mgr Arkadiusz Smolarek
WASKO S.A. Gliwice
e-mail: arkadiusz.smolarek@gmail.com

dr inż. Tomasz Malinowski
Warszawska Wyższa Szkoła Informatyki, Warszawa
e-mail: tmalin@poczta.wwsi.edu.pl

PROTOKOŁY TRASOWANIA W SIECIACH AD HOC

Routing protocols in ad hoc networks

Streszczenie

W artykule dokonano charakterystyki protokołów trasowania stosowanych w bezprzewodowych sieciach Ad Hoc, a także zilustrowano na drodze symulacji komputerowej zachowanie wybranych protokołów trasowania. Badania porównawcze zrealizowane zostały w środowisku symulacyjnym ns-2.

Słowa kluczowe: protokoły routingu, sieci ad hoc, Manet, symulacja.

Abstract

The article presents the characteristics of routing protocols used in wireless ad hoc networks, and is illustrated through computer simulation of the behavior of some routing protocols. Comparative studies were carried out in an ns-2 simulation environment.

Keywords: routing protocols, ah hoc networks, Manet, simulation.

1. WSTĘP

Powszechna instalacja w komputerach przenośnych, tabletach czy nawet telefonach komórkowych urządzeń do komunikacji sieciowej ma znaczący wpływ na rozwój technologii związanych z bezpośrednią komunikacją pomiędzy różnymi urządzeniami. Wykorzystanie sieci Ad Hoc nie jest już ograniczone do badań naukowych czy zastosowań militarnych. Przy obecnej, rosnącej potrzebie łączności bezprzewodowej, sieci Ad Hoc stają się jednym z ważniejszych obszarów badań odkrywających możliwości rozwoju oraz praktycznego wykorzystania tego typu rozwiązań. Pomimo stale rozwijającej się w wielu miejscach infrastruktury sieci bezprzewodowych opartych na punktach dostępowych, potrzeba wzajemnej i bezpośredniej komunikacji stawia nowe wyzwania projektanckie.

Jednym z bardziej istotnych elementów architektury sieci Ad Hoc są protokoły trasowania. Ograniczenia urządzeń mobilnych wymuszają poszukiwanie jak najbardziej efektywnych rozwiązań w zakresie wyznaczania tras dla pakietów. Protokoły trasowania w sieciach Ad Hoc powinny być jak najprostsze, szybkie i w sposób minimalny wykorzystujące dostępne zasoby, zarówno sprzętowe jak i te dotyczące dostępnej szerokości pasma. W przeciwieństwie do sieci przewodowych protokół trasowania w sieciach Ad Hoc musi między innymi sprostać takim ograniczeniom jak wąskie pasmo, łącza jednokierunkowe, duża dynamika topologii czy ograniczone zasoby sprzętowe i energetyczne.

W artykule dokonano charakterystyki protokołów trasowania w bezprzewodowych sieciach Ad Hoc, a także zilustrowano na drodze symulacji komputerowej zachowanie wybranych protokołów trasowania.

2. TRASOWANIE W SIECIACH AD HOC

Sieć Ad Hoc jest zbiorem mobilnych urządzeń bezprzewodowych tworzących dynamicznie tymczasową infrastrukturę sieciową bez centralnego punktu administracji. Dość często są to wieloskokowe struktury, gdzie na odcinkach pomiędzy poszczególnymi węzłami mogą wystąpić bardzo niskie przepustowości, a komunikacja może odbywać się tylko w jednym kierunku. Węzły w takiej sieci są nie tylko odbiornikami informacji, ale również mogą pełnić funkcję urządzeń przesyłających dane do innych węzłów [1]. Pierwotnie badania nad sieciami Ad Hoc prowadzono pod kątem wykorzystania tego typu rozwiązań w obszarze militarnym. Jednak rosnąca potrzeba wzajemnej komunikacji urządzeń nie będących w bezpośrednim zasięgu nadajników sieci szkieletowej zaowocowała powołaniem przy organizacji IETF grup MANET (Mobile Ad Hoc Networking) oraz NEMO (Network Mobility). Zadaniem obu grup jest badanie i standaryzacja protokołów trasowania IP pod kątem użycia w topologiach o charakterze zarówno statycznym jak i dynamicznym, zbudowanych z różnych odmian sprzętu wykorzystującego komunikację bezprzewodową [2].

W tabeli 1 wymieniono główne cechy sieci Ad Hoc.

Tabela 1. Wybrane cechy charakterystyczne sieci Ad Hoc [3].

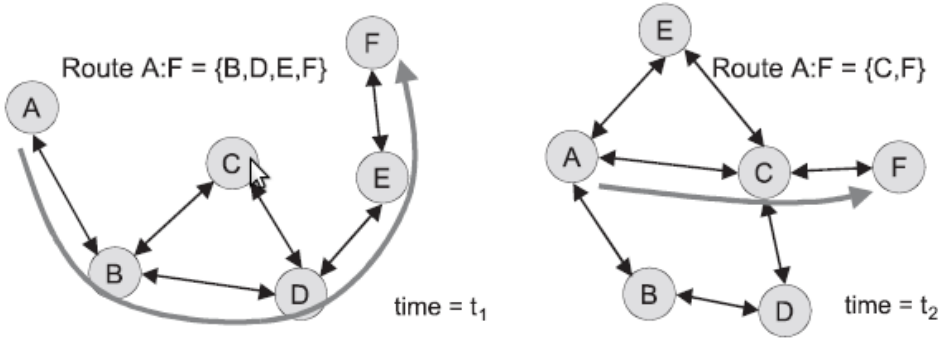
Cecha	Charakterystyka
Mobilność	Dynamicznie zmieniające się pozycje węzłów. Nieustalona liczba węzłów wchodzących w skład danej sieci.
Multihopping	Trasa od źródła do celu prowadzi przez kilka węzłów. Liczba przeskoków może się zmieniać w krótkim okresie czasu.
Samoorganizacja	Sieć Ad Hoc musi niezależnie określić swoje parametry konfiguracyjne, takie jak routing, pozycja, kontrola zasilania.
Oszczędzanie energii	Wiele urządzeń tworzących sieci Ad Hoc posiada ograniczone źródło zasilania. Wykorzystanie zoptymalizowanych pod kątem energetycznym protokołów pozwala wydłużyć czas pracy tych urządzeń.
Skalowalność	Charakter sieci Ad Hoc pozwala na dynamiczne budowanie sieci, które zawierać mogą duże ilości węzłów.
Bezpieczeństwo	Ze względu na swój charakter sieci Ad Hoc są jednym z najbardziej podatnych na ataki środowisk sieciowych.

