Wydział Fizyki, Astronomii i Informatyki Stosowanej

Uniwersytetu Jagiellońskiego,

Instytut Podstawowych Problemów Techniki

Polskiej Akademii Nauk

mgr Grzegorz Oryńczak

System agentowy dla wspomagania bezpiecznych usług czasu rzeczywistego

Rozprawa doktorska

Promotor

prof. dr hab. inż. Zbigniew Kotulski

Warszawa 2014

Podziękowania

Pragnę bardzo serdecznie podziękować Profesorowi Zbigniewowi Kotulskiemu. Oczywiste jest, iż bez jego dużego wsparcia oraz niezmiernie cennych wskazówek merytorycznych niniejsza praca nie mogłaby powstać. Szczególnie jednak dziękuję za wyrozumiałość, otwartość na nowe pomysły oraz stałą chęć do spotkań i dyskusji. Duża wiedza oraz poczucie humoru Pana Profesora potrafiły przywrócić chęć do dalszej pracy nawet podczas chwil zwątpienia.

Jednocześnie bardzo dziękuję mojej żonie Natalii za jej zainteresowanie prowadzonymi przeze mnie badaniami oraz nieocenioną pomoc w postaci zarówno celnych uwag merytorycznych, jak i uśmiechu oraz stałego wsparcia, które dawały mi dodatkową motywację do pracy.

Streszczenie

Niniejsza rozprawa doktorska poświęcona została zagadnieniu opracowania systemu wspomagającego transmisję czasu rzeczywistego w Internecie. Systemy tego typu, gdzie bardzo istotny jest czas pomiędzy wysłaniem wiadomości, a jej dotarciem do odbiorcy, zyskują coraz większą popularność. Przykładami mogą tu być telefonia internetowa VoIP, usługa strumieniowania obrazu wideo, telekonferencje, ale także zastosowanie militarne jak zdalna kontrola drona czy medyczne (np. wypadki zdalnych operacji chirurgicznych).

Projektując system na potrzeby transmisji danych czasu rzeczywistego w Internecie należy wziąć pod uwagę wiele czynników. Jednym z głównych jest wybór protokołu dla warstwy transmisyjnej. W przeciwieństwie do łącz komutacyjnych stosowanych w tradycyjnej telefonii stacjonarnej PSTN, gdzie dane przychodzą zgodnie z kolejnością wysłania oraz cechują się stałym opóźnieniem, przesyłanie danych w Internecie odbywa się przy użyciu komunikacji pakietowej. Różnica ta jest istotna, gdyż w tym przypadku każdy z pakietów podlega niezależnemu trasowaniu, a co za tym idzie kolejność ich odbioru nie musi być zgodna z kolejnością nadania. Dodatkowo każdy z pakietów może dotrzeć z innym opóźnieniem lub nie dotrzeć wcale. Projektując system do transmisji w Internecie niezbędne jest branie pod uwagę tych własności komunikacji pakietowej.

Jako cel pracy autor postawił sobie szereg wyzwań projektowych, związanych z zapewnieniem wysokiej jakości połączenia, bezpieczeństwem, niezawodnością, wsparciem dla wielu typów strumieni real-time oraz możliwością łatwej rozbudowy systemu w przyszłości. Ze względu na złożony charakter systemu autor wykorzystał przy jego projektowaniu zarówno istniejące już mechanizmy, jak i zaproponował wiele własnych rozwiązań. Architektura systemu oparta została o sieć peer-to-peer, dla której opracowany został mechanizm self-healing, istotnie zwiększający jej niezawodność. Wysoką jakość transmisji osiągnięto poprzez zastosowanie agentów stale kontrolujących jakość połączeń międzywęzłowych oraz adaptacyjnego protokołu routingu kontekstowego wspomaganego przez szereg dodatkowych mechanizmów, a także systemu reputacyjnego. Bezpieczeństwo systemu osiągnięte zostało poprzez zastosowanie zarówno standardowych metod kryptograficznych zwanych hard-security, jak i mechanizmów społecznych opartych o reputacji. Ponadto, w wyniku opracowania algorytmu wykrywającego złośliwe koalicje, zostało dodatkowo zwiększone bezpieczeństwo samego systemu reputacyjnego.

Efektem końcowym pracy jest system spełniający wszystkie z postawionych przez autora założeń, co potwierdzają wyniki przeprowadzonych testów.

Abstract

This dissertation is devoted to the problem of designing an agent-based system for supporting real-time communication in the Internet. This type of systems, where the end-to-end transmission delay is a very important factor, are becoming increasingly popular. There are many kinds of systems that are based on this type of communication. The most known example is Internet Telephony (VoIP). However real-time transmission is also used for video streaming, teleconference and telepresence systems and has its applications in medicine (e.g. in remote surgeries) or for the military purposes (e.g. to wirelessly control and obtain visual feedback from flying drones).

Due to the complex nature of the created system, in its design author has used many existing and well known mechanisms as well as new ones, developed especially to meet the needs of this research. Many factors need to be considered when designing system for real-time data transfer. One of them is the choice of transport layer protocol. As opposed to circuit-switched networks such as public switched telephone network (PSTN), where packet arrive in order and have fixed delay, communication over the Internet is based on packet-switched techniques. In this kind of networks packets can arrive without order, with different delay, or may be completely lost: these factors should be taken into account when developing new system.

As a goal of this study, the author has set a number of challenges in the system design, like providing mechanisms for ensuring the best possible connection quality in the changing network environment, providing high security and reliability level, supporting many different kinds of real-time streams, and creating design which allows to extend easily the system in the future.

The system is based on peer-to-peer architecture with a supernode, and the reliability of this architecture was greatly increased by designing a self-healing mechanism. The high quality of transmission was achieved by the use of agents that constantly observe transmission parameters, and by creating adaptive, context based routing framework, supported by the set of additional mechanisms and feedback for the reputation system. To provide high security level, not only the standard cryptographic methods (knows as hard-security) ware used, by also reputation system was designed, to provide social based, soft-security. Additionally, to ensure higher efficiency and security of the reputation system, a malicious coalitions detecting mechanism was designed.

The final result of this work is a system that is fulfilling all of the objectives set by the author, which was confirmed by the results presented in this dissertation.

Spis treści

Spis oznaczeń

t_s	interwał czasowy z jakim przesyła się oceny przebiegu interakcji
A	Zbiór wszystkich agentów
L	zbiór połączeń (linków)
B	zbiór obiektów ocenianych
π	funkcja ewaluacyjna służy do oceny jakości interakcji
Q	zbiór wszystkich możliwych ocen funkcji π
$T = [0,\ t_{act}]$	relatywny czas liczony od chwili rozpoczęcia pracy systemu
I	zbiór wszystkich zarejestrowanych bezpośrednich doświadczeń
Δt	okres obserwacji
T_D	dyskretny czas dla obserwacji
E	zbiór wszystkich bezpośrednich ocen z uwzględnieniem okresu obserwacji
$\tau: E \cup I \to T_D \cup T$	zwraca czas interakcji (dla czasu ciągłego lub dyskretnego)
$agr: A \times B \times I \times T_D \to Z$	funkcja agregująca obserwacje w danym czasie
E_b^a	podzbiór wszystkich ocen wystawionych przez agenta a na temat obiektu b
E_b	zbiór wszystkich ocen na temat obiektu b
$E_{b,t}$	zbiór wszystkich ocen na temat obiektu b w okresie obserwacji t
$E_b^{t_0}$	wszystkie najaktualniejsze oceny na temat $b \in B$ przesłane od chwili czasowej t_0
$L_{b,t}$	lista posortowanych względem czasu elementów zboru $E_{b,t}$
R_rec	reputacja rekomendacyjna - zdolność danego agenta do poprawnej oceny działania ocenianego obiektu
$rec: A \times B \times T \times E \to R_{rec}$	funkcja oceny reputacji rekomendacyjnej
R_{rec}	zbiór możliwych ocen reputacji rekomendacyjnej
$R_{rec,t}^{a,b}$	ocenę reputacji rekomendacyjnej agenta a względem obiektu b w danym okresie obserwacji t
REP	reputacja
R	zbiór możliwych ocen reputacyjnych
$r: B \times T \times R \times R\_rec \to R$	funkcja oceny reputacji
$\mathbb{E}(ev(q))$	wartość oczekiwana oceny q (dla oceny danej rozkładem beta)
$rec(aut(q), b, T-1)$	reputacja rekomendacyjna autora oceny q (aut(g)) na temat b, w chwili *t-1*
UC	kontekst użytkownika
NC	kontekst sieci
SC	kontekst dostawcy usługi
R	lista protokołów routingu
ERPP	sklasyfikowane Portfolio Protokołów Routingu
$\sigma: R \times CF \to \{4,3,2,1,0\}$	funkcja klasyfikująca
S	zbiór wszystkich obsługiwanych serwisów typu real-time
$PP^S \in [0,1]^3$	polityka priorytetów
rf	funkcja oceniająca

Rozdział I

Wstęp

Nie ulega wątpliwości, iż usługi bazujące na transmisji czasu rzeczywistego są bardzo istotną składową współczesnego sposobu komunikacji. Wraz z gwałtownym wzrostem popularności Internetu oraz związanych z nim usług, możliwa stała się budowa serwisów, które jeszcze do niedawna były wyłącznie domeną dużych firm komunikacyjnych oraz radiowo-telewizyjnych. Obecnie są nie tylko ogólnodostępne, ale także znacznie przewyższają wcześniejsze rozwiązania pod względem różnorodności dodatkowych usług. Jednym z najlepszych przykładów usług czasu rzeczywistego jest telefonia internetowa. Do pierwszych prób stworzenia narzędzia umożliwiającego przesłanie ludzkiej mowy przez sieć komputerową należało opracowanie w 1973 roku przez Dannyego Cohena z Information Science Institute (Uniwersytet Południowej Kalifornii) protokołu Network Voice Protocol (NVP) [1]. Protokół ten z powodzeniem został przetestowany na ówczesnej rozproszonej sieci ARPANET. Od tamtego czasu telefonia bazująca na protokole IP nieustannie zyskuje coraz większą popularność. Dalszy wzrost dostępnych przepustowości oraz jakości połączenia w sieciach Internet pozwolił na rozszerzenie zakresu usług o komunikację audiowizualną (m.in. programy Skype oraz AIM), systemy wideokonferencyjne czy mobilne roboty oferujące możliwość teleobecności. Obecnie komunikacja głosowa z wykorzystaniem Internetu stała się poważnym konkurentem standardowej telefonii stacjonarnej oraz komórkowej, a dzięki możliwości wykonywania tanich połączeń międzynarodowych oraz szerokiemu wachlarzowi dodatkowych usług będzie ona zapewne jednym z głównych sposobów komunikacji w przyszłości.

Systemy transmisji czasu rzeczywistego w Internecie nie ograniczają się jednak wyłącznie do komunikacji dźwiękowej oraz audiowizualnej. Popularność szerokopasmowych łącz pozwala na udostępnianie także usług powiązanych ze strumieniowaniem mediów, takich jak:

- systemy telewizji internetowej, gdzie protokół czasu rzeczywistego zarządza transmisją „na żywo”,
- sieciowe gry komputerowe, gdzie komunikacja w czasie rzeczywistym między graczami jest niejednokrotnie niezbędna dla uzyskania satysfakcjonującej jakości rozgrywki,
- usługi Wideo na Żądanie (VoD), gdzie mimo iż opóźnienie nie jest kluczowym czynnikiem, wciąż trzeba efektywnie zarządzać procesem strumieniowania mediów na bardzo dużą skalę.

Ze względu na spore zainteresowanie podmiotów komercyjnych tego typu usługami, rozwiązania, które pozwalają dostarczyć wspomniane funkcjonalności, zostały już zaprojektowane i są wciąż udoskonalane. Jednakże w dziedzinie tej jest jeszcze wiele istotnych wyzwań, które nie zostały kompleksowo rozwiązane. Jednym z najistotniejszych jest brak jednolitego mechanizmu mogącego zapewnić pożądaną wysoką jakość usługi (QoS). Także mechanizmy pozwalające na zapewnienie wysokiego poziomu bezpieczeństwa transmisji oraz wydajnego zarządzania dostępną przepustowością infrastruktury transmisyjnej muszą zostać przedefiniowane.

Projektując system na potrzeby transmisji danych czasu rzeczywistego w Internecie należy wziąć pod uwagę wiele czynników. Jednym z głównych jest wybór protokołu dla warstwy transmisyjnej. W przeciwieństwie do łącz komutacyjnych stosowanych w tradycyjnej telefonii stacjonarnej PSTN, gdzie dane przychodzą zgodnie z kolejnością wysłania oraz cechują się stałym opóźnieniem, przesyłanie danych w Internecie odbywa się przy użyciu komunikacji pakietowej. Różnica ta jest istotna, gdyż w wypadku komunikacji pakietowej każdy z pakietów podlega niezależnemu trasowaniu, a co za tym idzie kolejność ich odbioru nie musi być zgodna z kolejnością nadania, a dodatkowo każdy z pakietów może dotrzeć z innym opóźnieniem lub nie dotrzeć wcale. Projektując system do transmisji w Internecie niezbędne jest branie pod uwagę tych własności komunikacji pakietowej. Najczęściej używanymi protokołami transmisyjnymi w Internecie są TCP oraz UDP. Nie były one jednak projektowane na potrzeby transmisji real-time i nie dają kontroli nad stałym oraz zmiennym (tzw. jitter) opóźnieniem, a przez to w swojej niezmienionej formie nie mogą zostać użyte do tego specyficznego zastosowania. TCP jest protokołem niezawodnym, zapewniającym dotarcie pakietów w odpowiedniej kolejności, jednak jest on również protokołem bardziej złożonym i wolniejszym od UDP. Wbudowany w TCP mechanizm retransmisji utraconych pakietów jest zbyt wolny dla potrzeb transmisji real-time:

retransmitowany pakiet po dotarciu do miejsca docelowego może okazać się nieaktualny. Ponadto niezawodność transmisji nie jest zazwyczaj najistotniejszym czynnikiem w transmisji real-time. Utrata kilku klatek w strumieniu video lub milisekund strumienia audio nie jest tak uciążliwa, jak opóźnienie w komunikacji. Z tego też powodu UDP jest preferowanym wyborem jako podstawa do dalszego budowania własnego protokołu komunikacyjnego na potrzeby usług real-time. Nie bez znaczenia jest także długość nagłówka pakietu, który w wypadku UDP wynosi 8, a w TCP 20 bajtów. Jednak pomimo zysku płynącego z szybkości protokołu UDP, projektując własne, oparte na nim protokoły komunikacyjne, należy pamiętać o braku wsparcia dla wielu istotnych mechanizmów. Jednym z nich jest zarządzanie przeciążeniem, którego brak może doprowadzić do przeciążenia sieci oraz utraty transmitowanych danych. Kolejnym mechanizmem, który należy dodać, jest znakowanie czasowe pakietów, tak aby możliwe było ich odtworzenie we właściwej kolejności. Obecnie stosowane protokoły opracowane na potrzeby transmisji real-time, takie jak Real Time Protocol (RTP) obsługują te zagadnienia, lecz wciąż nie gwarantują wymaganej jakości usługi (QoS).

Kolejnym istotnym problemem związanym z transmisją czasu rzeczywistego jest sam proces routingu pakietów. Powszechnie stosowane algorytmy routingu nie są optymalizowane pod kątem transmisji real-time, a więc najbardziej istotne z punktu widzenia takiej transmisji aspekty jak stałe lub zmienne opóźnienie oraz liczba gubionych pakietów często nie są głównym czynnikiem w procesie trasowania. Próbami rozwiązania tego problemu są mechanizmy IntServ [2] oraz Diffserv [3] odpowiedzialne za rezerwację zdolności przesyłowej na całej trasie transmisji oraz różnicowanie i priorytetyzację strumieni. Mechanizmy te posiadają jednak problemy ze skalowalnością lub integralnością działania, a przez to często nie spełniają swojego zadania.

Bardzo ważnym aspektem jest także zapewnienie bezpieczeństwa transmisji oraz bezpieczeństwo samej infrastruktury. Obecnie stosowane lub proponowane rozwiązania są dość ograniczone w tej dziedzinie. O ile posiadają one możliwość szyfrowania komunikacji przy użyciu standardowych metod kryptograficznych, to problem bezpieczeństwa infrastruktury, szczególnie od strony zagrożenia z wewnątrz (wadliwy/złośliwy element infrastruktury), jest pomijany.

Ostatecznie, pomimo iż istnieje wielkie zapotrzebowanie na serwisy obsługujące transmisję danych czasu rzeczywistego w Internecie, brak jest spójnego rozwiązania mogącego

obsługiwać wiele tego typu usług jednocześnie. Różnorodność typów strumieni real-time, pociąga za sobą zazwyczaj różnice w systemach obsługujących konkretny rodzaj tego typu transmisji. Co za tym idzie, każdy system, często włącznie z fizyczną infrastrukturą, projektowany jest pod specyficzny typ transmisji i nie może być używany zamiennie z innym. Przykładowo, obecnych systemów opracowanych na potrzeby telefonii VoIP nie da się w prosty sposób rozbudować tak, aby obsługiwały dodatkowo strumieniowanie np. gier.

System dla wspomagania bezpiecznej transmisji czasu rzeczywistego w Internecie opracowany w ramach niniejszej pracy ma na celu sprostanie wszystkim wymienionym powyżej wyzwaniom. Oparcie architektury systemu o sieć peer-to-peer (P2P) daje większą kontrolę nad trasami przesyłania pakietów. Hybrydowa architektura oraz opracowany mechanizm samo uzdrawiania sprawia, iż system jest odporny na awarie oraz próby ataków. Ponadto, dzięki zaprojektowanemu w ramach rozprawy Frameworkowi służącemu do automatycznego wyboru protokołu routingu, system pozwala na jednoczesną obsługę wielu typów strumieni real-time. Duży nacisk został także położony na bezpieczeństwo systemu. W szczególności, poprzez wbudowanie w system mechanizmu reputacyjnego, możliwe stało się wykrywanie wadliwych oraz złośliwych elementów infrastruktury i ich odseparowywanie. Dokładny opis celów oraz zrealizowanych zadań związanych z budową systemu został przedstawiony w kolejnym rozdziale.

Rozdział II

Cel badań oraz teza rozprawy

2.1 Cel badań

Celem jaki postawił sobie autor niniejszej rozprawy było opracowanie systemu służącego do wspomagania transmisji czasu rzeczywistego w sieci Internet, który mógłby sprostać rosnącym wymogom stawianym przez coraz większą popularność oraz wzrost znaczenia tego typu transmisji. Głównym założeniem, jakie przyświecało autorowi pracy, była eliminacja wad obecnie stosowanych rozwiązań, opracowanie spójnego systemu mogącego obsługiwać jednocześnie wiele typów transmisji czasu rzeczywistego oraz zwrócenie szczególnej uwagi na bezpieczeństwo takiego systemu.

Na szczegółowe cele pracy składają się:

- określenie wpływu specyfiki transmisji pakietowej stosowanej w Internecie na jakość transmisji czasu rzeczywistego,
- zaprojektowanie schematu budowy systemu dla wspomagania transmisji czasu rzeczywistego wraz ze specyfikacją wszelkich wymaganych elementów,
- przeanalizowanie dostępnych typów infrastruktur P2P, a następnie opracowanie własnej, hybrydowej, łączącej najistotniejsze cechy infrastruktury rozproszonej (odporność na awarie) oraz scentralizowanej (szybkość wyszukiwania),
- przeanalizowanie dostępnych rozwiązań oraz algorytmów routingu ukierunkowanych na zapewnienie wymaganej jakości połączenia (QoS), a następnie opracowanie własnego Frameworku na potrzeby routingu w zaprojektowanym systemie,
- zaprojektowanie mechanizmu reputacyjnego w celu zwiększenia bezpieczeństwa systemu,

- przeprowadzenie szeregu testów obrazujących działanie systemu w zależności od różnych trybów pracy, specyfiki dostępnej infrastruktury, odporności na ataki zewnętrzne i wewnętrzne.

2.2 Teza rozprawy doktorskiej

Realizacja wymienionych celów pozwoliła na wysunięcie następującej tezy niniejszej rozprawy:

Możliwe jest opracowanie prostego w budowie systemu zdolnego do jednoczesnej obsługi wielu typów transmisji czasu rzeczywistego. System taki wykorzystywałby hybrydową infrastrukturę P2P cechującą się szybkim czasem wyszukiwania ścieżek, co jest charakterystyczne dla infrastruktury scentralizowanej, a jednocześnie odpornością na problem SPOF (Single Point of Failure), cechującą zazwyczaj infrastruktury całkowicie rozproszone. Wbudowanie mechanizmu reputacyjnego jest w stanie istotnie zwiększyć bezpieczeństwo systemu.

2.3 Zrealizowane zadania

Aby wykazać prawdziwość powyższej tezy, zrealizowano powyższe zadania:

- zbadano specyfikę transmisji real-time oraz transmisji pakietowych w Internecie,
- przeanalizowano różne typy transmisji czasu rzeczywistego oraz związane z nimi wymagania,
- opracowano mechanizm self-healing pozwalający w wyniku rozproszonego głosowania wyznaczyć nowy serwer centralny (w wypadku awarii/kompromitacji aktualnego), a następnie, używając mechanizmu dzielenia sekretu, w bezpieczny sposób odbudować bazę danych,
- dostosowano system reputacyjny oparty na rozkładzie beta do potrzeb projektu,
- zaprojektowano autorski algorytm wykrywania złośliwych koalicji w systemach reputacyjnych,
- przeanalizowano dostępne algorytmy routingu oraz mechanizmy wspomagające mogące polepszyć jakość transmisji real-time, a następnie opracowano własny mechanizm routingu bazujący na automatycznym wyborze najlepszego dostępnego protokołu routingu w zależności od danych kontekstowych (typu transmisji),
- zaimplementowano symulator służący do przetestowania prawidłowości opracowanych rozwiązań.

Należy zwrócić uwagę, iż niektóre z opracowanych w ramach pracy rozwiązań – jak mechanizm self-healing dla infrastruktur P2P czy algorytm wykrywania złośliwych koalicji – cechują się znacznie szerszym obszarem zastosowań i mogą być z powodzeniem stosowane w wielu innych projektach, niekoniecznie związanych z transmisją czasu rzeczywistego.

Rozdział III

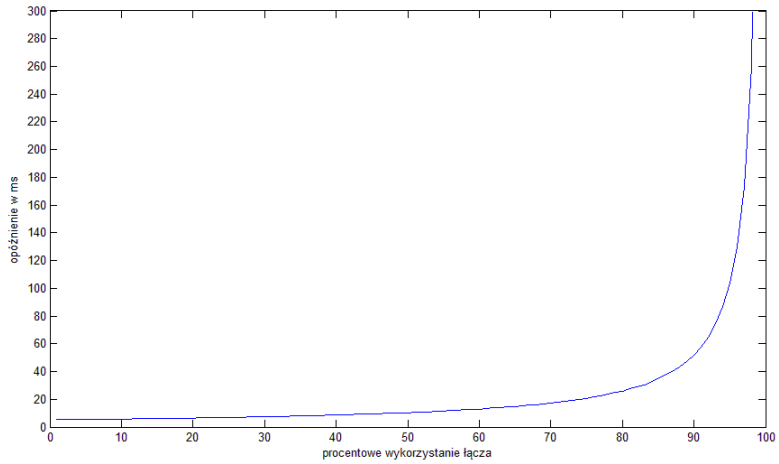
Transmisja czasu rzeczywistego

3.1 Transmisja czasu rzeczywistego w Internecie

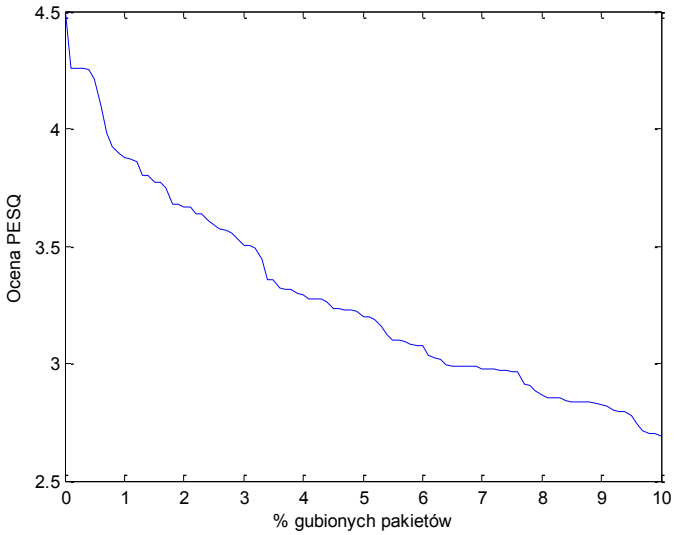
W odróżnieniu od łącz komutacyjnych, charakteryzujących się stałym opóźnieniem oraz zachowaniem poprawnej kolejności przy odbiorze nadesłanych danych, specyfika pakietowej transmisji danych w Internecie narzuca szereg problemów, z którymi borykają się systemy transmitujące dane czasu rzeczywistego w Internecie. Jak już wspomniano we wstępie, do problemów tych należy zarówno opóźnienie transmisji (stałe oraz zmienne) jak i zjawisko gubienia pakietów. Bardzo istotny wpływ na opóźnienie transmisji ma stopień wykorzystania łącza (rysunek 3.1). Przyjmując kilka założeń co do typu transmisji (np. czas dotarcia pakietów opisywany procesem Poissona, rozmiar pakietów zgodny z rozkładem wykładniczym, kolejkowanie M/M/1) można w prosty sposób wyliczyć średnie opóźnienie związane z procesem kolejkowania [4], korzystając ze wzoru:

$$d_q = \frac{1}{\frac{c}{ps} - aar}, \tag{3.1}$$

gdzie *aar* jest średnią prędkością nadsyłania pakietów (pps), *C* przepustowością łącza, a *ps* średnim rozmiarem pakietu. Fakt gubienia pakietów jest natomiast powodowany głównie przeciążeniami, anomaliami routingu czy też fizycznymi błędami transmisji na łączach i także bardzo negatywnie wpływa na jakość komunikacji. Zjawisko to zostało zilustrowane na rysunku 3.2, przedstawiającym zależność między procentowym gubieniem pakietów a spadkiem jakości usługi VoIP.



Rys 3.1 Wpływ stopnia wykorzystania łącza na opóźnienie transmisji.



Rys 3.2 Zależność między procentowym gubieniem pakietów a spadkiem jakości usługi VoIP.

Należy jednak zauważyć, iż zjawiska przeciążenia czy wzmożonego gubienia pakietów nie są częste, a w związku z tym średnie straty pakietów podczas komunikacji także nie są duże. Przeprowadzone w 1999 roku pomiary szacowały je na 0.6-5.2% [5]. Badania z 2003 roku wykazały średnie straty na poziomie 0.44% [6], a przeprowadzone w 2005 analizy oszacowały średnie straty na poziomie 0.26% [7]. Zdarzały się jednak momenty, trwające od kilku milisekund do kilku minut, gdy straty znajdowały się w zakresie od 10 do 100%. Powyższe statystyki potwierdzają również wyniki z 2011 roku [8], kiedy to pomiary wykazały straty na poziomie 0.27-1.11%. Wysokie odchylenie standardowe pomiarów (z zakresu 3.14-8.35)

pozwala jednak wnioskować o istnieniu stanów wzmożonych strat. Okresy te są szczególnie niekorzystne z punktu widzenia transmisji czasu rzeczywistego, gdyż podczas nich może zachodzić zauważalna degradacja jakości przekazu. Istotnym problemem są także awarie linii na trasie routingu, które w połączeniu z wolną zbieżnością protokołów BGP, mogą doprowadzić do zerwania komunikacji lub przekierowania jej na trasę charakteryzującą się znacznie wyższym opóźnieniem.

Wymienione powyżej problemy sprawiają, iż zapewnienie transmisji o wysokiej jakości w Internecie stanowi trudne zagadnienie i stawia przed projektantami systemów obsługujących tego typu transmisje liczne wyzwania do pokonania.

Kolejnym aspektem jest ogromna różnorodność systemów internetowych opierających swoje działanie na transmisji czasu rzeczywistego. Służą one wielu celom, poczynając od typowo rozrywkowych, jak gry sieciowe, a kończąc na zastosowaniach militarnych (np. sterowanie dronami) czy medycznych (przeprowadzanie zdalnych operacji chirurgicznych). Każde z tych zastosowań, pomimo iż bazuje na komunikacji należącej do wspólnej rodziny transmisji real-time, może istotnie różnić się wymogami zarówno co do samych parametrów transmisji jak i jej niezawodności. W kolejnej części rozprawy zostanie zaprezentowany przegląd najpowszechniejszych serwisów tego typu i rozwiązań w nich stosowanych.

3.2 Przegląd systemów transmisji czasu rzeczywistego

Niezależnie od spełnianej funkcji, każda aplikacja, która musi transmitować dane w czasie rzeczywistym, musi także spełniać określone wymagania nałożone przez niższe warstwy. Najważniejsze z nich są związane z opóźnieniem, opóźnieniem zmiennym (jitterem) oraz utratą pakietów. W zależności od typu serwisu, wymagania te mogą dotyczyć również sposobu zarządzania obciążeniem czy przepustowością oraz zastosowania określonych metod transmisji danych. W poniższej części zostanie zaprezentowany przegląd najpowszechniejszych serwisów tego typu.

3.2.1 Telefonia internetowa

Podczas korzystania z telefonii internetowej rozmówcy zazwyczaj są w stanie zauważyć całkowite opóźnienia głosu rzędu 250 ms lub dłuższe. Jedynie nieliczni mogą zauważyć około 200 ms opóźnienia. Jeżeli ten próg zostaje przekroczony, komunikacja staje się uciążliwa. Rekomendacja ITU-T G.114 [9] zaleca, aby czas opóźnienia nie przekraczał 150 ms, a ponieważ czas ten uwzględnia całkowitą trasę, jaką przemierza dźwięk (od mikrofonu do głośnika), opóźnienia transmisji sieci powinny być znacznie krótsze niż 150 ms. Opóźnienia są również przyczyną niepożądanego efektu „echo". W telefonii PSTN echo jest niezauważalne z powodu małych opóźnień transmisji, jednak w przypadku VoIP zjawisko to może być znacznie bardziej uciążliwe. Rozwiązaniem pozwalającym usunąć ten problem może być zastosowanie algorytmu eliminacji echa [10].

Kolejną różnicą pomiędzy standardową a internetową telefonią jest usługa lokalizacji użytkownika. W przypadku stałych linii telefonicznych występuje bezpośrednia relacja pomiędzy numerem telefonu użytkownika a jego fizyczną lokalizacją. Natomiast w przypadku VoIP użytkownik może zalogować się do systemu i skorzystać z klienta telefonicznego z dowolnego miejsca, w którym dostępne jest połączenie internetowe. Co więcej, możliwe jest również zarejestrowanie kilku możliwych lokalizacji, np. w domu, w pracy, itp. – wtedy sygnał przychodzącego połączenia będzie słyszany w każdym z tych miejsc jednocześnie lub w określonej lokalizacji. Pomimo zalet tego podejścia posiada ono wady w przypadku wykonywania połączeń alarmowych, gdyż trudno jest podać geograficzną lokalizację użytkownika VoIP, a w związku z tym połączenia alarmowe nie mogą być łatwo skierowane do najbliższego centrum pomocy. Użytkownicy internetowej telefonii są identyfikowani poprzez ich loginy. Na przykład w SIP używany jest Uniform Resource Identifier (URI) w formie sip:usename@host:port. Ta forma identyfikacji wymaga implementacji w VoIP mechanizmów uwierzytelniania użytkowników oraz ich rejestracji i lokalizacji. Ważnym jest również zapewnienie odpowiedniego mechanizmu bezpieczeństwa dla tych operacji. Obecnie najbardziej popularne aplikacje VoIP są oparte o SIP lub H.323, gdzie H.323 jest rekomendacją z ITU Telecommunication Standardization Sector (ITU-T), która opisuje użycie kilku protokołów dla komunikacji audio/wideo. SIP jest natomiast protokołem sygnalizacyjnym zdefiniowanym przez

Internet Engineering Task Force (IETF). Większość aplikacji telefonii internetowych używa do przekazywania danych w czasie rzeczywistym RTP w połączeniu z protokołem RTCP.

3.2.2 Wideokonferencje

Pierwszy system komercyjny służący komunikacji audiowizualnej został zastosowany w latach 70-tych przez firmę AT&T pod nazwą handlową „Picturephone products". Jednak popularność tego typu systemów umożliwiających pełną komunikację audio i wideo w czasie rzeczywistym zwiększyła się w pierwszej dekadzie XXI wieku wraz z powstaniem darmowych internetowych serwisów telefonii i wideo rozmów, takich jak Skype czy iChat. Obecnie systemy komunikacyjne wideo, będące w stanie zapewnić w czasie rzeczywistym wysokiej jakości obraz (30 fps i rozdzielczość 1280 na 720 pikseli), stają się standardem. Tradycyjne systemy wideokonferencji różnią się od wideo-telefonii ze względu na swoje przeznaczenie: powstały w celu obsługi grup a nie indywidualnych użytkowników. Jednak wraz z udoskonalaniem technologii i oprogramowania ta granica staje się coraz bardziej płynna. Pomimo wzrastającej popularności i ciągłego rozwoju nowych funkcji, wciąż pozostaje wiele do zrobienia w celu stworzenia w pełni niezawodnych systemów komunikacji audiowizualnej, charakteryzujących się wysoką jakością, a jednocześnie dobrą skalowalnością oraz oferujących wysoki poziom bezpieczeństwa.

Wymaganie względem dopuszczalnego opóźnienia transmisji dla tego typu serwisów jest podobne do opisanego wcześniej w przypadku telefonii internetowej: powinno ono być krótsze niż 150 ms. Kolejnym kluczowym warunkiem wymaganym w celu zapewnienia satysfakcjonującej jakości połączenia jest wysoka przepustowość łącza. Różnice w stosunku do VoIP mogą wystąpić również podczas usług rejestracji i lokalizacji. Ponieważ systemy wideokonferencji często wymagają kosztownego i pokaźnych rozmiarów sprzętu używanego w audytoriach, są nieprzenośne i z tego powodu mogą być na stałe powiązane z konkretnym hostem lub domeną. W tym przypadku dodatkowa rejestracja i lokalizacja serwisu nie są konieczne. Z drugiej strony potrzebny jest system kontrolny do alokowania zasobów, przeprowadzania routingu danych oraz dodawania i usuwania uczestników konferencji. Warstwa sygnalizująca jest również konieczna do kontroli połączeń i parametrów sesji. W większości przypadków sygnalizowanie jest przeprowadzane z użyciem protokołów H.323 lub SIP. Do

przekazywania danych audio i wideo w czasie rzeczywistym używany jest zazwyczaj protokół RTP/RTCP.

3.2.3 Usługi strumieniowania mediów

Już od lat 90-tych komputery osobiste stały się na tyle wydajne, aby umożliwić różne typy przekazu, takie jak audio czy wideo. Na początku zawartość tych przekazów była dostarczana metodami niestrumieniowymi (np. przez pobieranie). Jednak z początkiem pierwszej dekady XXI wieku, dzięki silniejszym procesorom i zwiększonej przepustowości sieci, serwisy mediów strumieniowych zaczęły stawać się coraz bardziej popularne. Zwłaszcza serwisy strumieniujące media „na żywo", takie jak internetowe radiostacje i telewizja zyskały sporą uwagę w ostatnich czasach. W obszarze tym został już osiągnięty znaczący postęp, jednak wciąż pozostaje wiele wyzwań, zwłaszcza jeżeli chodzi o wydajną dystrybucję strumienia w czasie rzeczywistym, wyposażoną dodatkowo w mechanizm zapewniający odpowiednie QoS, prywatność użytkowników i pozwalający zabezpieczać przesyłaną zawartość.

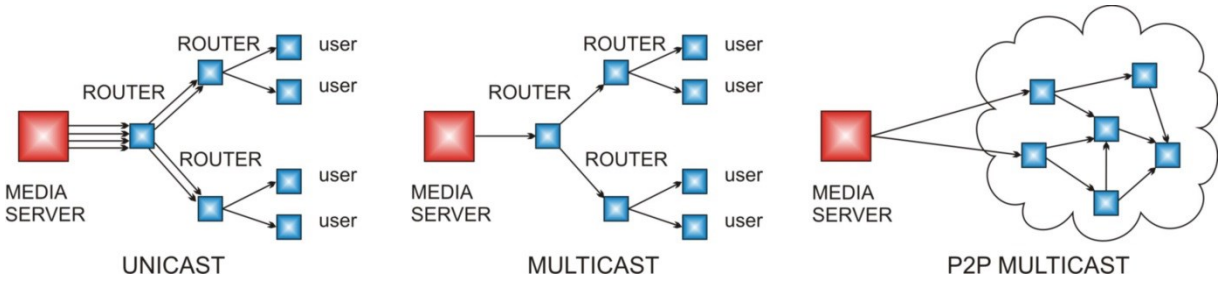
W uproszczeniu strumienie mediów można podzielić na dwie główne kategorie:

- strumienie „na żywo", np. internetowa telewizja i radio,
- strumienie nie czasu rzeczywistego, np. serwisy typu Wideo na Życzenie (VoD).

Wymaganie względem dopuszczalnego całkowitego opóźnienia dla serwisów wykorzystujących strumieniowanie „na żywo" jest również dość rygorystyczne, ale mniej restrykcyjne niż w poprzednio opisanych serwisach. W praktyce nawet połączenia charakteryzujące się względnie wysokimi opóźnieniami mogą być z powodzeniem używane przez większość serwisów. W szczególności możliwe jest zaprojektowanie usługi strumieniowania „na żywo", używając połączeń satelitarnych (opóźnienia rzędu 250 ms) [11]. Z drugiej strony, ponieważ serwisy te są bardzo wymagające pod względem dostępnej przepustowości, wydajne przesyłanie strumienia danych jest ważnym zagadnieniem. Systemy używające połączeń unicast w przypadku większej liczby użytkowników mogą charakteryzować się bardzo wysokim zużyciem zasobów związanych z dostępną przepustowością. Unicast jest najczęściej używanym typem transmisji w Internecie, ale nie skaluje się dobrze, gdy wielu

użytkowników chce równocześnie oglądać tę samą treść – serwer przesyła oddzielną kopię strumienia mediów do każdego odbiorcy, co może prowadzić do przeciążenia sieci. W przypadku setek równoczesnych użytkowników technologia IP multicasting [12] jest zdecydowanie lepszym wyborem, gdyż pozwala na wydajne przesyłanie datagramów IP do grupy zainteresowanych odbiorców. Ma ona jednak również pewne wady: do poprawnej pracy serwisy multicast wymagają zainstalowania na wszystkich poziomach sieci routerów zdolnych do obsługi tego typu połączeń. Z tego powodu rozwiązanie to jest niezbyt często stosowane w praktyce. Inne podejście pozwalające na zbudowanie efektywnych systemów live streaming jest oparcie ich o sieć P2P.

W przypadku multicast na poziomie sieci P2P [13], odbiorcy strumienia na żywo odgrywają jednocześnie rolę routerów dla innych użytkowników. Udowodniono, że podejście to jest użyteczne, skalowalne i wydajne [14] [15] oraz może być zastosowane w wielu rozpowszechnionych aplikacjach, np. Adobe Flash 10.1. Z drugiej strony, serwisy strumieniujące dane nie czasu rzeczywistego, takie jak Wideo na Życzenie, w większości przypadków wciąż są oparte o połączenia unicast. Indywidualne połączenie point-to-point jest ustanawiane pomiędzy serwerem strumieniującym i każdym użytkownikiem. W systemach tych opóźnienie łącza nie stanowi dużego problemu ze względu na użycie lokalnych buforów do przechwytywania porcji danych [16]. Użycie dzielonego bufora jako dodatku do macierzy dyskowych serwera mediów również poprawia wydajność systemu [17]. W większości przypadków zawartość multimedialna jest przekazywana z użyciem protokołów UDP lub RTP. Do kontroli serwera strumieniującego można zastosować Real Time Streaming Protocol (RTSP). Zapewnia on komendy typu VCR, takie jak “odtwórz” i “pauza”. Aby uniknąć nielegalnego kopiowania treści multimedialnych zawartość VoD powinna być szyfrowana – w tym celu może być użyty dodatkowy serwer Digital Rights Management.



Rys. 3.3 Porównanie metod transmisji strumieniowej.

3.3.4 Gry sieciowe

Jeżeli chodzi o gry sieciowe z dużą liczbą uczestników, to wymogi względem jakości i szybkości połączenia zależą mocno od dynamiki rozgrywki. Dokonując pewnego uproszczenia, można rozważać następujące trzy typy gier sieciowych:

- gry strategiczne czasu rzeczywistego (RTS),
- gry typu Massively Multiplayer Online Role-Playing Games (MMORPG),
- gry First Person Shouters (FPS),
- gry nie będące grami czasu rzeczywistego, jak np. szachy.

Zazwyczaj gry RTS lub MMORPG są mniej czułe na opóźnienia spowodowane jakością łącza. Testy pokazują, że w ich przypadku dopiero opóźnienia rzędu 500 ms są zauważalne [18]. Z drugiej strony dynamiczna specyfika gier typu FPS powoduje, że nawet względnie niewielkie opóźnienia mogą mieć duży wpływ na przebieg rozgrywki. Często opóźnienie większe niż 200 ms może być zauważalne i niekiedy nawet uniemożliwiać grę. Dlatego też niektóre serwery gier FPS nie kwalifikują użytkowników charakteryzujących się opóźnieniem rzędu 250 ms jako potencjalnych graczy.

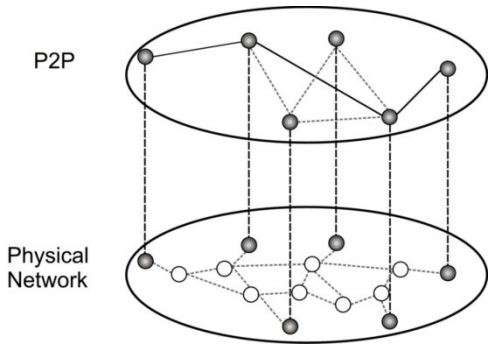
Problem całkowitego opóźnienia jest jeszcze bardziej istotny w coraz bardziej popularnych grach bazujących na chmurze, takich jak OnLive. W tego typu serwisach gry są przechowywane i uruchamiane na zdalnych serwerach, a następnie wygenerowany przez grę obraz jest przesyłany do użytkownika przez Internet. Dzięki temu rozwiązaniu nawet użytkownicy wyposażeni w mniej wydajne komputery mogą uczestniczyć w grach, które mają wysokie wymagania sprzętowe. Ze względu na to, że do chmury przesyłane są, oprócz strumieniowania obrazu video z gry, również polecenia z kontrolera, wymagane jest połączenie charakteryzujące się niewielkim opóźnieniem, aby umożliwić zadowalającą jakość gry.

Rozdział IV

Architektura Systemu

4.1 Budowa systemu

Aplikacja zaprezentowana w niniejszej pracy została oparta o sieć peer-to-peer (P2P). W przeciwieństwie do tradycyjnej architektury klient/serwer, węzły w systemie P2P pełnią jednocześnie zarówno funkcję klienta jak i serwera, a także mogą służyć jako pośrednicy w komunikacji pomiędzy innymi węzłami sieci. Przy pomocy sieci P2P możliwe jest zaimplementowanie abstrakcyjnej sieci nakładkowej, zbudowanej na poziomie warstwy aplikacji, z węzłami sieci utworzonymi z hostów infrastruktury P2P. Tak opracowana abstrakcyjna sieć pozwala na budowę systemów cechujących się niezależnością od fizycznej topologii sieci, gdyż usługa transmisji danych zapewniona jest przez warstwę transportową wchodzącą w skład bazowej, fizycznej infrastruktury.



Rys. 4.1 Model sieci nakładkowej.

Wybór sieci P2P jako infrastruktury używanej w opracowanym w ramach pracy systemie ma swoje uzasadnienie składające się z kilku ważnych powodów.

Po pierwsze, wewnątrz utworzonej sieci nakładkowej możliwe jest podejmowanie własnych decyzji w kwestii routingu i uniezależnienie się w większym stopniu od zewnętrznych protokołów routingu. Jest to szczególnie ważne ze względu na fakt, iż powszechnie używane protokoły routingu często nie są optymalizowane pod względem transmisji czasu rzeczywistego. Istnienie sieci nakładkowej pozwala natomiast na cykliczny pomiar jakości połączenia między węzłami i wyznaczenie alternatywnych ścieżek routingu, znacznie bardziej odpowiadających wymogom transmisji real-time.

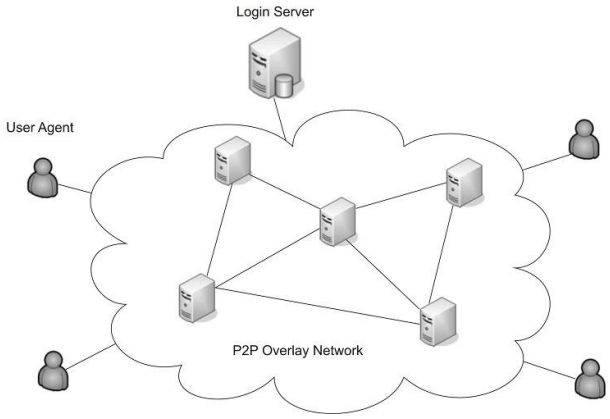
Kolejną cechą sieci P2P, która jest istotna w zaprojektowanym systemie, jest zdolność do samoorganizacji, tj. zakłada się, iż każdy węzeł może przyłączyć się do sieci lub ją opuścić w dowolnym czasie bez ryzyka utraty stabilności systemu. Dzięki samoorganizacji system może zostać łatwo rozbudowany, istotnie wzrasta również jego niezawodność i zmniejsza się podatność na awarie i ataki. Wybrany model infrastruktury posiada także zalety jeżeli chodzi o wydajną dystrybucję mediów strumieniowanych, gdyż na sieci nakładkowej możliwa jest implementacja usługi multicast (na poziomie warstwy aplikacji), a w wypadku usług typu VoD możliwe jest przechowywanie w pamięci podręcznej najczęściej strumieniowanych filmów oraz bilansowanie obciążenia sieci poprzez wprowadzenie mechanizmu load-balancing.

Następną zaletą wybranej infrastruktury jest automatyczna eliminacja problemów występujących w wypadku klientów znajdujących się poza NATem. W normalnych okolicznościach, kiedy obaj klienci znajdują się poza NATem, nie mogą ustanowić bezpośredniego połączenia. Istnieją co prawda techniki takie jak Session Traversal Utilities for NAT (STUN) [19], które mogą wykrywać obecność translatora adresów sieciowych i uzyskać numer portu, który NAT przydzielił połączeniu UDP aplikacji, jednak są one często nieefektywne [20]. Dlatego zazwyczaj w wypadku takich użytkowników wymagany jest dodatkowy serwer do przekazywania transmisji pomiędzy klientami. W większości przypadków dodatkowy serwer pośredniczący nie stanowi optymalnej trasy pomiędzy tymi dwoma klientami, a więc wprowadza on dodatkowe opóźnienie w komunikacji w czasie rzeczywistym. Z drugiej strony, w przypadku infrastruktury opartej o sieć nakładkową, węzły sieci są używane do routingu danych, więc żaden dodatkowy serwer pośredniczący nie jest konieczny, a ponieważ wybrano węzły, które tworzą optymalną ścieżkę pomiędzy klientami, opóźnienie transmisji jest zminimalizowane. Ponadto, dzięki architekturze sieci overlay i własnemu protokołowi routingu,

możliwe było zaimplementowanie dodatkowych mechanizmów bezpieczeństwa; są one opisane w rozdziale IX.

4.2 Komponenty składowe

Zaprojektowana aplikacja składa się z trzech głównych elementów: Serwera Głównego, węzłów sieci i terminali końcowych [21].



Rys. 4.2 Składniki infrastruktury.

Serwer Centralny (węzeł centralny)

Jak wskazuje nazwa, głównym zadaniem serwera jest zapewnienie usług uwierzytelnienia i autoryzacji. Każdy uczestnik, który chce się połączyć z systemem musi być najpierw zalogowany na tym serwerze. Rejestracja nowych użytkowników i zarządzanie kontami również są przez niego obsługiwane. Jeśli zachodzi taka potrzeba może on również wyliczać koszty związane z realizacją danej usługi.

Każdy węzeł musi przejść proces autoryzacji zanim zostanie przyłączony do infrastruktury. Odbywa się to w następujący sposób. W procesie logowania każdy z użytkowników zostaje przypisany do konkretnego węzła. Na wybór węzła mają wpływ lokalizacja geograficzna i informacje o obciążeniu. Serwer posiada aktualne informacje o stanie każdego z użytkowników, jego bieżący adres IP, numer portu i ID przypisanego węzła, jest więc

używany do określenia aktualnego położenia użytkownika przez innych, co jest niezbędne np. w usłudze VoIP, aby nawiązać połączenie. Ponieważ serwer ma pełną wiedzę na temat bieżącego stanu węzłów oraz jakości tras, odgrywa on główną rolę w procesie informowania innych węzłów o wszelkich zmianach w sieci, a zatem węzły mogą szybko przeliczać swoje tablice routingu i uniknąć tras powolnych/awaryjnych. Jak można zauważyć, obecność tego serwera jest kluczowa dla prezentowanej aplikacji, jest więc bardzo ważne aby zapewnić odpowiednie zasoby sprzętowe do jego działania.

Węzły sieci

Węzły są podstawowymi elementami składowymi każdej sieci P2P. Są one także zasadniczymi elementami opisywanego systemu. Ich głównym zadaniem jest pośredniczenie w transmisji danych oraz obsługa bezpośredniej komunikacji z terminalami użytkowników końcowymi. Jak zostało wspomniane poprzednio, podczas procesu logowania użytkowników końcowych do systemu, każdemu z nich przydzielany jest konkretny węzeł, używany następnie do pośredniczenia w procesie komunikacji między tym użytkownikiem a innymi elementami infrastruktury. Aby zwiększyć odporność infrastruktury na próby zablokowania dostępu (np. przez dostawcę Internetu), węzły wyposażone zostały w mechanizm pośredniczący, pozwalający im na przeprowadzenie w bezpieczny sposób procesu mediacji między użytkownikiem końcowym a serwerem głównym. Dotyczy to w szczególności procesu logowania. Każdy z węzłów wchodzących w skład infrastruktury posiada pełną wiedzę na temat wszystkich innych węzłów oraz jakości ścieżek pomiędzy nimi. Wiedza ta używana jest do budowy tablic routingu. Węzły w ramach współpracy z głównym serwerem wymieniają się informacjami odnośnie jakości ścieżek – cyklicznie raportują stan jakości połączeń ze swoimi sąsiadami oraz pobierają informacje o stanie pozostałych ścieżek. Ponieważ liczba zadań wykonywanych przez węzły jest znaczna, w celu usprawnienia modularności systemu oprogramowanie węzłów zostało opracowane zgodnie z paradygmatami programowania agentowego. W ten sposób system zyskał dodatkową elastyczność, gdyż możliwe stało się proste aktualizowanie oprogramowania – także zdalnie – działającego na węzłach. Ponadto, w zależności od aktualnych potrzeb, mogą być używane różne typy agentów – np. agenci inteligentni mający zdolność uczenia się i adaptowania do panujących warunków sieciowych. Poprzez wzajemną komunikację między-agentową mogą

dzielić się zdobytą wiedzą, a dzięki temu sprawniej wykonywać powierzone funkcje (np. routing).

Każdy z agentów działa wewnątrz własnego, pojedynczego węzła. Głównym zadaniem agenta jest stała kontrola jakości transmisji danych przesyłanych pomiędzy sąsiednimi węzłami. Agenci analizują takie parametry jak: stopień wykorzystania dostępnej przepustowości, liczbę gubionych pakietów oraz opóźnienia w transmisji. Ich zadaniem jest także testowanie innych, aktualnie aktywnie niewykorzystywanych połączeń, aby wykryć potencjalne zmiany jakości tych połączeń. W zależności od typu wykonywanych usług, mogą one dodatkowo analizować np. popularność aktualnie strumieniowanych mediów w usłudze VoD, a następnie automatycznie budować bazę cache'a, aby zmniejszyć obciążenie infrastruktury.

Z punktu widzenia bezpieczeństwa, każdemu z węzłów przypisany jest określony poziom wiarygodności, wyliczany na podstawie długoterminowej obserwacji zachowania tego węzła przez scentralizowany system reputacyjny działający na głównym serwerze. Dane dotyczące reputacji używane są następnie m.in. w procesie routingu oraz podczas wyboru nowego węzła centralnego.

