

ROZPRAWA DOKTORSKA

---

**Zapewnianie jakości transmisji  
danych w sieci komputerowej w  
sposób zdecentralizowany z  
zastosowaniem metod agentowych  
oraz uczenia maszynowego**

---

*Autor:*

mgr inż. Dariusz ŻELASKO

*Promotor:*

dr hab. inż. Paweł PŁAWIAK,  
prof. PK i IITiS PAN

7 grudnia 2021



# Spis treści

|  |            |
|--|------------|
| <b>Wstęp</b>   | <b>1</b>   |
| <b>1 Metody zapewniania jakości</b>  | <b>9</b>   |
| 1.1 Technologie sieciowe . . . . .   | 9          |
| 1.2 QoX . . . . .  | 16         |
| 1.3 Ruting . . . . .   | 18         |
| 1.4 Zastosowania technik zapewniania jakości . . . . .                             | 23         |
| <b>2 Agenty</b>  | <b>27</b>  |
| 2.1 Koncepcja agentowości . . . . .  | 27         |
| 2.2 Alokacja dóbr . . . . .  | 30         |
| 2.3 Agenty w sieciach komputerowych . . . . .                                      | 34         |
| <b>3 Materiały i metody</b>  | <b>37</b>  |
| 3.1 Koncepcja Pay&Require . . . . .  | 37         |
| 3.2 Projekt i realizacja emulatora . . . . .                                       | 44         |
| 3.3 Symulacja z wykorzystaniem NS3 . . . . .                                       | 49         |
| 3.4 Zastosowanie agentów . . . . .   | 52         |
| 3.5 Klasyfikacja jakości transmisji z wykorzystaniem uczenia maszynowego . . . . . | 53         |
| <b>4 Eksperymenty</b>  | <b>57</b>  |
| 4.1 Emulacja . . . . .   | 57         |
| 4.2 Symulacja . . . . .  | 74         |
| 4.3 Aukcje . . . . .   | 78         |
| 4.3.1 Jednokrotne aukcje jednego dobra . . . . .                                   | 78         |
| 4.3.2 Wielokrotne aukcje jednego dobra . . . . .                                   | 83         |
| 4.3.3 Aukcje drugiej ceny . . . . .  | 85         |
| 4.3.4 Aukcje kombinatoryczne . . . . .   | 86         |
| 4.4 Uczenie maszynowe . . . . .  | 87         |
| 4.4.1 Etap I . . . . .   | 88         |
| 4.4.2 Etap II . . . . .  | 98         |
| 4.4.3 Etap III . . . . .   | 110        |
| <b>Podsumowanie</b>  | <b>119</b> |
| Oryginalne elementy pracy . . . . .  | 123        |
| Kierunki dalszych badań . . . . .  | 124        |
| <b>Publikacje autora</b>   | <b>125</b> |

|                      |            |
|----------------------|------------|
| <b>Spis rysunków</b> | <b>127</b> |
| <b>Spis tabel</b>    | <b>131</b> |
| <b>Bibliografia</b>  | <b>133</b> |



# Wstęp

Na przestrzeni czasu obserwować można ciągły proces rozwoju techniki. Powstają nowe urządzenia, maszyny oraz technologie, a te istniejące dotychczas są modyfikowane i unowocześniane. Postęp techniczny to proces, do którego społeczeństwo przywykło i nauczyło czerpać z niego korzyści. Jednym z istotnych elementów wspomnianego postępu są urządzenia technologii komputerowej. Bez nich zmiany, które można obserwować nie byłyby możliwe. Równocześnie z rozwojem komputerów zauważono potrzebę wymiany informacji pomiędzy maszynami. Tak właśnie, powstała koncepcja sieci komputerowej umożliwiającej transmisję danych na odległość. Za moment, w którym zaczęto prace nad koncepcją sieci komputerowej przyjmuje się rok 1958. To właśnie w tym roku powstała agencja ARPA (Advanced Research Project Agency). ARPA to agencja utworzona w USA, na potrzeby prowadzenia projektów związanych z zastosowaniami militarnymi. W roku 1962 zauważono potrzebę opracowania metody połączenia komputerów bez wykorzystania punktu centralnego. Takie podejście miało zapewnić możliwość funkcjonowania sieci mimo jej częściowego uszkodzenia w wyniku działań militarnych. W 1963 roku zauważono konieczność opracowania sieci działającej na większą skalę niż tylko terytorium USA.

Głównym założeniem sieci komputerowej jest możliwość przekazywania informacji na odległość. Koncepcja transmisji z wykorzystaniem przewodów sięga roku 1876, kiedy to szkocki naukowiec Alexander Bell opatentował telefon. Sieć telefoniczna była przykładem sieci z komutacją obwodów. Centrale telefoniczne obsługiwane przez personel łączyły w sposób fizyczny obwody zestawiając pożądane połączenie. W ten sposób sygnał mógł być przekazywany pomiędzy stronami (nadawcą i odbiorcą). Wszelkie problemy fizyczne związane z transmisją powodowały przerwanie połączenia – stąd właśnie określenie sieć połączeniowa. Zaletą, ale i równocześnie wadą komutacji obwodów było zajmowanie całego łącza przez jedno fizyczne połączenie. Stanowiło to spore ograniczenie, jak chodzi o wydajność całego rozwiązania. Już na etapie komutacji obwodów zdefiniować można było pojęcie jakości usługi. W przypadku tego typu sieci, ze względu na wykorzystanie fizycznych obwodów na potrzeby połączenia, usługa była świadczona na stałym

poziomie jakości. Wynikało to z faktu, iż fizyczny obwód wykorzystywany był wyłącznie przez jedno połączenie. Co za tym idzie nie występowały zakłócenia transmisji. Stała jakość usługi była niewątpliwą zaletą komutacji obwodów. Niestety taki sposób wykorzystania fizycznych obwodów był niewydajny.

Zupełnie innym podejściem było to zaproponowane w roku 1961 przez Paula Barana. Zasada ta określona została przez jej autora mianem „gorącego kartofla”. Transmisja pomiędzy węzłami sieci polegała na przekazywaniu danych na podstawie informacji zawartej w nagłówku – następowała weryfikacja i przekazanie danych. Co istotne dane pierwotne były dzielone na mniejsze fragmenty zwane pakietami. Każdy pakiet mógł być przekazywany zupełnie różnymi ścieżkami. Podejście to stanowi podwaliny dla współczesnych zasad działania sieci komputerowych tj. dla sieci Ethernet/IP.

Za faktyczny początek istnienia Internetu przyjęto rok 1969 – to właśnie w tym roku powstały pierwsze węzły sieci ARPANET. Sieć ta powstała na potrzeby obronności USA. Na początku ARPANET składała się tylko z kilku węzłów pomiędzy którymi możliwa była transmisja danych. W latach 70-tych XX wieku wysłana została pierwsza wiadomość e-mail. W tej dekadzie ARPANET stała się siecią o zasięgu międzynarodowym. Kolejne lata przyniosły wzrost popularności sieci, a prawdziwą rewolucję przyniosło opracowanie technologii stron internetowych i powstanie pierwszej wyszukiwarki internetowej. Powstanie modelu TCP/IP przez DARPA stało się podstawą struktury współczesnych sieci – wykorzystano bramy dokonujące konwersji pakietów pomiędzy różnymi sieciami.

Od samego początku jednym z kluczowych problemów związanych z funkcjonowaniem sieci komputerowej było zapewnienie odporności na awarie. To właśnie w tym celu zaczęto w sieciach wykorzystywać komutację pakietów. Dane dzielone są na pakiety, które następnie przekazywane są za pośrednictwem urządzeń między nadawcą, a odbiorcą. Wykorzystywane narzędzia pozwalały na retransmisję pakietów, które nie zostały doręczone. Pakiety mogą być przesyłane z wykorzystaniem metody połączeniowej lub bezpołączeniowej. Metoda połączeniowa wymaga zestawienia wirtualnego połączenia przed rozpoczęciem transmisji. Z kolei metoda bezpołączeniowa oznacza brak konieczności zestawiania połączenia zanim rozpoczęta zostanie transmisja. Dzięki temu zwiększona została wydajność w porównaniu z siecią z komutacją obwodów oraz podniesiony został poziom dostępności usługi – awaria jednej ścieżki nie oddziaływała negatywnie na transmisję, ponieważ możliwe było jej przeprowadzenie przy wykorzystaniu innej ścieżki.

W miarę rozwoju sieci komputerowych proponowano różne techniki transmisji. Obecnie najbardziej znaną techniką jest standard Ethernet opisujący specyfikację przewodów, przesyłanych nimi sygnałów, format ramek oraz protokoły służące do komunikacji. Dla sieci wielousługowych zdefiniowano kompletną technikę ATM, która działa na zasadzie zestawiania wirtualnych kanałów i ścieżek (połączeniowo) [1]. Technika ATM wypierana jest przez MPLS, który daje możliwość definiowania etykiet na podstawie, których podejmowane są decyzje dotyczące obsługi poszczególnych pakietów [2]. MPLS jest z kolei przykładem techniki bezpołączeniowej. Współcześnie funkcjonują również koncepcje, takie jak Software-Defined Networking (SDN), w której założono separację warstwy transportowej od sterowania uzyskując w ten sposób większą elastyczność sieci [3, 4]. Administrator nie musi posiadać specjalistycznej wiedzy, aby móc konfigurować urządzenia – wystarczy umiejętność konfiguracji reguł, które następnie przekazywane są przez kontroler do urządzeń.

Praktycznie od samego początku istnienia sieci komputerowych zaczęto dostrzegać jak bardzo istotna jest niezawodność transmisji. Aby transmisja była zgodna z wymaganiami klienta zdefiniowano pojęcie QoS (z ang. Quality of Service) opisujące charakterystykę usług transmisyjnych będących podstawą do zaspokajania potrzeb wyrażonych przez usługobiorcę [5, 6, 7]. Od wielu lat prowadzone są prace mające na celu zdefiniowanie mechanizmów oraz architektur, dzięki którym możliwe byłoby różnicowanie jakości realizowanych usług transmisji danych oraz dodatkowo definiowanie i realizacja transmisji zgodnie z określonymi parametrami jakościowymi QoS [8, 9, 10, 11, 12, 13].

Obecnie dostawcy usługi transmisji danych pobierają zazwyczaj opłatę opierając się na maksymalnej możliwej do uzyskania przepływności łącza. Niestety, w rzeczywistości przepływność ta rzadko kiedy jest dostępna głównie ze względu na chwilowe nadmierne obciążenie poszczególnych węzłów i łączy w sieci. Chwilowa przepływność powinna być określana na podstawie przepływności uzyskiwanej na najmniej wydajnym ogniwie w ramach całej ścieżki transmisji pomiędzy nadawcą, a odbiorcą [14]. Tego typu wiedza dostępna jest wyłącznie, gdy ścieżka transmisji traktowana jest jako pewna całość, a nie jako zbiór łączy, co do wykorzystania których decyzja podejmowana jest niezależnie na różnych węzłach [15]. Zdefiniowane zostały różne metody i architektury pozwalające na zarządzanie ruchem sieciowym korzystające z odmiennych podejść do typów usług, priorytetów i klasyfikacji przesyłanych informacji. Dla klasycznych sieci IP zaproponowano modele usług IntServ [16] i DiffServ [17, 18, 19]. Podstawowym założeniem w IntServ jest rezerwacja zasobów dla strumieni danych i implementacja tego rozwiązania na każdym urządzeniu w sieci. DiffServ z kolei jest mechanizmem służącym do klasyfikowania w

oparciu o skończoną liczbę klas oraz priorytetyzowania ruchu dzięki wykorzystaniu wielu kolejek. Jednakże problem jakości transmisji jest na tyle złożony, że należy szukać innych rozwiązań dzięki którym podniesiony zostanie poziom świadczonej usługi. Od wielu lat prowadzone są prace mające na celu zdefiniowanie mechanizmów oraz architektur, dzięki którym możliwe byłoby różnicowanie jakości realizowanych usług transmisji danych oraz dodatkowo definiowanie i realizacja transmisji zgodnie z określonymi parametrami jakościowymi QoS [8, 9, 10, 11, 12, 13]. Usługa świadczona na rzecz klienta realizowana jest z pewną jakością. Istotne jest to, aby zastanowić się czy jakość jest wystarczająca. Kwestię wystarczalności rozpatrywać należy na dwa sposoby – czy jakość usługi jest zgodna z wymaganiami klienta oraz czy nie jest lepsza niż klient potrzebuje. Jeśli jakość usługi jest na wyższym poziomie niż ten wymagany przez klienta wówczas mowa o marnowaniu zasobów – niewykorzystaną jakością można by alokować na rzecz innego użytkownika [20, 21].

