
- Priemerná úspešnosť výmeny správ
- Celkový počet inicializovaných procesov hľadania cesty
- Počet prijatých REQ správ škodlivými uzlami

3.4 RER záplava

Na zaplavenie siete môžeme taktiež využiť aj správy slúžiace na notifikáciu zmeny topológie, neschopnosti doručenia správy a podobne. Takéto správy označíme ako RER (Route Error). Keď uzly príjmu takúto správu, tak najčastejšie inicializujú opätovný mechanizmus hľadania najkratšej cesty. Pri pravidelnom generovaní týchto správ škodlivým uzlom môžeme dosiahnuť nedostupnosť uzlov v sieti a takisto aj značné spotrebovanie dostupnej šírky pásma z dôvodu neustáleho zaplavovania siete správami na hľadanie cesty.

Generovať tieto správy by sme mali len pre uzly, ktoré sa v sieti naozaj nachádzajú. Inak síce spotrebujeme značnú časť dostupnej šírky pásma, ale správy ako také nebudú mať žiadny ďalší efekt. Takýto spôsob by bol navyše veľmi ľahko odhaliteľný. Preto si budeme ukladať už známe uzly v sieti z odchytených správ. Dozvieme sa tak nielen o susedných uzloch, ale zo správ REQ a RER aj o ostatných uzloch v sieti.

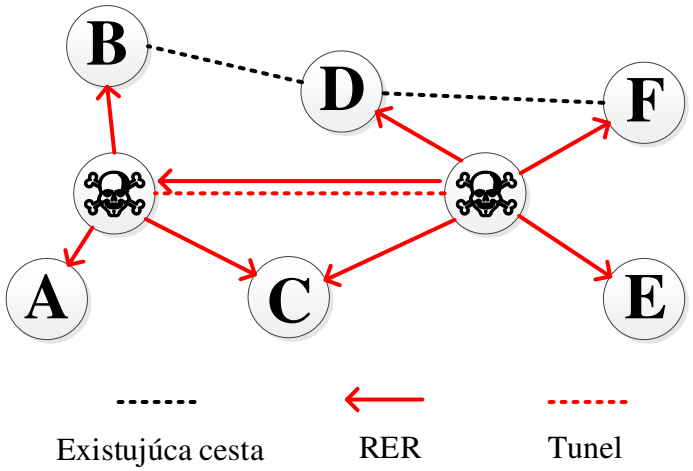
V určitých časových intervaloch bude škodlivý uzol vysielať RER správy, ktoré spôsobia vymazanie aktívnych ciest zo smerovacích tabuliek ostatných bežných uzlov. Pre väčšie pokrytie siete ďalej navrhujeme použitie viacerých uzlov, ktoré budú prepojené pomocou rýchlych fyzických spojení. Cez tieto spojenia si môžu škodlivé uzly vymieňať informácie o doteraz objavených uzlov v sieti, alebo aj použiť na odoslanie vygenerovaných RER správ na inom mieste v sieti.

Vhodné je použitie uzlov v anonymnom režime, pri ktorom nebudú viditeľné ostatným uzlov v sieti. Pri generovaní RER správ sa budú navonok javiť ako iný štandardný uzol v sieti (Prevezme jeho identitu).

Konkrétna implementácia útoku závisí od spôsobu fungovania RER správ daného protokolu. Napríklad pri protokole AODV analyzovanom v časti 1.2.2 sa v RER správe posiela zoznam nedostupných uzlov. Uzol následne skontroluje, či cestu do týchto uzlov cez uzol od ktorého RER správu prijal pozná. Ak áno, vymaže si dané cesty zo smerovacej tabuľky a susedným uzlom oznámi nedostupnosť len tých ciest, ktoré cez neho viedli. Pre kompletné vymazanie úplne všetkých ciest je nutné vygenerovať RER správy za všetky uzly v sieti.

Na obr. 3.4 môžeme vidieť použitie RER záplavy s dvomi spolupracujúcimi uzlami. Predpokladáme existenciu cesty z uzla B do uzla F cez uzol D a opačne. Taktiež predpokladáme, že sa škodlivé uzly naučili o prítomnosti všetkých ostatných uzlov v sieti. Obrázok zobrazuje len jednu z vygenerovaných RER správ, konkrétne tú ktorej zdrojvá

adresa bola nastavená na adresu uzla D.. Táto správa bude obsahovať zoznam všetkých ostatných známych uzlov. Rozšírenie správy v sieti je znázornené na obrázku. Uzol F po prijatí tejto správy zistí, že pozná cestu do uzla nachádzajúceho sa v zozname (Uzol B), cez uzol od ktorého správa prichádza (Uzol D). Preto túto cestu vymaže zo smerovacej tabuľke a ostatným uzlom odošle RER správu, ktorá ich informuje, že cez uzol F sa už nedá dostať do uzla B. Táto správa nie je kvôli prehľadnosti v obrázku znázornená. Analogicky uzol B po prijatí tejto správy zistí, že pozná cestu do uzla nachádzajúceho sa v zozname (Uzol F), cez uzol od ktorého správa prichádza (Uzol D). Následne taktiež zopakuje rovnaký postup ako v prípade uzla F. Docielili sme tak vymazanie cesty medzi uzlom B a F, ktoré pre ďalšiu vzájomnú komunikáciu budú musieť zopakovať proces hľadania cesty odznova.



Obr. 3.4 RER záplava

3.4.1 Testovací scenár útoku RER záplavy

Testovací scenár pozostáva z dvoch režimov RER záplavy.

Prvý scenár nepodporuje komunikáciu medzi škodlivými uzlami. Navzájom sa neučia existujúce uzly v sieti ani nepreposielajú RER správy druhého uzla.

Druhý scenár podporuje komunikáciu medzi škodlivými uzlami. Navzájom sa učia existujúce uzly v sieti a preposielajú správy druhého RER uzla. Učenie známych uzlov v sieti by tak malo byť rýchlejšie a takisto aj efektivita útoku vďaka vzájomnému preposielaniu RER správ by mala byť vyššia.

3.4.2 Sledované parametre RER záplavy

Oba definované režimy budú porovnávané z hľadiska nasledovných parametrov.

- Čas, za ktorý sa oba uzly naučili všetky existujúce uzly v sieti

- Počet RER správ odoslaných škodlivými uzlami
- Počet RER správ odoslaných v sieti
- Počet zduplikovaných bežných RER správ
- Priemerná úspešnosť doručenia správ v sieti
- Počet inicializácii procesu hľadania cesty v sieti
- Celková veľkosť prenesených dát v sieti
- Počet vymazaných ciest zo smerovacích tabuliek

3.5 Čierna červia diera

Spojením čiernej a červej diery môžeme vytvoriť veľmi silný útok, ktorý môže vykompenzovať nedokonalosti útoku čierna diera. Tento je neefektívny v prípade, ak pole počet skokov (alebo podobné pole slúžiace na určenie najkratšej cesty) nie je možné upraviť, z dôvodov rôznych zabezpečení, alebo sa v protokole nenachádza vôbec. Príkladom takéhoto protokolu je napríklad ARAN zanalyzovaný v časti 1.2.5.

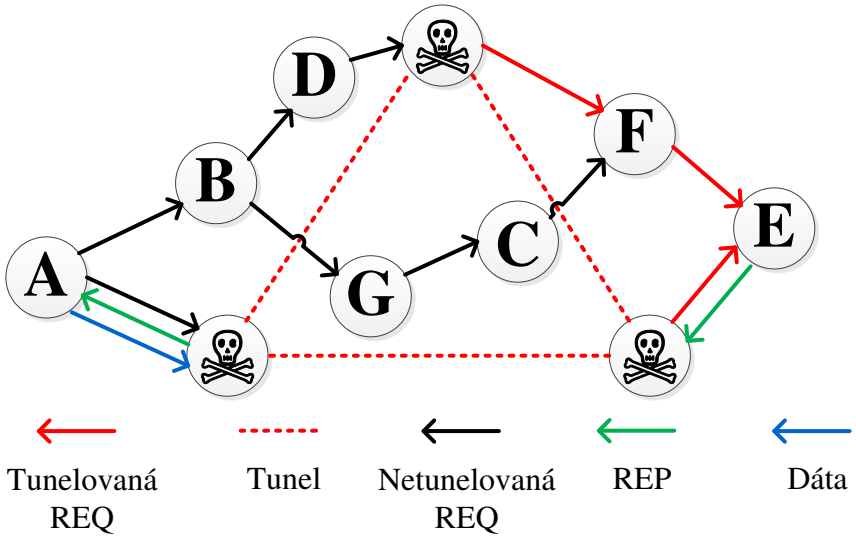
Protokoly nevyužívajúce polia ako počet skokov na určenie najkratšej cesty nevedia zaručiť najkratšiu cestu do cieľa a výsledkom hľadania cesty je spravidla najrýchlejšia cesta. Pomocou červej diery môžeme vytvoriť zdanlivú najkratšiu cestu medzi uzlami a po jej vytvorení prepnúť uzly do režimu čiernej alebo sivej diery. Prekonáme tak obmedzenie útoku čiernej diery.

Pre čo najťažšiu detekciu útoku použijeme červie diery v skrytom režime. Uzly tak nebudú viditeľné ostatným uzlom v sieti. Možné je použiť dva a viacero takýchto spolupracujúcich uzlov pre čo najväčšie pokrytie siete. Tunelované budú iba správy slúžiace na nájdenie najkratšej cesty a odpovede na tieto správy. Pri opätovnom iniciovaní mechanizmu hľadania cesty, z dôvodu nedoručenia bežných správ, bude cesta cez červiu dieru vždy najkratšia, čo spôsobí opätovné zahadzovanie bežných správ.

Z dôvodu ťažšieho odhalenia môže byť režim čiernej diery nahradený režimom sivej diery, ktorý bude zahadzovať iba správy od určitých uzlov, alebo ich bude zahadzovať len náhodné správy. Pri použití tohto útoku odporúčame použitie práve náhodného zahadzovania správ.

Na obr. 3.5 môžeme vidieť použitie čiernej červej diery s tromi spolupracujúcimi škodlivými uzlami. Tieto uzly pokrývajú veľkú časť siete. Predpokladáme použitie protokolu, ktorý neobsahuje pole zabezpečujúce nájdenie najkratšej cesty, ktoré by bolo možné modifikovať. Uzol A sa pokúša nájsť cestu do uzla E a pošle broadcastom správu REQ. Táto správa je prijatá uzlom B a škodlivým uzlom, ktorý túto správu odošle tunelom

obom spolupracujúcim uzlom. Tieto uzly správu odošlú ďalej štandardným spôsobom. Uzol E tak dostane ako prvú správu, ktorá bola tunelovaná a táto cesta sa teda použije na komunikáciu medzi uzlami A a E. Odpoveď na hľadanie cesty je taktiež tunelovaná naspäť. Po vytvorení cesty tak škodlivý uzol môže náhodne zahadzovať dáta posielané z uzla A do uzla E a spôsobiť týmto uzlom rôzne problémy.



Obr. 3.5 Čierna červia diera

3.5.1 Testovací scenár útoku čiernej červej diery

Testovací scenár útoku čiernej červej diery sa skladá z piatich rôznych režimov.

Prvý režim využíva iba červie diery, pomocou ktorých vytvorí najkratšie cesty cez škodlivé uzly bez nutnej modifikácie správ procesu hľadania cesty. Preferovaný je anonymný režim, pri ktorom nie sú škodlivé uzly viditeľné ostatným uzlom v sieti.

Uzly v druhom režime fungujú len ako bežné čierne diery. Generujú správy, pomocou ktorých sa snažia presvedčiť ostatné uzly v sieti, že cez nich prechádza najkratšia cesta do hľadaného cieľového uzla. Po ovplyvnení smerovacieho procesu začnú všetky správy zahadzovať. Komunikácia medzi zdrojovým a cieľovým uzlom tak nebude možná.

Tretí režim je obmenou druhého režimu. Namiesto čiernych dier sú použité sivé diery, ktoré po ovplyvnení smerovacieho procesu zahadzujú prichádzajúce správy náhodne. Niektoré správy tak budú doručené a niektoré nie. Výhodou môže byť zložitejšia detekcia a takisto aj nevyvolanie procesu opätovného vyhľadávania cesty, ktoré by už nemuselo byť opäť ovplyvnené sivou dierou.

Štvrtý režim využíva už vyššie popísanú čiernu červiu dieru. Pomocou spojenia oboch prístupov by mali byť využité silné stránky spojených útokov.

Posledný piaty režim upravuje štvrtý režim o použitie sivej diery. Rozdiel medzi štvrtým a piatym režimom by mal byť podobný ako v prípade druhého a tretieho režimu.

3.5.2 Sledované parametre útoku čiernej červej diery

Definované režimy budú porovnávané z hľadiska nasledovných parametrov.

- Priemerná úspešnosť doručenia správ v sieti
- Priemerný počet skokov do cieľového uzla
- Počet ovplyvnených procesov hľadania ciest
- Počet zahodených správ
- Celková veľkosť prenesených dát v sieti

3.6 Návrh zabezpečení

V tejto časti predstavíme možnosti ochrany pred navrhnutými útokmi. Zmienime existujúce riešenia, ale navrhneme aj naše vlastné koncepty na detekciu pridania oneskorenia a červej diery.

3.6.1 Zabezpečenie pridania oneskorenia

Existujúce zabezpečenia proti oneskoreniu sú založené najmä na vyhodnocovaní rôznych časov. V [22] autor deteguje škodlivý uzol pomocou vytvorenia TCP spojení so všetkými uzlami. Vypočítaný je čas, ktorý uplynie od odoslania SYN správy po prijatie odpovedi ACK. Ak je tento čas väčší ako definovaná prahová hodnota, tak je uzol označený za škodlivý, uložený do pamäte uzlu a následne vylúčený pri ďalších procesoch hľadania cesty.

3.6.1.1 Návrh zabezpečenia pridania oneskorenia

Pre detekciu pridania oneskorenia pomocou jedného, alebo viacerých uzlov, sme navrhli vlastný koncept, ktorý môže byť použitý na návrh finálneho riešenia detekcie a odolnosti proti tomuto útoku. Základom je využitie protokolu, ktorý v smerovacích tabuľkách ukladá zoznam uzlov na celej ceste od zdroja, až do cieľa. Príkladom takéhoto protokolu je zanalyzovaný protokol DSR v časti 1.2.1.

Princípom je využitie týchto známych ciest na identifikáciu škodlivého uzla, ktorý môže byť vylúčený z ďalšieho procesu hľadania cesty na základe informácií od uzlov nenachádzajúcich sa na podozrivej ceste. Uzol môže označiť cestu za podozrivú z rôznych dôvodov. V našich útokoch nepridávame oneskorenie správam smerovacieho protokolu, ale len dátovým správam. Uzol tak môže definovať priemerné oneskorenie na základe časov potrebných na prijatie správ REP. Za podozrivú cestu tak môžeme považovať tú, na

ktorej dosahujeme oneskorenie väčšie ako *k* . *priemer* pričom konštantu *k* môžeme zvoliť na základe parametrov našej siete.

Proces nájdenia škodlivého uzla je potom nasledovný. Uzol pošle všetkým známym uzlom nenachádzajúcim sa na podozrivej ceste špeciálnu správu RVALREQ (Route Validation Request), ktorej navrhovaný formát môžeme vidieť v Tab. 3.1. Jednotlivé polia správy sú samo vysvetľujúce.

Tab. 3.1 Formát správy RVALREQ

<table>
<tr><td colspan="32">0 1 2 3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1</td></tr>
<tr><td colspan="32">Zdrojový uzol</td></tr>
<tr><td colspan="8">Dĺžka škodlivej cesty</td><td colspan="24">Rezervované</td></tr>
<tr><td colspan="32">Adresa uzla 1 na škodlivej ceste</td></tr>
<tr><td colspan="32">Adresa uzla 2 na škodlivej ceste</td></tr>
<tr><td colspan="32">...</td></tr>
<tr><td colspan="32">Adresa uzla n na škodlivej ceste</td></tr>
</table>

Uzly ako odpoveď odošlú správu RVALREP (Route Validation Reply), v ktorej sa nachádza zoznam overených uzlov. Za overený uzol považujeme ten uzol, cez ktorý prebieha komunikácia bez výraznej odchýlky oneskorenia a bol súčasťou zoznamu v správe RVALREQ. Uzly môžu túto správu vygenerovať na základe oneskorení aktuálnych ciest, alebo môžu vyskúšať oneskorenie pre všetky uzly z prijatého zoznamu. Druhá možnosť zvýši presnosť metódy, ale pri dlhých zoznamoch podozrivých ciest môže spôsobiť výrazné vyťaženie zdrojov siete. Navrhovaný formát správy RVALREP môžeme vidieť v Tab. 3.2.

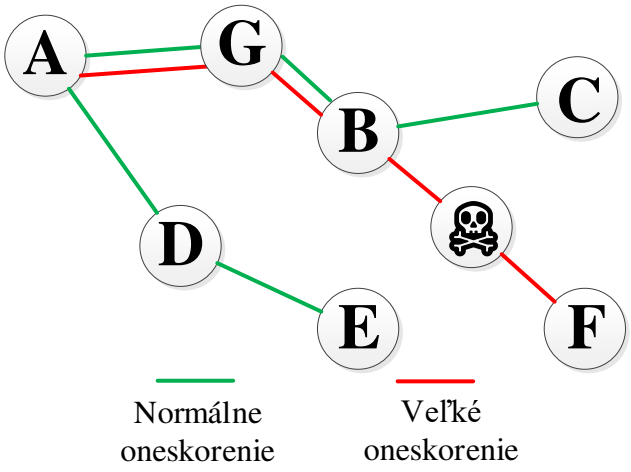
Tab. 3.2 Formát správy RVALREP

<table>
<tr><td colspan="32">0 1 2 3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1</td></tr>
<tr><td colspan="32">Zdrojový uzol</td></tr>
<tr><td colspan="8">Počet uzlov</td><td colspan="24">Rezervované</td></tr>
<tr><td colspan="32">Adresa overeného uzla 1</td></tr>
<tr><td colspan="32">Adresa overeného uzla 2</td></tr>
<tr><td colspan="32">...</td></tr>
<tr><td colspan="32">Adresa overeného uzla n</td></tr>
</table>

Pôvodný uzol po prijatí týchto správ môže použiť viacero metód na identifikáciu škodlivých uzlov. Pri základom princípe jednoducho odčíta množinu uzlov na podozrivej ceste od množiny uzlov prijatých v správach RVALREP. Týmto spôsobom nám ostanú najpravdepodobnejšie uzly, ktoré spôsobujú zvýšené oneskorenie. Ďalšie možnosti môžu zohľadniť počty dopytovaných uzlov a overených uzlov.

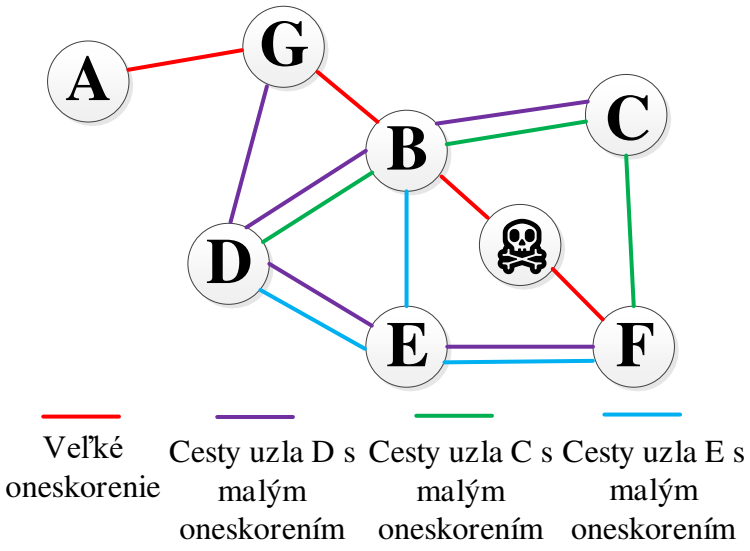
Pri ďalších procesoch hľadania cesty môžeme rozšíriť pôvodné protokoly o pole obsahujúce zoznam nedôveryhodných uzlov. Ak uzol príjme smerovaciu správu od uzla v tomto zozname, tak správu zahodí a tým pádom sa cesty cez podozrivé uzly nevytvoria. Nutnosťou je zabezpečenie všetkých správ smerovacieho protokolu a aj našich navrhnutých správ proti zmene jednotlivých polí. Príklad, ako to docieliť, sme mohli vidieť v analýze zabezpečených smerovacích protokolov. Z hľadiska rozsahu práce nebolo toto navrhnuté riešenie implementované. Vyžadovalo by si to špeciálne testovacie scenáre odsimulované pre rôzne možnosti výberu škodlivých uzlov na základe ktorých by bolo možné porovnanie jednotlivých prístupov.

Príklad detekcie škodlivého uzla môžeme vidieť na Obr. 3.6 a Obr. 3.7. Uzol A na Obr. 3.6 deteguje pomalé správy na ceste k uzlu F a tak odošle správu RVALREQ všetkým známym uzlom, ktoré neležia na podozrivej ceste (C, D, E). Táto správa obsahuje zoznam uzlov podozrivej cesty (G, B, škodlivý uzol, F).