Terminale końcowe

W zależności od typu wybranej usługi, terminale końcowe mogą stanowić pluginy do domowych odtwarzaczy sieciowych, aplikacje zainstalowane na komputerze (lub smartphonie), który używany jest przykładowo do nawiązywania i odbierania połączeń, itp. W standardowej konfiguracji terminale końcowe używają tylko dwóch portów: portu TCP na potrzeby sygnalizacji oraz UDP dla przesyłania strumienia real-time. Po zalogowaniu się do głównego serwera (bezpośrednio lub – gdy połączenie z serwerem jest blokowane – pośrednio z udziałem dowolnego węzła należącego do infrastruktury), terminal łączy się z węzłem przydzielonym przez ten serwer i jest gotowy do użycia.

Na potrzeby usług komunikacyjnych takich jak VoIP, możliwe jest także wsparcie mechanizmu wyszukiwania użytkowników oraz aktualizowanie statusu innych użytkowników znajdujących się w liście kontaktów. Gdy dany użytkownik zmienia swój status, pozostali użytkownicy, posiadający go na liście kontaktów, są automatycznie powiadamiani o tej zmianie. Mechanizm ten działa dzięki bilateralnej relacji pomiędzy użytkownikami przechowywanymi w bazie danych na serwerze głównym. Po dodaniu nowego znajomego do listy kontaktów system

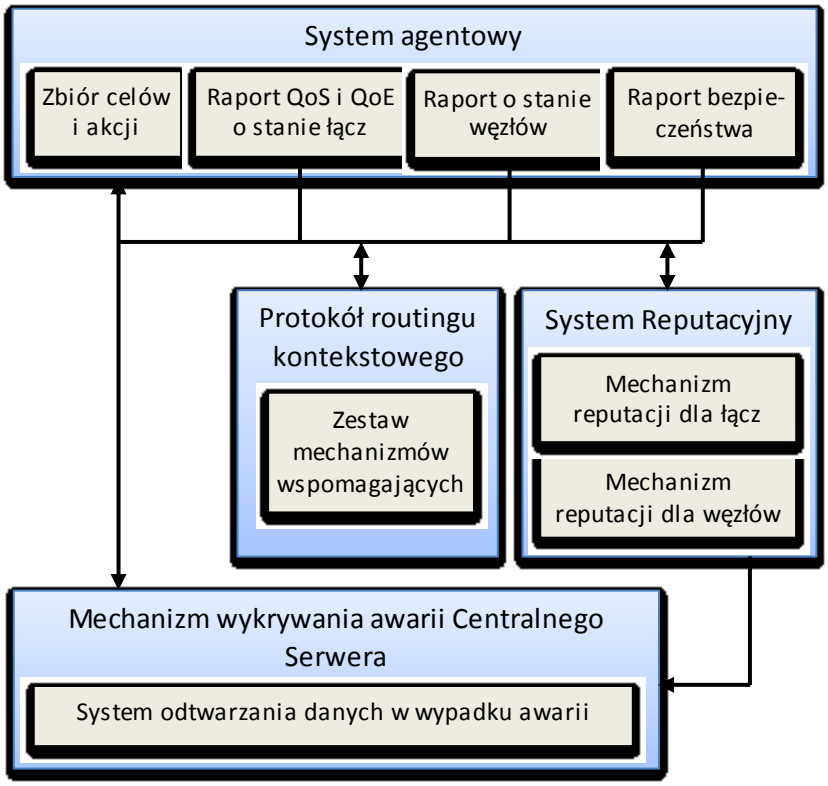
prosi dodawaną osobę o pozwolenie na sprawdzenie jej statusu. W przypadku, gdy takie pozwolenie zostanie udzielone, tworzona jest w bazie danych relacja pomiędzy tymi użytkownikami, tak aby możliwe było natychmiastowe wzajemne informowanie się o zmianie statusu.

Pozostałe elementy

Poza dotychczas wymienionymi elementami, prezentowany system może być w prosty sposób rozszerzony o dodatkowe wyspecjalizowane węzły, takie jak: bramy łączące z tradycyjną siecią telefonii stacjonarnej PSTN, bramy do innych standardów telefonii VoIP (np. SIP lub Skype), bazy danych z materiałami multimedialnymi, systemy telewizyjne itd.

4.3 Wykorzystane mechanizmy

Aby system mógł prawidłowo funkcjonować, niezbędne było podjęcie wielu decyzji konstrukcyjnych oraz opracowanie szeregu mechanizmów wspomagających. Począwszy od decyzji odnośnie rodzaju infrastruktury P2P, poprzez wykorzystanie algorytmu routingu na potrzeby przesyłania danych czasu rzeczywistego, kończąc na mechanizmach niezbędnych dla zapewnienia wysokiej niezawodności działania oraz bezpieczeństwa pracy całego systemu. Ze względu na złożony charakter tych mechanizmów, każdemu z nich został poświęcony osobny rozdział niniejszej rozprawy. Przegląd przez typowe architektury sieci P2P oraz cechy architektury opracowanej na potrzeby systemu opisane zostały w rozdziale V. Rozdział VI poświęcony jest mechanizmowi reputacyjemu, będącemu integralną częścią opracowanego systemu i istotnie wpływającemu na niezawodność oraz bezpieczeństwo jego pracy. Natomiast w rozdziale VII zawarte są informacje związane z routingiem oraz szczegółowo opisany jest Framework Routingu kontekstowego opracowany na potrzeby prezentowanego w niniejszej rozprawie systemu. Ponadto w rozdziale VIII opisany został autorski mechanizm odtwarzania centralnego serwera używany do przywrócenia systemowi funkcjonalności w wypadku niedostępności tego serwera (np. w wyniku awarii lub ataku). Aspekty związane z bezpieczeństwem systemu opisane zostały natomiast w rozdziale IX.



Rys. 4.3 Mechanizmy wykorzystane w systemie.

4.4 Bezpieczeństwo systemu

Aby zapewnić wysoki poziom bezpieczeństwa pracy systemu postanowiono posłużyć się jednocześnie metodami z zakresu hard-security jak i soft-security. W skład tych pierwszych wchodzą standardowe mechanizmy kryptograficzne używane do uwierzytelniania, autoryzacji oraz szyfrowania przesyłanych danych. Aspekt soft security realizowany jest natomiast przez wbudowany w system mechanizm reputacji [22]. Reputacja jest szczególnie ważna w wypadku systemów o infrastrukturze opartej na sieci P2P, gdzie członkami sieci mogą być anonimowe węzły dołączające do infrastruktury i opuszczające ją w dowolnym czasie [23]. Zachowanie tego typu węzłów w sieci jest nieprzewidywalne, a w wypadku ich dużej awaryjności lub złośliwego działania, system pozbawiony mechanizmów reputacji jest narażony na znaczne szkody lub nawet paraliż.

W celu zachowania spójności pracy, szczegółowy opis mechanizmów związanych z bezpieczeństwem został podzielony na podrozdziały poświęcone poszczególnym

mechanizmom. Aspekty związane z eliminacją wadliwych oraz złośliwych węzłów zostały omówione w części 6.6, problemy związane z możliwością istnienia złośliwych koalicji oraz sposoby ich identyfikacji w podrozdziale 6.7. Aspekty związane z procesem autoryzacji oraz uwierzytelniania, w szczególności sposób generowania biletu, opisane zostały natomiast w podrozdziałach 8.5-8.6.

Rozdział V

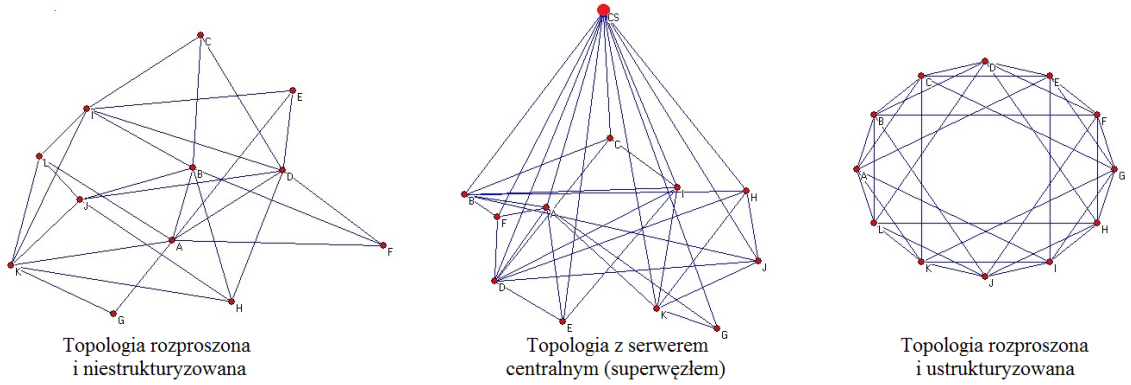
Sieci peer-to-peer

5.1 Wstęp do sieci P2P

Istnieje kilka typów sieci peer-to-peer, które różnią się od siebie sposobem ustrukturyzowania elementów składowych w ramach sieci, czasem niezbędnym do wyszukiwania danych, poziomem niezawodności w wypadku awarii ich elementów składowych oraz odpornością na próby blokowania/ataki. Opis trzech podstawowych klas sieci P2P, wraz z porównaniem ich podstawowych cech, zamieszczony jest w podrozdziale 5.2. Z kolei w podrozdziale 5.3 opisana została architektura opracowana na potrzeby systemu będącego tematem rozprawy, łącząca w sobie najistotniejsze cechy z istniejących infrastruktur, a jednocześnie niwelująca ich największe słabości.

5.2 Porównanie różnych typów sieci P2P

Standardowo możemy podzielić sieci peer-to-peer na trzy kategorie, w zależności od zastosowanej topologii [24]: sieci z serwerem centralnym, sieci rozproszone i niestrukturyzowane oraz sieci rozproszone i ustrukturyzowane. Schematy ich struktury pokazane zostały na rysunku 5.1.



Rys. 5.1 Typy sieci peer-to-peer.

5.2.1 Topologia z serwerem centralnym

Jest to najprostsza i najłatwiejsza w implementacji topologia. W jej przypadku serwer centralny używany jest przede wszystkim do zarządzania zasobami udostępnianymi przez poszczególne węzły sieci. Zakłada się, iż każdy węzeł należący do systemu informuje serwer centralny o swoich dostępnych zasobach, dzięki czemu mogą one zostać zaindeksowane w bazie danych serwera. Podejście to pozwala w prosty sposób zaprojektować wydajny i niezawodny mechanizm przeszukiwania wszystkich dostępnych zasobów. Po zlokalizowaniu konkretnego zasobu, dalsza interakcja z serwerem centralnym jest zbędna, gdyż zainteresowane węzły mogą bezpośrednio uczestniczyć w wymianie danych. Ponadto serwer centralny może być także używany do autoryzacji i uwierzytelniania oraz udostępniania usługi odnajdowania (lookup service), która jest m.in. szczególnie istotnym elementem każdego systemu telefonii internetowej. Jednakże znaczącą wadą tej topologii P2P jest fakt, iż serwer centralny stanowi pojedynczy punkt awarii (tzw. „single point of failure"), a więc sieć oparta o tę topologię jest wysoce wrażliwa na ataki typu DDOS oraz inne próby blokowania jej działania. Ze względu jednak na prostotę budowy, szybkie i niezawodne usługi zarządzania użytkownikami oraz bardzo szybkie wyszukiwaniem zasobów (w czasie $O(1)$), topologia ta wciąż jest preferowanym wyborem dla wielu usług P2P (m.in. aplikacja Skype używa centralnego serwera logowania [25]).

5.2.2 Sieć rozproszona i niestrukturyzowana

Jest to sieć, w której każdy z węzłów tworzących infrastrukturę traktowany jest jednakowo, a topologia ma postać losowej siatki częściowej (tzw. „random mash"). Zaletą takiej topologii jest dobra skalowalność oraz odporność na awarie. Sieci tego typu, ze względu na brak pojedynczych punktów awarii są także znacznie odporniejsze na próby ataków i blokowania usługi. Jednakże w strukturach tego typu brak jest wydajnego mechanizmu wyszukiwania. Przeszukiwanie zasobów odbywa się przy wykorzystaniu techniki zalewania (tzw „network flooding"), która jest czasochłonna, prowadzi do generowania wewnątrz sieci znacznej ilości transferu związanego z sygnalizacją oraz nie jest w pełni wiarygodna (nie wszystkie zasoby mogą zostać wyszukane w akceptowalnym czasie). Analogicznie do wyszukiwania, także autoryzacja

i uwierzytelnianie użytkowników oraz proces ich odnajdowania nie mogą zostać wykonane w efektywny sposób.

5.2.3 Sieci rozproszone i ustrukturyzowane

W sieciach tego typu topologia ma ustalona strukturę. Jest ona najczęściej budowana poprzez użycie rozproszonych tablic mieszających (DHT) [26] i może przyjmować rożne postacie, jak np. siatki, pierścienia (Chord [27]) albo torusa [28]. DHT pozwala na implementację mechanizmu wyszukiwania podobnego do tablic mieszających (tzw. „hash table"), w którym konkretne zasoby (jak np. prawa własności do pliku) powiązywane są z konkretnym węzłem sieci P2P. Topologia ta ma liczne zalety: dobry poziom skalowalności, satysfakcjonujący współczynnik odporności na błędy oraz awarie, a także szybki i niezawodny mechanizm wyszukiwania (najczęściej wymaga rozesłania $O(logN)$ wiadomości w celu przeszukania sieci składającej się z N węzłów). Z tych względów sieci te są często używane do implementacji usług wymiany plików, jednakże w wielu aspektach (typu zarządzanie kontami użytkowników oraz ich szybkie odnajdowanie) topologia P2P z centralnym serwerem jest wciąż znacznie wydajniejsza.

Obszerniejsze porównanie infrastruktur można znaleźć między innymi w pracach [26] oraz [24].

5.3 Architektura hybrydowa z mechanizmem self-healing

W przypadku systemu będącego tematem rozprawy, aspekty takie jak szybki czas wyszukiwania informacji o użytkownikach (np. w celu nawiązania połączenia), sprawna aktualizacja stanu sieci oraz dystrybucja danych odnośnie reputacji węzłów są bardzo istotne. Topologia z serwerem centralnym charakteryzowałaby się najwyższą wydajnością w realizacji tych zadań, jednak ryzyko awarii centralnego serwera stwarza istotne zagrożenia dla niezawodności i bezpieczeństwa pracy systemu. Z drugiej strony infrastruktury całkowicie rozproszone lub ustrukturyzowane znacznie lepiej spełniałyby wymogi bezpieczeństwa i niezawodności, lecz nie spełniałaby podstawowych założeń dotyczących wydajności. Aby zaradzić temu problemowi, autor opracował mechanizm self-healing, mający zastosowanie w wypadku topologii z serwerem centralnym i nadający jej niezawodności znanej z infrastruktur

całkowicie rozproszonych, przy jednoczesnym zachowaniu zalet związanych z szybkim czasem wyszukiwania i aktualizacji. Idea mechanizmu polega na obserwacji zachowania centralnego serwera, a wypadku jego awarii lub innych problemów z jego funkcjonalnością, automatycznego wyznaczania nowego serwera centralnego oraz odbudowy w bezpieczny sposób bazy danych z informacjami krytycznymi dla działania systemu (np. bazy danych o użytkownikach). Szczegółowy opis działania mechanizmu znajduje się w rozdziale VIII.

Rozdział VI

Systemy reputacyjne

6.1 Wstęp do systemów reputacyjnych

W rozdziale tym omówiony zostanie system reputacyjny zastosowany w opracowanym w ramach niniejszej rozprawy systemie wspomagającym transmisję czasu rzeczywistego. Na wstępie przedstawiona zostanie definicja i zadania reputacji, podstawowe typy systemów reputacyjnych oraz zdefiniowany zostanie matematyczny model takiego systemu. Szczegółową prezentację systemu opracowanego w niniejszej pracy poprzedzi przedstawienie konkretnych przykładów infrastruktur reputacyjnych opisanych w literaturze. Poruszony zostanie również temat bezpieczeństwa systemów reputacyjnych, ze szczególnym naciskiem położonym na problem istnienia szkodliwych koalicji, który jest zagadnieniem złożonym i częściowo wciąż otwartym. W ostatnim podrozdziale opisany będzie opracowany w ramach niniejszej pracy algorytm służący do wykrywania i blokowania tego typu koalicji. Na początku jednak, aby zobrazować sens użycia reputacji w projekcie, przedstawione zostaną jej podstawowe zadania.

System reputacyjny jest bardzo istotną częścią niniejszego projektu. Reputację zastosowano zarówno do oceny poszczególnych węzłów tworzących sieć P2P, jak i oceny jakości łącz transmisyjnych między nimi. Najistotniejsze korzyści, jakie pozwoliło osiągnąć wprowadzenie reputacji, są następujące:

a) W przypadku reputacji węzłów:
   - detekcja węzłów wadliwych lub złośliwych (tzw. malicious), których istnienie pogarsza ogólną wydajność systemu lub które aktywnie dążą do paraliżu systemu,
   - pomoc w wyborze najlepszych (charakteryzujących się najwyższą niezawodnością, poziomem bezpieczeństwa i jakością usługi) węzłów przy definiowaniu bezpiecznych i optymalnych tras routingu,
   - wykrywanie i eliminowanie „czarnych dziur" w routingu,

- wybór węzłów krańcowych, do których przyłączani są użytkownicy (w pierwszej kolejności przyłączani są do węzłów z lepszą reputacją),
- detekcja węzłów negatywnie ocenianych przez użytkowników końcowych, oferujących niski standard usługi lub posiadających luki bezpieczeństwa,
- wyznaczanie kandydatów na nowy centralny serwer w wypadku awarii obecnego,
- wyznaczanie węzłów, które mogą zostać notariatami (idea notariatu zdefiniowana zostanie w rozdziale VIII),
- detekcji węzłów typu selfish, które wykorzystują infrastrukturę dla swoich celów (np. obsługi tylko wybranych, przyłączonych bezpośrednio użytkowników), lecz nie chcą działać jako przekaźniki do transmisji zewnętrznej.

b) W przypadku reputacji łącz:
   - reputacja używana jest do opracowywania tras routingu. Bierze ona pod uwagę nie tylko aktualny stan łącza, ale także jego niezawodność i stałość w czasie. Na jej podstawie wybierane są najlepsze ścieżki. Każda ścieżka jest oceniana przez dwa węzły – bezpośrednio połączonych ze sobą sąsiadów. W ten sposób unika się sytuacji, gdy pojedynczy węzeł, chcąc przechwycić transmisję, będzie oszukiwał podając zawyżoną jakość łącza.

6.2 Definicja reputacji

Reputacja często definiowana jest jako „opinia, jaką ktoś lub coś ma wśród ludzi" [29]. Reprezentuje więc ona kolektywne zdanie członków danej społeczności na temat rzetelności lub niezawodności konkretnej jednostki. A zatem, w odróżnieniu od takich cech jak np. zaufanie, które może być wynikiem indywidualnej i subiektywnej oceny, reputacja, przy założeniu, że do jej wyliczenia użyto pełnej dostępnej wiedzy, jest oceną obiektywną i odzwierciedla opinię całej grupy. Cecha ta czyni z niej użyteczne narzędzie także w systemach IT.

Systemy reputacyjne, mające na celu zarządzanie reputacją, z powodzeniem używane są w wielu rozmaitych zastosowaniach. Do najpopularniejszych z nich należą systemy wspomagania decyzji, szczególnie gdy dany użytkownik dysponuje ograniczoną wiedzą na temat

innych członków systemu. Przykładowo, w systemach aukcyjnych typu Allegro[1] lub eBay[2] system reputacyjny pomaga przy wyborze kontrahentów, generując na podstawie przesłanych wcześniej opinii ocenę reputacyjną każdego z użytkowników. Reputacja okazuje się także bardzo pomocna przy detekcji nieprawidłowo funkcjonujących węzłów w sieciach P2P. W sieciach tego typu ataki na infrastrukturę ze strony jej członków, zwykle anonimowych, są zjawiskiem bardzo częstym [30]. Skutki występowania wadliwych węzłów w sieci P2P są szczególnie dotkliwe w systemach wymiany plików, gdzie złośliwe węzły odpowiadają na dowolne zapytanie wyszukiwania, po czym dostarczają celowo niewłaściwy lub uszkodzony plik. Powszechność tego problemu oraz realna groźba paraliżu sieci P2P doprowadziła do znacznego wzrostu zainteresowania systemami reputacyjnymi i opracowania wielu nowych modeli w dużej mierze niwelujących to zjawisko. Przegląd przez najpopularniejsze z nich można znaleźć między innymi w pracy [31]. Problem jednak nie został całkowicie rozwiązany, gdyż wraz z rozwojem zaawansowanych systemów detekcji wiążących zagadnienia reputacji, zaufania oraz wirtualnej tożsamości, pojawiają się także nowe, bardziej rozbudowane metody ataków [32].

Z punktu widzenia budowy systemy reputacyjne możemy podzielić na [22]:

a) Systemy Scentralizowane – systemy tego typu charakteryzują się istnieniem centralnej jednostki odpowiedzialnej za gromadzenie, przetwarzanie i przesyłanie ocen reputacyjnych.

   Podczas każdej interakcji uczestniczący w niej agenci obserwują wzajemne zachowanie, a po zakończeniu interakcji przesyłają ocenę jej przebiegu do centralnej jednostki gromadzącej. Na podstawie nadesłanych danych zostają wyliczone nowe oceny reputacyjne, a następnie są one przesyłane do zainteresowanych jednostek. Scentralizowane systemy reputacyjne, ze względu na swoją prostotę budowy i szybkość działania, są bardzo popularne.

b) Systemy Rozproszone – w systemach tego typu brak jest konkretnej centralnej jednostki, która odpowiadałaby za gromadzenie i przetwarzanie nadesłanych ocen. W jej miejsce stosuje się wiele lokalnych centrów reputacji lub też każdy z członków systemu zajmuje się indywidualnie jej obsługą.

[1] www.allegro.pl
[2] www.ebay.com

Agenci po każdej dokonanej transakcji zapamiętują jej wynik budując w ten sposób bazę danych nt. przeszłych doświadczeń. Dysponują oni ponadto mechanizmem pozwalającym na wymianę zgromadzonych opinii między sobą. Ostatecznie każdy z nich, po wcześniejszym zgromadzeniu wszystkich dostępnych ocen od pozostałych agentów, wylicza reputację indywidualnie. Systemy reputacji rozproszonej stosuje się najczęściej w wypadkach, gdy infrastruktura na której działają, jest również rozproszona i nie dysponuje żadną centralną jednostką. Bazuje na niej wiele serwisów P2P takich jak Napster, KaZaA lub Gnutella. Używana jest także w sieciach Ad-Hoc przy definiowaniu poziomu zaufania węzłów.

Innym kryterium charakteryzacji wielu istniejących odmian systemów reputacyjnych jest podział ze względu na metodę obliczania ocen reputacyjnych:

a) Proste systemy sumujące – są to najprostsze systemy, w których ocenę reputacyjną definiujemy jako różnicę pomiędzy nadesłanymi pozytywnymi i negatywnymi opiniami. Do tej kategorii można również zaliczyć systemy wyliczające reputację jako średnią z przesłanych opinii.
b) Systemy reputacyjne oparte na teorii prawdopodobieństwa – w literaturze znaleźć można dwa rodzaje tego typu systemów: systemy oparte o prawdopodobieństwo Bayesowskie [33] oraz o logikę probabilistyczną [34]. Pierwsza z metod bazuje na przewidywaniu przyszłego zachowania ocenianego agenta na podstawie analizy funkcji gęstości prawdopodobieństwa. W systemach tego typu dane są gromadzone poprzez jednostkę centralną, a oceny liczone są biorąc pod uwagę wszystkie nadesłane opinie – stanowią zatem dobry wybór w wypadku scentralizowanych systemów reputacyjnych. Systemy oparte o logikę probabilistyczną mogą być natomiast używane w systemach rozproszonych, gdy nie dysponuje się pełną aktualną wiedzą, choć mogą one wtedy prowadzić do powstania subiektywnych, niejednakowych ocen.
c) Systemy bazujące na logice rozmytej – używa się ich najczęściej w sytuacji, gdy niemożliwe było zgromadzenie wystarczającej ilości danych statystycznych, aby użyć metod probabilistycznych (np. próbując sklasyfikować rzadkie zjawiska). Szerszy ich opis można znaleźć m.in. w pracy [35].

Bardziej szczegółowy opis konkretnych implementacji systemów reputacyjnych znajduje się podrozdziale 6.4.

6.3 Model reputacji

Zagadnienia związane z reputacją są tematem badań wielu dziedzin nauki, także humanistycznych, czego wynikiem jest trudność w jednoznacznym scharakteryzowaniu tego pojęcia oraz brak zdefiniowanego uniwersalnego modelu systemu reputacyjnego. Jednakże istnieje kilka modeli dość dobrze opisujących większość tych aspektów reputacji, które są najistotniejsze w niniejszej pracy. Jednym z nich jest model oparty o zagadnienia teorii mnogości przedstawiony w pracy [36]. Model ten nie obejmuje jednak ważnych kwestii związanych z zaufaniem oraz reputacją rekomendacyjną. Dlatego też, w oparciu o ideę przedstawienia systemu reputacyjnego w ujęciu teorii mnogości, w prezentowanej pracy został zdefiniowany i przedstawiony poniżej nowy, rozszerzony model sytemu reputacyjnego. Zarówno w opisie ogólnego modelu, jak i późniejszym szczegółowym opisie użytego w pracy konkretnego systemu reputacyjnego, główny nacisk położono na scentralizowany system reputacyjny. Jednak projekt ten w prosty sposób można rozszerzyć do modelu rozproszonego. Interesującym zagadnieniem z punktu rozwoju projektu opisywanego w niniejszej pracy byłoby także opracowanie hybrydowego systemu reputacyjnego. W systemie takim główną rolę pełniłby scentralizowany mechanizm reputacyjny, jednak w wypadku braku komunikacji z serwerem centralnym lub w wypadku podejrzenia, iż uległ on awarii/przejęciu, mogłoby nastąpić czasowe przełączenie na system rozproszony.

Podstawą tworzenia oceny reputacyjnej są cząstkowe oceny wynikające wyłącznie z bezpośrednich doświadczeń powstałych w wyniku wzajemnej interakcji agentów. Po każdej zakończonej interakcji lub po ustalonym stałym interwale czasowym t_s (w wypadku gdy interakcja wciąż przebiega), biorący udział w interakcji agent ocenia jej przebieg. Ocenie podlega zarówno zachowanie drugiego agenta, jak i jakość łącza komunikacyjnego pomiędzy nimi. Do tego celu wykorzystywana jest funkcja ewaluacyjna π. Zakładając dowolną postać funkcji oceny dla każdego z agentów, można ją zdefiniować jako:

$$\pi: A \times B \times T \rightarrow Q, \tag{6.1}$$

gdzie $B \coloneqq A \cup L$ jest zbiorem obiektów ocenianych, zdefiniowanym jako suma zbioru agentów A oraz zbioru połączeń L. Kolejno, $T = [0,\ t_{act}]$ jest względnym czasem liczonym od chwili rozpoczęcia pracy systemu, natomiast Q jest zbiorem wszystkich możliwych ocen. Wielkość Q może być zdefiniowana dowolnie – w zależności od przyjętej konfiguracji systemu najczęściej jest to zbiór binarny $\{0,1\}$ lub zbiór ciągły na przedziale $[0,1]$.

Zdefiniujmy następnie zbiór wszystkich zarejestrowanych i ocenionych bezpośrednich doświadczeń dla danego zbioru agentów A:

$$I \coloneqq \{(a,b,t,c)\ \in A \times B \times T \times Q\ :\ a \neq b\}. \tag{6.2}$$

Po wprowadzeniu dyskretnego czasu $T_D = \{0,1,\dots,n\}$ o okresie obserwacji $\Delta t = \left\lceil \frac{t_{act}}{n} \right\rceil$, zbiór wszystkich bezpośrednich ocen z uwzględnieniem okresu obserwacji dany jest jako:

$$\begin{aligned} E \coloneqq \{(a,b,t,s) \in A \times B \times T_D \times S : \exists_{i \in I} i = (a,b,\cdot,\cdot) \wedge\ s \\ = agr(a,b,t,I) \wedge t\Delta t - \tau(i) \leq \Delta t\} \end{aligned} \tag{6.3}$$

Funkcja $\tau: E \cup I \to T_D \cup T$ zwraca czas interakcji bądź wystawienia oceny, natomiast $agr: A \times B \times I \times T_D \to Z$ jest funkcją agregującą, służącą do przetwarzania zgromadzonych w danym okresie obserwacji. Najczęściej ma ona postać sumy, średniej lub funkcji wyboru najaktualniejszej z ocen.

Podzbiór wszystkich ocen wystawionych przez agenta a na temat obiektu b przyjmuje postać:

$$E_b^a = \{e \in E : e = (a,b,\cdot,\cdot)\}. \tag{6.4}$$

Zbiór wszystkich ocen na temat b dany jest przez:

$$E_b \coloneqq \{e \in E : e = (\cdot,b,\cdot,\cdot)\} = \bigcup_{a \in A} E_b^a, \tag{6.5}$$

a zbiór wszystkich ocen na temat b w okresie obserwacji $t \in T_D$ oznaczamy jako:

$$E_{b,t} \coloneqq \bigcup_{a \in A} E_{b,t}^a. \tag{6.6}$$

Natomiast wszystkie najaktualniejsze oceny na temat $b \in B$ przesłane od chwili czasowej t_0 dane są przez:

$$E_b^{t_0} = \bigcup_{a \in A} \left\{ e \in E_b^a : \tau(e) \geq t_0 \wedge \forall_{x \in E_b^a} \tau(e) \leq \tau(x) \Rightarrow x = e \right\}. \tag{6.7}$$

Dodatkowo listę posortowanych względem czasu elementów zbioru $E_{b,t}$ oznaczamy jako $L_{b,t}$.

Możemy teraz zdefiniować reputację rekomendacyjną R_rec, rozumianą jako prawdomówność danego agenta przy ocenie działania danego obiektu w postaci:

$$R\_rec \coloneqq \{(a,b,t,j) \in A \times B \times T_D \times R_{rec} \mid j = rec(a,b,t,E)\}, \tag{6.8}$$

gdzie $rec: A \times B \times T \times E \to R_{rec}$ jest funkcją oceny reputacji rekomendacyjnej oraz R_{rec} jest zbiorem możliwych ocen reputacji rekomendacyjnej. Ocenę reputacji rekomendacyjnej agenta a względem obiektu b w danym okresie obserwacji t oznaczamy jako $R_{rec,t}^{a,b} \coloneqq rec(a,b,t,E,REP)$. Należy jednak zauważyć, że reputacja rekomendacyjna może nie być uwzględniana w niektórych systemach reputacyjnych (oceny przesyłane przez agentów mają zawsze taką samą wagę) lub jej rolę może pełnić zaufanie (często odgórnie ustalone jako stała wartość). Ponieważ funkcja rec oceny reputacji rekomendacyjnej wymaga ocen ogólnej reputacji REP jako jednego z parametrów wejściowych, proces wzajemnego wyliczania R_rec/REP przebiega sekwencyjnie. Do wyliczanie R_rec w danej chwili t^n używa się ocen REP wyliczonych w chwili t^{n-1}. Natomiast sama reputacja jest zdefiniowana w następujący sposób:

$$REP \coloneqq \{(b,t,j) \in B \times T_D \times R \mid j = r(b,t,E,R\_rec)\}. \tag{6.9}$$

Funkcję $r: B \times T \times R \times R\_rec \to R$ nazywamy funkcją oceny reputacji, gdzie R jest zbiorem możliwych ocen reputacyjnych. Ocenę reputacyjną obiektu b w okresie obserwacji t oznaczamy jako R_t^b:=r(b,t,E,R_rec). Dla uproszczenia aktualną reputację oznaczamy jako R^b. Początkowa reputacja oraz początkowa reputacja rekomendacyjna dane są odpowiednio jako r_0 oraz $r_{rec,0}$.

Przyjmując powyższy model, działanie systemu reputacyjnego można scharakteryzować zbiorem następujących parametrów:

$$P = (\pi, Q, \Delta t, agr, rec, R_{rec}, r, R, r_0, r_{rec,0}). \tag{6.10}$$

Wartości powyższych parametrów dobiera się w zależności od wymagań stawianych systemowi reputacyjnemu.

6.4 Przegląd istniejących systemów reputacyjnych

W przypadku najprostszych systemów sumujących ocenę reputację wyznacza się jako różnicę pomiędzy ilością ocen pozytywnych oraz negatywnych. Definiując Q jako {-1,0,1}, a za funkcję reputacji rekomendacyjnej przyjmując funkcję stałą o wartości 1, otrzymujemy następującą postać funkcji reputacyjnej:

$$r(a) = \sum_{e \in E_a} ev(e), \tag{6.11}$$

gdzie funkcja $ev(e) = s \,|e = (a, b, t, s)$ zwraca ocenę bezpośredniej obserwacji w danym okresie. Dodatkowo, aby zminimalizować wpływ pojedynczych agentów na ocenę końcową (np. poprzez generowanie wielu ocen w krótkim czasie), oceny cząstkowe wystawiane przez poszczególnych agentów w pojedynczym przedziale czasowym bywają normalizowane poprzez zastosowanie odpowiedniej funkcji agregującej, np.:

$$agr\,(a, b, t_D) = \begin{cases} 1 & : \sum_{\substack{(a,b,t,\cdot)\in I \\ t_D \Delta t - t \le \Delta t}} \pi(a, b, t) > 0 \\ 0 & : \sum_{\substack{(a,b,t,\cdot)\in I \\ t_D \Delta t - t \le \Delta t}} \pi(a, b, t) = 0 \\ -1 & : \sum_{\substack{(a,b,t,\cdot)\in I \\ t_D \Delta t - t \le \Delta t}} \pi(a, b, t) < 0 \end{cases} \tag{6.12}$$

Technika ta stosowana jest np. w serwisie aukcyjnym eBay, gdzie w okresie tygodnia, niezależnie od ilości wystawionych ocen, ich wpływ na ocenę kontrahenta zawiera się w przedziale [–1,+1].

Innym przykładem algorytmu stosowanego w systemach reputacyjnych jest PageRank [37] przedstawiony w 1999 roku przez Larrego Paga. Algorytm ten służy do pozycjonowania stron internetowych w wyszukiwarce na podstawie ich reputacji (jest między innymi podstawą silnika Google[3]). Istota algorytmu polega na ocenie strony na podstawie liczby innych stron, które na nią wskazują. Pojedynczy link do danej strony traktowany jest jako pozytywna ocena, a zbiór możliwych ocen określony jest jako {0,1}. Ponieważ w systemie tym nie istnieje negatywna reputacja, sam algorytm PageRank nie jest zdolny do blokowanie stron w wynikach

[3] www.google.com

wyszukiwania. Dla danego zbioru stron P, reputację strony $u \in P$ wyznacza się przy pomocy tego algorytmu w następujący sposób:

$$R_{t+1}^{u} = cE(u) + \sum_{v \in N_u^-} \frac{R^v}{\#N_u^+}, \tag{6.13}$$

gdzie N_u^- oraz N_u^+ są odpowiednio zbiorami stron z linkiem prowadzącym do u, oraz stron do których prowadzą linku z u, natomiast E wektorem określającym startową reputację strony. Stały parametr c, zgodnie z założeniami podanymi w pracy [37] dobrany jest tak, aby $\sum_{u \in P} R^u = 1$. Ocena reputacji zawiera się w przedziale [0,1] (choć w przypadku wyszukiwarki Google skaluje się go do [0,10]).

Następnym przykładem systemu reputacyjnego, opracowanego tym razem z myślą o sieciach P2P jest mechanizm EigenTrust [38]. Mechanizm ten może być używany zarówno w sieciach scentralizowanych jak i rozproszonych, a ponadto jest prosty w implementacji. W systemie tym, na podstawie wektora lokalnych opinii c_i tworzonego przez każdego z agentów, generowana jest globalna macierz opinii C. Wektor ocen reputacyjnych t uzyskuje się poprzez wielokrotne mnożenie macierzy C przez siebie samą oraz przez wektor reputacji startowej t_0 $\mathrm{t} = (C^T)^n t_0$. W rozszerzonej wersji tej metody zakłada się istnienie w systemie z góry zaufanych agentów, dla których uwzględniana jest poprawka przy liczeniu reputacji: $t^{n+1} = (1-a) * C^T t^n + ap$, gdzie p jest wektorem poziomów zaufania agentów. Mechanizm ten w swojej rozszerzonej wersji jest zdolny do poprawnej oceny dobrych/uczciwych agentów, nawet gdy w systemie istnieje do 70% agentów wadliwych/złośliwych.

Z kolei w pracy [39] został zaprezentowany system reputacyjny beta oparty o metody statystyczne. Reputacja jest w nim wyliczana na podstawie analizy funkcji gęstości prawdopodobieństwa rozkładu beta o parametrach tworzonych przy wykorzystaniu nadesłanych ocen interakcji. Jego głównymi zaletami są duża elastyczność oraz silne ugruntowanie w teorii prawdopodobieństwa. Niestety jego podstawowa forma jest jednak dość prosta i często nie radzi sobie z wykrywaniem złośliwych węzłów.

6.5 System reputacyjny zaprojektowany w ramach pracy

W początkowej fazie projektowania systemu do wspomagania transmisji real-time rozważane było zaadaptowanie na jego potrzeby mechanizmu EigenTrust. Jednak jego działanie oparte jest na istnieniu w systemie agentów o predefiniowanym wysokim poziomie zaufania, co w wypadku awarii lub przejęcia zaufanego agenta stwarza istotne zagrożenie dla bezpieczeństwa całego systemu. Ponadto schemat połączeń w infrastrukturze peer-to-peer projektu będzie miał w większości przypadków strukturę grafu rzadkiego (połączenia między ograniczoną liczbą węzłów charakteryzujących się lokalnie najwyższą jakością łącza). W takiej sytuacji wymagane byłoby istnienie większej liczby lokalnych agentów o ustalonym wysokim poziomie zaufania. Ostatecznie więc postanowiono zbudować własny system reputacyjny bazujący na systemie beta. System ten w stosunku do modelu zaproponowanego w [39] został istotnie rozszerzony. Przede wszystkim został on wzbogacony o mechanizm reputacji rekomendacyjnej oraz mechanizm wygaszania reputacji w czasie. Dodatkowo opracowano na jego potrzeby algorytm wykrywania koalicji, co znacznie zwiększyło poziom bezpieczeństwa i niezawodność systemu. Szczegółowy opis zastosowanego w projekcie mechanizmu reputacyjnego zamieszczony został poniżej.

Zadaniem systemów reputacyjnych opartych na metodach bayesowskich jest przewidywanie przyszłego zachowania ocenianego obiektu na podstawie jego wcześniejszych zachowań oraz dodatkowej, posiadanej na jego temat wiedzy. Niech $X = (X_1, \dots, X_n)$ będzie pozyskanym w wyniku obserwacji ciągiem dyskretnych zmiennych losowych opisujących obserwowane zachowanie obiektu A. Zakładając, iż zmienne te przyjmują wartość 1 lub 0, odpowiednio w wypadku właściwego lub niewłaściwego zachowania obiektu A, faktyczny wynik obserwacji dany jest jako $x = (x_1, \dots, x_n)$ gdzie $\forall_{i=1}^{n} x_i \in \{0,1\}$. Dodatkowo niech θ będzie parametrem charakteryzującym zachowanie obiektu A. W statystyce bayesowskiej parametr θ nie jest traktowany jako stała, lecz jako zmienna losowa, co pozwala odzwierciedlić niepewność co do jej wartości. Przyjmujemy, że dla zmiennej tej istnieje rozkład prawdopodobieństwa dany jako $f(\theta)$ i reprezentuje on dotychczasową (*a priori*) dostępną wiedzę nt. zachowania A. Korzystając z twierdzenia Bayesa:

$$P(\theta|X = x) = \frac{P(X = x|\theta)P(\theta)}{P(X = x)}, \tag{6.14}$$

gdzie $P(X = x|\theta)$ jest prawdopodobieństwem warunkowym otrzymania danego ciągu próbek x przy założonym θ, możliwe jest wyznaczenie $P(\theta|x)$, zwanego rozkładem *a posteriori* zmiennej θ. Ponieważ mianownik $P(X = x)$ jest wartością stałą, można zapisać następującą zależność proporcjonalną:

$$P(\theta|X = x) \propto P(X = x|\theta)P(\theta). \tag{6.15}$$

Analogicznie w wypadku funkcji gęstości prawdopodobieństwa, prawdziwa jest proporcjonalność:

$$f(\theta|X = x) \propto P(X = x|\theta)f(\theta). \tag{6.16}$$

Traktując elementy zbioru X jako kolejne próby procesu Bernoulliego, chcemy wyznaczyć prawdopodobieństwo s, z jakim obiekt A zachowuje się poprawnie. Po przyjęciu początkowego rozkładu gęstości prawdopodobieństwa *a priori* $f(s)$, gęstość rozkładu prawdopodobieństwa *a posteriori* będzie proporcjonalna do: $f(s|X) \propto P(X|s)f(s)$. Przyjmując $k \leq n$ jako liczbę raportowanych zachowań prawidłowych obiektu A w ciągu x, $P(X = x|s)$ można wyliczyć jako:

$$P(x|s) = s^k(1 - s)^{n-k}, \tag{6.17}$$

więc ostatecznie funkcja gęstości rozkładu prawdopodobieństwa zmiennej s jest proporcjonalna do:

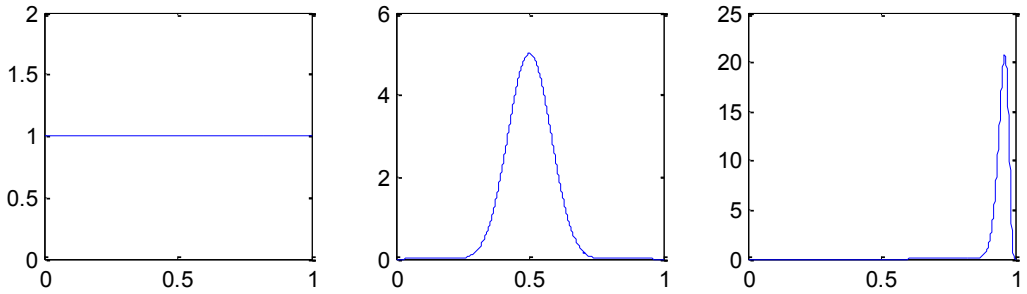
$$f(s|x) \propto s^k(1 - s)^{n-k}f(s). \tag{6.18}$$

Następnie, jako wiedza *a priori* został użyty rozkład beta dany następującą funkcją gęstości:

$$f(s|\alpha, \beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(a)\Gamma(\beta)} s^{\alpha-1}(1 - s)^{\beta-1} \ , \quad \text{gdzie } 0 \leq s \leq 1, \ \alpha, \beta \geq 0. \tag{6.19}$$

W projektowanym systemie reputacyjnym przyjęto, iż wstępnie nie jest posiadana żadna wiedza na temat ocenianych obiektów. Stąd też założenie, iż dowolna wartość s jest równie prawdopodobna w całym przedziale [0,1]. Dlatego też początkowe wartości α oraz β rozkładu beta zostały ustalone na 1, co daje rozkład jednostajny w [0,1].

W przypadku dysponowania dodatkową wiedzą, możliwe jest użycie innych parametrów α, β rozkładu. Ich wpływ na kształt funkcji gęstości prawdopodobieństwa został zilustrowany przykładami na rysunku.6.1.



beta(1,1) *beta*(20,20) *beta*(200,20)

Rys. 6.1 Postać funkcji gęstości prawdopodobieństwa dla różnych parametrów rozkładu beta(α,β).

Podstawiając za rozkład *a priori* rozkład $beta(\alpha,\beta)$, dla rozkładu *a posteriori* prawdziwe jest:

$$f(s|x) \propto s^k(1-s)^{n-k}\frac{\Gamma(\alpha+\beta)}{\Gamma(a)\Gamma(\beta)}s^{\alpha-1}(1-s)^{\beta-1} = \frac{\Gamma(\alpha+\beta)}{\Gamma(a)\Gamma(\beta)}s^{k+\alpha-1}(1-s)^{n-k+\beta-1}. \quad (6.20)$$

Wynika z tego, że rozkład *a posteriori* jest również rozkładem beta danym jako $beta(k+\alpha, n-k+\beta)$. Zakładając, że nieznana wartość *s* jest stała, przy dużej liczbie obserwacji danego obiektu $(n\to\infty)$, wartości rozkładu beta będą dążyły odpowiednio do $\alpha = n*s$ oraz $\beta = n*(1-s)$. Sam rozkład natomiast spełni zależność $\delta(x-a) = \lim_{\sigma^2} f(s)$, gdzie δ jest deltą Diraca, a jest wartością oczekiwaną rozkładu, a σ^2 wariancją.

Należy zauważyć, że funkcja *a posteriori* gęstości prawdopodobieństwa $f(s|x)$ odzwierciedla rozkład zmiennej *s*, która sama również jest prawdopodobieństwem. W związku z tym, aby określić prawdopodobieństwo (obarczone niepewnością) prawidłowego zachowania się obiektu *A* w przyszłości, można wyznaczyć prawdopodobieństwo przynależności *s* do danego zakresu $[s_1, s_2]$ jako $\int_{s_1}^{s_2} beta(s|\,k+\alpha, n-k+\beta)ds$ lub wyznaczyć wartość oczekiwaną rozkładu, daną jako $E(s) = \frac{\alpha}{\alpha+\beta}$.

Mimo iż przedstawiony powyżej schemat polegał na analizie procesu binarnego z tylko dwoma możliwymi wynikami $x_i \in \{0,1\}$, systemy reputacyjne oparte na rozkładzie beta można stosować także w sytuacji, gdy oceny nie są binarne. Jest to szczególnie istotne w wypadku systemu reputacyjnego zaprojektowanego w ramach niniejszej pracy, gdyż ocena będąca

wynikiem bezpośredniej interakcji ma bardziej złożoną postać. Wiąże się to ze znaczną ilością parametrów, jakie podlegają ocenie (zarówno przy ocenie używanego łącza, jak i przy ocenie innych węzłów) oraz potrzebie dokładniejszego stopniowania satysfakcji z interakcji. W związku z tym przyjęto, iż każda ocena wystawiana danemu obiektowi $B \in A \cup L$ będzie parą $(a, b) \in \mathbb{R}^2$, taką że $beta(a, b)$ jest rozkładem prawdopodobieństwa odzwierciedlającym poziom zadowolenia oceniającego z przebiegu interakcji z ocenianym obiektem B.

W wypadku gdy wstępna ocena cząstkowa g_i interakcji dana jest liczbą rzeczywistą z przedziału $[p, s]$ i wyznaczana jest periodycznie z danym odstępem czasu, w ostatecznej ocenie przesyłanej do systemu reputacyjnego w prosty sposób zawrzeć można także informację o wolumenie przeprowadzonych interakcji. Dla danego zbioru $\tilde{g} = \{\tilde{g}_1, \dots, \tilde{g}_n\}$, będącego zbiorem ocen cząstkowych g_i po normalizacji przedziału $[p, s]$ do [0,1], po uwzględnieniu wolumenu transakcji n, ocena przesyłana do systemu reputacyjnego będzie dana jako:

$$e = (n * \sum_{i=1}^{n} \tilde{g}_i \, , n * \sum_{i=1}^{n} (1 - \tilde{g}_i)). \tag{6.21}$$

Podsumowując powyższy model, prosty schemat działania systemu reputacyjnego opartego o metody bayesowskie można scharakteryzować następującymi krokami:

1. Początkowa reputacja dla każdego ocenianego obiektu ustalona jest na (1,1).
2. Każdy agent, używając zadanej funkcji π, ocenia innych agentów, z którymi ma bezpośrednie interakcje oraz jakość używanych łącz transmisyjnych.
3. Cząstkowe oceny są agregowane do oceny ostatecznej i przesyłane do scentralizowanego systemu reputacyjnego (nie częściej niż co ustalony interwał czasowy t_s).
4. Moduł agregujący gromadzi wszystkie najaktualniejsze (mieszczące się w interwale czasowym Δt) oceny oraz, używając funkcji agregującej agr (w najprostszej wersji jest to suma poszczególnych ocen) dla każdego ocenianego obiektu $x \in B$, wylicza nowe oceny w postaci par (a^x, b^x).
5. Najnowsze oceny są wyliczane zgodnie ze wzorami:

$$\alpha_{i+1}^x = \alpha_i^x + a^x, \tag{6.22}$$

$$\beta_{i+1}^x = \beta_i^x + b^x, \tag{6.23}$$

$$f_{i+1}^{x}(s|\alpha_{i+1}^{x},\beta_{i+1}^{x}) = \frac{\Gamma(\alpha_{i+1}^{x}+\beta_{i+1}^{x})}{\Gamma(\alpha_{i+1}^{x})\Gamma(\beta_{i+1}^{x})} s^{\alpha_{i+1}^{x}-1}(1-s)^{\beta_{i+1}^{x}-1}. \quad (6.24)$$

6. Aktualna ocena reputacyjna jest wyliczana jako wartość oczekiwana rozkładu f:

$$r(x) = E\left(f_{i+1}^{x}(s|\alpha_{i+1}^{x},\beta_{i+1}^{x})\right) = \frac{\alpha_{i+1}^{x}}{\alpha_{i+1}^{x}+\beta_{i+1}^{x}}. \quad (6.25)$$

7. System reputacyjny rozsyła najnowsze oceny do wszystkich agentów.

Zaprezentowany powyżej prosty schemat wymaga jednak pewnych usprawnień, gdyż w tej formie charakteryzuje się powolną reakcją na zmiany reputacji oraz nie jest odporny na wiele możliwych do przeprowadzenia ataków. W następnym podrozdziale przedstawione zostały dodatkowe usprawnienia, których użyto w końcowym, zaprojektowanym na potrzeby niniejszej pracy systemie.

a) Reputacja rekomendacyjna

W podstawowej wersji systemu reputacyjnego opartego na modelu bayesowskim przyjmuje się równe wagi dla każdego głosu. Założenie to nie zawsze jest właściwe i może prowadzić do manipulacji wynikami oraz destabilizacji systemu. Ponieważ prezentowany w obecnej pracy system oparty jest na sieci P2P, nowe węzły o niepewnej reputacji oraz nieznanych zamiarach mogą w dowolnej chwili dołączać do sieci. Ponadto aktualnie działające węzły mogą ulec awarii lub zostać przejęte i zacząć działać na szkodę infrastruktury. Wymagany jest więc mechanizm, który będzie obserwował nie tylko samo zachowanie węzłów, ale także ich zdolność do rzetelnej oceny innych. Opinie agentów przesyłających fałszywe informacje powinny być odrzucane przy wyliczaniu oceny reputacyjnej, natomiast opinie agentów, którzy charakteryzują się długim stażem oraz sprawdzoną rzetelnością powinny być brane pod uwagę z wyższą wagą niż opinie nowych agentów. Cel ten osiągnięty został poprzez użycie dodatkowej reputacji rekomendacyjnej, której ogólna idea została przedstawiona w pracy [40]. System reputacji rekomendacyjnej został zaprojektowany jako system scentralizowany, oparty także na modelu bayesowskim i rozkładzie beta.

W niniejszym systemie przyjęto, że reputacja rekomendacyjna danego agenta a będzie wyznaczana niezależnie względem każdego ocenianego przez a obiektu x, w odróżnieniu od ogólnej reputacji $r(a)$, którą przedstawić można jako liczbę z zakresu [0,1]. Będzie ona

stanowiła zatem zbiór cząstkowych reputacji $R_{rec}^{a,b}$ rekomendacyjnych, określających jaka jest zdolność agenta *a* do rzetelnej oceny agenta *b*. Podejście to pozwala na dokładniejsze zamodelowanie aspektów społecznych związanych z interakcją między agentami oraz chroni przed atakami mających na celu zaniżenie reputacji rekomendacyjnej (np. poprzez zamierzone, niesprawiedliwe traktowanie wybranego agenta gorzej niż pozostałych, aby wystawione przez niego oceny zostały następnie zdyskredytowane w wyniku porównania z ocenami nadsyłanymi przez większość). Przyjęto ponadto, że reputacja rekomendacyjna wyliczana będzie wyłącznie w wypadku interakcji między agentami.