W niniejszej pracy zaproponowano rozwiązanie, dzięki któremu jakość usługi transmisji danych jest gwarantowana. W rozwiązaniu tym do zarządzania siecią wykorzystano system wieloagentowy [22], a zasoby nie są marnowane w związku z nadmiarową rezerwacją. Wykorzystanie tego typu systemu pozwoliło uzyskać rozwiązanie zdecentralizowane tzn. takie, w którym nie występuje centralny kontroler. Rozważając opisane kwestie dotyczące zapewniania jakości transmisji danych zaproponowano rozwiązanie stanowiące system zdecentralizowany wykorzystujący środowisko wieloagentowe w ramach którego użytkownik płaci za wymaganą przez siebie jakość transmisji - jakość ta jest gwarantowana poprzez ciągłe monitorowanie parametrów sieci takich jak przepływność (bandwidth), opóźnienie transmisji (delay), fluktuacja opóźnienia (jitter), współczynnik utraty pakietów (packet loss ratio) i w razie konieczności rekonfigurację tzn. zmianę aktualnie wykorzystywanych tras tak, aby dostosować parametry transmisji do wymaganej jakości. W prezentowanej pracy rozwiązanie ograniczono do najbardziej popularnych rozwiązań sieciowych tj. protokołu IP i standardu Ethernet oraz sieci o zasięgu lokalnym. W sieciach Ethernet/IP stosowane są protokoły routingu, w ramach których ścieżka transmisji uzależniona jest wyłącznie od celu transmisji [23]. Różnicowanie ścieżek transmisji w połączeniu z monitorowaniem jakości każdej z nich pozwala zagwarantować pewne parametry użytkownikowi. W prezentowanym rozwiązaniu ścieżka nie zależy wyłącznie od celu, ale i od źródła transmisja, a nawet konkretnej usługi. Prezentowane rozwiązanie pozwala na dynamiczny zakup usługi transmisji danych z określonymi parametrami jakościowymi. Poprzez zastosowanie taryfikacji opartej o metody rynkowe możliwe jest dynamiczne kształtowanie cen parametrów dotyczących jakości transmisji. Nowatorskim jest wykorzystanie podejścia agentowego w sieci teleinformatycznej do dynamicznego definiowania reguł trasowania

i różnicowania ścieżek transmisji pakietów w oparciu o wymagania użytkownika. Kolejnym nowatorskim rozwiązaniem jest wykorzystanie uczenia maszynowego (z ang. Machine Learning, ML) do translacji parametrów opisujących jakość transmisji w pewnej skali. Dzięki takiemu podejściu uzyskano bardziej wiarygodny opis jakości usługi transmisji danych.

Tematyka zaprezentowana w niniejszej pracy jest istotna z punktu widzenia racjonalnego gospodarowania zasobami w sieciach komputerowych oraz z punktu widzenia świadczenia usług zgodnie z wymaganiami klienta. Dzięki zastosowaniu proponowanego rozwiązania w lokalnej sieci Ethernet/IP, klient zawsze otrzymuje usługę na poziomie jaki jest przez niego oczekiwany i zgodny z wniesioną opłatą. Dzięki temu nie występuje problem nadmiarowej rezerwacji zasobów lub świadczenia usługi na nieakceptowalnie niskim poziomie wynikającym z niedostatecznej ilości zasobów. Przy dalszym wzroście liczby użytkowników sieci (urządzeń) racjonalne gospodarowanie zasobami staje się znaczącą kwestią.

Biorąc pod uwagę powyżej nakreślone zagadnienia zdefiniowano następująca tezę:

*Możliwe jest różnicowanie jakości usługi, jaką jest transmisja danych, poprzez zastosowanie systemu wieloagentowego do sterowania siecią tak, aby usługa realizowana była na poziomie oczekiwanym przez użytkownika.*

Praca zorganizowana została w następujący sposób:

- W rozdziale 1 zaprezentowano techniki związane z zapewnianiem jakości transmisji wykorzystywane we współczesnych sieciach komputerowych. Omówiono zostały techniki tzw. QoS opisujące różne podejścia do jakości, przedstawiono umowę dotyczącą jakości usług informatycznych oraz różne techniki routingu, które można wykorzystać, jako element systemu zapewniania jakości. W rozdziale tym zawarto również opis protokołu RSVP dzięki któremu możliwe jest rezerwowanie zasobów. Opiszano także technikę stanowiącą wzór dla systemów zapewniania jakości. Kolejnym ważnym elementem opisanym w rozdziale 1 są standardy sieciowe, w ramach których zdefiniowane zostały mechanizmy zapewniania jakości.
- Rozdział 2 zawiera charakterystykę systemów agentowych, w ramach której opisano koncepcję systemu agentowego, metody rynkowe alokacji dóbr ze szczególnym naciskiem na teorię gier, negocjacje oraz aukcje stanowiące ważny element niniejszej pracy. Przedstawiono zastosowania agentów w rozwiązaniach związanych z sieciami komputerowymi.

- W rozdziale 3 zaprezentowano koncepcję zapewniania jakości transmisji w sieciach komputerowych z wykorzystaniem agentów. Rozdział ten podzielono na kilka podrozdziałów. W podrozdziale 3.1 zaprezentowano koncepcję Pay&Require (P&R) jako rozwiązanie umożliwiające zapewnienie usługi transmisji danych na oczekiwanym przez klienta poziomie. Cel ten osiągnięto poprzez różnicowanie ścieżek transmisji. Dobór ścieżek podyktowany jest zmierzonymi wartościami parametrów opisujących jakość transmisji, takimi jak przepływność, opóźnienie transmisji, fluktuacja opóźnień i współczynnik utraty pakietów. W tradycyjnych sieciach komputerowych pakiety trasowane są na podstawie celu transmisji. W prezentowanym rozwiązaniu decyzja dotycząca routingu podejmowana jest nie tylko na podstawie celu, ale i źródła transmisji. Dzięki temu dla różnych źródeł i tego samego celu transmisji możliwe jest wykorzystanie różnych ścieżek, którymi przesłane zostaną pakiety.

Aby zweryfikować sensowność koncepcji przeprowadzono serię badań z wykorzystaniem emulatora stworzonego na potrzeby niniejszej pracy oraz wiarygodnego symulatora. W podrozdziale 3.2 przedstawiono opis emulatora stworzonego na potrzeby prowadzenia badań. Emulator stanowi kompletne rozwiązanie - sieć komputerową w której wykorzystano Pay&Require.

W podrozdziale 3.3 przedstawiono ogólną koncepcję eksperymentu w którym zastosowano symulator NS3. Zaprezentowano również algorytm wykorzystany w eksperymentach.

Bardzo ważnym elementem prezentowanego w niniejszej pracy rozwiązania było wykorzystanie agentów. W podrozdziale 3.4 opisano typy implementowanych agentów.

W podrozdziale 3.5 opisano koncepcję zastosowania ML do translacji parametrów opisujących jakość transmisji w pewnej skali. Jakość transmisji danych opisywana jest przez wartości kilku parametrów tj. przepływności, opóźnień transmisji, fluktuacji opóźnień oraz współczynnika utraty pakietów. Zrozumienie kombinacji wartości poszczególnych parametrów jest trudne dla klienta oraz w przypadku wykorzystania ich w systemie. W związku z tym podjęto się translacji parametrów do pewnej skali. Zdecydowano się na wykorzystanie uczenia maszynowego.

- W rozdziale 4 przedstawiono wyniki przeprowadzonych eksperymentów. Rozdział ten został podzielony na kilka podrozdziałów. Wyniki emulacji przedstawione zostały w podrozdziale 4.1. Emulacja przeprowadzona została dla

dwóch różnych sieci - składającej się z 4 ruterów oraz 9 ruterów. W obu przypadkach zweryfikowano ścieżki wybrane w zależności od klienta i jego oczekiwań odnośnie jakości transmisji. Dla mniejszej sieci tj. z 4 ruterami sprawdzono wybrane ścieżki dla:

- sieci działającej z maksymalnymi parametrami,
- sieci w której pogorszeniu uległy parametry jakościowe poszczególnych łączy. Pogorszenie parametrów uzyskano poprzez emulowanie dużego natężenia ruchu na danym łączy.

W ramach przeprowadzonej emulacji, dla sieci składającej się z 4 ruterów i różnych parametrów łączy zweryfikowano średnie opóźnienie transmisji oraz średni czas transmisji pliku o określonej wielkości. Zauważono, że wartości uzyskane dla różnych klientów uzależnione były od wybranej ścieżki i jej parametrów. Różnica w wynikach uzyskanych dla poszczególnych klientów wynikała z oczekiwań co do jakości transmisji. Ponieważ wyniki uzyskane dla sieci składającej się z 4 ruterów wykazały, że różnicowanie ścieżek transmisji ma wpływ na jakość transmisji przeprowadzono kolejne badania - dla większej sieci, tj. składającej się z 9 ruterów. Również w przypadku tych badań w pierwszej kolejności zweryfikowano wybór ścieżek w zależności od oczekiwań klienta i parametrów ścieżek.

Kolejne badania przeprowadzono z wykorzystaniem symulatora NS3. Miały na celu potwierdzenie poprawności uzyskanych wyników poprzez wykorzystanie dobrze znanego narzędzia. W podrozdziale 4.2 zaprezentowano uzyskane wyniki. Również w przypadku symulacji przeprowadzono dwa eksperymenty - pierwszy dla sieci składającej się z 4 ruterów, tej samej co w przypadku emulacji. Dla uzyskania wiarygodnej analizy wyniki uzyskane dla sieci składającej się z 4 ruterów zestawiono z wynikami emulacji. Wykazano tą samą zależność zarówno w przypadku emulacji, jak i symulacji - uzyskane wyniki uzależnione są od ścieżek, którymi przesyłane są dane, czyli oczekiwań klienta względem jakości transmisji. Kolejny eksperyment przeprowadzony został z wykorzystaniem sieci składającej się z 12 ruterów. Zweryfikowano wybór ścieżek w zależności od parametrów łączy. Zaprezentowano wyniki dotyczące średniego opóźnienia oraz średniego czasu transmisji dla sieci, w której nie zastosowano Pay&Require oraz takiej, w której koncepcja była wykorzystywana. Ponownie stwierdzono różnice w zmierzonych czasach zgodne z oczekiwaniami.

W podrozdziale 4.3 zaprezentowano wyniki badań dotyczących aukcji i rynkowego kształtowania cen usługi, jaką jest transmisja danych z określonymi parametrami opisującymi jakość transmisji.

W podrozdziale 4.4 przedstawiono wyniki badań dotyczących zastosowania ML do translacji parametrów opisujących jakość transmisji w pewnej skali.

- Rozdział ostatni stanowi podsumowanie całej pracy. W pierwszej kolejności w rozdziale tym zaprezentowano wnioski z przeprowadzonych badań oraz ustosunkowano się do postawionej tezy. Następnie rozdział podzielono na dwie sekcje. W pierwszej opisano oryginalne elementy pracy, a w kolejnej zebrano i opisano kierunki dalszych badań.



## Rozdział 1

# Metody zapewniania jakości

W rozdziale tym opisano metody zapewniania jakości transmisji w sieciach komputerowych. Rozwiązań związanych z zapewnianiem jakości transmisji jest wiele, opisane zostały najważniejsze z nich, a w szczególności te posiadające związek z rozwiązaniem prezentowanym w niniejszej pracy. Tematyka pracy jest ściśle związana z sieciami komputerowymi i jakością usługi transmisji danych. Poniższy przegląd metod ma na celu ulokowanie rozwiązania prezentowanego w niniejszej pracy wśród innych metod związanych z jakością. Zauważyć można, że techniki zapewniania jakości zastosowane mogą zostać na wielu płaszczyznach funkcjonowania sieci komputerowych.

### 1.1 Technologie sieciowe

W miarę rozwoju sieci komputerowych zaczęto zauważać problem związany z jakością świadczonej usługi transmisji danych. Sieci komputerowe funkcjonowały bez jakichkolwiek gwarancji jakości usług - klasycznym podejściem jakie można zaobserwować w sieci Internet jest best-effort [24]. W podejściu tym nie ma jakiegokolwiek gwarancji, że przesyłane dane trafią do odbiorcy. Nie ma także żadnej weryfikacji parametrów z jakimi dane są przesyłane. Każdy użytkownik otrzymuje usługę na takim samym poziomie wynikającym ściśle z warunków aktualnie panujących w sieci, a w szczególności z natężenia ruchu w sieci. W związku z dynamicznym rozwojem sieci komputerowych na przestrzeni lat różne podmioty podjęły się trudu zdefiniowania pojęcia jakim jest zapewnianie jakości usług (QoS). W pierwszej kolejności ważnym było zdefiniowanie czym jest usługa. W [6] usługę zdefiniowano jako możliwość wymiany informacji za pośrednictwem pewnego medium transmisyjnego. Możliwość ta jest dostarczana klientowi przez operatora.