Obr. 3.6 Identifikácia podozrivých ciest

Všetky uzly, ktoré túto správu dostanú, odošlú naspať zoznam tých uzlov z podozrivej cesty, cez ktoré prijímajú správy bez výraznej odchýlky v oneskorení. Na základe Obr. 3.7 môžeme povedať, že uzol D odošle zoznam uzlov G,B a F. Uzly C a E odošlú zoznam uzlov B a F.



Obr. 3.7 Nájdenie škodlivého uzla

Uzol A tak od pôvodnej množiny odčíta prijaté zoznamy uzlov a nájde škodlivý uzol. Pri ďalšom procese hľadania cesty tak v správe tento uzol explicitne označí a vytvorí sa nová cesta do uzla F bez pridaného oneskorenia. Môžeme si všimnúť, že ak by uzol D nemal vytvorenú cestu s uzlom G, tak by bol aj tento uzol označený za škodlivý a takisto by bol vylúčený z procesu hľadania cesty. Metóda tak v tomto prípade nie je dokonalá. Vylepšenie presnosti by sme mohli zvýšiť už spomínanými kontrolami všetkými uzlami na všetky uzly zo zoznamu podozrivých uzlov. Preverené by tak boli všetky uzly tohto zoznamu, ale za cenu zvýšenej záťaže na sieť.

3.6.2 Zabezpečenie REQ a RER záplav

Základom ochrany proti týmto útokom je zabezpečenie nemennosti polí smerovacích správ ako sú sekvenčné čísla, zdrojové a cieľové uzly. Využiť na to môžeme rôzne kryptografické mechanizmy od asymetrických až po symetrické. Príklady takýchto zabezpečení sme popísali pri zabezpečených smerovacích protokoloch v častiach 1.2.3, 1.2.4 a 1.2.5. Účinnosť zabezpečenia protokolu SAODV budeme môcť vidieť aj v simuláciách v časti 5.

Kritická je samozrejme ochrana šifrovacích kľúčov, ktorú môžeme zabezpečiť napríklad pomocou centrálnej autority ako v protokole ARAN. Nevýhodou je ale následná závislosť uzlov v sieti, ktoré musia byť schopné s autoritou komunikovať a overovať platnosti jednotlivých certifikátov. Charakteristika decentralizácie MANET sietí je tak čiastočne porušená.

3.6.3 Zabezpečenie červej diery

Ochrániť sa pred naším navrhnutým útokom čiernej červej diery je možné pomocou samotnej ochrany proti červej diere. Tá je ale veľmi náročná, pretože uzly, už ako bolo spomenuté, môžu pracovať v odhalenom a skrytom režime. Pri odhalenom režime sú uzly viditeľné ostatným uzlom v sieti, no pri skrytom režime nie sú viditeľné ostatným uzlom v sieti. Dostupných článkov zaoberajúcich sa ochranou a detekciou červích dier je viacero, pričom niektoré z metód sú zhrnuté v [23]. Dokonalá ochrana ale neexistuje a preto pri každej narazíme na prípady falošnej detekcie útoku.

Všeobecne môžeme mechanizmus detekcie čiernych dier rozdeliť do dvoch kategórií: validácia susedov a detekcia na celej ceste. Pri validácii susedov je možné pridať do správ ďalšie informácie, pomocou ktorých je možné limitovať maximálnu vysielaciu vzdialenosť. Tento spôsob ale nie je efektívny v prípade skrytého režimu.

Detekcie na celej ceste typicky merajú a analyzujú sieťové uzly a vlastnosti ciest, ako frekvencia výskytu uzlov v ceste, geografická pozícia uzlov, počty skokov, čas odpovedí a ďalšie. Frekvencia výskytu uzlov v cestách sa ukázala ako efektívna v prípade odhaleného režimu a málo dynamických sietí. V prípade vysoko dynamických sietí môže byť zložité udržiavať frekvencie výskytu jednotlivých uzlov. Z tohto dôvodu, sa na tento účel môže použiť špeciálny centrálny uzol.

Myšlienka detekcie na základe geografickej polohy je veľmi zaujímavá, pretože jednoducho rieši problém zistenia vzdialeností medzi uzlami. Problémom je nutnosť prítomnosti GPS na každom zo zariadení, čo značne obmedzuje jej využitie v MANET sieťach s nízkonákladovými zariadeniami.

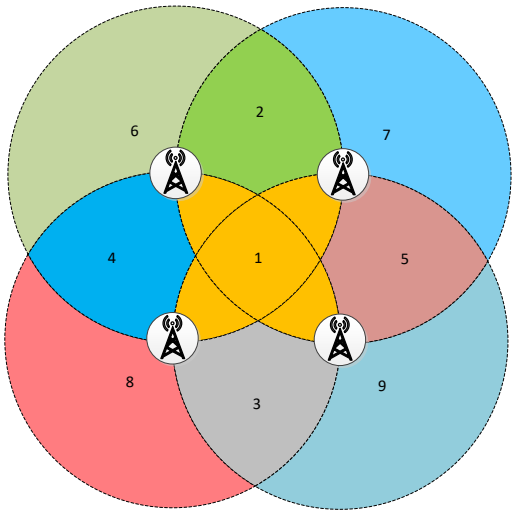
Ďalšou možnosťou je využitie predpokladu, že ak priemerná doba odozvy na jeden skok je väčšia ako definovaná hranica, tak ide o cestu s červou dierou. Tento spôsob má potenciál odhaliť oba režimy červej diery. Nevýhodou ale je, že v praxi sa v týchto sieťach vyskytujú nepredvídateľné toky dát, ktoré môžu tieto metriky ľahko ovplyvniť.

Ďalšie rôzne metódy majú taktiež svoje určité výhody a nevýhody. Žiadne z riešení nie je dokonalé a preto by mali byť zvolené zabezpečenia vybrané na základe vlastností danej siete.

3.6.3.1 Návrh zabezpečenia červej diery

Pri návrhu konceptu tohto zabezpečenia sme sa inšpirovali myšlienkou využitia polohy jednotlivých zariadení. Nevyužijeme však GPS, ale pevné stanice pomocou ktorých budeme môcť zistiť v akej zóne sa konkrétny uzol nachádza. Stanice by mali byť od seba

v takej vzdialenosti, ktorá umožňuje ich vzájomnú komunikáciu. Uzly v dosahu stanice tak budeme vedieť pomocou špeciálnych správ jednoznačne zaradiť do konkrétnych zón ako na Obr. 3.8.



Obr. 3.8 Definovanie zón pri štyroch staniciach

Pre zistenie minimálneho počtu skokov zavedieme dve jednoduché nové správy RMHREQ (Route Min Hop Request) a RMHREP (Route Min Hop Reply). Požiadavku odosiela bežný uzol v sieti stanici po prijatí odpovede na proces hľadania cesty a obsahuje adresu cieľového uzla cesty. Ďalší obsah správy ponechávame na návrh implementujúceho. Odporúčame ale zabezpečiť všetky správy proti sfalšovaniu pomocou rôznych kryptografických riešení.

Stanica po prijatí správy RMHREQ kontaktuje ostatné stanice a zistí vzdialenosť medzi zdrojovým uzlom správy a cieľovým uzlom nachádzajúcim sa v správe. Vzdialenosť samozrejme nemôže byť zistená presne, ale pomocou definovania minimálneho počtu skokov medzi zónami môžeme získať odhad minimálneho počtu skokov medzi požadovanými uzlami. Táto hodnota je odoslaná dopytujúcemu uzlu v správe RMHREP.

Uzol následne skontroluje, či nie je táto hodnota menšia ako v prijatej správe. Ak áno, inicializuje ďalší proces hľadania cesty v ktorom pomocou rozširujúceho poľa nastaví minimálny počet skokov na hodnotu o 1 vyššiu ako bola prijatá v predchádzajúcej odpovedi požiadavky na cestu. Cieľový uzol tak zahodí požiadavku na cestu v prípade, že počet skokov je menší ako hodnota rozširujúceho poľa. Vytvorí sa tak dlhšia cesta, ktorá by už nemala viesť cez červiu dieru.

Navrhovaný koncept má potenciál odhaliť odhalený aj skrytý režim červích dier, no nie je dokonalý, aj po predĺžení cesty môže byť vytvorená cesta súčasťou červej diery. Dôležité bude vyladenie metódy odhadu minimálneho počtu skokov medzi uzlami.

3.7 Zhodnotenie návrhu

V tejto časti sme navrhli štyri nové verzie útokov, pričom niektoré z nich môžeme vykonať v rôznych režimoch. Princípy navrhnutých útokov boli podrobne popísané a ich funkcionalita bola znázornená na obrázkoch. Spolu s týmito útokmi sme navrhli aj testovacie scenáre. Tieto scenáre sa skladajú z viacerých režimov útokov, ktoré majú podobný dopad na sieť.

Pomocou týchto scenárov budeme môcť dôkladne otestovať jednotlivé zraniteľnosti smerovacích protokolov. Vyhodnotenie odolnosti a účinnosti jednotlivých útokov v scenári budeme posudzovať pomocou definovaných sledovaných parametrov.

Na základe predchádzajúcej analýzy zabezpečených protokolov sme navrhli koncepty bezpečnostných riešení pre útoky pridania oneskorenia a červej diery. Z dôvodu rozsahu a zamerania práce neboli tieto koncepty ďalej rozpracované ani implementované.

4 Implementácia

Pred implementáciou útočných scenárov bolo najskôr nutné vybrať implementačné prostredie a implementovať v ňom príslušné analyzované protokoly. Po implementácií týchto protokolov boli vytvorené jednotlivé škodlivé uzly, ktoré vznikli modifikáciou základných uzlov jednotlivých smerovacích protokolov.

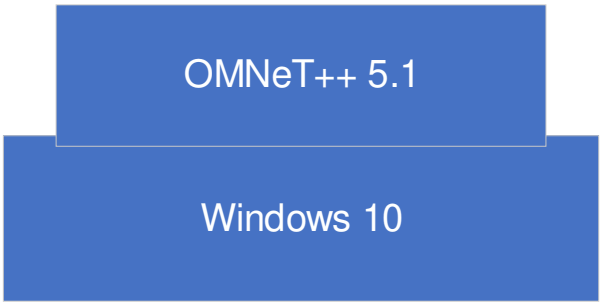
Ako implementačné prostredie sme si vybrali OMNeT++. Tento simulátor je rozšíriteľná, modulárna, komponentovo založená C++ knižnica určená primárne na simuláciu sietí rôznych druhov.

4.1 Simulačné prostredie

Doménové špecifické funkcionality konkrétnych druhov sietí sú v OMNeT++ implementované pomocou rôznych rámcov (frameworkov). Najpoužívanejším z nich je INET, v ktorom je implementovaný aj protokol AODV. Rozšírenie tohto frameworku o vlastné riešenie je zložité z dôvodu vzájomných závislostí modulov. INET býva navyše často a razantne menený, takže by naše riešenie nemohlo byť využité v novších alebo starších verziách. Rozhodli sme sa preto nevyužiť tento rámec a správanie protokolov implementovať samostatne, keďže s použitím rámca by bolo aj tak nutné implementovať všetky požadované protokoly, okrem protokolu AODV. Výstupom nášho projektu tak budú nielen testovacie scenáre a ich vyhodnotenie ale aj implementácie zvolených smerovacích protokolov.

Implementácia samotných protokolov pre otestovanie navrhnutých scenárov nevyžaduje implementáciu všetkých vrstiev modelu RM OSI. Preto sú tieto implementácie zamerané hlavne na správanie protokolov so zjednodušením ostatných vrstiev len na nevyhnutné funkcionality.

Pri implementácii a testovaní bol použitý operačný systém Windows 10, ktorý nevyžadoval žiadne špeciálne úpravy simulátora, ktorý bol v čase implementácie vo verzii 5.1. Grafické znázornenie prostredia je na obr. 4.1.



Obr. 4.1 Implementačné prostredie

Jednotlivé súbory v OMNeT++ sú odlíšené pomocou ich koncoviek. Ich význam je nasledovný:

- .cc - popisujú funkcionalitu modulu v jazyku C++
- .msg - definujú správy a ich polia. Na základe týchto súborov dokáže OMNeT++ vygenerovať príslušné .cc súbory pre jednotlivé správy.
- .ned - definujú moduly, ich parametre, vstupy a výstupy. Takisto definujú aj celé simulačné siete, ktoré môžu byť vytvorené napevno, alebo parametricky
- .ini - nastavujú hodnoty parametrov simulácie, siete a modulov definovaných v .ned
- .sca, .vec - výstupné súbory so štatistikami simulácie

4.2 Základná implementácia protokolov

Všetky implementované protokoly majú spoločné parametre, ktoré definujú správanie uzlov v sieti. Tieto parametre musia byť pri testovaní rovnaké u všetkých protokolov, aby nedošlo ku skresleniu výsledkov. Medzi tieto parametre uzlov patria:

- pings_times_to_try - počet pokusov opätovného odoslania ping správ, pri ich zlyhaní
- transmission_distance - vzdialenosť do akej je uzol schopný odoslať správu
- ping_retransmit_wait - čas, po ktorého uplynutí sa považuje príchod odpovede na ping za zlyhanie
- transmit_delay - čas, ktorý je nutný na odoslanie správy inému uzlu
- mobility_delay - čas definujúci ako často sa má uzol pohnúť o 1 jednotku
- position_x - aktuálna súradnica x daného uzla
- position_y - aktuálna súradnica y daného uzla

Okrem parametrov uzlov sú definované aj nasledovné parametre siete:

- max_x_position - maximálna povolená súradnica x daného uzla
- max_y_position - maximálna povolená súradnica y daného uzla
- numNodes - počet uzlov v sieti

Konkrétne použité hodnoty parametrov sú uvedené v časti 5.

4.2.1 Odosielanie správ

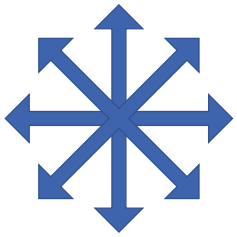
Z dôvodu nepoužitia INET rámca boli implementované aj vlastné funkcie na odosielanie správ, ktoré musia zohľadňovať bezdrôtový charakter siete a obmedzenie na maximálnu vzdialenosť odoslania správ. Na tento účel slúžia 2 nasledovné metódy:

- sendUnicast (cMessage *msg, const char* destination) - odošle správu konkrétnemu uzlu
- sendToAllNodesInRange (cMessage *msg) - odošle správu všetkým uzlom v dosahu

V bezdrôtových sieťach zachytia správu všetky uzly, ktoré sú v dosahu od vysielajúceho uzla. Unicastová správá je následne na uzloch ktorým nie je určená, zahodená na základe MAC adries. Toto správanie sme zjednodušili na priame odoslanie pomocou metódy sendUnicast. Pre bežný broadcast sa používa metóda sendToAllNodesInRange.

4.2.2 Mobilita uzlov

Ďalšou dôležitou vlastnosťou MANET sietí je mobilita uzlov. Uzly sa môžu voľne pohybovať a tým spôsobovať zmeny topológie. Uzly sa preto pohybujú náhodne v rámci vymedzeného priestoru definovaným parametrami max_x_position a max_y_position. Začiatočné parametre pozície position_x a position_y každého uzla sú vygenerované náhodne na začiatku každej simulácie. Takisto je vygenerovaný náhodný smer, ktorým sa bude uzol pohybovať. Možné smery sú zobrazené na obr. 4.2.



Obr. 4.2 Povolené smery pohybu uzla

Uzol zachováva náhodne vygenerovaný smer pohybu dovtedy, kým sa nepriblíži k hraniciam povoleného priestoru. Vtedy je vygenerovaný nový smer vzhľadom na dosiahnutú hranicu priestoru. Teda ak sa uzol priblíži k hornej hranici, tak následne zvolený náhodný pohyb bude z množiny vľavo dole, dole a vpravo dole. Tento princíp bol zvolený z toho dôvodu, pretože pri náhodnom výbere smeru pri každom pohybe sa uzol, kvôli rovnomernému rozdeleniu náhodne generovaných hodnôt, pohyboval len v blízkom okolí štartovacej pozície.

4.3 AODV

Funkcionalita protokolu AODV bola implementovaná na základe analýzy z časti 1.2.2. Medzi rozširujúce parametre tohto protokolu patria:

- rt_table_lifetime - maximálna životnosť ciest v smerovacej tabuľke
- rt_table_lifetime_update_interval - ako často je aktualizovaná životnosť záznamov v smerovacej tabuľke

4.3.1 Použité správy

Protokol využíva 7 rôznych správ, z ktorých sú 3 vlastné. Je nutné dodať, že veľkosti nevlastných správ sú definované na základe reálnych správ protokolu a nie na základe polí nachádzajúcich sa v správe. Niektoré z týchto polí môžu byť len pomocné pre uľahčenie implementácie. Vlastné správy posiela uzol sám sebe. Týmto spôsobom je možné simulovať rôzne oneskorenia, alebo časovače. Medzi vlastné správy patria:

- Self_ping_retransmit - jediná vlastná správa s parametrom. Slúži na opätovné odoslanie ping správy, pokiaľ nebola do definovaného času prijatá odpoveď už na odoslaný ping. Parameter obsahuje ping správu, ktorá má byť odoslaná.
- SELF_MOBILITY - spôsobí pohnutie uzla v rámci vymedzeného priestoru o 1 jednotku a naplánovanie odoslania ďalšej rovnakej vlastnej správy s definovaným oneskorením v parametri mobility_delay
- RT_LIFETIME - zabezpečuje aktualizovanie životnosti záznamov v smerovacej tabuľke a taktiež naplánovanie opätovného odoslania rovnakej vlastnej správy na základe parametra rt_table_lifetime_update_interval.

4.3.1.1 Správa RREQ

Polia správy RREQ o veľkosti 118 B sú nasledovné:

1. source_MAC - zdrojová MAC adresa

2. destination_MAC - cieľová MAC adresa
3. source - zdrojová IP adresa
4. destination - cieľová IP adresa
5. type - typ nastavený na hodnotu 1
6. flags - príznaky
7. reserved - rezervované bity nastavené na hodnotu 0
8. hop_count - počet skokov
9. rreq_id - identifikátor RREQ správy
10. dst_ip - cieľová IP adresa hľadania cesty
11. dst_seq - cieľové sekvenčné čislo
12. src_ip - zdrojová IP adresa uzla, ktorý inicializoval hľadanie cesty
13. src_seq - zdrojové sekvenčné číslo
14. data_size - veľkosť dát nastavená na hodnotu 118 B

4.3.1.2 Správa RREP

Polia správy RREP o veľkosti 114 B sú nasledovné:

1. - 4. - Rovnaké ako pri správe RREQ
5. type - typ nastavený na hodnotu 2
6. - 12. - Rovnaké ako pri správe RREQ s vynechaním rreq_id
13. lifetime - životnosť cesty
14. data_size - veľkosť dát nastavená na hodnotu 114 B

4.3.1.3 Správa RERR

Polia správy RRER o premenlivej veľkosti sú nasledovné:

1. source_MAC - zdrojová MAC adresa
2. type - typ nastavený na hodnotu 3
3. dstcount - počet prvkov polí 4 a 5
4. dst_IParray[] - pole IP adries, do ktorých už nie je cesta dostupná
5. dst_Seqarray[] - pole sekvenčných čísel, do ktorých už nie je cesta dostupná
6. data_size - základná veľkosť dát nastavená na hodnotu 114 B

4.3.1.4 Správa PING

Polia správy PING o veľkosti 94 B sú nasledovné:

1. type - 1 pre požiadavku, 2 pre odpoveď
2. id - náhodne vygenerované číslo pre rozlíšenie jednotlivých odpovedí
3. src_ip - zdrojová IP adresa

4. dst_ip - cieľová IP adresa
5. create_time - čas vytvorenia požiadavky. Slúži na výpočet oneskorenia odpovede.
6. payload - veľkosť payloadu. Prednastavený na 0.
7. data_size - veľkosť dát nastavená na hodnotu 94 B

4.4 SAODV

Funkcionalita protokolu SAODV bola implementovaná na základe analýzy z časti 1.2.3. Predpokladáme, že všetky uzly majú pravé verejné kľúče všetkých ostatných uzlov v sieti pretože cieľom našich simulácií je otestovanie zabezpečenia protokolu a nie výmeny kľúčov. Medzi rozširujúce parametre tohto protokolu patria:

- rt_table_lifetime - maximálna životnosť ciest v smerovacej tabuľke
- rt_table_lifetime_update_interval - ako často je aktualizovaná životnosť záznamov v smerovacej tabuľke
- sha256_verification - čas potrebný na výpočet výsledku hašovacieho algoritmu vynásobený číslom 253 (Pri overovaní sa viacnásobne hašuje prijatý reťazec)
- rsa_sign - čas potrebný na vytvorenie RSA podpisu
- rsa_verify - čas potrebný na overenie RSA podpisu

Časové hodnoty boli odvodené z výsledkov nameraných na Raspberry Pi 3 model B pomocou programu openssl. Pri RSA bol použitý priemer časov pre kľúče o dĺžke 1024 a 2048 bitov. Konkrétne namerané hodnoty môžeme vidieť v tabuľkách Tab. 4.1 a Tab. 4.2. Výsledky testu pre sha256 definujú koľko bytov bolo spracovaných za sekundu.