Reputacja rekomendacyjna R_{rec} wyliczana jest na podstawie funkcji rec promującej agentów, których oceny są zbliżone oraz przyjmującej niską wartość w wypadku agentów, którzy przesyłają oceny istotnie różniące się od ocen większości. Funkcja rec została zaprojektowana tak, aby odzwierciedlać specyfikę opracowanego systemu P2P, czyli sytuacje, gdy istnieje mało ocen na temat danego agenta, oraz gdy któryś z oceniających agentów przesyła fałszywe opinie. W pierwszym kroku w danym okresie obserwacji *T* dla każdego agenta $a \in A$ oraz ocenianego przez niego agenta *b* wyliczany jest spółczynnik *m(a,b,T)*, będący różnicą pomiędzy średnią ważona (wagami są poprzednie reputacje rekomendacyjne) wszystkich ocen nadesłanych na temat *b*, a ocenami na jego temat nadesłanymi przez agenta *a*. Zgodnie z oznaczeniami wprowadzonymi w podrozdziale 6.3. współczynnik *m* przyjmuje postać:

$$m(a,b,T) = \left| \left(\frac{\sum_{q \in E_{b,T}} \mathbb{E}\big(ev(q)\big) * rec\,(aut\,(q), b, T-1)}{\sum_{q \in E_{b,T}} rec\,(aut(q), b, T-1)} \right) - \left(\frac{1}{\#E_{b,T}^{a}} \sum_{p \in E_{b,T}^{a}} \mathbb{E}(ev\,(p)) \right) \right|, \tag{6.26}$$

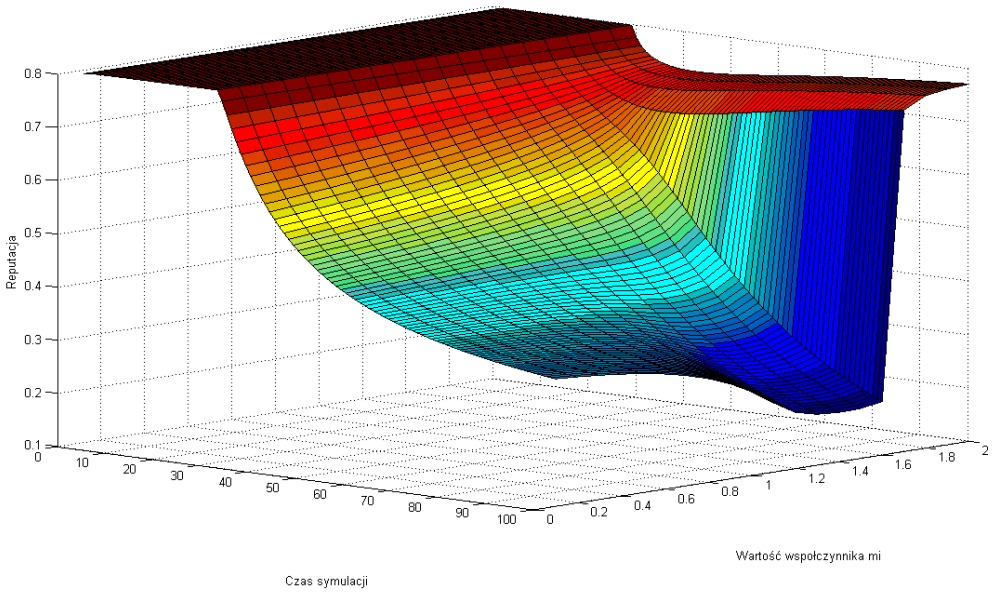
przy czym $\mathbb{E}\big(ev(q)\big)$ zwraca wartość oczekiwaną z danej oceny, natomiast $aut(q)$ podaje agenta, który jest autorem oceny q.

Wartość $m(a,b,T)$, wyliczona w ten sposób, jest w następnym kroku mnożona przez współczynnik wzmacniający *mi*, tak aby większe odchylenia od średniej były bardziej istotne. Dodatkowo wprowadzone zostały ograniczenia na dopuszczalny zakres zmienności reputacji rekomendacyjnej w pojedynczym okresie, tak aby mieściła się ona w przedziale [0,1]:

$$m'(a,b,T) = \begin{cases} 1 & : mi * m(a,b,T) > 1 \\ mi * m(a,b,T) & : 0 < mi * m(a,b,T) < 1 \end{cases} \tag{6.27}$$

Wartość współczynnika mi została dopasowana w ten sposób, by system był w stanie efektywnie blokować nieprawidłowe oceny nadsyłane przez agentów o niskiej prawdomówności.

Jako kryterium zachowania satysfakcjonującego progu efektywności przyjęto, że agenci nieprawdomówni mogą stanowić maksymalnie 60% wszystkich oceniających danego agenta. Na wykresie 6.2 przedstawiony został wpływ współczynnika mi na zmianę reputacji. Na potrzeby symulacji założono istnienie agenta o początkowej reputacji równej 0.8, względem którego począwszy od chwili czasowej t=30 zaczęto wysyłać negatywne opinie. Autorzy negatywnych opinii stanowili ostatecznie 60% wszystkich oceniających, a ich początkowa reputacja rekomendacyjna nie przekraczała wartości 0.4.



Rys. 6.2 Wpływ współczynnika mi na zmianę reputacji agenta w wyniku przesyłania negatywnych opinii na jego temat.

Symulacja wykazała, iż dla wartości współczynnika $mi < 1.7$ w wyniku nadsyłania negatywnych opinii reputacja uległa szybkiej degradacji, natomiast dla wartości powyżej 1.7 system był w stanie zignorować negatywne oceny i utrzymać wysoki wskaźnik reputacji, bazując na pozostałych 40% pozytywnych ocen nadsyłanych przez agentów o wysokiej reputacji rekomendacyjnej. Dla wyższych wartości współczynnika blokowanie również było efektywne, lecz zwiększało się także ryzyko ignorowania poprawnych opinii. Ostatecznie więc w mechanizmie reputacji rekomendacyjnej ustalono wartość mi = 1.7, jako spełniającą wymagane kryterium efektywności blokowania, a jednocześnie pozwalającą na uwzględnienie szerokiej skali poprawnych ocen.

Podsumowując powyższe rozważania, dla danego agenta $a \in A$ wartość reputacji rekomendacyjnej wyliczana jest zgodnie ze schematem:

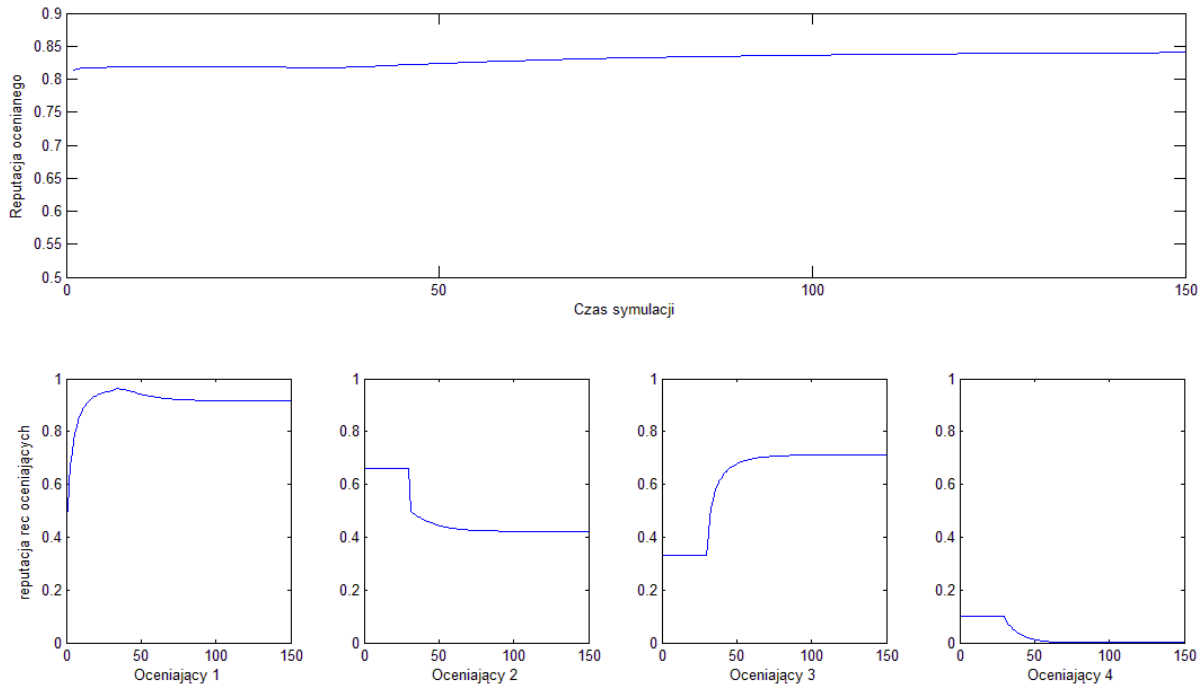
$$rec\_\alpha_{i+1}^{a,b} = rec\_\alpha_{i}^{a,b} + m'(a,b,t), \tag{6.28}$$

$$rec\_\beta_{i+1}^{a,b} = rec_{\beta_i^{a,b}} + \big(1 - m'(a,b,t)\big), \tag{6.29}$$

$$rec(a,b) = E\left(beta\left(rec_{\alpha_{i+1}}^{a,b}, rec_{\beta}{}_{i+1}^{a,b}\right)\right) = \frac{rec\_\alpha_{i+1}^{a,b}}{rec\_\alpha_{i+1}^{a,b} + rec\_\beta_{i+1}^{a,b}}. \tag{6.30}$$

Aby jednak nastąpiła aktualizacja reputacji rekomendacyjnej, wymagane jest, aby danego agenta oceniało jednocześnie co najmniej trzech innych agentów. Początkowa wartość reputacji rekomendacyjnej zależy od konkretnej realizacji systemu (wielkość sieci P2P, typ transmisji real-time, przyjęty poziom bezpieczeństwa, wymagana szybkość reakcji). Założono także, że nowi agenci dołączający do infrastruktury otrzymują reputację równą beta(1,9)=0.1, choć w wypadku zaufanych węzłów istnieje możliwość nadania im wyższych wartości.

Przykład działania reputacji rekomendacyjnej został zaprezentowany na rysunku 6.3.



Rys. 6.3 Zmiana reputacji ocenianego węzła (górny wykres) oraz reputacji rekomendacyjnej oceniających (dolne wykresy) w czasie.

W pierwszej fazie agent oceniany jest uczciwie przez 4 innych agentów (Oceniający 1), otrzymując ocenę reputacyjną równą 0.82. W czasie t = 30 sekund dołącza się kolejnych 6 agentów – dwóch z nich (Oceniający 2) o reputacji początkowej danej parą liczb (2,1), ocenia go 35% poniżej rzeczywistej wartości; dwóch następnych (Oceniający 3) o reputacji początkowej (1,2) ocenia go 25% wyżej, a dwóch ostatnich (Oceniający 4) o reputacji początkowej (1,10) ocenia go bardzo nisko. W systemie pozbawionym reputacji rekomendacyjnej reputacja ocenianego spadłaby do wartości 0.64, natomiast użycie reputacji rekomendacyjnej pozwoliło na filtrację nieprawidłowych ocen, przez co ocena uległa tylko niewielkim fluktuacjom.

b) Mechanizm zapominania

Podczas funkcjonowania systemu mogą pojawiać się istotne zmiany w zachowaniu jego elementów, które mogą mieć zarówno charakter negatywny, jak np. awarie węzła lub łącza, przejęcie kontroli nad którymś z węzłów itp. lub charakter pozytywny, np. udostępnienie większej liczby zasobów. Pożądane jest, aby system reputacyjny był w stanie dynamicznie reagować na tego typu zmiany i przez to ograniczać możliwe ryzyko wybrania aktualnie niepożądanego węzła/łącza, a jednocześnie wspomagał efektywne wykorzystanie bieżących zasobów. Jednym ze sposobów zwiększenia dynamiki systemu reputacyjnego jest wprowadzenie mechanizmu zapominania zaproponowanego m.in. w pracy [41]. Mechanizm ten oparty jest na idei, iż starsze oceny reputacyjne mogą nie odzwierciedlać aktualnego stanu ocenianego obiektu, a więc powinny mieć mniejszy udział na ocenę końcową niż oceny aktualne. Wprowadzając dodatkowy współczynnik μ, aktualną wartość współczynników α_n^x i β_n^x wylicza się zgodnie ze wzorami:

$$\alpha_n^x = a_n^x + \mu a_{n-1}^x + \mu^2 a_{n-2}^x + \cdots + \mu^n a_0 = \sum_{i=1}^{i=\#L_{x,n}} \mu^{\#L_{x,n}-i} ev(a_i) | a_i \in (a,b) = L_{x,n}[i], \tag{6.31}$$

$$\beta_n^x = b_n^x + \mu b_{n-1}^x + \mu^2 b_{n-2}^x + \cdots + \mu^n b_0 = \sum_{i=1}^{i=\#L_{x,n}} \mu^{\#L_{x,n}-i} ev(b_i) | b_i \in (a,b) = L_{x,n}[i], \tag{6.32}$$

co można zapisać także w postaci rekurencyjnej:

$$\alpha_{i+1}^x = \mu * \alpha_i^x + a^x, \tag{6.33}$$

$$\beta_{i+1}^{x} = \mu * \beta_{i}^{x} + b^{x}. \tag{6.34}$$

Równania w przypadku reputacji rekomendacyjnej mają analogiczną postać, przy czym współczynnik zapominania dany jest jako μ_{rec} .

Mechanizm zapominania można także zrealizować poprzez użycie tzw. okna przesuwnego, czyli uśredniając ostatnie w ocen:

$$\alpha_{i+1}^{x} = \frac{\sum_{j=i-w+2}^{i} \alpha_{j}^{x} + \alpha^{x}}{w}. \tag{6.35}$$

Przeprowadzone symulacje wykazały zbieżność obu mechanizmów – mechanizm oparty o wykładnicze wygaszanie ze współczynnikiem μ=0.9993 daje wyniki zbliżone do mechanizmu z wykorzystaniem okna przesuwnego o rozmiarze $w = 50$.

c) Mechanizm wygaszania

Podstawowym celem użycia systemu reputacyjnego w systemie wspomagającym transmisję real-time jest zwiększenie niezawodności infrastruktury i eliminacja wadliwych węzłów sieci. Jednakże ze względu na otwarty charakter infrastruktury przyjęto, iż węzły które zostały zablokowane jako posiadające złą reputację, nie są wyłączane na stałe. Dzięki temu węzły, których reputacja pogorszyła się w wyniku awarii lub chwilowego przejęcia nad nimi kontroli, po upływie określonego czasu mogą ponownie podłączyć się do infrastruktury i budować swoją reputację. Cel ten jest realizowany przez powolne wygaszanie przeszłej reputacji oraz przesuwanie jej w kierunku wartości oczekiwanej równej $\frac{1}{2}$. Mechanizm wygaszania, w odróżnieniu od mechanizmu zapominania, aktualizuje reputację wraz z upływem czasu, a nie tylko w przypadku nadsyłania kolejnych, nowszych ocen.

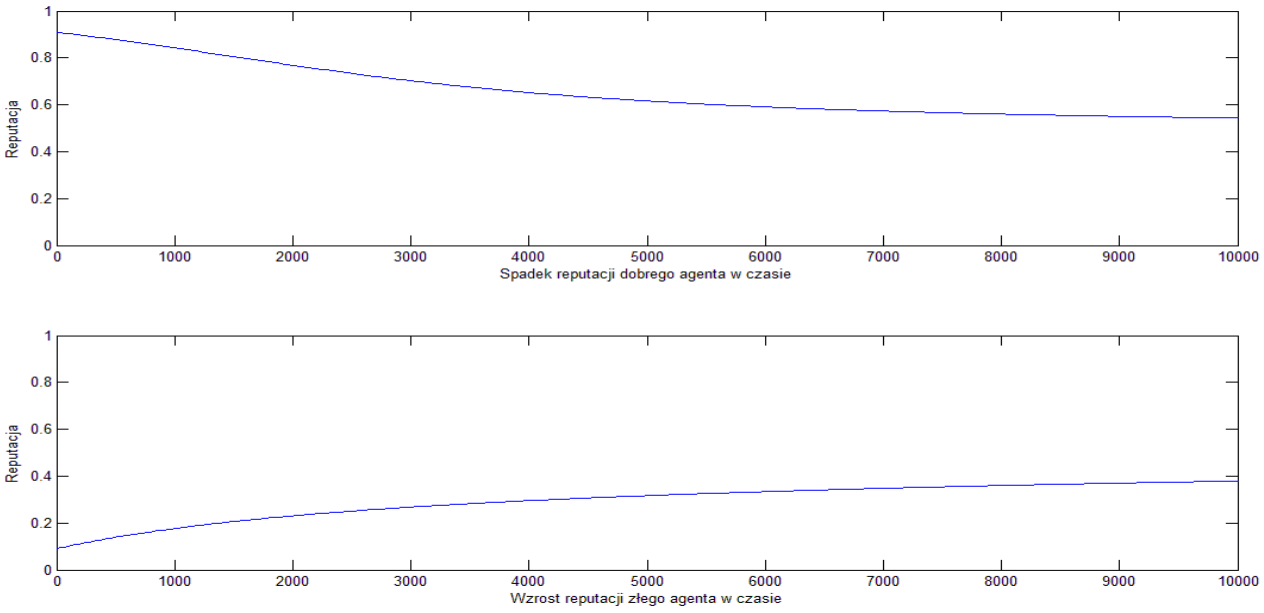
Dla każdego ocenianego obiektu $b \in B$, z zadanym okresem T_p reputacja jest wygaszana zgodnie z równaniami:

$$\alpha_{i+1}^{b} = (p * \alpha_{i}^{b} + c_i), \tag{6.36}$$

$$\beta_{i+1}^{b} = \left(p * \beta_{i}^{b} + c_i\right), \tag{6.37}$$

$$c_i = \begin{cases} \frac{\beta_i+1}{\alpha_i} & dla\ a_i > b_i \\ \frac{\alpha_i+1}{\beta_i} & wpp \end{cases}, \tag{6.38}$$

gdzie p jest współczynnikiem wygaszania z zakresu $p \in [0,1]$. Jego wartość jest bliska 1 i standardowo ustalona została na 0.99. Natomiast współczynnik c_i został tak skonstruowany, aby prędkość zmiany reputacji była proporcjonalna do zgromadzonych pozytywnych i negatywnych ocen. Jak widać z powyższego wzoru, wygaszaniu ulega reputacja każdego ocenianego obiektu (także tych z wysoką reputacją). Jest to zjawisko pożądane, gdyż w wypadku mało aktywnych agentów, a przez to nie posiadających od dłuższego czasu nowych ocen, prowadzi ono po dłuższym czasie także do uśrednienia ich reputacji. Natomiast w wypadku obiektów aktywnych, zjawisko to jest zaniedbywalne gdyż nowe oceny reputacyjne mają znacznie istotniejszy wpływ na ich ocenę końcową. Na rysunku 6.4. przedstawiony został przykład zmiany reputacji w czasie pod wpływem mechanizmu zapominania dla obiektów o reputacji wejściowej (50,5) oraz (10,100).



Rys. 6.4 Zmiana reputacji w czasie w wyniku działania mechanizmu wygaszania. Przykład dla reputacji początkowej (50,5) (na górze) oraz (10,100) (na dole).

d) Integracja mechanizmów

Podrozdział ten przedstawia kompletną charakterystykę systemu reputacyjnego, który został zaprojektowany z uwzględnieniem wymienionych powyżej mechanizmów i użyty do oceniania węzłów oraz linków w prezentowanym systemie wspomagającym transmisję czasu rzeczywistego. W systemie tym zbiór ocenianych obiektów dany jest jako suma zbiorów wszystkich agentów powiązanych z węzłami sieci, oznaczonych jako A, oraz zbiór wszystkich połączeń między nimi, oznaczony jako L. Do chwilowej oceny jakości łącza używana jest funkcja π_L, zwracająca liczbę z przedziału [0,1]. Oceny chwilowe danych obserwacji o_1 do o_k są sumowane, po czym przesyłane z interwałem czasowym T_s do serwera centralnego jako pary $\left(k\sum_{i=1}^{k}\tilde{g}_i, k\sum_{i=1}^{k}(1-\tilde{g}_i)\right)$, gdzie $\tilde{g}_i = \pi_L(o_i)$ jest wynikiem pojedynczej i-tej obserwacji, natomiast k liczbą cząstkowych ocen. Podstawą do oceny węzłów jest bezpośrednia interakcja między nimi – dany agent ocenia współuczestników transakcji przy pomocy funkcji π_A. Chwilowe oceny węzłów są ocenami binarnymi ze zbioru {G,B}; w wypadku gdy transakcja przebiega pozytywnie wystawiana jest ocena G. Natomiast jeżeli wystąpią problemy wynikłe z niewłaściwego zachowania uczestnika transakcji, takie jak przesyłanie błędnych danych, niewłaściwe trasy routingu, brak kooperacji itp., wystawiana jest ocena B. Ostatecznie oceny węzłów sieci dane są również jako pary (x, y), gdzie $x = \#G$ jest sumą transakcji ocenionych pozytywnie, natomiast $y = \#B$ sumą transakcji o negatywnym przebiegu.

Zgodnie z przyjętą w podrozdziale 6.3 charakterystyką systemu reputacyjnego jako zbioru parametrów $P = (\pi, Q, \Delta t, agr, rec, R_{rec}, r, R, r_0, r_{rec,0})$, na funkcję oceny π składają się funkcje składowe π_A oraz π_L. Zbiór Q będący zbiorem dopuszczalnych ocen wynikających z bezpośrednich obserwacji dany jest jako $\mathbb{R} \times \mathbb{R}$. Okres wyliczania aktualizacji oceny reputacyjnej Δt, jest parametrem zmiennym wynikającym z konkretnej implementacji systemu oraz kompromisu pomiędzy szybkością reakcji na zmiany w systemie, a ilością transmitowanych danych. W testowej wersji systemu przyjęto $\Delta t = 5$ s, przy czym jeśli aktualnie nadsyłane oceny wskazują na znaczny spadek reputacji danego obiektu (np. awaria łącza lub węzła), a jednocześnie reputacja rekomendacyjna nadsyłającego ocenę jest wysoka, wtedy centralny system reputacyjny rozsyła zaktualizowaną reputację natychmiastowo w celu minimalizacji ryzyka związanego z degradacją jakości transmisji.

Cząstkowe oceny nadsyłane w pojedynczym okresie obserwacji przez konkretnego agenta są agregowane do oceny ostatecznej przy pomocy funkcji agregującej agr. Funkcja ta ma postać sumy ograniczonej, tzn. poszczególne oceny są sumowane dla każdego elementu pary z osobna. Gdy suma któregoś z elementów przekroczy ustalony próg określający maksymalny wpływ oceny danego agenta na ocenę reputacyjną w pojedynczym okresie obserwacji, jest ona wtedy skalowana liniowo do dopuszczalnego zakresu. Idea ta może być wyrażona przy pomocy poniższego równania:

$$agr(i,j,Tn) = \begin{cases} \left(Z, \frac{yZ}{x}\right) gdy\ x > Z\ \wedge x \geq y \\ (x,y) = \sum_{t\in[\Delta tTn,\Delta t(Tn+1)]} ev(E^i_{j,t})\ dla\ x,y < Z \\ \left(\frac{xZ}{y}, Z\right) gdy\ y > Z\ \wedge y > x \end{cases} \quad (6.39)$$

Przyjęto ponadto następujące oznaczenia: funkcja $ev\left(q = E^i_{j,t}\right)$ zwraca parę (a,b) będącą oceną agenta j-tego przez agenta i-tego w chwili t, funkcje $ev(q)[1] = a$ oraz $ev(q)[2] = b$ odpowiednio pierwszy i drugi element pary, natomiast $\mathbb{E}\left(ev(q)\right)$ zwraca wartość oczekiwaną rozkładu beta o parametrach (a,b).

Funkcja wyliczająca reputację rekomendacyjną, po uwzględnieniu wzorów 6.28 oraz 6.31 przyjmuje postać:

$$rec\_\alpha^{i,j}_{Tn+1} = \mu_{rec} * rec_{\alpha}{}^{i,j}_{Tn} + m'(i,j,Tn), \quad (6.40)$$

$$rec_{\beta}{}^{i,j}_{Tn+1} = \mu_{rec} * rec_{\beta}{}^{i,j}_{Tn} + \left(1 - m'(i,j,Tn)\right), \quad (6.41)$$

$$Rrec^{i,j}_{Tn+1} = \frac{rec\_\alpha^{i,j}_{Tn+1}}{rec\_\alpha^{i,j}_{Tn+1} + rec\_\beta^{i,j}_{Tn+1}}, \quad (6.42)$$

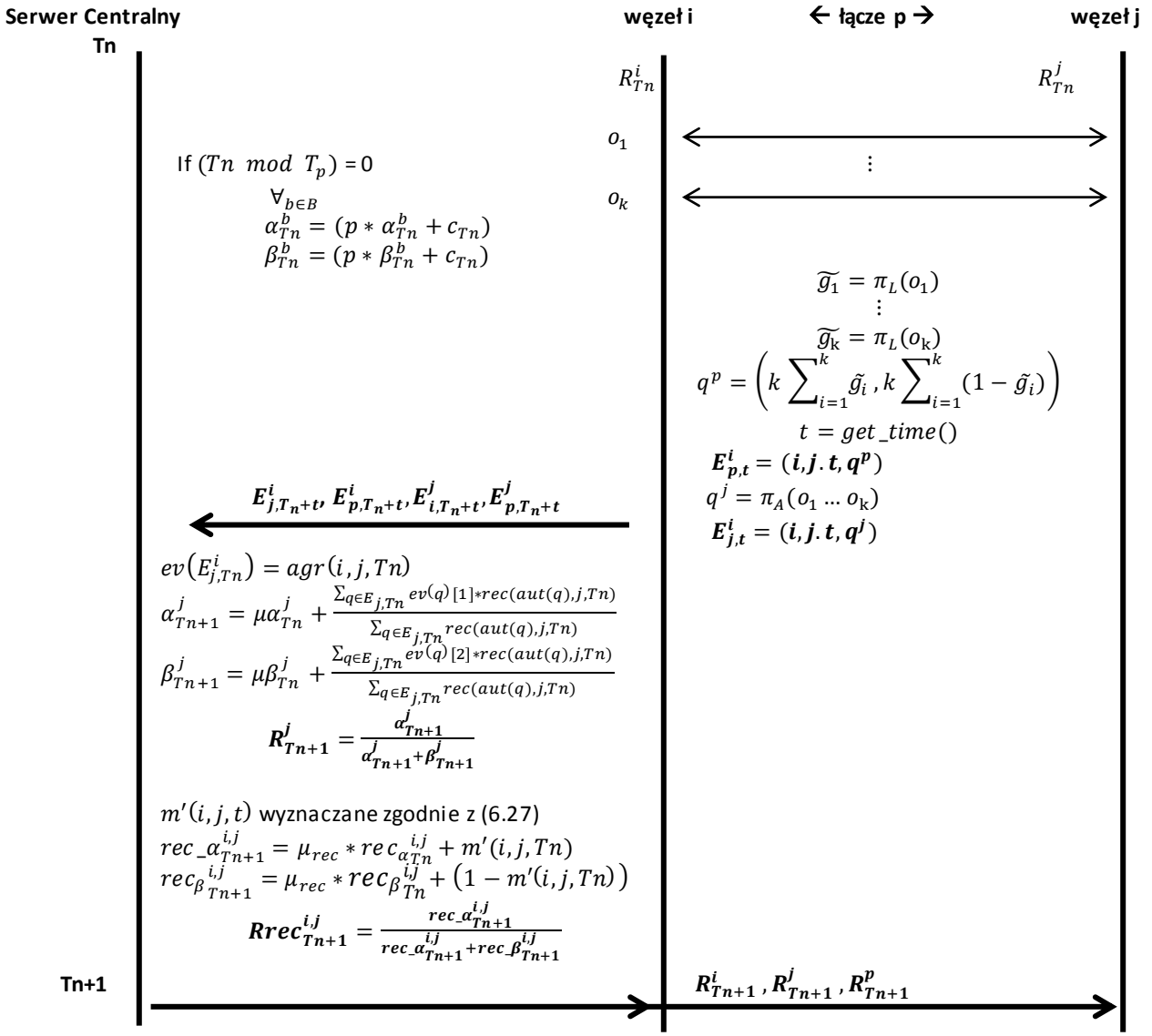
a zbiór R_{rec} możliwych ocen reputacji rekomendacyjnej określony jest na [0,1]. Ostatecznie reputację obiektu j, z uwzględnieniem współczynnika zapomnienia μ oraz reputacji rekomendacyjnej rec wyznacza się następująco:

$$\alpha^j_{Tn+1} = \mu\alpha^j_{Tn} + \frac{\sum_{q\in E_{j,Tn}} ev(q)[1] * rec(aut(q),j,Tn)}{\sum_{q\in E_{j,Tn}} rec(aut(q),j,Tn)}, \quad (6.43)$$

$$\beta^j_{Tn+1} = \mu\beta^j_{Tn} + \frac{\sum_{q\in E_{j,Tn}} ev(q)[2] * rec(aut(q),j,Tn)}{\sum_{q\in E_{j,Tn}} rec(aut(q),j,Tn)}, \quad (6.41)$$

$$R^{j}_{Tn+1} = \frac{\alpha^{j}_{Tn+1}}{\alpha^{j}_{Tn+1}+\beta^{j}_{Tn+1}}. \tag{6.42}$$

Z każdym interwałem $T_p = 100 * \Delta t$ reputacja jest także zmieniana w wyniku działania mechanizmu wygaszania opisanego w punkcie c). Jako wyjściową reputację przyjęto wartość 0.5 ($\mathbb{E}(beta(1,1))$), natomiast reputacja rekomendacyjna nowo przyłączanych węzłów została ustalona na poziomie 0.1. Diagram obrazujący proces wyznaczania reputacji przedstawiony jest na rysunku 6.5.



Rys. 6.5 Schemat protokołu wyznaczania reputacji łącza oraz węzła.

6.6 Bezpieczeństwo systemów reputacyjnych

Istnieje szereg ataków, których celem jest destabilizacja systemu reputacyjnego lub wpłynięcie na końcową ocenę reputacyjną konkretnego obiektu. W podrozdziale tym zostały scharakteryzowane najpowszechniejsze typy ataków oraz przedstawiony został sposób obrony przed nimi.

6.6.1 Ataki typu Sybil

Atak typu Sybil [42] to jeden z najpopularniejszych ataków na systemy reputacyjne, przeciw któremu wciąż nie opracowano uniwersalnego mechanizmu ochrony. Należy on do ataków na tożsamość w infrastrukturze, kiedy pojedyncza jednostka ukrywa swoją prawdziwą tożsamość i generuje wiele innych fikcyjnych. Większość systemów P2P zakłada, iż do określonej jednostki przypisana jest tylko jedna, konkretna tożsamość, dlatego też posiadanie kontroli nad większa liczbą tożsamości pozwala skuteczniej wpływać na pracę infrastruktury. Dobrym przykładem ilustrującym tę technikę jest głosowanie on-line, gdzie pojedyncza jednostka, posiadając wiele tożsamości, ma możliwość oddania wielu głosów. W wielu przypadkach atak taki jest w stanie zdyskredytować wyniki ocen systemu reputacyjnego.

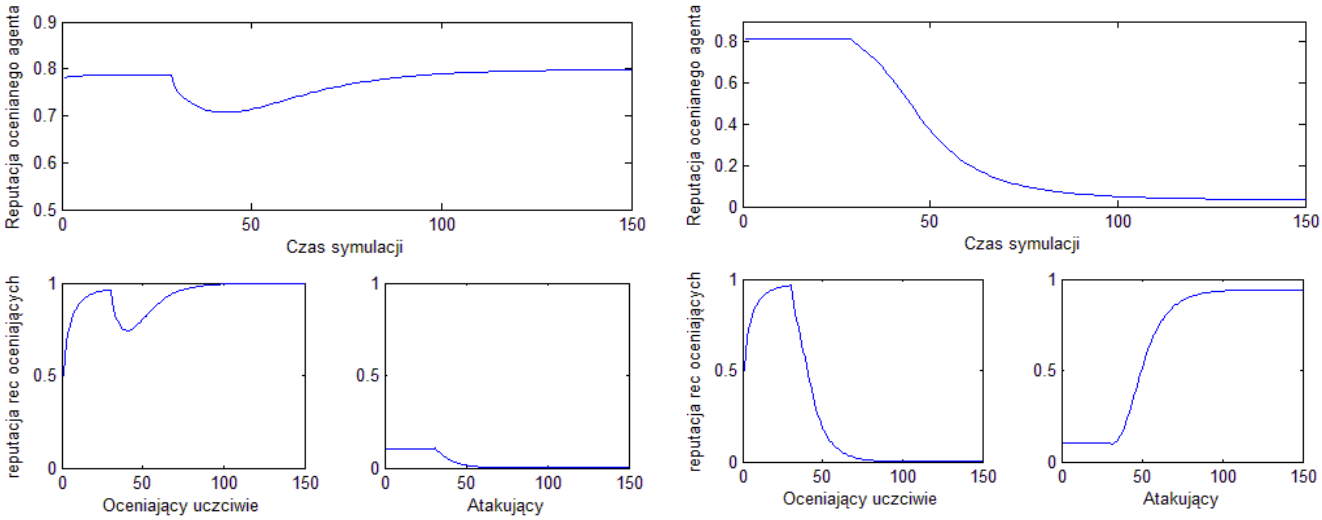
Ponieważ krytyczne dla poprawnego działania systemu reputacyjnego jest prawidłowe i bezpieczne przydzielanie tożsamości, wszelkie luki w mechanizmie jej przydzielania mogą znacznie ułatwić przeprowadzanie ataków (w szczególności typu Sybil). Dlatego też mechanizm przydzielania tożsamości musi spełniać kilka podstawowych warunków, takich jak niepowtarzalność, odrębność oraz trwałość w czasie.

Jednym z najczęściej stosowanych rozwiązań zapewniających bezpieczeństwo procesu przydzielania tożsamości oraz ochronę przed atakami z nią związanymi jest certyfikacja poprzez zaufaną stronę. Jednak mechanizm ten wymaga przeważnie fizycznego udziału osób trzecich, co w wypadku dużego systemu o możliwości ciągłego, dynamicznego przyłączania nowych węzłów jest rozwiązaniem nieefektywnym.

W niniejszym systemie użyty został kilkuetapowy mechanizm, który ma na celu zminimalizowanie zagrożenia spowodowanego atakiem typu Sybil. W pierwszym kroku procesu rejestracji i przydzielania tożsamości konieczne jest przejście testu Turinga (wpisanie kodu CAPTCHA) oraz potwierdzenie rejestracji z użyciem adresu e-mail. Kolejny etap wymaga

udowodnienia posiadania odpowiednich zasobów poprzez test przepustowości łącza oraz mocy obliczeniowej – jest to realizowane poprzez rozwiązanie zadanego problemu obliczeniowego w określonym czasie. Ostatecznie tożsamość powiązana zostaje z adresem IP nowego węzła, z możliwością jego późniejszej aktualizacji. Oprócz tego, w zależności od konkretnej implementacji systemu, istnieje możliwość dodania warunku związanego z kosztem uzyskania tożsamości (np. opłata finansowa).

Kolejnym czynnikiem minimalizującym zagrożenie jest reputacja rekomendacyjna, która dla nowo przyłączonych węzłów jest ustalana na niskim poziomie. Dzięki temu nawet w wypadku uzyskania znacznej liczby tożsamości przez jednego agenta, ich skumulowany wpływ na infrastrukturę P2P będzie znikomy. Na rysunku 6.6 przedstawiony został rezultat symulacji obrazującej przykładową próbę manipulacji oceną reputacyjną przy pomocy nowo zarejestrowanych agentów o fałszywej tożsamości.



Rys. 6.6 Symulacja nieudanego (po lewej) oraz udanego (po prawej) ataku typu Sybil mającego na celu zaniżenie reputacji atakowanego.

Początkowa reputacja atakowanego węzła wynosiła 0.8, a ocena końcowa generowana była na podstawie średniej z ośmiu ocen od różnych agentów. W czasie t=30 do grona oceniających dołączyło 12 nowo zarejestrowanych agentów próbujących zaniżyć reputację, lecz atak zakończył się niepowodzeniem. W kolejnej próbie ataku użyto 30 nieuczciwych agentów, co stanowiło 79% wszystkich oceniających; w tym wypadku atak odniósł oczekiwany skutek – reputacja atakowanego agenta została zaniżona. Na podstawie przeprowadzonych symulacji określono iż opracowany system reputacyjny jest w stanie efektywnie bronić się przed atakami

typu Sybil, gdy liczba atakujących nie przekracza 75% wszystkich oceniających. W wypadku gdy jest ona wyższa, a celem ataków jest większa grupa, do ochrony może zostać wykorzystany mechanizm wykrywania koalicji opisany szerzej w podrozdziale 6.7

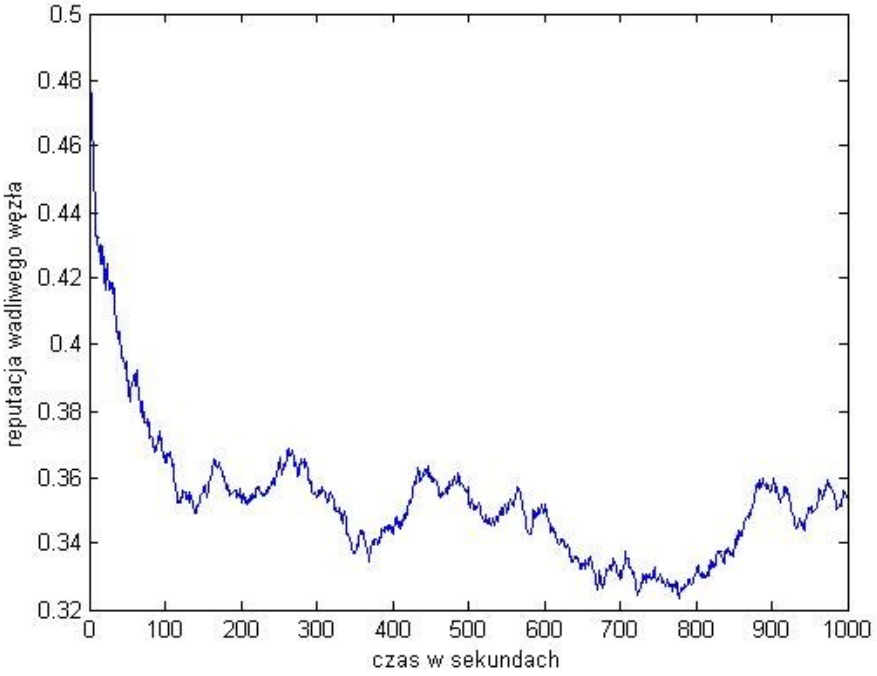
6.6.2 Wadliwe węzły

Węzły wadliwe można podzielić na dwie kategorie: węzły ulegające typowym awariom oraz kłamcy.

Typowe awarie węzłów

Tego typu awarie są stosunkowo łatwe do wykrycia. W wypadku gdy oceniający agent nie może się połączyć z danym węzłem, obniżana jest wyłącznie reputacja łącza pomiędzy nimi, natomiast reputacja samego węzła pozostaje bez zmian. Natomiast gdy zostanie wykryte błędne zachowanie samego węzła, np. przesyłanie niepoprawnych pakietów, niski standard bezpieczeństwa lub korzystanie z błędnej trasy routingu, reputacja węzła ulegnie zaniżeniu i w efekcie zostanie on odizolowany od reszty systemu. Dzięki mechanizmowi zapominania po pewnym czasie jego ocena wzrośnie i będzie miał on ponownie szansę podłączyć się do infrastruktury.

Poniższa symulacja (Rys. 6.7) przedstawia czas wykrycia i odizolowania wadliwych węzłów w sieci. Przyjęto, iż każdy wadliwy węzeł w danej chwili połączony jest z losową liczbą (z zakresu [1-5]) węzłów funkcjonujących prawidłowo, oraz że prawdopodobieństwo przesłania błędnych danych w ciągu jednej interakcji wynosi 70%. Jako wartość współczynnika zapominania przyjęto $\mu = 0.99$. Symulacja pokazuje iż czas detekcji jest stosunkowo szybki, wadliwy węzeł osiągnął reputację niższą niż 0.4 już po 40 sekundach.



Rys. 6.7 Spadek reputacji wadliwego węzła w czasie.

Kłamcy

Odrębną grupą są agenci przesyłający nieprawidłowe dane w celu zrealizowania określonego zadania, jakim może być np. przechwycenie transmisji lub zmiana reputacji innych agentów. Aby agent X mógł przechwycić transmisję pomiędzy agentami A-B, trasa przebiegająca przez niego musi uzyskać lepszą ocenę od dotychczasowej trasy routingu pomiędzy A-B. W tym celu musi on zadeklarować wysoką jakość połączenia między wybranymi węzłami A' B', takimi że trasa T'=(A-…-A'-X-B'-…-B) uzyska najwyższą ocenę. Jednakże jeśli X w rzeczywistości nie dysponuje wydajnymi łączami pomiędzy A' oraz B', fakt ten zostanie wykryty przez sąsiadujących z nim agentów, co doprowadzi do zaniżenia reputacji łącza deklarowanego przez X jako wydajne, a zatem ostatecznie trasa między A-B nie ulegnie zmianie na T'.

Jeżeli natomiast X chce wpłynąć na reputację innego agenta (Y), może przyjąć kilka możliwych strategii postępowania. W najprostszym przypadku zakłada się, iż X próbując w możliwie najkrótszym czasie zmienić reputację Y, będzie zawsze przesyłał maksymalnie pozytywne lub maksymalnie negatywne opinie na jego temat, w zależności od zamierzonego celu. W takiej sytuacji możliwe są dwa scenariusze:

- X jest jedynym oceniającym Y; wówczas reputacja Y będzie w każdym kroku aktualizowana zgodnie z opinią X, lecz z uwzględnieniem ograniczenia na maksymalną dozwoloną zmianę w pojedynczym okresie. Natomiast reputacja rekomendacyjna X na temat Y, zgodnie z przyjętym limitem na minimalną liczbę oceniających, nie będzie ulegać zwiększeniu. Gdy w kolejnych okresach obserwacji pojawią się inne oceny na temat Y, wystawione przez uczciwych agentów, zacznie działać mechanizm reputacji rekomendacyjnej i ocena wystawiona Y zostanie skorygowana.
- X jest jednym z wielu oceniających Y, a ponieważ początkowa reputacja rekomendacyjna X jest niska, będzie on miał znikomy wpływ na ocenę końcową agenta Y.

W bardziej rozbudowanych strategiach kłamca może próbować najpierw zyskać zaufanie (zwiększyć swoją reputację rekomendacyjną) przesyłając przez pewien czas poprawne opinie, a dopiero w następnym kroku zacząć kłamać. Skuteczność tego typu ataków jest szczególnie wysoka, gdy reputacja rekomendacyjna zdefiniowana jest jako pojedyncza ocena dla każdego z agentów (analogicznie do zwykłej reputacji). W projekcie tym reputacja rekomendacyjna wyznaczana jest jednak niezależnie dla każdej pary agentów, więc fakt, iż dany agent ocenia poprawnie $a_{1..n}$ innych agentów, nie ma żadnego wpływu na jego reputację rekomendaycjną względem agenta a_{n+1}. Zatem jedynym sposobem na jej zwiększenie jest przesyłanie przez dłuższy czas prawidłowych ocen na temat a_{n+1}. Skutkiem tego będzie jednak dodatkowe ugruntowanie opinii odnośnie a_{n+1} (dodatkowe poprawne oceny zwiększą parametry α oraz β rozkładu beta), przez co późniejsza próba zmiany reputacji a_{n+1} będzie wymagała większej ilości opinii/czasu. Dodatkowo jeżeli w tym okresie będą pojawiać się także uczciwe opinie odnośnie a_{n+1}, reputacja rekomendacyjna atakującego agenta znowu zostanie zmniejszona.

W powyższych analizach przyjęto, że większość wszystkich przesyłanych ocen jest prawidłowa, oraz że każdy agent otrzymuje najczęściej takie właśnie oceny. Pierwszy warunek jest wymagany dla prawidłowego funkcjonowania infrastruktury, drugi natomiast nie zawsze może być spełniony. Wynika to z charakteru systemu, w którym mogą istnieć agenci rzadko biorący udział w transmisji real-time np. ze względu na niepopularną lokalizację lub relatywnie słabe łącza. W ich wypadku możliwe staje się przeprowadzenie udanego ataku skutkującego ich czasowym zablokowaniem, szczególnie gdy kłamcy zaczną współpracować i wspólnie przesyłać nieprawdziwe opinie. Jednakże uznano, iż nie jest to element krytyczny dla działania infrastruktury – wykluczeni w wyniku ataku agenci, dzięki mechanizmowi wygaszania reputacji

zostaną po pewnym czasie odblokowani, a ponieważ z definicji byli mało aktywni, istnieje niewielkie prawdopodobieństwo, że ich chwilowa blokada znaczne obniży wydajność całej infrastruktury.

Osobną klasę zagrożeń stanowią ataki przeprowadzane z udziałem skoordynowanej grupy agentów tworzących koalicje. Ponieważ ataki tego typu są szczególnie groźne i mogą doprowadzić do paraliżu całej infrastruktury, a ich wykrycie jest problemem złożonym, zagadnieniu temu został poświęcony kolejny podrozdział.

6.7 Mechanizm wykrywania złośliwych koalicji

6.7.1 Złośliwe koalicje

Jak zostało pokazane w poprzednim podrozdziale, zaprojektowany system reputacyjny oparty o rozkład beta potrafi skutecznie wykrywać agentów charakteryzujących się niewłaściwym działaniem i poprzez przypisanie im niskiej reputacji minimalizować ich rolę w systemie. Ponadto zastosowanie mechanizmu reputacji rekomendacyjnej pozwala na wykrycie agentów, których działanie ma na celu destabilizację systemu lub osiągnięcie własnych korzyści poprzez przesyłanie nieprawdziwych opinii. W takiej sytuacji oceny przesłane przez tych agentów zostaną częściowo lub całkowicie zignorowane w procesie obliczania reputacji. Jednak niezwykle istotny problem, który nie został dotychczas omówiony, związany jest z działaniami koalicji złośliwych agentów. Przez koalicję rozumie się tu grupę agentów, którzy są świadomi wzajemnego istnienia, posiadają zdolność wzajemnej komunikacji i koordynują swoje działania w celu osiągnięcia zamierzonego efektu (najczęściej destabilizacji systemu lub wzmocnienia własnej w nim pozycji). Należy zwrócić uwagę, że jednym z podstawowych założeń przyjętych przy projektowaniu systemu reputacyjnego jest założenie, iż większość agentów uczciwie ocenia innych. Łatwo jednak zauważyć, że w przypadku sieci, której schemat wzajemnych interakcji między agentami da się przedstawić w formie grafu rzadkiego, już niewielka grupa współdziałających agentów może istotnie wpłynąć na pracę całego systemu. Sytuacja taka może także zaistnieć w systemie będącym przedmiotem niniejszej pracy, gdzie agenci, odzwierciedlający węzły sieci P2P, komunikują się w większości sytuacji tylko z niewielką liczbą innych agentów (grupa o lokalnie najlepszej jakości łącz). W związku z tym grupa

oceniająca danego agenta również nie jest liczna, a więc czasem skoordynowana koalicja nawet niewielu agentów może być w stanie dowolnie zmienić oceny reputacyjne pozostałych.

Problem wykrywania koalicji jest bardzo złożony i w wielu wypadkach wciąż otwarty. W pracy [38] zostało pokazane, że system reputacyjny oparty o mechanizm eigentrust radzi sobie dobrze z tym zagadnieniem (zadowalające wyniki można uzyskać nawet przy 70% złośliwych agentów). Jednak ze względu na wymóg istnienia predefiniowanych zaufanych węzłów mechanizm ten został odrzucony jako kandydat do użycia w systemie opracowanym w ramach niniejszej rozprawy. Z kolei zastosowany ze względu na swoją uniwersalność system reputacyjny beta nie jest niestety odporny na działanie złośliwych koalicji. Reputacja rekomendacyjna wyznaczana jest w nim na podstawie odchylenia od średniej z nadsyłanych ocen. Gdy w wyniku ataku średnia osiągnie wartość istotnie różniącą się od uczciwej oceny, jaką powinien otrzymać dany agent, nastąpi odwrócenie ocen reputacji rekomendacyjnej (osiągnie ona niską wartość dla agentów raportujących prawdziwe oceny, wzrośnie natomiast dla członków koalicji).

Ze względu na istotność zagrożeń związanych z istnieniem koalicji, w ostatnim czasie zaproponowany został szereg metod mających na celu ich identyfikację. Podejścia zaproponowane w pracy [43] uwzględniają próby rozwiązania tego problemu na gruncie teorii gier, ale jednocześnie wymagają przyjęcia założeń co do zdolności graczy oraz oczekiwanej nagrody, co jest trudne do oszacowania w wypadku węzłów sieci P2P. Inne podejście, przedstawione w pracy [44], bazuje na metodzie rozpoznawania wzorców, gdzie zachowanie węzłów jest obserwowane w celu wykrycia podejrzanej, szkodliwej działalności. Jednakże metoda ta jest skuteczna jedynie w przypadku, gdy złośliwe koalicje zachowują się zgodnie z wcześniej znanym planem. Z kolei w opracowaniu [45] autorzy wprowadzają pojęcie „przestrzeni korzyści", definiowanej jako wielowymiarowa przestrzeń odzwierciedlająca poziom korzyści (ocen) wygenerowanych względem każdego agenta w systemie. Wychodząc z założenia, iż członkowie koalicji mają podobne preferencje, a więc oceniają innych w podobny sposób, koalicje takie identyfikowane są poprzez szukanie agentów położonych najbliżej w przestrzeni korzyści. Podejście to nie wymaga żadnej dodatkowej wiedzy na temat modelu zachowania koalicji lub zdolności agentów i może być używane zarówno do wykrywania szkodliwych koalicji, jak i koalicji ustanowionych w celu sztucznego zawyżania wzajemnej oceny. Jednakże, ze względu na założenie mówiące o podobnym zachowaniu członków koalicji, metoda ta może nie być efektywna w wypadku koalicji bardziej inteligentnych, maskujących

swoje działanie. W ich wypadku można założyć, iż agenci w celu ukrycia swojej przynależności do koalicji będą dysponowali pewną autonomią, pozwalającą im na swobodny wybór partnerów do komunikacji, a przez to odgrywać będą rolę niezależnych oraz godnych zaufania jednostek. W ich wypadku dyscyplina koalicyjna objawiać się będzie jedynie w przesyłaniu ocen na temat wybranych agentów, będących celem ataku. Stworzona w ten sposób koalicja jest znacznie trudniejsza do wykrycia, ponieważ przestrzenie korzyści poszczególnych jej członków mogą się mocno różnić między sobą.

Ponieważ każda z omówionych dotychczas metod cechuje się pewnymi ograniczeniami, w celu skutecznego przeciwdziałania zagrożeniom płynącym z działalności koalicji, autor niniejszej pracy zdecydował się opracować własny algorytm mający za zadanie wykrycie i zniwelowanie skutków działania koalicji [46]. Idea algorytmu wywodzi się z założenia, iż podstawową cechą charakterystyczną koalicji jest współdziałanie grupy agentów mające na celu zwiększenie ich własnej pozycji w systemie poprzez grupowe wysyłanie ocen nieprawdziwych: wzajemne wystawianie zawyżonych ocen oraz/lub zaniżanie ocen innych agentów.

W związku z powyższym, w pierwszym etapie wykonania algorytmu potencjalni członkowie koalicji identyfikowani są w wyniku obserwacji agentów, którzy są kontrowersyjnie oceniani. Oceniający każdego z takich agentów zostają podzieleni na dwie grupy: wysyłających wysokie oraz niskie oceny. Następnie przy pomocy algorytmu klasteryzacji wyznaczane są potencjalne grupy agentów, którzy często współdziałają przy wysyłaniu kontrowersyjnych ocen. Aby uniknąć sytuacji, gdy zaniżone lub zawyżone oceny wynikają jedynie z przypadku lub preferencji pewnej grupy agentów, w kolejnym kroku przeprowadza się ponowną klasteryzację na uprzednio zidentyfikowanych grupach. Na tym etapie, bazując na założeniu, iż zwyczajowo członkowie koalicji wzajemnie oceniają się wysoko, jako kryterium klasteryzacji przyjęto wartość wzajemnych ocen przesyłanych do systemu reputacyjnego. Grupy wyznaczone w wyniku powyższego procesu mogą być albo grupami agentów, którzy działają poprawnie, wzajemnie oceniają się wysoko i jednocześnie uczciwie oceniają innych agentów lub też mogą być złośliwymi koalicjami sztucznie zaniżającymi lub zawyżającymi oceny innych. Aby to zweryfikować, opracowano miarę podobieństwa między stronami oceniającymi przeciwnie przypadki kontrowersyjne.