Ogromnym wyzwaniem w sieciach Ad Hoc jest wyznaczanie i utrzymanie trasy dla pakietów. Trasa pakietu w środowisku sieci Ad Hoc może przebiegać przez jeden lub więcej węzłów, przy czym topologia sieci podlega dynamicznym zmianom, często w sposób nieprzewidywalny [4]. Protokół trasowania w sieci Ad Hoc powinien być automatycznie uruchamiany wedle potrzeby i wyznaczać wolną od pętli trasę do żądanego punktu docelowego. Powinien dostarczać mechanizm zapewniający jak najszybszą zbieżność sieci w warunkach często zmieniającej się topologii. Poza tym przy wykonywaniu wszystkich swoich zadań protokół powinien mieć minimalne zapotrzebowanie na zasoby sprzętowe urządzenia oraz pasmo [5]. Główne problemy dotyczące wyznaczania i utrzymania tras dla pakietów w sieciach Ad Hoc to:

- Szybko, dynamicznie i nieprzewidywalnie zmieniająca się topologia,
- Niska dostępność pasma,
- Brak centralnego zarządzania,
- Duża rozpiętość sieci,
- Występowanie łączy jednokierunkowych,
- Niska skalowalność.

W sieciach Ad Hoc każdy węzeł należący do sieci może brać udział w przekazywaniu pakietów. Dzięki

wykorzystaniu specjalnych protokołów węzeł może znacznie wydajniej monitorować najbliższą okolice i adaptować się do zmieniających się warunków. Rysunek 1 przedstawia prosty przykład, w którym w celu prawidłowego przesłania danych konieczne jest wykorzystania protokołów trasowania, potrafiących dynamicznie dostosować się do zmiennych warunków panujących w sieci. W czasie t1 najkrótsza trasa od węzła A do F wiedzie przez węzły B, D i E. W kolejnej jednostce czasu t2 topologia ulega zmianie, co powoduje konieczność ponownego wyznaczenia trasy dla pakietów biegnących od źródła A do celu F.



Rys. 1. Proces dynamicznego wyznaczania trasy w zmieniającej się topologii [6].

Podobnie jak w przypadku sieci przewodowych, protokół trasowania ma zapewnić drogę dla pakietów z węzła źródłowego do węzła będącego punktem docelowym. Biorąc jednak pod uwagę wymienione wcześniej ograniczenia sieci Ad Hoc, realizacja tego zadania jest o wiele większym wyzwaniem technologicznym.

Dostępnych jest wiele różnych protokołów trasowania, które mogą być zaimplementowane w mobilnych sieciach Ad Hoc. Istniejące rozwiązania można sklasyfikować jako [6]:

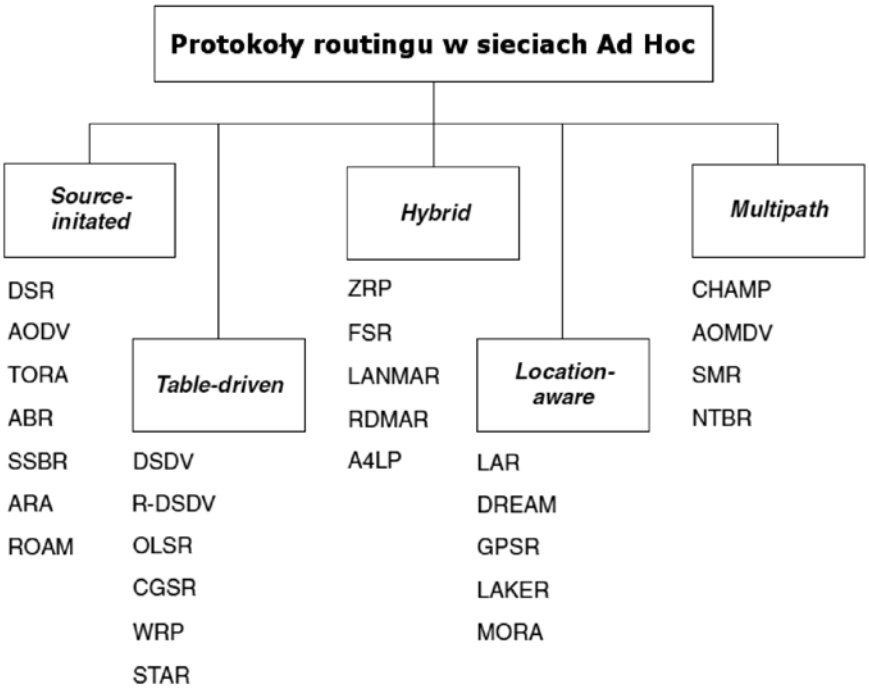
- Protokoły proaktywne (ang. proactive) – w każdym węźle utrzymywane są zawsze możliwie najświeższe informacje na temat tras do pozostałych węzłów. Trasy przechowywane są w tablicach routingu, które są regularnie aktualizowane. Protokoły z tej kategorii nie są zalecane dla dużych środowisk z bardzo dynamicznie zmieniającą się topologią,
- Protokoły reaktywne (ang. reactive) – znane również jako protokoły trasowania na żądanie (on-demond). Jest to klasa protokołów, w których trasa wyznaczana jest w momencie, gdy węzeł źródłowy potrzebuje przesłać pakiety do określonego celu. Sieć zalewana jest specjalnymi pakietami żądania trasy, począwszy od najbliższych węzłów sąsiadujących ze źródłem. Po zakończeniu procedury wyznaczania trasy jest ona utrzymywana do zakończenia jej wykorzystywania,
- Protokoły hybrydowe (ang. hybrid) – protokoły te łączą właściwości protokołów proaktywnych i reaktywnych. W większości protokołów z tej grupy działanie polega na podziale sieci na mniejsze fragmenty, a węzły utrzymują tablice tras dla tych wydzielonych obszarów.

Dodatkowo, w literaturze podawane są jeszcze dwie kategorie protokołów, a mianowicie protokoły geograficzne i energooszczędne. Wykorzystują one odpowiednio rozgłaszania określone przez położenie geograficzne (ang. geocast) i transmisje przez kanał wielościeżkowy (ang. mulitpath) [7].