Tab. 4.1 Hodnoty testu rýchlosti sha256 na Raspberry Pi 3

Algoritmus	**16B**	**64B**	**256B**	**1024B**
Sha256	10 593,59 k	26 750,25 k	49 139,46 k	61 687,47 k

Tab. 4.2 Hodnoty testu rýchlosti RSA na Raspberry Pi 3

RSA dĺžka kľúča	**Podpis**	**Overenie**
1024	0,004006 s	0,000198 s
2048	0,025 s	0,000451 s

4.4.1 Použité správy

Implementovaný protokol využíva rovnaké správy ako v prípade protokolu AODV popísané v časti 4.3.1. Líšia sa iba pridanou bezpečnostnou SAODV hlavičkou pri správach RREQ, RREP a RERR.

4.4.1.1 Rozširujúca hlavička

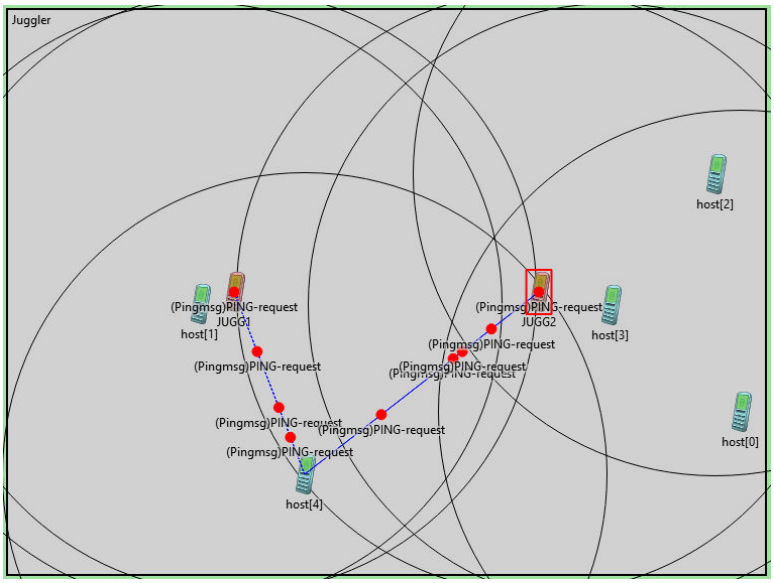
Rozširujúca hlavička o veľkosti 448 B (v prípade správy RERR 384 B z dôvodu chýbajúcich hašov) je implementovaná v zjednodušenej podobe. V našom prípade nám plne postačuje simulácia podpisovania a hašovania správ, ktorá je zabezpečená pomocou pridaného času pre spracovanie správy. V prípade, že uzol vygeneruje správu, ktorej overenie by zlyhalo buď na overovaní podpisu alebo hašovacej hodnoty, tak to oznámi na základe nasledovných polí:

1. hash - definuje či je hodnota vypočítaného hašu zhodná s vrchným hašom
2. signature - definuje či je podpis zhodný s vygenerovaným podpisom

4.5 Žonglér

Scenár útoku žonglér sa skladá z troch režimov, ktoré sú implementované v uzle typu JugglerAttack, ktorý je rozšírením základného uzla Aodv, pričom sú upravené niektoré z metód na zabezpečenie požadovaného správania. Implementovaný scenár pozostáva z dvoch útočných uzlov JUGG1 a JUGG2. Tie sú umiestnené tak, aby pokryli čo najväčšiu časť siete a zároveň boli od seba v dostatočnej vzdialenosti nato, aby medzi nimi nebola možná priama komunikácia.

Príklad so piatimi bežnými uzlami môžeme vidieť na obr. 4.3, kde môžeme vidieť viacero ping správ ktoré sú v oneskorovacej slučke medzi škodlivými uzlami JUGG1 a JUGG2.

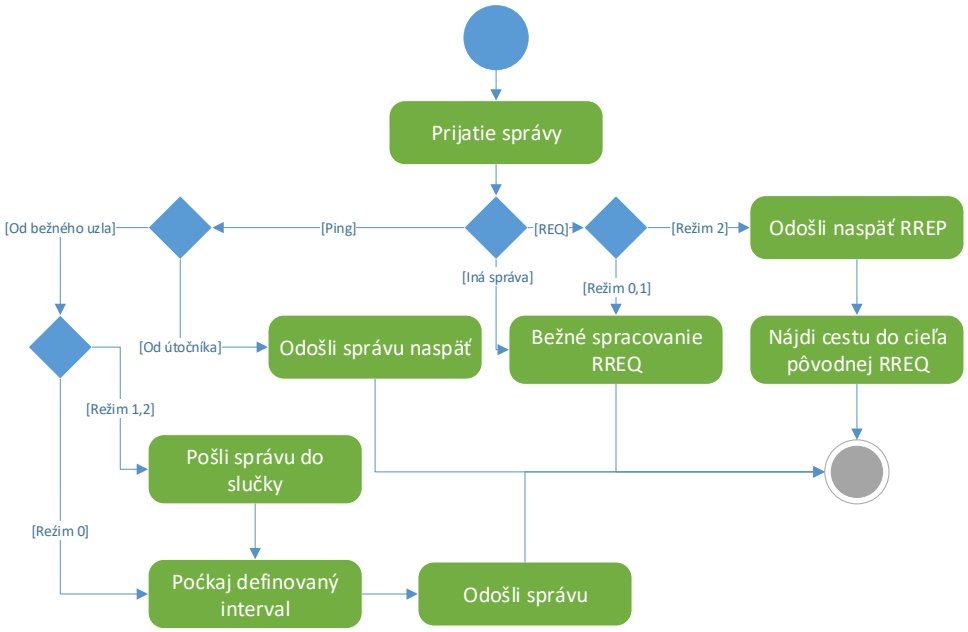


Obr. 4.3 Žonglér s piatimi bežnými uzlami

Jednotlivé režimy a parametre je možné ovládať priamo zo súboru .ini. Medzi tieto parametre patria:

- active
  - 0 uzol neaktívny
  - 1 uzol aktívny
- mode
  - 0 režim základného oneskorenia
  - 1 režim žongléra bez ovplyvňovania ciest
  - 2 režim žongléra s ovplyvňovaním ciest
- attack_route_update_interval - pravidelný interval hľadania ciest medzi jednotlivými útočnými uzlami na zabezpečenie udržiavania existencie oneskorovacej slučky
- secs_to_delay - definuje o aký dlhý čas v sekundách má byť správa oneskorená

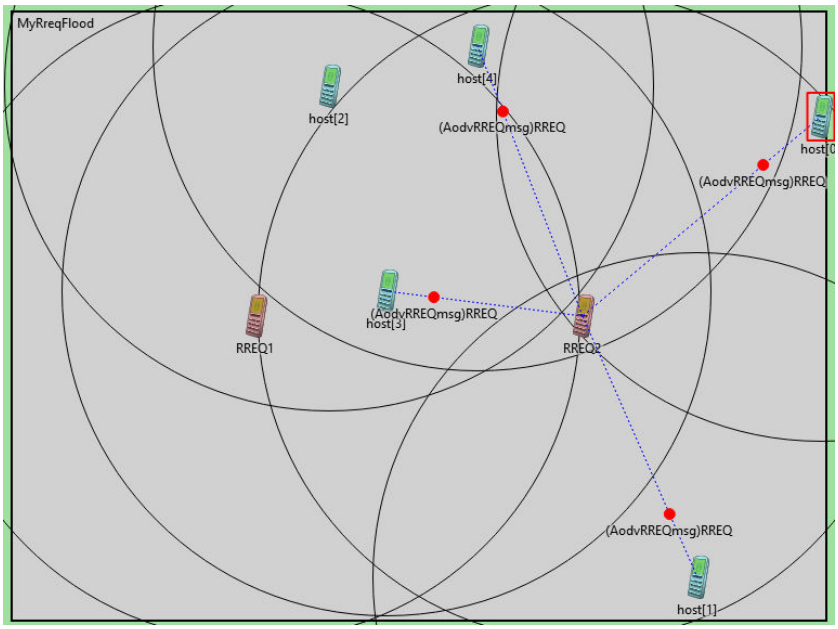
Zjednodušený diagram princípu uzla žonglér v scenári môžeme vidieť na obr. 4.4. Použité čísla režimov pri rozhodovaní vychádzajú z parametra mode.



Obr. 4.4 Základný princíp uzla Žonglér

4.6 REQ záplava

Scenár REQ záplavy tvoria dva režimy popísané v časti 3.3.1. Implementovaný uzol RreqAttack je takisto rozšírením normálneho uzla s upravenými metódami. Uzly sú rovnako ako pri žonglérovi umiestnené tak, aby pokryli čo najväčšiu časť siete a sú od seba natoľko vzdialené, aby neboli schopné priamo medzi sebou komunikovať. Uzly sú pomenované RREQ1 a RREQ2. Na obr. 4.5 môžeme vidieť vygenerované REQ správy uzlom RREQ2 v protokole AODV.

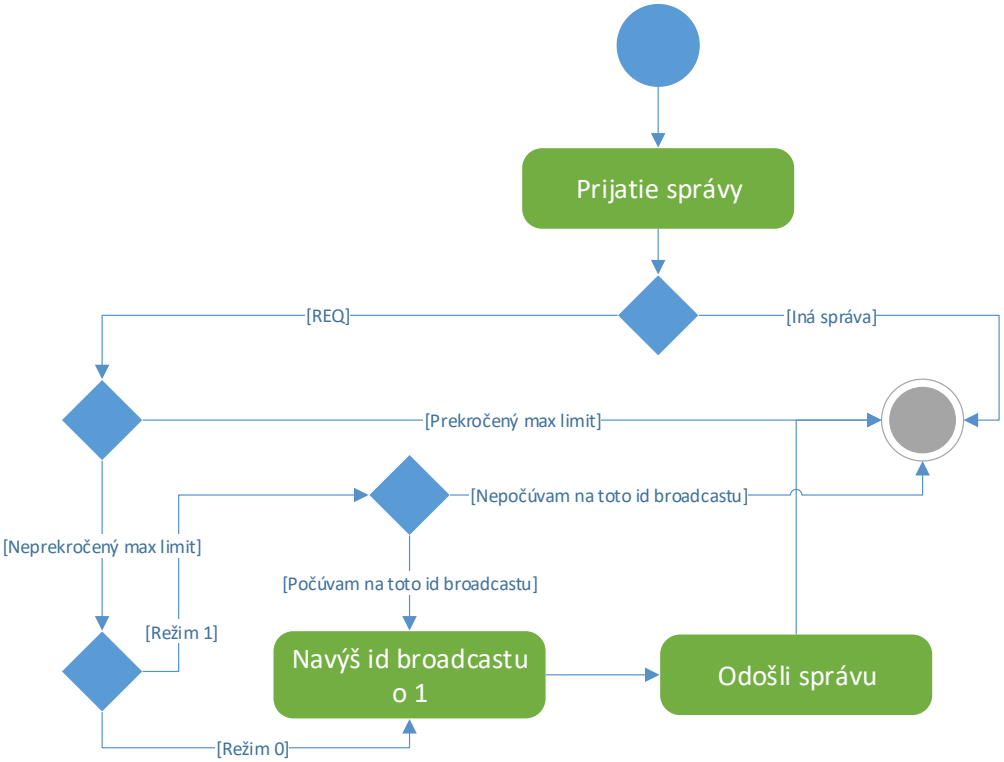


Obr. 4.5 Implementovaná REQ záplava

Jednotlivé režimy a parametre útoku je možné ovládať z .ini súboru. Medzi tieto parametre patria:

- active
  - 0 uzol neaktívny
  - 1 uzol aktívny
- mode
  - 0 režim klasickej REQ záplavy
  - 1 režim navrhnutej REQ záplavy pri ktorej škodlivé uzly reagujú len na určité sekvenčné čísla
- my_flood_param - berie sa do úvahy len v režime 1. Uzol s hodnotou 0 reaguje len na párne sekvenčné čísla a uzol s hodnotou 1 len na nepárne sekvenčné čísla.
- rate_limit - maximálny povolený počet vygenerovaných REQ správ za sekundu

Zjednodušený princíp uzla REQ záplavy môžeme vidieť v diagrame na obr. 4.6. Použité čísla režimov pri rozhodovaní vychádzajú z parametra mode.



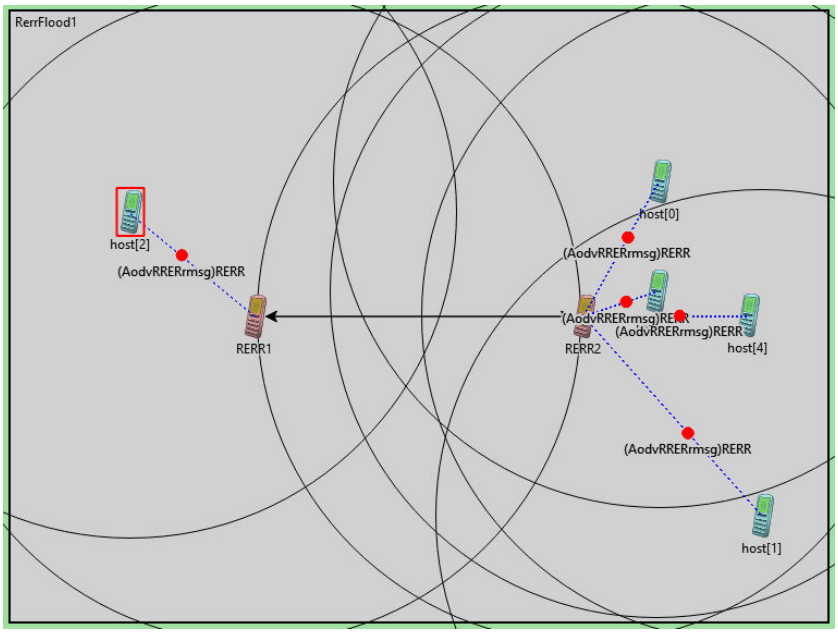
Obr. 4.6 Princíp uzla REQ záplavy

4.7 RER záplava

Uzol RER záplavy RerrFlood je taktiež ako ostatné útoky rozšírením bežného uzla s implementovanými dvoma režimami popísanými v časti 3.4.2. Uzly pokrývajú čo najväčšiu časť siete pričom sú kvôli druhému režimu navzájom obojsmerne prepojené vysokorýchlostným spojením. Toto spojenie nie je pri prvom režime využité.

Oba útočné uzly nie sú viditeľné ostatným uzlom v sieti. Viditeľnosť nie je nutná, pretože oba uzly odosielajú správy, ktoré podľa ich obsahu pochádzajú od iných uzlov v sieti.

Príklad útoku môžeme vidieť na obr. 4.7, kde oba škodlivé uzly RERR1 a RERR2 generujú RER správy, ktorými sa snažia spôsobiť vymazanie existujúcich ciest zo smerovacích tabuliek bežných uzlov.

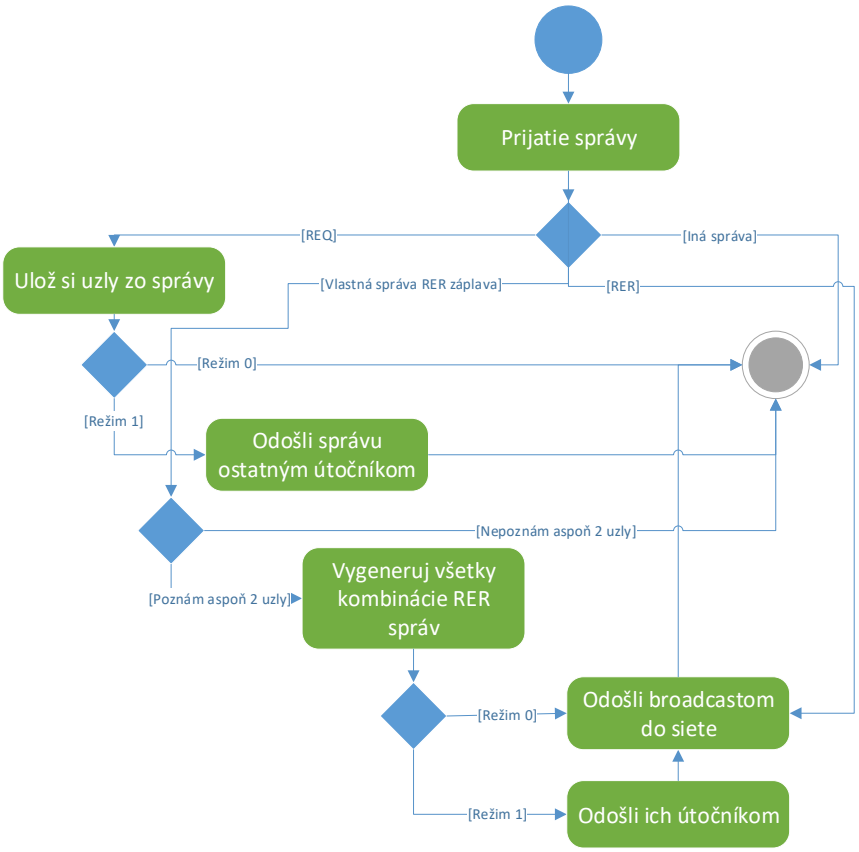


Obr. 4.7 Implementovaná RER záplava

Jednotlivé režimy a parametre RER záplavy je možné ovládať z .ini súboru. Medzi tieto parametre patria:

- active
  - 0 uzol neaktívny
  - 1 uzol aktívny
- mode
  - 0 režim bez komunikácie medzi útočnými uzlami
  - 1 režim s komunikáciou medzi útočnými uzlami
- flood_interval - definuje ako často v sekundách majú škodlivé uzly generovať RER správy. Vysoká hodnota parametra spôsobí expiráciu ciest, ale nebude mať vplyv na úspešnosť doručenia správ.

Zjednodušený princíp funkcionality uzla RER záplavy môžeme vidieť na obr. 4.8. Použité čísla režimov pri rozhodovaní vychádzajú z parametra mode.



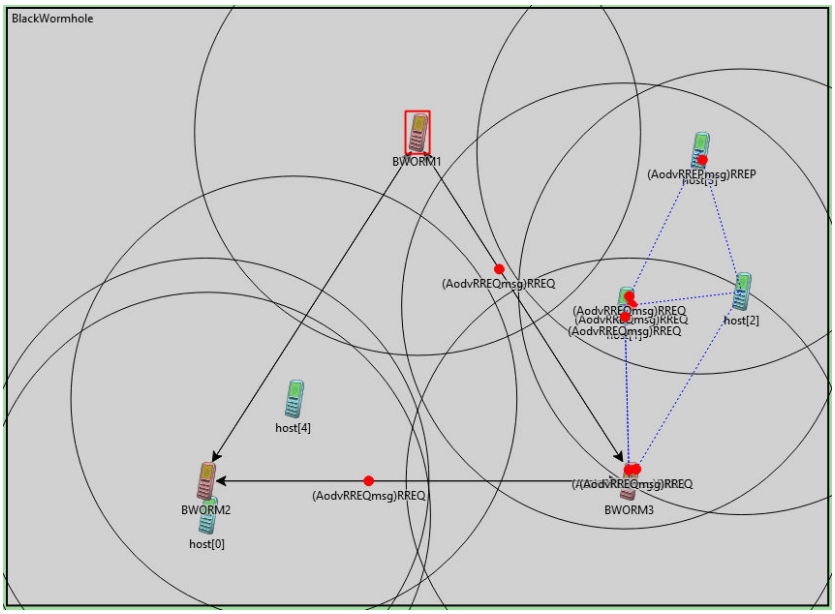
Obr. 4.8 Princíp uzla RER záplavy

4.8 Čierna červia diera

Uzol čiernej červej diery BlackWormhole je taktiež rozšírením bežného uzla s implementovanými piatimi režimami popísanými v časti 3.5.2. Pre dosiahnutie najväčšieho dopadu útoku sú škodlivé uzly rozmiestnené v tvare trojuholníka s dostatočnými odstupmi. Škodlivé uzly sú navzájom prepojené pomocou vysokorýchlostného spojenia a spolu tak pokrývajú veľkú časť siete.

Viditeľnosť škodlivých uzlov z hľadiska bežných uzlov sa líši v závislosti od použitého režimu. Neviditeľné sú pri režimoch odvodených od červej diery, a to červia diera, čierna červia diera a sivá čierna diera. Viditeľné sú pri režimoch čiernej a sivej diery.

Na obr. 4.9 môžeme vidieť využitie spojení medzi škodlivými uzlami BWORM1, BWORM2 a BWORM3 na preposlanie správy REQ, ktorá bola prijatá uzlom BWORM3.