Jeśli algorytm detekcji zidentyfikuje daną grupę agentów jako złośliwą koalicję, reputacja rekomendacyjna jej członków względem kontrowersyjnie ocenianych węzłów zostaje obniżona

do 0. W bardziej restrykcyjnej konfiguracji systemu reputacyjnego możliwe jest także obniżenie reputacji członków danej koalicji, ignorowanie nadsyłanych przez nich ocen wzajemnych (reputacja rekomendacyjna równa stale 0) lub obniżenie ich całościowej (względem wszystkich innych) reputacji rekomendacyjnej. W dalszej części tego rozdziału zostało pokazane, że metoda ta pozwala wykryć nie tylko koalicje złożone z agentów o bardzo zbliżonym profilu oceniania, lecz także te, w których agenci posiadają swobodę wyboru partnerów, z którymi się komunikują i oceniają, a koalicyjna dyscyplina w przesyłaniu ocen obowiązuje wyłącznie względem atakowanej grupy.

6.7.2 Schemat działania algorytmu detekcji

Poniżej zamieszczony jest szczegółowy opis algorytmu służącego do wykrywania koalicji w systemach reputacyjnych beta z uwzględnieniem poszczególnych etapów jego pracy.

1) Analiza kontrowersyjnie ocenianych agentów.

Mechanizm wykrywania koalicji uruchamiany jest cyklicznie z interwałem czasowym K_{kd}. Interwał ten został ustalony tak, aby nie obciążać nadmiernie serwera, a jednocześnie zdążyć zebrać wymaganą ilość danych do analizy. Przyjęte zostało, iż czas ten zależny będzie liniowo od współczynnika zapominania μ używanego przez system reputacyjny. To założenie wynika z następującego faktu: jeżeli koalicja chce, aby zmienione w wyniku jej działania oceny reputacyjne pozostały aktualne, musi przesyłać fałszywe opinie nie rzadziej niż co K_{kd}, wtedy bowiem zostaną one każdorazowo zarejestrowane przez mechanizm detekcji. Przy każdym uruchomieniu algorytmu detekcji przeglądane są oceny zarejestrowane w czasie K_{kd} i dla każdego agenta $j \in A$ wyznaczany jest współczynnik odchylenia ocen zgodnie ze wzorem:

$$\sigma_j = \frac{1}{\#n} \sum_{i \epsilon n} |rep_i^j - avg_j|, \tag{6.43}$$

gdzie n jest zbiorem wszystkich agentów przesyłających oceny na temat agenta j-tego, natomiast avg_j jest średnią oceną wyliczoną na podstawie nadesłanych ocen. Jeśli σ_j przekroczy ustalony próg σ_{st}, agent j-ty zostaje zaklasyfikowany jako kontrowersyjnie oceniany, a agenci przesyłający pozytywne i negatywne oceny na jego temat zostają dodani odpowiednio do zbiorów L_{poz}^j oraz L_{neg}^j. Mimo iż przy wyznaczaniu σ_j reputacja rekomendacyjna nie jest brana

pod uwagę, w obliczeniach przyjęte zostało, że pomimo to pomijane są oceny nadesłane przez agentów oznaczonych jako kłamcy, tzn. o bardzo niskiej średniej reputacji rekomendacyjnej (poniżej 0.1).

2) Budowa macierzy binarnych oraz macierzy uśrednionej opisującej współdziałanie agentów.

W przypadku najprostszych koalicji można założyć, że wszyscy jej członkowie przesyłają w trakcie ataku nieprawdziwe oceny. Jednak jeśli koalicja dysponuje większą liczbą członków, możliwe jest, że będzie ona próbowała w jakiś sposób zamaskować swoją działalność. Przykładowo, w przypadku niewielkiego systemu agentowego o średnim współczynniku interakcji między agentami równym 4, wystarczy koalicja złożona z 4 agentów, aby móc istotnie zmieniać oceny reputacyjne. Jednak jeśli koalicja ma 5 członków, to chcąc jednocześnie wpływać na oceny czterech wybranych agentów $\{a_1, a_2, a_3, a_4\}$, zamiast wysyłać cztery opinie jako grupa {1,2,3,4,5}, może próbować pozornie maskować swoje działanie, np. wysyłając oceny według schematu:

(a_1) {1 2 3 4}
(a_2) {5 2 3 4}
(a_3) {1 5 3 4}
(a_4) {1 2 5 4}

Jak widać w takim wypadku żaden z członków koalicji nie występuje w parze z żadnym innym jednocześnie we wszystkich czterech zestawieniach. Aby system detekcji koalicji zdolny był wykrywać także tego typu maskowania, postanowiono utworzyć macierz częstości współdziałania poszczególnych agentów. Początkowo, dla każdego kontrowersyjnie ocenianego agenta $k \in n$, na podstawie zbiorów L^k_{poz} oraz L^k_{neg}, tworzona jest binarna macierz M_k reprezentująca współdziałanie agentów oceniających k:

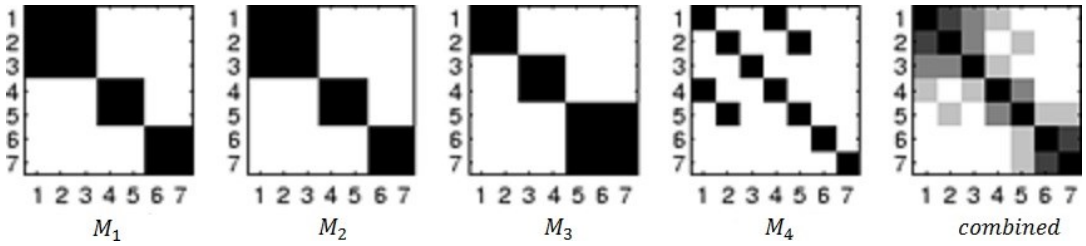
$$\begin{cases} M_k(i,j) = 1 & gdy\ i,j \in L^k_{neg} \vee i,j \in L^k_{poz} \\ M_k(i,j) = 0 & wpw \end{cases}. \tag{6.44}$$

Tak utworzone macierze $M_{i\in[1\ldots n]}$ są następnie agregowane do ogólnej macierzy M, opisującej częstość współdziałania dowolnych dwóch agentów przy wysyłaniu ocen na temat agentów uznanych za kontrowersyjnych:

$$M(i,j) = \frac{1}{\#i + \#j} \sum_{k=1}^{N} 2 * M_k(i,j), \tag{6.45}$$

gdzie #i oznacza liczbę zbiorów $L^k_{poz} + L^k_{neg}, k \in n$, w których występuje agent i-ty. Skonstruowana w ten sposób macierz jest symetryczna, w związku z tym można się ograniczyć tylko do wyznaczenia macierzy górnotrójkątnej.

Poglądowy schemat tworzenia uogólnionej macierzy M został przedstawiony na rysunku 6.8



Rys. 6.8 Schemat działania algorytmu klasteryzacji.

3) Klasteryzacja na podstawie macierzy częstości współpracy.

Potencjalne koalicje wyznaczane są w procesie klasteryzacji (użyto w tym celu algorytmu Similarity Partitioning Algorithm (CSP) [47]):

- W pierwszym etapie każdy z agentów $i \in n$ tworzy osobny jednoelementowy zbiór Z_i.
- Tworzona jest macierz częstości współpracy $\mathcal{M}$ pomiędzy zbiorami agentów, która w pierwszym etapie klasteryzacji (dla zbiorów jednoelementowych) przyjmuje postać $\mathcal{M}^{t_0} = M$.
- Dla każdego ze zbiorów wyznaczany jest współczynnik podobieństwa określający jak często agenci z danych dwóch zbiorów współpracowali ze sobą:

$$\rho(Z_a, Z_b) = \frac{\sum_{i \epsilon Z_a} \sum_{j \epsilon Z_b} M(i,j)}{\#Z_a \#Z_b}. \tag{6.46}$$

- Dwa zbiory o najwyższym wzajemnym współczynniku ρ zostają agregowane do jednego tworząc macierz $\mathcal{M}^{t+1}$ o jednym zredukowanym wierszu i kolumnie.
- Analogicznie wyznaczany jest współczynnik podobieństwa między zbiorami, już z uwzględnieniem nowopowstałego zbioru. Tak jak poprzednio, dwa zbiory o najwyższym współczynniku zostają agregowane w jeden zbiór.
- Scalanie zbiorów według powyższego schematu zostaje przerwane, gdy najwyższy współczynnik podobieństwa między zbiorami będzie mniejszy niż ustalony próg ε_S.

4) Ponowna klasteryzacja wyznaczonych grup agentów.

Na tym etapie określane są relacje między agentami podobnie oceniającymi innych (kontrowersyjnych) agentów. W tym celu wyznaczone w kroku 3) grupy agentów przechodzą ponowną klasteryzację (każda grupa niezależnie). Zakłada się, iż agenci wewnątrz koalicji nie oceniają się negatywnie, tak więc algorytm klasteryzacji ma na celu wybranie podgrupy agentów, których wzajemne oceny są wysokie a przez to zidentyfikowanie tej koalicji. Przebieg algorytmu klasteryzacji jest analogiczny do opisanego w punkcie 3), przy czym jako funkcję stopnia podobieństwa zbiorów przyjmuje się:

$$\rho_2(Z_a, Z_b) = \frac{\sum_{i \epsilon Z_a} \sum_{j \epsilon Z_b} rep(i,j)}{\#Z_a \#Z_b}. \tag{6.47}$$

Zatrzymanie procesu klasteryzacji następuje, gdy maksymalny stopień podobieństwa pomiędzy dostępnymi zbiorami jest niższy niż ustalony próg ε_{GS}. Aby odróżnić przypadkowy zbiór agentów od agentów tworzących koalicję, próg ten powinien być dobrany względnie wysoko. Dokładna wartość progu może zależeć od przewidywań nt. sposobu współpracy agentów. Dla prostych form koalicji można przyjąć $\varepsilon_{GS} = 1$, jednak zakładając iż koalicja w swoim zachowaniu może używać pewnej strategii w celu zminimalizowania szansy jej wykrycia, warto lekko obniżyć ε_{GS} (np. do 0.8).

5) Klasyfikacja koalicji jako szkodliwych.

Odróżnienie grupy przypadkowych węzłów o podobnych preferencjach od rzeczywiście szkodliwej koalicji bywa często nietrywialnym zadaniem. Czasem (jak np. w [45]) klasyfikacja oparta jest na założeniu, iż agenci wewnątrz koalicji oceniają się wzajemnie wyżej niż losowo wybrana próbka agentów spośród wszystkich dostępnych. Jednak podejście to nie zawsze musi być prawdziwe, a dodatkowo może doprowadzić do niesłusznego zaklasyfikowania grupy dobrych (a więc wzajemnie wysoko się oceniających węzłów) jako koalicji. Aby uniknąć tego typu problemów i zwiększyć szansę na poprawną klasyfikację w niniejszym mechanizmie postanowiono zastosować autorską metodę wykrywania koalicji. Polega ona na porównywaniu podobieństwa zbiorów agentów oceniających przypadki kontrowersyjne podobnie jak członkowie potencjalnej koalicji, do zbiorów agentów oceniających te same przypadki przeciwnie. Warto zauważyć, iż koalicja dysponuje ograniczonymi zasobami, więc aby zmienić oceny reputacyjne wybranej grupy agentów, przy wysyłaniu sfałszowanych ocen na ich temat musi współpracować zdecydowana większość członków koalicji. Z drugiej strony oceny przeciwne do wystawionych przez członków koalicji podają w większości uczciwi agenci, oceniający węzły kontrowersyjne zgodnie z prawdą. Tacy agenci mają własne preferencje, każdy z nich może oceniać wszystkich bądź tylko wybraną grupę kontrowersyjnych węzłów. Można więc wyciągnąć wniosek, że poziom współpracy agentów tworzących koalicję przy ocenie agentów kontrowersyjnych jest zazwyczaj wyższy (wyłączając sytuacje, gdy koalicja dysponuje wieloma członkami i stosuje zaawansowane metody maskowania) niż poziom współpracy agentów oceniających przypadki kontrowersyjne przeciwnie.

Etapy działania algorytmu używanego do opisanej klasyfikacji koalicji przedstawione są poniżej:

a) Dla każdej z grup $A \in PK$ agentów określonych w punkcie 3) jako potencjalna koalicja, wyznaczana jest podgrupa węzłów kontrowersyjnych $K_A \in K$, przy ocenie których uczestniczyła większość członków grupy A, a ich oceny były pozytywne:

$$K_A^{poz} = \{i \in K | \exists_{A' \subseteq A} : \forall_{x \in A'} x \in L_{poz}^i \wedge \#A' > \sigma_{k\_min} \#A\}, \quad (6.48)$$

lub negatywne:

$$K_A^{neg} = \{i \in K | \exists_{A' \subseteq A} : \forall_{x \in A'} x \in L_{neg}^i \wedge \#A' > \sigma_{k\_min} \#A\}, \quad (6.49)$$

σ_{k_min} oznacza minimalną liczbę agentów ze zbioru A, którzy wspólnie wysyłali oceny na temat konkretnego agenta ocenionego kontrowersyjnie. W prezentowanym modelu przyjęto $\sigma_{k\_min} = 0.7$.

b) Wyznaczona zostaje grupa agentów, którzy w wypadku kontrowersyjnych K_A oceniali analogicznie do potencjalnej koalicji A (negatywnie bądź pozytywnie):

$$Collab'_A = \left\{ \bigcup_{i \in K_A^{poz}} L^i_{poz} \cup \bigcup_{i \in K_A^{neg}} L^i_{neg} \right\}, \tag{6.50}$$

oraz zbiór agentów oceniających przeciwnie do A:

$$Oppon_A{}' = \left\{ \bigcup_{i \in K_A^{poz}} L^i_{neg} \cup \bigcup_{i \in K_A^{neg}} L^i_{poz} \right\}. \tag{6.51}$$

Następnie w celu minimalizacji błędów spowodowanych istnieniem agentów sporadycznie przesyłających nieprawdziwe oceny, usuwani są agenci występujący jednocześnie w obu podzbiorach $A_s = Collab_A{}' \cap Oppon_A{}'$. W ten sposób utworzone zostają ostatecznie zbiory: $Collab_A = Collab_A{}'/A_s$ oraz $Oppon_A = Oppon_A{}'/A_s$.

c) Budowana jest macierz M^{K_A} opisująca częstość współdziałania każdej pary agentów przy wysyłaniu kontrowersyjnych ocen na temat K_A. Proces budowy macierzy jest analogiczny do przestawionego w punkcie 2) przy czym ogranicza się tylko do zbioru K_A kontrowersyjnie ocenianych agentów.

d) Wyznaczanie stopnia współpracy pomiędzy członkami zbioru $Collab_A$ oraz $Oppon_A$. W tym celu używana jest miara *m* bazująca na macierzy M^{K_A}

$$m(X) = \frac{2 \sum_{i \in X} \sum_{j \in X, j > i} M^{K_A}(i,j)}{(\#m)(\#m-1)}. \tag{6.52}$$

e) Klasyfikacja potencjalnej koalicji *A*.

Przyjmuje się, że potencjalna koalicja *A* zostaje zaklasyfikowana jako rzeczywista, gdy $m(Collab_A) > m(Oppon_A)$. Dla minimalizacji ryzyka niewłaściwej klasyfikacji postanowiono ponadto przyjąć dodatkowy limit, który musi zostać spełniony: $m(Collab_A) > \sigma_{coal}$. Próg ten pozwala sterować czułością detekcji w przypadku, gdy w systemie występuje znaczna ilość agentów losowo oceniających błędnie oraz gdy koalicja stosuje maskowanie. Standardowo przyjęto iż $\sigma_{coal} = 0.55$.

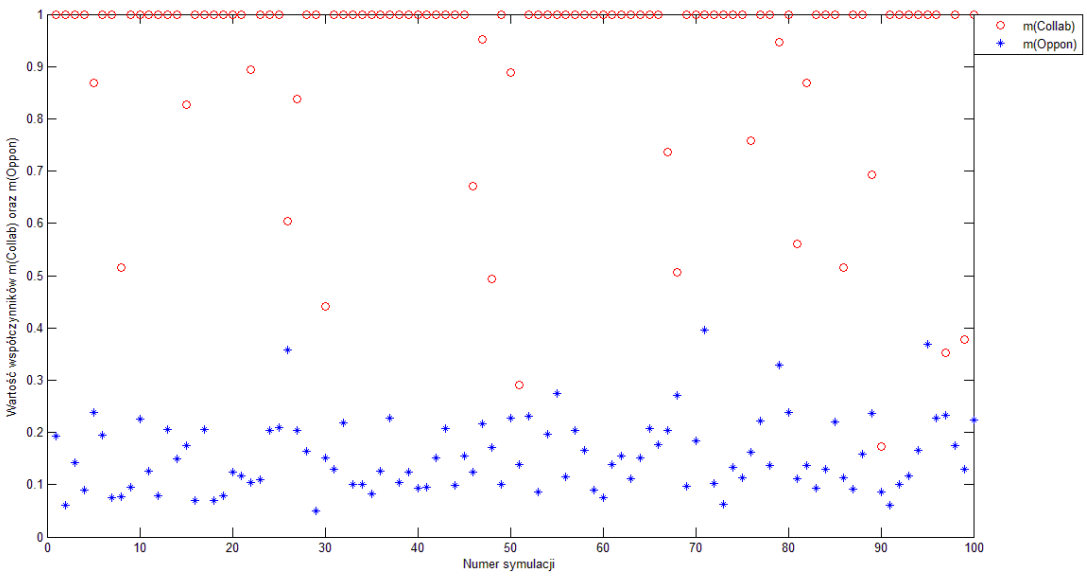
6.7.3 Symulacja działania

Aby przetestować poprawność działania algorytmu, został przeprowadzony szereg symulacji. W tym celu stworzono grupę testową składająca się ze 100 agentów. Do każdego z nich zostały przypisane dwa współczynniki z zakresu [0-1]: współczynnik poprawności działania, na podstawie którego liczona była reputacja, oraz współczynnik prawdomówności odpowiadający za reputację rekomendacyjną.

Na wykresie 6.9 przedstawiono wynik 100 symulacji dla następujących parametrów:

- aby odzwierciedlić strukturę sieci P2P podczas każdej z symulacji losowano poziom interakcji między agentami – mógł on przyjmować wartość z przedziału [5%-25%],
- założono losowy rozkład pozostałych parametrów, przy czym agenci uczciwi stanowili minimum 60% wszystkich agentów w systemie.

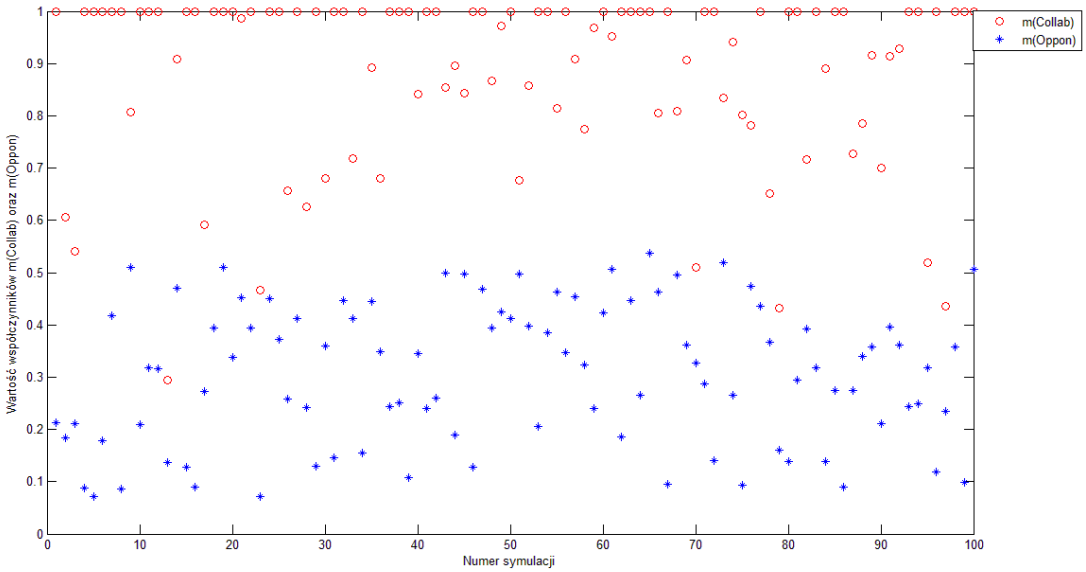
W czasie symulacji generowane były koalicje składające się z losowo wybranych agentów o rozmiarze dowolnym z przedziału [5-30]. Zadaniem koalicji było zaniżenie reputacji danej grupy agentów (liczebność grupy dobierana losowo z przedziału [3-30]). Dodatkowo przyjęto, iż agenci tworzący koalicję głosują w podobny sposób w co najmniej 90% przypadków.



Rys. 6.9 Wynik 100 symulacji obrazujący poziom współczynników $m(Collab_A)$ oraz $m(Oppon_A)$ przyjmując minimum 90% poziomu podobieństwa głosowania koalicjantów.

Poziom wykrywalności koalicji wyniósł 94%. Jednocześnie nie stwierdzono przypadków niewłaściwej klasyfikacji grupy agentów w rzeczywistości nie tworzących koalicji.

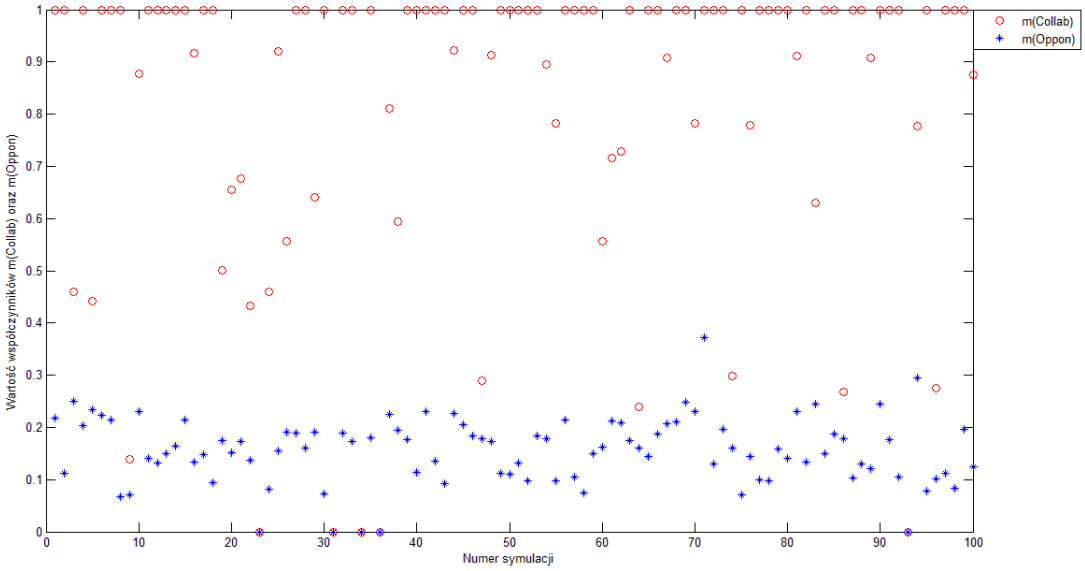
W kolejnej symulacji, której wyniki przedstawione są na rysunku 6.10, zmianie uległ poziom interakcji: jego górna granica została zwiększona dwukrotnie, do 50%, przy jednoczesnym zachowaniu wartości pozostałych parametrów. W tym przypadku wykrytych zostało 93% koalicji.



Rys. 6.10 Wynik 100 symulacji obrazujący poziom współczynników $m(Collab_A)$ *oraz* $(Oppon_A)$ *przy założeniu minimum 90% poziomu podobieństwa głosowania koalicjantów i maksymalnym poziomie interakcji między agentami równym 50%.*

Na tym etapie przeprowadzono także dodatkowe symulacje, które pozwoliły stwierdzić brak wrażliwości mechanizmu wykrywania koalicji na poziom reputacji agentów, rozmiar koalicji oraz rozmiar atakowanej grupy węzłów.

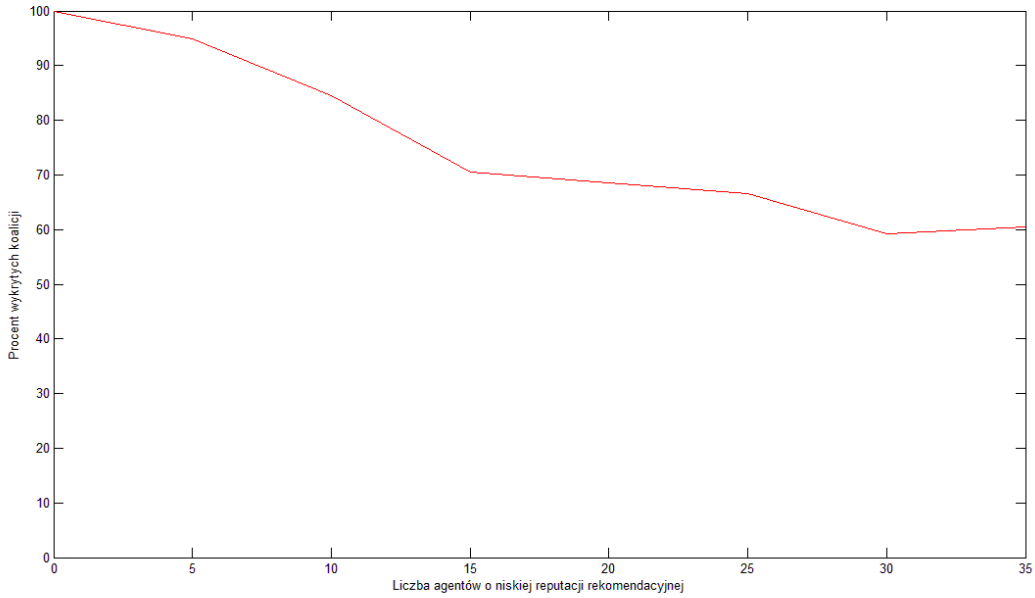
W kolejnej symulacji agenci tworzący koalicje mieli pełną dowolność indywidualnego wyboru agentów, z którymi się komunikują oraz możliwość przesyłania niezależnych ocen. Dyscyplina podczas głosowania obowiązywała tylko w wypadku agentów wyznaczonych jako cele ataku koalicji. Reszta parametrów pozostała niezmieniona w stosunku do pierwszej symulacji.



Rys. 6.11 Wynik 100 symulacji obrazujący poziom współczynników $m(Collab_A)$ *oraz* $(Oppon_A)$. *Dyscyplina głosowania wewnątrz koalicji obowiązywała tylko odnośnie atakowanych agentów. Maksymalny poziom interakcji między agentami wynosił 25%.*

Tym razem wykrytych zostało 83.6% koalicji – wynik jest ułamkowy, gdyż w kilku przypadkach nie zostali zidentyfikowani wszyscy agenci wchodzący w skład koalicji. Co więcej, trzykrotnie większa koalicja została zaklasyfikowana jako kilka niezależnych mniejszych.

Ustalono jednocześnie, iż na niższy poziom wykrywalności koalicji wpływają agenci scharakteryzowani jako ”kłamiący często, ale nie notorycznie”. System wykrywania koalicji został tak zaprojektowany, aby ignorować notorycznych kłamców na etapie wyznaczania agentów ocenianych kontrowersyjnie. Kłamiący umiarkowanie oceniają niekiedy zgodnie a niekiedy przeciwnie do koalicji, co pozwala ich zidentyfikować i wykluczyć (co ma miejsce w etapie 5b algorytmu). Kłamiący często ale nie notorycznie są natomiast trudniejsi do wykrycia, ich większa liczba doprowadza do generowania kontrowersyjnie ocenianych agentów, co może prowadzić do błędów przy późniejszej klasteryzacji na etapie 3) algorytmu. Zależność pomiędzy liczbą takich agentów, a poziomem wykrywania koalicji została ustalona w wyniku symulacji i przedstawiona na wykresie 6.12 (parametry zgodne z symulacją trzecią):



*Rys. 6.12 Zależność pomiędzy liczbą agentów określonych jako „*kłamiący często, ale nie notorycznie*", a poziomem detekcji koalicji.*

Rozdział VII

Routing

7.1 Wprowadzenie do routingu w sieciach P2P

Wybór sposobu routingu pakietów, obok przyjętej metody kompresji i przesyłania danych, jest najważniejszym aspektem wpływającym na jakość transmisji czasu rzeczywistego w Internecie. Jak wspomniano w poprzednich rozdziałach, transmisja czasu rzeczywistego wiąże się z zachowaniem restrykcyjnych ograniczeń wynikających z dopuszczalnego stałego (delay) oraz zmiennego (jitter) opóźnienia w przesyłaniu danych. Ograniczenia te w połączeniu z dodatkowymi wymogami związanymi z zapewnieniem zarówno niezawodności jak i wysokiego poziomu bezpieczeństwa transmisji sprawiają, iż problem routingu jest zagadnieniem złożonym i skomplikowanym. Podstawowe trudności wynikają z samej konstrukcji sieci Internet, której pierwotny projekt nie zakładał tego typu transmisji. Stąd też standardowy sposób przesyłania informacji w Internecie opiera się na metodzie best-effort, w której pakiety wysyłane są w najszybszy możliwy sposób, zgodnie z kolejnością z jaką napływają. Rozwiązanie to sprawdza się w wypadku standardowej transmisji danych, gdzie opóźnienia związane z kolejkowaniem nie są mocno odczuwalne, a zastosowane protokoły transmisji (najczęściej TCP) pozwalają na retransmitowanie pakietów utraconych w wyniku błędów na trasie routingu. Jednakże w odniesieniu do transmisji real-time, metody te nie są adekwatne, gdyż w wypadku przeciążonych łącz proces kolejkowania może wprowadzać nadmierne opóźnienia, a stosowana w TCP retransmisja utraconych pakietów przeważnie okazuje się zbyt wolna. W standardowej konfiguracji TCP brak danego pakietu wykrywany jest z opóźnieniem, gdyż w celu zwiększenia wydajności transmisji potwierdzenie ACK przesyłane jest zazwyczaj dla większej grupy pakietów. Nawet po wprowadzeniu odpowiedniej zmiany w konfiguracji stosu TCP/IP, umożliwiającej szybsze wykrycie i retransmisję brakującego pakietu, przesłany ponownie może

dotrzeć już po czasie przyjętym w specyfikacji danej usługi real-time i co za tym idzie, zostać odrzucony. Z tego powodu przy konstrukcji protokołów komunikacyjnych na potrzeby usług czasu rzeczywistego zazwyczaj rezygnuje się z mechanizmów kontroli przepływu i retransmisji znanych z TCP na rzecz protokołu UDP, będącego protokołem bezpołączeniowym, nie gwarantującym integralności i transmisji, ale przez to także szybszym. Ponieważ nagłówek UDP nie zawiera takich danych jak numer sekwencyjny czy czas wysłania pakietu, informacje te muszą zostać dodane do końcowego, zbudowanego nad UDP protokołu. Najpopularniejszym protokołem tego typu jest Real-time Transport Protocol (RTP) [48] opracowany w 1996 roku i wciąż powszechnie używany przez wiele serwisów.

Inne często stosowane w Internecie protokoły routingu również nie uwzględniają aspektów istotnych przy transmisji czasu rzeczywistego [49]. Pomimo iż łącza internetowe w sieci szkieletowej posiadają zazwyczaj nadmiarowe zdolności przesyłowe i w znacznej liczbie przypadków sieć ta zapewnia wymaganą jakość połączenia, mogą jednak pojawić się krótkie okresy, w których procent traconych pakietów jest znacznie wyższy. Pakiety tracone są przede wszystkim w wyniku chwilowych przeciążeń, awarii fizycznych oraz błędów w routingu. Znaczny wpływ na degradację jakości usługi VoIP mają także problemy związane z niezawodnością sieci. Badania ilościowe strat w pakietach wskazują, iż pomimo że średni współczynnik strat nie jest wysoki (0.6-5.2% oraz 0.44% według opracowań [5] i [6]), zdarzają się także okresy, gdy straty te sięgają 13% (w ciągu jednej godziny). Kompleksowe badanie strat w jakości VoIP związanych z przesyłaniem głosu w sieci szkieletowej zostały opisane miedzy innymi w pracy [50].

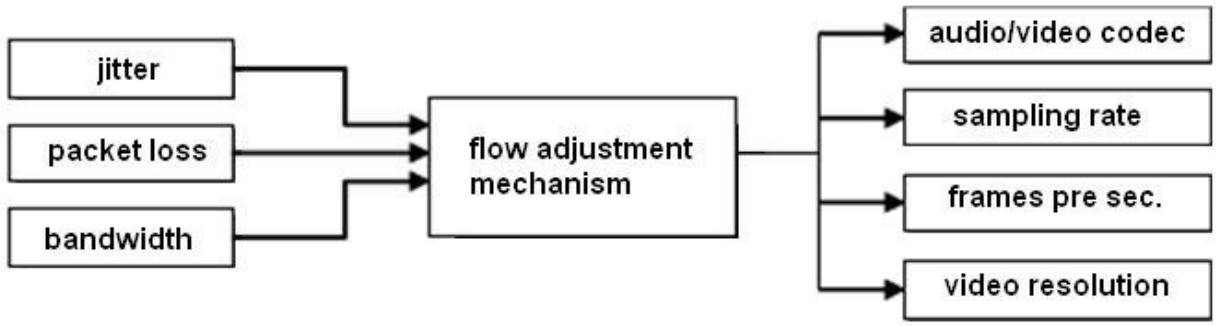
Kolejnym istotnym problemem w przesyle danych real-time przez Internet jest zagadnienie komunikacji z urządzeniami znajdującymi się w sieci wewnętrznej. Kończące się zasoby adresów IP oraz przyjęty model tworzenia sieci internetowych sprawił, iż wiele komputerów, szczególnie w sieciach korporacyjnych oraz osiedlowych, ma przydzielony jedynie wewnętrzny adres IP. Taki numer jest indywidualny jedynie wewnątrz danej sieci, natomiast globalny adres IP, używany do komunikacji z jednostkami zewnętrznymi jest współdzielony dla całej podsieci. Za przydzielanie poszczególnych portów komunikacyjnych odpowiedzialny jest działający po stronie routera mechanizm Network Adres Translation (NAT). Wiąże on (według przyjętego algorytmu) porty właściwe zewnętrznemu adresowi IP z odpowiednim portem i lokalnym numerem IP komputera chcącego zainicjować połączenie wychodzące. Jak można

łatwo zauważyć, aby urządzenie będące w wewnętrznej sieci zdolne było do odbioru komunikacji z sieci zewnętrznej, musi nastąpić wpierw powiązanie NAT, a co za tym idzie, jednostka z wewnętrznym adresem IP musi zawsze być stroną inicjującą połączenie (wyjątek stanowi tu sytuacja z aktywną usługą tunelowania). Jest to najistotniejsza wada wynikająca ze stosowania mechanizmu NAT, która w praktyce najczęściej uniemożliwia inicjację połączenia pomiędzy dwoma komputerami znajdującymi się wewnątrz różnych sieci lokalnych. Problem ten częściowo da się rozwiązać stosując mechanizmy takie jak STUN (Session Traversal Utilities for NAT) [19], które potrafią wykryć obecność NAT-u, a następnie próbują odgadnąć numery portów przez niego przydzielane. Ponieważ jednak wśród producentów sprzętu istnieje różnorodność w projektowaniu algorytmów wiążących (najprostsze z nich przydzielają kolejne wolne numery z puli dostępnych portów lub numery losowe) mechanizmy typu STUN często są nieefektywne [20]. Ostatecznie problem komunikacji pomiędzy urządzeniami będącymi w rożnych sieciach lokalnych sprowadza się do konieczności korzystania z trzeciej jednostki, stanowiącej serwer pośredniczący, która przekazuje dane pomiędzy komunikującymi się urządzeniami. Poza koniecznością utrzymywania serwera pośredniczącego, rozwiązanie takie może być szczególnie nieefektywne w wypadku usług czasu rzeczywistego, gdyż serwer ten najczęściej nie leży na optymalnej trasie między komunikującymi się jednostkami, a co za tym idzie wprowadza on dodatkowe znaczne opóźnienia w transmisji.

Transmisja real-time wiąże się ściśle z problemem zapewnienia odpowiedniej jakości przekazu zwanej QoS. Jednakże, pomimo iż zagadnienie QoS jest jednym z najistotniejszych wyzwań stawianych sieci Internet, wciąż nie powstał jednolity, w pełni funkcjonalny mechanizm odpowiedzialny za jego realizację. Techniki te, z których najpopularniejsze to IntServ [2] oraz Diffserv [3], opierają się na wcześniejszej rezerwacji wymaganych zasobów sieciowych oraz klasyfikacji typów transmisji (na podstawie pola Type of Service w nagłówku IP przesyłanych pakietów). Należy jednak zauważyć, że metody te są ściśle zależne od polityki przyjętej przez danego dostawcę usługi i wymagają współpracy wszystkich routerów będących na danej trasie. To z kolei sprawia, że często nie są one w stanie spełnić swojej funkcji i zapewnić pożądanej jakości usługi na całej trasie routingu.

Wraz z nieustannie rosnącą popularnością usług czasu rzeczywistego w Internecie, problem zapewnienia odpowiedniego poziomu danej usługi nieustannie zyskuje na znaczeniu. W ciągu ostatnich lat zostało zaprezentowanych wiele mechanizmów mających na celu

zwiększenie niezawodności oraz jakości transmisji. Ich podstawowym zadaniem jest niwelowanie skutków problemów pojawiających się podczas routingu. Są one związane zarówno ze sposobem kompresji i przesyłania danych, jak i z ich kolejkowaniem i ostatecznie routingiem [51]. Jedną z powszechnie używanych metod z pierwszej grupy jest mechanizm automatycznej adaptacji jakości transmisji do bieżących warunków sieciowych [52]. Na podstawie informacji na temat aktualnie panujących warunków, mechanizm ten ma możliwość dostosowania parametrów używanego kodeka, takich jak częstotliwość próbkowania, wielkość ramki oraz rozmiar bufora odbiorczego. W celu dokonania właściwych korekt, niezbędne jest określenie rzeczywistej jakości trasy, po której odbywa się transmisja. Używa się do tego informacji zwrotnej, która może być przesyłana dodatkowym kanałem komunikacyjnym (jak to ma miejsce w wypadku RTCP [53]), dodawana bezpośrednio do transmisji czasu rzeczywistego (przy wykorzystaniu np. techniki znakowania wodnego [54]) lub też wpisywana do nagłówka zastosowanego protokołu.



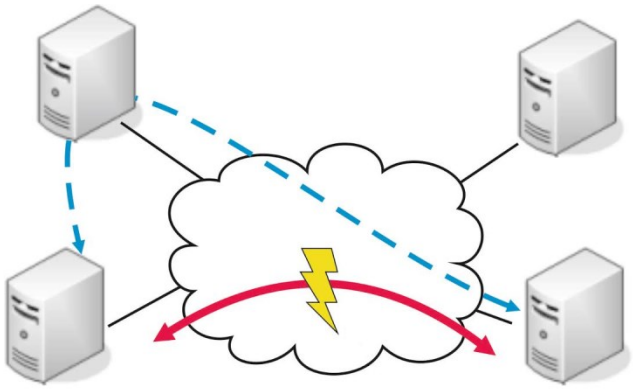
Rys. 7.1 Mechanizm automatycznego dostosowywania parametrów transmisji.

Odpowiednio skonfigurowany mechanizm adaptacyjny jest istotnym narzędziem, które może być użyte w celu podtrzymania funkcjonowania danej usługi w sytuacjach, gdy akceptowalna jest częściowa degradacja jakości audio/video.

Innym mechanizmem, stosowanym w celu kompensacji błędów pojawiających się podczas routingu, jest mechanizm kodowania korekcyjnego FEC [55] oparty na dodawaniu do właściwej komunikacji dodatkowych, redundantnych danych, które mogą być wykorzystane do korekty błędów transmisji. Wadą tego rozwiązania jest jednak znaczne zwiększenie narzutu związanego z transmisją. Przykładowa realizacja FEC wykorzystująca kodowanie Reeda-Solomona [56] o parametrach (n, k) polega na utworzeniu $n - k$ bloków FEC dla każdej grupy k bloków danych. Dodatkowy narzut związany z FEC wynosi tu $(n - k)$, natomiast kodowanie jest w stanie poprawić maksymalnie do $(n - k)/2$ przekłamań.

Powyższe dwa mechanizmy wciąż jednak są wyłącznie narzędziami komplementarnymi, nie rozwiązującymi bezpośrednio problemów pojawiających się w trakcie routingu.

Ponieważ sam proces routingu w Internecie jest procesem zewnętrznym, konkretny serwis ma zazwyczaj bardzo ograniczone możliwości wpływu na jego przebieg. Sytuacja ta może jednak ulec zmianie, jeśli architekturę serwisu oprze się na infrastrukturze P2P. Jak już wspomniano w rozdziale IV, komputery podpięte do takiej sieci stanowią jej węzły i dzięki zdolności do równoległego działania zarówno jako klienci jak i serwery, aktywnie uczestniczą w procesie komunikacji. Jedną z istotnych funkcjonalności wynikających z użycia sieci P2P jest możliwość tworzenia tzw. sieci nakładkowych (tzw. overlay networks), będących abstrakcyjną warstwą sieciową zbudowaną ponad fizyczną strukturą sieci Internet. Pomimo iż komunikacja między węzłami takiej sieci wciąż odbywa się zgodnie z powszechnymi zasadami routingu obowiązującymi w Internecie, jej dokładna charakterystyka staje się mniej istotna, a wirtualne łącze miedzy węzłami przypisane ma jedynie parametry odpowiadające sumarycznej charakterystyce fizycznej ścieżki, po której przebiega transmisja. Możliwe staje się zatem opracowywanie własnych algorytmów routingu, działających wewnątrz wirtualnej sieci nakładkowej: na jej węzłach oraz wirtualnych łączach. Algorytm taki, w wypadku awarii lub znacznego pogorszenia się jakości któregoś z elementów na fizycznej ścieżce, zaktualizuje parametry wirtualnego łącza, a następnie wyznaczy zastępczą, najbardziej optymalną ścieżkę, prowadzącą przez inne węzły sieci P2P.



Rys. 7.2 Przykład działania routingu w sieci nakładkowej w wypadku awarii łącza.

Należy jednak zwrócić uwagę na pewną wadę takiego rozwiązania, a mianowicie na to, że trasa routingu wyznaczona w oparciu o sieć nakładkową może okazać się gorsza niż istniejące

trasy bazujące na fizycznej strukturze Internetu. Wynika to z faktu, iż węzły sieci P2P nie muszą leżeć bezpośrednio na optymalnej trasie, a zatem wyznaczona w oparciu o nie trasa routingu będzie suboptymalna. Przyjmując jednak, iż używana sieć nakładkowa jest niewielka względem fizycznej struktury sieci Internet, opóźnienia wynikające z suboptymalnego położenia węzłów będą nieznaczne, a ponadto straty te będą rekompensowane przez zysk wynikający z możliwości aktywnego monitorowania stanu łącz oraz szybkiego reagowania na wszelkie awarie. Jednocześnie, ponieważ na poziomie serwisu istnieje możliwość wyboru konkretnego węzła, do którego podłączony jest dany użytkownik, jeśli węzeł taki wybrany zostanie w sposób optymalny (np. położony będzie w tym samym systemie autonomicznym co użytkownik), dodatkowe opóźnienia w transmisji real-time wynikające z użycia przekaźników będą niewielkie.

Sieć nakładkowa może być także z powodzeniem wykorzystywana do optymalizacji procesu przesyłania strumieniowanych danych w czasie rzeczywistym jednocześnie do licznej grupy odbiorców. Zagadnienie to, zwane multicast, zostało opisane w podrozdziale 3.2.3.

Także opisany powyżej, związany z mechanizmem NAT, problem z nawiązaniem połączenia pomiędzy komputerami będącymi w rożnych sieciach lokalnych może zostać rozwiązany na bazie sieci nakładkowej P2P. Jako że każdy z węzłów tworzących taką sieć może działać jako przekaźnik, może on również zostać użyty w celu asystowania w wymianie informacji pomiędzy jednostkami za NAT-em.

Ze względu na opisane powyżej zalety sieci nakładkowych, w ostatnich latach prowadzone są intensywne prace nad konstrukcją takich sieci oraz opracowaniem dla nich wydajnych algorytmów routingu. Jedną z pierwszych implementacji takiej sieci jest opracowana w roku 2001 roku sieć RON (Resilient Overlay Network) [57]. Jej zadaniem było między innymi monitorowanie stanu ścieżek na trasie routingu między węzłami i w wypadku ich awarii wyznaczanie alternatywnych dróg komunikacji. Sieć ta posłużyła do wielu badań nad routingiem oraz gubieniem pakietów. Badania analizujące zysk wynikający z wykorzystania prostego mechanizmu routingu opartego o sieć nakładkową wykazały, iż sieć ta zdolna jest do zamaskowania średnio 60% awarii poprzez zaproponowanie alternatywnej trasy. Architektura ta pozbawiona jednak była mechanizmu gwarantującego QoS, tak więc wyznaczone ścieżki mogły nie spełniać wymogów związanych z usługami real-time.

7.2 Routing QoS

Zwyczajowo protokoły routingu przy wyznaczaniu trasy posługują się pewnymi metrykami. Prostsze z nich bazują na wektorze odległości od celu (np. liczbie przeskoków, jak w wypadku protokołu RIP). Drugą klasę, dającą większą elastyczność, stanowią protokoły trasowania oparte na stanie łącza (przepustowość, niezawodność, odległość od routera). Parametry opisujące stan łącz na konkretnej trasie możemy podzielić na dwie zasadnicze grupy:

- parametry addytywne – propagują się w sposób addytywny na danej trasie. Dzięki temu ich wartość dla całej trasy p można wyznaczyć w bardzo prosty sposób zgodnie z zależnością: $c(p) = \sum_{l \in p} c_l$, gdzie c_l jest wartością parametru c na danym linku l. Przykładem takich parametrów jest opóźnienie, liczbę gubionych pakietów, jitter lub koszt energii związanej z transmisją. Do tej grupy zaklasyfikować można także parametry, które propagują się w sposób multiplikatywny, gdyż propagacja taka da się prosto zapisać w postaci sumy przy użyciu funkcji logarytmicznej. Dla przykładu całkowitą liczbę utraconych pakietów wyznaczyć można zgodnie ze wzorem: $e^{\sum_{l \in p} \ln(lr_l)}$, gdzie lr_l jest współczynnikiem straty powiązanym z linkiem l.

- parametry nieaddytywne, które jak wskazuje ich nazwa nie zachowują cechy addytywnej propagacji wzdłuż trasy. Ustanawiają one zazwyczaj globalne ograniczenia na ścieżce, takie jak minimalna przepustowość, minimalny poziom stosowanych zabezpieczeń lub minimalna reputacja danego węzła.

Obie z wymienionych klas parametrów muszą być wzięte pod uwagę podczas procesu optymalnego wyboru trasy routingu. Parametry nieaddytywne nie wpływają jednak znacznie na złożoność obliczeniową algorytmu wyboru trasy. Wynika to z faktu, iż stawiane przez nie ograniczenia sprowadzają się do problemu znalezienia wąskiego gardła, a więc mogą być spełnione poprzez redukcję przestrzeni rozwiązań (czyli w praktyce struktury sieci) danej grafem $G = (E, V)$ do swojego podgrafu $G' = (E', V')$, gdzie $E' = \left\{e \in E : \forall_{i=[1..k]} c_i^e \geq L_i^{NA}\right\}$ oraz L_i^{NA} dla $i = 1 \ldots k$ są ograniczeniami danymi parametrami nieaddytywnymi, natomiast c_i^e stanowi aktualną wartość parametru nieaddytywnego powiązanego z linkiem e.

Z drugiej strony, ograniczenia dane parametrami addytywnymi stanowią zagadnienie znacznie trudniejsze do rozwiązania. Zakładając koszt transmisji dla danego łącza $l \in p$ jako s_l

i biorąc pod uwagę zbiór ograniczeń danych parametrami addytywnymi, możemy zdefiniować problem znalezienia optymalnej ścieżki jako minimalizacji funkcji kosztu $S(p) = \sum_{l \in p} s_l$ przy wymogu spełnienia $i = 1 \ldots n$ ograniczeń na parametry addytywne: $\sum_{l \in p} c_i^l \leq L_i^A$.

Przy problemie minimalizacji względem tylko jednego parametru (jak to ma miejsce w większości stosowanych algorytmów routingu) jest to zagadnienie proste i można je rozwiązać stosując algorytm szukania najkrótszej ścieżki (najczęściej Dijkstry lub Bellman-Forda). Jednakże minimalizacja względem pojedynczego parametru nie gwarantuje zapewnienia żądanej jakości (QoS) dla trasy routingu, dlatego konieczne jest uwzględnienie większej liczby ograniczeń. Użyteczne jest także dodanie dodatkowych wag do poszczególnych ograniczeń wprowadzając w ten sposób pewne priorytety. Zagadnienie to jest szczególnie istotne w przypadku usług czasu rzeczywistego, gdzie trasa szybka, ale gubiąca wiele pakietów może okazać się gorsza od wolniejszej ale bardziej niezawodnej. Jednak wzrost liczby parametrów istotnie zwiększa złożoność obliczeniową problemu i sprawia, że jego rozwiązanie może być niemożliwe w akceptowalnym czasie.

Podsumowując dotychczasowe rozważania, przy konstruowaniu funkcji kosztu ścieżki (τ_p), uwzględnić należy następujące wymagania:

1) Wszystkie wymagane do procesu optymalizacji aspekty związane z typem usługi, wymogami użytkowników oraz dostępną infrastrukturą fizyczną.
2) Każde z $c_{[1\ldots n]}$ wymagań stawianych szukanej ścieżce powinno zostać spełnione; w przeciwnym wypadku funkcja kosztu powinna przyjąć wartość zerową:

$$\tau_p(c_1, \ldots, c_n) > 0 \Rightarrow \forall_{i \in [1,n]} f_p(c_i) > 0, \tag{7.1}$$

gdzie $f_p(c_i)$ jest funkcją określającą stopień spełnienia kryterium c_i (addytywnego lub nieaddytywnego) zadanego szukanej ścieżce. Wymóg ten eliminuje możliwość wyboru trasy nie spełniającej choć jednego z zadanych kryteriów (np. gdy połączenie jest bardzo dobrej jakości, lecz nie zostało sklasyfikowane jako bezpieczne).

Wymóg 2) powoduje, iż funkcja kosztu powinna przyjąć formę iloczynu poszczególnych kosztów w miejsce ich sumy. Dodatkowo, aby mieć możliwość dynamicznego modyfikowania istotności poszczególnych wymogów (i ich całkowitego wyłączania), indywidualne wagi mogą być podane w formie wykładniczej, przez co końcową funkcję kosztu można zapisać jako:

$$\tau_p(c_1, \ldots, c_n) = \prod_{i=[1\ldots n]} f_p(c_i)^{w_i}. \quad (7.2)$$

Zagadnienie tego rodzaju należy do rodziny problemów zwanych Multi-Constrained Optimal Path (MCOP), czasem definiowanych także jako Restricted Shortest Path (RSP), i poza znajdowaniem tras spełniających założenia QoS jest także szeroko używane w wielu innych dziedzinach, takich jak zarządzanie ruchem ulicznym lub łańcuchami zaopatrzenia. Jednakże znalezienie optymalnego rozwiązania MCOP w akceptowalnym czasie możliwe jest tylko w przypadku niewielkich sieci, gdyż problem ten charakteryzuje się NP-zupełną złożonością obliczeniową [58]. Inne podobne zagadnienia optymalizacyjne z tej klasy, jak np. Multi-Constrained Path (MCP) znajdujący jedynie trasę spełniającą wymogi QoS (bez optymalizacji), są także problemami NP-zupełnymi. Istnieje szereg algorytmów rozwiązujących tego typu zadania [59], lecz stosowane są one zazwyczaj tylko w małych sieciach, gdyż bazują na metodzie brute force. W przypadku gdy wymagane jest uzyskanie szybszego wyniku (np. szybka zmiana trasy w wypadku awarii łącza), można posłużyć się algorytmami heurystycznymi, dającymi przybliżone rozwiązanie problemu MCOP. Przegląd przez dokładne i przybliżone algorytmy rozwiązujące problem MCOP można znaleźć w [59] [60] [61].