Bazując na metodzie dostarczania danych, protokoły trasowania w sieciach Ad Hoc można sklasyfikować w następujący sposób:

- Unicastowe protokoły trasowania – protokoły przesyłające informacje do pojedynczego miejsca docelowego z pojedynczego źródła,
- Multicastowe protokoły trasowania – protokoły dostarczające informacje do grupy odbiorców jednocześnie z użyciem wybranej, efektywnej strategii wykorzystania struktury sieci [2].

Rysunek 2 przedstawia kolejny sposób klasyfikacji protokołów trasowania stosowanych w sieciach Ad Hoc. Szczegółową charakterystykę wymienionych tutaj grup protokołów znaleźć można w [7].

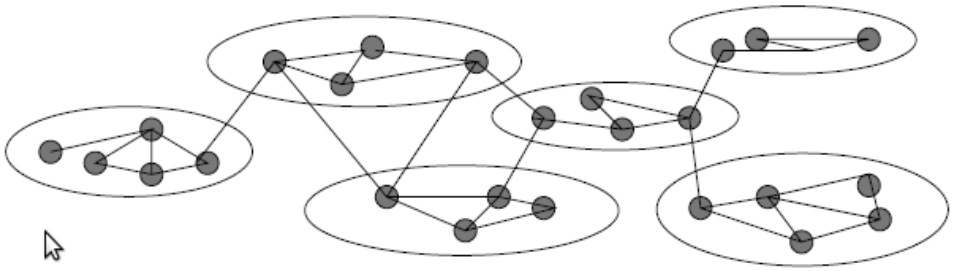


Rys. 2. Podział protokołów trasowania w sieciach Ad Hoc [7].

Architektura trasowania w sieciach samoorganizujących Ad Hoc może być płaska lub hierarchiczna. W płaskiej architekturze każdy z węzłów należących do takiej sieci jest niezależnym routerem identyfikowanym przez swój adres. Nie jest wymagane zarządzanie ruchem, ponieważ każdy węzeł jest widoczny poprzez protokół trasowania. Przykładowymi protokołami (algorytmami) dla trasowania płaskiego są Destination-Sequenced Distance Vector (DSDV) i Wireless Routing Protocol (WRP). Protokoły te w swoich tablicach trasowania posiadają wpisy tras do wszystkich węzłów danej sieci. Niestety, tego typu podejście sprawdza się w sieciach Ad Hoc jedynie do chwili, kiedy rozmiary sieci zaczynają się zwiększać, a dynamika zmian topologii wzrasta. Wówczas skalowalność takiego protokołu jest ograniczona. W takim wypadku powinien zostać zastosowany protokół bazujący na hierarchicznym modelu trasowania.

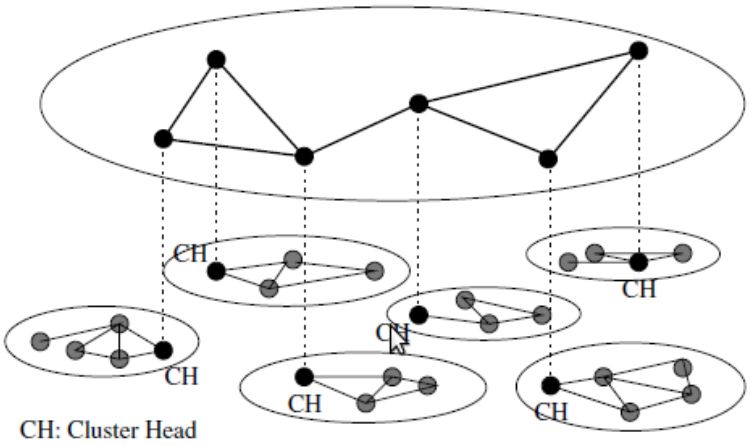
Ideą trasowania hierarchicznego jest łączenie węzłów w grupy zwane klastrami. Jeden z węzłów wyznaczany jest na węzeł główny, w którym przechowywane są informacje dotyczące przynależności węzłów do klastra. Kiedy węzeł chce przesłać pakiet wysyła go do węzła głównego, będącego brzegowym węzłem klastra.

Na rysunku 3 przedstawiona jest sieć o płaskiej architekturze, w której wielokrotne połączenia do poszczególnych obszarów umożliwiają wybór różnych tras dla pakietów. Sytuacja ta zwiększa jednak wykorzystanie zasobów niezbędnych do utrzymywania przez poszczególne węzły pełnej informacji o topologii sieci.



Rys. 3. Płaska infrastruktura trasowania.

W architekturze hierarchicznej, tak jak na rysunku 4, tylko wyznaczone węzły w danym obszarze są odpowiedzialne za ruch międzyobszarowy, co pozwala na uproszczenie procesu trasowania, a także zminimalizowanie stopnia użycia potrzebnych do jego realizacji zasobów.



Rys. 4. Hierarchiczna infrastruktura trasowania.