W sieciach z komutacją komórek takich, jak ATM [1] QoS pojawia się w podstawowej definicji protokołu. Rezerwacja zasobów, a co za tym idzie zapewnianie jakości transmisji dokonywane jest na każdym etapie transmisji. Wynika to ze specyfiki działania ATM - na potrzeby transmisji zestawiane jest połączenie między

stacjami końcowymi. Połączenie w tym wypadku oznacza zagwarantowanie dostępności zasobów w czasie przesyłania danych.

Dostawca usługi może zawrzeć z klientem umowę dotyczącą poziomu świadczonej usługi (z ang. Service Level Agreement, SLA). Umowa SLA może zostać zawarta pomiędzy klientem, a dostawcą (dostawcami) usługi. Jej celem jest zapewnienie świadczenia usługi na określonym poziomie. SLA nie definiuje konkretnych mechanizmów monitorowania i utrzymywania jakości. Umowa SLA zawiera informacje takie jak:

- Typ usługi objętej umową – czego dana umowa dotyczy, przy czym może to być usługa ogólna, jak np. dostęp do sieci internet lub usługa szczegółowa, jak np. telewizja przez internet, telefonia internetowa,
- Jakość lub parametry z nią związane – określenie na jakim poziomie ma być dana usługa świadczona i przy użyciu jakich parametrów ten poziom jest wyznaczany,
- Określenie sposobów monitorowania jakości – w jaki sposób odczytywane są parametry transmisji, z jaką częstotliwością oraz w jaki sposób można uzyskać dostęp do wyniku pomiarów,
- Metody informowania o występujących błędach – w jaki sposób użytkownik może zgłosić zauważone nieprawidłowości,
- Czas reakcji – w jakim czasie od zgłoszenia dostawca musi interweniować oraz ile czasu na naprawę ma dostawca,
- Kary umowne – wszelkiego rodzaju określenie kar w przypadku niedotrzymania warunków umowy.

Bardzo często w umowach zawieranych z dostawcami usług internetowych nie ma wzmianki o gwarantowanych parametrach, najczęściej jest tylko informacja o maksymalnej możliwej do uzyskania przepływności. Rzadko w umowach pojawia się informacja o minimalnych parametrach opisujących jakość transmisji. W koncepcji zaprezentowanej w niniejszej pracy założono, że użytkownik płaci za określoną jakość transmisji danych, która jest mu gwarantowana na podstawie umowy zawartej z dostawcą usług internetowych.

W sieciach z komutacją pakietów wykorzystujących metodę bezpołączeniową wydajne mechanizmy QoS są trudniejsze w zastosowaniu. Dane przesyłane przez

tego typu sieci przekazywane są za pośrednictwem wielu urządzeń sieciowych, często z wykorzystaniem infrastruktury różnych dostawców łączących różne sieci prywatne. Co za tym idzie zachowanie sieci jest trudne do określenia w momencie rozpoczęcia transmisji. Międzynarodowy Związek Telekomunikacyjny (z ang. International Telecommunications Union, ITU) definiuje usługę w sieci IP jako świadczoną przez warstwę usługową użytkownikowi końcowemu (np. hostowi lub elementowi sieci), która wykorzystuje możliwości transmisji IP oraz powiązane z nią funkcje sterowania i zarządzania, w celu dostarczania informacji użytkownika określonych przez pewną umowę opisującą poziom danej (świadczonej) usługi [25]. Z kolei jakość definiowana jest jako pojęcie pozwalające na ocenę czy dana usługa zaspokaja oczekiwania klienta [6]. Trzy najbardziej znane podmioty międzynarodowe tj. ITU, Grupa Zadaniowa ds. Inżynierii Internetowej (z ang. Internet Engineering Task Force, IETF) i Europejski Instytut Norm Telekomunikacyjnych (European Telecommunications Standards Institute, ETSI) na przestrzeni czasu w różny sposób podchodziły do jakości usług. Najbliższe sobie podejścia prezentowały ITU i ETSI [6]. Jakość usługi została zdefiniowana jako zbiorowy efekt wykonania usługi, który determinuje stopień zadowolenia użytkownika. W prezentowanej definicji istotna jest determinacja jakości usługi przy określeniu stopniu zadowolenia użytkownika. Oznacza to, że określenie jakości usługi jest ściśle związane z empirycznym odbiorem przez użytkownika. ITU oraz ETSI wyszczególniają następujące sformułowania:

- Jakość usługi oczekiwana przez klienta – określenie oczekiwań klienta względem świadczonej usługi tj. wszelki opis (zarówno techniczny, jak i nietechniczny) zrozumiały dla klienta i dostawcy.
- Jakość usługi oferowana przez dostawcę – parametry możliwe do dostarczenia przez operatora modyfikowane w oparciu o wewnętrzną politykę dostawcy (np. koszt, zasoby).
- Jakość usługi osiągnięta przez dostawcę – parametry opisujące jakość powinny być monitorowane i porównywane z tymi zaoferowanymi przez dostawcę.
- Jakość usługi postrzegana przez klienta – ocena na podstawie empirycznych doznań użytkownika.

Podejście prezentowane przez IETF [26, 27] dotyczy aspektów technicznych funkcjonowania sieci. Odczucia użytkownika wydają się nie być istotne. Jakość usługi definiowana jest jako zestaw wymagań względem usług, które musi spełnić sieć

podczas przesyłania strumienia informacji [26]. IETF zaproponowało dwie architektury ściśle związane z jakością świadczonych usług w sieciach z komutacją pakietów tj. IntServ oraz DiffServ.

IntServ (z ang. Integrated Services) jest architekturą specyfikującą elementy niezbędne do zagwarantowania jakości usług w sieciach komputerowych [28]. Głównym założeniem tej architektury jest rezerwacja zasobów wykorzystywanych do transmisji w oparciu o żądania. IETF podjęło się zadania mającego na celu standaryzację protokołu, którego zadaniem była sygnalizacja (sterowanie połączeniem) ściśle związana z zapewnianiem jakości usługi. Na potrzeby przesyłania żądań opracowano RSVP (z ang. Resource Reservation Protocol) [29]. Opis RSVP znajduje się w dalszej części niniejszego rozdziału. Największym problemem w przypadku IntServ okazała się skalowalność. Dla dużych sieci routery działające w rdzeniu sieci musiały obsługiwać ogromną ilość rezerwacji. Zauważono problem w przypadku dalszego rozwoju sieci.

DiffServ (z ang. Differentiated Services) z kolei jest architekturą służącą do zapewniania parametrów transmisji, takich jak przepływność, opóźnienie transmisji, fluktuacja opóźnienia, współczynnik utraty pakietów w sieciach komputerowych [30, 17]. Pozwala klasyfikować ruch sieciowy i różnicować go. Znakowanie pakietów dokonywane jest przez źródło transmisji lub urządzenie brzegowe. Znakowanie jest rozwiązaniem działającym w ramach pewnej domeny, czyli sieci, w której wykorzystywany jest DiffServ. W danej sieci urządzenia są skonfigurowane tak, aby odpowiednio obsługiwany był ruch należący do danej klasy zgodnie z określonymi wymaganiami. Rozwiązanie to jest rozwiązaniem lokalnym (obszar należący do jednej domeny) wykorzystującym wiele kolejek dla pakietów, które oczekują na przekazanie. Zazwyczaj możliwe jest konfigurowanie liczby kolejek, zasobów udostępnianych na rzecz danej kolejki oraz priorytetów kolejek. DiffServ dla pakietów należących do tej samej klasy nie pozwala na definiowanie różnych zachowań. Oznacza to, że niezależnie od nadawcy pakiety należące do tej samej klasy będą zawsze obsługiwane w taki sam sposób.

Kolejną techniką związaną z zapewnianiem jakości transmisji jest MPLS [2]. Jest to technika stosowana w sieciach komputerowych w której wykorzystano etykiety służące do znakowania strumieni danych. Ruter brzegowy nadaje etykietę, dzięki której pakiety należące do jednego strumienia są w taki sam sposób obsługiwane przez urządzenia w sieci. Urządzenia weryfikują etykietę, a następnie podejmują określone działania. Dzięki takiemu podejściu mimo podobieństwa strumienia, których źródłem są różni użytkownicy mogą być obsługiwane w różny sposób. Zastosowanie etykiet uprościło proces decyzyjny - nie ma konieczności weryfikowania docelowego adres i tablicy routingu, wystarczy weryfikacja krótkiej etykiety i na jej

podstawie określone działanie. W porównaniu z sieciami Ethernet/IP, MPLS jest rozwiązaniem droższym we wdrożeniu co stanowi jego główną wadę.

Protokołem związanym z zapewnianiem jakości w sieciach komputerowych jest RSVP. Jest to protokół wykorzystywany przez użytkowników w celu uzyskania określonych parametrów transmisji (jakości) [29]. Protokół ten jest również wykorzystywany przez routery w celu poinformowania innych urządzeń należących do tej samej ścieżki transmisji o konieczności zapewnienia określonych parametrów [31]. W ramach RSVP możliwa jest rezerwacja parametrów w jednym kierunku transmisji. Za inicjalizację procesu rezerwacji odpowiada odbiorca – strona stanowiąca cel transmisji informuje urządzenia w sieci o akceptowanych parametrach transmisji. RSVP nie jest protokołem routingu – może zostać wykorzystany wraz z dowolnym protokołem routingu, lecz sam nie posiada takiej funkcjonalności. Umożliwia dynamiczną automatyczną adaptację w przypadku wystąpienia zmian w sieci.

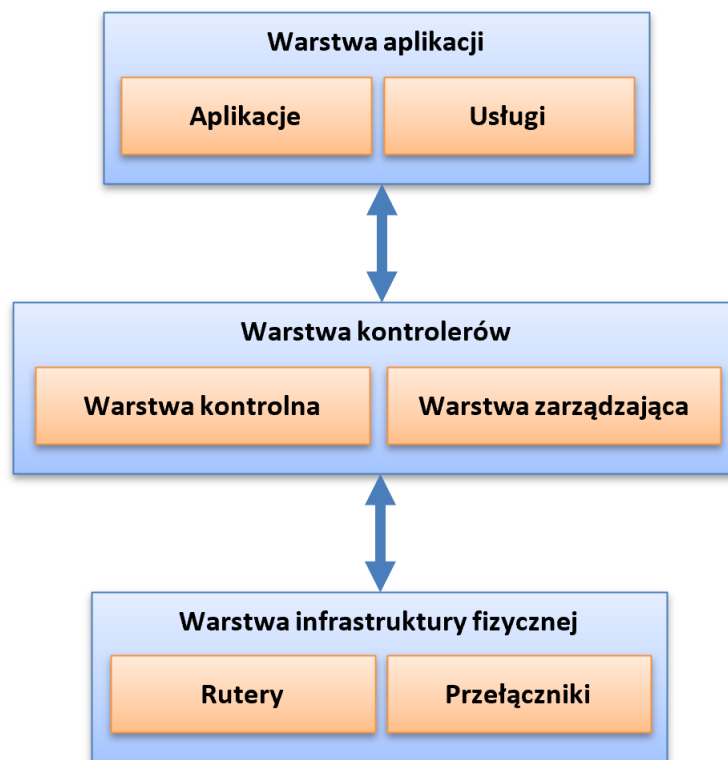
Obecnie RSVP często rozszerzane jest przy użyciu Traffic Engineering (TE) [32, 33]. TE to zagadnienie dotyczące pomiarów, oceny i optymalizacji wydajności w sieciach IP [34]. Istotną kwestią jest optymalizacja zarówno na poziomie ruchu sieciowego, jak i na poziomie zasobów. Do kluczowych parametrów, jak chodzi o pomiary zorientowane na ruch sieciowy zaliczyć można opóźnienie transmisji, fluktuację opóźnienia, współczynnik utraty pakietów oraz przepływność [6]. Ważne jest, aby świadczona usługa była niezawodna – należy zapewnić mechanizmy poprawiające integralność sieci oraz minimalizować błędy, uszkodzenia i awarie. W sieciach komputerowych bardzo istotną kwestią jest transmisja pomiędzy węzłem źródłowym, a docelowym, która nie byłaby możliwa bez routingu. W związku z tym jednym z kluczowych zadań TE jest nadzór i optymalizacja procesu routingu tak, aby przebiegał on w optymalny sposób. W TE najważniejszy jest użytkownik i jego odczucia – jakość usługi musi być na odpowiednim, akceptowanym przez użytkownika poziomie. Co za tym idzie bardzo istotne jest, aby prawidłowo zidentyfikować parametry, które należy zoptymalizować – tak, aby usługa była ciągle na odpowiednim poziomie. Optymalizacja polega na właściwym planowaniu pojemności sieci, nadzorze nad routingiem, zarządzaniu zasobami, przy czym zasoby to przepływność łączy, rozmiar bufora, moc obliczeniowa. TE odnosi się również do kształtowania ruchu w sieci, zarządzania kolejkami, planowaniu i wszystkich innych funkcjach służących do regulacji ruchu. Ze względu na to, iż sytuacja w sieciach komputerowych dynamicznie się zmienia (zapotrzebowanie, nowe technologie) proces optymalizacji nie powinien być traktowany jako jednorazowy, lecz jako proces ciągły. Ciężko jest zdefiniować uniwersalne parametry, które można zastosować w dowolnej sieci – optymalizacja jest uzależniona od wielu czynników specyficznych dla konkretnych sieci (konfiguracji, wymagań). System zarządzający zgodny z TE powinien