7.3 Przegląd literatury dotyczącej routingu QoS

Zadaniem prawidłowego protokołu routingu QoS jest wybór trasy spełniającej ograniczenia określone parametrami QoS, przy jednoczesnej próbie minimalizacji zasobów zużywanych przez infrastrukturę sieciową. Podstawowym zagadnieniem rozpatrywanym w ramach routingu QoS są problemy należące do wspomnianej wcześniej klasy MCOP. Ze względu na dużą złożoność obliczeniową klasycznego problemu MCOP, opracowany został szereg alternatywnych metod pozwalających na wybór tras spełniających wymogi QoS. Metody te opierają się zazwyczaj na podejściu hierarchicznym, charakteryzującym się niższą złożonością obliczeniową, osiąganą kosztem dokładności rozwiązania.

W wypadku usług czasu rzeczywistego, na których koncentruje się niniejsza rozprawa, jednym z głównych elementów wpływających na wybór trasy routingu jest całkowite opóźnienie transmisji, będące parametrem addytywnym łącza. Kolejne ograniczenia mogą natomiast stanowić kombinację dodatkowych, zarówno addytywnych jak i nieaddytywnych parametrów. Często w celu uproszczenia problemu zagadnienie routingu QoS ogranicza się do zastosowania

kombinacji pojedynczego parametru addytywnego z parametrami nieaddytywnymi. Jak pokazano na wykresie 3.1, przy dużym stopniu wykorzystania łącza zwiększa się opóźnienie transmisji, dlatego jako parametr nieaddytywny przyjmowana jest zazwyczaj minimalna przepustowość łącza. Ograniczony w taki sposób problem, zwany wówczas Bandwidth-Restricted-Path (BRP), można rozwiązać jednym z poniższych dwóch algorytmów.

Widest-shortest path (WSP) – znajduje najkrótszą ścieżkę spośród wszystkich spełniających wymagania na przepustowość. WSP wyznaczyć można przy użyciu zmodyfikowanej wersji algorytmu Bellmana-Forda, zwanego Iterative Shortest Path [62], lub też zmodyfikowanym algorytmem Dijkstry, gdzie wprowadzona została dodatkowa, drugorzędna względem odległości, metryka powiązana z dostępną przepustowością.

Shortest-widest path (SWP) – znajduje trasę o największej dostępnej przepustowości, spośród wszystkich dostępnych tras, a następnie – jeśli jest ich wiele – wybiera tę spośród nich, która charakteryzuje się najkrótszą odległością. Realizacja algorytmu opiera się na dwukrotnym wywołaniu algorytmu Dijkstry.

Powyższe algorytmy można w prosty sposób rozszerzyć tak, aby mogły uwzględnić większą liczbę ograniczeń danych parametrami nieaddytywnymi poprzez uprzednie zawężenie przestrzeni rozwiązań (zgodnie z metodą opisaną dla zmiennych nieaddytywnych w podrozdziale 7.2).

Powyższe algorytmy różnią się od siebie głównym celem optymalizacyjnym. WSP ukierunkowany jest na zminimalizowanie wykorzystania zasobów sieciowych, co osiągane jest poprzez faworyzowanie długości nad przepustowością. Algorytm WSP sprawdza się zatem lepiej w sytuacjach, gdy sieć jest przeciążona. Natomiast SWP, poprzez preferowanie przepustowości nad czasem dostarczenia, skierowany jest na równoważenie obciążenia sieciowego.

Poza dwoma powyższymi algorytmami w pracy [62] omówiony został algorytm Dynamic-Alternative-Path (DAP). Jest to algorytm zbliżony do WSP, lecz z dodanym ograniczeniem górnym na maksymalną długość ścieżki. DAP wybiera ścieżkę o maksymalnej długości $n + 1$, gdzie n jest długością ścieżki najkrótszej liczonej bez uwzględnienia ograniczeń wynikających z przepustowości. W powyższej pracy przedstawione zostały również rezultaty symulacji, z których wynika, iż DEP charakteryzuje się wyższą wydajnością od SWP oraz WSP w wypadku, gdy infrastruktura sieciowa jest mocno obciążona. Jednakże w wypadku

niewielkiego obciążenia systemu przewaga ta zanika i ostatecznie wydajność DEP plasuje się poniżej wydajności dwóch wspominanych algorytmów.

Powyższe algorytmy wymagają informacji na temat aktualnej przepustowości łącz, które można uzyskać korzystając z istniejących narzędzi. Najpopularniejsze z nich to pathChirp [63], pathLoad [64], czy cprobe [65]. Bazują one zazwyczaj na różnych wariantach pomiaru opóźnienia pakietów próbkujących. Dla przykładu w metodzie Self-Loading-Periodic-Stream (SLoPS) nadawca wysyła strumień pakietów (np. 100) o identycznych rozmiarach, z ustaloną szybkością R, natomiast odbiorca monitoruje zmianę opóźnienia pomiędzy kolejnymi strumieniami. Jeśli w danym strumieniu szybkość nadsyłania pakietów jest wyższa, niż pozwala na to zdolność przesyłowa łącza, to przesłanie takiego strumienia doprowadza do krótkiego przeciążenia w procesie kolejkowania, co skutkuje dodatkowym opóźnieniem. Zadaniem metody testującej jest takie dobranie szybkości nadsyłania pakietów, aby możliwie najlepiej zbliżyć się do aktualnej przepustowości łącza. Używa się w tym celu algorytmu analogicznego do wyszukiwania binarnego. Pełny przegląd metod wyznaczania dostępnej przepustowości można znaleźć w pracy [66].

Innym podejściem do problemu QoS w wypadu transmisji czasu rzeczywistego jest konstrukcja metryk wiążących istotne dla tej transmisji parametry w pojedynczą miarę, a następnie użycie klasycznego algorytmu wyszukiwania najkrótszej ścieżki. Należy tu jednak zaznaczyć, iż zależności pomiędzy parametrami łącza a jakością transmisji są nietrywialne, gdyż w grę wchodzi tu także szereg aspektów z warstwy aplikacji (m.in. rodzaj użytego kodeka, wielkość buforu oraz stosowane algorytmy korekcyjne). W pracy [67] zaproponowana została metoda aproksymacji kosztu danego łącza przy użyciu metryki zależnej od opóźnienia na tym łączu oraz współczynnika strat. Aby uniknąć złożoności obliczeniowej związanej z kilkoma parametrami addytywnymi, postanowiono tu założyć, iż utracony pakiet dotarł, lecz z opóźnieniem T_{max} większym od przyjętego maksymalnego progu akceptacji. Wówczas koszt danej ścieżki można zapisać jako:

$$cost = (1 - p) * T + p * T_{max} , \tag{7.3}$$

gdzie p jest prawdopodobieństwem utraty pakietu, a T opóźnieniem transmisji.

W pracy [68] zaproponowano natomiast użycie jako metryki kosztu łącza miary danej w modelu ITU-T E, stosowanej do oszacowania jakości dźwięku na podstawie parametrów sieci. Zdefiniowany jest w niej współczynnik R, wiążący różne aspekty jakości dźwięku:

$$cost \equiv R = R_0 - I_s - I_e - I_d + A, \tag{7.4}$$

gdzie R_0 opisuje efekty szumowe, I_s odzwierciedla inne zaburzenia pojawiające się równolegle z dźwiękiem, I_e odpowiedzialny jest za zaburzenia związane ze stratami pakietów, I_d określa zaburzenia spowodowane opóźnieniem, natomiast A dodatkowo kompensuje te zaburzenia uwzględniając różne aspekty konfiguracji użytkownika. Jednakże w wypadku VoIP przyjmuje się za zmienne tylko parametry I_e oraz I_d, a wstawiając w miejsce pozostałych ustalone wartości, otrzymuje się zależność:

$$cost \equiv R = 94.2 - I_e - I_d\,. \tag{7.5}$$

Ponieważ I_e oraz I_d wiążą straty pakietów oraz ich opóźnienie całkowite, a więc efekty związane zarówno z parametrami łącza jak i mającymi miejsce w warstwie aplikacji (m.in. buforowanie, kompresja), relacja pomiędzy tymi współczynnikami a fizycznymi parametrami nie jest oczywista. Jednakże w wyniku przeprowadzonych testów [69] opracowano modele, które mogą być użyte do wyznaczenia obu tych parametrów. Co ważne, istnieje także powiązanie pomiędzy współczynnikiem R a popularną miarą jakości dźwięku MOS:

$$MOS = 1 + 0.035R + 7 * 10^{-6}\mathrm{R}(R - 60)(100 - R). \tag{7.6}$$

Miara ta zatem może być użyta do routingu w oparciu o Quality of Experience (QoE) (należy tu zauważyć, że badania potwierdzają istnienie prostej relacji pomiędzy miarami QoS oraz QoE [70]).

Przekrój przez przykładowe możliwe do użycia funkcje kosztu przedstawiony został w tablicy 7.1.

Tablica 7.1 Przykłady miar kosztu ścieżki między węzłami sieci P2P.

Nazwa miary	**Koszt**
Liczba przeskoków	1
Opóźnienie łącza	T
Współczynnik strat	$-\log(1-p)$
Spodziewane opóźnienie [67]	$(1-p) * T + p * T_{max}$
Spodziewane opóźnienie uwzględniając mechanizm retransmisji [67]	$(1-p) * T + (p - 2p^2 + 3p^3)$
Miara jakości dźwięku [68]	$94.2 - I_e - I_d$
Przepustowość łącza [71]	$bandwidth$
Rozmiar bufora w węźle [72]	$buff\_size$

Symulacje przeprowadzone między innymi w [67] wykazały, iż w zależności od typu infrastruktury, lepsze wyniki można uzyskać za pomocą różnych miar. Przykładowo przy

sieciach o małych rozmiarach najlepiej sprawdza się miara oparta jedynie na współczynniku strat, jednak już w wypadku większych sieci ta sama miara uzyskuje najgorsze wyniki.

Zastosowane metody routingu mogą się także różnić w zależności od typu transmisji, a także w ramach jednego rodzaju przekazu. Szczególnie w wypadku przesyłania strumienia wideo w czasie rzeczywistym, gdzie oprócz opóźnienia w przekazie istotne jest również buforowanie danych oraz równoważenie obciążenia węzłów. W tego typu transmisji istotna jest przyjęta metoda rozsyłania danych w infrastrukturze, a więc właściwy dobór kolejnych węzłów uczestniczących w transmisji. W literaturze dostępne jest wiele rozwiązań tego problemu, między innymi dobór węzłów w oparciu o przepustowość łącz jakimi dysponują [71], wielkość ich buforu [72], ich wcześniejszą reputację [73] lub też fizyczną odległość od źródła [74]. Popularne są też rozwiązania oparte na systemach agentowych, takie jak architektura belief-desire-intention [75], gdzie agent dysponuje pewną wiedzą oraz własnymi przypuszczeniami na temat infrastruktury (belief), zestawem możliwych celów (desire) oraz podzbiorem tych celów (intentions), do których aktualnie dąży. Agent taki dysponuje zestawem gotowych planów i sam decyduje, który z nich aktualnie wykonać.

Jak widać w powyższym zestawieniu, istnieje szereg możliwych metod oceny i wyboru trasy routingu. Każda z nich ma swoje zalety oraz wady i najlepiej sprawdza się w określonych warunkach. Co więcej, niejednokrotnie przydatność danej metody można określić dopiero po przeprowadzeniu testów na docelowej infrastrukturze oraz konfiguracji. Dodatkowo dochodzi także rozbieżność wymagań oraz preferencji, w zależności od danej usługi real-time jak i samych użytkowników. Spostrzeżenie to skłoniło autora niniejszej pracy do odrzucenia podejścia bazującego na jednym konkretnym algorytmie routingu, a zamiast tego do skupienia się na opracowaniu mechanizmu pozwalającego dynamicznie wybierać najlepszą z dostępnych metod. Zaprojektowany specjalnie w tym celu Framework przedstawiony został w następnym podrozdziale [76].

7.4 Mechanizm wspomagania routingu dla systemu transmisji danych czasu rzeczywistego

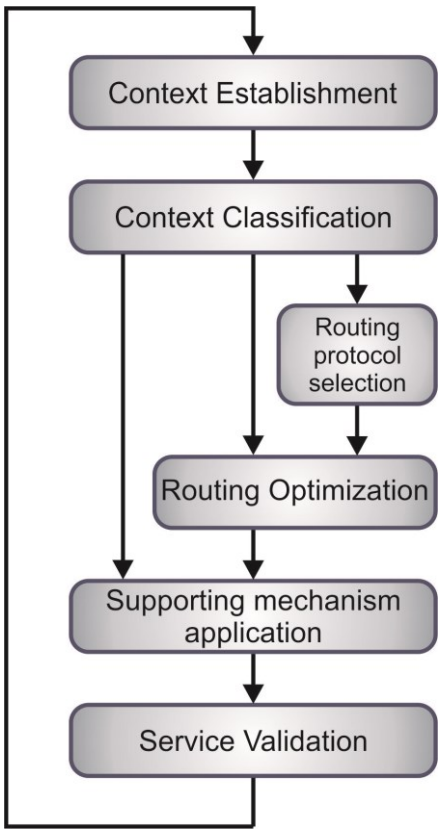
W celu optymalizacji procesu routingu danych czasu rzeczywistego, w ramach pracy zaprojektowany został Mechanizm Wspomagania Routingu (zwany dalej Frameworkiem) bazujący na danych kontekstowych. Jego głównym zadaniem jest optymalizacja samego procesu routingu, jednak został w nim także zaimplementowany szereg dodatkowych mechanizmów, pozwalających na zwiększenie poziomu bezpieczeństwa transmisji oraz zwiększenie samej jakości komunikacji, ze szczególnym naciskiem położonym na wskaźniki takie jak Quality of Service (QoS) oraz Quality of Experience [70]. Schemat Frameworku został przedstawiony na rysunku 7.3. W jego skład wchodzi sześć głównych warstw, przy czym niektóre z nich mogą być aktywowane opcjonalnie. Poniżej przedstawiona jest krótka charakterystyka wprowadzonych warstw:

- **warstwa ustanawiania kontekstu** – wykrywa ona potencjalne źródła danych kontekstowych, wyznacza istotne czynniki kontekstowe i ostatecznie dostarcza informacje kontekstowe potrzebne do wyboru algorytmu routingu oraz schematu bezpieczeństwa,
- **warstwa klasyfikacji kontekstu** – waliduje ona dostępne czynniki kontekstowe, a następnie buduje na ich podstawie kontekstowe rekomendacje dla protokołu routingu oraz systemu bezpieczeństwa,
- **warstwa wyboru protokołu routingu** – w oparciu o wyznaczone na podstawie kontekstu rekomendacje, wyznacza odpowiedni protokół routingu, wybierając go z dostępnego portfolio. W wypadku braku protokołu spełniającego dane wymogi, informuje ona o tym fakcie zainteresowane strony i w procesie negocjacji wymogi mogą zostać obniżone,
- **warstwa optymalizacji routingu** – dobiera konfigurowalne parametry protokołu routingu oraz aplikuje dodatkowe mechanizmy bezpieczeństwa,
- **warstwa mechanizmów wspomagających** – zawiera dodatkowe mechanizmy zwiększające jakość transmisji i niwelujące skutki błędów komunikacji (takich jak szybka retransmisja pakietów),

- **warstwa walidacji usługi** – pozwala uwzględnić informacje dostarczone w procesie trójstronnej walidacji (użytkownik końcowy, dostawca usługi oraz operator systemu) oraz, jeśli jest to wymagane, wprowadzić poprawki w proces routingu.

Dodatkowo, w celu ułatwienia testowania i wdrażania nowych protokołów przewidziane zostały także trzy tryby pracy Frameworku:

- **tryb konfiguracji** – wszystkie warstwy są aktywne, system jest dostrajany lub istotnie rekonfigurowany względem potrzeb danej usługi,
- **tryb niestabilny** – faza wyboru protokołu routingu jest pominięta, dostosowywane są mechanizmy wspomagające oraz bezpieczeństwa,
- **tryb stabilny** – faza wyboru protokołu routingu oraz jego optymalizacji zostaje deaktywowana. System działa w oparciu o wcześniej wybrany algorytm routingu oraz na podstawie ustalonej konfiguracji.



Rys. 7.3 Schemat Frameworku mechanizmu routingu kontekstowego dla potrzeb usług czasu rzeczywistego.

Poszczególne mechanizmy użyte we Frameworku opisane zostały w kolejnych podrozdziałach.

7.4.1 Warstwy ustanawiania oraz klasyfikacji kontekstu

Głównym zadaniem mechanizmu ustanawiania kontekstu (CEM) jest gromadzenie i wstępne przetwarzanie danych kontekstowych. Na potrzeby systemu będącego tematem pracy zdefiniowane zostały cztery główne źródła informacji kontekstowych:

- informacje profilowe użytkowników, prezentowane na etapie rejestracji do systemu. Każda aplikacja kliencka, w procesie przyłączania do systemu, generuje automatycznie opis profilu użytkownika. Przesyłane są dane takie jak: rodzaj używanych usług (typ strumienia real-time), obsługiwane mechanizmy bezpieczeństwa oraz związane z nim wymogi, aspekty socjalne (reputacja klienta), przyjęty model biznesowy (klient standardowy lub premium), itp. Jeśli jest taka możliwość, udostępniane są także dane odnośnie lokalizacji oraz ewentualnej mobilności. Profil stworzony w ten sposób może być dowolnie aktualizowany.

- dane przestawione przez dostawców usług. Dostawcy usług są także zobowiązani do dostarczania informacji odnośnie swojej specyfikacji. Poza punktami analogicznymi do tych z informacji użytkowników, przesyłane są dodatkowe dane odnośnie dostępnych zasobów, typów preferowanych klientów oraz aspektów ekonomicznych.
- dane dostarczane przez system reputacyjny – reputacja węzłów uwzględniana jest w procesie routingu, w szczególności eliminowane są z niego węzły zidentyfikowane jako niegodne zaufania.
- system monitorowania sieci – gromadzone przez agentów znajdujących się w węzłach informacje dotyczące topologii sieci oraz jakości i niezawodności łącz są analizowane. Każdy węzeł sieci (poza węzłami oznaczonymi jako ograniczone energetycznie) jest okresowo testowany i wyznaczane są parametry, takie jak opóźnienie, jitter, współczynnik gubienia pakietów, błędy transmisji oraz dostępna przepustowość na łączach.

Zgromadzone informacje są w następnym kroku walidowane, usuwane są dane błędne oraz redundantne, a następnie zostaje zdefiniowana przestrzeń kontekstowa (CS), zawierająca

wszystkie unikalne aspekty kontekstowe (CA). W kolejnym kroku różne aspekty kontekstowe klasyfikowane są, w zależności od posiadanych cech, do odpowiednich grup, takich jak: informacje sieciowe, schematy bezpieczeństwa, informacje socjalne (zaufanie, reputacja, model biznesowy), itp. W przypadku posiadania niepełnych informacji, system próbuje odgadnąć brakujące dane bazując na metodach wnioskowania (np. na podstawie typu deklarowanej usługi możliwe jest określenie wymogów względem jakości łącz komunikacyjnych oraz poziomu bezpieczeństwa). Aspekty kontekstowe, które nie zostały zaklasyfikowane do żadnej ze zdefiniowanych grup, zostają zignorowane.

Proces klasyfikacji kontekstu może uwzględniać wiele różnych kryteriów (przykłady dostępne np. w [77]). Może on być wewnętrzny lub zewnętrzny, powiązany z daną jednostką, danym działaniem lub aktualną sytuacją, a także z miejscem, czasem, otoczeniem itp. Na potrzebny opisywanego systemu zdefiniowane zostały trzy klasy kontekstu:

- **Kontekst Użytkownika (UC):** obejmujący oczekiwania i rekomendacje użytkownika, a także jego ekonomiczne oraz socjalne ograniczenia, pożądany poziom bezpieczeństwa oraz poziom jego reputacji i zaufania,
- **Kontekst Sieci (NC):** zawierający informacje odnośnie aktualnego stanu sieci, jej topologii, reputacji węzłów oraz łącz pomiędzy nimi, jej słabych punktów oraz potencjalnych zagrożeń,
- **Kontekst Dostawcy Usługi (SC):** obejmuje ograniczenia związane z przyjętym modelem biznesowym, dostępnymi zasobami oraz informacje odnośnie reputacji.

W zależności od wymagań, dane kontekstowe oraz klasy mogą być dowolnie modyfikowane/rozszerzane. W podstawowej konfiguracji mechanizmu, użytej w ramach systemu opisywanego w niniejszej pracy, dla każdego użytkownika $u \in U$ wyróżniamy następujące aspekty kontekstowe:

- współczynnik bezpieczeństwa, opisujący pożądane oraz obsługiwane mechanizmy bezpieczeństwa:

$$S: U \rightarrow \{trans\_sec, signaling\_sec, authentication, authorization\}$$

- wymagania sieciowe, związane z usługami real-time używanymi przez użytkownika:

$$NR: U \rightarrow \{best\_effort, \max\_delay, \max\_packet\_loss, \max\_jitter, reliability\}$$

- przyjęty model biznesowy:

$$BM: U \rightarrow \{free_{account}, normal_{user}, premium_{user}\}.$$

- aktualny profil energetyczny:

$$ES: U \rightarrow \{energy_{aware}, net_{powered}\}.$$

- aspekty socjalne:

$$SA: U \rightarrow R \times R_{rec} \times T,$$

gdzie R jest reputacją danego użytkownika, R_{rec} jego reputacją rekomendacyjną, natomiast T jego współczynnikiem zaufania (jeśli taki jest dostępny).

Na podstawie powyższych informacji, kontekst użytkownika można więc opisać za pomocą poniższego zestawu parametrów:

$$UC(u) = \{NR(u), S(u), BM(u), ES(u), SA(u)\}. \quad (1.12)$$

Czynniki kontekstowe dla sieci (*NC*) oraz dostawcy usługi (*SC*) tworzone są w analogiczny sposób. Następnie, na bazie utworzonych czynników, możliwe jest zdefiniowanie wymogów oraz rekomendacji kontekstowych. Przykładowe mapowanie czynników kontekstu na oceny w skali [0-4] przedstawione zostało w tablicy 7.2.

Tablica 7.2. Przykład przypisania ocen współczynnikom gubienia pakietów.

Współczynnik gubienia pakietów (plr)	Ocena
$< 0.001\%$	4
$0.01\% \geq \text{plr} > 0.001\%$	3
$0.1\% \geq \text{plr} > 0.01\%$	2
$1\% \geq \text{plr} > 0.1\%$	1
$> 1\%$	0

7.4.2 Wybór protokołu routingu

Aby dokonać wyboru konkretnego sposobu routingu w zależności od dostępnych informacji kontekstowych, w zaprojektowanym Frameworku postanowiono wprowadzić idee Portfolio Protokołów Routingu oraz funkcję oceniającą. Portfolio, utworzone na etapie konfiguracji Frameworku, stanowi logiczną strukturę służącą do przechowywania istotnych

informacji o wszystkich obsługiwanych przez infrastrukturę algorytmach routingu. W połączeniu z funkcją oceniającą, używane jest ono do zautomatyzowania procesu wyboru najlepszego dostępnego algorytmu routingu w zależności od wymogów i rekomendacji każdej ze stron biorących udział w transmisji (zdefiniowanych poprzez czynniki kontekstowe).

Portfolio protokołów routingu

W związku z tym, że nowe protokoły routingu mogą być dodawane do Frameworku na etapie konfiguracji, może on obsługiwać wiele różnych protokołów odpowiednich dla różnych usług czasu rzeczywistego. Protokoły te mogą zostać klasyfikowane pod względem wielu cech, a z punktu widzenia usług czasu rzeczywistego najistotniejsze z nich to:

- **Wsparcie dla mechanizmu Quality-of-Service (QoS).** W zależności od rodzaju użytego sposobu routingu oraz przyjętej miary dla ścieżek, dany algorytm zostaje sklasyfikowany jako wspierający QoS bądź nie. Możliwe są tu dodatkowe podklasy odzwierciedlające główny priorytet danego mechanizmu routingu (minimalizacja opóźnienia, gubienie pakietów, zapewnianie wysokiej przepustowość itp.).

- **Dostępność mechanizmów zapobiegających degradacji jakości transmisji.** Jak już zaznaczano wielokrotnie, usługi czasu rzeczywistego są bardzo wrażliwe na spadki jakości transmisji. Z tego powodu preferowane są algorytmy routingu potrafiące wykryć takie spadki jakości, a następnie szybko wyznaczyć alternatywną ścieżkę. W tym celu w niniejszym Frameworku zastosowano agentów (aplikacje zainstalowane po stronie węzłów sieci) monitorujących stan infrastruktury, jednak decyzja odnośnie konkretnej reakcji pozostawiona jest w gestii danego protokołu routingu.

- **Energooszczędność**. W przypadku urządzeń opartych na zasilaniu bateryjnym (jak np. sieci sensorowe) energooszczędność jest jednym z najistotniejszych aspektów ich funkcjonowania. Pomimo iż zdecydowana większość algorytmów routingu używanych w systemach real-time pozbawiona jest mechanizmów wspomagania oszczędzania energii, istnieją także protokoły projektowane właśnie z myślą o pracy na węzłach o ograniczonych zasobach energetycznych. Bazują one zazwyczaj na zaawansowanych

mechanizmach sterujących wielostopniowymi stanami uśpienia oraz adaptacyjnym dostosowywaniu mocy transmisji (w wypadku komunikacji radiowej). Przykładową implementację takiego algorytmu routingu można znaleźć miedzy innymi w [78].

- **Bezpieczeństwo komunikacji.** Istnieje szeroki wachlarz ataków skierowanych przeciw protokołom routingu. Ich szerszego opisu dokonano w podrozdziale 7.5. Różne protokoły routingu mogą się charakteryzować odpowiednio: brakiem jakichkolwiek mechanizmów bezpieczeństwa, istnieniem mechanizmów ochrony przed podsłuchiwaniem lub przechwyceniem transmisji (poprzez szyfrowanie komunikacji oraz autoryzację węzłów) lub dodatkową ochroną przed złośliwymi węzłami wewnątrz sieci [79]. Jednakże nawet w wypadku, gdy dany protokół routingu nie dysponuje mechanizmami bezpieczeństwa, to ochrona ta częściowo zapewniana jest między innymi poprzez wykorzystanie reputacji i eliminowanie negatywnie ocenianych węzłów z infrastruktury.

- **Przyjęta metoda wyznaczania trasy routingu.** Obok metod hybrydowych, istnieją dwa najpopularniejsze sposoby wykrywania trasy routingu. Pierwszy z nich to routing proaktywny, opierający się na ciągłej kontroli stanu sieci oraz stałym przechowywaniu i aktualizowaniu tablic routingu. Pakiety typu keep-alive/hello są regularnie przesyłane między węzłami, a w sytuacji wykrycia zmian stanu łącza, aktualizacja jest rozsyłana do pozostałych węzłów sieci. Drugi sposób wykrywania trasy, zwany routingiem reaktywnym, polega na wykrywaniu trasy na życzenie. Podejście to, mimo większych opóźnień podczas nawiązywania połączenia, jest znacznie lepszym wyborem w wypadku urządzeń charakteryzujących się znacznymi ograniczeniami energetycznymi.

Poza wymienionymi powyżej cechami, istnieje także szereg innych aspektów (jak użyte techniki optymalizacyjne, istnienie mechanizmu retransmisji bądź korekcji błędów itp.) które, w zależności od danego kontekstu, mogą być dodatkowo brane pod uwagę.

Każdy nowy, dodany do Frameworku protokół routingu musi być (manualnie) oceniony pod względem zachowania cech określonych w czynnikach kontekstowych (*CF*). W taki sposób tworzone są odpowiednio podzbiory protokołów routingu, charakteryzujące się zapewnieniem bezpieczeństwa transmisji, zachowaniem QoS, energooszczędnością itp. Dla danej listy

protokołów routingu R oraz listy czynników kontekstowych, każdy z protokołów zawartych w R oceniany jest za pomocą funkcji klasyfikującej σ:

$$\sigma: R \times CF \rightarrow \{4,3,2,1,0\}, \tag{7.7}$$

gdzie liczby 0-4 odpowiadają poziomowi spełnienia danego wymogu: odpowiednio 4 dla całkowitego spełnienia oraz 0 dla jego braku. Dodatkowo każdy protokół musi dostarczać zbiór wymagań (jeśli takie istnieją) niezbędnych do jego działania:

$$req := \{(cf, v): cf \in CF\}, \tag{7.8}$$

gdzie v jest minimalną wartością, którą kontekst Sieci, Użytkownika lub Dostawcy Usługi musi spełnić.

W kolejnym kroku, sklasyfikowane Portfolio Protokołów Routingu (*ERPP*) definiujemy jako:

$$ERPP: = \left\{(r, \pi, req): r \in R \wedge \pi = \left\{\bigcup_{cf \in CF}\left(cf, \sigma(r, cf)\right)\right\}\right\}, \tag{7.9}$$

W zaprojektowanym w ramach niniejszej pracy systemie wspomagającym transmisję czasu rzeczywistego zakłada się, iż wiele typów strumieni real-time może być transmitowanych jednocześnie z wykorzystaniem tej samej infrastruktury. A zatem różne protokoły routingu mogą być używane przez różne grupy użytkowników oraz dostawców usług. Dla każdej z takich grup, w sposób automatyczny, wybierany jest najbardziej odpowiedni protokół routingu. Wykorzystywana w tym celu jest zdefiniowana poniżej funkcja oceniająca rf.

Definicja funkcji oceniającej

Dla danej klasy użytkowników U', z ich kontekstem danym jako $UC' \subset UC$, niech:

- $SC' \subseteq SC$ będzie Kontestem Dostawcy Usługi (*SC*), określonym dla serwisów skorelowanych z U';
- *CF* będzie zbiorem czynników kontekstowych, względem których ocenione zostały protokoły routingu zawarte w portfolio;
- *NC* będzie kontekstem sieci, opisującym aktualny stan infrastruktury;
- $USNr$ będzie zbiorem wymagań określonych przez UC', SC' oraz NC;

- R będzie listą protokołów routingu zawartych w portfolio;
- $ERPP$ będzie Sklasyfikowanym Portfolio Protokołów Routingu zdefiniowanym zgodnie z (7.9);
- Wymagania protokołu routingu (dla każdego z protokołów z ERPP) dane jako req;
- $PP \in [0,1]^3$ będzie przyjętą polityką priorytetów, określających wagi istotności wymagań użytkowników, dostawców usług oraz Operatora Sieci.

Kolejno definiujemy Funkcję Oceniającą rf jako:

$$rf: UC' \times SC' \times CF \times NC \times ERPP \times R \times PP \to \mathbb{R} \cup \{null\}. \tag{7.10}$$

Funkcja rf działa w następujący sposób:

1) Biorąc pod uwagę aktualny stan infrastruktury dany przez *NC* oraz konkretne wymagania każdego z protokołów routingu (req) wyznaczany jest podzbiór możliwych do użycia protokołów routingu.
2) Każdy z protokołów wyznaczonych w kroku 1) oceniany jest poprzez porównanie czynników danych przez *ERPP* z czynnikami danymi w rekomendacji kontekstowej.

Prosty przykład funkcji rf przedstawiony został poniżej:

$$rf(UC', SC', CF, NC, ERPP, r, PP) =$$

$$\prod_{x \in req} p(x, UC' \cup SC' \cup NC) \prod_{y \in USNr} s(y, r, ERPP) \left(\alpha \sum_{i \in UC' \cap CF} q(\sigma(r,i), \tilde{\imath}) + \beta \sum_{j \in SC' \cap CF} q(\sigma(r,j), \tilde{\jmath}) + \gamma \sum_{k \in NC \cap CF} q\left(\sigma(r,k), \tilde{k}\right) \right), \tag{7.10}$$

gdzie $\alpha, \beta, \gamma \in PP, \sigma(r,i)$ jest wartością współczynnika $i \in CF$ dla protokołu r (uzyskaną z *ERPP*), natomiast $\tilde{\imath}, \tilde{\jmath}$ oraz $\tilde{k}$ są wartościami tego współczynnika odpowiednio dla UC', SC' oraz NC. Dodatkowo używamy dwóch prostych funkcji kontrolnych p i s:

$$p: req \times \{UC' \cup SC' \cup NC\} \to \{1, null\}, \tag{7.11}$$

funkcji p w celu sprawdzenia, czy wymogi protokołu routingu zostały spełnione, oraz funkcji s

$$s: USNr \times r \times ERPP - \{1, null\}. \tag{7.12}$$

sprawdzającej czy dany protokół routingu spełnia wymagania zdefiniowane przez użytkownika, dostawcę usługi oraz infrastrukturę sieciową. W kroku końcowym, funkcja q używana jest w celu oceny poziomu spełnienia danego współczynnika kontekstowego (z konkretnego protokołu routingu), w odniesieniu do zadanych rekomendacji kontekstowych. Aby złamać symetrię i zwiększyć wpływ niespełnionych rekomendacji na wynik końcowy, postanowiono dodatkowo mnożyć przez 2 wartości ujemne:

$$q(a,b) = \begin{cases} a-b & jeśli\ a \geq b \\ 2(a-b) & jeśli\ a < b \end{cases} \tag{7.13}$$

W zależności od przyjętej polityki, wybór protokołu routingu może być zorientowany na użytkowników ($\alpha = 1,\ \beta = 0,\ \gamma = 0$), lub być wybierany zgodnie z innymi priorytetami. Ostatecznie, wybranym protokołem routingu jest protokół maksymalizujący wartość funkcji rf po całej przestrzeni (portfolio) obsługiwanych protokołów. W wypadku braku możliwości wyboru konkretnego protokołu (brak spełnienia któregokolwiek z wymogów), możliwe jest przeprowadzenie renegocjacji wymogów, czyli np. zaniżanie poziomu bezpieczeństwa, wymogów względem QoS itp.

7.4.3 Blok optymalizacyjny

Zadaniem bloku optymalizacyjnego jest zarówno dostarczanie mechanizmów zwiększających jakość/niezawodność transmisji, jak i dodatkowych mechanizmów bezpieczeństwa.

W skład tych pierwszych wchodzi przede wszystkim mechanizm redystrybucji informacji o stanie sieci i trasach routingu. Korzystając tu z faktu posiadania centralnego serwera, jest on używany do gromadzenia informacji o stanie łącz, wyliczania nowych tras oraz rozsyłania w bezpieczny sposób wyznaczonych tablic routingu. Podejście to pozwala istotnie zredukować ilość danych sygnalizacyjnych wewnątrz sieci, osiągnąć szybszą zbieżność oraz szybszą reakcję na awarie/spadki jakości połączeń wewnątrz sieci. Jednocześnie, jeśli dany algorytm routingu posiada własny lokalny mechanizm wyboru trasy (np. autonomiczni agenci podejmujący indywidualną decyzję odnośnie partnera do komunikacji), może on ignorować informacje

nadsyłane przez centralny serwer. Ponadto, jeśli dany algorytm routingu dysponuje parametrami konfiguracyjnymi, jak np. wielkość bufora, czas do retransmisji itp., to blok ten udostępnia mechanizm pozwalający na przeprowadzanie automatycznej próby doboru najlepszego ustawienia takich parametrów. Po podaniu zakresu początkowego i końcowego danego parametru, stosowany jest algorytm zbliżony do wyszukiwania binarnego, gdzie poprzez obserwację wpływu zmiany parametru na zagregowaną jakość transmisji wyznaczana jest najlepsza konfiguracja.

W ramach bloku optymalizacyjnego możliwa jest także automatyczna obserwacja stanu sieci oraz, jeśli wymagane, zmiana przyjętej polityki priorytetów przy wyborze metody routingu. Zjawisko to może mieć miejsce np. w wypadku zaobserwowania nadmiernego obciążenia infrastruktury. W tej sytuacji możliwe jest zwiększenie wpływu kontekstu sieci (*NC*) w funkcji rf, a co za tym idzie wybranie protokołu routingu charakteryzującego się większym równoważeniem zasobów sieciowych (istotne szczególnie przy wielkoskalowym strumieniowaniu video). Niekiedy występuje także odwrotne zjawisko: jeśli obciążenie sieci jest niewielkie, a priorytet optymalizacji sieci duży, to poprzez jego zmniejszenie możliwy jest wybór algorytmu w lepszym stopniu spełniającego wymogi użytkowników.

W odniesieniu do bezpieczeństwa głównym zadaniem tego bloku jest obserwacja reputacji węzłów oraz odłączanie tych charakteryzujących się jej niskim poziomem. Węzły takie ignorowane są przy rozsyłaniu informacji o trasach routingu, a co za tym idzie, pomija się je w procesie routingu.

7.4.4 Mechanizmy wspomagające

W chwili obecnej we Frameworku przewiduje się wsparcie dla dwóch mechanizmów kompensujących straty pakietów w procesie routingu:

a) Mechanizm szybkiej retransmisji (FRM) [67] zaproponowany dla sieci P2P – może zostać użyty do odtworzenia pakietów utraconych podczas transmisji. W przeciwieństwie do relatywnie powolnego mechanizmu retransmisji używanego w TCP, retransmisja przy użyciu FRM jest znacznie szybsza, gdyż zachodzi między bezpośrednio komunikującymi się węzłami. Mechanizm ten działa poprzez wyposażenie każdego z węzłów w niewielki

bufor, gdzie przechowywane są ostatnio transmitowane pakiety. Podczas procesu routingu, każdy węzeł pracujący jako przekaźnik sprawdza numer sekwencyjny aktualnie routowanego pakietu i w wypadku nieprawidłowości prosi najbliższego bezpośrednio połączonego sąsiada o ponowne przesłanie zagubionego pakietu.

b) Kodowanie korekcyjne (FEC) – może być wykorzystane do wysyłania nadmiarowych danych i zminimalizowania problemów wywołanych zagubionymi pakietami [80]

7.4.5 Blok walidacyjny

Każda z jednostek zaangażowanych w transmisję (dostawca usługi, operator systemu oraz użytkownik końcowy) może ocenić działanie procesu routingu od swojej strony. Prezentowany Framework dostarcza dwa mechanizmy na potrzeby walidacji jakości transmisji

a) Automatyczny – informacje na temat wydajności procesu (obiektywne miary jakościowe) oraz informacje na temat ocen jednostek zaangażowanych w transmisję (oceny reputacyjne) są przesyłane automatycznie przez agentów do centralnego serwera.
b) Manualna – użytkownicy infrastruktury mogą dostarczać informacji zwrotnych na temat jakości usług, z których korzystają. W szczególności każde złowrogie działanie, które nie może być automatycznie wykryte, powinno zostać raportowane.

System reputacyjny używany jest do zbierania i odfiltrowania nieuczciwych głosów. Następnie, w zależności od przyjętej strategii, użytkownicy typu „premium", lub też dostawcy danej usługi mogą wymusić dokonanie zmian w stosowanej polityce routingu. W szczególności różne mechanizmy wsparcia, takie jak FRM lub FEC, mogą być automatycznie uruchamiane.

7.4.6 Przykładowa konfiguracja

Ze względu na swoją elastyczność, opisywany Framework może zostać zaadaptowany do szerokiego zakresu usług oraz różnych typów sieci. Poniżej przedstawiona została prosta

przykładowa konfiguracja. Na początku sześć różnych algorytmów routingu zostało wprowadzonych do Frameworku:

A. Prosty algorytm bazujący na wektorze odległości,
B. Algorytm Shortest Path First (SPF) [62],
C. QRON [81] – o działaniu podobnym do B. lecz wyposażony w dodatkowy mechanizm balansowania węzłów.
D. "1-800 overlay" [82] – protokół routingu zaprojektowany na potrzeby VoIP,
E. Protokół routing bazujący na rezerwacji przepustowości oraz priorytetowych klasach (zorientowany biznesowo).
F. Protokół routingu analogiczny do B. z dodatkowymi mechanizmami zorientowanymi na bezpieczeństwo transmisji, możliwością używania tylko zaufanych węzłów oraz mechanizmem tzw. anonimizacji (protokół Onion [83]).

Każdy z powyższych protokołów routingu został ręcznie oceniony (przypisano wartości w skali od 0 do 4) względem poniższej listy współczynników kontekstowych:

Tablica 7.3 Wyniki ewaluacji algorytmów routingu względem współczynników kontekstowych.

Współczynniki kontekstowe/alg. routingu	**A**	**B**	**C**	**D**	**E**	**F**
Model biznesowy	0	0	0	0	**4**	0
Obsługa QoS (jitter, delay, packet loss)	1,1,1	2,2,2	2,2,2	3,2,1	4,4,4	1,2,2
Koegzystencja z innymi protokołami	4	4	4	4	2	2
Poziom balansowania zasobami sieci	2	3	4	3	3	2
Wymogi związane z sygnalizacją (wyższa ocena – mniejsze wymogi)	3	2	2	1	2	2
Dostępne mechanizmy bezpieczeństwa (tylko zaufane węzły, anonimizacja)	1	2	2	2	3	4

Jedynym algorytmem, dla którego zdefiniowano dodatkowe wymagania kontekstowe, jest protokół "E". Ze względu na obsługę priorytetową cechuje się on znacznie niższym poziomem współpracy i koegzystencji z innymi protokołami routingu i w związku z tym jest on dostępny jedynie dla użytkowników typu „premium".

Założono istnienie pięciu różnych klas użytkowników systemu. Ich informacje kontekstowe zostały przedstawione w tablicy 7.3. Jedyną klasą definiującą wymagania kontekstowe stanowią użytkownicy wymagający bezpiecznego połączenia, pozostałe współczynniki kontekstowe traktowane są jako rekomendacje.

Tablica 7.4 Klasy użytkowników i ich współczynniki kontekstowe (zmapowane do skali 0-4).

User/Context	Security requirements	Business model	Energy scheme	QoS requirements
VoIP client normal account	2	0	0	3,2,1
VoIP client premium account	2	4	0	3,2,1
VoIP client secure connection	4	4	0	3,2,1
Video on Demand	1	0	0	2,2,2
Telemetry	3	0	0	1,2,4

Zakładamy ponadto listę rekomendacji kontekstowych zdefiniowanych przez operatora sieci jako: 0 dla modelu biznesowego, 2 dla koegzystencji oraz 2 dla balansowania obciążeniem i poziomem transmisji związanej z sygnalizacją.

Ostatecznie każdy z przedstawionych algorytmów routingu został automatycznie oceniony przy użyciu funkcji ewaluacyjnej 7.10. Jako politykę priorytetową przyjęto 0.8 dla użytkownika, 0.2 dla operatora sieci oraz 0 dla dostawcy usługi. Wyniki oceny zostały zaprezentowane w tablicy 7.5.

Tablica 7.5 Wyniki ewaluacji protokołów routingu dla danych klas użytkowników.

	A	B	C	D	E	F
VoIP normal	-5.6	-0.2	0	**0.2**	null	-0.8
VoIP premium	-5.6	-0.2	0	0.2	**6.4**	-0.8
VoIP secure	null	null	null	null	null	**-2.4**
Video on Demand	-	1.4	**1.6**	0.2	null	0.8
Telemetry	-4.2	-1.8	**-1.6**	-3	null	**-1.6**

Wybrane protokoły (cechujące się najwyższą oceną) zostały wyróżnione pogrubioną czcionką. W przypadku gdy więcej niż jeden protokół osiągnie tę samą ocenę, algorytm wybierany jest losowo spośród nich. Na podstawie przeprowadzonej powyżej prostej symulacji widać więc wyraźnie, że bazując na informacji kontekstowej oraz używając funkcji rankingowej do oceny, można dokonać wyboru optymalnych algorytmów dla każdej z różnych klas użytkowników.

7.5 Bezpieczeństwo routingu

Istnieje szereg ataków, które mogą zostać przeprowadzone w celu przechwycenia transmisji lub destabilizacji działania sieci. Wśród nich wyróżnić można dwie podstawowe grupy:

- ataki pasywne, podczas których atakujący ogranicza się jedynie do podsłuchiwania transmisji, nie prowadząc przy tym żadnych szkodliwych działań,
- ataki aktywne, cechujące się prowadzeniem przez atakującego kroków mających na celu destabilizację pracy systemu. Do tego typu działań należą próby modyfikacji tras routingu w celu przechwycenia transmisji, celowe nieprzekazywanie pakietów (tzw. black hole), czy też generowanie wzmożonego ruchu związanego z sygnalizacją (np. poprzez przesyłanie błędnych tablic routingu). Szerszy opis częstych ataków na mechanizm routingu został przedstawiony w [84].

W przypadku Frameworku opracowanego w ramach niniejszej pracy, zaproponowane zostały dwa mechanizmy możliwe do zastosowania w celu minimalizacji zagrożenia związanego z działalnością wadliwych/złośliwych węzłów w sieci.

Pierwszy z nich opiera się na użyciu standardowych metod kryptograficznych (tzw. hard security). W szczególności, aby zapobiec ingerencji w trasy routingu, przyjęto, iż dane sygnalizacyjne podpisywane są kluczem prywatnym serwera centralnego. Także transmisja danych między węzłami jest szyfrowana przy użyciu strumieniowej wersji algorytmu AES (tzw. counter mode). Podejście to uniemożliwia przeprowadzenie ataku pasywnego oraz ingerencję w trasy routingu przez atakującego nie będącego członkiem systemu.

Mechanizm drugi bazuje natomiast na metodach soft-security, a w szczególności na reputacji. Został on zaimplementowany, aby chronić system przed złośliwymi lub wadliwymi

węzłami, które stanowią elementy składowe infrastruktury P2P, a przez to metody typu hard-security mogą okazać się w ich przypadku nieskuteczne. Jak już wspomniano w rozdziale poświęconym reputacji, po każdej interakcji bezpośrednio zaangażowane w nią węzły oceniają drugą ze stron pod względem poprawności jej działania. Scentralizowany system reputacyjny gromadzi oraz przetwarza nadesłane opinie, a następnie wylicza współczynnik reputacji dla każdego z węzłów. Obliczone w taki sposób współczynniki brane są pod uwagę w procesie wyznaczania tras routingu, co pozwala na wykluczenie węzłów o niskiej reputacji. Również dane sygnalizacyjne nadesłane przez takie węzły są ignorowane. Reputacja używana jest także do detekcji „czarnych dziur" na trasach routingu, co zostało przedstawione w formie symulacji w podrozdziale 9.3.

Ostatecznie, istnieje szereg możliwości rozbudowy Frameworku o dodatkowe mechanizmy bezpieczeństwa. W szczególności warte rozważenia jest dodanie mechanizmów wykrywania anomalii [85].

Rozdział VIII

Mechanizm odtwarzania serwera centralnego

8.1 Wstęp

Zgodnie z porównaniem przedstawionym w podrozdziale 5.2, każda z różnych topologii sieci P2P posiada zarówno wady jak i zalety. Aby uniknąć konieczności wyboru pomiędzy szybkością wyszukiwania a bezawaryjnością i odpornością na ataki, autor niniejszej pracy postanowił opracować własny typ infrastruktury P2P. Będzie on w sobie łączyć szybki i niezawodny mechanizm wyszukiwania oraz zarządzania użytkownikami, charakterystyczny dla struktur scentralizowanych, z wysokim współczynnikiem odporności na awarie oraz próby ataków będący cechą infrastruktury w pełni rozproszonej [86]. Ponadto opracowany mechanizm może być w prosty sposób zaadaptowany w już istniejących systemach, działających w oparciu o sieci P2P z serwerem centralnym – dzięki temu ich poziom bezpieczeństwa istotnie wzrośnie przy jednoczesnym zachowaniu wcześniejszej wydajności.

Główna idea działania sytemu odtwarzania centralnego serwera opiera się na połączniu trzech mechanizmów odpowiedzialnych odpowiednio za: obserwację i wykrywanie wadliwej pracy centralnego serwera, przeprowadzenie procesu szybkiego rozproszonego głosowania mającego na celu wyłonienie nowego centralnego serwera oraz odbudowę centralnej bazy danych. Idea działania oraz główne cechy algorytmu zostały opisane w kolejnym podrozdziale. Natomiast podrozdziały 8.3-8.5 zostały poświęcone szczegółowemu opisowi działania jego poszczególnych modułów.

8.2 Idea działania systemu odtwarzania centralnego serwera

System został przeznaczony do pracy w sieciach P2P dodatkowo wyposażonych we wspomagający serwer centralny, w którym mogą znajdować się dane kluczowe dla poprawnej pracy infrastruktury (jak informacje o użytkownikach i dane niezbędne do poprawnego procesu logowania). Przyjęte założenia sprawiają, iż mechanizm odtwarzania ma na celu nie tylko wykrycie błędnego zachowania aktualnego serwera centralnego i wyznaczenie nowego, ale także zapewnienie bezpiecznego sposobu na przechowywanie kopii zapasowych danych krytycznych dla pracy systemu, jak również ich ciągłą synchronizację oraz, w razie potrzeby, szybkie odtworzenie. Zadania te realizowane są przez trzy niezależnie mechanizmy:

1) **Mechanizm detekcji błędnego zachowania centralnego serwera.**

Nieprawidłowości związane z funkcjonowaniem centralnego serwera zostały podzielone na dwie kategorie:

a) awarie natury fizycznej związane z danym serwerem lub siecią, do której jest przyłączony
b) nieprawidłowe działanie lub luki bezpieczeństwa powstałe w wyniku błędów oprogramowania, infekcji złośliwym kodem lub będące wynikiem ataku na infrastrukturę (np. atak DDOS).

Awarie pierwszego typu są relatywnie proste do wykrycia w wyniku obserwacji jakości połączenia z centralnym serwerem, gdyż ich skutki objawiają się niedostępnością danej usługi lub znaczną degradacją jej jakości. Znacznie większym wyzwaniem może być natomiast prawidłowa detekcja problemów mających źródło w błędach oprogramowania lub kompromitacji serwera i przejęcia nad nim kontroli w wyniku przeprowadzonego ataku. Następstwa tego typu nieprawidłowości mogą również być groźniejsze, gdyż poza ewentualnym paraliżem usługi może dojść do kradzieży poufnych danych zgromadzonych na serwerze oraz wciąż nadsyłanych przez nieświadomych zagrożenia użytkowników. W celu wykrycia tego typu nieprawidłowości, mechanizm detekcji postanowiono wyposażyć w system bezpieczeństwa bazujący na zaufaniu. Zasada działania polega na wyznaczeniu grupy zaufanych serwerów zwanych Notariatami.

W zależności od przyjętego sposobu implementacji, są to serwery zewnętrzne lub wybrane spośród zaufanych węzłów P2P wchodzących w skład infrastruktury. Zadaniem Notariatów jest gromadzenie informacji społecznych (m.in. raportów nadsyłanych przez użytkowników systemu) odnośnie zachowania centralnego serwera i, w oparciu o tak zgromadzoną wiedzę, wyznaczanie w sposób kolektywny aktualnego poziomu zaufania względem centralnego serwera. Jeśli zaufanie to spadnie poniżej pierwotnie ustalonego progu, automatycznie wyznaczany jest nowy serwer centralny.

2) Mechanizm wyboru nowego serwera

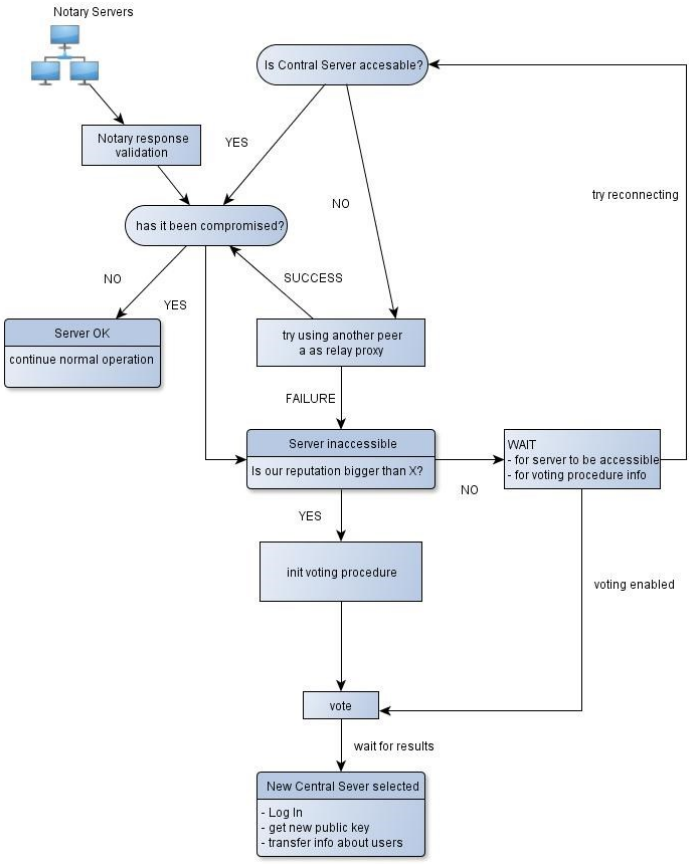
W wypadku wykrycia awarii, bądź stwierdzenia kompromitacji obecnego serwera, nowy centralny serwer jest wyłaniany spośród węzłów tworzących sieć P2P. Na tym etapie istotne jest, aby wybór został dokonany w możliwie krótkim czasie oraz aby nowy serwer spełniał wymagania stawiane mu przez pozostałe elementy infrastruktury (w szczególności szybki czas dostępu). W celu wyznaczenia nowego serwera w sposób obiektywny oraz optymalny z punktu widzenia większości uczestników systemu, zadecydowano o wyłonieniu go w procesie automatycznego rozproszonego głosowania. Pierwotna pula kandydatów wybierana jest na podstawie analizy danych pochodzących z systemu reputacyjnego, także proces przydzielania głosów jest oparty o reputację, tak aby agenci o wyższej ocenie mieli istotniejszy wpływ na końcowy wynik głosowania. Szczegóły działania algorytmu głosowania zostały opisane w podrozdziale 8.3.3.