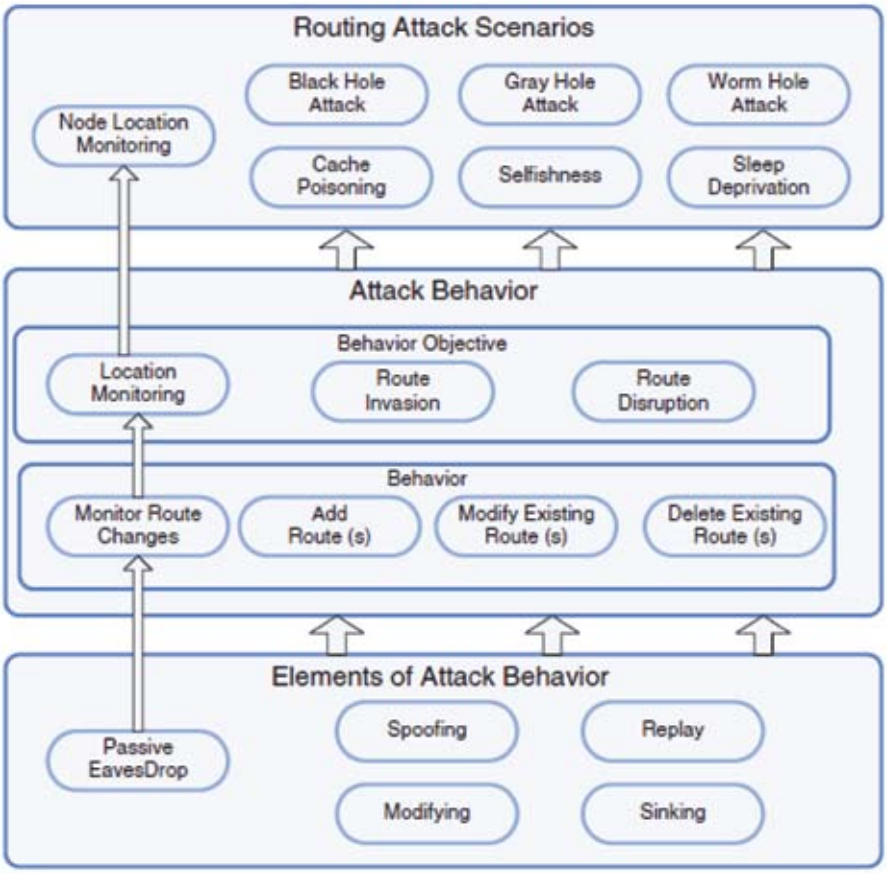
Ważnym aspektem towarzyszącym trasowaniu w sieciach Ad Hoc jest ograniczona energia zasilania węzła. Działania wykonywane przez protokoły trasowania mogą znacznie wykorzystywać dostępne zasoby energetyczne. Dlatego protokoły powinny być tak projektowane, aby uwzględniać to wymaganie i efektywnie wykorzystywać źródło zasilania w trakcie realizacji zadania wyznaczania tras [2]. Oszczędność energii dotyczy nie tylko warstwy związanej z trasowaniem, ale również warstwy łącza danych, jak i warstwy aplikacji. Do technik stosowanych w sieciach Ad Hoc mających za zadanie oszczędzanie dostępnej energii należą między innymi [7]:

- Transmisja multicastowa,
- Korzystanie ze stanu aktywnego (ang. active) urządzenia i stanu gotowości (ang. standby),
- Unikanie retransmisji danych.

Ze względu na swój charakter sieci Ad Hoc są podatne na różnego rodzaju ataki sieciowe. Jako jedna z krytycznych usług w sieci, trasowanie jest częstym celem ataków [2]. Wszystkie węzły w sieci muszą ze sobą współpracować i wymieniać informację w celu wyznaczania i utrzymywania tras. Fakt ten oraz natura sieci sprawiają, że protokoły trasowania są trudnym do ochrony obszarem działania sieci. Podobnie jak w sieciach przewodowych, infrastruktura Ad Hoc podatna jest na typowe ataki związane z podsłuchiwaniem, blokowaniem usług, podmianą pakietów czy innymi działaniami związanymi z nieautoryzowanym badaniem topologii i właściwości urządzeń należących do sieci [8].

Ataki skierowane na protokoły trasowania mogą być zorganizowane w pewną hierarchię zachowań, zależną od złożoności działań oraz wielkości zniszczeń przez nie powodowanych. Rysunek 5 przedstawia 7 głównych scenariuszy ataków na protokoły trasowania w sieciach Ad Hoc. Scenariusze te zawierają zestaw działań oraz zachowań, które wykonane w odpowiedniej kolejności prowadzą do osiągnięcia przez atakującego zamierzonego celu. Poszczególne zachowania zostały sklasyfikowane jako służące:

- Podglądaniu tras,
- Modyfikowaniu tras,
- Uszkadzaniu tras.

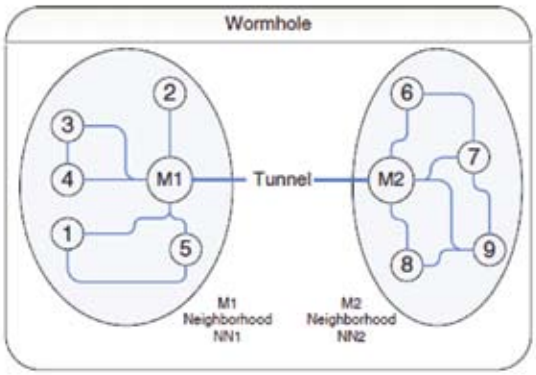


Rys. 5. Hierarchia ataków na protokoły trasowania [2].

Przy aktywnych atakach na mechanizmy trasowania celem może być nie tylko zmodyfikowanie lub zablokowanie procesu trasowania, ale również takie wpłynięcie na jego działanie, aby możliwe stało się przeprowadzenie ataku w wyższych warstwach modelu ISO OSI. Pasywne ataki mogą mieć na celu między innymi podsłuchanie sieci w celu lokalizacji danego węzła, co dla przykładu w rozwiązaniach militarnych może mieć krytyczne znaczenie. Ataki pasywne są znacznie trudniejsze do wykrycia i wyeliminowania.

Poniżej przedstawiono krótkie charakterystyki poszczególnych rodzajów ataków w sieciach Ad Hoc:

- Atak typu „czarna dziura" (ang. Black hole) – w ataku tego typu pakiety są przekierowywane do nieistniejące punku docelowego, gdzie znikają. Można wyróżnić dwie odmiany ataku typu „czarna dziura". W pierwszej odmianie cały ruch jest kierowany do nieistniejącego węzła. W drugim natomiast osoba manipulująca trasami pakietów przekierowuje jedynie określony strumień pakietów do punktu pełniącego role „czarnej dziury",
- Atak typu „tylne wejście" (ang. wormhole) – atak ten polega na stworzeniu tunelu pomiędzy dwoma zaatakowanymi węzłami należącymi do oddzielnych sieci tak, jak to zostało pokazane na rysunku 6. Pomiędzy węzłami M1 i M2 został stworzony tunel będący jedynym łącznikiem komunikacyjnym dla węzłów sąsiadujących z M1 oraz M2.



Rys. 6. Tunel między atakowanymi węzłami przy ataku wormhole [2].