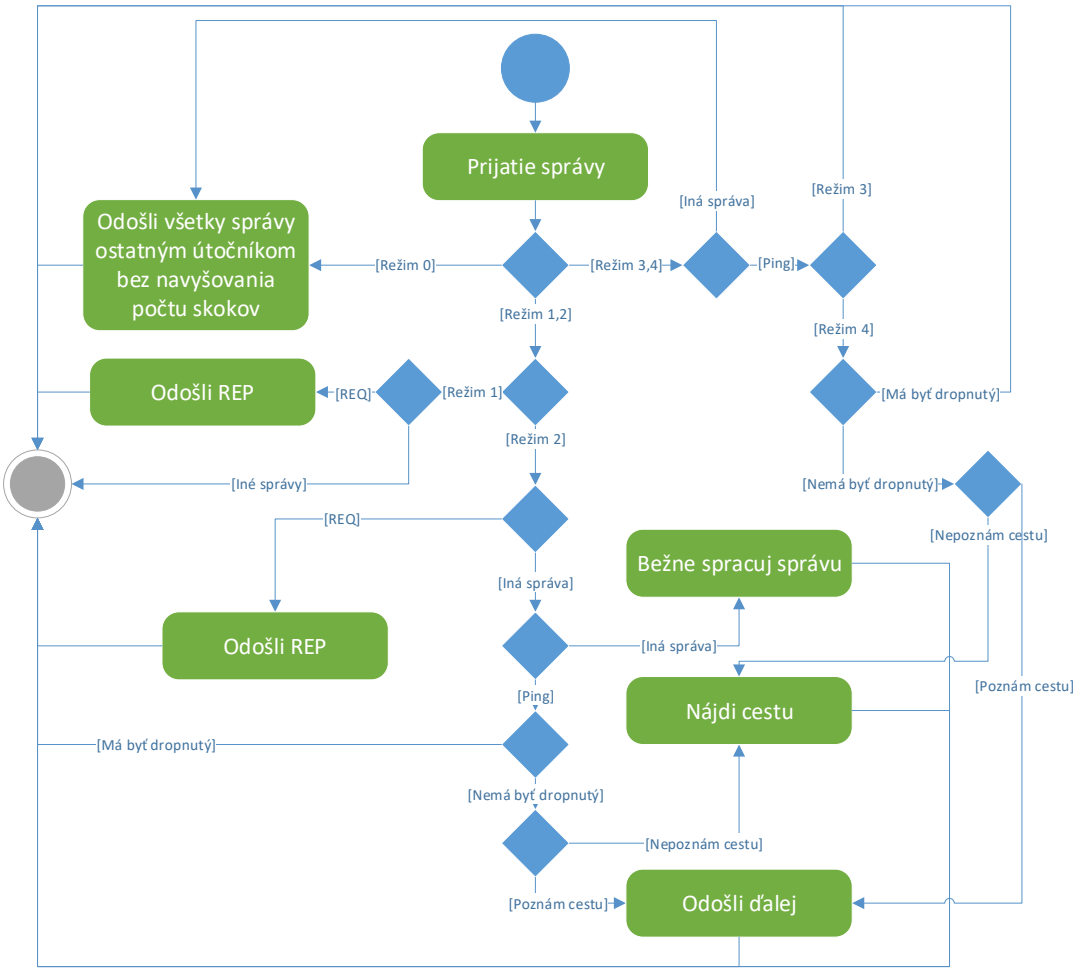


Obr. 4.9 Implementovaná čierna červia diera

Jednotlivé režimy a parametre čiernej červej diery je možné ovládať z .ini súboru. Medzi tieto parametre patria:

- active
  - 0 uzol neaktívny
  - 1 uzol aktívny
- mode
  - 0 režim červej diery
  - 1 režim čiernej diery
  - 2 režim sivej diery
  - 3 režim čiernej červej diery
  - 4 režim sivej čiernej diery
- drop_percentage - parameter pre režimy 2 a 4. Určuje pravdepodobnosť zahodenia správ sivou dierou.

Zjednodušený princíp funkcionality uzla čiernej červej diery môžeme vidieť na obr. 4.10. Použité čísla režimov pri rozhodovaní vychádzajú z parametra mode.



Obr. 4.10 Princíp uzla čiernej červej diery

5 Testovanie a simulácie

V tejto časti sa nachádza testovanie a vyhodnotenie simulácií jednotlivých navrhnutých testovacích scenárov. V prvej časti sa nachádza zvolené nastavenie sieťových parametrov pre všetky testovacie scenáre. Špecifické hodnoty parametrov budú uvedené pri konkrétnych scenároch. Nasleduje vyhodnotenie simulácií jednotlivých scenárov na základe definovaných parametrov. V závere zhodnotíme výsledky všetkých simulácií.

5.1 Všeobecné parametre scenárov

Konkrétne použité hodnoty všeobecných parametrov môžeme vidieť v tab. 5.1. Okrem hodnôt parametrov je dôležitý aj pomer súvisiacich parametrov ako sú napríklad ping_retransmit_wait a transmit_delay.

Tab. 5.1 Hodnoty všeobecných parametrov scenárov

Parameter	Hodnota
pings_times_to_try	5
transmission_distance	300
ping_retransmit_wait	0,5
transmit_delay	0,01
mobility_delay	0,1

Na základe hodnoty parametra mobility_delay vieme povedať, že uzly sa pohybujú rýchlosťou 10 jednotiek za sekundu v prípade horizontálneho a vertikálneho pohybu a rýchlosťou 14 jednotiek za sekundu v prípade diagonálneho pohybu.

Všetky testovacie scenáre boli simulované s premenlivým paramerom numNodes, ktorý definuje počet uzlov. Tento parameter bol testovaný v rozmedzí hodnôt <5,15>. Každá simulácia bola zopakovaná 5 krát pre každú hodnotu patametra numNodes. Tým sme dosiahli rôzne štartovacie pozície a pohyb uzlov a výsledky simulácií tak dávali presnejšie výsledky. Celkový počet simulácii pre jeden protokol tak spolu so simuláciami bez útoku dosiahol hodnotu 880.

5.1.1 AODV

Tab. 5.2 Hodnoty parametrov protokolu AODV

Parameter	Hodnota
rt_table_lifetime	3
rt_lifetime_update_interval	1

5.1.2 SAODV

Tab. 5.3 Hodnoty parametrov protokolu SAODV

Parameter	Hodnota
rt_table_lifetime	3
rt_lifetime_update_interval	1
sha256_verification	0,0005981
rsa_sign	0,014503
rsa_verify	0,000451

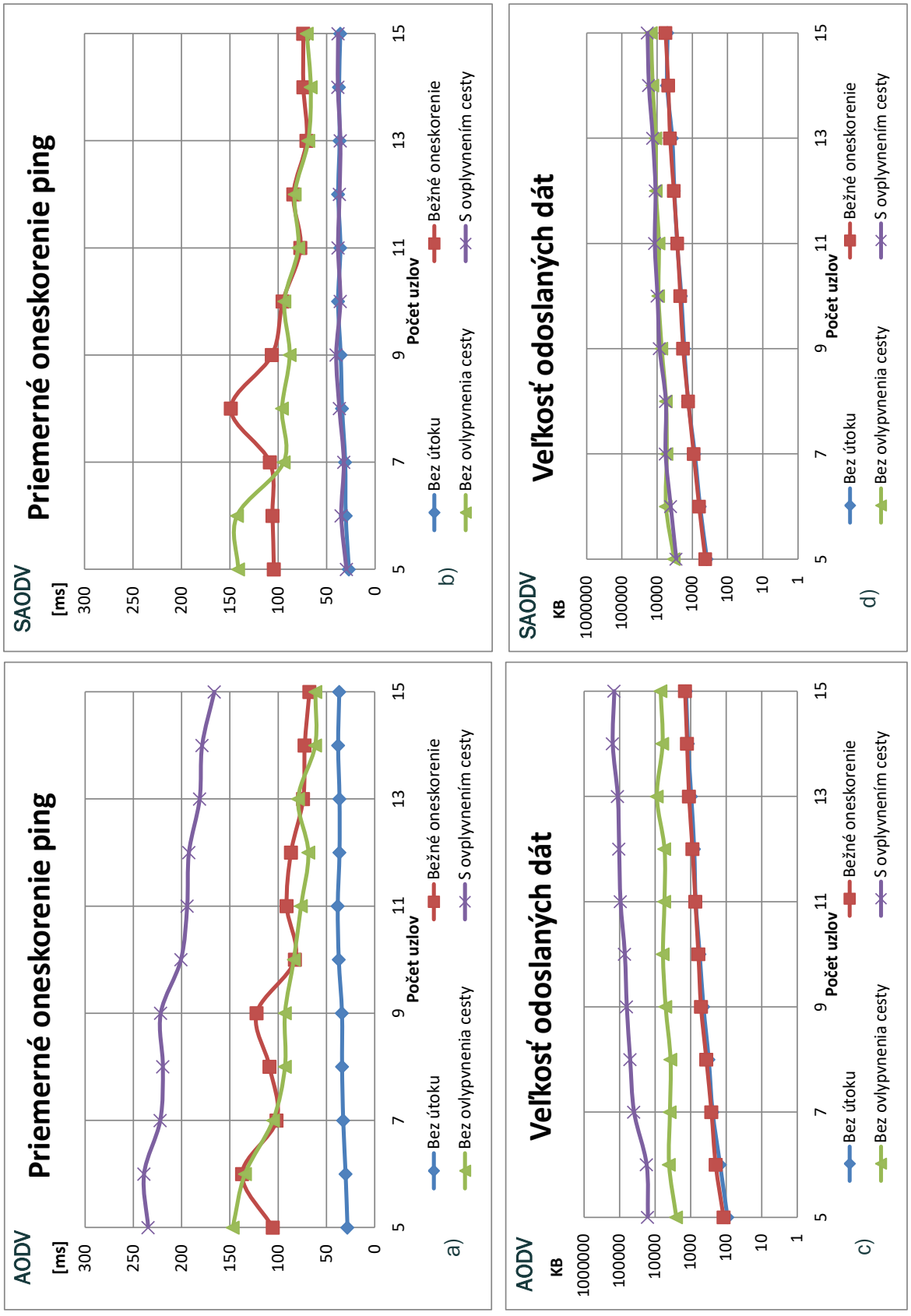
5.2 Testovací scenár žonglér

Hodnoty parametrov testovacieho scenára žonglér pre všetky testované protokoly môžeme vidieť v tab. 5.4.

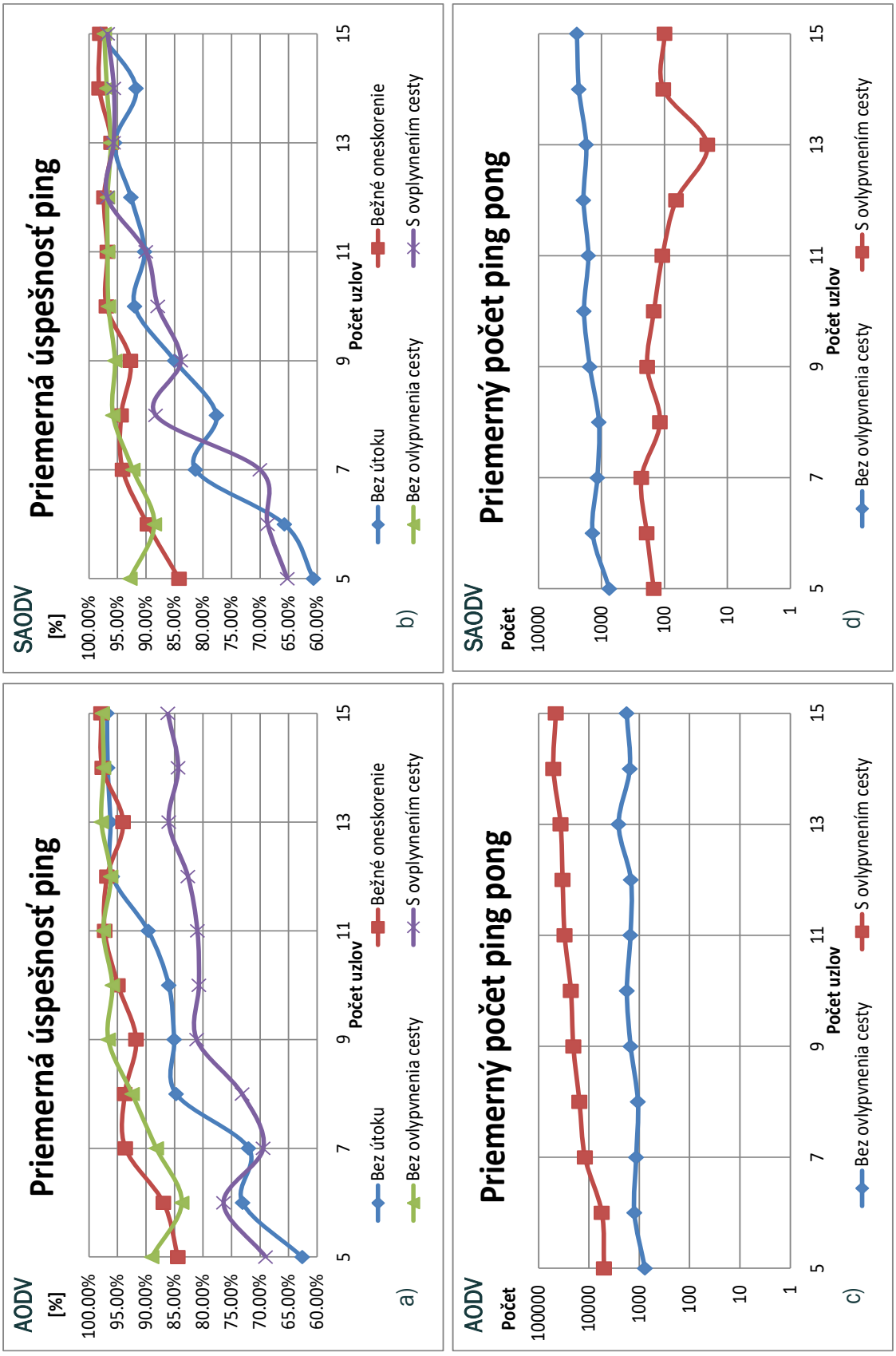
Tab. 5.4 Hodnoty parametrov scenára žonglér

Parameter	Hodnota
max_x_position	750
max_y_position	560
secs_to_delay	0,15
attack_route_update_interval	0,8

Grafy sledovaných parametrov útoku môžeme vidieť na Obr. 5.1 a Obr. 5.2. Režimy, ktoré neobsahovali konkrétne sledované parametre, alebo boli ich hodnoty pre daný parameter nulové, nie sú v jednotlivých grafoch zobrazené. Dôležité je upozorniť, že mierka v grafe veľkosť odoslaných dát a priemerný počet ping pong (výmen oneskorovacích správ medzi škodlivými uzlami) je logaritmická. Všetky zobrazené hodnoty parametrov sú priemery piatich simulácií pre fixný počet uzlov.



Obr. 5.1 Výsledky scenára žonglér pre protokoly AODV a SAODV (1)



Obr. 5.2 Výsledky scenára žonglér pre protokoly AODV a SAODV (2)

5.2.1 Vyhodnotenie scenára žonglér

Na základe výsledkov môžeme povedať, že každý režim útoku bol v prípade protokolu AODV úspešný a pridal správam ping určité oneskorenie. Režim s bežným oneskorením (0) bol v tomto prípade takmer zhodný s režimom bez ovplyvnenia cesty (1), ktorý dosiahol o trochu vyššie priemerné oneskorenia správ. Najefektívnejší bol režim s ovplyvnením cesty (2), ktorý bol od režimu 1 priemerne o 56 % účinnejší. Taktiež vieme usúdiť, že so stúpajúcim počtom uzlom efektivita útoku klesala, keď sme dosiahli pri počte uzlov 15 pri režimoch 0 a 1 len približne menej ako dvojnásobné zvýšenie oneskorenia. Pri protokole SAODV si môžeme všimnúť, že režim 2 nebol vôbec úspešný z dôvodu generovania falošných správ, ktoré protokol zdetegoval a zahodil. Ostatné režimy dosiahli v tomto parametri podobné hodnoty ako pri protokole AODV, na základe čoho môžeme tvrdiť, že zvýšená časová náročnosť na spracovanie správ s podpismi a hašami mala nebadateľný vplyv na oneskorenie smerovacích správ.

Veľkosť odoslaných dát bola taktiež závislá od zvoleného režimu a použitého protokolu. Očakávane bol najefektívnejší režim 2, pri ktorom sme v protokole AODV dosiahli priemerné navýšenie o 13 653 %. Za ním nasledoval režim 1 s priemerným navýšením o 1266 %. Režim 0 nedosiahol žiadne navýšenie, pretože nevytváral oneskorovaciu slučku a negeneroval tak žiadne správy naviac. Z dôvodu zachyteného útoku sme najväčšie množstvo dát v sieti v protokole SAODV nedosiahli iba v režime 2, ale aj v režime 1 kde sa hodnoty týchto režimov prekrývajú. Všimnime si ale až 10 násobný nárast vygenerovaných dát v režimoch 0 a bez útoku. Túto skutočnosť spôsobila veľká hlavička protokolu SAODV.

Takisto aj pri priemernej úspešnosti ping správ bol najúspešnejší režim 2 s priemerom 79,08 % pri protokole AODV vďaka tomu, že škodlivé uzly predstierali existenciu najkratšej cesty do cieľového uzla cez nich. Túto cestu sa snažili následne nájsť, ale ak ju nenašli tak prijaté správy neboli schopné doručiť a tak bola dosiahnutá aj nižšia úspešnosť doručenia ping správ. Ostatné režimy zvýšili úspešnosť ping správ z dôvodu pridaných ďalších dvoch uzlov, ktoré zvýšili celkovú dostupnosť uzlov v sieti. Pri protokole SAODV sme v tomto parametri zaznamenali rozdiel len pri detegovanom režime 2 s hodnotami podobnými režimu bez útoku.

Pri režimoch 1 a 2 sme sledovali aj počet správ vymenených v oneskorovacej slučke a očakávane dosiahol režim 2 v protokole AODV oveľa vyššie hodnoty ako pri režime 1. So stúpajúcim počtom uzlov tento počet logicky narastal, pričom pri režime 2 išlo o oveľa prudší nárast ako v prípade režimu 1. Pri 15tich uzloch išlo konkrétne o 904% nárast oproti

simulácii s 5timi uzlami. Situácia sa v protokole SAODV vymenila. Počet vymenených správ bol v režime 2 asi 10 násobne menší. Spôsobila to už spomínaná detekcia tohto útoku.

Celkovo tak najlepšie výsledky očakávane dosiahol režim 2, ktorý bol ale v protokole SAODV detegovaný na základe generovania falošných správ. Ostatné režimy ale týmto protokolom detegované neboli. Náš útok v režimoch 2 a 1 bol celkovo efektívnejší ako bežné pridanie oneskorenia režimu 0.

5.2.2 Vyhodnotenie nesimulovaných protokolov

Protokol DSR je zraniteľný pre všetky režimy útokov v tomto scenári pretože nemá žiadne mechanizmy zabraňujúce generovaniu falošných správ alebo analýze sieťovej premávky pre detekciu oneskorenia správ. Takisto je tu možnosť útoku pri ktorom vieme meniť zoznam uzlov v správach procesu hľadania cesty. Môžeme tak vytvoriť dlhšie komunikačné cesty, ktoré budú sami o sebe zvyšovať oneskorenie medzi komunikujúcimi uzlami.

Protokol SRP by mal byť vďaka kontrolným súčtom schopný detegovať režim s posielaním falošných správ. Ostatné režimy by nemali byť detegované. Rovnaký útok pridania oneskorenia pomocou modifikácie správ protokolu DSR môžeme využiť aj v protokole SRP pretože toto bolo nie je používané pri výpočte kontrolného súčtu kvôli jeho premenlivosti.

Ak predpokladáme, že v protokole ARAN majú certifikáty aj škodlivé uzly, tak bude jeho detekčná schopnosť na úrovni protokolu SRP. V prípade, že tieto uzly certifikát nemajú, tak by sa vôbec nemohli zúčastňovať komunikácie v sieti a tým pádom by nebol úspešný žiadny z režimov.

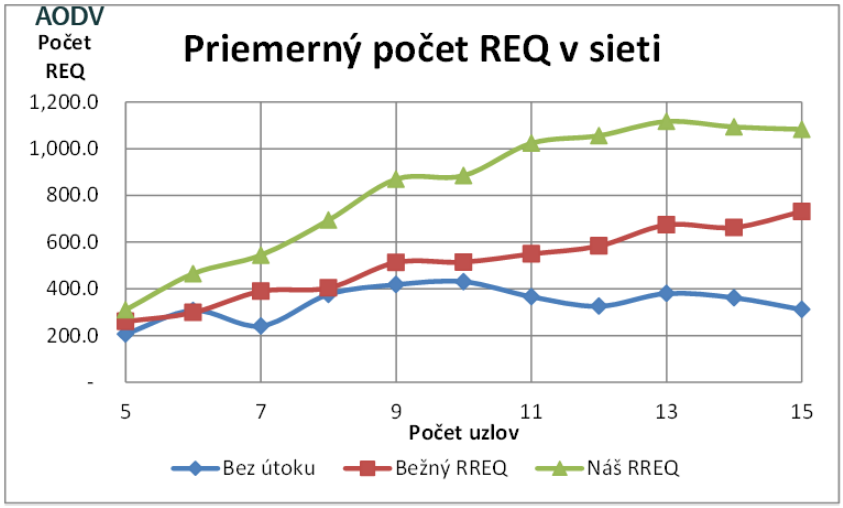
5.3 Testovací scenár REQ záplava

Hodnoty parametrov testovacieho scenára REQ záplava pre všetky testované protokoly môžeme vidieť v tab. 5.4.