3) Odbudowa bazy danych

Jedną z pierwszych czynności, jakie musi wykonać nowo wybrany centralny serwer, jest odbudowa bazy zawierającej dane kluczowe dla pracy infrastruktury (przede wszystkim dane użytkowników, takie jak login, hasło i informacje o koncie). Dane używane do odbudowy bazy są rozproszone pomiędzy węzłami sieci P2P i przechowywane w zaszyfrowanej postaci. Proces aktualizacji danych odbywa się automatycznie podczas logowania się użytkownika do systemu. W zależności od typu infrastruktury oraz rozległości wcześniejszej awarii, podczas procesu scalania informacji możliwe jest pozyskanie i odtworzenie także innych danych aktualnie

dostępnych w poszczególnych węzłach (np. w wypadku usługi strumieniowania danych audio/video część kolekcji może zostać odtworzona na podstawie danych znajdujących się w buforach poszczególnych węzłów).

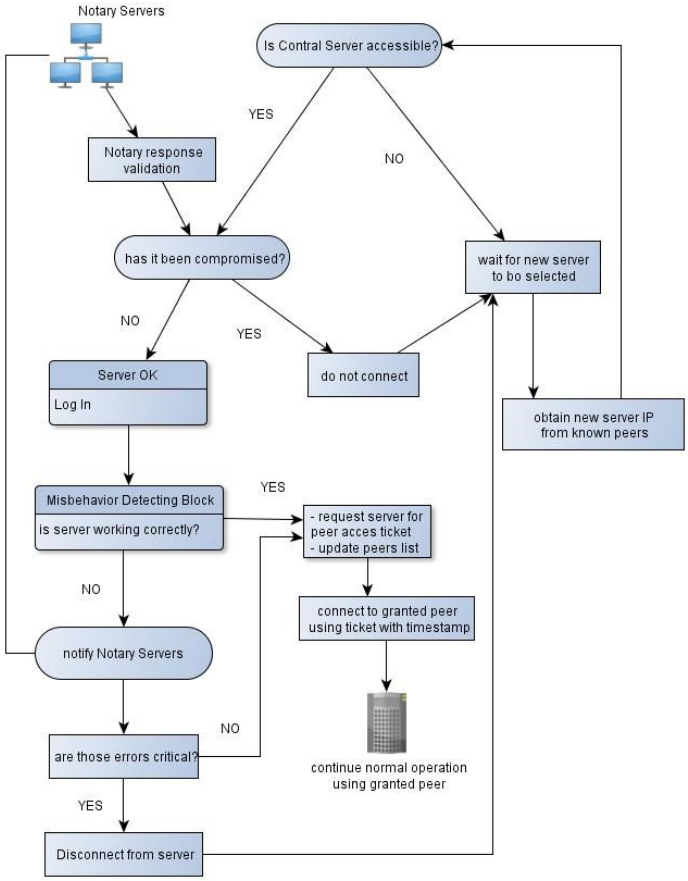
Aby system mógł działać poprawie, wymagane było opracowanie szczegółowego schematu postępowania w wypadku wykrycia awarii centralnego serwera. Schematy takie zostały zaprojektowane zarówno dla agentów zlokalizowanych w węzłach sieci P2P, jak i dla końcowych użytkowników danego serwisu (np. aplikacji VoIP zainstalowanej po stronie klienta). Na rysunku 8.1 przedstawiony został schemat działania, opracowany dla agentów umiejscowionych w węzłach sieci P2P, obrazujący przebieg wykonywanych akcji w wypadku awarii centralnego serwera.



Rys. 8.1 Mechanizm odtwarzania serwera centralnego – schemat blokowy mechanizmu po stronie serwera.

Ponieważ każdy agent może obserwować zachowanie centralnego serwera, w wypadku wykrycia związanych z nim awarii, ma on możliwość zainicjowania procesu odtwarzania.

Wymogiem jest jednak, aby agent inicjujący rozproszone głosowanie posiadał wysoką reputację. W zależności więc od swojej oceny reputacyjnej, może sam być inicjatorem zmiany lub przejść w tryb oczekiwania na zgłoszenie błędu przez innego agenta. Aby uniknąć prób destabilizacji infrastruktury poprzez ciągłe zmiany nowego serwera, przyjęto, że wszelkie nieuzasadnione próby inicjacji mechanizmu wyboru nowego serwera (np. jeśli aktualny serwer działa poprawnie i ma wysoki współczynnik zaufania) kończą się przerwaniem procedury głosowania oraz ukaraniem inicjującego agenta poprzez zaniżenie jego reputacji. Dane odnośnie reputacji agentów dostarczane są w wyniku pracy systemu reputacyjnego opisanego w rozdziale VI.



Rys. 8.2 Mechanizm odtwarzania serwera centralnego – schemat blokowy mechanizmu po stronie użytkownika.

Diagram działania agentów końcowych powiązanych z aplikacją użytkownika (user-agent) przedstawiony został na rysunku 8.2. W wypadku użytkowników końcowych szczególny nacisk położony został na kwestie bezpieczeństwa i ochrony prywatnych danych. Dlatego też,

przed każdorazowym rozpoczęciem procesu logowania, user-agent łączy się z Notariatami i pobiera aktualną wartość współczynnika zaufania centralnego serwera. W przypadku pozytywnej weryfikacji user-agent loguje się na serwer i pobiera bilet uprawniający go do połączenia się z przydzielonym mu węzłem sieci P2P. Na użytkowniku końcowym spoczywa także zadanie raportowania o wszelkich nieprawidłowościach związanych z pracą centralnego serwera (podejrzane zachowania, jak np. zmiana polityki bezpieczeństwa, monity o podanie dodatkowych danych itp.).

Aby zapewnić wysoki poziom bezpieczeństwa, poza szyfrowaniem połączeń, jest przeprowadzane obustronne uwierzytelnienie. Pozwala to uniknąć problemów związanych z podmianą serwera centralnego – np. w wyniku ataku man-in-the-middle, który w przypadku braku odpowiednich zabezpieczeń infrastruktury może być w prosty sposób przeprowadzony wewnątrz sieci LAN, przy wykorzystaniu techniki ARP spoofing [87]. Powszechnie stosowanym sposobem uwierzytelnienia jest użycie certyfikatów podpisanych przez PKI (Public Key Infrastructure) lub certyfikatu podpisanego samodzielnie (tzw. self signed). Rozwiązanie to ma jednak istotne wady, które sprawiają, iż jego implementacja może być problematyczna w przypadku przyjętego modelu P2P, w którym klucz publiczny serwera może się często zmieniać (m.in. wraz z wyborem kolejnego serwera). Utrzymanie infrastruktury PKI jest zazwyczaj bardzo kosztowne, natomiast rozwiązanie polegające na podpisywaniu certyfikatów samodzielnie jest z kolei wrażliwe na ataki sieciowe i naraża użytkowników na ryzyko interakcji z niezaufanym serwerem. W niniejszej pracy problem ten postanowiono rozwiązać przy wykorzystaniu zdefiniowanych pierwotnie Notariatów, czyli grupy zaufanych węzłów sieci lub niezależnych serwerów zewnętrznych. Poza zbieraniem i przetwarzaniem danych odnośnie pracy centralnego serwera, Notariaty obserwują zmiany klucza publicznego centralnego serwera w czasie, w rożnych lokalizacjach sieci. Pozostali użytkownicy infrastruktury P2P mogą pobrać tak utworzone logi, a następnie porównać je z kluczem przedstawionym przez aktualny serwer centralny, co pozwala wykryć wiele typowych ataków. Szczegółowy opis oraz przykładowa implementacja mechanizmu śledzenia zmian klucza przez Notariaty można znaleźć w pracy [88].

8.3 Proces rozproszonego głosowania

8.3.1 Definicja wymogów

Algorytm głosowania jest jednym z kluczowych elementów mechanizmu odtwarzania centralnego serwera. Sam proces głosowania jest zagadnieniem bardzo często omawianym, a różne algorytmy głosowania używane są w bardzo wielu dziedzinach. Standardowym przykładem jest podejmowanie różnych decyzji społecznych, jak np. wybory polityczne lub głosowanie akcjonariuszy w korporacjach. Algorytm ten jest przydatny również przy rozwiązywaniu problemów technicznych, takich jak analiza wyników pozyskanych z rozproszonych sieci sensorowych, gdzie różniące się od siebie, niejednokrotnie niekompletne lub błędne wyniki muszą być zagregowane. Także rozproszone systemy komputerowe często bazują na procesie głosowania, gdzie przed przeprowadzeniem niektórych rozproszonych transakcji musi być uprzednio osiągnięte kworum [89]. Proces głosowania jest także istotnym elementem systemów niezawodnościowych, gdzie redundantne systemy kontrolne używają go, aby zmniejszyć wpływ nieprawidłowo działających modułów (rozwiązania stosowane m.in. w statkach kosmicznych projektowanych przez NASA). Natomiast w tym podrozdziale zostało opisane, jak proces głosowania może być wykorzystany w celu wyznaczenia nowego serwera centralnego.

Procesy głosowania można sklasyfikować na wiele sposobów. Do podstawowych z nich należy uwzględnienie takich aspektów jak: rodzaj użytego algorytmu zliczania głosów, liczba niezbędnych do przeprowadzenia rund w celu osiągnięcia końcowego rezultatu oraz uzyskany w ten sposób wynik (dokładny lub przybliżony).

Biorąc pod uwagę założenia projektowe mechanizmu odtwarzania centralnego serwera, zdefiniowano szereg wymogów względem algorytmu głosowania:

a) Decentralizacja – ponieważ w chwili awarii centralny serwer nie jest dostępny, brak jest głównej jednostki przetwarzającej, która mogłaby gromadzić głosy. Wymagane jest zatem, aby każdy agent w sieci P2P zliczał głosy niezależnie we własnym zakresie. Biorąc pod uwagę infrastrukturę P2P, w ramach której odbywa się wymiana głosów, możliwa jest sytuacja pojawienia się nieścisłości wyników - nie każdy agent otrzyma identyczny wynik głosowania, jednakże konieczne jest, aby zdecydowana większość agentów wybrała tego samego kandydata.

b) Szybkość – wynik powinien być ustalany w pojedynczej rundzie.

c) Odporność na błędy/próby oszustw – mała grupa wadliwych/złośliwych agentów nie powinna mieć możliwości ponadprzeciętnego wpływu na końcowy wynik głosowania.

d) Niejednolitość w potencjale wyborczym – agenci z dobrą reputacją i dłuższym stażem powinni mieć większy wpływ na wynik głosowania niż nowi członkowie sieci.

e) Bezstronność i rzetelność – węzeł wybrany do roli centralnego serwera nie powinien faworyzować jednej grupy użytkowników kosztem innej lecz dokonywać optymalnego wyboru mając na uwadze ogólną wydajność infrastruktury.

Należy zauważyć, iż głosowanie jest krytycznym elementem całego procesu odtwarzania centralnego serwera. Jeśli podczas tego procesu pojawią się nieprawidłowości, mogą one być groźne w konsekwencjach: prowadzić do awarii całej infrastruktury P2P i powodować poważne luki w mechanizmie bezpieczeństwa. W ekstremalnych przypadkach możliwa jest sytuacja, gdy koalicja złośliwych węzłów będzie w stanie przejąć kontrolę nad infrastrukturą – np. w sytuacji, gdy atakującym uda się stworzyć dużą liczbę sztucznych bytów i z ich pomocą zyskać znaczny wpływ na wynik (analogicznie jak w wypadku ataku sybil na system reputacyjny). Jednakże w przypadku systemu opracowanego w niniejszej pracy, poza standardowymi mechanizmami bezpieczeństwa opisanymi w części poświęconej reputacji, kolejną linię obrony stanowią Notariaty. W sytuacji gdy nowy serwer zacznie wykazywać oznaki nieprawidłowej pracy, zachowanie to zostanie wykryte, a sam proces głosowania powtórzony. Notariaty, mimo iż chronią przed przejęciem kontroli nad infrastrukturą, nie eliminują jednak zagrożenia związanego z kradzieżą danych przez nowo wybrany serwer – stąd też sam proces głosowania powinien być przeprowadzany w taki sposób, aby minimalizować ryzyko wyboru wadliwego lub zwodniczego serwera centralnego.

Kluczowe elementy użytego algorytmu głosowania zostały omówione w dwóch kolejnych podrozdziałach.

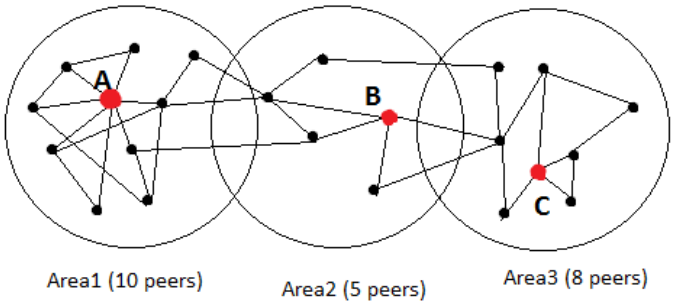
8.3.2 Wybór kandydatów oraz przydzielanie wag głosującym

Ponieważ system do wspomagania transmisji czasu rzeczywistego, zrealizowany w ramach niniejszej pracy doktorskiej, zawiera jako element składowy system reputacyjny, postanowiono użyć ocen reputacyjnych także do wspomagania procesu wyboru nowego serwera centralnego. Jak zostało omówione w rozdziale VIII, zaprojektowany system reputacyjny oparty jest na scentralizowanym modelu, a więc każdy agent w infrastrukturze powinien posiadać identyczne dane na temat reputacji innych agentów (z dokładnością do czasu synchronizacji). Taka spójność ocen pozwala na jednoznaczny wybór ograniczonej liczby kandydatów na nowy serwer, poprzez wybór niewielkiej grupy L węzłów, charakteryzujących się najwyższą reputacją. Podejście takie pozwala na minimalizację rywalizacji między węzłami (agenci nie będą wybierać siebie samych) oraz sprawniejsze wyznaczenie kandydata odpowiadającego wymogom większości głosujących. W obecnej implementacji jako liczbę kandydatów przyjęto $L = 10\%$ najwyżej ocenianych węzłów.

Kolejną zaletą użycia systemu reputacyjnego jest możliwość rozróżnienia pomiędzy węzłami z większym stażem i dobrą historią współpracy, a węzłami relatywnie nowymi lub mogącymi przesyłać nieuczciwe/błędne informacje. W tym przypadku do weryfikacji głosujących użyta została reputacja rekomendacyjna, czyli ich zdolność do uczciwej oceny innych. Natomiast aby zdefiniować grupę kandydatów, posłużono się ogólną oceną reputacyjną. Bazując na reputacji rekomendacyjnej R_{REC} możliwe jest nierównomierne rozdysponowanie liczby możliwych do oddania głosów pomiędzy agentów wchodzących w skład infrastruktury. W ten sposób dla N głosujących generowany jest wektor wag $W = [w_1, w_2, \dots, w_N]$, gdzie $w_i = f(R_{REC\,i})$ jest wagą głosu agenta i-tego, natomiast $f(R_{REC}) \rightarrow \mathbb{R}$ funkcją normalizującą. Liczba przyznanych głosów zależy silnie od rozmiaru infrastruktury oraz rozkładu ocen reputacji rekomendacyjnej R_{REC}. Wymagane jest ponadto, aby zapobiegać możliwości pojawienia się dyktatorów (agentów z wagą głosu pozwalającą samodzielnie decydować o wyniku głosowania). Aby sprostać tym wymogom, w obecnej implementacji jako $f(R_{REC})$ użyto prostej funkcji progowej zwracającej liczby naturalne z przedziału $[0, \dots, \frac{(N-1)(N-1)}{2N}]$.

8.3.3 Protokół głosowania

Zgodnie z listą wymagań (rozdział 8.1) stawianych przez mechanizm odtwarzania serwera algorytmowi głosowania, jednym z najistotniejszych warunków do spełnienia jest szybkość procesu uzgadniania decyzji. Ponieważ transmisja danych między agentami wprowadza istotne opóźnienia, preferowane są algorytmy jednorundowe. Ograniczenie to eliminuje więc standardowe metody absolutnej większości, często używane w wypadku głosowania z wagami, ze względu na trudną do osiągnięcia jednomyślność większości głosujących w pojedynczej turze. Kolejna prosta i popularna metoda głosowania opiera się na względnej większości, gdzie wybrany zostaje kandydat, który otrzymał najwyższą liczbę głosów (lecz niekoniecznie większość wszystkich oddanych). Metoda ta charakteryzuje się szybkością oraz prostotą w implementacji, jednak również nie nadaje się ona do aplikacji w omawianym zagadnieniu wyboru centralnego serwera, gdyż wyznaczony na jej podstawie zwycięzca nie musi być optymalny z punktu widzenia całej infrastruktury. Aby to udowodnić, rozważmy prosty przykład sieci P2P składającej się z 23 węzłów. Załóżmy, iż na podstawie ich lokalizacji geograficznej można podzielić je na 3 podgrupy: strefa 1, 2 i 3 z odpowiednio 10, 5 i 8 węzłami, jak zostało przedstawione na rysunku 8.3.



Rys. 8.3 Przykładowa sieć P2P podzielona na 3 podgrupy.

Załóżmy także, iż mamy trzech kandydatów do roli nowego centralnego serwera, oznaczonych odpowiednio na rysunku A, B i C. Ponadto przyjmijmy zadane średnie czasy opóźnień w transmisji pomiędzy poszczególnymi kandydatami, a węzłami poszczególnych stref (tablica 8.1).

Tablica 8.1. Przykładowe opóźnienia kandydatów względem stref.

	Area1	Area2	Area3
A	<20ms	<50ms	<100ms
B	<50ms	<20ms	<50ms
C	<100ms	<50ms	<20ms

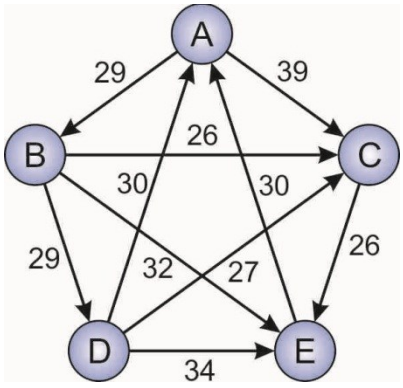
Ponieważ czas dostępu do serwera jest jednym z najistotniejszych kryteriów wyboru, możemy założyć, że każdy węzeł odda swój głos na najbliższego kandydata, tj. o najniższym czasie dostępu. Użycie w tym wypadku algorytmu względnej większości doprowadzi do wyboru kandydata A jako nowego centralnego serwera. Jak można jednak łatwo zauważyć, z punktu widzenia całej infrastruktury wybór ten nie będzie optymalny, gdyż zaowocuje on dużymi (czasem nieakceptowalnymi) opóźnieniami transmisji pomiędzy nowym serwerem a węzłami wewnątrz strefy 3. Z tego względu lepszym wyborem byłby kandydat B. Aby lepiej oszacować jakość wybranego kandydata, postanowiono posłużyć się kryterium Condorceta [90], mówiącym iż końcowym zwycięzcą (zwanym też „zwycięzcą według Condorceta") jest kandydat, który w zestawieniu ze wszystkimi innymi kandydatami preferowany jest przez większość. Algorytmy głosowania oparte na tym kryterium są znacznie lepsze w dokonywaniu optymalnego wyboru od wspomnianych wcześniej innych metod głosowania, gdyż dzięki niemu zwycięzca będzie stanowił najlepszy wybór dla większości głosujących i spełni wymóg optymalnego wyboru z punktu widzenia całej infrastruktury (w powyższym przykładzie zwycięzcą Condorceta zostałby kandydat B). Należy jednak zauważyć, iż głosowanie metodą Condorceta posiada także pewne wady: przede wszystkim nie bierze pod uwagę szeregu problemów związanych z istnieniem koalicji. Co więcej, istnieją przypadki, gdy dla danego zestawu głosujących zwycięzca w sensie Condorceta nie istnieje. Drugi z problemów, zwany także paradoksem głosowania, ma miejsce gdy relacja preferencji danego kandydata przez większość nie jest przechodnia (z ARB oraz BRC nie wynika ARC, czyli preferencje danych grup wyborców tworzą cykl). Jednakże na potrzeby systemu opracowanego w niniejszej pracy, przyjęto, iż jeśli koalicje istnieją, to z dużym prawdopodobieństwem zostaną one wykryte przez mechanizm opisany w rozdziale 6.7 i ich członkowie dostaną niską wagę głosu. Natomiast w ramach algorytmu głosowania postanowiono opracować hybrydowy mechanizm, który bazuje na metodzie Condorceta jako podstawowej, a w wypadku niemożliwości wyznaczenia zwycięzcy, używany jest algorytm pomocniczy – oparty na metodzie rankingu Borda [91]. Jako algorytm realizujący

metodę głosowania Condorceta, ze względu na prostotę w implementacji, postanowiono wykorzystać metodę Markusa Schulzego [90] opracowaną w 1997 roku. W metodzie tej każdy głosujący ocenia każdego z kandydatów zgodnie ze swoimi preferencjami, tworząc w ten sposób listę rankingową $v_i \in S(\{c_1, \dots, c_L\})$, gdzie S jest zbiorem wszystkich permutacji na zbiorze kandydatów oraz dla każdego $a, b : 1 \geq a > b \leq L$ kandydat $v_i(a)$ jest preferowany nad kandydatem $v_i(b)$. System zliczający głosy, bazując na utworzonych listach rankingowych, wylicza indeks m, gdzie $m(c_i, c_j)$ jest liczbą głosujących preferujących kandydata c_i nad c_j. Niech ścieżka P o sile s od kandydata c_i do kandydata c_j będzie zdefiniowana następująco:

$$\forall_{k\in(i,\dots,j-1)}\ m(c_k, c_{k+1}) > m(c_{k+1}, c_k) \tag{8.1}$$

$$\forall_{k\in(i,\dots,j-1)}\ m(c_k, c_{k+1}) \geq s \tag{8.2}$$

Natomiast $P_s(c_i, c_j) = \max\left(0, P(c_i, c_j)\right)$ niech będzie najsilniejszą ścieżką z c_i do c_j. Mówimy, że kandydat c_i jest lepszy od kandydata c_j, gdy zachodzi relacja $P_s(c_i, c_j) > P_s(c_j, c_i)$. Ponadto, kandydat c_w jest potencjalnym zwycięzcą, gdy $\forall_{k\in(1,\dots,L)}\ P_s(c_w, c_k) > P_s(c_k, c_w)$. W celu wyznaczenia siły ścieżek P_s tworzony jest graf skierowany $G(V, E)$, gdzie $V = (c_1, \dots, c_L)$ oraz $E \subseteq \left(\{u, v\} : u, v \in V\ \ i\ \ 0 < m(u, v) > m(v, u)\right)$. Przykładowy przebieg głosowania z 5 kandydatami oraz 50 oddanymi głosami pokazany jest na rysunku 8.4.



5	ABCDE
7	ACBED
3	ADECB
2	BAEDC
15	BDEAC
8	CDEAB
3	CEADB
3	DACBE
4	EDBAC

Rys. 8.4 Przykładowy graf głosowania dla 5 kandydatów i 50 oddanych głosów.

Dla procesu głosowania zaprezentowanego w formie grafu najsilniejsza ścieżka pomiędzy każdą parą kandydatów $P_s(u, v)$ możliwa jest do wyznaczenia przy pomocy algorytmu szukania

najkrótszej ścieżki w grafie. W obecnej implementacji systemu użyto algorytmu Floyda-Warshalla o złożoności $\theta(n^3)$. Zwycięzca głosowania jest następnie wyznaczony poprzez wzajemne porównywanie siły ich najsilniejszych ścieżek.

8.4 Schemat procesu wyboru nowego serwera centralnego.

Proces wyznaczania nowego centralnego serwera wiąże się z opracowaniem schematu rozproszonego głosowania bazującego na mechanizmach opisanych w poprzednich podrozdziałach (wyznaczania kandydatów oraz przydziału wag głosującym), prostym mechanizmie korekcji błędów oraz algorytmie głosowania według metody Schulzego. Główne kroki procesu opisane zostały poniżej:

1) **Wybór kandydatów** – każdy agent p_i, bazując na danych z systemu reputacyjnego, wybiera L węzłów charakteryzujących się najwyższą reputacją, a następnie ocenia każdego z nich tworząc listę rankingową $v_i = [c_1, \dots, c_L]$ będącą jednocześnie jego głosem.
2) **Broadcast** – głosy rozsyłane są do każdego węzła w sieci.
3) **Gromadzenie danych** – w ciągu zdefiniowanego przedziału czasu każdy agent gromadzi otrzymane w wyniku broadcastu głosy, tworząc listę $V_i = [v_1, \dots, v_N]$.
4) **Walidacja danych wejściowych** – każdy z agentów przegląda swoją listę V_i pod kątem błędnych danych. Jeśli takie napotka, podmienia w ich miejsce swój własny głos v_i. Jeśli w liście brakuje danych od któregoś z głosujących, dane te są także uzupełniane poprzez wstawienie własnego głosu v_i.
5) **Zliczanie głosów:**
   a) Uwzględniając wagi poszczególnych głosujących, wyznaczony zostaje indeks m, zgodnie ze schematem:

```
for(k=1; k<=N; k++)
    for(i=1; i<=L-1; i++)
        for(j=i+1; j<=L; j++)
            m(V[k][i],V[k][j]) = m(V[k][i],V[k][j]) + f(R_REP(k));
```

   b) Zbudowany zostaje graf głosowania G.

c) Zmodyfikowana wersja algorytmu Floyda-Warshalla zostaje użyta do wyznaczenia najsilniejszych ścieżek (P_s):

```
for (i=1; i<=L; i++)
    for (j=1; j<=L; j++)
        if (i!=j)
            if ( m[i,j] > m[j,i] ) p_s[i,j]=m[i,j]
            else p_s[i,j]=0;
for (k=1; k<=L; k++)
    for (i=1; i<=L; i++)
        for (j=1; j<=L;j++)
            if ( (k!=j) && (i!=j))
            p_s[i,j]= max (p_s[i,j], min(p_s[i,k],p_s[k,j]));
```

d) Zwycięzca wyznaczany jest w wyniku porównywania wyznaczonych we wcześniejszym kroku najsilniejszych ścieżek. Jeśli zaistniał cykl w preferencjach głosujących i zwycięzca jest niemożliwy do ustalenia, głosowanie ponawiane jest z użyciem metody Borda.

e) Nowy centralny serwer został wyznaczony. Notariaty zostają zaktualizowane, a serwer przystępuje do fazy odbudowy bazy danych.

Jeśli zdarzyło się, że dany agent lokalnie wyznaczył innego zwycięzcę niż większość (np. w wyniku błędów przy transmisji rozsyłanych głosów, odmiennych ocen reputacyjnych lub innych błędów) może połączyć się on z Notariatami i pobrać adres nowego serwera.

8.5 Odbudowa bazy danych

Odbudowa danych krytycznych dla poprawnej pracy systemu jest drugim obok wyboru nowego serwera, fundamentalnym elementem działania Modułu Odbudowy centralnego serwera. Zanim dany serwis P2P odzyska pełną sprawność po awarii/kompromitacji/ataku na centralny serwer, dane takie jak konta użytkowników (loginy, hasła, stan konta itp.) muszą zostać przywrócone na nowo wybranym centralnym serwerze. W zależności od typu serwisu, także dodatkowe dane, jak informacje konfiguracyjne, adresy zasobów, itp. mogą wymagać odtworzenia. Najpowszechniejszą metodą ochrony przed utratą danych jest tworzenie kopii zapasowych i przechowywanie ich na innych, wybranych serwerach. Jednakże rozwiązanie to

minimalizuje jedynie ryzyko utraty danych, a ponadto tworzy dodatkowe problemy związane z koniecznością ich synchronizacji. Ponadto koszt utrzymywania dedykowanych bezpiecznych serwerów w wielu różnych obszarach geograficznych (na wypadek awarii sieci w jednym z centrów gromadzących dane) może okazać się znaczący. Jednocześnie w wyniku większego rozproszonego ataku na infrastrukturę serwisu, celem ataku mogą być również serwery z kopiami bezpieczeństwa, co doprowadzi do paraliżu danego serwisu. Ostatecznie, rozwiązanie bazujące na serwerach z kopiami bezpieczeństwa wymaga wcześniejszej konfiguracji, co eliminuje jedną z głównych zalet użycia infrastruktury P2P – zdolności do samoorganizacji i autonomicznego działania. Infrastruktury pozbawione konkretnych serwerów predefiniowanych do gromadzenia kopii zapasowych są znacznie bardziej odporne na awarie, ataki oraz próby blokowania dostarczanej usługi, więc rozwiązania tego typu są bardziej preferowane w wypadku serwisów charakteryzujących się dużą niezależnością i autonomicznością (np. system waluty elektronicznej BitCoin).

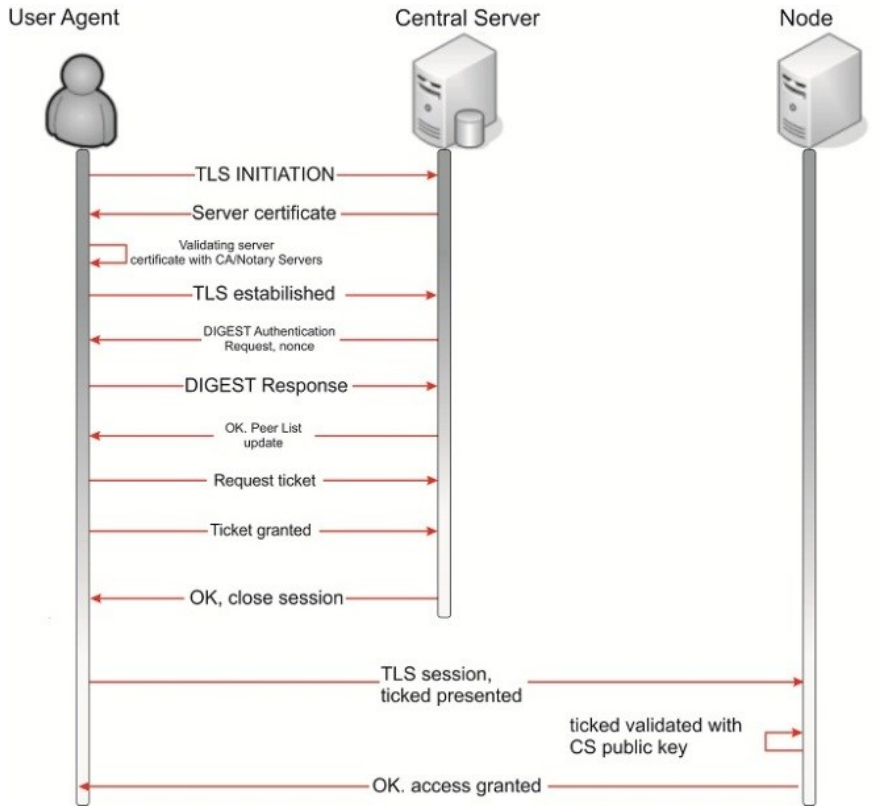
Kolejnym podejściem do minimalizacji ryzyka utraty danych w wyniku awarii centralnego serwera jest gromadzenie kopii bezpieczeństwa tych danych w zwykłych węzłach sieci P2P. Uzyskuje się w ten sposób znacznie wyższą elastyczność systemu oraz pewność odzyskania danych (znaczna ilość potencjalnych dostawców kopii zapasowych). Jednak wadą tego rozwiązania jest szereg problemów związanych z synchronizacją danych oraz ich bezpieczeństwem (zwykły węzeł może być znacznie łatwiejszym celem ataku mającego na celu kradzież danych).

Ponieważ obecne rozwiązania przechowywania i odbudowy danych (w szczególności kont użytkowników) nie spełniały wymogów postawionych na etapie projektowania mechanizmu odbudowy centralnego serwera, konieczne okazało się opracowanie własnego systemu zarządzającego kopiami zapasowymi. Kluczowe wymagania względem takiego systemu przedstawione są poniżej:

- Bezpieczeństwo – dane przechowywane w węzłach sieci P2P muszą być zaszyfrowane kluczem nieznanym żadnemu z węzłów. W tej sytuacji atakujący, nawet po pozyskaniu kopii zapasowej, nie będzie zdolny do jej odszyfrowania.
- Niezawodny protokół odbudowy danych, który musi być zdolny do automatycznego odtworzenia danych bez udziału starego centralnego serwera.
- System synchronizacji danych powinien być rozwiązany w sposób efektywny.

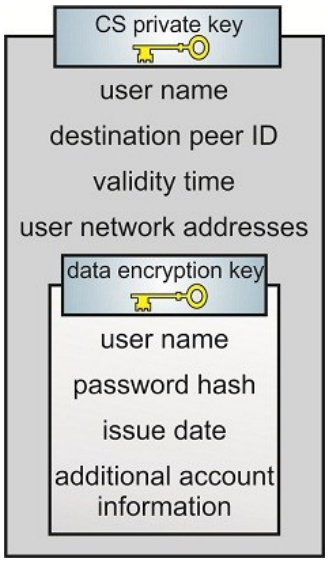
Opracowana w ramach niniejszej pracy metoda spełniająca te założenia składa się z dwóch etapów.

Podczas pierwszego etapu odbywa się zabezpieczenie danych i ich redystrybucja pomiędzy węzły infrastruktury. Proces ten ma przede wszystkim miejsce podczas logowania się użytkownika do systemu. Gdy użytkownik chce mieć dostęp do konkretnej usługi, łączy się wpierw z centralnym serwerem (poprzez szyfrowane połączenie TLS z obustronnym uwierzytelnieniem), gdzie następnie przechodzi proces autoryzacji. W kolejnym kroku serwer wybiera konkretny węzeł z sieci P2P, który będzie obsługiwał danego użytkownika (na podstawie algorytmu lokalizacji użytkownika oraz bilansu obciążeń węzłów) oraz generuje i przesyła użytkownikowi bilet. Bilet ten użytkownik przedstawia przydzielonemu węzłowi (w celu autoryzacji). Schemat procesu logowania przedstawiony został na rysunku 8.5.



Rys. 8.5 Schemat przebiegu procesu logowania użytkownika do systemu.

Bilet (rys. 8.6) wygenerowany przez centralny serwer jest zaszyfrowany kluczem prywatnym serwera KCS1, więc każdy węzeł jest w stanie sprawdzić jego autentyczność, używając klucza publicznego KCS2. Pierwsza część biletu posiada strukturę podobną do tej stosowanej w Kerberosie [92], a zatem zawiera nazwę użytkownika, ID przydzielonego węzła, początkowy i końcowy czas ważności biletu oraz listę adresów sieciowych, z których użytkownik może użyć danego biletu.



Rys. 8.6 Budowa biletu przydzielanego użytkownikowi w celu autoryzacji w węźle.

Druga część biletu stanowi natomiast kopię danych opisujących konto danego użytkownika, a więc jego login, hasło, datę utworzenia kopii, inne dane prywatne oraz pozostałe informacje odnośnie konta. Ta część biletu jest dodatkowo zaszyfrowana innym, symetrycznym kluczem K3 znanym tylko i wyłącznie serwerowi centralnemu. Każdy z węzłów w procesie autoryzacji użytkownika, dzięki użytej kryptografii asymetrycznej, jest w stanie odszyfrować pierwszą warstwę biletu, niezbędną do autoryzacji danego użytkownika. Drugą, wciąż zaszyfrowaną cześć biletu, węzeł zapisuje w swojej bazie danych (w połączeniu z nazwą użytkownika i czasem otrzymania biletu). W przypadku, gdy użytkownik loguje się po raz kolejny, a stary bilet utracił ważność i przedstawiony został nowy, dane z drugiej części biletu są podmieniane w bazie danych (ze względów bezpieczeństwa, stare dane nie są natychmiast usuwane lecz zachowywane jeszcze przez niedługi czas). W wyniku tego procesu każdy z węzłów mających interakcje z konkretnym użytkownikiem, przechowuje w swojej bazie danych zaszyfrowane informacje na temat jego konta.

Drugi etap mechanizmu odtworzenia baz danych centralnego serwera odpowiedzialny jest za bezpieczną dystrybucję klucza K3 pomiędzy węzły sieci P2P należące do systemu oraz faktyczną odbudowę bazy danych. W celu odzyskania informacji kluczowych do działania systemu, nowo wybrany serwer musi pozyskać dane zgromadzone w bazach danych poszczególnych węzłów, a następnie je odszyfrować i scalić. Do odszyfrowania informacji niezbędny jest klucz K3, który został wcześniej rozdystrybuowany pomiędzy węzły używając metody dzielenia sekretu [93]. Ponieważ w danym momencie nie wszystkie węzły muszą być aktywne (część z nich mogła ulec awarii, zostać skompromitowana lub po prostu się wylogować), pozostała część węzłów musi wciąż być zdolna do odtworzenia klucza K3, co z kolei wymusza zastosowanie progowej metody podziału sekretu. Dodatkowo założono, iż podział nie powinien być równomierny – węzły cieszące się większym zaufaniem powinny otrzymać większą liczbę części klucza. W zależności od typu danej usługi oraz rodzaju węzłów tworzących infrastrukturę P2P (np. sieć o mało zmiennej strukturze złożona z dedykowanych jednostek lub mocno zmienna, oparta w dużym stopniu np. o komputery użytkowników), odpowiednio dobrana powinna zostać progowość. W obecnej implementacji systemu przyjęto $t = 80\%$, wymagając współpracy co najmniej 80% węzłów w celu odtworzenia klucza K3.

Istnieje wiele algorytmów podziału sekretu, które mogą zostać użyte w celu redystrybucji klucza K3. Przegląd głównych z nich można znaleźć w opracowaniu [93]. Na potrzeby niniejszej pracy postanowiono wykorzystać zmodyfikowaną wersję schematu Shamira, ze względu na jego popularność oraz łatwość rozbudowy. Przeprowadzona modyfikacja polega na nierównomiernym podziale sekretu, w oparciu o reputację węzłów, zgodnie z metodą opisaną w podrozdziale 8.3.2. Sam algorytm podziału opiera się na wyznaczeniu wartości wielomianu $t-1$ stopnia w punkcie x_0 na podstawie danego zbioru t punktów. Dla danego progu (t, n) oraz k-bitowego klucza $K3$, serwer centralny wybiera n $(x_1, \dots, x_n)$ różnych, niezerowych elementów z $\mathbb{Z}_p$, gdzie p jest dużą (większą zarówno od n jak i K3) liczbą pierwszą. Następnie losowane (z rozkładu jednorodnego nad $\mathbb{Z}_p$) jest $t-1$ współczynników $a_1, \dots, a_{t-1}$, oraz tworzony zostaje wielomian

$$f(x) = K3 + a_1 x + \cdots + a_{t-1} x^{t-1}. \tag{8.3}$$

Ostatecznie, dla każdego węzła i:$1 \le i \le n$, zostaje wyznaczona para (x_i, f_i), gdzie $f_i = K3 + \sum_{i=1}^{t-1} a_i x^i \ mod\ p$, która następnie jest przesyłana węzłowi korzystając z bezpiecznego połączenia TLS. W celu odbudowy klucza K3 użyć można wielomianu Lagrange'a. Po udanym procesie

głosowania nowo wybrany serwer centralny zbiera przyznane według powyższego schematu udziały od węzłów oraz wyznacza klucz K3 jako $K3 = L(0)$, gdzie

$$L(x) = \sum_{i=1}^{t} f_i \prod_{j=1, j\neq i}^{t} \frac{x_j}{x_i - x_j} \; mod \; p. \tag{8.4}$$

Następnie zbierane są informacje na temat kont użytkowników, przechowywane dotychczas w węzłach w zaszyfrowanej formie. Przy użyciu K3 dane te są odszyfrowywane – w ten sposób na nowym serwerze scalona zostaje baza danych. Ponieważ dane często są zwielokrotnione, zazwyczaj wybierana jest ich najaktualniejsza wersja. Dodatkowo, jeśli zajdzie taka potrzeba, dzięki archiwalnym informacjom z baz danych węzłów, możliwa jest także odbudowa głównej bazy w postaci sprzed danego okresu czasowego (co może być użyteczne w wypadku, gdy stary centralny serwer zdążył wygenerować pewną ilość nieprawdziwych/błędnych biletów, zanim jego wadliwe działanie zostało wykryte przez Notariaty).

Prezentowana metoda, mimo licznych zalet posiada jednak pewną wadę, związaną z sytuacją, gdy dany użytkownik komunikował się z systemem tylko przy użyciu niewielkiej liczby węzłów i wszystkie z nich były aktualnie niedostępne w momencie odbudowy bazy danych. W takiej sytuacji informacja odnośnie konta tego użytkownika nie będzie mogła zostać przywrócona do chwili ponownego pojawienie się któregoś z tych węzłów. Problem ten jednak powinien dotyczyć bardzo niewielkiej liczy użytkowników, sporadycznie korzystających z systemu. Dodatkowo można się przed nim zabezpieczać przesyłając okresowe pełne kopie (również zaszyfrowane kluczem K3) całej bazy danych do wybranych, cechujących się najwyższa reputacją węzłów.

8.6 Bezpieczeństwo

Należy zauważyć, że w powyższym schemacie omówione zostały jedynie techniki „miękkiego bezpieczeństwa", bazującego na reputacji. Jednak system ten w prosty sposób może zostać rozbudowany o standardowe metody kryptograficzne z zakresu tzw. „twardego bezpieczeństwa". Dla przykładu, w sieciach P2P o strukturze grafu rzadkiego, gdzie wiele węzłów nie jest połączonych bezpośrednio, a więc głosy muszą być przekazywane przez inne węzły, zalecane jest stosowanie metody kryptografii klucza publicznego w celu podpisywania głosów.

Rozdział IX

Środowisko symulujące oraz wyniki symulacji

9.1 Wstęp

Na potrzeby przetestowania zastosowanych rozwiązań opracowane zostało środowisko symulacyjne pozwalające sprawdzać zachowanie poszczególnych elementów systemu w zależności od warunków sieciowych, stanu aktywności węzłów oraz awarii i spadku jakości łącz komunikacyjnych. W pierwszym etapie, w celu dokładnego zrozumienia charakterystyki opóźnień w sieciach komputerowych stworzony został model łącza internetowego (opisany w rozdziale 9.2.1), a następnie przeanalizowane zostały metody oceny jakości dźwięku służące do ewaluacji uzyskanych wyników (rozdział 9.2.2). Aplikacja symulatora została natomiast opisana w podrozdziale 9.2.3.

9.2.1 Model łącza internetowego

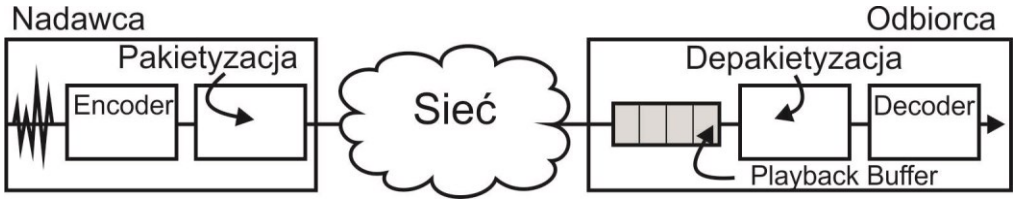
Oczywistym jest, iż jakość transmisji czasu rzeczywistego jest bezpośrednio skorelowana z jakością łącz internetowych wykorzystywanych do takiej transmisji. Dlatego też symulator mający na celu oszacowanie kluczowych parametrów systemu wspomagającego ten rodzaj transmisji, musi być oparty na odpowiednim modelu sieci internetowej. Model taki powinien być zdolny do odzwierciedlania podstawowych parametrów łącz sieciowych oraz realistycznego symulowania zaburzeń mogących pojawiać się w czasie transmisji.

Jak już wielokrotnie wspomniano, podstawowymi parametrami wpływającymi na jakość transmisji real-time są: całkowite opóźnienie (end-to-end), zmienne opóźnienie (jitter) oraz

zjawisko gubienia pakietów. Istotna jest również przepustowość łącza, gdyż warunkuje ona możliwość przeprowadzania transmisji danego typu oraz wpływa na wybór odpowiedniego algorytmu kodującego, co także często przekłada się na dodatkowe opóźnienie w transmisji.

a) Opóźnienie całkowite w transmisji real-time

Opóźnienie całkowite definiuje się jako czas, który upłynął pomiędzy chwilą, gdy sygnał przeznaczony do transmisji został wygenerowany (np. dźwięk wypowiedziany przez rozmówcę lub obraz zarejestrowany przez kamerę), a chwilą gdy sygnał ten został odtworzony po stronie odbiorcy. W skład tego opóźnienia wchodzi zatem proces kodowania danych, ich pakietyzacji, opóźnienie związane z transmisją danych poprzez łącze oraz dodatkowe opóźnienia po stronie odbiorcy – depakietyzacja i dekodowanie sygnału. Często po stronie odbiorcy występuje także dodatkowy bufor mający na celu zminimalizowanie wpływu jitteru i/lub przywrócenie właściwej kolejności odebranym pakietom. Bufor taki wprowadza dodatkowe opóźnienie proporcjonalne do swojej długości.



Rys. 9.1 Składniki opóźnienia powstającego podczas transmisji RT.

Opóźnienie wynikające z procesu kodowania zależy także od zastosowanego mechanizmu, którego rodzaj najczęściej podyktowany jest wymogami jakościowymi transmisji lub limitem przepustowości łącza. Przykładowo popularny standard kodowania dźwięku używany w telefonii VoIP ITU G.711 [9] wymaga przepustowości łącza na poziomie 64 kbps (82.4 kbps uwzględniając dodatkowe nagłówki IP oraz RTP) i wprowadza opóźnienie rzędu jedynie 0.125 ms. Jednak stosowany w przypadku łącz o niższej przepustowości standard G.729A [94] o wymaganej przepustowości 8 kbps (26.4kbps z nagłówkami) charakteryzuje się już 15 ms opóźnieniem algorytmu kodującego oraz dodatkowymi 5 ms na potrzeby zapełnienia bufora używanego do kompresji. W przypadku obu standardów przesyłanych jest 50 pakietów na sekundę, tak więc dodatkowe opóźnienie pakietyzacji wynosi 20 ms.

Opóźnienie związane z transmisją danych składa się natomiast z sumy opóźnień występujących na każdym etapie trasy. Są to odpowiednio:

a) opóźnienia propagacji – czas wymagany na przesłanie sygnału po łączu liczony jako $\frac{d}{s}$, gdzie d jest odległością między wysyłającym a odbiorcą, natomiast s jest prędkością propagacji fali elektromagnetycznej w danym ośrodku (równą prędkości światła w wypadku transmisji bezprzewodowej oraz 0.5c – 0.8c w wypadku łącz miedzianych). Opóźnienie to jest szczególnie istotne w przypadku połączeń międzykontynentalnych, gdzie potrafi przekroczyć wartość 100 ms,
b) opóźnienia transmisji – wynikają bezpośrednio z określonej przepustowości łącza,
c) opóźnienia wynikające z kolejkowania pakietów – ponieważ w danej chwili pakiety mogą być przesyłane wyłącznie pojedynczo, w routerach wymagany jest system kolejkowy. Zakładając brak priorytetów, kolejka taka przyjmuje postać standardowej kolejki FIFO. W przypadku łącz przeciążonych lub o małej przepustowości opóźnienie to może być decydującym czynnikiem degradacji jakości transmisji real-time. Dla przykładowego łącza o przepustowości 1.5 Mbps i przy 5 zakolejkowanych pakietach po 2 kB każdy, opóźnienie to wyniesie 53 ms.

Podsumowując, całkowite opóźnienie transmisji można opisać wzorem (9.1):

$$\begin{aligned} d_{end\_to\_end} &= d_{enc} + d_{pack} \\ &+ \sum_{r \in route} \left(d_{prop}(r) + d_{trans}(r) + d_{queuing}(r) \right) + d_{depac} \\ &+ d_{dec} + d_{buff} \end{aligned} \tag{9.1}$$

Jak łatwo zauważyć, jedynym zmiennym czynnikiem w powyższym wzorze jest opóźnienie wynikające z kolejkowania, które jest także główną przyczyną jitteru w sieci. Z równania tego można również bezpośrednio wyliczyć maksymalny dopuszczalny jitter dla danej metody kodowania.

b) Zmienne opóźnienie

W sieciach opartych o pakietową transmisję danych można wyróżnić dwa podstawowe typy jitteru:

- **Jitter stały**

  Jest rejestrowany pomiędzy każdym kolejnym pakietem przesyłanym przez sieć. Jego źródłem są miedzy innymi stałe fluktuacje związane z szumami cieplnymi powodującymi np. dryf czasu w zegarach stosowanych w urządzeniach sieciowych lub ze sposobem kolejkowania pakietów (niektóre routery posiadają równoległe kolejki, które z powodu różnej długości wprowadzają dodatkowe zmienne opóźnienie). Niewielkie zmienne opóźnienia mogą także wynikać z zastosowanej metody routingu polegającej na użyciu kilku równoległych tras przez dostawców usług IP (technika stosowana w celu bilansowania obciążenia łącz oraz zwiększenia niezawodności usługi).

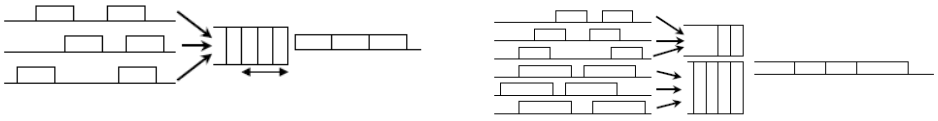
- **Jitter okresowy** – jitter o charakterze okresowym, na który składają się:

  - chwilowe przeciążenia w sieci lokalnej - mimo iż w typowych sieciach LAN przeciążenia występują rzadko, mogą jednak pojawiać się w nich opóźnienia o charakterze wąskiego impulsu. Wynikają one ze specyfiki algorytmu dostępu CSMA/CD stosowanego w sieciach Ethernet, który w wypadku wykrycia kolizji może cofnąć pakiet o pewien ograniczony okres o losowej długości. Wartości tych opóźnień wynoszą zazwyczaj do kilku ms.

  - okresowe przeciążenia na łączach dostępowych – sieci dostępowe przeważnie są wąskim gardłem na całej ścieżce pakietu, z tego też powodu są one głównym źródłem jitteru. Dla danego łącza o przepustowości B, opóźnienie pakietu spowodowane kolejkowaniem określone jest wzorem:

$$d_{queuing} = \frac{\sum_{p \in Q} size(p)}{125 * B} \tag{9.2}$$

  Problem kolejkowania można zminimalizować poprzez stosowanie kolejek priorytetowych oraz mechanizmów rezerwacji łącza [2] [3].



Rys. 9.2. Standardowa kolejka FIFO oraz model z kolejką priorytetową.

- odświeżanie tras routingu – ponieważ pakiety zawierające nowe tablice routingu są przesyłane z wyższym priorytetem, operacja odświeżania trasy może powodować periodyczne, chwilowe opóźnienia.

- migotania trasy – jest to zjawisko pojawiające się w przypadku specyficznej awarii interfejsu sieciowego, gdy zmienia on swój stan (aktywny/nieaktywny) periodycznie. Zjawisko to ma swoje odzwierciedlenie w tablicach routingu i może wprowadzać dłuższe okresowe zmiany w czasie dostarczania pakietów.

Poza wymienionymi, istnieją także inne zjawiska wprowadzające zmienne opóźnienie (jak np. zmiany związane z przydzielaniem mocy obliczeniowej przez CPU czy zastosowane mechanizmy bezpieczeństwa, np. firewalle), jednak ich wpływ jest mniej znaczący w porównaniu do powyższych.