W skutek manipulacji trasami węzły należące do obu sieci posiadają informacje, że droga z jednej sieci do drugiej wiedzie przez jeden przeskok wiodący przez zaatakowane węzły. Dzięki temu w chwili, kiedy jeden z węzłów będzie chciał skomunikować się z węzłem należącym do drugiej sieci, trasa wędrówki pakietów będzie przebiegać przez jeden z zaatakowanych węzłów. Stworzenie takiego środowiska przez atakującego daje mu możliwości przeprowadzenia aktywnego lub pasywnego ataku, polegającego na monitorowaniu i analizie ruchu sieciowego w celu określenia lokalizacji poszczególnych węzłów lub modyfikacji przesyłanych danych. Jest to jeden z trudniejszych do wykrycia atakówa.

- Atak typu „podział sieci" (ang. network partitioning) – atak ma na celu odizolowanie części sieci poprzez usunięcie tras odnoszących się do tego obszaru.
- Atak typu „zatruwanie pamięci podręcznej" (ang. cache poisoning) – atak ten polega na manipulacji informacjami o trasach znajdującymi się w pamięci podręcznej danego węzła.
- Atak typu „ograniczenie uśpienia" (ang. steep deprivation) – atak ten ma na celu wyeliminowanie danego węzła poprzez pozbawienie go źródła zasilania. Sieć Ad Hoc składa się głównie z mobilnych urządzeń wyposażonych w baterie mające ograniczone czasowo możliwości dostarczania energii. Jeden z węzłów sieci, będący pod kontrolą atakującego, przesyła do innego węzła błędne informację związane z trasowaniem. Informację te są cały czas analizowane przez zaatakowany węzeł, co może znacznie skrócić czas jego działania [2].

Jedną z metod mających podnieść bezpieczeństwo w procesie trasowania w sieciach Ad Hoc jest dołączanie do istniejących protokołów trasowania rozszerzenia SRP (Secure Routing Protocol). Opiera się ono na negocjacji tajnego klucza podczas procedury nawiązywania połączenia pomiędzy węzłami. Jednym z protokołów posiadających wbudowane mechanizmy zabezpieczeń oparte na certyfikatach jest protokół ARAN (Authenticated Routing for Ad Hoc Network). Posiada on nie tylko mechanizmy autentykacji, ale również zapewnia integralność danych [1].

3. SYMULACYJNE BADANIE PROTOKOŁÓW ROUTINGU W SIECIACH AD HOC – PRZYKŁADOWY EKSPERYMENT

W celu porównania cech i właściwości protokołów trasowania w sieciach Ad Hoc posłużono się symulacją komputerową. Głównym celem przykładowego eksperymentu symulacyjnego było porównanie dwóch protokołów wykorzystywanych w sieciach Ad Hoc, protokołu DSDV i protokołu AODV. Pierwszy z nich to przedstawiciel grupy protokołów proaktywnych, a drugi to protokół należący do grupy protokołów reaktywnych, wyznaczających trasę na żądanie.

Do wykonania eksperymentu wykorzystano symulator NS2 (Network Simulator 2) w wersji 2.3.4, działający w systemie Ubuntu 10.10. NS2 jest darmowym, zorientowanym obiektowo, symulatorem napisanym w języku C++. Nie posiada interfejsu graficznego. Definiowanie poszczególnych scenariu-

szy odbywa się przy użyciu języka TCL (Tool Command Language). Wyniki symulacji zapisywane są w specjalnym pliku śladu, posiadającym rozszerzenie *.tr. W celu przetworzenia wyników posłużono się dostępnymi w Internecie skryptami języka AWK, służącego głównie do wyszukiwania i przetwarzania wzorców w plikach lub strumieniach danych. Aby zilustrować zebrane dane wykorzystano narzędzie Microsoft Excel.

Przygotowanie środowiska symulacyjnego

W celu przeprowadzenia prostego eksperymentu symulacyjnego po zainstalowaniu programu NS2 należy utworzyć odpowiedni skrypt definiujący model symulowanej sieci. Skrypt jest podstawowym elementem scenariusza symulacyjnego. Może zawierać takie informacje jak opis topologii badanej sieci, parametry określające położenie poszczególnych węzłów, model generatora błędów transmisji, parametry źródeł ruchu sieciowego. Plik posiada rozszerzenie *.tcl. Uruchomienie symulacji odbywa się po wydaniu komendy:

```
$ns nazwa_pliku.tcl
```

W pierwszej części pliku definiowane są parametry wykorzystywane w modelu symulacyjnym. Tabela 2 zawiera informacje o poszczególnych ustawieniach.

Tabela 2. Parametry modelu sieci.

Parametr	Ustawienie	Opis
set val(chan)	Channel/WirelessChannel	Typ kanału
set val(prop)	Propagation/TwoRayGround	Model propagacji
set val(ant)	Antenna/OmniAntenna	Typ anteny
set val(ll)	LL	Typ warstwy łącza
set val(ifq)	Queue/DropTail/PriQueue	Typ kolejki
set val(ifqlen)	50	Maksymalna ilość pakietów w kolejce
set val(netif)	Phy/WirelessPhy	Typ interfejsu
set val(mac)	Mac/802_11	Warstwa MAC
set val(rp)	DSDV (AODV)	Protokół trasowania
set val(nn)	50	Deklaracja ilości węzłów
set val(sc)	nazwa_pliku	Ścieżka do pliku z topologią
set val(cp)	nazwa_pliku	Ścieżka do liku z deklaracją ruchu
set val(x)	500	Rozmiar obszaru działania symulacji
set val(y)	500	

Po definicji parametrów rozpoczyna się zasadnicza część skryptu. W pierwszej części tworzona jest instancja symulatora:

```
set ns_ [new Simulator]
```

W kolejnej części skryptu definiującego model symulacji tworzona jest topografia oraz uchwyt do pliku zawierającego dane z symulacji oraz uchwyt do pliku *.nam, wykorzystywanego do wizualizacji symulacji w programie NAM (Network Animator).

```
set topo [new Topography]
```

```
$topo load_flatgrid $val(x) $val(y)
$ns_ use-newtrace;
set namfd [open w_nam.nam w];
$ns_ namtrace-all-wireless $namfd $val(x) $val(y)
$ns_ use-newtrace
set tracefd     [open wyniki.tr w];
$ns_ trace-all $tracefd
```

W kolejnej linii skryptu tworzony jest obiekt GOD (General Operations Director), który przechowuje globalne informacje na temat stanu środowiska, sieci lub węzłów:

```
set god_ [create-god $val(nn)]
```