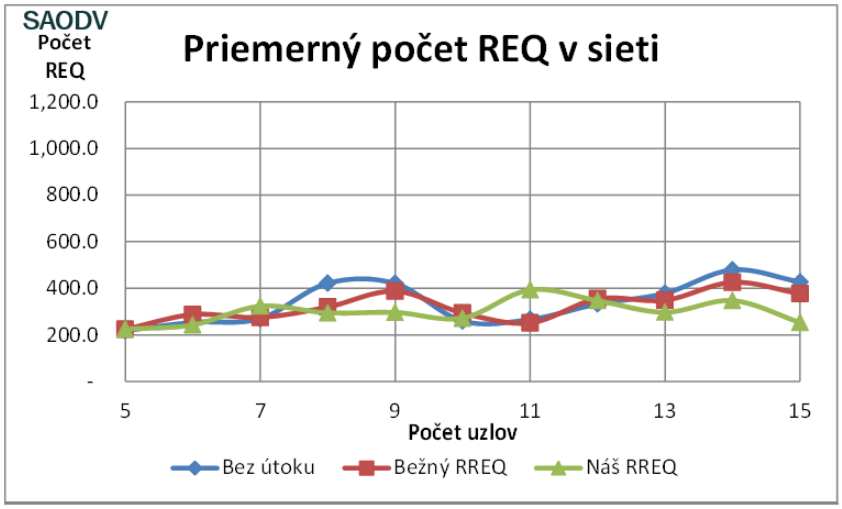
Tab. 5.5 Hodnoty parametrov scenára REQ záplava

Parameter	Hodnota
max_x_position	750
max_y_position	560
rate_limit	10

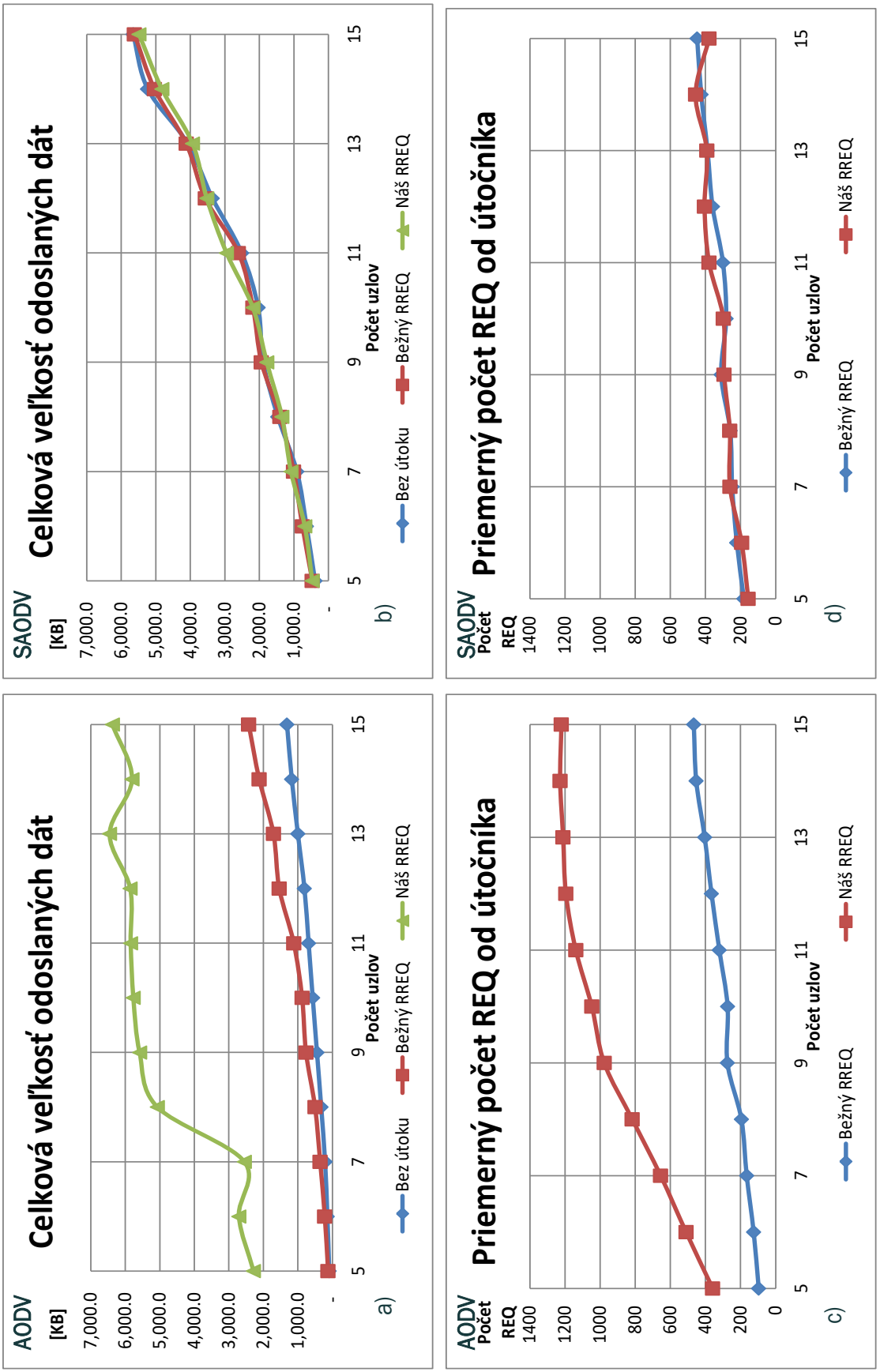
Grafy sledovaných parametrov útoku môžeme vidieť na Obr. 5.3, Obr. 5.4, Obr. 5.5 a Obr. 5.6. Režimy, ktoré neobsahovali konkrétne sledované parametre, alebo boli ich hodnoty pre daný parameter nulové, nie sú v jednotlivých grafoch zobrazené. V žiadnom z grafov nie je použitá logaritmická mierka.



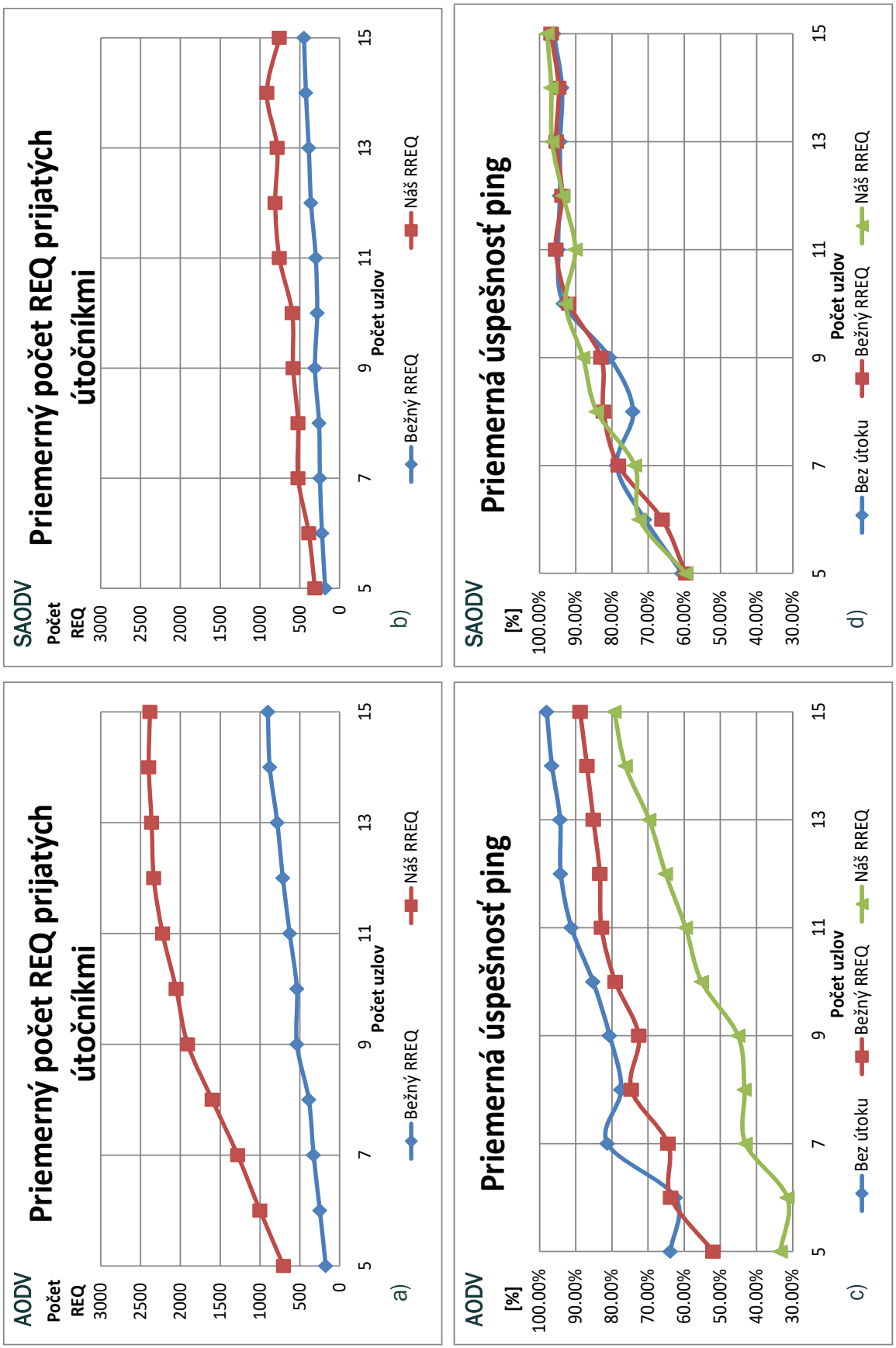
Obr. 5.3 Priemerný počet REQ správ protokolu AODV



Obr. 5.4 Priemerný počet REQ správ protokolu SAODV



Obr. 5.5 Výsledky scenára REQ záplava pre protokoly AODV a SAODV (1)



Obr. 5.6 Výsledky scenára REQ záplava pre protokoly AODV a SAODV (2)

5.3.1 Vyhodnotenie scenára REQ záplava

Cieľom REQ záplavy bolo najmä preťaženie dostupnej šírky pásma. Na základe grafov môžeme povedať že oba režimy zvýšili celkovú veľkosť odoslaných dát v sieti. Režim našej navrhnutej záplavy (2) bol ale oveľa účinnejší ako bežný režim (1), keď pri našom útoku dosiahla veľkosť dát v niektorých prípadoch približne až 6-násobok základného scenára a pri režime 1 približne len 2-násobok. V prípade protokolu SAODV boli detegované oba režimy z dôvodu generovania falošných správ, ktorým bolo zmenené sekvenčné číslo. Celková veľkosť odoslaných dát tak bola vo všetkých režimoch rovnaká ale môžeme si všimnúť rozdiel oproti protokolu AODV kvôli veľkej SAODV hlavičke. Rozdiely nie sú až také výrazné pretože pri protokole AODV boli oba režimy úspešné a vygenerovaných tak bolo oveľa viac REQ správ, ktoré sa na tomto grafe prejavili.

Pri parametri počtu REQ správ vygenerovaných útočníkom v protokole AODV bol náš režim 2 oveľa oveľa úspešnejší v porovnaní s režimom 1. Aj keď uzly reagovali len na určité sekvenčné čísla, tak vďaka existujúcim cestám medzi týmito uzlami vznikol opakujúci sa cyklus reakcie na vzájomne vygenerované správy a to sa odrazilo aj na priemernom počte REQ správ prijatých útočníkmi. Pri protokole SAODV sú hodnoty tohto parametra pre oba režimy rovnaké, pretože náš režim nedokázal vytvoriť opakovaciu slučku z dôvodu jeho detekcie.

Záplavy pôsobili aj na úspešnosť ping správ. Najväčší vplyv mal v protokole AODV opäť náš režim 2, ktorý dosiahol priemerne 54 % úspešnosť. Vplyv na úspešnosť mal aj režim 1 ale nebol taký výrazný ako v prípade režimu 2. S narastajúcim počtom uzlov úspešnosť ping správ stúpala. Z rovnakého dôvodu ako v predchádzajúcich prípadoch boli hodnoty tohto parametra v protokole SAODV rovnaké pre všetky režimy.

Taktiež aj pri celkovom počte REQ správ v sieti bol najúčinnejší režim 2 s maximálne 5-násobným nárastom oproti režimu bez útoku. Pri režime 1 sme zaznamenali viac ako 3-násobný nárast. Ako už bolo povedané, protokol SAODV detegoval oba režimy útokov a tak sa ich vplyv v tomto sledovanom parametri neprejavil.

Celkovo bol tak náš návrh útoku efektívnejší ako bežne používaný princíp režimu 1. SAODV útoku úspešne zabránil na úkor väčšieho objemu vygenerovaných dát.

5.3.2 Vyhodnotenie nesimulovaných protokolov

Protokol SRP nie je chránený proti žiadnemu z režimov v tomto scenári. Princíp protokolu pri ktorom si ukladá zoznam uzlov priamo v smerovacích správach by navyše spôsobil ešte väčšie zahltenie.

Protokol DSR by mal byť rovnako ako odsimulovaný SAODV odolný proti obom režimom útokov. Falošné správy identifikuje na základe kontrolného súčtu vypočítaného z obsahu správy a zdieľaného kľúča (Predpokladáme iba útoky na smerovací protokol).

Nezávisle od predpokladu vlastnenia certifikátu škodlivými uzlami by bol protokol ARAN zabezpečený proti všetkým režimom tohto scenára. Jediným rozdielom pri vlastníctve certifikátov by bolo väčšie množstvo vygenerovaných dát, ktoré by ale nedosiahlo silu útoku. Zvýšiť by sme ho v tomto prípade mohli jedine pomocou zníženia periódy generovania záplavových správ, ktoré by boli aj tak zahodené a mohli by potencionálne zahltiť len časť siete. Toto tvrdenie platí aj pre ostatné zabezpečené protokoly.

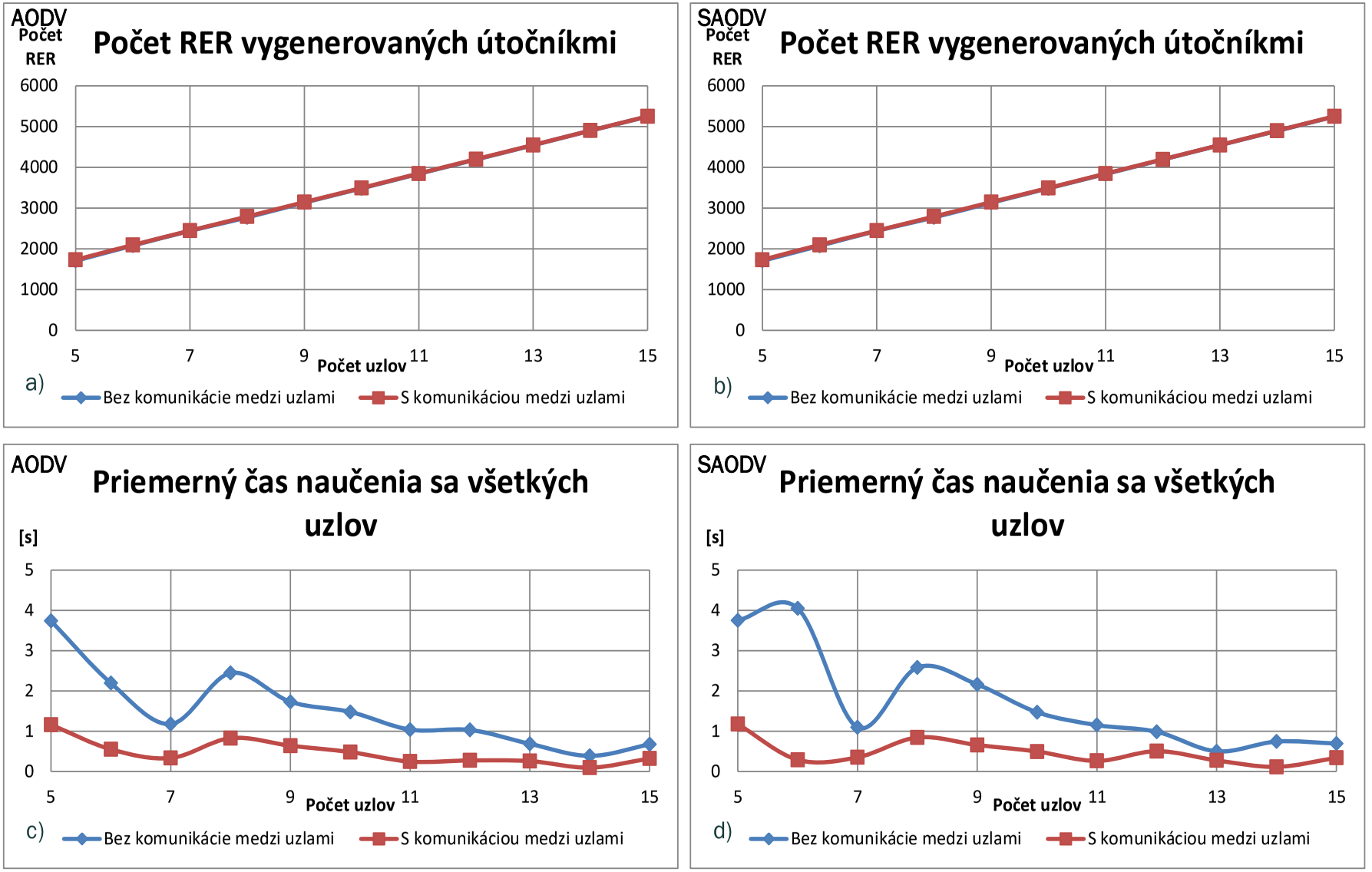
5.4 Testovací scenár RER záplavy

Hodnoty parametrov testovacieho scenára REQ záplava pre všetky testované protokoly môžeme vidieť v tab. 5.5.

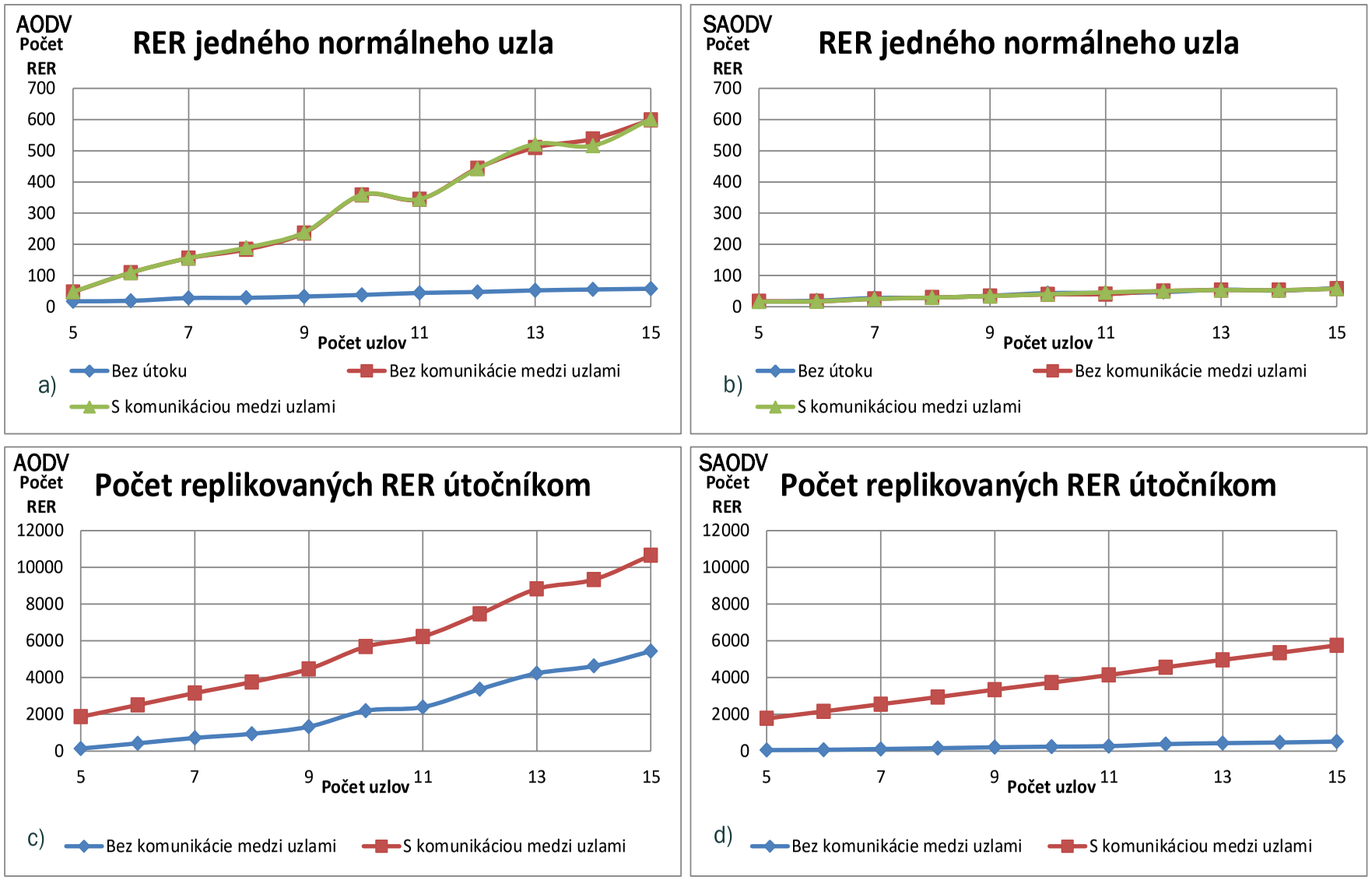
Tab. 5.6 Hodnoty parametrov scenára RER záplava