Ponieważ liczność i różnorodność potencjalnych czynników wprowadzających zmienne opóźnienie jest znaczna, opóźnienie to ma dość złożony charakter. Istnieją metody pozwalające, dla danej funkcji gęstości prawdopodobieństwa opisującej jitter, wyodrębnić jego deterministyczną część [1], ale wciąż pozostaje trudna do zamodelowania część losowa. Badania przeprowadzone miedzy innymi w pracach [1], [2] oraz [5] wskazują na to, iż rozkład prawdopodobieństwa zmiennego opóźnienia w sieci można opisać przy pomocy rozkładu Laplace'a, o dystrybuancie danej jako:

$$F(x) = \begin{cases} \frac{1}{2} e^{\left(-\frac{\mu - x}{b}\right)} & dla\ x < \mu \\ 1 - \frac{1}{2} e^{\left(-\frac{\mu - x}{b}\right)} & dla\ x \geq \mu \end{cases} \tag{9.3}$$

Daniel, White oraz Teauge [95] zaproponowali prosty model jitteru oparty o generator liczb losowych zgodnych z rozkładem Laplace'a, blok dodający szum biały oraz mechanizm rozkładający impulsy skokowe.

Rys. 9.3 Model jitteru oparty o rozkład Laplace'a zaprezentowany w [95]

Istnieją także inne badania [96], z których wynika, iż w niektórych wypadkach rozkład jitteru charakteryzuje się dłuższym ogonem i odbiega od wprowadzonego modelu opartego o rozkład Laplace'a. Rizo et.al [96] pokazują, iż niekiedy rozkład Cauchy'ego może dać akceptowalne przybliżenie rzeczywistego rozkładu zmienności opóźnienia.

Jednakże, ponieważ w powyższych modelach opóźnienia generowane są na podstawie rozkładu prawdopodobieństwa, nie istnieją korelacje pomiędzy kolejnymi opóźnieniami pakietów. Również dodatkowo zaproponowane mechanizmy tylko w niewielkim stopniu odzwierciedlają zjawisko korelacji. Z tego też powodu, na potrzeby opisywanego w tej pracy symulatora, został opracowany model jitteru oparty na szeregu czasowym. Przykład modelowania jitteru za pomocą prostego dwustanowego układu został przedstawiony w [97]. W celu zastosowania w prezentowanym symulatorze model ten został znacznie rozbudowany. Zmienne opóźnienie modelowane jest jako seria wielu niezależnych zdarzeń, reprezentujących potencjalne źródła jitteru, przy czym każde z tych zdarzeń może z zadanym prawdopodobieństwem generować impuls o losowej amplitudzie (z określonego przedziału wysokości). Do zdarzeń przypisane są także niezależne funkcje filtru. W użytym w pracy prostym modelu zastosowany został filtr oparty o ruchomą średnią o zmiennym oknie q daną jako:

$$m_k = \frac{1}{q} \sum_{i=k-q+1}^{k} y(i) \tag{9.4}$$

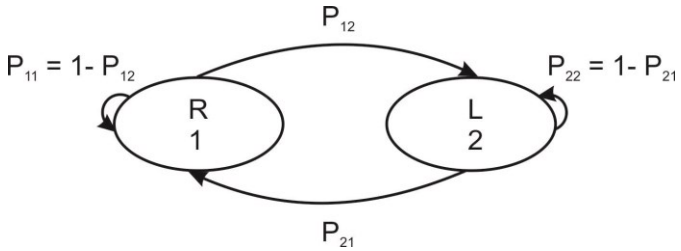
Dodatkowo model jitteru został połączony z opisanym w następnym podrozdziale modelem gubienia pakietów, dzięki czemu istnieje korelacja pomiędzy wzmożonym zmiennym opóźnieniem oraz prawdopodobieństwem utraty pakietów.

Modelowanie zjawiska gubienia pakietów

Jednym z najprostszych i często stosowanych modeli do symulowania zjawiska gubienia pakietów jest proces Bernoulliego. Model taki charakteryzuje się tylko jednym parametrem P_{loss}, który jest prawdopodobieństwem utraty pakietu w danej sieci. Jednocześnie zakłada się, iż gubienie kolejnych pakietów jest zjawiskiem niezależnym od stanu poprzednich pakietów. Przy takim założeniu prawdopodobieństwo zgubienia x pakietów w ciągu n kolejnych pakietów dane jest jako:

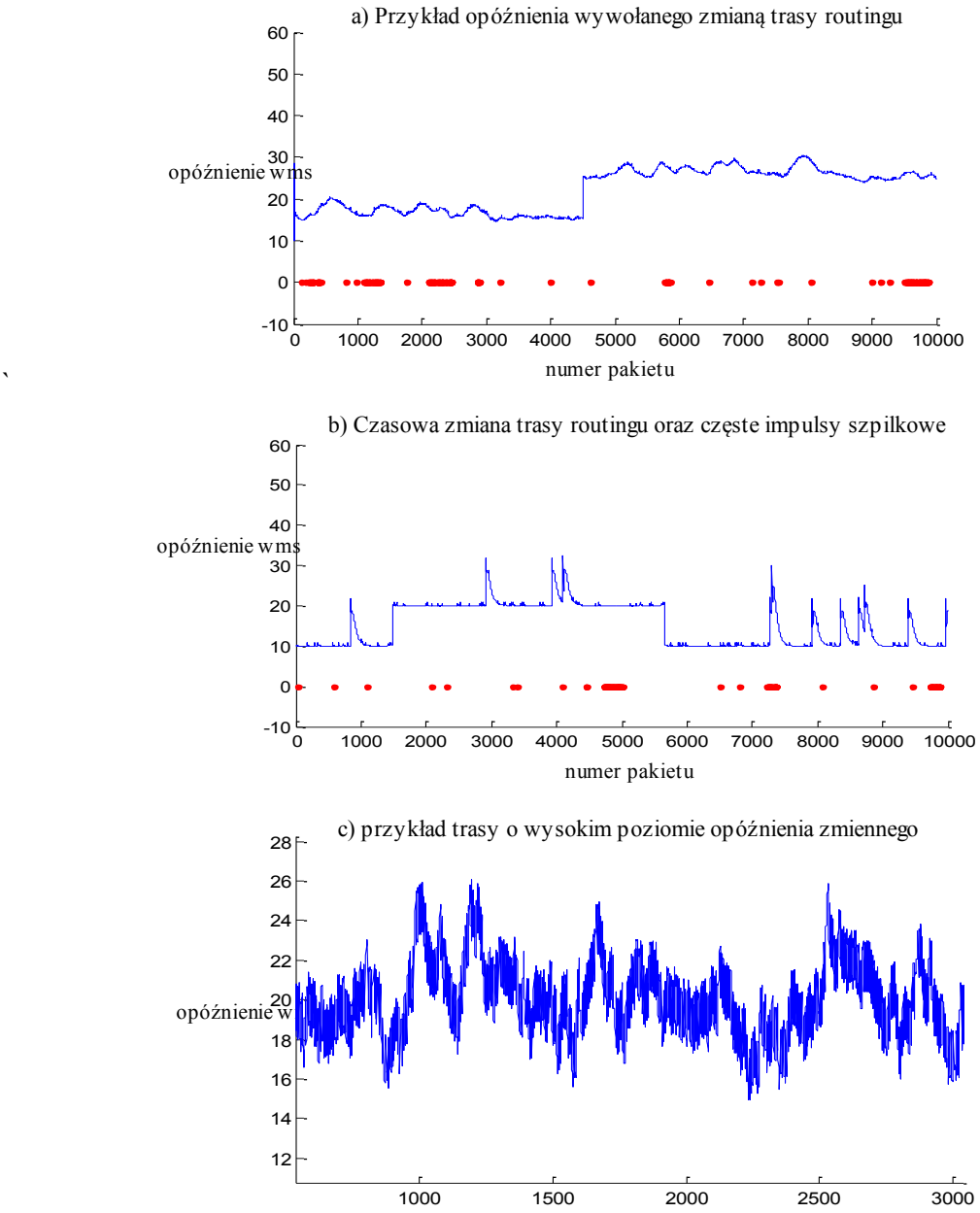
$$P(x|n) = \binom{n}{x} P_{loos}^{x}(1 - P_{loos})^{n-x} \tag{9.5}$$

Niestety model ten nie pozwala na dokładniejsze symulowanie zjawiska gubienia pakietów, które, jak pokazują obserwacje, charakteryzuje się silnie nieregularnym natężeniem. W związku z tym zostały opracowane bardziej rozwinięte modele, których zdecydowana większość oparta jest na łańcuchach Markowa. Dwustanowy model Gilberta [97] uwzględniający nieregularność natężenia zakłada istnienie stanu odbioru (R), w którym wszystkie pakiety są odbierane oraz stanu straty (L), w którym wszystkie pakiety są tracone:



Rys. 9.4 Prawdopodobieństwo przejścia pomiędzy stanami R i L

Prawdopodobieństwo przejścia pomiędzy stanami R i L dane jest odpowiednio $P_{12} = P(q_t = L|q_{t-1} = T)$ oraz $P_{21} = P(q_t = T|q_{t-1} = L)$. Model ten został rozwinięty przez Elliotta [98] poprzez wprowadzenie dodatkowo dla każdego ze stanów prawdopodobieństwa sukcesu $s_1\ i\ s_2$ oraz porażki $s_1\ i\ s_2$. Przegląd przez pozostałe modele gubienia pakietów można znaleźć w pracach [99], [100].



Rys. 9.5 Przykładowe symulacje opóźnienia oraz gubienia pakietów (czerwone kropki).

9.2.2 Metody oceny jakości dźwięku

Istnieje szereg metod oceny jakości dźwięku. Zasadniczo dzielą się one na dwie klasy: metody subiektywne oraz metody obiektywne. Metody z pierwszej grupy, jak wskazuje nazwa,

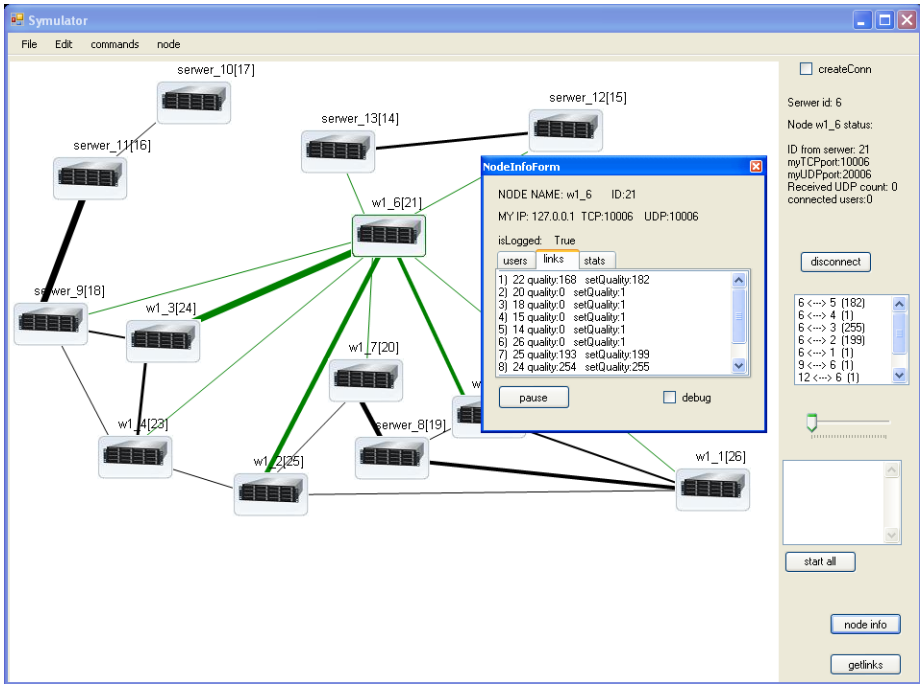
opierają się na subiektywnych wrażeniach grupy odbiorców, którzy mają za zadanie odsłuchać zestaw testowych sygnałów, a następnie ocenić jakość każdego z nich (najczęściej według ustalonej skali z przedziału 1-5). Natomiast w metodach obiektywnych grupa słuchaczy zostaje zastąpiona algorytmem perceptualnym. Dokładny przegląd wyżej wymienionych metod można znaleźć w [101]. W niniejszej pracy autor użył obiektywnej metody Perceptual Evaluation of Speech Quality (PESQ). Metoda ta, zgodnie z rekomendacją ITU-T P.862 (02/01), uznawana jest obecnie za światowy standard w testowaniu jakości mowy.

9.2.3 Aplikacja symulatora

Środowisko symulacyjne opracowane w ramach niniejszej rozprawy składa się z pięciu podstawowych modułów:

- Moduł węzła, odpowiedzialny za operacje wykonywane w ramach pojedynczego węzła, tj.: pośredniczenie w procesie routingu, testowanie jakości połączenia z innymi węzłami oraz komunikację z serwerem centralnym. Moduł ten wyposażony został w prosty system agentowy, przy czym agent rozumiany jest tu jako dodatkowa klasa uruchamiana jako niezależny wątek i posiadająca pełny dostęp do zmiennych i funkcji modułu, a także własny protokół komunikacyjny oparty o gniazdka TCP. Moduł węzła posiada także możliwość zdalnej konfiguracji parametrów pracy.
- Moduł superwęzła, będący implementacją algorytmów działających w ramach serwera centralnego, czyli mechanizmów odpowiedzialnych za wyliczanie ocen reputacyjnych, mechanizm routingu kontekstowego oraz mechanizm logowania dla węzłów i użytkowników końcowych.
- Moduł terminala końcowego, będący oprogramowaniem działającym jako terminal VoIP.
- Moduł łącza internetowego, opracowany na podstawie modelu z podrozdziału 9.2.1.
- Moduł symulatora infrastruktury. Jest to nadrzędny program zawierający w sobie wszystkie powyższe moduły. Pozwala on na graficzne zaprojektowanie pożądanej infrastruktury sieciowej poprzez dodawanie węzłów oraz definiowanie połączeń między nimi. Interfejs użytkownika symulatora przedstawiony został na rysunku 9.6.

Każdy z powyższych modułów został napisany w języku C#. W celu uzyskania bezpośredniego dostępu do bufora dźwięku wykorzystano oprogramowanie Microsoft DirectSound. Strumień danych real-time uzyskiwany jest poprzez transmisję sygnału mowy z wybranego pliku audio pomiędzy węzłami sieci, z wykorzystaniem modulacji G.711. Parametry takie jak zachowanie węzłów oraz jakość linii transmisyjnych mogą być dowolnie modyfikowane w celu symulacji różnych scenariuszy.



Rys. 9.6 Interfejs symulatora infrastruktury VoIP.

Moduły projektowane były w taki sposób, aby mogły stanowić niezależne programy uruchamiane na rzeczywistej infrastrukturze. Oprogramowanie symulatora uruchamia wiele instancji takich modułów jako niezależne wątki i konfiguruje je do pracy na pojedynczej maszynie (zastosowanie tego samego numeru IP oraz różnych numerów portów, tak aby możliwa była wzajemna komunikacja poprzez TCP oraz UDP). Profil zachowania węzłów oraz parametry łącz są następnie pobierane z pliku konfiguracyjnego z opisem symulacji. Każdy z modułów w trakcie symulacji, przed wysłaniem konkretnego pakietu poprzez gniazdko UDP, pobiera informacje z powiązanego z daną trasą modułu łącza. Następnie zgodnie z nimi przesyła pakiet z opóźnieniem lub nie przesyła go wcale, symulując zjawisko gubienia pakietów.

Ponadto, część z opracowanych w pracy rozwiązań, która do skontrolowania poprawności działania nie wymagała infrastruktury real-time (jak mechanizmy reputacyjne i wykrywanie koalicji) testowane były niezależnie, z wykorzystaniem środowiska Matlab.

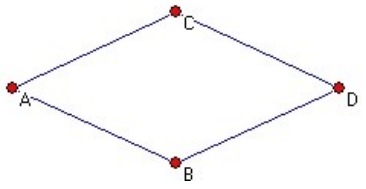
9.3 Wyniki przeprowadzonych symulacji.

W podrozdziale tym przedstawione zostały wyniki przeprowadzonych symulacji. Proces symulacji odbywał się dwuetapowo. W pierwszym etapie posłużono się bardzo prostym modelem składającym się jedynie z czterech węzłów i jednego strumienia audio (rysunek 9.7). Ten uproszczony schemat struktury pozwolił na dokładną analizę działania, na poziomie pojedynczej trasy, mechanizmu routingu adaptacyjnego oraz reputacji dla ścieżek. Drugi etap symulacji odbywał się na dużej, składającej się z 50 węzłów infrastrukturze, w której transmitowane było jednocześnie wiele strumieni mowy. Pozwoliło to z kolei na dokładniejszą analizę wydajności mechanizmu routingu oraz reputacji dla węzłów.

9.3.1 Symulacje z wykorzystaniem prostej infrastruktury

a) Przykład działania mechanizmu adaptacyjnego doboru ścieżki.

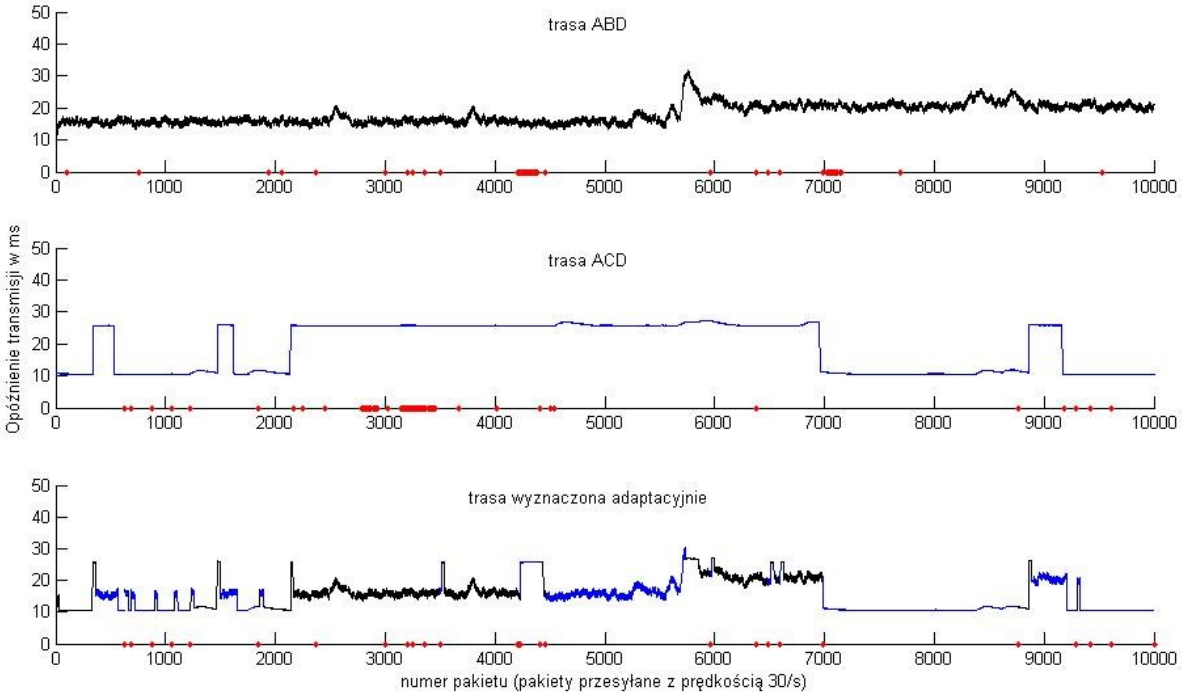
W przypadku transmisji przebiegającej z punktu A do D możliwe są trasy ABD oraz ACD (rysunek 9.7)



Rys. 9.7 Schemat sieci z dwiema alternatywnymi trasami.

Jakość transmisji dla tras ABD oraz ACD została wygenerowana z wykorzystaniem metody opisanej w podrozdziale 9.2.1 i przedstawiona odpowiednio w górnej oraz środkowej części rysunku 9.8. Dolna część rysunku obrazuje trasę powstałą w wyniku działania mechanizmu adaptacyjnego. Kolorem czarnym zaznaczone zostały odcinki należące do trasy ABD, a

niebieskim – należące do ACD. Czerwone punkty na wykresach odzwierciedlają zjawisko gubienia pakietów.



Rys. 9.8 Wykres jakości transmisji (opóźnienia i pakietów) odpowiednio dla tras ABD, ACD oraz trasa powstała w wyniku działania mechanizmu wyboru ścieżki.

Istotne parametry dla każdej ze ścieżek przedstawione zostały w tablicy 9.1.

Tablica 9.1 Parametry ścieżek o jakości przedstawionej na wykresie 9.8.

	ABD	**ACD**	**adaptacyjna**
Średnia ocena PESQ	4.1479	4.0972	4.2336
% utraconych pakietów	0.84	1.04	0.34
Średnie opóźnienie[ms]	18.1845	18.9639	15.0724

Rezultaty przedstawione w tablicy 9.1 jednoznacznie wskazują na istotny zysk wynikający z zastosowania mechanizmu adaptacyjnego wyboru ścieżki. Również kolejne symulacje

przeprowadzone dla innych parametrów ścieżek, jednoznacznie potwierdzają celowość stosowania wspomnianego mechanizmu.

b) Wpływ rozmiaru okna pomiarowego na jakość transmisji.

Ponieważ decyzja o zmianie aktualnej trasy na inną podejmowana jest w oparciu o obserwację każdej ze ścieżek, istotny jest dobór czasu takiej obserwacji. Z jednej strony pożądany jest krótki czas obserwacji, pozwalający szybko reagować na pogorszenie jakości transmisji, z drugiej natomiast, bardzo częste przełączanie ścieżek wiąże się ze znacznym wzrostem transmisji sygnalizacyjnej w sieci oraz ze wzrostem nakładu obliczeniowego związanego z ciągłym przeliczaniem nowych tras. Dla przykładu, systemy takie jak Spines [102] reagują na awarię ścieżki dopiero po czasie 10 sek. Istotny wpływ na ten czas ma przede wszystkim okres niezbędny do synchronizacji tablic routingu między węzłami, tak aby nie dopuścić do zjawiska migotania trasy i tworzenia cykli. Architektura systemu zaprojektowanego w niniejszej rozprawie pozwala bardzo efektywnie rozwiązać ten problem poprzez wykorzystanie Super Węzła (centralnego serwera), gdyż za proces gromadzenia informacji, obliczania tras i rozsyłania nowych tablic routingu odpowiada ta sama maszyna.
W tym wypadku czas niezbędny do detekcji i wyznaczenia nowej trasy można obliczyć zgodnie ze wzorem:

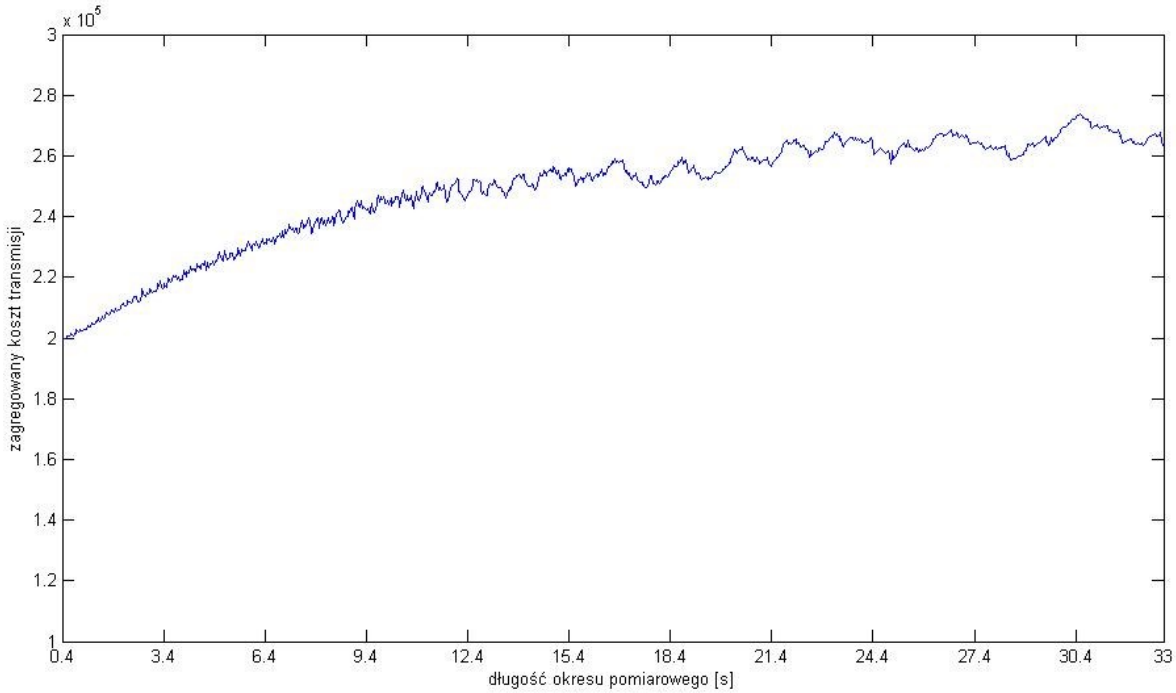
$$T_{reRoute} = wd_{size} + T_{tr} + T_q + T_{calc} + T_{dist}\,, \tag{9.6}$$

gdzie wd_{size} jest rozmiarem okna obserwacyjnego,

T_{tr} czasem transmisji nowych pomiarów do serwera centralnego,
$T_q < wd_{size}$ czasem kolejkowania nadesłanych pakietów,
T_{calc} czasem przetwarzanie danych i wyliczania nowych tras routingu,
T_{dist} czasem niezbędnym do rozesłania nowo wyliczonych tablic routingu pomiędzy węzły.

Proces obliczania nowych tras nie jest zadaniem czasochłonnym. Implementacja użyta w symulatorze opracowanym na potrzeby niniejszej pracy bazuje na algorytmie Floyda-Warshalla o złożoności obliczeniowej $\theta(n^3)$, który do działania na komputerze wyposażonym w procesor Intel i5 4260U potrzebuje około 5 ms dla sieci o 50 węzłach. Sam proces transmisji zależy od fizycznych właściwości sieci i trwa zazwyczaj 2-30 ms. Jak zatem widać największy wpływ na czas $T_{reRoute}$ ma rozmiar okna pomiarowego wd_{size}. Wykres powstały w wyniku uśrednienia 20

losowo wygenerowanych przypadków różnych tras dla zagadnienia a) i obrazujący zależność między rozmiarem tego okna, a kosztem wyznaczonej adaptacyjnie ścieżki przedstawiony został na rysunku 9.9.

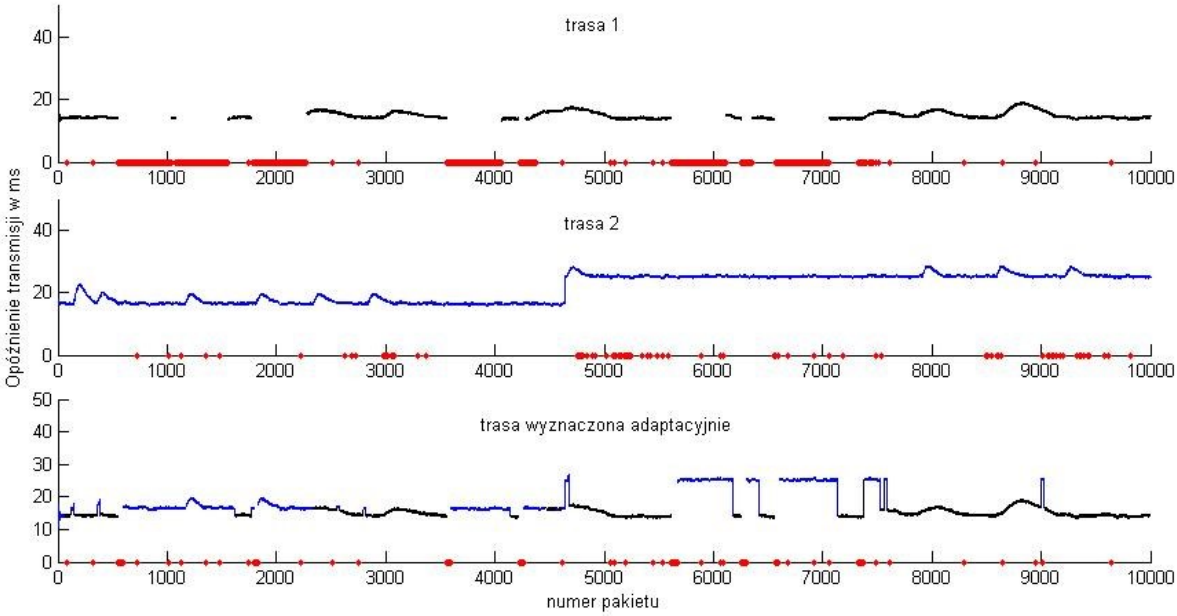


Rys. 9.9 Zależność między rozmiarem okna pomiarowego a kosztem wyznaczonej trasy. Jako funkcji kosztu użyto miary spodziewanego opóźnienia (tablica 7.1).

Można łatwo zauważyć, iż najwyższy zysk z użycia metody adaptacyjnej uzyskuje się przy oknie o niewielkich rozmiarach. Z tego też powodu system został skonfigurowany do pracy z oknem pomiarowym o długości 0.6s, co odpowiada 30 pakietom w standardzie kodowania G.711 (w rzeczywistości próbka będzie znacznie większa, ponieważ najczęściej wiele strumieni jest przesyłanych równolegle).

c) Wpływ reputacji linii na jakość transmisji.

Celem zastosowania w opracowanym systemie mechanizmu reputacyjnego dla połączeń międzywęzłowych jest chęć minimalizacji ryzyka obniżenia jakości transmisji w wypadku okresowych awarii lub bardzo silnych degradacji jakości łącz. Opisany powyżej mechanizm adaptacyjnego dobru trasy pozwala w istotny sposób zwiększyć jakość transmisji poprzez obserwację, a następnie wybór tras o lepszej jakości. Jednakże konstrukcja tego mechanizmu narzuca minimalny okres $T_{reRoute}$, niezbędny do wykrycia awarii i wyznaczenia nowej trasy. Mimo, iż czas ten jest stosunkowo niewielki (w przyjętej konfiguracji wynosi poniżej 2 s), to w przypadku całkowitej awarii łącza przerwa ta jest zauważalna i znacznie wpływa np. na komfort prowadzonej rozmowy. Zjawisko to zostało zaprezentowanie na rysunku 9.10, gdzie jedna z linii (trasa 1) ulega okresowym awariom.



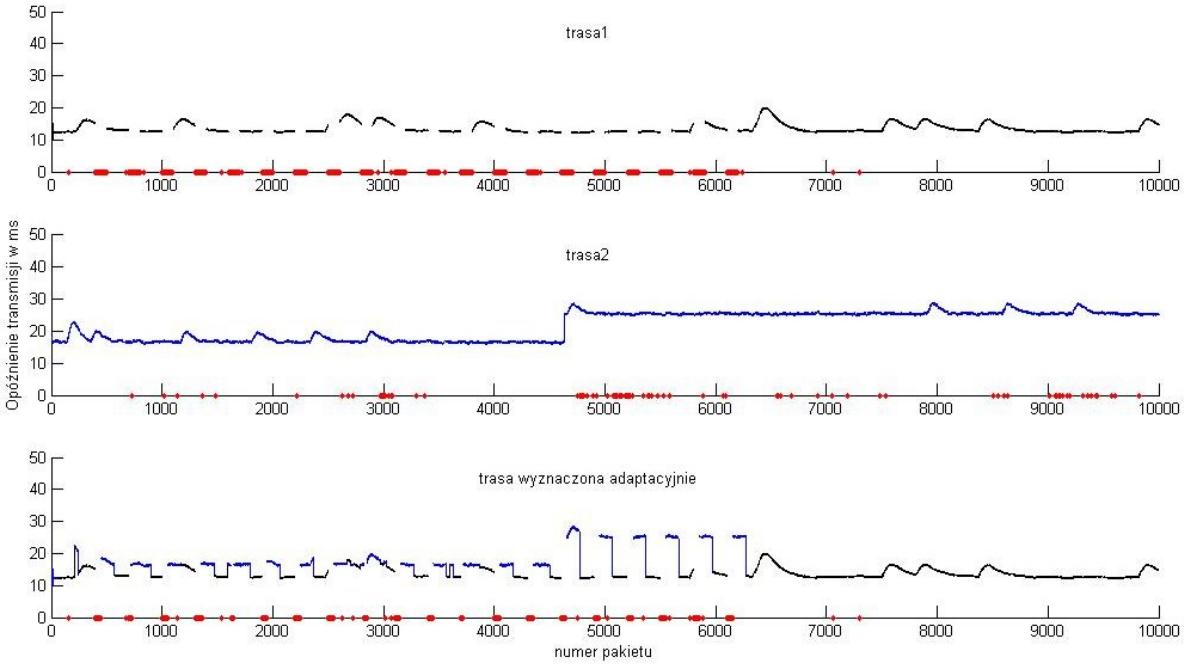
Rys. 9.10 Wynik działania mechanizmu adaptacyjnego w wypadku częstych awarii połączenia.

Przedstawiona w górnej części rysunku 9.10 trasa 1 cechowała się lepszą jakością niż alternatywna trasa 2 w okresach, gdy połączenie funkcjonowało prawidłowo i dlatego była ona preferowana przez mechanizm adaptacyjny. Występujące na niej awarie były jednak niemożliwe do przewidzenia przez ten mechanizm i skutkowały przerwami w transmisji (o czasie $T_{reRoute}$) w wyznaczonej adaptacyjnie trasie (dolny wykres rys. 9.2). Wpływ tego zjawiska przedstawiony został w tablicy 9.2:

Tablica 9.2. *Parametry ścieżek o jakości przedstawionej na wykresie 9.10.*

	Trasa1	**Trasa2**	**Adaptacyjna bez reputacji**	**Adaptacyjna z reputacją**
Średnia ocena PESQ	2.5119	3.9450	3.6477	3.9521
% utraconych pakietów	31.98	0.94	3.47	1.55
Średnie opóźnienie [ms]	15	22	17	19

Szczególnie pesymistyczny przypadek zaistnieje, gdy mamy do czynienia z łączem o bardzo dobrych parametrach transmisyjnych, lecz ulegającym bardzo częstym cyklicznym awariom. W okresach poprawnego działania będzie ono preferowane przez mechanizm adaptacyjny, lecz notoryczne awarie będą skutkowały okresowymi przerwami o czasie $T_{reRoute}$ niezbędnym do zmiany trasy na alternatywną. Ostatecznie zaowocuje to bardzo niską jakością transmisji. Sytuacja taka została przedstawiona na rysunku 9.11:



Rys. 9.11 Pesymistyczny przypadek z łączem o częstych cyklicznych awariach.

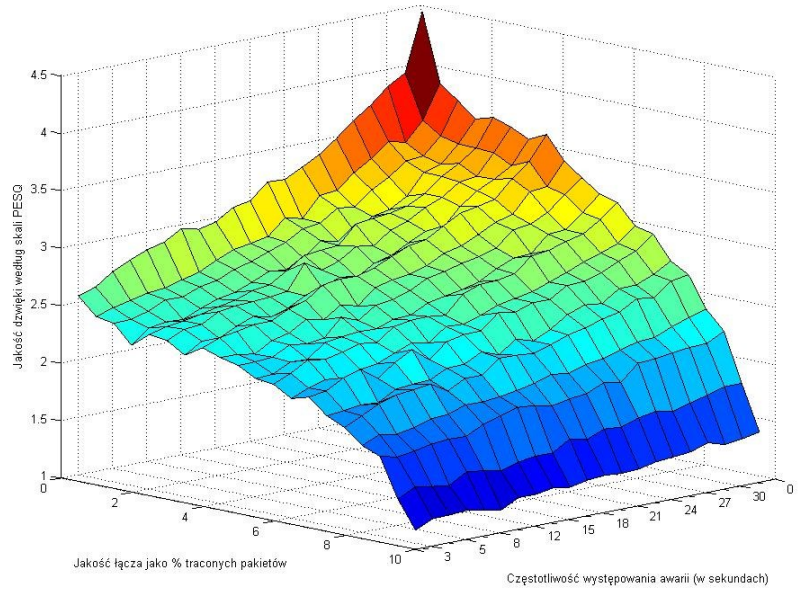
Cykliczne awarie łącza (średnio co 7 sekund) doprowadziły do istotnego spadku jakości transmisji trasy wyznaczonej adaptacyjnie. Szczegółowe dane dla każdej z tras przedstawione zostały w tablicy 9.3.

Aby zminimalizować zademonstrowane powyżej negatywne efekty wywoływane awariami łącza, autor postanowił rozszerzyć mechanizm adaptacyjnego przełączania ścieżek tak, aby przy wyliczaniu każdego z bezpośrednich połączeń uwzględniać także jego reputację, zgodnie ze wzorem:

$$r\_cost(i,j) = \frac{r\_cost(i,j)'}{rep(i,j)}. \tag{9.7}$$

Jednym z problemów, które należało w tym wypadku rozwiązać, był właściwy dobór parametrów systemu reputacyjnego tak, aby możliwe było dokonanie optymalnego wyboru pomiędzy awaryjnym łączem o dobrej jakości, a poprawnie działającym łączem o jakości niższej. Wynik symulacji wpływu awaryjności łącza oraz spadku jakości trasy na jakość sygnału mowy (ocenianego przy pomocy PESQ) przedstawiono na wykresie 9.12. Porównane w niej zostały straty wynikające z konieczności re-routingu (trwającego 1 sekundę, a więc tracone jest 50 kolejnych pakietów w wypadku kodowania G.711) oraz straty spowodowane losowym gubieniem pakietów na linii. Okres symulacji trwał 5 minut, przy czym każda minutowa próbka oceniania była niezależnie, a następnie wynik PESQ został uśredniony.

W praktyce okazuje się, iż w pewnych sytuacjach rozsądniej jest wybrać trasę o wysokiej jakości transmisji, która ulega sporadycznie awarii/degradacji, niż zdecydować się na transmisję po łączu cechującym się stałą niską jakością, lecz przy tym stabilnym działaniem. Przykładowo trasa ulegająca awarii co 30 sekund wciąż daje lepszy wynik PESQ niż alternatywna trasa o stałym współczynniku gubienia pakietów równym 3%.

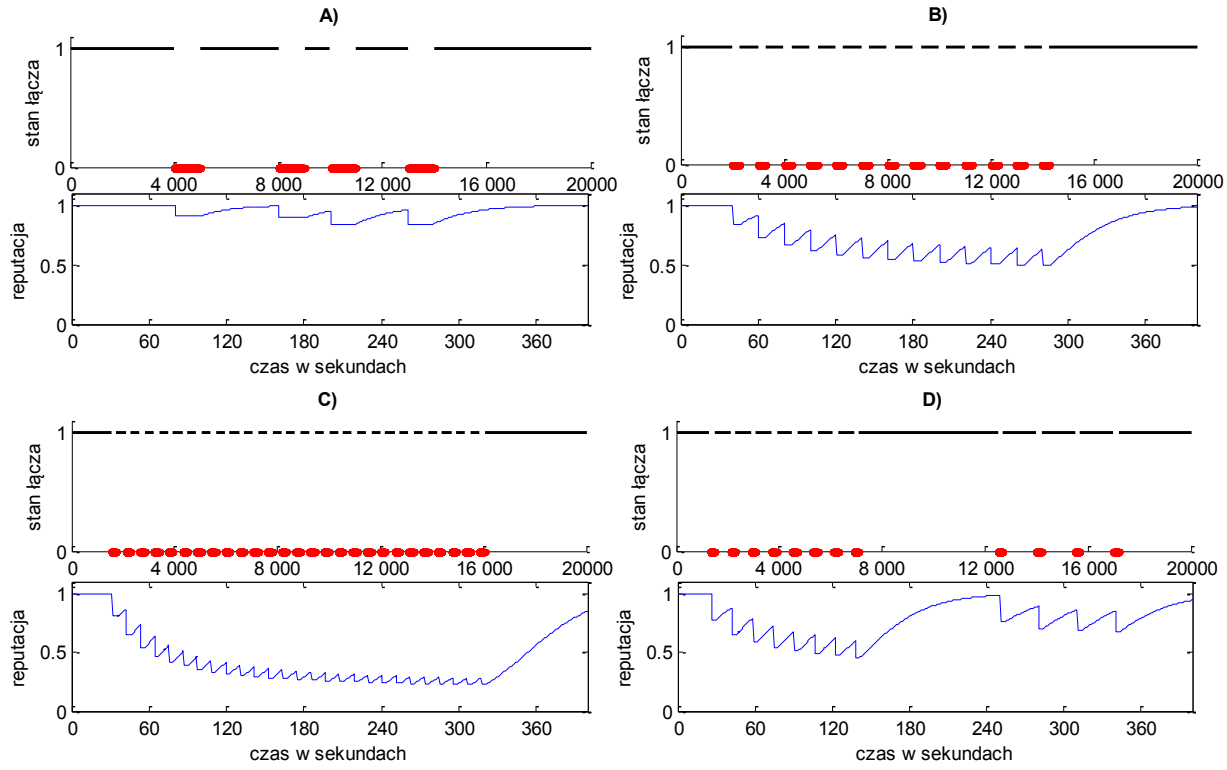


Rys. 9.12 Wpływ jakości łącza oraz przerw w transmisji spowodowanych awariami łącza na jakość dźwięku podczas 5 minutowej rozmowy.

Parametry systemu reputacyjnego zostały dobrane w sposób doświadczalny, tak aby wynikowy koszt $r\_cost(i,j)$ spełniał powyższą zależność. Sam system reputacyjny zlicza jedynie częstotliwość występowania awarii a nie ich długość (ponieważ istotny jest jedynie czas re-routingu), a jego parametry zostały dobrane następująco:

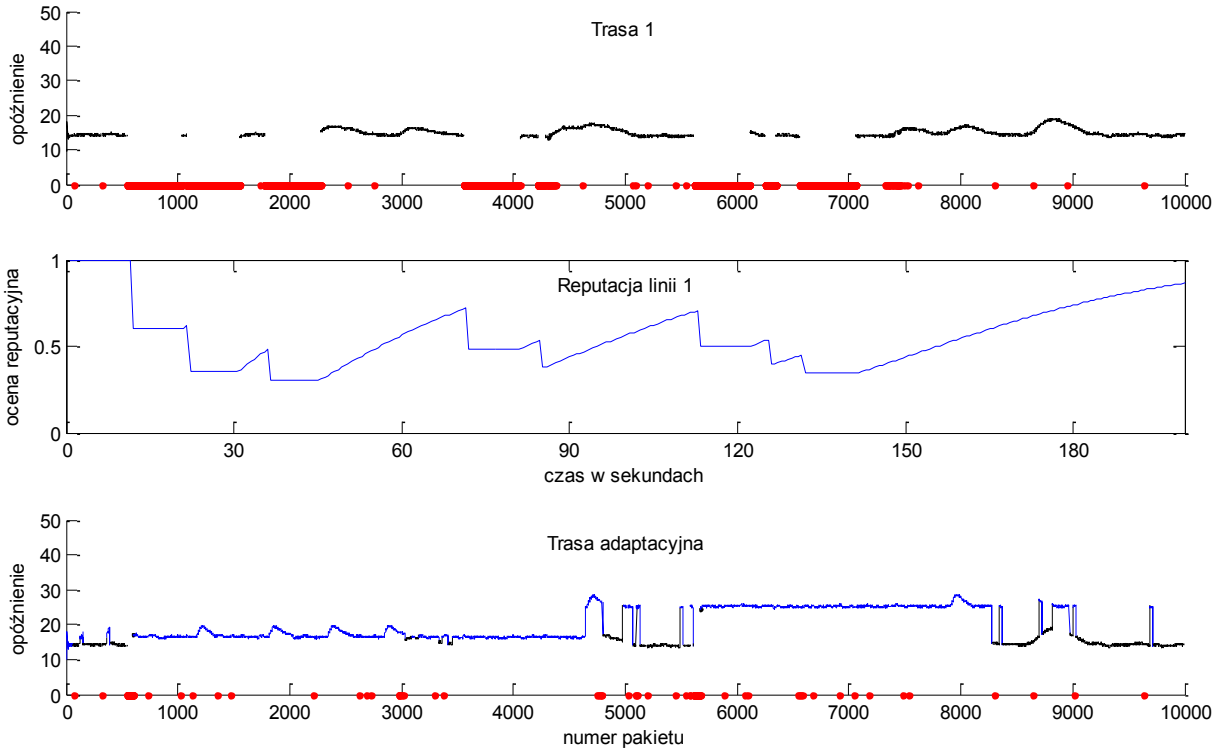
Współczynnik zapominania dla pozytywnych ocen disc_a=0.9999;
Współczynnik zapominania dla negatywnych ocen: disc_b=0.98;
Ocena w wypadku poprawnego działania a=0.08 b=0;
Ocena w wypadku wykrycia awarii a=0 b=failure_count, gdzie failure count jest licznikiem zliczającym kolejne wystąpienia awarii.
W wypadku trwania awarii a=8

Przykład wyliczania reputacji dla różnego typu awarii został przedstawiony na rysunku 9.13.

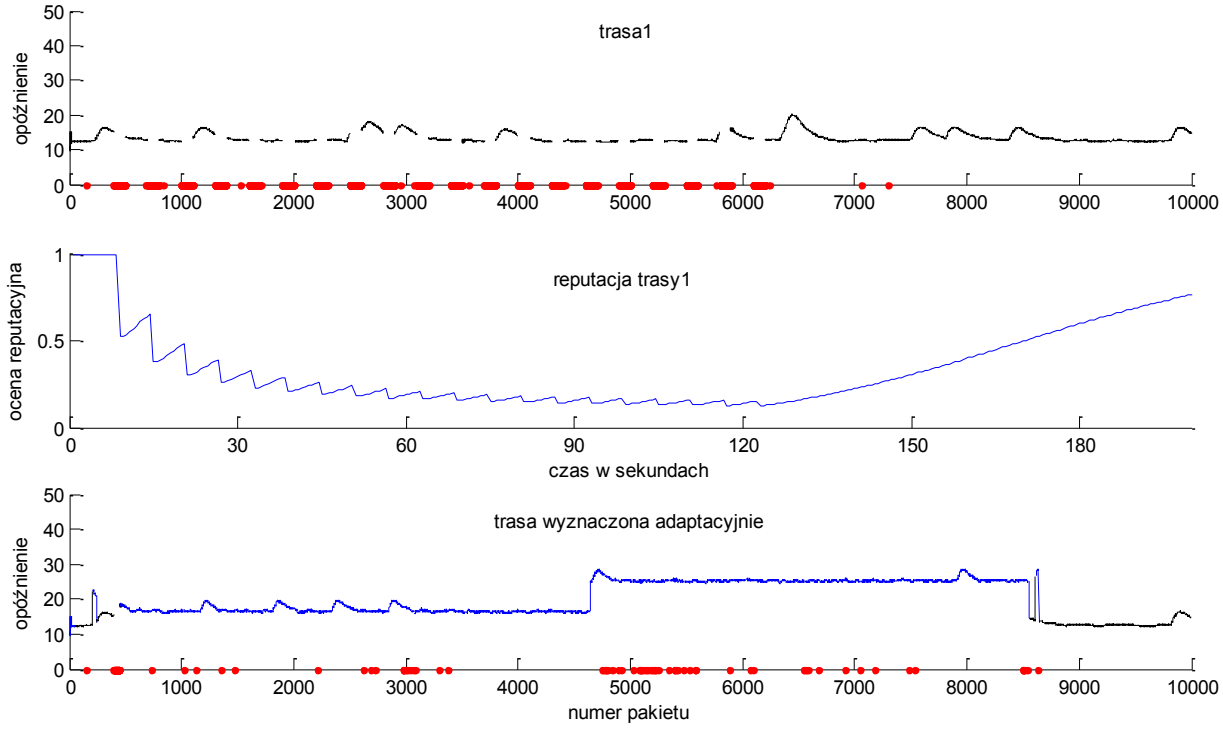


Rys. 9.13 Wykres zmiany reputacji w czasie dla linii cechujących się awariami o okresie a) 2000 b) 700 c) 350 oraz d) 600 i 1300 pakietów.

Skonfigurowany zgodnie z powyższymi parametrami system reputacyjny został przetestowany na dwóch wcześniejszych przypadkach. Rezultat zastosowania reputacji dla przykładu danego na rysunku 9.10 został pokazany na rysunku 9.14, natomiast wynikający z niej zysk przedstawiony w ostatniej kolumnie tablicy 9.2.



Rys. 9.14 Wynik działania mechanizmu adaptacyjnego wspomaganego reputacja linii dla przykładu przedstawionego na w rys.9.10.



Rys. 9.15 Wynik działania mechanizmu adaptacyjnego wspomaganego reputacją linii dla przykładu pesymistycznego, przedstawionego na rys.9.11.

Wyniki uzyskane dla pesymistycznego przypadku pokazanego na rysunku 9.11 po uwzględnieniu reputacji linii przedstawione zostały na rysunku 9.13, natomiast dane liczbowe znajdują się w tablicy 9.3.

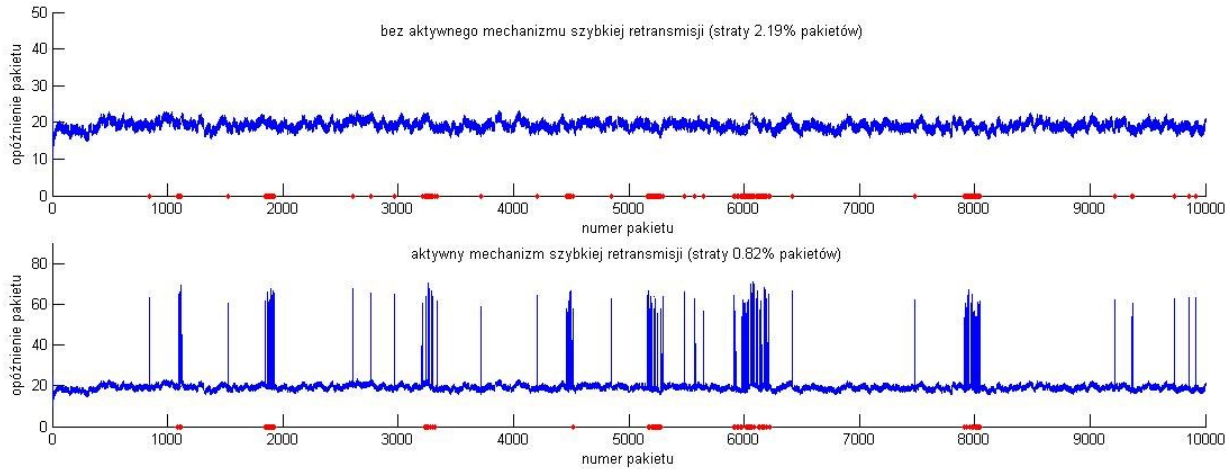
Tablica 9.3. Uzyskana jakość transmisji przy wykorzystaniu tras danych na rys. 9.11.

	Trasa1	**Trasa2**	**Adaptacyjna bez reputacji**	**Adaptacyjna z reputacją**
Średnia ocena PESQ	2.8225	3.9450	3.3330	3.9634
% utraconych pakietów	20.38	0.94	9.23	1.25
Średnie opóźnienie [ms]	14	22	15	20

Przedstawione powyżej przykłady wyraźnie obrazują celowość zastosowania mechanizmu reputacji dla połączeń międzywęzłowych. Mechanizm ten szczególnie dobrze sprawdza się w przypadku łącz o bardzo niestabilnej naturze, gdzie zaobserwowano 19% wzrost oceny jakości transmitowanego sygnału mowy względem routingu adaptacyjnego niewspomaganego reputacją.

d) Mechanizmy wspomagające routing.

Ponieważ zaproponowane w rozdziale o routingu mechanizmy wspomagające, takie jak FEC czy szybka retransmisja, nie stanowią oryginalnych rozwiązań opracowanych przez autora rozprawy, lecz zostały jedynie zaadaptowane do użycia w projekcie, nie zostanie im poświęcone dużo uwagi. Jednakże, ze względu na swoją szczególną użyteczność w infrastrukturze zastosowanej w projekcie, zaimplementowany i przetestowany został mechanizm szybkiej retransmisji pakietów. Przykład jego działania zaprezentowano na rysunku 9.16.