Kolejna część skryptu zawiera definicję pojedynczego węzła oraz kod (pętla for) „powoływania do życia" kolejnych węzłów:

```
$ns_ node-config -adhocRouting $val(rp) \
                 -llType $val(ll) \
                 -macType $val(mac) \
                 -ifqType $val(ifq) \
                 -ifqLen $val(ifqlen) \
                 -antType $val(ant) \
                 -propType $val(prop) \
                 -phyType $val(netif) \
                 -topoInstance $topo \
                 -channelType $val(chan) \
                 -agentTrace ON \
                 -routerTrace ON \
                 -macTrace OFF \
                 -movementTrace OFF
      for {set i 0} {$i < $val(nn) } {incr i} {
            set node_($i) [$ns_ node ]
            $node_($i) random-motion 0
         }
```

W następnym fragmencie linii skryptu następuje załadowanie plików z informacjami dotyczącymi ruchu sieciowego, a także zmian topologii sieci:

```
source $val(sc)
source $val(cp)
```

Ostatnia część skryptu to definicja zakończenia symulacji oraz uruchomienie samej symulacji:

```
for {set i 0} {$i < $val(nn) } {incr i} {
              $ns_ at 200.0 „$node_($i) reset";
              }
$ns_ at 199.0001 „stop"
      $ns_ at 199.0002 „puts \"ZAKONCZENIE SYMULACJI..\" ; $ns_ halt"
```

```
        proc stop {} {
                global ns_ tracefd
                close $tracefd
        }
$ns_ run
```

Definiowanie topologii oraz ruchu sieciowego

Istnieje możliwość ręcznego zdefiniowania topologii sieci oraz ruchu sieciowego bezpośrednio w pliku skryptu opisującego model symulacyjny. Jednak dla sieci Ad Hoc, gdzie wymagana jest pewna dynamika sieci, dla dużej liczby węzłów byłoby to zajęcie żmudne i czasochłonne. Dlatego warto posłużyć się gotowymi programami i skryptami, które wygenerują losowe dane. Narzędziem służącym do utworzenia pliku z topologią jest program „setdest". Składnia polecenia jest następująca:

```
setdest -v 1 -n $numnodes -M $speed -M $maxspeed -t $simtime -x $maxx -y $maxy
```

gdzie:
- n : liczba węzłów w topologii
- M : maksymalna prędkość przemieszczania się węzłów
- t : czas trwania symulacji
- x,y : obszar ruchu węzłów [9]

Na potrzeby eksperymentu utworzono cztery pliki zawierające odpowiednio prędkości poruszania się węzłów 1, 10, 50 i 100 m/s. Liczba węzłów została ustawiona na 50. Czas symulacji to 200 sekund. Pozostałe parametry zostały ustawione identycznie, jak w pliku z modelem symulacji.

Do utworzenia pliku zawierającego informację o pojawiającym się ruchu w sieci wykorzystano skrypt „cbrgen.tcl" dołączony, podobnie jak program „setdest", do instalacji symulatora NS2. Składnia polecenia to:

```
ns cbrgen.tcl [-type cbr|tcp] [-nn nodes] [-seed seed] [-mc connections][-rate rate]
```

gdzie:
-type : typ połączenia (wykorzystany w symulacji cbr/udp)
-nn : liczba węzłów
-seed : odstęp pomiędzy losowymi połączeniami
-mc : maksymalna ilość połączeń pomiędzy węzłami
-rate : wartość wykorzystywana do wyliczenia interwału czasu pomiędzy pakietami [9].

Utworzone zostały również cztery pliki opisujące ruch sieciowy, wykorzystywane razem z plikami topologii w różnych scenariuszach eksperymentu.

Analiza i ocena wyników symulacji

Dane wygenerowane podczas przeprowadzania symulacji zapisywane są w pliku śladu. Aby uzyskać pożądane dane wyjściowe konieczne jest dalsze przetwarzanie pliku śladu. Symulator NS2 nie zawiera żadnych narzędzi pozwalających na interpretacje wyników symulacji, dlatego w eksperymencie posłużono się dostępnymi w Internecie skryptami do obróbki danych zawartych w plikach *.tr. Istnieje również możliwość wykorzystania innych narzędzi systemu Linux do obróbki danych tekstowych, jak np. programu „grep". Aby odpowiednio zinterpretować dane zapisane w pliku śladu trzeba poznać jego strukturę. Rysunek 7 przedstawia strukturę danych zapisywanych przez symulator.

Zdarze-nie	Czas	Węzeł źródło-wy	Węzeł docelo-wy	Typ pakietu	Wiel-kość pakietu	Flaga	Identy-fikator strumie-nia	Adres źródło-wy	Adres docelo-wy	Numer sekwen-cyjny	Uni-kalny identy-fikator pakietu

Rys.7 Struktura pliku śladu.

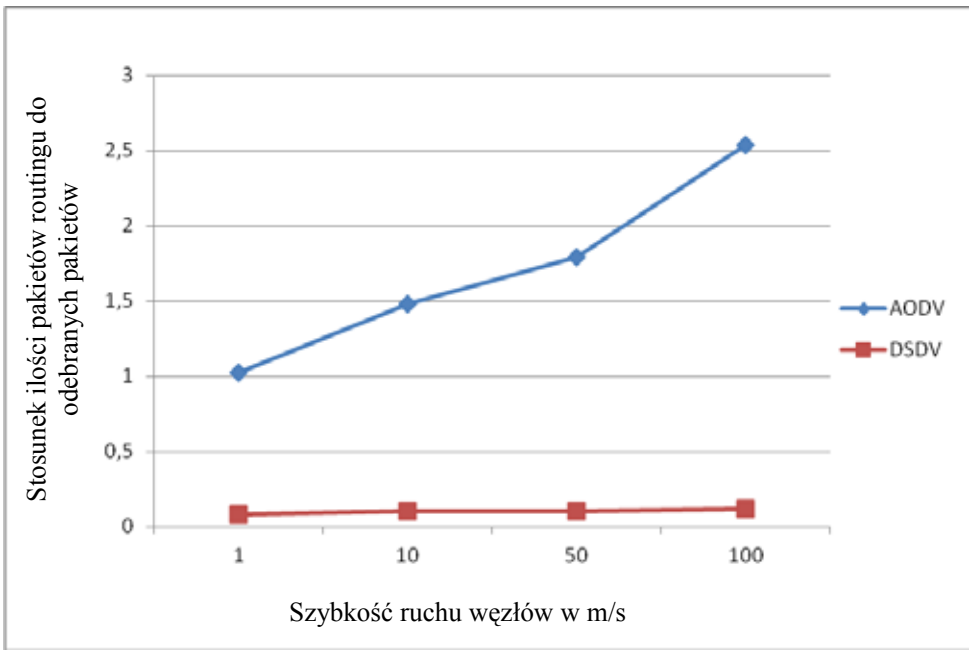
Na podstawie danych zapisanych podczas symulacji można wyznaczyć różne parametry sieci. Kilka z nich to:

- Przepustowość protokołu trasowania,
- Procent utraconych pakietów,
- Opóźnienie „end-to-end”,
- Narzut związany z trasowaniem,
- Procent wykorzystania dostępnej szerokości pasma.