działać szybko, niezawodnie i efektywnie w przypadku zmian w sieci. Przy tego typu zmianach sieć powinna nadal działać stabilnie i bez zakłóceń. Ocena wydajności może odbywać się na kilka sposobów – metody analityczne, symulacje oraz na podstawie pomiarów rzeczywistych parametrów. Przy czym metody analityczne i symulacje mogą zostać wykorzystane głównie w przypadku wstępnego modelowania oraz określania charakterystyki ruchu (np. symulacja obciążenia sieci lub wystąpienia jakiegoś zjawiska). Ocena na podstawie rzeczywistych parametrów może być skomplikowana, lecz dostarcza bardzo cennych informacji z punktu widzenia zarządzania siecią. Bardzo istotne jest, aby rozwiązanie TE było dopasowane do konkretnego systemu i wymagań przy jednoczesnym zachowaniu elastyczności i dobrej skalowalności – tak, aby w prosty sposób można było je rozszerzyć o nowe wymagania. RSVP-TE jest rozszerzeniem protokołu RSVP stworzonym na potrzeby kształtowania ruchu sieciowego [33, 32]. Pozwala na rezerwację zasobów (przepływność, fluktuacja opóźnienia, itp.) w sieciach IP.

W sieciach komputerowych nie tylko poszczególne protokoły mogą mieć wpływ na jakość świadczonych usług, ale także całe architektury. Przykładem jest SDN (z ang. Software-Defined Networking). SDN to technologia pozwalająca na zarządzanie, konfigurację oraz kontrolę sieci w sposób dynamiczny z wykorzystaniem otwartych standardów [3, 4]. W ramach SDN istotną kwestią jest strona programistyczna działania sieci i w związku z tym w tej architekturze dokonano separacji warstwy odpowiedzialnej za przekazywanie danych od warstwy odpowiedzialnej za kontrolę i sterowanie (rysunek 1.1). Kontroler zarządza urządzeniem lub urządzeniami znajdującymi się w warstwie fizycznej (kontrolowanej) za pośrednictwem interfejsu. Interfejs zazwyczaj działa na zasadzie przekazywania żądań przy użyciu bibliotek lub wywołań systemowych (interfejs lokalny) lub przy użyciu protokołów (otwarte standardy) w przypadku zarządzania zdalnego. Z założenia SDN musi być:

- Programowalne - bez konieczności dbania o wszelkie narzędzia służące do przekazywania informacji,
- Elastyczne – wykorzystanie otwartych standardów,
- Łatwe w zarządzaniu – administrator nie musi posiadać specjalistycznej wiedzy na temat konfiguracji konkretnych urządzeń sieciowych, wystarczy mu ogólna wiedza i umiejętność pracy z aplikacjami SDN,
- Zarządzane centralnie – możliwym jest kontrolowanie całej sieci z jednego punktu. Punkt ten posiada wiedzę na temat całej sieci.

W modelu SDN poszczególne warstwy mają następujące zadania:



RYSUNEK 1.1: SDN - separacja warstw

- Warstwa aplikacji – w tej warstwie działają aplikacje i usługi, które bezpośrednio przekazują wymagania oraz reguły dotyczące działania sieci do warstwy kontrolerów SDN. Warstwa ta może posiadać wiedzę na temat topologii sieci na potrzeby definiowania reguł.
- Warstwa kontrolerów – jest scentralizowanym systemem, który tłumaczy wymagania pochodzące z warstwy aplikacji do postaci zrozumiałej dla urządzeń sieciowych (np. komendy konfiguracyjne). Warstwa ta może udostępniać informacje niezbędne do funkcjonowania warstwy aplikacji – wszelkie parametry dotyczące sieci, informację o topologii (warstwa kontrolna) lub wyłącznie odpowiadać za konfigurację urządzeń (warstwa zarządzająca).
- Warstwa infrastruktury fizycznej – w warstwie tej działają wszelkiego rodzaju urządzenia sieciowe obsługujące SDN. Warstwa ta jest odpowiedzialna za odbieranie i przetwarzanie pakietów transmitowanych przez sieć. Działa zgodnie z konfiguracją otrzymaną z warstwy kontrolerów. Urządzenia fizyczne mogą również przekazywać pewne parametry dotyczące swojej pracy do podwarstwy kontrolnej działającej w ramach warstwy kontrolerów.

SDN jest rozwiązaniem scentralizowanym tzn. takim, które posiada centralny kontroler [4]. Prowadzone są prace mające na celu decentralizację rozwiązania [35, 36].

Jednym z rozwiązań, które mogą być użyteczne przy decentralizacji są systemy agentowe [37].

## 1.2 QoX

Każdy ze wspomnianych w podrozdziale 1.1 podmiotów (ITU, ETSI oraz IETF) na przestrzeni lat podjął się zdefiniowania kilku pojęć dotyczących jakości w sieciach [6, 7]. Są to między innymi QoS, CoS (z ang. Class of Service) oraz QoE (z ang. Quality of Experience). Każde ze wspomnianych wykorzystywane jest w kontekście zapewniania jakości. Założenia poszczególnych określeń różnią się od siebie.

QoS jest to ogół cech usługi telekomunikacyjnej, które mają wpływ na zdolność zrozumienia i zaspokojenia potrzeb użytkownika [38]. W sieciach komputerowych pojęcie to skojarzone jest ze świadczeniem usługi na poziomie oczekiwanym przez użytkownika oraz możliwym do zapewnienia przez dostawcę. Głównie założenie QoS to klasyfikacja i priorytetyzacja ruchu sieciowego w zależności np. od konkretnego użytkownika czy usługi [9, 39, 40]. QoS pozwala na zagwarantowanie parametrów, takich jak np. szybkość transmisji, stały poziom opóźnienia transmisji, brak fluktuacji opóźnienia. W ramach QoS wykorzystywane są inne techniki związane z jakością usługi – CoS, QoR [41]. Na jakość usługi wpływ ma wiele czynników, takich jak stabilność usługi, skalowalność, dostępność, konserwacja czy opóźnienia. Aby usługa świadczona była na określonym, zagwarantowanym poziomie, QoS powinien być zapewniony na każdym etapie transmisji źródło – cel. Oznacza to, że QoS nie powinno ograniczać się wyłącznie do sieci lokalnej – gwarancja powinna również dotyczyć sieci rozległej. Czynnikiemami wpływającymi negatywnie na jakość transmisji są:

- Niska przepływność – tworzenie się tzw. wąskich gardeł, czyli odcinków sieci, które powodują pogorszenie jakości w ramach całej trasy,
- Odrzucanie pakietów – routery w momencie, gdy są nadmiernie obciążone odrzucają pakiety. Nadawca może zdecydować się na retransmisję pakietów, co dodatkowo obciąży łącze,
- Błędy transmisji – w momencie, gdy dane podczas transmisji zostaną uszkodzone,
- Fluktuacja opóźnienia – stanowi duży problem, ponieważ pakiety docierają do celu z różnym opóźnieniem, a co za tym idzie mogą zostać błędnie złożone przez cel transmisji,



- Opóźnienie – może być wynikiem nadmiernie wypełnionych buforów w routach i koniecznością dłuższego niż normalnie oczekiwania na przetworzenie.

Liczba usług świadczonych w sieciach wciąż rośnie natomiast można wyróżnić pewne usługi, które kojarzone są z zapewnianiem jakości: telewizja internetowa (IPTV), telefonia internetowa (VoIP), wideo na życzenie (VoD), wideokonferencje, gry on-line, aplikacje krytyczne. Są to usługi w ramach, których jakość powinna być nieustannie na takim samym poziomie. Istotne jest również to, aby klasyfikacja odbywała się w czasie rzeczywistym bez dodatkowych narzutów na czas transmisji [39].

Z kolei CoS jest parametrem, który służy do klasyfikowania przesyłanych danych [42]. Wykorzystywany jest on w protokołach komunikacyjnych między innymi do priorytetyzacji transmisji danych. Informacja o klasie ruchu umieszczana jest w polu PCP oznaczonej ramki 802.1Q. PCP to 3 bitowe pole wykorzystywane do oznaczania priorytetu ramki. W zaleceniu dotyczącym CoS (pierwotnie 802.1p) zdefiniowano 8 klas (priorytetów). Klasy te są wykorzystywane również przez inne techniki zapewniania jakości. ITU jakiś czas temu zajęło się również zdefiniowaniem jakości usługi jako elementu postrzeganego przez użytkownika, a nie opisywanego wyłącznie przez parametry ściśle związane z funkcjonowaniem sieci komputerowej. To właśnie z konieczności opisu jakości usługi poprzez empiryczne doznania użytkownika zdefiniowane zostało pojęcie QoE - ogólna akceptowana jakość usługi, postrzegana subiektywnie przez użytkownika [43]. Na postrzeganie przez użytkownika końcowego wpływ mają wszystkie czynniki występujące pomiędzy źródłem danych, a celem. Czynniki wpływające na jakość to np. stosowane media transmisyjne i parametry z nimi skojarzone, urządzenia sieciowe wykorzystywane do transmisji, narzut protokołów sieciowych. Znaczący wpływ na ocenę mają użytkownicy – ich cechy psychiczne, fizyczne, doświadczenie, otoczenie oraz opinie innych osób [44]. QoE znajduje zastosowanie w przypadku transmisji obrazu [45], głosu lub danych – czyli takich usług, które można w prosty sposób ocenić na podstawie zmian jakości w trakcie trwania transmisji. W rekomendacji [46] zdefiniowana została pięciopunktowa bezwzględna skala MOS (z ang. Mean Opinion Score, tabela 1.1), zalecenia dotyczące stosowania różnych skali ocen do porównywania (tabela 1.2) oraz zalecenia dotyczące miejsca, w którym przeprowadzany jest pomiar, np. w przypadku wideo dystans od ekranu, rozdzielczość ekranu, jasność, kontrast oraz inne elementy mogące mieć wpływ na wynik pomiaru.

W MOS można wyszczególnić następujące sposoby oceny jakości:

- Pełne odniesienie – gdy porównywany jest sygnał pierwotny (wzór) i zniekształcony,

| MOS | Jakość      |
|-----|-------------|
| 1   | Zła         |
| 2   | Słaba       |
| 3   | Dostateczna |
| 4   | Dobra       |
| 5   | Doskonała   |

TABELA 1.1: Skala MOS

| Ocena | Jakość        |
|-------|---------------|
| 3     | Dużo lepsza   |
| 2     | Lepsza        |
| 1     | Trochę lepsza |
| 0     | Taka sama     |
| -1    | Trochę gorsza |
| -2    | Gorsza        |
| -3    | Dużo gorsza   |

TABELA 1.2: Skala porównawcza

- Brak odniesienia – gdy nie ma możliwości porównania z sygnałem wzorcowym, ocena wyłącznie na podstawie sygnału zniekształconego,
- Częściowe odniesienie – gdy dostępna jest tylko częściowa informacja dotycząca sygnału wzorcowego.

Ponieważ technika ta bazuje na ocenach wystawianych przez użytkowników, a co za tym idzie dość skomplikowanym procesie uzyskiwania informacji dotyczących jakości, prowadzone są prace nad przełożeniem zmierzonych (w ramach innych technik) parametrów (np. QoS) na parametry QoE [47]. Ze względu na fakt, iż internet cały czas ewoluuje niewystarczające stają się podstawowe parametry, na bazie których stwierdzana jest jakość usługi. Obecnie coraz częściej parametry definiowane są w zależności od usługi, która ma podlegać ocenie [48, 49], a nie w oparciu o ogólnie przyjęte parametry dla techniki QoE.