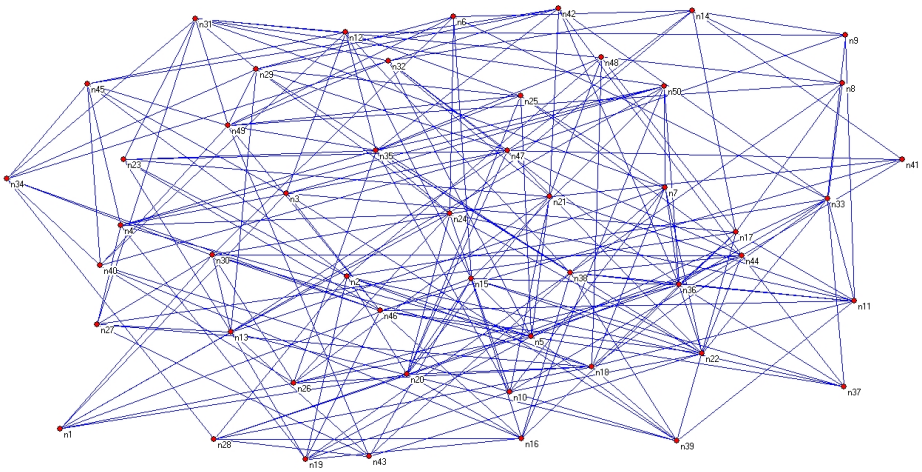


Rys. 9.16. Przykład zastosowania mechanizmu szybkiej retransmisji na linii międzywęzłowej

Przeprowadzone symulacje potwierdziły, iż mechanizm ten w pełni integruje się z innymi rozwiązaniami stosowanymi w projekcie i pozwala istotnie zwiększyć jakość transmisji (zdolny był on do odtwarzania od 60 do 90% traconych pakietów).

9.3.2 Testy na dużej infrastrukturze

Ze względu na mocno niedeterministyczny charakter sieci Internet, symulacja jej działania jest zagadnieniem bardzo złożonym [4]. Jednak możliwe jest stworzenie symulatorów (np. network NS2 [103]) pozwalających testować dany system pod względem konkretnych scenariuszy i warunków zbliżonych do tych panujących w rzeczywistej sieci. Prosty symulator tego typu, nastawiony konkretnie na zagadnienia związane z transmisją real-time, został opracowany przez autora niniejszej rozprawy (podrozdział 9.2) w celu przetestowania działania opisywanego systemu. Przy pomocy tego symulatora zostało wygenerowanych wiele możliwych struktur sieciowych dla różnych warunków. Przykładowa infrastruktura, na której przeprowadzona została większość przedstawionych poniżej symulacji, składa się z 50 węzłów oraz 233 linii międzywęzłowych i jest pokazana na rysunku 9.17. Jak można się spodziewać, wyniki symulacji zależą mocno do konkretnych cech scenariusza testowego. Jednakże na podstawie analizy znacznej liczby różnorodnych przypadków, zaobserwowano wiele prawidłowości wpływających na wynik końcowy.



Rys. 9.17 Model sieci użytej do testów (50 węzłów, 233 połączenia).

Podstawową z nich jest charakterystyka strat pakietów występujących na liniach. Zgodnie z opisem przedstawionym w podrozdziale dotyczącym modelu linii (9.2.1), linia taka może znajdować się w jednym z dwóch stanów: normalnym oraz w tzw. „stanie burst" (wzmożonego gubienia pakietów). Różnice pomiędzy pakietami traconymi w każdym z tych stanów istotnie wpływają na jakość działania systemu. W celu zademonstrowania tej zależności przygotowane zostały cztery scenariusze testowe:

Scenariusz 1.

Częste losowe straty pakietów poza stanem wzmożonych strat (stan burst), wysokie prawdopodobieństwo wejścia linii w stan burst oraz wysokie prawdopodobieństwo wyjścia z niego (liczne krótkie stany wzmożonych strat pakietów).

Scenariusz 2

Niższe (względem scenariusza 1) straty poza stanem burst. Rzadsze, ale dłużej trwające stany wzmożonej utraty pakietów.

Scenariusz 3.

Jeszcze niższe straty pakietów poza stanem burst (obecnie 0.1%, względem 1% w scenariuszu 1) oraz równie częste jak w scenariuszu 1 stany burst lecz trwające znacznie dłużej.

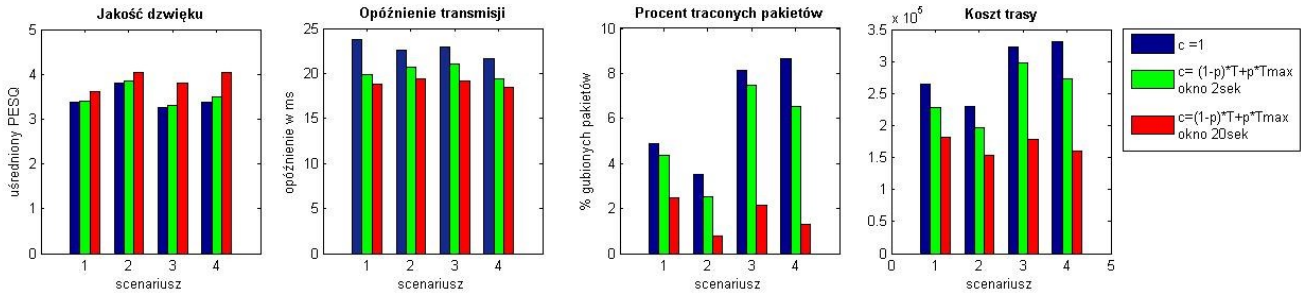
Scenariusz 4

Dziesięciokrotnie rzadsze, lecz także znacznie dłuższe stany burst w porównaniu do scenariusza 3.

Tablica 9.4 Parametry symulacji.

	S1	S2	S3	S4
loss_probability(1)	0.01	0.005	0.001	0.001
loss_probability(2)	0.4	0.4	0.5	0.5
transition_probability(1)	0.001	0.000125	0.001	0.0001
transition_probability(2)	0.1	0.00125	0.005	0.0005
% traconych pakietów	4.46	3.89	8.06	7.19

Każdy z powyższych scenariuszy został przetestowany poprzez analizę przesyłanych równolegle 60 strumieni audio trwających po 5 min, a następnie określenie parametrów transmisji. Porównane zostały także 3 metody routingu: oparta na mierze odległości, oparta na mierze kosztu z mechanizmem adaptacyjnym o czasie reakcji 2 s, oraz na tej samej mierze lecz z czasem reakcji 20 s. Wyniki zostały przedstawione na wykresach 9.18 a) b) c) oraz d).



Rys. 9.18 Wyniki symulacji dla scenariuszy 1-4.

Na podstawie analizy przypadków testowych pierwszym nasuwającym się wnioskiem jest to, że bardzo istotna jest szybkość, z jaką system reaguje na zmiany jakości transmisji. Okno pomiarowe o rozmiarze 20 sekund jest stanowczo za duże i nie pozwala odpowiednio reagować na krótsze, okresowe zaburzenia pojawiające się na linii. Kolejno można stwierdzić, że zgodnie z przewidywaniami routing adaptacyjny przynosi największy zysk w przypadku występowania dłuższych zaburzeń, tj. dla sieci ze scenariusza 3. Pomimo, iż sama sieć gubi niemal dwukrotnie więcej pakietów niż w scenariuszu 1 (odpowiednio 8.06% i 4.46%), to jednak mechanizm adaptacyjny zdołał osiągnąć lepsze wyniki. Analogiczny wniosek można wyciągnąć

z porównania scenariuszy 3 oraz 4. Najgorszy wynik został osiągnięty w scenariuszu 1, co spowodowane jest większymi niż w pozostałych trzech scenariuszach stratami poza stanem burst. Podstawowym wnioskiem wynikającym z powyższych symulacji jest fakt, iż routing adaptacyjny doskonale sprawdza się w wypadku dłuższych degradacji, lecz nie zapobiega stratom wynikającym ze zjawiska pojedynczego, niezależnego gubienia pakietów. Jest on zatem komplementarnym rozwiązaniem do mechanizmu szybkiej retransmisji, która najlepiej sprawdza się w drugim wypadku, ale jest mało efektywna podczas gdy linia jest w stanie burst.

Wpływ nieprzewidzianych awarii na jakość transmisji.

W wypadku sieci P2P, poza typowymi awariami łącza transmisyjnego pomiędzy węzłami na trasie routingu, częstym zjawiskiem jest także nagłe opuszczenie sieci przez jeden z węzłów. Jest to charakterystyczne dla otwartych sieci P2P i może być również interpretowane jako awaria łącza. Ponadto możliwy jest atak, mający na celu sabotowanie działania systemu, poprzez wprowadzanie częstych cyklicznych zaburzeń jakości transmisji. Aby przetestować zachowanie się systemu w przypadku nieprzewidzianych awarii linii międzywęzłowych, przygotowano trzy różne scenariusze testowe:

Scenariusz 1
10% linii ulega sporadycznym awariom (jedna lub dwie awarie w czasie symulacji). Awarie te są długotrwałe, ich okres wynosi przynajmniej 1 min.

Scenariusz 2
Jest rozszerzoną wersją scenariusza pierwszego: występuje dodatkowo 10% awaryjnych linii. W tym przypadku linie ulegają awariom częściej (średnio dwa razy na minutę), ale zaburzenia trwają znacznie krócej (10 sekund).

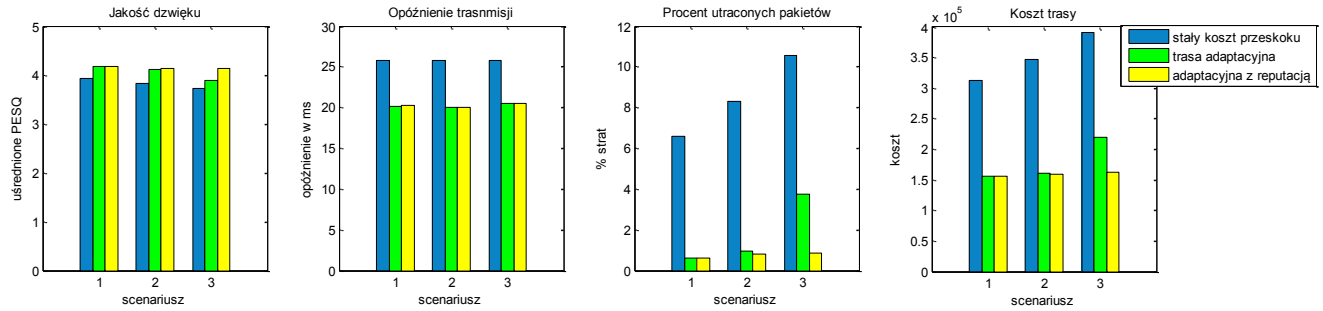
Scenariusz 3
Stanowi rozszerzenie scenariusza drugiego o dodatkowe 10% awaryjnych linii, tym razem o bardzo niestabilnej naturze. Linie te charakteryzują się bardzo silną cykliczną degradacją jakości połączenia. Spadek następuje co 3 sekundy, po czym po kolejnych 3 sekundach następuje powrót do wysokiej jakości transmisji.

Straty pakietów dla każdego z przypadków przedstawione zostały w tablicy 9.5.

Tablica 9.5. Średnie straty pakietów w scenariuszach 1-4.

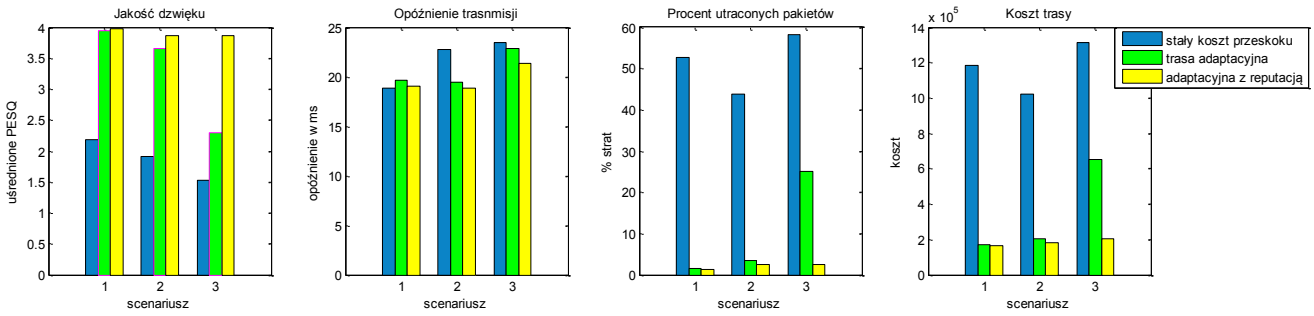
	Sieć bazowa	Scenariusz 1	Scenariusz 2	Scenariusz 3
% strat	1.16	4.5	7.4	11.7

Symulacja została przeprowadzona poprzez równoległą transmisję 60 strumieni audio. Uśrednione wyniki przedstawione zostały na rysunku 9.19. Do porównania wydajności użyto trzech różnych konfiguracji mechanizmu routingu: routingu na podstawie liczby przeskoków, routingu z wykorzystaniem mechanizmu adaptacyjnego oraz routingu z wykorzystaniem mechanizmu adaptacyjnego wspomaganego reputacją linii.



Rys. 9.19 Wyniki symulacji wpływu awarii linii międzywęzłowych na jakość transmisji.

Ponieważ duża część transmisji odbywała się z całkowitym pominięciem awaryjnych tras, aby lepiej zobrazować wpływ awarii na jakość transmisji na rysunku 9.20 przedstawione zostały wyniki dla 10% najgorszych transmisji.

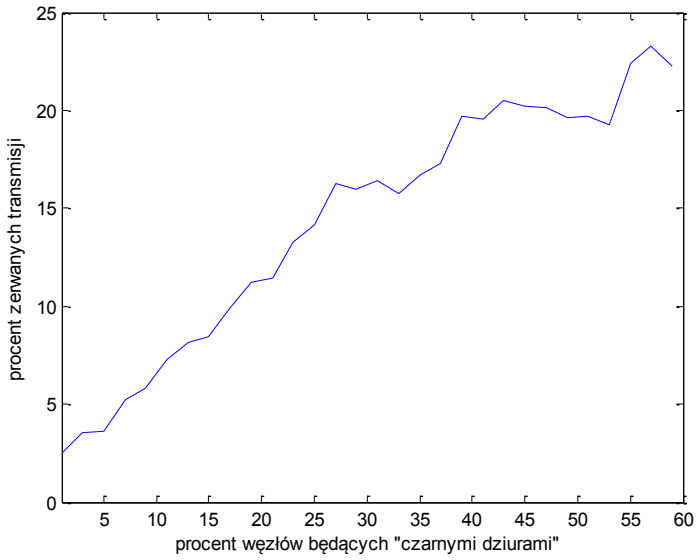


Rys. 9.20 Wyniki symulacji wpływu awarii linii międzywęzłowych na jakość transmisji dla 10% najgorszych przypadków.

Na postawie powyższych wyników można jednoznacznie stwierdzić, iż zaprojektowany system jest w stanie szybko i skutecznie reagować na nieprzewidziane awarie powstałe na trasach międzywęzłowych. W przypadku scenariusza pierwszego widoczny jest wyraźny zysk wynikający z szybkiej reakcji routingu adaptacyjnego (PESQ 3.97 względem wcześniejszych 2.18), natomiast reputacja wpływa w bardzo znikomym stopniu na wynik, ponieważ awarie nie są częste. Reakcja systemu na awarie w scenariuszu drugim także była oparta o proces adaptacji; tu widoczny jest bardzo niewielki wzrost jakości wynikający z użycia reputacji (1.05%). Znaczny zysk w jakości połączenia wynikający z użycia reputacji (odpowiednio 154% w wypadku routingu opartego o przeskoki i 68% względem routingu adaptacyjnego pozbawionego reputacji) można zaobserwować dopiero w scenariuszu trzecim, gdzie niestabilne linie otrzymały niską ocenę reputacyjną i zostały wykluczone z procesu routingu.

Wykrywanie „czarnych dziur" w routingu.

Atak typu „packet drop" polega na celowym gubieniu pakietów przez jeden z routerów na trasie routingu – router ten zwany wówczas jest czarną dziurą. Taki atak jest często bardzo groźny w skutkach i zazwyczaj ciężki do wykrycia. Jeśli węzeł/router odrzuca wszystkie nadsyłane do niego pakiety, to takie złośliwe zachowanie można szybko wykryć, gdyż zostanie ono potraktowanie na równi z awarią łącza. Jednakże sytuacja komplikuje się, gdy złośliwy węzeł poprawnie komunikuje się i przesyła prawidłowo wszystkie pakiety kontrolne, włącznie z prawdziwą informacją o stanie ścieżki, ale jednocześnie celowo odrzuca pakiety należące do transmisji, w której zgodnie z tablicami routingu powinien pośredniczyć. W sytuacji takiej kolejny węzeł na trasie, nie otrzymując oczekiwanych informacji od poprzednika, nie jest w stanie ocenić, czy pakiety zostały zablokowane w węźle bezpośrednio go poprzedzającym, czy gdziekolwiek wcześniej na ścieżce routingu. Symulacja tego zjawiska została przedstawiona na rysunku 9.21.

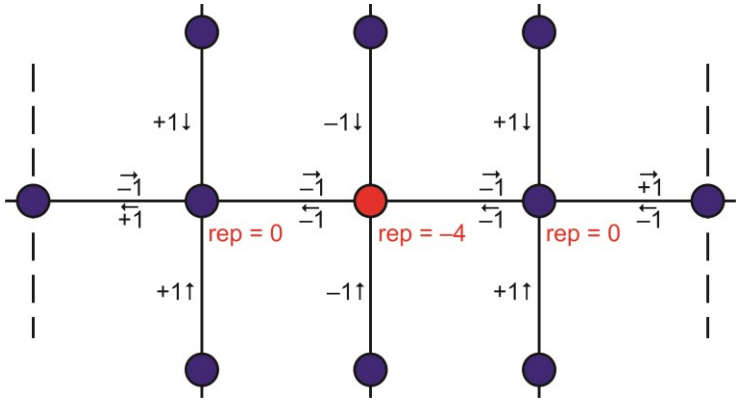


Rys. 9.21 Wpływ węzłów będących czarnymi dziurami na ilość zerwanych połączeń.

Ponieważ symulacja odbywała się na sieci testowej (rys. 9.17) oraz z użyciem transmisji real-time, znaczna część transmisji odbywa się bez udziału węzłów pośrednich. Mimo to, gdy węzły te stanowią 20%, wciąż potrafią zablokować 10% transmisji. Dodatkowe zagrożenie (szczególnie uciążliwe w wypadku sieci bezprzewodowych) stanowi fakt, iż jakość transmisji ma główny wpływ na wybór tras, a zatem złośliwy węzeł poprzez rozgłaszanie nieprawdziwych informacji o bardzo wysokiej jakości połączenia może najpierw przejąć znaczną część komunikacji, a następnie ją przerwać. W opracowanym w ramach niniejszej rozprawy systemie zjawisko przejęcia komunikacji jest jednak bardzo ograniczone w wyniku uśredniania informacji o jakości trasy pochodzących od węzłów z obu stron łącza, a także dzięki możliwości niecałkowitego ignorowania takich informacji od węzłów oznaczonych jako notoryczni kłamcy (niska reputacja rekomendacyjna).

W celu wykrycia czarnych dziur na trasie routingu autor postanowił opracować prosty system bazujący na reputacji węzłów. Ponieważ każda transmisja wiąże się z obustronnym przesyłaniem komunikatów pomiędzy stronami (gdzie w nagłówku pakietu zawarta jest informacja o ilości przesłanych pakietów, a sam pakiet kontrolny podpisany jest cyfrowo), agent znajdujący się w węźle ma za zadanie monitorować transmitowane strumienie i w wypadku nie wykrycia strumienia zwrotnego zaniżyć reputację węzła poprzedzającego go bezpośrednio na

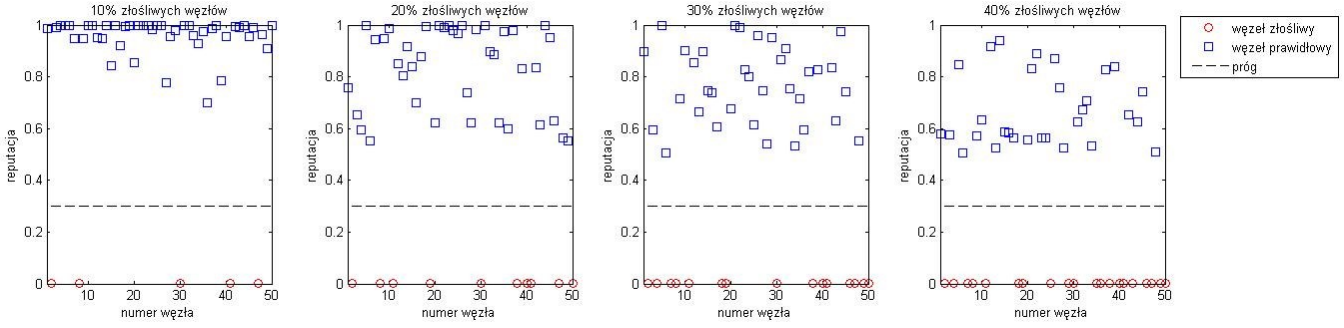
trasie routingu (mimo, iż nie musi on być odpowiedzialny za gubienie pakietów). W ten sposób każdy z węzłów uzyska niższą ocenę reputacyjną w odróżnieniu od sytuacji, gdy węzeł uczestniczy także w wielu innych poprawnych transmisjach i ocena zostaje jedynie nieznacznie obniżona. Natomiast sam węzeł będący czarną dziurą otrzyma znacznie niższą ocenę niż pozostałe. Idea działania algorytmu przedstawiona została na rysunku 9.22.



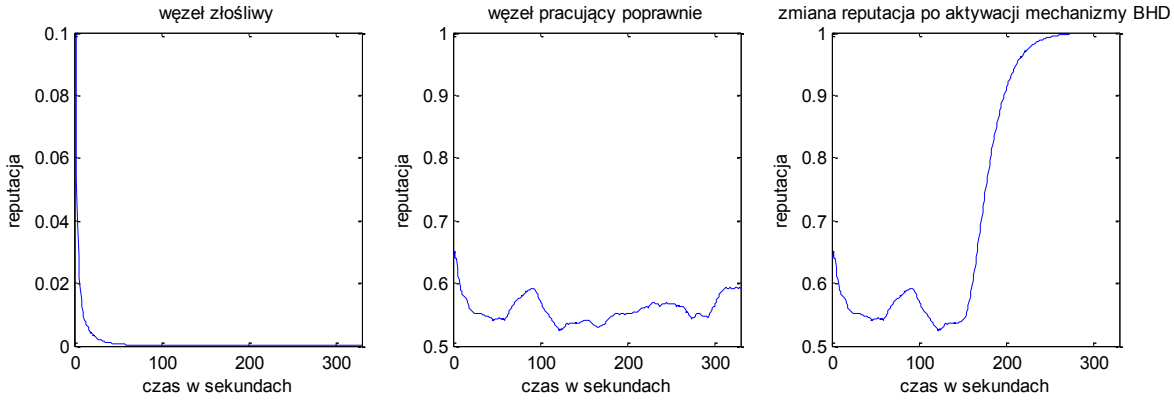
Rys. 9.22 Przykład działania mechanizmu wykrywania „czarnych dziur".

Algorytm ten sprawdza się nawet w wypadku, gdy złośliwy węzeł celowo wystawia nieprawdziwą, niską ocenę wszystkim swoim sąsiadom.

Symulacja działania algorytmu detekcji czarnych dziur stanowiących odpowiednio 10, 20, 30 oraz 40% wszystkich węzłów sieci przedstawiona została na rysunku 9.23, natomiast zmiana w czasie reputacji węzła będącego czarną dziurą oraz węzła działającego poprawnie została przedstawiona na rysunku 9.24



Rys. 9.23 Wynik działania mechanizmu wykrywania „czarnych dziur" w routingu.



Rys. 9.24 Reputacja węzła złośliwego, reputacja węzła pracującego poprawnie zaniżona w wyniku nieuczciwych ocen węzłów złośliwych (wykres w środku) oraz reputacja tego samego węzła po zablokowaniu węzłów będących czarnymi dziurami (w chwili t=150 s).

W każdym z powyższych przypadków algorytm zdołał bezbłędnie wykryć wszystkie węzły będące czarnymi dziurami. Na uwagę zasługuje również fakt, że nie zaobserwowano przy tym żadnych przypadków błędnej klasyfikacji.

Rozdział X

Podsumowanie i wnioski

Cel, jaki postawił przed sobą autor niniejszej rozprawy, to jest stworzenie systemu wspomagającego transmisję czasu rzeczywistego w Internecie jest zagadnieniem złożonym. Aby dobrze zdefiniować ten problem oraz poznać związane z nim ograniczenia, należy zrozumieć istotę pakietowej transmisji danych w Internecie. Wiedza ta jest niezbędna, aby zaprojektować odpowiednie mechanizmy routingu, bezpieczeństwa oraz kompensacji strat związanych z transmisją danych. Sam system z założenia powinien być możliwie niezawodny, a przy tym umożliwiać w przyszłości łatwą rozbudowę. Wymaga to już na etapie wstępnego projektowania architektury zadbania o dostarczenie mechanizmów umożliwiających obsługę nie tylko obecnie używanych usług (jak VoIP czy media strumieniowe), ale także usług przyszłych, które mogą charakteryzować się innymi priorytetami czy wymogami względem routingu oraz bezpieczeństwa.

Autor rozpoczął pracę od analizy wad i zalet systemów opisanych w literaturze, aby zdefiniować ich braki, a w ten sposób i wymogi względem własnego systemu. Analiza cech różnych typów transmisji czasu rzeczywistego przedstawiona w rozdziale III pozwoliła na zrozumienie ograniczeń związanych z maksymalnymi opóźnieniami i tolerancją na gubienie pakietów. Natomiast opracowany i opisany w podrozdziale 9.2.1 model łącza internetowego zobrazował istotę tych opóźnień i pozwolił przewidzieć skuteczność niektórych mechanizmów, jak na przykład szybkiej, międzywęzłowej retransmisji pakietów.

Modularność i łatwość rozbudowy systemu została zapewniona w szczególności poprzez dwa mechanizmy:

- System agentowy, który z założenia powinien pozwalać agentom (umieszczonym w każdym z węzłów sieci) na autonomiczność oraz możliwość wzajemnej komunikacji,

- Framework routingu kontekstowego (podrozdział 7.4): w miejsce ograniczenia się do jednego, np. najpowszechniej stosowanego protokołu routingu, autor wprowadził ideę „ocenianego portfolia protokołów routingu" oraz opracował mechanizmy pozwalające na automatyczny wybór najlepszego protokołu względem wymagań przedstawionych przez dany serwis lub grupę użytkowników. Podejście to pozwala nie tylko na jednoczesną, bardziej efektywną obsługę wielu typów serwisów i grup użytkowników, ale także na znacznie łatwiejsze dodawanie w przyszłości nowych metod routingu (wystarczy dodać nowy protokół do portfolio). Opracowany Framework pozwala także na dodanie dodatkowych, zwiększających jakość transmisji mechanizmów, jak wspomniana wcześniej szybka retransmisja pakietów lub kodowanie korekcyjne.

Bardzo istotnym zagadnieniem była kwestia bezpieczeństwa systemu. Szczególnie, że z założenia, w zależności od przyjętej konfiguracji, jego elementy mogą stanowić węzły o nieznanym poziomie zaufania, tworzone z anonimowych komputerów przyłączanych do sieci peer-to-peer. Autor postanowił nie tylko skorzystać ze standardowych technik kryptografii, zwanych hard-security, ale także z mechanizmów społecznych, w szczególności z reputacji, definiowanych jako metody soft-security. Decyzja o wzbogaceniu systemu o mechanizm reputacji wpłynęła na znaczne zwiększenie jego bezpieczeństwa. Symulacje zachowania mechanizmu w wypadku zagrożenia spowodowanego różnymi typami ataków zostały przedstawione w podrozdziale 6.6, 6.7.3 oraz 9.3.

Ze względu na stosunkowo rozbudowany charakter systemu, przy jego projektowaniu niezbędna była konsolidacja wielu mechanizmów, które same w sobie stanowią rozległe zagadnienia z zakresu informatyki. Aby system mógł funkcjonować poprawnie, autor musiał posłużyć się zarówno istniejącymi rozwiązaniami jak i zaproponować swoje własne. Poniżej przedstawiona została lista oryginalnych, opracowanych przez autora niniejszej pracy, rozwiązań:

- zaprojektowanie architektury stworzonego systemu,
- opracowanie mechanizmu self-healing, pozwalającego w wyniku rozproszonego głosowania wyznaczyć nowy serwer centralny w wypadku awarii/kompromitacji aktualnego, a następnie w bezpieczny sposób odbudować bazę danych (używając kryptograficznego protokołu dzielenia sekretu),
- dostosowanie systemu reputacyjnego opartego na rozkładzie beta do potrzeb projektu,

- zaprojektowanie algorytmu do wykrywania złośliwych koalicji w systemach reputacyjnych,
- opracowanie własnego Frameworku routingu kontekstowego bazującego na automatycznym wyborze najlepszego dostępnego protokołu routingu w zależności od danych kontekstowych (wymogi i rekomendacje użytkowników, operatora sieci oraz dostawcy danej usługi),
- implementacja symulatora służącego do przetestowania prawidłowości opracowanych rozwiązań.

Opisane powyżej decyzje, opracowane mechanizmy oraz przeprowadzone następnie testy pozwoliły w pełni potwierdzić tezę postawioną na wstępie rozprawy:

Opracowanie prostego w budowie systemu zdolnego do jednoczesnej obsługi wielu typów transmisji czasu rzeczywistego okazało się możliwe, a przeprowadzone symulacje potwierdziły jego prawidłowe funkcjonowanie.

Autorowi udało się zaprojektować założoną, hybrydową infrastrukturę P2P cechującą się szybkim czasem wyszukiwania ścieżek, co jest charakterystyczne dla infrastruktury scentralizowanej, a jednocześnie odpornością na problem SPOF (Single Point of Failure) cechującą zazwyczaj infrastruktury całkowicie rozproszone. System taki, dzięki opracowanemu mechanizmowi self-healing odporny jest na awarie i próby ataków, a poprzez wykorzystanie kryptograficznego schematu dzielenia sekretu możliwe stało się bezpieczne przechowywanie, synchronizacja i odbudowa, niezbędnej do ciągłej pracy infrastruktury, bazy danych.

Przeprowadzone w podrozdziale 9.3 symulacje na sieci składającej się z 50 węzłów dowiodły, iż opracowany system jest w stanie istotnie poprawić jakość połączenia. W zależności od warunków panujących w sieci, średni wzrost jakości transmisji VoIP względem systemu pozbawionego adaptacyjnego routingi i reputacji, oscylował pomiędzy 7 a 154%. Wysoki wzrost jakości połączenia, będący zasługą mechanizmu reputacji dla łącz międzywęzłowych, można było zaobserwować szczególnie w wypadku połączeń wysoce niestabilnych.

Wyniki badań przedstawione w niniejszej rozprawie są także potwierdzeniem faktu, iż wprowadzenie mechanizmu reputacyjnego jest w stanie istotnie zwiększyć bezpieczeństwo tego typu systemów. Przedstawione w rozdziale 6.6 symulacje dowodzą, iż poprzez wykorzystanie mechanizmu reputacji, system jest w stanie prawidłowo wykrywać i unieszkodliwiać wadliwe

węzły. Jest on także odporny na ataki typu Sybil (nawet gdy liczba atakujących dochodzi do 75% wszystkich oceniających) oraz na działania węzłów złośliwych, próbujących destabilizować pracę systemu, poprzez przesyłanie nieprawidłowych informacji o stanie łącz i zachowaniu innych węzłów (podrozdział 6.6.2). W przypadku tych ostatnich poziom detekcji wynosi 100%, gdy atakujący działają niezależnie. W przypadku, gdy atakujący będą formować koalicję, do jej unieszkodliwienia wykorzystywany jest mechanizm detekcji koalicji (opisany w podrozdziale 6.7), zdolny zazwyczaj do wykrycia od 65% do 100% węzłów ją tworzących (wyniki zależne od stopnia maskowanie się koalicji). Wysoka użyteczność reputacji w wykrywaniu „czarnych dziur" na trasach routingu została przedstawiona w podrozdziale 9.3, gdzie symulacje dowiodły 100% skuteczności detekcji.

Poza przykładami opisanymi w powyższej rozprawie, istotny jest także fakt, iż niektóre z opracowanych w ramach pracy rozwiązań, jak mechanizm self-healing dla infrastruktur P2P czy algorytm wykrywania złośliwych koalicji, cechują się znacznie szerszym obszarem zastosowań i mogą być z powodzeniem stosowane w wielu innych projektach, niekoniecznie związanych z transmisją czasu rzeczywistego. Z przeprowadzonych przez autora wstępnych symulacji wynika, iż zaproponowany algorytm detekcji koalicji okazuje się skuteczny przy wykrywaniu zmów mających na celu dyskredytację konkurencji w internetowych serwisach aukcyjnych. Natomiast mechanizm self-healing może posłużyć do budowy szerokiej gamy systemów internetowych, w szczególności mogących działać całkowicie autonomicznie, bez konkretnego nadzorcy, gdzie wszelkie decyzje związane z jej funkcjonowaniem podejmowane są w wyniku głosowania [86].

Bibliografia

[1] D. Cohen, "Specifications for the Network Voice Protocol," *Technical Report No. ISI/RR-75-39,* 1976.

[2] R. Braden, D. Clark and S. Shenker, "Integrated Services in the Internet Architecture: an Overview," *RFC 1633,* 1994.

[3] D. Grossman, "New Terminology and Clarifications for Diffserv," *RFC 3260,* 2002.

[4] M. Pióro and D. Medhi, "Routing, flow, and capacity design in communication and computer networks," 2004.

[5] V. Paxson, "End-to-end Internet packet dynamics," *ACM SIGCOMM Computer Communication Review,* vol. 27, no. 4, pp. 139-152, 1997.

[6] D. G. Andersen, A. C. Snoeren and H. Balakrishnan, "Best-path vs. multi-path overlay routing," *Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement,* vol. ACM, 2003.

[7] A. Markopoulou, T. Fouad and M. Karam, "Loss and delay measurements of internet backbones," *Computer Communications 29,* pp. 1590-1604.

[8] S. Sundaresan, W. D. Donato, N. Feamster, R. Teixeira, S. Crawford and e. al., "Broadband Internet Performance: A View From the Gateway," *ACM SIGCOMM,* pp. 134-145, 2013.

[9] "One way transmission time," *ITU-T Recommendation G.114,* 2003.

[10] J. H. Gross and D. M. Etter, "Comparison of echo cancellation algorithms for the adaptive delay filter," *IEEE 42nd Vehicular Technology Conference ,* pp. 574-576, 1992.

[11] T. R. Henderson and R. H. Katz, "Transport protocols for Internet-compatible satellite networks," *IEEE Journal on Selected Areas in Communications,* vol. 17, no. 2, pp. 326--344, 1999.

[12] B. Quinn and K. Almeroth, "IP Multicast Applications: Challenges and Solutions," *RFC 3170,* 2001.

[13] M. Hefeeda, A. Habib, B. Botev, D. Xu and B. Bhargava, "PROMISE: Peer-To-Peer Media Streaming," 2003.

[14] F. Picconi and L. Massoulié, "Is there a future for mesh-based live video streaming?," *Eighth International Conference on Peer-to-Peer Computing ,* pp. 289-298, 2008.

[15] H. Xiaojun, L. Yong and K. Ross, "IPTV over P2P streaming networks: the mesh-pull approach," *IEEE Communications Magazine,* vol. 46, no. 2, pp. 86-92, 2008.

[16] K. Almeroth and M. Ammar, "The use of multicast delivery to provide a scalable and interactive video-on-demand service," *IEEE Journal on Selected Areas in Communications,* vol. 14, no. 6, pp. 1110-1122, 2004.

[17] S. Sengodan and V. O. K. Li, "A shared buffer Architecture for Interactive VOD servers," *Proceedings IEEE Sixteenth Annual Joint Conference of the IEEE Computer and Communications Societies,* vol. 3, pp. 1341-1348, 1997.

[18] T. Henderson, "The effects of relative delay in networked games," *PhD thesis, University College London,* 2003.

[19] J. Rosenberg, R. Mahy, P. Matthews and D. Wing, "Session Traversal Utilities for NAT," *RFC 5389,*

2008.

[20] Z. Hu, "NAT traversal techniques and peer-to-peer applications," *Telecommunications Software and Multimedia Laboratory / HUT T-110.551 Seminar on Internetworking,* pp. 04-26, 2005.

[21] G. Oryńczak and Z. Kotulski, "Agent based infrastructure for real-time applications," *Annales UMCS, Informatica,* vol. 11, no. 4, pp. 33-47, 2011.

[22] G. Oryńczak and Z. Kotulski, "Non cryptographic methods for improving real time transmission security and integrity," *Annales UMCS, Informatica,* vol. 11, no. 3, pp. 71-86, 2011.

[23] A. Wierzbicki, "Trust and fairness in open, distributed systems," *Heidelberg: Springer,* vol. 298, 2010.

[24] E. K. Lua, J. Crowcroft, M. Pias, R. Sharma and S. Lim, "A survey and comparison of peer-to-peer overlay network schemes," *EEE Communications Surveys & Tutorials,* vol. 7, no. 2, pp. 72-93, 2005.

[25] S. A. Baset and H. Schulzrinne, "An analysis of the skype peer-to-peer internet telephony protocol," *Technical Report CUCS-039-04 / arXiv preprint cs/0412017,* 2004.

[26] C. Wang and B. Li, "Peer-to-peer overlay networks: A survey," *Department of Computer Science, The Hong Kong University of Science and Technology,* 2003.

[27] I. Stoica, R. Morris, D. Karger, F. Kaashoek and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup service for Internet applications," *ACM SIGCOMM Computer Communication Review,* vol. 31, no. 4, pp. 149-160, 2001.

[28] S. Ratnasamy, P. Francis, M. Handley, R. Karp and S. Shenker, "A scalable content addressable network," *Proceedings of ACM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications SIGCOMM'01,* vol. 31, no. 4, 2001.

[29] "Merriam-Webster Online Dictionary," *available from http://www.m-w.com/,* accessed August 2014.

[30] J. Liang, R. Kumar, Y. Xi and K. W. Ross, "Pollution in P2P file sharing systems," *INFOCOM 2005. Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies,* vol. 2, pp. 1174-1185, 2005.

[31] S. Marti and H. Garcia-Molina, "Taxonomy of trust: Categorizing P2P reputation systems," *Computer Networks,* vol. 50, no. 4, pp. 472-484, 2006.

[32] K. Hoffman, D. Zage and C. Nita-Rotaru, "A survey of attack and defense techniques for reputation systems," *ACM Computing Surveys (CSUR),* vol. 42, no. 1, 2009.

[33] Y. Wang and J. Vassileva, "Trust and reputation model in peer-to-peer networks," *Proceedings Third International Conference on Peer-to-Peer Computing,* pp. 150-157, 2003.

[34] A. Jøsang, "A logic for uncertain probabilities," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems ,* vol. 9, no. 03, pp. 279-311, 2001.

[35] J. Sabater and C. Sierra, "Reputation and social network analysis in multi-agent systems," *Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 1,* vol. ACM, 2002.

[36] A. Schlosser, M. Voss and L. Bruckner, "Comparing and evaluating metrics for reputation systems by simulation," *A Workshop on Reputation in Agent Societies,* 2004.

[37] L. Page, S. Brin, R. Motwani and T. Winograd, "The PageRank citation ranking: Bringing order to the web," 1999.

[38] S. D. Kamvar, M. T. Schlosser and H. Garcia-Molina, "The eigentrust algorithm for reputation management in p2p networks," *Proceedings of the 12th international conference on World Wide*

Web, pp. 640-651, 2003.

[39] A. Jsang and R. Ismail, "The beta reputation system," *Proceedings of the 15th bled electronic commerce conference,* pp. 41-55, 2002.

[40] J. Liu and V. Issarny, "Enhanced reputation mechanism for mobile ad hoc networks," *Trust management,* pp. 48-62, 2004.

[41] A. Whitby, A. Jøsang and J. Indulska, "Filtering out unfair ratings in bayesian reputation systems," *Proc. 7th Int. Workshop on Trust in Agent Societies,* vol. 6, 2004.

[42] B. N. Levine, C. Shields and N. B. Margolin, "A survey of solutions to the sybil attack," *University of Massachusetts Amherst, Amherst, MA,* 2006.

[43] G. Ciccarelli and R. L. Cigno, "Collusion in peer-to-peer systems," *Computer Networks,* vol. 55, no. 15, pp. 3517-3532, 2011.

[44] G. Sukthankar and K. Sycara, "Robust recognition of physical team behaviors using spatio-temporal models," *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems,* pp. 638-645, 2006.

[45] R. Kerr and R. Cohen, "Detecting and identifying coalitions," *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems,* vol. 3, pp. 1363-1364, 2012.

[46] G. Oryńczak and Z. Kotulski, "On a mechanism of detection of coalitions for reputation systems in P2P networks," *International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS 2014) IEEE,* pp. 578-584, 2014.

[47] A. Strehl and J. Ghosh, "Cluster ensembles - a knowledge reuse framework for combining multiple partitions," *The Journal of Machine Learning Research,* vol. 3, pp. 583-617.

[48] V. Jacobson, R. Frederick, S. Casner and H. Schulzrinne, "RTP: A transport protocol for real-time applications," vol. The Internet Engineering Task Force, 2004.

[49] N. Kushman, S. Kandula and D. Katabi, "Can you hear me now?!: it must be BGP," *ACM SIGCOMM Computer Communication Review,* vol. 37, no. 2, pp. 75-84, 2007.

[50] A. P. Markopoulou, F. A. Tobagi and M. J. Karam, "Assessing the quality of voice communications over internet backbones," *IEEE/ACM Transactions on Networking (TON),* vol. 5, pp. 747-760, 2003.

[51] C. Aurrecoechea, A. Campbell and L. Hauw., "A survey of QoS architectures," *Multimedia systems,* vol. 6, no. 3, 1998.

[52] I. Aktas, F. Schmidt and E. Weingärtner, "An adaptive codec switching scheme for SIP-Based VoIP," *Internet of Things, Smart Spaces, and Next Generation Networking,* pp. 347-358, 2012.

[53] D. Wing, "Symmetric RTP/RTP Control Protocol (RTCP)," *RFC4961,* 2007.

[54] W. Mazurczyk and Z. Kotulski, "Adaptive voip with audio watermarking for improved call quality and security," *Journal of Information Assurance and Security,* vol. 2, no. 3, pp. 226-234, 2007.

[55] W. Jiang and H. Schulzrinne, "Comparison and optimization of packet loss repair methods on VoIP perceived quality under bursty loss," *Proceedings of the 12th international workshop on Network and operating systems support for digital audio and video.,* pp. 73-81, 2002.

[56] S. B. Wicker and V. K. Bhargava, "Reed-solomon forward error correction (FEC) schemes," *Reed-Solomon codes and their applications,* 1999.

[57] D. Andersen, H. Balakrishnan, F. Kaashoek and R. Morris, "Resilient Overlay Networks," *Proceedings of the eighteenth ACM symposium on Operating systems,* 2001.

[58] M. Gary and D. Johnson, "Computers and intractability: a guide to the theory of NP-completeness," *WH Freman and Co,* 1979.

[59] R. G. Garroppo, S. Giordano and L. Tavanti, "A survey on multi-constrained optimal path computation: Exact and approximate algorithms," *Computer Networks,* vol. 54, no. 17, pp. 3081-3107, 2010.

[60] F. Kuipers, P. Van Mieghem, T. Korkmaz and M. Krunz, "An overview of constraint-based path selection algorithms for QoS routing," *IEEE Communications Magazine,* vol. 40, no. 12, 2002.

[61] B.-L. Wenning, D. Pesch, A. Timm-Giel and C. Görg, "Environmental monitoring aware routing in wireless sensor networks," *Wireless and Mobile Networking,* pp. 5-16, 2008.

[62] Q. Ma and P. Steenkiste, "On path selection for traffic with bandwidth guarantees," *International Conference on Network Protocols,* pp. 191-202, 1997.

[63] V. Ribeiro, R. Riedi, R. Baraniuk, J. Navratil and L. Cottrell, "pathchirp: Efficient available bandwidth estimation for network paths," *Passive and active measurement workshop,* vol. 4, 2003.

[64] J. Manish and C. Dovrolis, "Pathload: A measurement tool for end-to-end available bandwidth," *In Proceedings of Passive and Active Measurements (PAM) Workshop,* 2002.

[65] R. L. Carter and M. E. Crovella, "Dynamic server selection using bandwidth probing in wide-area networks," *Proceedings of IEEE INFOCOM,* pp. 123-128, 1997.

[66] E. Goldoni and M. Schivi, "End-to-end available bandwidth estimation tools, an experimental comparison," *Traffic Monitoring and Analysis,* pp. 171-182, 2010.

[67] Y. Amir, C. Danilov, S. Goose, D. Hedqvist and A. Terzis, "An overlay architecture for high-quality VoIP streams," *IEEE Transactions on Multimedia,* vol. 8, no. 6, pp. 1250-1262, 2006.

[68] S. Tao, K. Xu, A. Estepa, T. F. L. Gao, R. Guerin, J. Kurose, D. Towsley and Z.-L. Zhang, "Improving VoIP quality through path switching," *INFOCOM 2005. Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies,* vol. 4, pp. 2268--2278, 2005.

[69] L. Ding and R. A. Goubran, "Speech quality prediction in VoIP using the extended E-model," *Global Telecommunications Conference GLOBECOM'03,* pp. 3974-3978, 2003.

[70] M.Fiedler, T.Hossfeld and P.Tran-Gia, "A generic quantitative relationship between quality of experience and quality of service," *Network,* vol. 24, no. 2, 2010.

[71] X. Zhang, J. Liu, B. Li and T.-S. P. Yum, "CoolStreaming/DONet: a data-driven overlay network for peer-to-peer live media streaming," *Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies,* vol. 3, p. 2102–2111, 2005.

[72] X. Hei, Y. Liu and K. W. Ross, "Inferring network-wide quality in P2P live streaming systems," *Selected Areas in Communications, IEEE,* vol. 25, no. 9, pp. 1640-1654, 2007.

[73] S. L. Blond, F. L. Fessant and E. L. Merrer, "Choosing partners based on availability in P2P networks," *ACM Transactions on Autonomous and Adaptive Systems (TAAS),* vol. 7, no. 2, p. 25, 2012.

[74] Y. Li, D. Ren, S. G. Chan and A. C. Begen, "Low-delay mesh with peer churns for peer-to-peer streaming," *IEEE International Conference on Multimedia and Expo,* p. 1546–1547, 2009.

[75] S. Yildirim, M. Sayit and G. Kardas, "A belief-desire-intention agent architecture for partner selection in peer-to-peer live video streaming applications," *Expert Systems,* 2014.

[76] G. Oryńczak and Z. Kotulski, "Context-Aware Secure Routing Protocol for Real-Time Services," *Cryptography and Security Systems. Springer Berlin Heidelberg,* pp. 193-207, 2014.

[77] K. Wrona and L. Gomez, "Context-aware security and secure context-awareness in ubiquitous computing environments," *Annales UMCS, Informatica,* vol. 4, no. 1, pp. 332-348, 2006.

[78] O. Chipara, Z. He, G. Xing, Q. Chen, X. Wang, C. Lu, J. Stankovic and T. Abdelzaher, "Real-time

power-aware routing in sensor networks," *14th IEEE International Workshop on Quality of Service,* pp. 83-92, 2006.

[79] M. G. Zapata and N. Asokan, "Securing ad hoc routing protocols," *Proceedings of the 1st ACM workshop on Wireless security,* pp. 1-10, 2002.

[80] A. Nafaa, T. Taleb and L. Murphy, "Forward error correction strategies for media streaming over wireless networks," *IEEE Communications Magazine,* vol. 46, no. 1, pp. 72-79, 2008.

[81] Z. Li and P. Mohapatra, "QRON: QoS-aware routing in overlay networks," *IEEE Journal on Selected Areas in Communications,* vol. 22, no. 1, pp. 29-40, 2004.

[82] Y. Amir, C. Danilov, S. Goose, D. Hedqvist and A. Terzis, "1-800-OVERLAYS: using overlay networks to improve VoIP quality," *roceedings of the international workshop on Network and operating systems support for digital audio and video.,* pp. 51-56, 2005.

[83] D. Goldschlag, M. Reed and P. Syverson, "Onion routing," *Communications of the ACM,* vol. 42, no. 2, pp. 39--41, 1999.

[84] Y. Zhang, L. Xu and X. Wang, "A cooperative secure routing protocol based on reputation system for ad hoc networks," *Journal of Communications,* vol. 3, no. 6, pp. 43-50, 2008.

[85] F. Seredynski and P. Bouvry, "Anomaly detection in TCP/IP networks using immune systems paradigm," *Computer Communications,* vol. 30, no. 4, pp. 740-749, 2007.

[86] G. Oryńczak and Z. Kotulski, "Notary-based self-healing mechanism for centralized peer-to-peer infrastructures," *Annales UMCS Informatica,* vol. 12, no. 4, pp. 97-112, 2012.

[87] F. Callegati, W. Cerroni and M. Ramilli, "Man-in-the-middle attack to the HTTPS protocol," *IEEE Security and Privacy,* vol. 7, no. 1, pp. 78-81, 2009.

[88] D. Wendlandt, D. G. Andersen and A. Perrig, "Perspectives: Improving SSH-style host authentication with multi-path probing," *Proceedings of USENIX Annual Technical Conference,* pp. 321-334, 2008.

[89] D. K. Gifford, "Weighted Voting for Replicated Data," *Proceedings of the seventh ACM symposium on Operating systems principles,* pp. 150-162, 1979.

[90] M. Schulze, "A new monotonic, clone-independent, reversal symmetric, and condorcet-consistent single-winner election method," *Social Choice and Welfare,* vol. 2, pp. 267-303, 2011.

[91] P. Emerson, "The original Borda count and partial voting," *Social Choice and Welfare,* vol. 40, no. 2, pp. 353-358, 2013.

[92] C. Neuman, S. Hartman, T. Yu and K. Raeburn, "The Kerberos network authentication service (V5)," *Network,* vol. 6649, p. 6806, 2005.

[93] E. D. Karnin, J. Greene and M. E. Hellman, "On secret sharing systems," *IEEE Transactions on Information Theory,* vol. 29, no. 1, pp. 35-41, 1983.

[94] R. Salami, C. Laflamme, B. Bessette and J.-P. Adoul, "ITU-T G. 729 Annex A: Reduced complexity 8 kb/s CS-ACELP codec for digital simultaneous voice and data," *IEEE Communications Magazine,* vol. 35, no. 9, pp. 56-63.

[95] E. J. Daniel, C. M. White and K. A. Teague, "An interarrival delay jitter model using multistructure network delay characteristics for packet networks," *Conference Record of the Thirty-Seventh Asilomar Conference on Signals, Systems and Computers,* vol. 2, pp. 1738-1742, 2004.

[96] L. Rizo-Dominguez, D. Torres-Roman, D. Munoz-Rodriguez and C. Vargas-Rosales, "Jitter in IP networks: a cauchy approach," *Communications Letters IEEE,* vol. 14, no. 2, pp. 190-192, 2010.

[97] E. N. Gilbert, "Capacity of a Burst-Noise Channel," *Bell system technical journal,* vol. 39, no. 5, pp.

1253-1265, 1960.

[98] E. O. Elliott, "Estimates of Error Rates for Codes on Burst-Noise Channels," *Bell system technical journal,* vol. 42, no. 5, pp. 1977-1997, 1963.

[99] W. Jiang and H. Schulzrinne, "Modeling of packet loss and delay and their effect on real-time multimedia service quality," *PROCEEDINGS OF NOSSDAV'2000,* 2000.

[100] G. Haßlinger and O. Hohlfeld, "The Gilbert-Elliott model for packet loss in real time services on the Internet," *14th GI/ITG Conference - Measuring, Modelling and Evaluation of Computer and Communication Systems (MMB),* pp. 1-15, 2008.

[101] Ł. Apiecionek, "Metoda oceny jakości transmisji głosowej w telefonii VoIP," *Rozprawa doktorska, Instytut Podstawowych Problemów Techniki Polskiej Akademii Nauk,* 2010.

[102] Y. Amir, C. Danilov, R. Musualoiu-Elefteri and N. Rivera, "The Spines Overlay Network". *dostępny na http://www. spines. org.*

[103] I. Teerawat and E. Hossain, "Introduction to network simulator NS2," *Springer Science & Business Media,* 2011.

[104] X. Chen and e. al., "Survey on QoS management of VoIP," *International Conference on Computer Networks and Mobile Computing IEEE,* 2003.

Spis rysunków

Spis tablic