Parameter	**Hodnota**
max_x_position	750
max_y_position	560
flood_interval	0,2

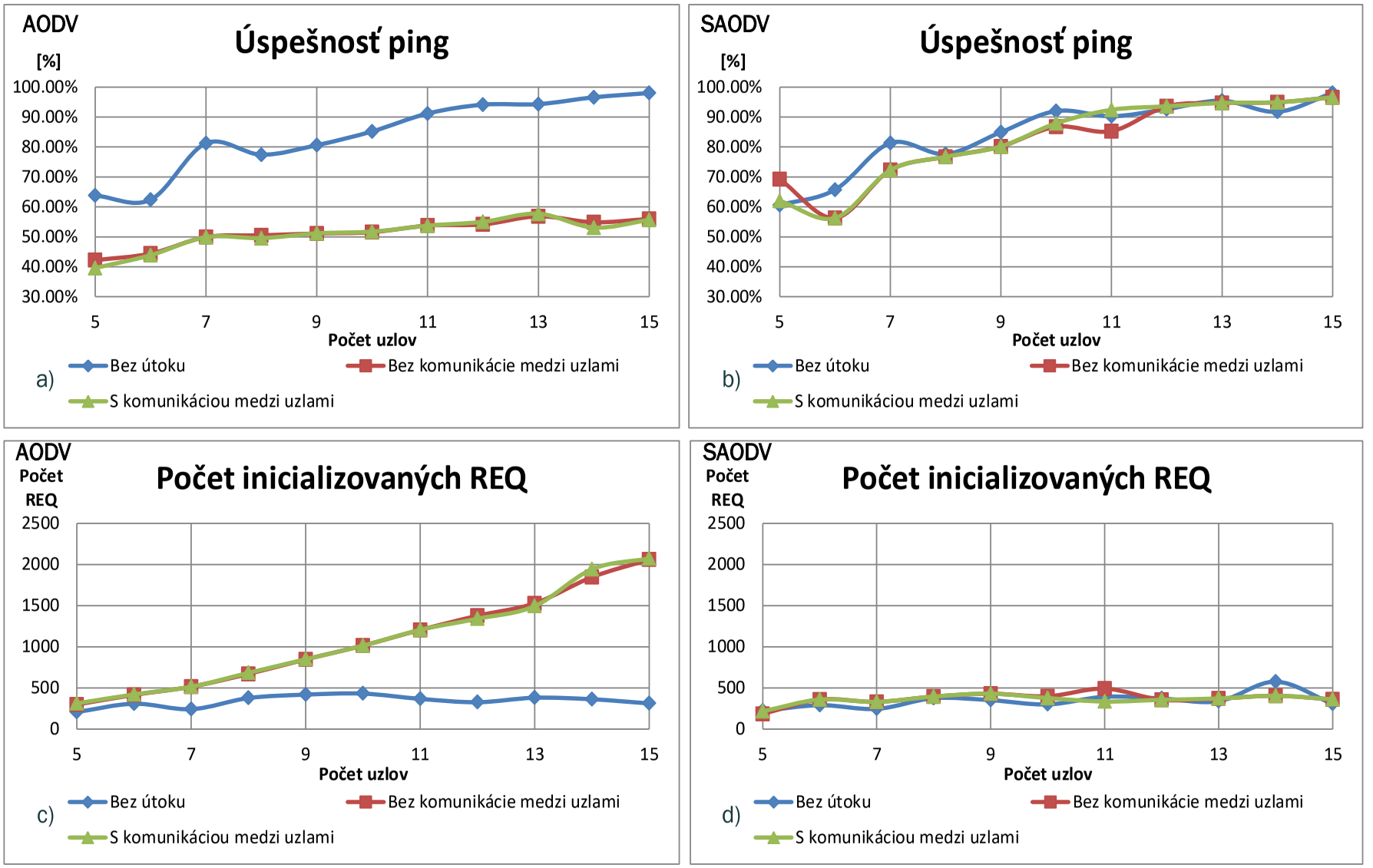
Grafy jednotlivých sledovaných parametrov môžeme vidieť na obrázkoch 5.4 a 5.5. Režimy, ktoré daný parameter neobsahovali alebo bola hodnota daného parametra nulová nie sú v grafoch zobrazené.



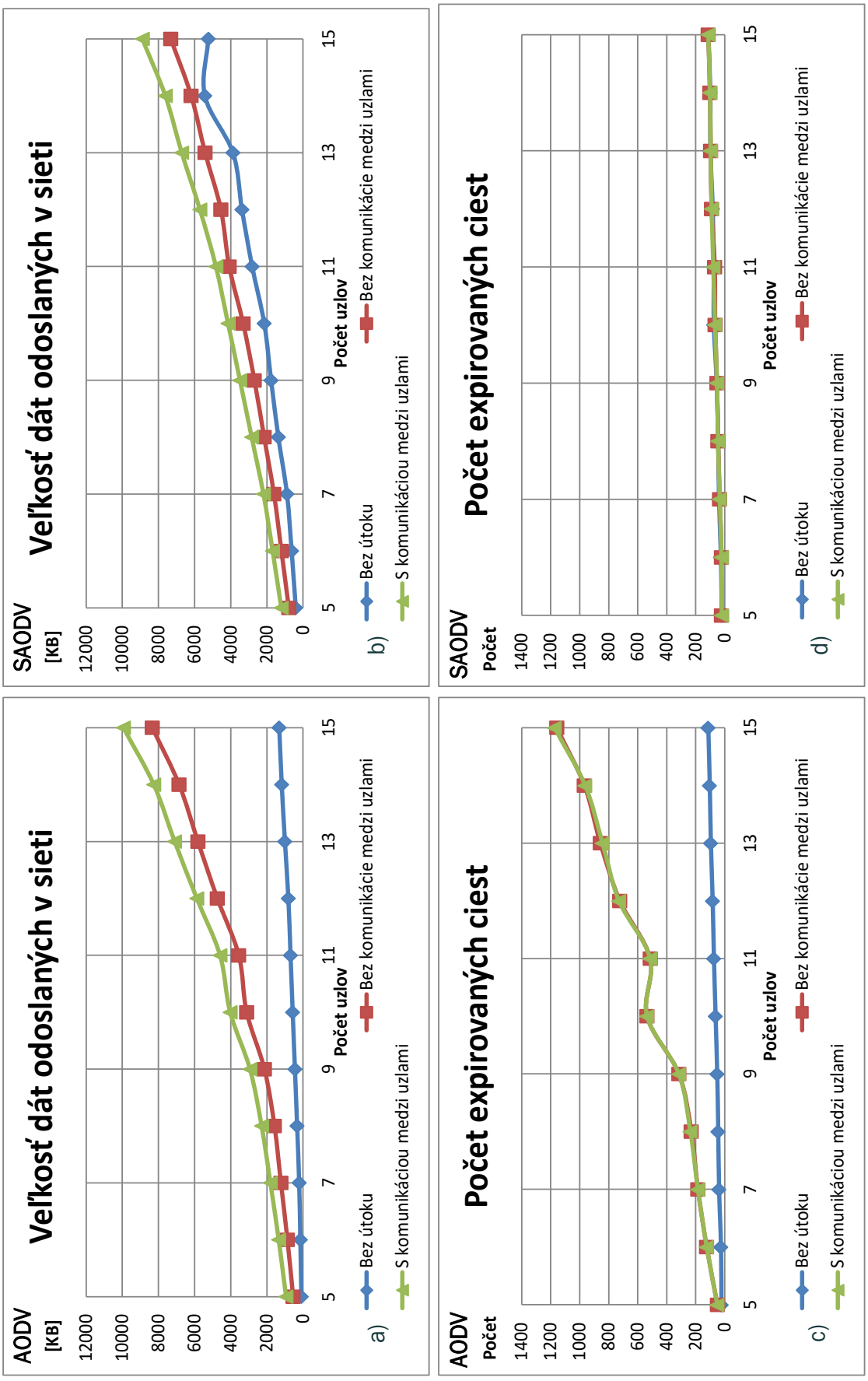
Obr. 5.7 Výsledky scenára RER záplava pre protokoly AODV a SAODV (1)



Obr. 5.8 Výsledky scenára RER záplava pre protokoly AODV a SAODV (2)



Obr. 5.9 Výsledky scenára RER záplava pre protokoly AODV a SAODV (3)



Obr. 5.10 Výsledky scenára RER záplava pre protokoly AODV a SAODV (4)

5.4.1 Vyhodnotenie scenára

Na základe simulácii vieme povedať, že režim bez komunikácie medzi uzlami (1) bol takmer totožný s režimom s komunikáciou medzi uzlami (2). Vidieť to môžeme napríklad pri porovnaní počtu RER správ vygenerovaných útočníkmi, ktorý je v oboch režimoch rovnaký z dôvodu fixnej periódy generovania správ.

Režimy ale mali vplyv na rýchlosť naučenia sa všetkých uzlov v sieti, kde režim 2 dosiahol očakávane lepšie časy ako režim 1. Rozdiel medzi týmito časmi bol menší so stúpajúcim počtom uzlov. Odchýlky medzi protokolmi AODV a SAODV boli v tomto parametri minimálne.

Rovnaké hodnoty dosiahli režimy v protokole AODV aj z hľadiska priemerného počtu vygenerovaných RER správ jedného normálneho uzla. Oba režimy značne zvýšili počet týchto správ oproti režimu bez útoku. V prípade protokolu SAODV boli oba režimy detegované na základe generovania falošných RERR správ útočnými uzlami.

Rozdielnosť režimov sa prejavila v počte replikovaných RER správ útočníkom. Tento rozdiel vyplýva z replikovania RER správ druhého škodlivého uzla pri režime 2 s komunikáciou medzi uzlami v prípade oboch simulovaných protokolov.

Najdôležitejším sledovaným parametrom bola úspešnosť ping správ, kde oba režimy dosiahli v protokole AODV požadované výsledky, keď značne znížili celkovú úspešnosť správ. Táto úspešnosť je navyše takmer nezávislá od počtu uzlov v sieti. Najväčší rozdiel dosiahnutý medzi režimom bez útoku a režimom s útokom bol približne 45%. Detekcia útoku pri SAODV nespôsobila žiadne významnejšie odchýlky pri oboch režimoch.

Počet inicializovaných RERR protokolu AODV bol v oboch režimoch vyšší v porovnaní s režimom bez útoku. Táto skutočnosť bola spôsobená úspešnými útokmi, ktoré vyvolali procesy hľadania ciest.

Malý rozdiel bol zaznamenaný v celkovej veľkosti odoslaných dát v sieti, kde logicky režim s komunikáciou dosiahol trochu vyššie hodnoty. Zaujímavé je, že aj napriek neúspešnosti útokov v protokole SAODV sa hodnoty príliš nelíšia od dosiahnutých hodnôt v protokole AODV vďaka veľkej hlavičke správ zabezpečeného protokolu.

Úspešnosť jednotlivých režimov môžeme vidieť aj pri počte expirovaných ciest v smerovacích tabuľkách bežných uzlov. Tento počet sa pri úspešnom útoku výrazne líšil od režimu bez útoku a narastal so stúpajúcim počtom uzlov v sieti.

Celkovo tak oba naše režimy dosiahli požadované výsledky, pričom rozdiely medzi režimami boli len veľmi malé a z hľadiska dopadu na sieť zanedbateľné. SAODV útok zdetegoval ale nezabránil väčšiemu množstvu odoslaných dát.

5.4.2 Vyhodnotenie nesimulovaných protokolov

Rovnako ako pri REQ záplave nie je protokol SRP chránený proti žiadnemu z režimov v tomto scenári. Protokol DSR by mal byť opäť odolný proti týmto útokom, ktoré odhalí na základe kontrolného súčtu. Takisto by bol odolný aj protokol ARAN na základe rovnakého princípu ako v testovacom scenári REQ záplava.

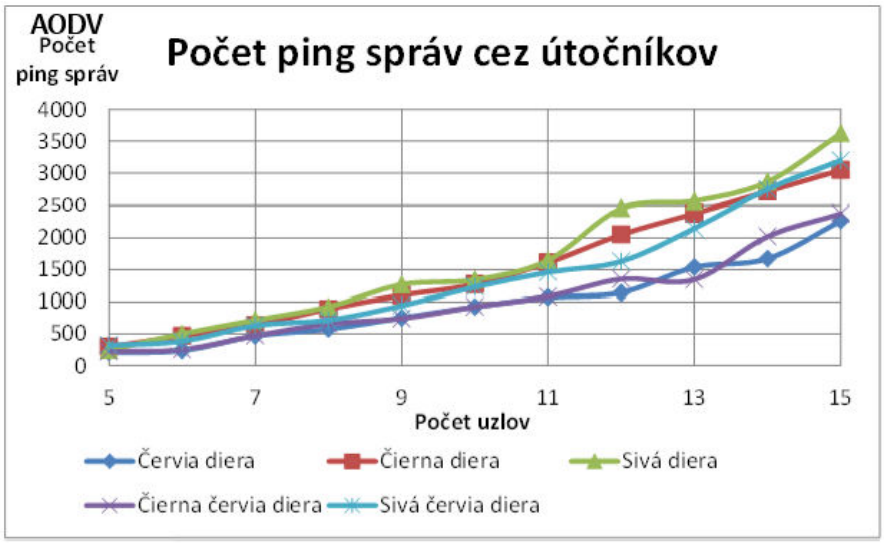
5.5 Testovací scenár čiernej červej diery

Hodnoty parametrov testovacieho scenára čiernej červej diery pre všetky testované protokoly môžeme vidieť v tab. 5.6.

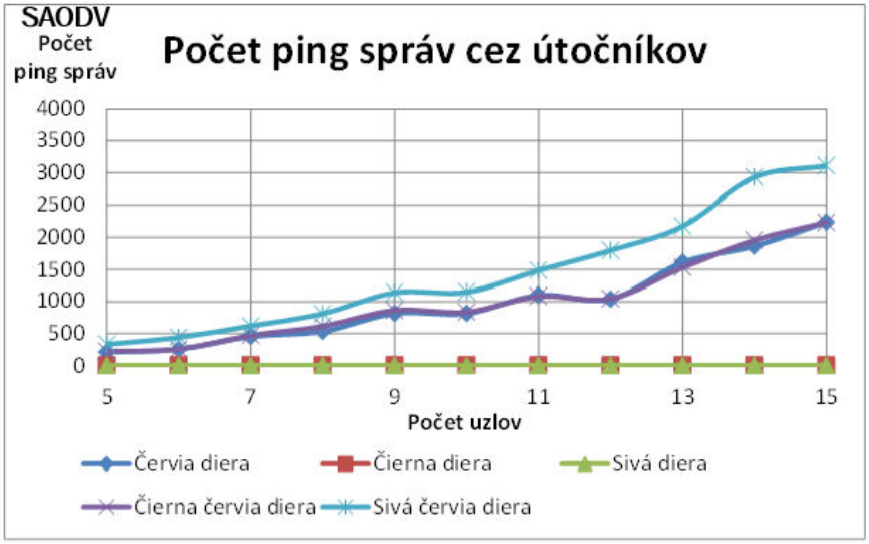
Tab. 5.7 Hodnoty parametrov scenára čiernej červej diery

Parameter	Hodnota
max_x_position	1100
max_y_position	800
drop_percentage	20

Grafy jednotlivých sledovaných parametrov môžeme vidieť na obrázkoch 5.6 a 5.7. Režimy, ktoré daný parameter neobsahovali alebo bola hodnota daného parametra nulová nie sú v grafoch zobrazené. V žiadnom z grafov nie je použitá logaritmická mierka

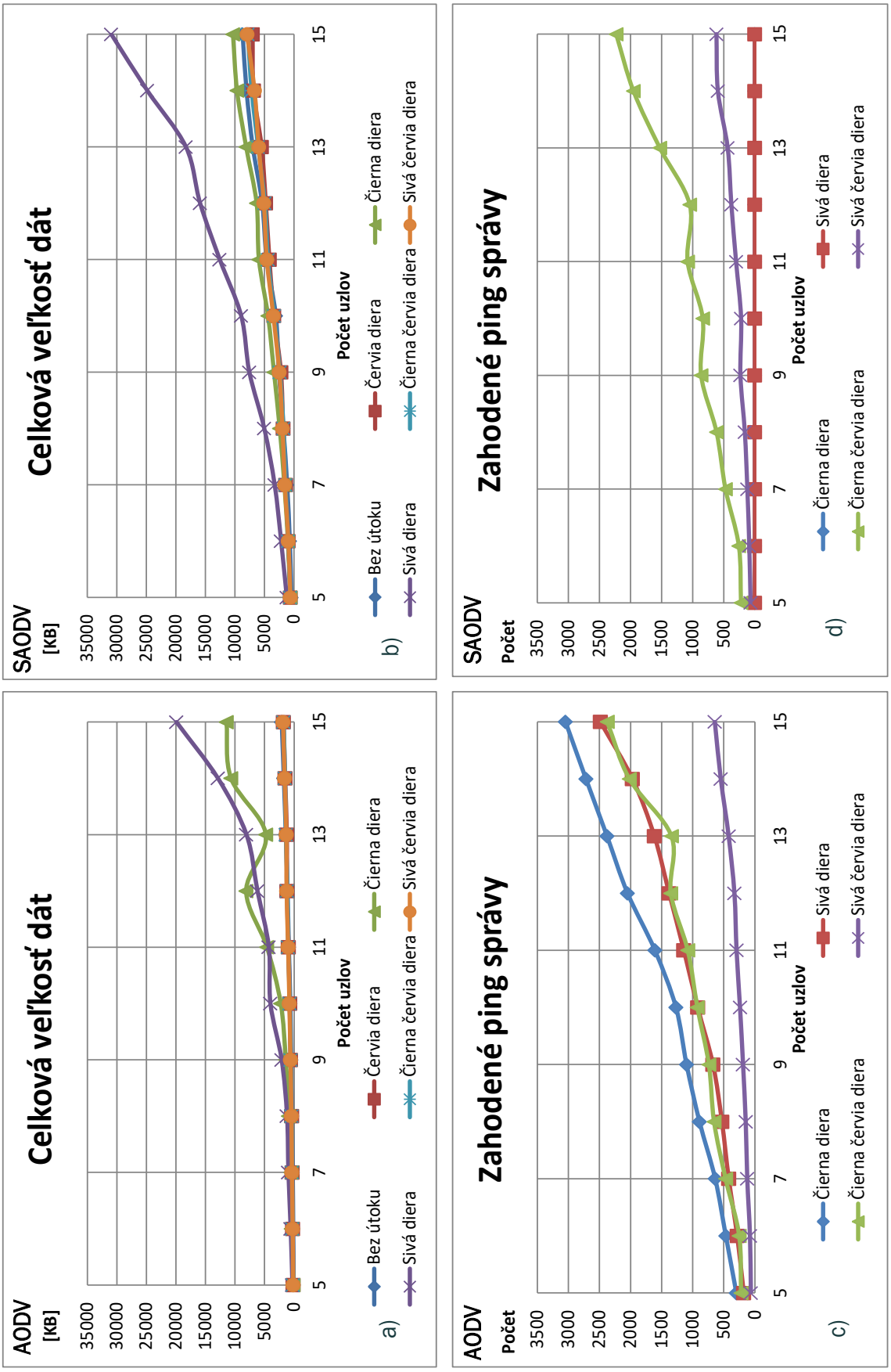


Obr. 5.11 Počet ping správ cez útočníkov v protokole AODV



Obr. 5.12 Počet ping správ cez útočníkov v protokole SAODV

Obr. 5.13 Výsledky scenára čierna červia diera pre protokoly AODV a SAODV (1)



Obr. 5.14 Výsledky scenára čierna červia diera pre protokoly AODV a SAODV (2)

5.5.1 Vyhodnotenie scenára čiernej červej diery

Scenár čiernej červej diery tvorilo až päť útočných režimov a režim bez útoku. Jednotlivé režimy boli nasledovné:

- Režim 0 - bez útoku
- Režim 1 - červia diera
- Režim 2 - čierna diera
- Režim 3 - sivá diera
- Režim 4 - čierna červia diera
- Režim 5 - sivá čierna diera

Na základe grafov sledovaných parametrov vieme usúdiť rozličné vlastnosti jednotlivých režimov. Najdôležitejším sledovaným parametrom pri tomto scenári bola priemerná úspešnosť ping správ. Podľa výsledkov bolo poradie režimov v protokole AODV v efektívnosti nasledovné : 2, (3,4), 0, 5, 1. Režimy 2 a 3 boli detegované protokolom SAODV čo môžeme usúdiť na základe podobných hodnôt s režimom bez útoku. V protokole AODV dopadli režimy nasledovne.