### 1.3 Ruting

Z zapewnianiem jakości w sieciach komputerowych często związane są różne techniki rutingu [50, 13]. Ruting [51] jest techniką wykorzystywaną do kierowania pakietów danych w ramach transmisji pomiędzy źródłem, a celem. Wyróżnić można dwie grupy rutingu:

- Statyczny – trasy definiowane są ręcznie poprzez podanie adresu następnego skoku dla danej sieci docelowej. Konfiguracja jest wykonywana ręcznie

przez administratora na każdym urządzeniu działającym w sieci, przez które przesyłane będą pakiety. Rozwiązanie to daje największą władzę administratorowi, a zarazem jest bardzo uciążliwe i czasochłonne. Administrator musi pamiętać o skonfigurowaniu wszystkich ścieżek w obie strony na wszystkich urządzeniach do każdej sieci. Ruting statyczny umożliwia definiowanie ścieżek wyłącznie w oparciu o docelową sieć (adres), nie umożliwia definiowania reguł w oparciu o inne informacje niesione w nagłówku pakietu. W przypadku awarii administrator musi samodzielnie wykonać rekonfigurację.

- Dynamiczny – dynamiczne protokoły routingu ułatwiają wymianę informacji pomiędzy ruterami oraz upraszczają proces budowania tablic routingu. Wszelkie informacje dotyczące sieci pozyskiwane są dynamicznie, a następnie podejmowana jest decyzja o tym, które z nich należy umieścić w tablicy routingu. Decyzja o wyborze konkretnej trasy podejmowana jest przez protokół routingu w oparciu o pewne parametry zależne od protokołu. Administrator ma ograniczoną możliwość konfiguracji, a po wprowadzeniu niezbędnych ustawień sieć działa samodzielnie (bez ingerencji administratora). W przypadku awarii tablice routingu modyfikowane są automatycznie. Wadą protokołów routingu dynamicznego jest większe wykorzystanie zasobów (w stosunku do routingu statycznego) – wynika to z faktu, iż urządzenia wymieniają między sobą komunikaty dotyczące tras co powoduje wzrost użycia procesora oraz dodatkowe wykorzystanie pasma.

Protokoły routingu dynamicznego mogą zostać podzielone na kilka różnych sposobów. Ze względu na domenę routingu można wyróżnić protokoły IGP (z ang. Interior Gateway Protocol) – protokoły bramy wewnętrznej, używane wewnątrz konkretnej domeny oraz EGP (z ang. Exterior Gateway Protocol) – protokoły bramy zewnętrznej, wykorzystywane w przypadku routingu pomiędzy różnymi domenami. Kolejnego podziału dokonać można w oparciu o informacje zawarte w aktualizacjach przesyłanych pomiędzy urządzeniami. Wyróżnić można klasowe protokoły routingu – w aktualizacjach nie jest uwzględniana informacja o masce podsieci. Maski określana jest na podstawie konfiguracji interfejsu lub klasy adresu. Z kolei bezklasowe protokoły routingu w aktualizacjach dodają informację o masce, co umożliwia podział na podsieci o dowolnej wielkości. Protokoły można również podzielić ze względu na sposób wyznaczania tras. W tym wypadku wyróżnić można protokoły wektora odległości, łącze-stan, hybrydowe oraz wektora ścieżki.

W protokołach wektora odległości trasy rozgłaszane są jako wektory odległości i kierunku. Odległość definiuje się przy wykorzystaniu metryki – zazwyczaj liczba skoków, czyli urządzeń pomiędzy źródłem, a celem transmisji. Jako kierunek

rozumiany jest ruter następnego skoku, czyli urządzenie, do którego powinny zostać skierowane pakiety, aby zbliżyć się do celu transmisji. Zazwyczaj w tego typu protokołach wykorzystywany jest algorytm Bellmana-Forda, który pozwala znaleźć najkrótszą ścieżkę (o najmniejszej wadze) pomiędzy dwoma wierzchołkami w grafie. Rutery wymieniają pomiędzy sobą (tylko najbliżsi sąsiedzi) w stałych odstępach czasu kompletną informację o tablicach routingu. Ponadto nie znają pełnej topologii sieci – posiadają wyłącznie informację o następnym skoku, czyli kolejnym urządzeniu, do którego skierują pakiety. Przykładowymi protokołami są RIP (z ang. Routing Information Protocol) i IGRP (z ang. Interior Gateway Routing Protocol).

W protokołach łącze-stan informacje dotyczące topologii zbierane są ze wszystkich węzłów dzięki czemu każdy ruter zna pełną topologię sieci. Komunikaty przekazywane są okresowo i zawierają tylko częściową informację – wyłącznie o podsięciach bezpośrednio podłączonych do rutera z uwzględnieniem metryki. Bazy danych, w których przechowywane są informacje o topologii sieci – pełna wiedza o wszystkich ruterach w sieci, tworzone są na podstawie informacji wysyłanych pomiędzy ruterami, a po zebraniu wszystkich informacji tworzone jest drzewo SPF (z ang. Shortest Path First) i z wykorzystaniem algorytmu Dijkstry (do znajdowania najkrótszej ścieżki z pojedynczego źródła w grafie, wagi krawędzi nie mogą być ujemne) określana jest droga o najmniejszym koszcie. Baza przechowująca informacje o topologii jest aktualizowana wtedy, gdy pojawi się jakaś zmiana w sieci. Do tej grupy protokołów routingu dynamicznego zalicza się protokół OSPF (z ang. Open Shortest Path First) oraz IS-IS (z ang. Intermediate System to Intermediate System).

Hybrydowe protokoły routingu to takie, które posiadają zarówno cechy protokołów stanu łącza, jak i wektora odległości. Do tej grupy protokołów zaliczyć można bezklasowy protokół routingu EIGRP (z ang. Enhanced Interior Gateway Routing Protocol). Jest to protokół stworzony przez firmę Cisco (protokół własnościowy), jako następcę protokołu IGRP, w 2013 roku został przekształcony w protokół otwarty, a w 2016 roku ukazał się formalny dokument zawierający jego specyfikację [52]. EIGRP często jest zaliczany do grupy protokołów wektora odległości ze względu na metodę działania. W EIGRP wykorzystano algorytm DUAL (Diffusing Update Algorithm), dzięki któremu dana ścieżka wyznaczana jest globalnie bez możliwości powstania pętli routingu. Aktualizacje routingu są częściowe – zawierają wyłącznie informacje o tym, co uległo zmianie, dzięki temu zredukowano liczbę przesyłanych komunikatów.

W przypadku protokołów wektora ścieżki różnica w stosunku do protokołów wektora odległości i stanu łącza jest taka, że w tablicy routingu oprócz sieci docelowej i następnego skoku umieszczana jest również informacja, o całej ścieżce do celu. Przykładem protokołu z tej grupy jest BGP (z ang. Border Gateway Protocol).

BGP [53] jest protokołem, którego głównym zadaniem jest wymiana informacji o osiągalności poszczególnych sieci pomiędzy innymi urządzeniami, na których działa ten protokół wraz z listą systemów autonomicznych, do których informacja dotarła. Dzięki wymianie tego typu informacji możliwe jest zbudowanie grafu reprezentującego całą topologię sieci z uwzględnieniem różnych systemów autonomicznych. W ramach BGP dostępne są mechanizmy umożliwiające bezklasowy ruting pomiędzy domenami (CIDR, z ang. Classless Inter-Domain Routing) – wyeliminowano klasowość oraz umożliwiono agregację tras również w przypadku systemów autonomicznych. Decyzja o kierowaniu pakietu w odpowiednie miejsce podejmowana jest wyłącznie na podstawie adresu celu transmisji znajdującego się w nagłówku pakietu oraz reguł lub zestawu reguł dotyczących sieci skonfigurowanych przez administratora.

W klasycznym ujęciu trasowanie pakietów odbywa się na podstawie informacji niesionej przez pakiet jaką jest adres docelowy [51]. Jednakże stosowane są również techniki pozwalające na trasowanie pakietów nie tylko na podstawie celu transmisji, ale i jego źródła [54, 55].

Poza klasycznym routingiem spotkać można również technikę, dzięki której decyzje dotyczące routingu mogą być podejmowane na podstawie różnych informacji przenoszonych w nagłówku pakietu. Aby omówić technikę PBR (z ang. Policy-Based Routing) [56] należy najpierw omówić dwie inne techniki, z którymi PBR może zostać skojarzone tj. Source Routing (SR) [51] oraz Dynamic Source Routing (DSR) [51]. SR to technika pozwalająca nadawcy określić ścieżkę, po jakiej odbyć ma się transmisja z wyszczególnieniem wszystkich węzłów pośredniczących. W ramach protokołu IP w przypadku tej techniki wykorzystywane są dwa pola nagłówka:

- SSRR (z ang. Strict Source and Record Route) – określona zostaje dokładna ścieżka, którą pakiet ma zostać przesłany. Wyszczególniony zostaje adres każdego urządzenia, do którego pakiet ma trafić. Rutery przekazują pakiet zgodnie z podaną informacją do urządzenia następnego skoku. W przypadku błędu (np. urządzenie następnego skoku jest nieosiągalne) pakiet zostanie odrzucony.
- LSRR (z ang. Loose Source and Record Route) – zdefiniowane zostają węzły, przez które pakiet ma przejść, lecz nie zostaje zdefiniowana dokładna ścieżka. W pakiecie przechodzącym przez kolejne węzły zmieniany jest adres urządzenia docelowego na adres urządzenia następnego skoku.

W przypadku sieci, w której istnieje kilka ścieżek dotarcia do tego samego celu wykorzystując SSRR możliwym jest przekazanie pakietów różnymi ścieżkami w

celu przeprowadzenia oceny poszczególnych ścieżek i wyboru odpowiedniej dla konkretnego zastosowania. Technika SR stworzona została głównie z myślą o diagnozowaniu i monitorowaniu działania sieci komputerowej.

DSR to rozproszony protokół routingu stworzony na potrzeby rozproszonych sieci komputerowych, przy czym można go również wykorzystać w sieciach komórkowych. Ścieżka tworzona jest na żądanie wtedy, gdy zachodzi taka potrzeba. W przypadku tego protokołu cała informacja dotycząca routingu przechowywana i zarządzana jest przez węzły (np. nadawcę). Węzły we własnym zakresie pozyskują informacje o topologii sieci. W momencie, gdy cel transmisji odbierze pakiety i chce przesłać odpowiedź to sprawdzi swoją pamięć podręczną ścieżek, gdy znajdzie wpis to wykorzysta ścieżkę zgodną z nim, a gdy nie znajdzie odpowiedniego wpisu wykorzysta informację niesioną przez odebrany pakiet. Protokół ten stworzony został, aby zmniejszyć ilość informacji sterujących przesyłanych przez sieć, a co za tym idzie, aby lepiej zagospodarować pasmo.

PBR to technika, która umożliwia podejmowanie decyzji dotyczącej routingu (kierowania pakietu) na podstawie reguł zdefiniowanych przez administratora [10, 57]. Technika ta działa na zasadzie routingu statycznego – tablice routingu budowane są przez administratora i nie ulegają zmianie automatycznie tak, jak ma to miejsce w przypadku routingu dynamicznego. Administrator może samodzielnie zdecydować o tym, jak kierowane będą pakiety przesyłane przez sieć. W przypadku klasycznego routingu, gdy odebrany zostanie pakiet następuje proces przeszukiwania tablicy routingu – w momencie, gdy wpis dotyczący danej sieci docelowej zostanie znaleziony to pakiet zostanie przekierowany w oparciu o wyszczególniony adres następnego skoku (kolejne urządzenie lub sieć docelowa), z kolei w sytuacji, gdy wpis nie zostanie znaleziony pakiet zostanie odrzucony. Gdy w sieci wykorzystywana jest technika PBR to decyzja dotycząca kierowania pakietu nie jest podejmowana wyłącznie na podstawie adresu docelowego, lecz również na podstawie innych informacji niesionych przez pakiet (np. adres źródłowy IP, rozmiar pakietu, port źródłowy i docelowy). Głównym założeniem PBR jest wykorzystanie wielu tablic routingu równocześnie. W klasycznym routingu wykorzystywana jest jedna tablica routingu zawierająca wszystkie niezbędne informacje. W PBR definiuje się oddzielne tablice routingu dla każdej tworzonej reguły (np. sieć źródłowa, port). Dzięki temu możliwym jest różnicowanie ścieżek, którymi przesyłane są pakiety w zależności od źródła transmisji (konkretny użytkownik, usługa).