Jako zmienną podczas eksperymentu zastosowano maksymalną prędkość, z jaką węzły mogą zmieniać swoją pozycję. Wartość ta jest ustalana podczas generowania pliku z topologią za pomocą narzędzia „set-dest”. Wartości porównywane to:

– Opuźnienie – średnie opóźnienie podczas dostarczania pakietów,
– Narzut związany z trasowaniem – stosunek liczby pakietów protokołu routingu do liczby wszystkich odebranych pakietów,
– Przepustowość protokołu trasowania – całkowity rozmiar danych odebranych przez węzły docelowe w zadanej jednostce czasu. W eksperymencie wartość ta jest mierzona przez cały czas trwania symulacji.

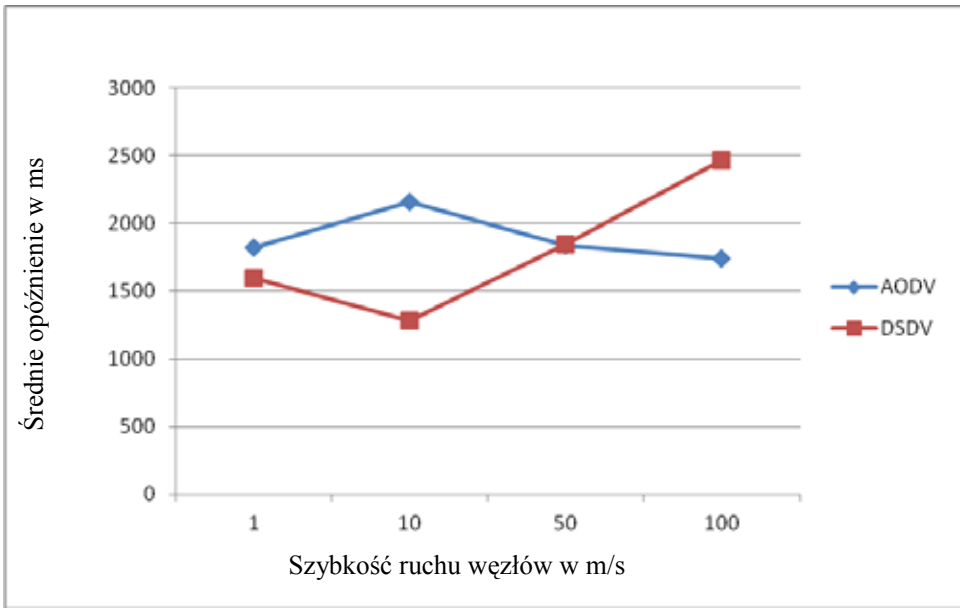
Zgodnie z wykresem na rysunku 8, liczba pakietów wygenerowanych na potrzeby wyznaczenia tras dla pakietów jest w przypadku protokołu AODV dużo większa niż przy protokole DSDV. Podczas procesu odnajdywania trasy do punktu docelowego protokół AODV rozgłasza pakiety RREQ do wszystkich węzłów w sieci. Częstsze zmiany pozycji węzłów zachodzące w badanej topologii sieci zwiększają dodatkowo tę liczbę. Przy zwiększaniu prędkości przemieszczania się węzłów buforowane lokalnie przez węzeł informacje o trasach musza być częściej odświeżane. Rozwiązaniem tego problemu są protokoły hybrydowe, sposób działania których pozwala na ograniczenie ruchu generowanego na potrzeby wyznaczania tras.



Rys. 8. Narzut związany z trasowaniem.

Protokół DSDV korzysta z przechowywanych tablic routingu. Jedynie w momencie znacznej zmiany topologii lub wysyłania aktualizacji okresowych zwiększa się ilość ruchu związanego z routingiem. Trzeba również pamiętać, że protokół DSDV jedynie okresowo przesyła całe swoje tablice do sąsiadów. Przy tzw. aktualizacjach wyzwalanych, mających miejsce w chwili zmiany topologii, przesyłana jest jedynie aktualizacji przyrostowa zawierające informacje o zmienionych trasach. Zasoby potrzebne do przechowywania i przetworzenia informacji zapisanych w lokalnej tablicy każdego węzła wzrastają wraz ze zwiększeniem liczby węzłów w sieci.

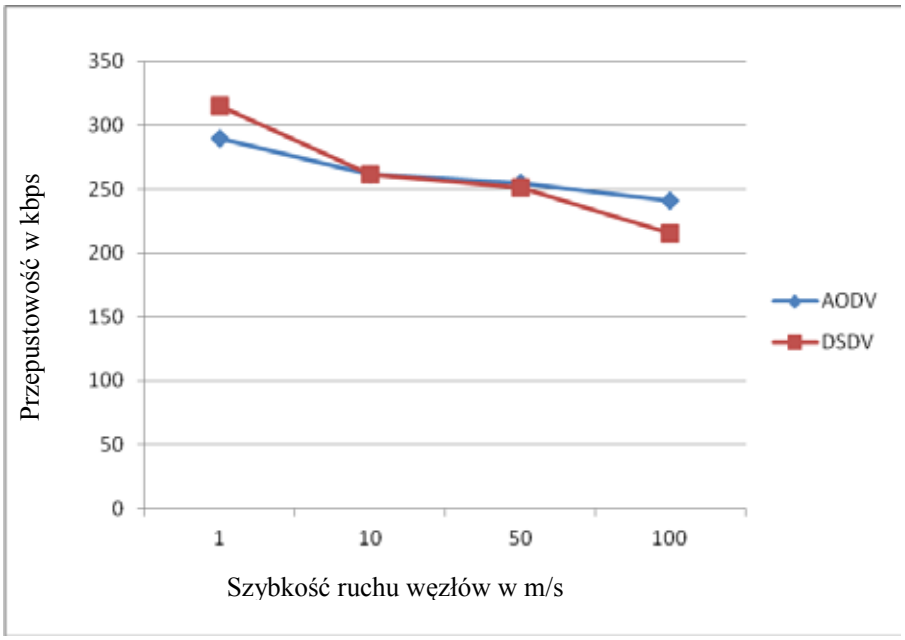
Charakterystyka pracy AODV odzwierciedla się również w kwestii opóźnień w dostarczaniu pakietów do celu. Wykres na rysunku 9 ilustruje trendy opóźnień obu protokołów w odniesieniu do zwiększającej się szybkości poruszania się węzłów.