Najväčšiu účinnosť mala čierna diera z dôvodu jej povahy. Uzly generovali falošné správy, čím spôsobili smerovanie do ostatných uzlov cez seba a následne zahadzovali všetky prichádzajúce ping správy.

Na druhom mieste skončili režimy 3 a 4 pričom oba používali rozdielny princíp ovplyvňovania cesty. Režim 3 používal princíp čiernej diery, ale prichádzajúce ping správy zahadzoval s pravdepodobnosťou 20 %. Ak zarátame cestu ping správy tam a späť celková pravdepodobnosť zahodenia bola 36 %. Z tohto dôvodu boli niektoré ping správy cez sivú dieru úspešné a niektoré nie, čo sa odrazilo na priemernej úspešnosti doručenia ping správ. Ak vezmeme do úvahy detekciu režimu 3 v protokole SAODV, tak bol práve náš navrhnutý útok v tomto parametri najúspešnejší.

Režim 4 využíval náš navrhnutý spôsob ovplyvnenia cesty pomocou červej diery a následne zahadzovanie systémom čiernej diery. Detekcia takéhoto prístupu je oveľa náročnejšia ako v prípade režimov 2 a 3 keďže negenerujeme žiadne falošné správy. Zahadzovaním všetkých správ cez takto ovplyvnené cesty sme sa dostali na hodnoty podobné režimu 3.

Na treťom mieste sa umiestnil režim bez útoku. Je to z toho dôvodu, že za ním nasledujú už len režimy založené na červej diere, ktoré vďaka zvýšenému pokrytiu siete

umožnili komunikáciu vzdialených uzlov, ktoré spolu nemohli v režime bez útoku komunikovať. Nie je to ale chyba režimov založených na červej diere. Vďaka tomuto prístupu sme síce umožnili komunikáciu viacerým uzlom, ale táto komunikácia prechádzala cez útočné uzly čo môžeme využiť na sledovanie komunikácie medzi uzlami, alebo aj na úpravu týchto správ.

Štvrté miesto patrí režimu 5 z rovnakého dôvodu ako pri režimoch 2 a 3. Správanie sivej diery ovplyvnilo úspešnosť kvôli 36 % šanci zahodenia, ale detekcia takéhoto útoku by mala byť náročnejšia ako v režime 4.

Najhoršie z pohľadu tohto parametra skončil režim 1. Zapríčinené to je už spomínaným zlepšením dostupnosti uzlov. Na druhej strane nám tento režim taktiež dovoľuje odpočúvať alebo meniť správy jednotlivých komunikácií.

Ďalším sledovaným parametrom bol priemerný počet skokov. Môžeme si všimnúť, že režimy založené na červích dierach mali očakávane priemernú hodnotu tohto parametra najnižšiu. Naopak, samotné režimy 2 a 3 vykázali v tomto parametri vyššie hodnoty ako pri režime bez útoku.

Celková veľkosť odoslaných dát v sieti sa výraznejšie v oboch protokoloch líšila v režime 3 z dôvodu nutnosti hľadania ciest inicializovaných škodlivými uzlami a viacnásobným posielaním ping správ z dôvodu ich náhodného zahadzovania. Veľkosť vygenerovaných dát bola očakávane vyššia v protokole SAODV.

Pri počte zahodených ping správ môžeme vidieť ktoré režimy úspešne zahadzovali správy. Hodnoty reflektujú graf úspešnosti doručenia ping správ pričom pri protokole AODV dosiahol najvyššie hodnoty režim 2. Na druhom mieste sa umiestnili režimy 3 a 4. Náš útok v režime 4 tak dosahuje identické parametre ako režim 3, ktorý používa generovanie falošných správ a tým je ľahšie odhaliteľný a nepoužiteľný pri niektorých zabezpečených verziách smerovacích protokolov ako môžeme vidieť na grafe protokolu SAODV.

Posledným sledovaným parametrom bol počet ping správ cez škodlivé uzly. V tomto parametri sme medzi jednotlivými režimami nezaznamenali veľmi výrazné odchýlky okrem detegovaných režimov v protokole SAODV.

Celkovo tak naše režimy 4 a 5 obstáli veľmi dobre. Hlavne režim 4, ktorý dosiahol takmer identické výsledky s režimom 3, ktorý ale využíva generovanie falošných správ a bol odhalený protokolom SAODV. Dosiahli sme tak dostatočnú silu útoku bez zjavného zásahu do siete.

5.5.2 Vyhodnotenie nesimulovaných protokolov

Protokol SRP nie je chránený proti žiadnemu z režimov v tomto scenári. Neobsahuje žiadny mechanizmus na detekciu čiernych alebo červích dier.

Protokol DSR bude zraniteľný v rovnakých režimoch ako protokol SAODV z dôvodu neschopnosti detekcie červích dier. Naše navrhnuté útoky tak budú účinné aj pri tomto zabezpečenom protokole.

Nezávisle od vlastníctva certifikátov škodlivými uzlami by bol protokol ARAN zraniteľný proti našim navrhnutým útokom pokiaľ by boli červie diery implementované v skrytom režime. Uzly by tak tieto škodlivé uzly nevideli a červie diery by dokázali vytvoriť zdanlivo krátke cesty na dlhé vzdialenosti. Oproti režimom so samostatnou čiernou a sivou dierou by ale bol odolný nezávisle od vlastníctva certifikátov.

5.6 Zhodnotenie simulácií

Pomocou simulácií navrhnutých testovacích scenárov sme dokázali určité výhody našich navrhnutých útokov oproti ich bežným implementáciám. Podrobné rozdiely sme zobrazili v grafoch jednotlivých sledovaných parametrov a výsledky sme podrobne popísali.

Útok žonglér spôsobil okrem pridania oneskorenia aj spotrebovanie určitej šírky pásma pričom vďaka viacerým uzlom dokázal zvýšiť aj priemerné oneskorenie správ v porovnaní s bežnou verziu útoku. Detegovaný bol v prípade zabezpečených protokolov režim s ovplyvňovaním mechanizmu hľadania ciest. Protokol ARAN je teoreticky odolný proti všetkým režimom tohto scenára.

REQ záplava vďaka viacerým uzlom a reagovaním len na určité sekvenčné čísla spôsobila vygenerovanie väčšieho množstva falošných REQ správ, ako pri bežnej verzii útoku. Tento rozdiel bol dosiahnutý najmä vytvorením akejsi slučky, kde škodlivé uzly reagovali na správy vygenerované inými škodlivými uzlami. Zabezpečené protokoly boli proti režimom odolné s výnimkou možnosti zaplavenia len určitej časti siete.

RER záplava úspešne docielila zahadzovanie správ v smerovacích tabuľkách uzlov čím sme dosiahli výrazný pokles doručenia správ v sieti. Rozdiely medzi dvoma našimi navrhnutými režimami boli z hľadiska sily útoku minimálne. Tak ako aj v prípade REQ záplavy boli tieto režimy neúčinné pri zanalyzovaných smerovacích protokoloch.

Veľmi dobré výsledky sme dosiahli pri čiernej červej diere, kde sme napriek negenerovaniu falošných správ dosiahli silu útoku sivej diery, ktorá tieto falošné správy využíva. Útok je tak rovnako silný ale s oveľa väčšou náročnosťou detekcie čo potvrdzuje aj naša simulácia protokolov AODV a SAODV. Teoreticky by proti nemu nemali byť

odolné ani protokoly SRP a ARAN. Preto je tento útok našim najúspešnejším navrhnutým útokom.

Dosiahli sme tak cieľ našej práce a vytvorili sme scenáre, ktoré dokážu dôkladnejšie otestovať zabezpečenie reaktívnych smerovacích protokolov MANET sietí, ako pri bežných prístupoch. Výborné výsledky sme dosiahli najmä s čiernou červiou dierou, ktorá nebola odhalená žiadnym zabezpečeným protokolom. Tieto scenáre tak môžu byť použité aj pri návrhu nových protokolov.

Zhodnotenie a záver

V našej diplomovej práci sme sa venovali návrhu a otestovaniu testovacích scenárov pre reaktívne protokoly MANET sietí. Predstavili a zanalyzovali sme základné vlastnosti MANET sietí ako aj možnosti ich aplikácie a problémy ich bezpečnosti.

Pri analýze smerovacích protokolov sme si z nezabezpečených protokolov vybrali protokoly DSR a AODV. Takisto sme vybrali aj tri zabezpečené verzie smerovacích protokolov, ktoré používajú rôzne metódy zabezpečenia. Ide o protokoly SAODV, SRP a ARAN. Všetky tieto protokoly sme medzi sebou porovnali z rôznych hľadísk. Útoky v MANET sieťach sme rozdelili podľa vrstiev modelu RM OSI. Najväčší dôraz sme kládli na útoky odohrávajúce sa na sieťovej vrstve v ktorej smerovacie protokoly pracujú. Na konci analýzy sme v tabuľke zhrnuli zabezpečenie zanalyzovaných smerovacích protokolov na zanalyzované útoky.

Špecifikácia stanovuje základné požiadavky pre testovacie scenáre v MANET sieťach. Na základe špecifikácie a analýzy sme navrhli štyri nové útoky, ktoré sme zakomponovali do nami vytvorených testovacích scenárov. Tieto útoky sú založené najmä na spolupráci medzi viacerými uzlami v sieti, čím dosahujeme väčšiu silu útoku a jeho náročnejšiu detekciu. Súčasťou návrhu sú aj koncepty bezpečnostných riešení pre detekciu nami navrhnutých útokov.

Vybrané smerovacie protokoly a útoky scenárov sme implementovali v simulátore OMNeT++. Tieto modely sú postavené nezávisle od dostupných aplikačných rámcov, ktorých časté a výrazné zmeny by mohli spôsobiť nefunkčnosť implementácie v novších verziách.

Jednotlivé testovacie scenáre sme odsimulovali a vyhodnotili na základe definovaných parametrov a porovnali rozdiely medzi jednotlivými útokmi v scenári ako aj medzi protokolmi navzájom. Na základe týchto simulácií sme dokázali účinnosť našich navrhnutých útokov a ich lepšie vlastnosti v porovnaní s bežne známymi útokmi. Obzvlášť by sme chceli upozorniť na útok nami vytvorenej čiernej červej diery, ktorý nebol odhalený žiadnym z analyzovaných zabezpečených protokolov pri dosiahnutí sily útoku sivej diery, ktorá ale bola odhalená zabezpečenými protokolmi.

Pokračovaním práce by mohlo byť rozpracovanie navrhnutých bezpečnostných konceptov s navrhnutím scenárov pre vyhodnotenie dopadu jednotlivých parametrov

týchto zabezpečení na úspešnú detekciu útokov. Následne by mohli byť tieto výsledky porovnané s inými existujúcimi bezpečnostnými riešeniami.

Prínos práce je vo vytvorení štandardizovaných testovacích scenárov pre vyhodnotenie a porovnanie navrhovaných bezpečnostných riešení. Testovacie scenáre obsahujú doteraz neexistujúce útoky pre otestovanie bezpečnosti reaktívnych smerovacích protokolov v MANET sieťach. Testované riešenia tak môžu byť overené v rovnakých testoch.

Vývoj v tejto oblasti dovolí používať čoraz výkonnejšie zariadenia, ktoré budú schopné používať náročnejšie a efektívnejšie mechanizmy zabezpečenia. S víziou využitia týchto sietí v rôznych sférach tak bude problematika bezpečnosti vždy aktuálnou témou.

Literatúra

[1] LIU, J., CHLAMTAC, I.: Mobile Ad-hoc networking with a view of 4G Wireless: Imperatives and Challenges. In: Mobile ad hoc networking, 2004, s. 13-64.

[2] Tyagi, S. et al.: Study Of Manet: Characteristics, challenges, application and security attacks. In: International Journal of Advanced Research in Computer Science and Software Engineering, vol. 3, 2013, no.5, pp. 252-257.

[3] RAJA, L., BABOO, S. et al.: An Overview of MANET: Applications, Attacks and Challenges. In: International Journal of Computer Science and Mobile Computing, vol. 3, 2014, no.1, pp 408-417.

[4] HONG, X., XU, K. et al.: Scalable routing protocols for mobile ad hoc networks. In: IEEE network, vol. 16, 2002, no.4, pp 11-21.

[5] ALSLAIM, M., ALAQEL, H. et al.: A comparative study of MANET routing protocols. In: e-Technologies and Networks for Development (ICeND), 2014, pp 178-182.

[6] JOHNSON, D., MALTZ, D.: DSR: the dynamic source routing protocol for multi-hop wireless ad hoc networks. In: Ad hoc networking, 2006, s. 1-25.

[7] JOHNSON, D., HU, Y.: The dynamic source routing protocol (DSR) for mobile ad hoc networks for IPv4. RFC 4728, 2007.

[8] JHAVERI, R., PATEL, A. et al .: MANET routing protocols and wormhole attack against AODV. In: International Journal of Computer Science and Network Security, vol. 10, 2010, no.4, pp 12-18.

[9] PERKINS, C., BELDING-ROYER, E.: Ad hoc on-demand distance vector (AODV) routing. RFC 3561, 2003.

[10] ZAPATA, M. et al.: Secure ad hoc on-demand distance vector routing. In: ACM SIGMOBILE Mobile Computing and Communications Review, vol. 6, 2002, no.3, pp 106-107.

[11] PAPADIMITRATOS, P., HAAS, Z.: The Secure Routing Protocol (SRP) for Ad Hoc Networks. INTERNET-DRAFT, 2002.

[12] SINGH, B., KUMAS, D. et al.: Jamming attack in MANET: A Selected Review. In: International Journal of Advanced Research in Computer Science and Software Engineering, vol. 5, 2015, no.4, pp 1264-1267.

[13] TSENG, F., CHOU, L. et al.: A survey of black hole attacks in wireless mobile ad hoc networks. In: Human-centric Computing and Information Sciences, vol. 1, 2011, no.4, pp 1-16.

[14] MUDGAL, R., GUPTA, R. et al.: Study of various wormhole attack detection techniques in mobile ad hoc network. In: Electrical, Electronics, and Optimization Techniques (ICEEOT), vol. 1, 2016, pp 3748-3754.

[15] SHRIVASTAVA, S. et al.: Rushing Attack and its Prevention Techniques. In: International Journal of Application or Innovation in Engineering & Management, vol. 2, 2013, no.4, pp 453-456.

[16] TANK, V., LATHIGARA, A. et al.: To Detect and Overcome Sinkhole Attack in Mobile Ad hoc Network. In: Communications on Applied Electronics (CAE), vol. 2, 2015, no.6, pp 1-5.

[17] RAJAKUMAR, P., PRASANNA, V.: Security Attacks and Detection Schemes in Manet. In: Electronics and Communications Systems (ICECS), 2014, s. 1-6.

[18] DANAILOV, Z.: Attacks on mobile ad hoc networks. Bochum: Ruhr-Universitat, 2012. 15 s. Seminararbeit.

[19] BISWAS, K.: Security threats in mobile ad hoc network. Blekinge Institute of Technology, 2007. 39 s. Master Thesis.

[20] TOMAR, P., SURI, P. et al.: A comparative study for secure routing in MANET. In: International Journal of Computer Applications, vol. 4, 2010, no.5, pp 17-22.

[21] SANZGIRI, K., DAHILL, B. et al.: A secure routing protocol for ad hoc networks. In: Network Protocols 10th IEEE International Conference, 2002, pp 78-87.

[22] KUMAR, S. et al.: Detection and Prevention of Jellyfish Attack in AODV Routing Protocol in MANET. In: IJSTE, vol.3, 2016, no.6, pp 36-38.

[23] KARLSSON, J., DOOLEY, L., PULKKIS, G. et al.: A new MANET wormhole detection algorithm based on traversal time and hop count analysis. In: Sensors, 2011, pp 11122-11140

Príloha A: Technická dokumentácia

A.1 Návod na inštaláciu simulátora

Návod je aplikovateľný pre 64-bitové verzie systémov Windows 7 a Windows 10. 32-bitové verzie nie sú podporované od verzie 5.0.

1. Stiahnite inštalátor OMNet++ z domovskej stránky (http://omnetpp.org). Uistite sa, že inštalátor je určený pre systém Windows (končí src-windows.zip).
2. Stiahnutý súbor obsahuje všetky potrebné súbory. Presuňte súbor do požadovaného priečinka, v ktorom chcete simulátor nainštalovať a rozbaľte súbor ZIP. Uistite sa, že cesta k súboru neobsahuje medzery (napr. Program Files).
3. Spustite migwenv.cmd v priečinku omnetpp-5.2 cez príkazový riadok, alebo dvojitým kliknutím. Otvorí sa konzola MSYS bash shell. Pri prvom spustení budete musieť čakať na extrahovanie priečinka tools.
4. Skontrolujte obsah súboru configure.user aby ste sa uistili, že obsahuje všetky potrebné nastavenia. Vo väčšine prípadov nie sú nutné žiadne zmeny tohto súboru.
5. Zadajte nasledovné príkazy
    - ./configure
    - Make
6. Po dokončení buildu môžete spustiť IDE. OMNeT++ obsahuje IDE založené na Eclipse. Spustite ho príkazom omnetpp.
7. Po prvom otvorení IDE sa zobrazí okno vyzývajúce na inštaláciu knižnice INET. Potvrdíme tlačidlom OK a počkáme na ukončenie inštalácie.
8. Následne klikneme na priečinok inet pravým tlačidlom a zvolíme *Project-> Build project.*
9. Po týchto krokoch je simulátor plne pripravený na prácu.

Príloha B: Obsah priloženého média

Priložené médium obsahuje súbory rozdelené do nasledovných adresárov:

- /Dokumentacia - elektronická forma dokumentu v rôznych formátoch
  - /Zdroje - elektronické verzie použitých zdrojov
  - /IIT_SRC - článok z IIT SRC
  - /EUCNC - článok z EUCNC
- /Omnet++ - zdrojové súbory a výstupy zo simulácií
  - /Zdrojove kody - zdrojové kódy jednotlivých implementovaných protokolov spolu so scenármi
    - /AODV
    - /SAODV
  - /Data - výstupy dát zo simulácií
    - /AODV
    - /SAODV
- /Programy - obsahuje inštalátor OMNeT++ vo verzii 5.1

Príloha C: Plán práce

V nasledujúcej tabuľke je zobrazený plán práce s konkrétnymi časťami a termínmi na ich splnenie.

Tab. C.1 Plán práce

Časť	Termín	Stav
Analýza MANET sietí	3.2.2017	Splnené načas
Analýza DSR	15.2.2017	Splnené načas
Analýza AODV	20.2.2017	Splnené načas
Analýza SAODV	25.2.2017	Splnené načas
Analýza SRP	1.3.2017	Splnené načas
Útoky na fyzickej vrstve	5.3.2017	Splnené načas
Útoky na linkovej vrstve	10.3.2017	Splnené načas
Útoky na sieťovej vrstve	15.3.2017	Splnené načas
Útoky na transportnej vrstve	20.3.2017	Splnené načas
Analýza ARAN	30.3.2017	Splnené načas
Útoky na aplikačnej vrstve	5.4.2017	Splnené načas
Návrh vlastných útokov	1.5.2017	Splnené načas
Dokončenie návrhu	1.10.2017	Nesplnené načas
Implementácia protokolu AODV spolu s testovacími scenármi	14.11.2017	Splnené načas
Vyhodnotenie simulácií a dokončenie DP2	12.12.2017	Splnené načas
Implementácia protokolu SAODV spolu s testovacími scenármi	1.2.2018	Časový plán bol posunutý na 10.3 a bol splnený v predstihu
Vyhodnotenie simulácií protokolu SAODV	7.2.2018	Časový plán bol posunutý na 1.4 a bol splnený v predstihu

Implementácia protokolu DSR spolu s testovacími scenármi	20.2.2018	Po konzultácii s vedúcim boli implementácie ostatných protokolov zrušené z dôvodu veľkého rozsahu práce. Zabezpečenie proti navrhnutým scenárom bolo vyhodnotené teoreticky na základe návrhov jednotlivých protokolov
Vyhodnotenie simulácií protokolu DSR	23.2.2018	Zrušené na základe zmienených skutočností
Implementácia protokolu SRP spolu s testovacími scenármi	1.3.2018	Zrušené na základe zmienených skutočností
Vyhodnotenie simulácií protokolu SRP	5.3.2018	Zrušené na základe zmienených skutočností
Implementácia protokolu ARAN spolu s testovacími scenármi	15.3.2018	Zrušené na základe zmienených skutočností
Teoretické vyhodnotenie nesimulovaných protokolov	1.4.2018	Splnené v predstihu
Návrh zabezpečenia proti útokom	7.4.2018	Splnené načas
Finalizácia dokumentu	20.4.2018	Splnené načas

Príloha D: Článok na konferenciu EUCNC

Testing Scenarios for MANET Routing Protocols

Martin Ilavský
Slovak University of Technology
Faculty of Informatics and Information Technologies
Bratislava, Slovakia
ilavskymartin@yahoo.com

Jozef Filipek
Slovak University of Technology
Faculty of Informatics and Information Technologies
Bratislava, Slovakia
xfilipekj1@is.stuba.sk

Abstract— **In this paper we propose new testing scenarios for evaluation of security in reactive MANET routing protocols. These scenarios contains new attacks that are more efficient and harder to detect than the basic attack types. Created scenarios were implemented in the network simulator and their impact on the network properties was evaluated.**

Keywords—MANET, testing scenarios, security, network attack

I. INTRODUCTION

MANET (mobile ad hoc networks) are nowadays one of the most evolving networks. The basic characteristic of MANETs is dynamic forming with independent mobile nodes, connected by various wireless technologies without previous network infrastructure and centralized administration. Nodes in these networks can move freely, which results in high dynamic topology changes. Their main disadvantages are limited bandwidth, dependency on energy sources and greater vulnerabilities to attacks [1] [2].