## 1.4 Zastosowania technik zapewniania jakości

Zapewnianie jakości transmisji danych jest bardzo istotną kwestią. Prowadzonych jest bardzo dużo badań dotyczących tego zagadnienia. Jakość transmisji to bardzo rozległe pojęcie i tak np. SLA nie jest bezpośrednio techniką służącą do zapewniania jakości, jednakże jest ściśle związane z tym zagadnieniem. Pojęcie to jest wykorzystywane zarówno w przypadku szeroko pojętego zapewniania jakości, jak i w przypadku konkretnych rozwiązań sieciowych. W [8] poruszono problem dotyczący zapewniania jakości w dużych sieciach komputerowych. Z kolei kwestie związane z SLA w przypadku sieci wykorzystujących konkretne protokoły sieciowe poruszono w [58, 59, 60]. Bardzo istotną kwestią jest, przy okazji zapewniania jakości umowa zawarta pomiędzy dostawcą, a klientem.

W [61, 62, 63] zaproponowano rozwiązanie stanowiące kompletną architekturę służącą do gwarantowania przepływności w sieciach Ethernet/IP [64]. Tego typu sieci są jednymi z najbardziej popularnych, jak chodzi o zastosowania lokalne. Najczęściej do zapewniania jakości w sieciach IP wykorzystywany jest DiffServ [65] w kombinacji z odpowiednim protokołem routingu. W przypadku protokołów routingu dynamicznego prowadzone są różnego rodzaju prace nad ich wydajnością [66, 67] oraz odpowiednim doбором – uzależnionym od konkretnego zastosowania i wymagań [68, 69]. Również kwestie związane z centralizacją i decentralizacją procesu jakim jest routing poruszane są w literaturze [23]. Poza protokołami routingu dynamicznego wykorzystać można również PBR, czyli technikę, która umożliwia definiowanie wielu ścieżek routingu pomiędzy różnymi źródłami i tym samym celem, a decyzja odnośnie tego którą ścieżkę wybrać uzależniona jest od adresu źródłowego [55]. Ze względów bezpieczeństwa tworzone są protokoły, których zadaniem jest dostarczanie danych w sposób bezpieczny tzn. uniemożliwiając podgląd przesyłanych informacji oraz ich modyfikację (np. VPN). Takim właśnie protokołem jest IPSec stanowiący rozszerzenie protokołu IP o metody bezpieczeństwa. Również i w przypadku tego protokołu prowadzone są prace nad zapewnieniem odpowiedniej jakości usługi. W [70] poruszono właśnie ten temat i zauważono, iż klasyfikacja ruchu sieciowego mająca miejsce w przypadku stosowania QoS może posłużyć do zwiększenia poziomu bezpieczeństwa - poprzez separację ruchu różnego typu.

Prace nie ograniczają się tylko do sieci Ethernet/IP, ale również do innych dobrze znanych technologii. Wiele kwestii dotyczących zapewniania jakości poruszanych jest w kontekście protokołu MPLS lub GMPLS. W [62] zaproponowano kompletne rozwiązanie służące do zapewniania jakości na całej ścieżce od źródła do celu w sieci wykorzystującej MPLS. W zaproponowanym rozwiązaniu wykorzystano DiffServ, który wspierany jest przez protokół MPLS. Wspomniany protokół jest także

badany pod kątem jakości w przypadku wykorzystania rozszerzenia TE [71, 72, 73]. Także w przypadku protokołu GMPLS prowadzone są badania dotyczące rozszerzenia TE [74, 75, 76]. Aby możliwym było gwarantowanie jakości bardzo ważne jest monitorowanie parametrów, które mają na nią wpływ np. opóźnienie transmisji, fluktuacja opóźnienia, współczynnik utraty pakietów, przepływność [77, 78]. Istotną kwestią w przypadku MPLS jest analiza również pod kątem protokołu routingu wykorzystywanego w sieci [79, 80]. Zazwyczaj jest to OSPF [81] lub OSPF-TE w kombinacji z rozwiązaniem DiffServ, choć również prowadzone są prace nad tzw. wirtualną topologią sieciową, która może ulec zmianie w zależności od natężenia ruchu [82].

Dzięki odpowiedniemu doborowi technik routingu możliwym jest definiowanie różnych ścieżek pomiędzy źródłem, a celem. Co za tym idzie możliwe jest uzależnienie wyboru ścieżki routingu od wymaganych parametrów jakościowych, np. przepływności [83, 84, 85] czy prawdopodobieństwa blokowania ruchu [86, 87]. Metody te wymagają przeprowadzenia klasyfikacji ruchu sieciowego, która może zostać wykonana z wykorzystaniem np. IntServ, DiffServ lub klasycznej metody transmisji bez gwarancji [88, 89]. Jednakże najczęściej badania prowadzone są wyłącznie pod kątem DiffServ [90, 91, 92]. Proponowane są również rozwiązania służące do agregacji klas ruchu co może wpłynąć na redukcję liczby ścieżek w sieci [93]. Zagadnienie TE może zostać również rozpatrzone osobno pod kątem każdej klasy DiffServ [94, 95]. W przypadku routingu z wieloma ścieżkami bardzo istotną kwestią jest odpowiednie rozłożenie ruchu tak, aby nie doprowadzić do przeciążenia łącza nadmiernym ruchem [96, 97].

Należy również pamiętać, że MPLS jest rozwiązaniem stosowanym w sieciach rozległych, a w tych z kolei zapewnianie jakości jest utrudnione, ponieważ dane przechodzą przez wiele domen, w ramach których definiowane mogą być różne mechanizmy. W [98] skupiono się na przepływności i niezawodności w ramach sieci rozległej, w której wykorzystano MPLS-TE. Dzięki wykorzystaniu odpowiednich technik odciążono ścieżki i zwiększono przepływność oraz niezawodność.

Bardzo popularną techniką sieciową na przestrzeni ostatnich kilku lat stał się SDN. W związku z tym pojawiają się metody, których zadaniem jest umożliwienie migracji ze starszych technologii do nowoczesnego SDN [99]. Prowadzonych jest coraz więcej badań dotyczących SDN, a także jakości w przypadku stosowania tej techniki [100, 101] oraz tworzone są testy porównujące SDN i inne dobrze znane techniki – np. routingu klasyczny [102]. Ogólne badania najczęściej dotyczą przypadków użycia SDN [103, 104, 105], symulacji oraz emulacji rozwiązań [106, 107]. Istotną kwestią są również metody zarządzania siecią, w której stosowany jest SDN [108,



109, 110]. Z założenia zarządzanie siecią powinno odbywać się w sposób scentralizowany. Ze względów wydajnościowych rozwiązania rozważane jest również podejście zdecentralizowane, jak chodzi o zarządzanie siecią [35, 111]. Zanim możliwym będzie gwarantowanie jakości świadczonej usługi [112] należy zastanowić się w jaki sposób będzie ona monitorowana i oceniana [113, 114, 115]. Zasobem, którego wykorzystanie można monitorować, oceniać i odpowiednio alokować jest np. przepływność [116, 117, 118] czy zapewnienie ścieżki źródło-cel [119]. Dla odpowiedniej alokacji można skorzystać z mechanizmów równoważenia obciążenia [120, 121].

Dla SDN definiowane są dwie główne metody zapewniania jakości – różnicowanie ścieżek [122, 123, 124] lub priorytetyzacja z kolejkowaniem transmisji [125]. W [126] zaproponowano rozszerzenie służące do zapewniania jakości dla sieci SDN działające na zasadzie różnicowania ścieżek oraz co istotne, rozwiązanie to może zostać zastosowane w przypadku sieci z wieloma operatorami dzięki wymianie informacji pomiędzy domenami. Z kolei zapewnianie jakości poprzez odpowiednią priorytetyzację i kolejkowanie zostało zaproponowane w [127, 128, 129]. Nadawanie odpowiednich priorytetów strumieniom informacji, a następnie ich odpowiednie kolejkowanie wydaje się być interesującym rozwiązaniem zwłaszcza w przypadku scentralizowanego SDN, w którym reguły definiowane są centralnie. Problemem może okazać się czas uzyskiwania dostępu do reguł.

W przypadku sieci komputerowych, należy również zastanowić się nad kwestiami związanymi z bezpieczeństwem. W kontekście SDN, ogólnej analizy bezpieczeństwa dokonano w [130]. Wydaje się, iż najbardziej podatny jest kontroler i w związku z tym, to właśnie jego działanie powinno podlegać kontroli dotyczącej bezpieczeństwa [131, 132, 133].

SDN może być bez problemów stosowany w sieciach lokalnych, zauważono natomiast problemy z jego stosowalnością w sieciach rozległych. Dystrybucja reguł może okazać się problematyczna ze względu na wielu operatorów, którzy w różny sposób definiować mogą zasady działania sieci [134, 135, 136].

W przypadku routingu zastosowana może zostać również koncepcja wykorzystania reguł routingu determinowanych przez źródło transmisji (z ang. source routing) – dzięki takiemu podejściu nadawca może sam zdefiniować, jak ma wyglądać ścieżka transmisji do celu [54]. Również w przypadku starszych technologii sieciowych prowadzono badania dotyczące zapewniania jakości. Ogólny zarys kwestii dotyczących jakości w ATM, z uwzględnieniem sieci z wieloma operatorami pojawił się w [137]. Kolejne badania dotyczyły przepływności i charakterystyki ruchu [138, 139, 140]. Również kwestie routingu są istotne z punktu widzenia zapewniania jakości [141, 142]. Priorytetyzacja danych czy to z wykorzystaniem DiffServ [143], czy bez niego [144] również stały się obiektami eksperymentów. Istotnym aspektem

jest bezpieczeństwo sieci i równoczesne zapewnianie jakości. Ze względu na dodatkowe obciążenie sieci związane z negocjacją parametrów bezpiecznego połączenia może okazać się, iż jakość w takiej sieci zostanie obniżona (dodatkowe opóźnienia, błędy, ograniczona przepływność). W [145] zaproponowano protokół, którego zadaniem jest zapewnienie odpowiedniego poziomu bezpieczeństwa bez pogarszania jakości transmisji.

Szeroki przegląd dotyczący zapewniania jakości przedstawiono w [146] – poruszono tę kwestię w kontekście sieci wykorzystujących ATM, GMPLS oraz MPLS. Kolejnym typem sieci, w ramach której można analizować jakość są sieci P2P (z ang. Peer-to-peer) [147].

Kwestie dotyczące jakości poruszone zostały również w kontekście sieci rozległych [148, 149, 150] i dostępowych, takich jak np. DSL [151, 152]. QoS może być rozważany także w odniesieniu do warstw modelu OSI (warstwa łącza danych i sieciowa) bez wskazywania konkretnych protokołów [153, 154], co stanowić może bardzo interesujące podejście. Różne rozwiązania proponowane są również w przypadku warstwy transportowej modelu OSI, a konkretnie protokołu TCP [155], który służy do niezawodnego dostarczania segmentów danych. Innym przykładem rozwiązań dotyczących zapewniania jakości jest odpowiednie wykorzystanie protokołów routingu niezależnie od technologii sieciowej. W [156] zaproponowano rozwiązanie wykorzystujące protokół OSPF oraz różnicowanie ścieżek. Często zdarza się, iż analiza dokonywana jest nie tylko pod kątem konkretnych technologii sieciowych, ale i usług jakie w danej sieci są świadczone. Przykładowo w [157] zaproponowano rozwiązanie służące do monitorowania i gwarantowania jakości telewizji internetowej w sieci wykorzystującej TCP/IP. Jakość może być także analizowana pod kątem wydajności i niezawodności świadczonej usługi [158]. W konkretnych zastosowaniach może okazać się, że ogólna kategoryzacja danych nie wystarczy i wystąpi konieczność zdefiniowania innych klas ruchu sieciowego [159]. Definicja klas w takim wypadku dokonywana może być na podstawie charakterystyki danej sieci.

## Rozdział 2

# Agenty

W rozdziale tym scharakteryzowano systemy agentowe. Przedstawiono koncepcję agentowości jako ogólny zarys tego podejścia, a następnie opisano wykorzystanie agentów w sieciach komputerowych. Ostatni element niniejszego rozdziału stanowi opis metod rynkowych zakupu dóbr tj. teoria gier oraz aukcje. W niniejszej pracy do zarządzania siecią komputerową wykorzystano system agentowy stąd też istotne było przybliżenie tej koncepcji.

### 2.1 Koncepcja agentowości

Systemy agentowe to koncepcja o szerokim spektrum zastosowań, będąca przedmiotem wielu badań. Głównym celem stosowania systemów agentowych jest poprawa sposobu projektowania i wdrażania oprogramowania. Agent stanowi pośrednika pomiędzy użytkownikiem, a pewnym środowiskiem, w którym się znajduje. System postrzegany jest przy pomocy sygnałów, a na wyjściu agent tworzy akcje wpływające na środowisko. Agent reprezentuje użytkownika. W ogólnym ujęciu agent powinien posiadać następujące cechy [22, 160]:

- Reaktywność – agent powinien być w stanie monitorować środowisko, w którym się znajduje i reagować w określonym czasie na zmiany które wystąpiły,
- Proaktywność – agent powinien wykazywać zachowanie zorientowane na osiągnięcie konkretnego, określonego celu poprzez podejmowanie własnej inicjatywy,
- Zachowania społeczne – agent powinien wchodzić w interakcję z innymi agentami.