Rys. 9. Czasy opóźnień w dostarczaniu pakietów dla obu protokołów.

Dzięki przechowywaniu informacji na temat topologii sieci w tablicy routingu, protokół DSDV może przy wolniej zmieniającej się topologii szybciej wskazać drogę pakietom niż protokół AODV. Wynika to z faktu, że protokół AODV musi poświęcić czas na wyznaczenie trasy. Jednak jak widać na wykresie, wraz ze wzrostem dynamiki sieci protokół AODV lepiej adaptuje się do zmieniających się warunków, co skutkuje mniejszymi opóźnieniami niż w przypadku korzystania z protokołu DSDV, który dodatkowo musi przeznaczać czas i zasoby na gromadzenie i odpowiednie przetworzenie informacji o trasach.

Lepsze przystosowanie się do bardziej dynamicznego środowiska można również zauważyć analizując wartość przepustowości danego protokołu. Porównanie tego parametru przestawia wykres na rysunku 10.



Rys. 10. Przepustowość protokołu.

Protokół DSDV przy małej szybkości ruchu węzłów w sieci posiada możliwość przesłania większej ilości danych, gdyż po wstępnym ustaleniu topologii może od razu przesyłać pakiety do celów zapisanych w tablicach trasowania. Protokół AODV wyznacza trasy na żądanie, tak więc nie może od razu wysłać pakietów. Najpierw musi wyznaczyć trasę do punktu docelowego, co zajmuje dodatkowy czas. Jednak zgodnie w wykresem na rysunku 10, w miarę wzrostu prędkości poruszania się węzłów protokół AODV lepiej radzi sobie ze zmieniającymi się warunkami, co skutkuje większą wydajnością podczas przesyłania danych. W przypadku protokołu DSDV, przy częstej zmianie topologii, pojawia się dodatkowy narzut związany z dostosowaniem i aktualizacją tablic routingu.

ZAKOŃCZENIE

Analiza wyników symulacji przeprowadzonych dla dwóch protokołów trasowania uwidacznia ich zalety i wady, a także możliwości przystosowania się do określonych warunków. Protokoły proaktywne, takie jak DSDV mogą sprawdzić się w sieciach o niezbyt dużej liczbie węzłów oraz statycznej topologii. Z racji pochodzenia tych rozwiązań bezpośrednio od protokołów przeznaczonych dla sieci przewodowych, nie do końca są one przystosowane do pracy w sieciach Ad Hoc, w których ciężko jest przewidzieć ilość węzłów czy szybkość ich poruszania się. Protokoły reaktywne wydają się być rozwiązaniem projektowanym z myślą przede wszystkim o sieciach Ad Hoc. Niestety również w przypadku tych protokołów, w określonych sytuacjach efektywność ich pracy spada. W przypadku protokołu DSDV przy znaczącym wzroście ilości węzłów w sieci i zwiększeniu dynamiki sieci można się spodziewać szybkiego wyczerpania dostępnych zasobów sprzętowych, niemogących pomieścić ani przetworzyć w rozsądnym czasie informacji dotyczących tras. Natomiast w przypadku protokołu AODV widzimy, że wzrost szybkości poruszania się węzłów prowadzi do częstych zmian tras dla pakietów, a co za tym idzie do bardziej intensywnego zalewania sieci pakietami biorącymi udział w wyznaczaniu ścieżek dla pakietów. Pewnym rozwiązaniem niedomagań protokołów proaktywnych i reaktywnych są protokoły hybrydowe. Pozwalają one na bardziej efektywne wyznaczanie tras dla pakietów przy użyciu mniejszych zasobów urządzeń.

Zagadnienie trasowania pakietów w sieciach Ad Hoc jest głównym tematem prac badawczych nad sieciami tego typu. Część z wymienionych w artykule protokołów jest już w użyciu, część jest dopiero rozwijana, a niektóre z nich staną się w przyszłości punktem wyjściowym dla projektantów poszukujących bardziej wydajnych rozwiązań.

BIBLIOGRAFIA

[1] S. K. Sarkar, T. G. Basavaraju, C. Puttamadappa, *Ad Hoc Mobile Wireless Networks Principles, Protocols, and Applications*, Taylor & Francis Group, 2007.

[2] S. Misra, S. C. Misra, I. Woungang, *Guide to Wireless Ad Hoc Networks,* Springer-Verlag, London 2009.

[3] P. Mohapatra, S. V. Krishnamurthy, *Ad Hoc networks (Technologies and Protocols)*, Springer, 2005.

[4] A. Boukerche, *Handbook of Algorithms for Wireless Networking and Mobile Computinng*, Chapman & Hall, 2006.

[5] NATO Research and Technology Organisation, *Awareness of Emerging Wireless Technologies: Ad-hoc and Personal Area Networks Standards and Emerging Technologies*, Online, 2007.

[6] S. Basagni, M. Conti, S. Giordano, I. Stojmenovic, *Mobile Ad Hoc Networking*, IEEE Press, New Jersey 2004.

[7] A. Boukerche, *Alhorithms and Protocols for Wireless and Mobile Ad Hoc Networks*, Wiley, Ottawa 2009.

[8] S. Pierre, M. Barbeau, E. Kranakis, Ad Hoc , *Mobile and Wireless Networks*, Springer, Montreal 2003.

[9] M. Greis, *Tutorial for the Network Simulator „NS"*, 2011, http://www.isi.edu/nsnam/ns/tutorial/.