Existing routing protocols must be quickly adaptable to frequent topology changes, while saving up limited computing resources. In general, they use link state or distance vector algorithms. We can divide them into proactive (link state) and reactive (distance vector) protocols based on the type of used algorithm [3]. In our paper, we focus on reactive protocols which nodes request path information only when it is necessary. There is no need for periodical flooding of topology requests. Examples of reactive routing protocols are AODV and DSR.

Several methods were proposed to secure these routing protocols. Their prevention mechanisms can be based on asymmetric cryptography (ARAN [4]), symmetric cryptography (SRP [5]), one-way hash chains (SAODV [6]) or hybrid principles (Ariadne [3]). Solutions of SRP and SAODV proposed extended secure header which can be used by other non-secure routing protocols, while ARAN and Ariadne are standalone protocols.

Security proposals are not standardized tested so their comparison based on their papers can be difficult. Therefore we proposed new attacks which can thoroughly verify proposed solutions. These attacks are part of the created testing scenarios which includes various modes of attack types that are focusing to influence the same network properties. Security proposals can be therefore compared based on these created scenarios.

II. RELATED WORK

Scenarios used in individual evaluations are often simplified and not very detailed. Proposals of SAODV [6] and SRP [5] contain only security assumptions without any evaluations. Authors in [7] were evaluating proposed mechanism on AODV protocol with only specific scenario for detection of blackhole. Other types of attacks focusing route discovery processes were not tested. Solution was verified with packet delivery, packet loss and throughput metrics.

Best detailed analysis of proposed security mechanism was done by authors of ARAN [4]. Protocol proposal was tested on attack types as modifications of protocols fields, tunnelling, spoofing and fabrication. Simulations were performed with 20 and 50 nodes with random initial positions. Results were evaluated using several metrics, such as average path length, routing load, average delay etc.

III. BASIC PRINCIPLES OF REACTIVE ROUTING PROTOCOLS

In general, reactive routing protocols are using three types of messages. Route request (RREQ), route reply (RREP) and route error (RERR).

RREQs and RREPs are used in the route discovery process. If node A wants to send data to node B and routing table of node A does not contain already created route, or this route has expired, then node A initializes route discovery process with broadcasting RREQ message. Every node, except destination node B, that receives this message rebroadcasts it to the network and saves the backward route to node A. Special field called broadcast id in RREQ message is often used for loop prevention, when node drops this type of message with already seen value of broadcast id.

Node B after receiving first RREQ sends unicast RREP via already created backward route. Other received RREQs are dropped. This ensures that created route is the fastest and not the shortest one. When node A receives RREP, route is established and communication between these two nodes can begin.

RERR messages serves as notifications of topology changes. They can be generated in multiple situations to cause reinitialization of the route discovery process. For example, when transmitting node detects unavailability of receiving node , data packet with unknown route to destination is received or other situations defined in the specific routing protocols.

IV. CREATED ATTACKS AND TESTING SCENARIOS

Created attacks are based on the use of communication between multiple cooperating malicious nodes.

Each testing scenario consists of multiple attacks called modes. Modes in scenario share common goal, such as prevention of regular message delivery. Each scenario contains known basic attacks as well as ours proposed attacks.

A. *Juggler attack*

The goal of the juggler attack is to cause delays in communication between nodes, consumption of the available bandwidth and increase load on nodes in the network. Messages will be delivered to the destination, but added delay can cause problems for different applications. Amount of added delay can be controlled. The assumptions for successful attack are:

Existence of route from source to destination node through malicious node

Existence of at least two malicious collaborating nodes

Attack consists of the three following steps:

Perform an attack to ensure routing to destinations nodes through malicious node (Optional).

Add delay by creating temporary communication loop between cooperating malicious nodes.

Deliver a message with a defined delay.

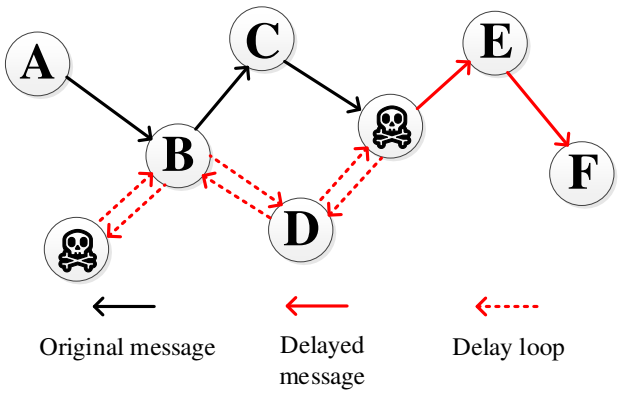
For first optional step, black hole mechanism to influence routing discovery process using modified RREQs can be used. These modified messages can contain very low hop count value and therefore will malicious node be chosen as a next hop for the destination node. More routes via malicious node will cause greater impact of the attack on the network.

Second step adds defined delay to the communication between source and destination node. Our solution exploits the existence of other malicious nodes in the network. Instead of the sending of received message to another node in the created route, original message is wrapped to another message which is send to randomly selected cooperating malicious node. Malicious node that received this message will send it again to other malicious node in the network. This process creates loop that adds delay to the original message and also consumes part of limited network resources.

The initial malicious node that created delay loop checks if the defined delay has been achieved (or count of exchanges between nodes). Once this condition is met, original message without any modification can be sent via defined next hop. Then a stable addition of delay in controlled communications will be achieved.

It is crucial to maintain active routes between cooperating malicious nodes in the network. It is advisable to periodically initiate route discovery process between these nodes to ensure stable connection.

Example of this attack can be seen in the Figure 1 where node A wants to send data message to node F.



Juggler attack with two malicious nodes

1) *Testing scenario of juggler attack*

In the case of a juggler attack, scenario consists of the three modes whose influence on the selected network parameters will be compared with other modes.

First mode adds static delay without creating delay loop. Malicious node after receiving data message from the basic nodes will hold this message for defined time interval and then send it to the next hop.

Second mode uses described juggler attack without the first optional step. Malicious nodes does not create any modified messages to exploit route discovery processes.

Third mode consists of juggler attack with first optional step. Malicious nodes will try to exploit route discovery processes to influence path selection.

2) *Monitored network parameters of of juggler testing scenario*

Modes will be evaluated within the following monitored network parameters:

Average ping delay

Total size of data transmitted in network

Number of messages, that were delayed

Number of messages, that were not delayed

Number of messages, exchanged between malicious nodes

B. *Black-wormhole attack*

By combining the blackhole and wormhole attack we can create a very strong attack that can offset the imperfections of single blackhole mechanism. Attack consists of two following rules:

1. If received message is used in the route discovery process, then act as wormhole.

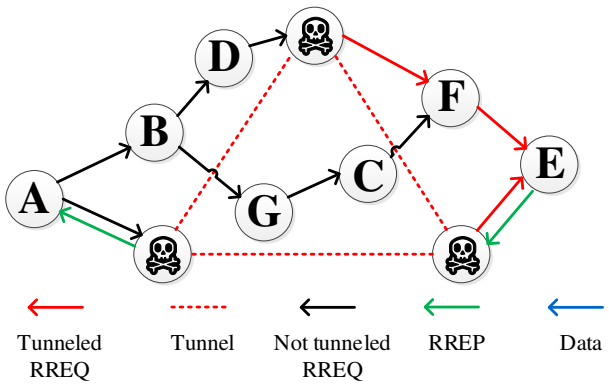
2. If received message is data message, then act as a blackhole.

Blackhole uses modified fake messages with low hop count values to pretend that the shortest path to destination node leads via this malicious node. All received data messages are then dropped. This mechanism is inefficient in cases where the modification of hop count values are not possible for reasons of secure routing protocols or when protocol does not use this field at all.

To overcome limitations of single blackhole attack, we propose to use wormholes to create seemingly shortest and fastest path between ordinary nodes in network. Wormholes are using high speed connection between malicious nodes to speed up the delivery of messages in the network. Ordinary nodes will than think that they are closer than they really are. RREQs will be then faster delivered via this created tunnel, which will result in creating routes via malicious nodes. After creation of these routes we can switch nodes to blackhole or greyhole (random drop) mode.

For the most difficult detection of the attack we can use wormholes in hidden modes, so they will be invisible to other nodes in the network. These nodes should be far from each other to cover as much space in network as possible.

Example of black-wormhole attack with three collaborating malicious nodes can be seen in the Figure 2. Node A wants to send data to node E but the route is not established yet, so it starts route discovery process. Initial RREQ will be tunnelled via wormhole, so the first message that node E will receive will be through our malicious nodes. RREP message will be generated and sent back via our wormhole. When node A starts to transmit the data through this new created route, malicious node will drop them and prevent any communication between these two nodes.



Black-wormhole attack with three collaborating nodes

1) Testing scenario of black-wormhole attack

In the case of a black-wormhole attack, scenario consists of the five modes whose influence on the selected network parameters will be compared to each other.

First mode uses nodes only in wormhole mode which will create shortest paths without the need of message modification. Anonymous mode in which these nodes are not visible to ordinary nodes is recommended.

Nodes in second mode are working only as regular blackholes which are generating fake messages to exploit route discovery processes. All messages routed via these nodes will be dropped.

Third mode uses only grey holes instead of black holes. All messages routed via these nodes are randomly dropped. Some of them will therefore be delivered and some not.

Fourth mode is based on our proposed black-wormhole attack which was described above.

The last fifth mode adjusts the fourth mode to use a greyholes instead of blackholes.

2) Monitored network parameters of of black-wormhole testing scenario

Modes will be evaluated within the following monitored network parameters:

Average percentage of successful pings

Average hop count

Number of exploited route discovery processes

Number of dropped messages

Total size of data transmitted in network

C. RREQ flood

Reactive protocols often use fields with sequence numbers (SEQ num) for loop prevention. Only first received RREQ message is processed and others are dropped based on the saved received sequence number. Basic RREQ floods are transmitting these messages with modified SEQ num field to prevent their messages from dropping and causing all nodes to rebroadcast these messages to the whole network.

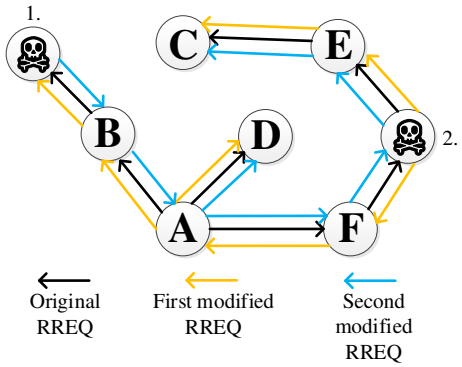
Our proposal of this attack uses malicious nodes, which are generating new RREQ messages with increased SEQ num value after receiving. But to prevent generation of same SEQ nums and to ensure almost infinite RREQ generating loop (under the condition of path existence between collaborating nodes), all malicious nodes in network will generate new message on exceptionally excluding conditions. Generally, if we have n nodes, then i-th node will generate new message when SEQ num s fulfils condition

$$s \bmod n = i - 1 \qquad (1)$$

Nodes thus generate messages with the modification that causes new message to be generated on other malicious node. Because the most common defense against this type of attack is to limit the number of RREQ messages generated by single node over a certain time period, we will be able to generate n-1 times more messages in a given time period compared to single node attack.

Example of our proposed attack can be seen in the Figure 3 with two collaborating nodes, where malicious node 1 reacts only to the even SEQ nums and malicious node 2 reacts to the odd SEQ nums. Original RREQ message is also rebroadcasted to the network. Only those messages that were not dropped by

loop prevention mechanism are displayed. Figure shows only two of theoretically infinite generated messages.



RREQ flood with two collaborating nodes

1) Testing scenario of RREQ flood

RREQ flood scenario consists of the two modes whose influence on the selected network parameters will be compared with other modes.

First mode uses only one malicious node which reacts to all received RREQ messages. This mode should be limited by maximum RREQs allowed by routing protocol.

Second mode uses two collaborating nodes. In total, they should respond to the same number of messages as one node in first mode, but as already mentioned, due to number of RREQs limitations, we will be theoretically able to generate two times more messages in a given time period than in the first mode. Malicious nodes should be located at a distance, which does not allow their direct communication.

2) Monitored network parameters of of RREQ flood testing scenario

Modes will be evaluated within the following monitored network parameters:

Total size of the transmitted data

Number of generated RREQs messages by malicious nodes

Average ping success rate

Number of the initiated route discovery processes

Number of the RREQs messages received by malicious nodes

a. EVALUATION

Selected routing protocols and proposed scenarios were implemented in OMNeT++ network simulator. All scenarios were tested on the network with 5 - 15 nodes which could move randomly. Every simulation was repeated five times to achieve average results. Every node tried five times to ping all other nodes in network during 60 second simulation. Values of network and routing protocol parameters can be seen in the Table I.

PARAMETERS OF NETWORK SIMULATIONS

Parameter	Value
Number of ping retransmits after failure	*5*
Transmission range	*300 m*
Ping retransmit wait	*500 ms*
Transmit delay	*10 ms*
Speed of nodes	*10 m/s*

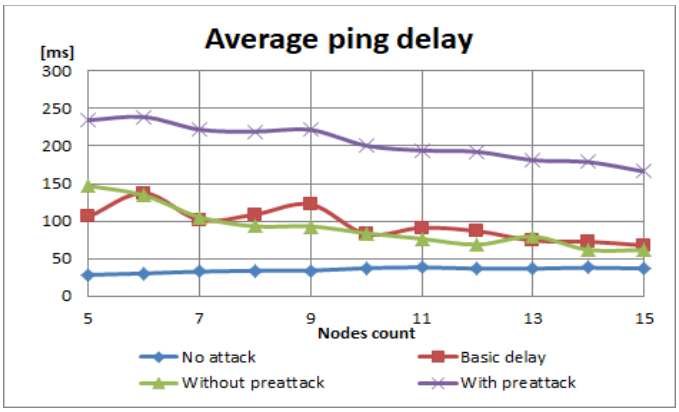
This paper presents evaluation results of the AODV protocol. Values of parameters for this protocol can be seen in the Table II.

PARAMETERS OF AODV PROTOCOL

Parameter	Value
Routing table lifetime	*3 s*
Routing table lifetime update interval	*1 s*

A. Juggler scenario evaluation

Results of the scenario simulation can be seen in the Figure 4. Each attack was successful and added certain delay to ping messages. Best results were achieved with our proposed attack which used first optional step. Other modes achieved significantly lower results in this metric.

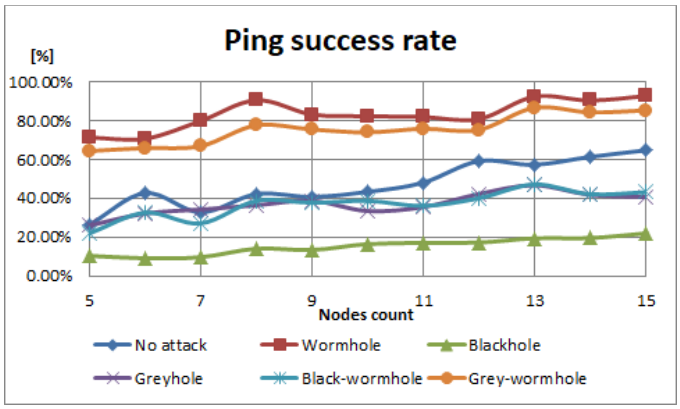


Average ping delay in Juggler scenario

Difference between basic delay and without pre-attack modes were noticed in total data transmitted metric, where mode without optional step generated 1 266 % more data than basic delay mode, which was not generating any additional data compared to simulation without attack. Attack with optional step achieved best results with 13 653 % more data generated than basic delay attack.

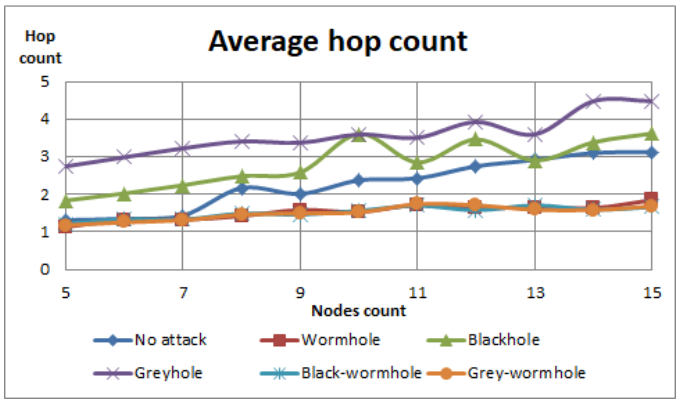
B. Black-wormhole scenario evaluation

Results of the scenario simulation can be seen in the Figures 5 and 6. Basic blackhole achieved lowest ping successful rate of all modes. Our proposed black-wormhole attack achieved same results as greyhole. This is very interesting because we did not generate any fake messages to exploit route discovery mechanism and therefore is our proposed attack much harder to detect. Modes based on wormholes achieved higher successful rate due to the nature of this attack which increased availability of the nodes in the network. This is not bad, because we could also use these attacks to eavesdrop network communication.



Ping success rate in Black-wormhole scenario

Lowest hop count values were achieved in modes based on wormholes with values from 1-2 hops (5-15 nodes) followed by mode without attack (1-3 hops), blackhole (2-3 hops) and greyhole (3-5 hops).



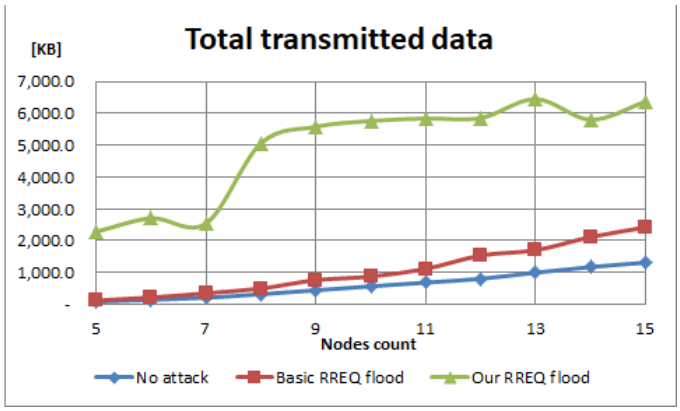
Average hop count in Black-wormhole scenario

C. *RREQ flood scenario evaluation*

Results of the scenario simulation can be seen in the Figure 7. Total size of transmitted data was significantly higher in our proposed attach with collaborating malicious nodes compared to the attack with basic RREQ flood. Our attack has transmitted up to 6 times more data in simulations with 10 regular nodes.

Total count of generated RREQs messages was also higher in our proposed mode thanks to created loop and doubled limit for RREQs.

Both attacks had impact on ping success rate. Our proposed mode achieved an average of 54 %. First mode also had impact on this monitored parameter, but it was not as large as in second mode with average of 75 %.



Total transmitted data in RREQ flood scenario

VI. CONCLUSION

In this paper we proposed testing scenarios for MANET routing protocols with three new attacks based on collaboration between multiple malicious nodes.

New proposed attacks are described in detail with assumptions for successful execution, individual steps and example diagrams. Scenarios are defined by multiple modes with specified monitored network parameters.

MANET routing protocols can therefore be compared with each other using our testing scenarios which were not created with focus on specific attacks that should be detected by proposed solutions. Scenarios contain also standard and non-standard attacks for thorough protocol evaulation and comparison.

REFERENCES

LIU, J., CHLAMTAC, I.: Mobile Ad-hoc networking with a view of 4G Wireless: Imperatives and Challenges. In: *Mobile ad hoc networking*, (2004), pp. 13-64.

Tyagi, S.: Study Of Manet: Characteristics, challenges, application and security attacks. In: *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 3, (2013), no.5, pp. 252-257.

ALSLAIM, M., ALAQEL, H.: A comparative study of MANET routing protocols. In: *e-Technologies and Networks for Development (ICeND)*, (2014), pp 178-182.

SANZGIRI, K., DAHILL, B.: A secure routing protocol for ad hoc networks. In: *Network Protocols 10th IEEE International Conference*, (2002), pp 78-87.

PAPADIMITRATOS, P., HAAS, Z.: The Secure Routing Protocol (SRP) for Ad Hoc Networks. INTERNET-DRAFT, (2002).

ZAPATA, M.: Secure ad hoc on-demand distance vector routing. In: *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 6, (2002), no.3, pp. 106-107.

TANK, V., LATHIGARA, A.: To Detect and Overcome Sinkhole Attack in Mobile Ad hoc Network. In: *Communications on Applied Electronics (CAE)*, vol. 2, (2015), no.6, pp. 1-5