Agent powinien być systemem autonomicznym (zdolność samodzielnego działania bez ludzkiej ingerencji), powinien kooperować z innymi agentami (z wykorzystaniem pewnego wspólnego protokołu) oraz powinien być w stanie uczyć się (zmiana zasad reagowania w oparciu o zaobserwowane zależności i zachowania w

trakcie działania agenta). Nie wszystkie warunki muszą być spełnione równocześnie. W zależności od spełniania poszczególnych warunków można w różny sposób klasyfikować systemy agentowe. Systemy agentowe można podzielić ze względu na [161]:

- Mobilność – zdolność przemieszczania się w obszarze określonej sieci,
- Sposób reagowania – może następować bezpośrednie przełożenie stanu środowiska na decyzję lub działanie na podstawie pewnego modelu, obserwacji środowiska i kooperacji z innymi agentami,
- Posiadane atrybuty – autonomiczność, zdolność do kooperacji, uczenie się,
- Pełnione funkcje – zależnie od roli jaką pełni dany agent w danym środowisku.

W oparciu o posiadane atrybuty wprowadzono cztery kategorie agentów [161]. *Collaboration agent* jest w stanie kooperować z innymi agentami oraz stanowi system autonomiczny, *interface agent* jest autonomiczny i posiada zdolność uczenia się. Z kolei system, który jest w stanie uczyć się oraz kooperować z innymi agentami określany jest jako *collaborative learning agent*. Agent posiadający wszystkie atrybuty określany jest mianem *smart agent*.

Tak jak w przypadku agentów, również w przypadku środowisk wprowadzono pewną klasyfikację [22]. Klasyfikacji dokonano na podstawie cech takich jak:

- Dostęp do systemu – agent może uzyskać pełną, kompletną i aktualną informacją o stanie środowiska lub informację częściową, ograniczoną w pewien sposób,
- Reakcja systemu – system może reagować w taki sposób, że pewnej akcji zawsze będzie towarzyszył zagwarantowany efekt, bez pewności co do stanu będącego wynikiem działania,
- Dynamika systemu – system może pozostać niezmienny tak długo, aż agent nie wykona jakiejś operacji lub system może być dynamicznie modyfikowany przez wiele różnych podmiotów działających w nim,
- Ciągłości działania – system może działać przez pewną określoną liczbę akcji i stanów, w których się znajdzie lub w sposób nieprzerwany, ciągły.

Systemy agentowe mogą zostać podzielone na wiele różnych sposobów – ze względu na sposób działania, pewne cechy. Wyróżnić można następujące koncepcje agentów [22, 162]:

- Inteligentny – to agent, który jest w stanie uczyć się w trakcie działania oraz wykorzystać swoją wiedzę, aby osiągnąć zamierzony cel,
- Autonomiczny – jest w stanie modyfikować metody osiągnięcia określonego celu lub celów w trakcie swojego działania,
- Rozproszony – agent uruchomiony na wielu maszynach równocześnie,
- Multi-agent systems – są to rozproszone agenty, które współpracują ze sobą, aby osiągnąć założony cel,
- Mobilny – agent jest w stanie przemieszczać się pomiędzy różnymi systemami.

Agenty podzielić można również ze względu na ich architekturę. Z punktu widzenia architektury i klasy wyróżnić można następujące agenty [163]:

- Logiczny - decyzja o akcji podejmowana jest na podstawie wnioskowania logicznego [162]. Tę architekturę zaliczyć można do klasy *Simple Reflex*, tzn. takiej w której decyzja podejmowana jest wyłącznie na podstawie stanu obecnego bez względu na to co było kiedyś,
- BDI – w architekturze tej założono dwuetapowy proces, w którym najpierw ustalany jest konkretny cel do osiągnięcia, a następnie ustalany jest sposób jego osiągnięcia [162]. Architekturę tą zaliczyć można do klasy *goal-based*, czyli takiej w której istotne jest osiągnięcie wyłącznie określonego celu,
- Reaktywny – sytuacje w środowisku są bezpośrednio mapowane na podejmowane akcje, zasada działania zbliżona jest do automatu skończonego [162, 164]. Ta architektura zgodna jest z założeniami klasy *simple reflex*.
- Kognitywny – złożony model agenta uwzględniający środowisko oraz inne agenty w nim działające [164]. W tym przypadku można mówić o klasie *model-based reflex* w ramach której środowisko obserwowane jest częściowo, a agent zapisuje swój aktualny stan. Decyzja podejmowana jest na podstawie historii stanów z wykorzystaniem wewnętrznego modelu.
- Deliberatywny – w architekturze tej agent uwzględnia otaczające środowisko oraz swoje istnienie [164]. Tę architekturę zaliczyć można do klasy *utility-based* oraz *learning*. Klasa *utility-based* określa działanie w celu osiągnięcia określonego celu z tym, że celów może być więcej, a ostateczna decyzja podejmowana jest zawsze na podstawie pożyteczności celu, który może zostać osiągnięty. Z kolei klasa *learning* określa, iż agent jest w stanie uczyć się w trakcie działania i wykorzystać swoją wiedzę, aby osiągnąć zamierzony cel.

Bardzo istotną kwestią jest rozróżnienie agentów od programów – nie każdy program może zostać określony mianem agenta. Cechami odróżniającymi agenty od zwykłego programu są jego podstawowe cechy tzn. reaktywność, proaktywność, autonomia. Zwykle programy komputerowe działają przez pewien określony czas wykonując określone operacje. Agent jest w stanie działać niezależnie tak długo, jak jest to konieczne obserwując przy tym środowisko oraz ucząc się określonych reakcji w oparciu o to, co zaobserwował. Z kolei odróżnić agenty od obiektów pozwalają cechy takie, jak większa autonomia, reaktywność, proaktywność, kooperacja. W kontekście systemów ekspertowych agenty posiadają cechy reaktywności i proaktywności oraz są w stanie kooperować. Tych cech systemy ekspertowe nie posiadają. W ujęciu sztucznej inteligencji agent może być odrębną maszyną, może wchodzić w interakcję ze środowiskiem ludzkim i z czymkolwiek innym, co wykazuje cechy umożliwiające osiągnięcie założonego celu.

## 2.2 Alokacja dóbr

Do alokacji dóbr, takich jak np. jakość transmisji wykorzystać można aukcje [165, 166]. Aukcje stanowią mechanizm wykorzystywany do osiągnięcia porozumienia w kwestii alokacji dóbr [167]. W tym przypadku dobrem może być dowolny przedmiot czy usługa. Jeśli dane dobro pożądane jest przez więcej niż jeden podmiot wówczas przydatne stają się aukcje. Z kolei, gdy dobro pożądane jest wyłącznie przez jeden podmiot, aukcje nie mają zastosowania – wystarczające są negocjacje dwustronne (sprzedawca – kupujący). Aukcje zazwyczaj prowadzone są pomiędzy jednym oferentem, a wieloma klientami. Celem jest alokacja dóbr na rzecz jednego z klientów. Generalną zasadą aukcji jest to, iż sprzedający dąży do maksymalizacji zysku, a kupujący do minimalizacji ceny. Każdy z uczestników posiada własną strategię, która nie powinna zostać ujawniona pozostałym uczestnikom. Na strategię wpływ ma osobista wycena przedmiotu przez uczestnika aukcji. Można wyróżnić trzy typy: wartość wspólna, wartość prywatna oraz skorelowana wartość. W przypadku wartości wspólnej, dla każdego uczestnika aukcji dany przedmiot wart jest dokładnie tyle samo. Z kolei, gdy mowa jest o wartości prywatnej, dany przedmiot dla każdego uczestnika może mieć różną wartość wynikającą np. z potencjału danego przedmiotu, jego przydatności, wartości sentymentalnej. Gdy dany przedmiot miałby zostać odsprzedany dalej przez uczestnika aukcji to jego wartość wyznacza korzyść, która może zostać osiągnięta przez uczestnika aukcji. Wówczas mowa o wartości skorelowanej.

Aukcje przebiegają zgodnie z określonym protokołem, czyli zasadami, które dotyczą sposobu prowadzenia aukcji oraz jej parametrów [167]. W ramach protokołu

bardzo istotne jest zdefiniowanie kto po zakończeniu aukcji uzyska dobro. Zazwyczaj uczestnik, który złożył najlepszą ofertę (najwyższa cena) wygrywa i otrzymuje przedmiot aukcji. W takim wypadku mowa jest o tzw. aukcjach *first-price*. Inny przypadek to tzw. aukcje *second-price* w których wygrywa klient oferujący najwięcej, lecz płaci drugą co do wielkości cenę. Ze względu na jawność informacji o działaniach poszczególnych uczestników aukcje można podzielić na *open-cry* i *sealed-bid*. Aukcje *open-cry* prowadzone są na zasadzie pełnej informacji tzn. działania uczestników są widoczne dla pozostałych uczestników. Z kolei w aukcjach *sealed-bid*, uczestnik nie widzi działań pozostałych uczestników. Aukcje mogą przebiegać w jednej rundzie (tzw. *one shot*) lub w wielu rundach. Aukcja, w której jest wyłącznie jedna runda, przebiega w taki sposób, iż uczestnicy składają ofertę raz po czym następuje rozstrzygnięcie aukcji. W przypadku, gdy aukcja przeprowadzana jest w wielu rundach, w każdej z nich uczestnicy składają ofertę, a rozstrzygnięcie następuje np. po określonym czasie lub liczbie rund. Jeśli aukcja prowadzona jest w wielu rundach, to przyjąć można dwa różne mechanizmy jej przebiegu – sprzedający ustala cenę początkową, a uczestnicy w kolejnych rundach składając oferty podnoszą cenę, jest to tzw. aukcja *ascending*. Z kolei, gdy sprzedający ustala cenę, a następnie obniża ją w kolejnych rundach tak długo aż nie pojawi się oferta, to mowa o tzw. aukcji *descending*.

Aukcje dotyczyć mogą jednego lub wielu dóbr. W przypadku aukcji jednego dobra wyróżnić można kilka różnych protokołów determinujących przebieg aukcji. Aukcje *First-price-sealed-bid* należą do grupy aukcji *first-price*, *sealed-bid*, *one-shot*. Aukcja przeprowadzana jest w jednej rundzie, a w ramach niej uczestnicy składają oferty. Wygrywa uczestnik, który złoży najwyższą ofertę. Uczestnicy nie mają możliwości zwiększenia wartości swoich ofert. Aby wygrać aukcję tego typu uczestnik od razu powinien złożyć ofertę stanowiącą przyjęte przez niego maksimum. Podobnym protokołem prowadzenia aukcji jest *Second-price-sealed-bid* (Vickrey). Różnica polega na tym, że wygrywający aukcję płaci kwotę równą drugiej najwyższej ofercie. Wygrywającym nadal pozostaje uczestnik, który zaoferował najwięcej. Ten typ aukcji jest interesujący z punktu widzenia strategii graczy. Jeśli gracz złoży ofertę wyższą niż swoje maksimum, możliwe jest to, że wygra aukcję, przy czym zapłacić będzie musiał cenę równą drugiej najwyższej ofercie. W takim przypadku pojawia się ryzyko, że gracz będzie musiał zapłacić więcej niż swoje maksimum i wówczas poniesie stratę. Najbezpieczniejszym rozwiązaniem jest złożenie oferty zgodnej z przyjętym maksimum. Ponieważ oferty składane są w sposób tajny ważne jest, aby sprzedający był uczciwy, ponieważ to on informuje zwycięzcę o tym, ile ma zapłacić, czyli jaka była druga najwyższa oferta.

Aukcje angielskie (z ang. English auctions) [168, 169] są jednym z najbardziej

znanych typów aukcji. Są to aukcje należące do grupy *first-price, open-cry, ascending*. Przebieg jest następujący:

1. Sprzedający wystawia przedmiot na sprzedaż określając jego wyjściową cenę.
2. Uczestnicy aukcji składają oferty, z których każda musi być wyższa niż obecna.
3. W momencie, gdy uczestnicy przestaną składać oferty aukcja jest kończona. Przedmiot otrzymuje uczestnik, który zaoferował najwięcej.

Czasem w przypadku tego typu aukcji spotkać można sytuację, w której sprzedający poza podaniem ceny wyjściowej, podaje również minimalną cenę za jaką jest w stanie sprzedać dobro. Taki sposób działania zaobserwować można np. w przypadku aukcji prowadzonych za pośrednictwem sieci internet. Kolejne rozwiązanie, z którym można się spotkać w przypadku aukcji internetowych jest określenie czasu trwania aukcji. Nawet jeśli uczestnicy przestali składać oferty aukcja nie kończy się. Sprzedający określa na początku aukcji, jak długo będzie ona trwać. Po upływie wskazanego czasu aukcja kończy się i wówczas następuje jej rozstrzygnięcie – wygrywa uczestnik, który zaoferował najwięcej lub w przypadku braku ofert aukcja pozostaje nierozstrzygnięta. Z punktu widzenia uczestnika najlepszą strategią jest podnoszenie oferty każdorazowo o małą kwotę tak długo, aż osiągnie przyjęte przez niego maksimum. Gdy uczestnik nie jest już w stanie złożyć kolejnej oferty wycofuje się z aukcji. Kolejną metodą przeprowadzania aukcji są tzw. aukcje holenderskie (z ang. Dutch auctions) [22, 170]. Ten typ aukcji należy do grupy *first-price, open-cry, descending*. Przebieg jest następujący:

1. Sprzedający jako cenę wyjściową przyjmuje pewną bardzo wysoką kwotę.
2. Sprzedający obniża cenę o pewną małą wartość tak długo, aż nie znajdzie się uczestnik, który jest w stanie zapłacić tą aktualną kwotę.
3. Aukcję wygrywa uczestnik, który złożył ofertę.

Z kolei aukcje Japońskie (z ang. Japanese auctions) należą do grupy *first-price, open-cry, ascending* [167]. W pierwszej kolejności sprzedający podaje cenę wyjściową, a uczestnicy decydują czy są w stanie ją zapłacić. Następnie w każdej rundzie cena jest podnoszona przez sprzedającego, a uczestnicy deklarują czy są w stanie zapłacić aktualną cenę. Deklaracja uczestnika oznacza kontynuację udziału w aukcji. Jeśli uczestnik zrezygnuje z udziału w danej rundzie to nie ma już możliwości wzięcia udziału w kolejnej. Aukcja kończy się w momencie, gdy w danej rundzie pozostał (złożył ofertę) tylko jeden uczestnik. Wygrywający płaci zaoferowaną przez siebie cenę.



Aukcje wielu dóbr (kombinatoryczne) [171] są bardziej skomplikowane, ponieważ wiele czynników może równocześnie wpływać na przebieg aukcji. Przykładowo jako dobro oferowane w ramach aukcji postrzegać można cenę produktu, jego ilość czy jakość. Uczestnicy zazwyczaj mają preferencje co do kombinacji oferowanych dóbr. Uczestnicy mogą skorzystać z różnych metod determinowania ofert. Jedną z metod są tzw. oferty XOR. Metoda ta polega na tym, że określanych jest wiele różnych ofert, ale uczestnik zapłaci co najwyżej kwotę zgodnie z jedną z nich. Jeśli równocześnie możliwym jest spełnienie więcej niż jednego warunku, wówczas uczestnik płaci najwyższą z kwot pojedynczej oferty. Przykładowo, jeśli dostępne są dobra  $d_1, d_2, d_3, d_4, d_5$  użytkownik może złożyć ofertę Of taką, że:

$$Of = (\{d_3, d_4, d_5\}, 100) XOR (\{d_2\}, 20) \quad (2.1)$$

Taka oferta oznacza, że użytkownik jest w stanie zapłacić 100, jeśli otrzyma dobro  $d_3, d_4, d_5$ , 20 jeśli otrzyma dobro  $d_2$  oraz 100, jeśli otrzyma dobro  $d_2, d_3, d_4, d_5$

Kolejnym przykładem ofert są tzw. oferty OR. Również i w tym przypadku, definiowanych jest wiele różnych ofert, a oferta końcowa składa się ze wszystkich ofert częściowych dla których spełniony był warunek. Przykładowo, jeśli dostępne są dobra  $d_1, d_2, d_3, d_4, d_5$  użytkownik może złożyć ofertę Of taką, że:

$$Of = (\{d_3, d_4, d_5\}, 100) OR (\{d_2\}, 20) \quad (2.2)$$

Oznaczać to będzie, że użytkownik jest w stanie zapłacić 100, jeśli otrzyma dobro  $d_3, d_4, d_5$ , jeśli otrzyma dobro  $d_2$  jest w stanie zapłacić 20, a gdy otrzyma dobro  $d_2, d_3, d_4, d_5$  może zapłacić 120.

Dla aukcji wielu dóbr trudnym staje się określenie zwycięzcy. Przykładowo, jeśli uczestnicy określą, że są w stanie dokonać zakupu określonej ilości (wagi) materiału za określoną cenę, to pojawia się problem komu sprzedać materiał. Najprostszą metodą w takim przypadku byłoby obliczenie ceny jednostkowej materiału. Problem pojawi się w momencie, gdy cena jednostkowa dla różnych wag materiału okaże się taka sama – czy bardziej opłacalne jest sprzedanie większej ilości materiału czy mniejszej, skoro cena jednostkowa jest taka sama. W takim przypadku wyłonienie zwycięzcy uzależnione jest od osobistych preferencji sprzedającego, które powinny zostać sprecyzowane w warunkach aukcji.

Opisane metody prowadzenia aukcji mogą zostać wykorzystane w sieciach komputerowych. Wszystkie metody i techniki wykorzystywane do gwarantowania jakości mogą bezpośrednio wpływać na kształtowanie cen. W momencie, gdy usługa

świadczona jest na określonym poziomie i ten poziom jest gwarantowany, możliwym staje się uzależnienie jakości transmisji od jej ceny [172, 173]. Problem ten jest przedmiotem badań, a propozycje wraz z wynikami opublikowane zostały w [174, 175]. Zapewnianie jakości transmisji jest bardzo istotne biorąc pod uwagę wykorzystanie sieci komputerowych w dzisiejszych czasach. Obserwować można stały wzrost liczby usług świadczonych przez sieć oraz stały wzrost liczby użytkowników sieci.

## 2.3 Agenty w sieciach komputerowych

W sieciach komputerowych często wykorzystuje się agenty. W zależności od konkretnego zastosowania mogą to być bardzo proste agenty, stanowiące pośredników w transmisji lub bardziej zaawansowane rozwiązania. Jednym z przykładów prostego agenta jest tzw. *user agent* wykorzystywany przy przeglądaniu stron internetowych [176]. Stanowi on pośrednika pomiędzy użytkownikiem, a serwerem stron internetowych. Agent ma za zadanie informowanie serwera o wykorzystywanym przez klienta oprogramowaniu (przeglądarka internetowa, system operacyjny). Informacje te są wykorzystywane m.in. do prowadzenia statystyk odwiedzin stron internetowych lub na potrzeby wyboru wariantu strony internetowej w zależności od typu urządzenia klienta. *User agent* przekazywany jest w nagłówku HTTP, składa się z nazwy produktu, wersji produktu oraz silnika układu i jego wersji. Są to podstawowe informacje, które mogą zostać rozszerzone o szereg dodatkowych elementów.

Kolejnym przykładem wykorzystania agentów w sieciach jest tzw. MTA (z ang. mail transfer agent lub message transfer agent). MTA odpowiada za transfer wiadomości email od nadawcy do odbiorcy [177]. Jego zadaniem jest zarówno wysyłanie, jak i odbiór wiadomości – w momencie, gdy nadawca wyśle wiadomość, serwer pocztowy podejmie próbę doręczenia jej do serwera odbiorcy. Zarówno po stronie serwera nadawcy, jak i odbiorcy musi działać odpowiednie oprogramowanie. Serwery poczty elektronicznej działają w architekturze klient-serwer. Z kolei, aby możliwym było przekazanie wiadomości pomiędzy serwerem odbiorcy, a odbiorcą na serwerze działać musi tzw. MDA (z ang. mail delivery agent), czyli oprogramowanie odpowiedzialne za doręczenie wiadomości do celu.

Agenty mogą być stosowane również w przypadku bardziej skomplikowanych zadań na różnych płaszczyznach działania sieci komputerowych [178]. Przykładem

takiego zadania jest automatyczne zarządzanie sieciami komputerowymi. Zarządzanie polega na monitorowaniu, wykrywaniu usterek oraz dynamicznym modyfikowaniu czy rekonfigurowaniu sieci, gdy jest to konieczne [179, 180, 181]. Rekonfiguracja może sprowadzać się do zmiany rezerwacji zasobów wykorzystywanych w czasie transmisji [182, 183] lub do zmiany ścieżek transmisji. Zmiana ścieżek transmisji, czyli modyfikacja tablic routingu prowadzi do zmiany trasy, którą przesyłane są pakiety pomiędzy źródłem, a celem [184, 185]. Proces rekonfiguracji można rozważać w związku z potrzebą równoważenia obciążenia łączy [186], a także np. zapewniania jakości usługi (QoS) [187]. Dla protokołu MPLS proponowane są rozwiązania w których wykorzystuje się agenty w procesie trasowania [188, 189, 190]. Prowadzone są również prace dotyczące wykorzystania agentów w sieciach GMPLS dla zapewniania QoS poprzez rozbudowany mechanizm sygnalizacji [191]. Dla sieci IP także proponowane są rozwiązania pozwalające na dynamiczne negocjacje, gwarantowanie i monitorowanie QoS [192]. Tego typu koncepcje są bardzo istotne dla usług transmisji wideo i dźwięku [193]. Także w przypadku SDN pojawiają się koncepcje wykorzystania agentów. Ich wykorzystanie ma na celu skuteczniejsze zapewnianie jakości usługi [194, 37].

Zarządzanie w sieciach komputerowych w sposób zdecentralizowany stanowi alternatywę dla klasycznych metod scentralizowanych [195, 196, 197]. Definiowane są nowoczesne modele agentów, którzy są w stanie dostarczyć tego typu funkcjonalności [198, 199, 200]. W przypadku decentralizacji proponowanym rozwiązaniem jest wykorzystanie agentów programowych [174, 201, 202]. Innym zastosowaniem dla agentów w sieciach komputerowych jest taryfikacja. Agenty mogą zostać wykorzystane w przypadku, gdy w sieci mają być stosowane metody rynkowe do określania ceny usługi transmisji danych z określonymi parametrami [166, 203, 204, 205].

Działanie sieci komputerowych oparte jest na sterowaniu ruchem (z ang. traffic management), które polega na klasyfikacji strumieni ruchu, analizie wymagań jakościowych QoS, definiowaniu trasy, definiowaniu i realizacji aktywnych metod zarządzania ruchem w węzłach. Zastosowanie agentów na potrzeby zapewniania parametrów jakościowych oraz do rezerwacji zasobów jest wciąż nowym podejściem.



## Rozdział 3

# Materiały i metody

W ramach tej pracy proponowana jest koncepcja Pay&Require (P&R) której głównym założeniem jest to, że użytkownik płaci za parametry opisujące jakość transmisji danych, które są mu gwarantowane. W niniejszym rozdziale zaprezentowano koncepcję P&R, a także przedstawiono metody wykorzystane w badaniach.

### 3.1 Koncepcja Pay&Require

W ramach koncepcji P&R przyjęto, iż użytkownik płacąc za określone parametry transmisji (jakość) otrzymuje usługę na określonym poziomie. Założenie to jest kluczowe, ponieważ w wielu przypadkach użytkownicy płacą za maksymalne możliwe do uzyskania parametry transmisji. Niestety w praktyce może to oznaczać, iż użytkownik nigdy nie otrzyma parametrów za które zapłacił. Postulat dostarczenia jakości usługi wymaganej przez klienta można zrealizować w taki sposób, że użytkownik płaci za zagwarantowaną jakość transmisji, czyli parametry pomiędzy węzłem źródłowym, a docelowym. Jakość gwarantowana jest poprzez wykorzystanie odpowiedniego protokołu routingu, stanowiącego połączenie protokołu statycznego i dynamicznego oraz technologii agentowej. Jakość transmisji można odnieść do najczęściej stosowanych parametrów QoS, takich jak opóźnienie, fluktuacja opóźnienia, czas transmisji oraz współczynnik utraty pakietów [6]. W oparciu o te parametry, możliwym jest zdefiniowanie poziomów jakości. Istotnym elementem koncepcji jest to, w jaki sposób jakość transmisji ma być gwarantowana. W sytuacji, w której użytkownik płaci za określone parametry, koniecznym jest ich monitorowanie i podejmowanie odpowiednich decyzji w przypadku, gdy odbiegają one nadmiernie od tych przyjętych. W tym celu wykorzystano agenty programowe, których zadaniem jest monitorowanie jakości. Każdy agent nadzoruje jedno lub wiele urządzeń sieciowych. Jego zadaniem jest weryfikacja parametrów i ich zgodności z oczekiwaniami klienta. Koncepcja P&R umożliwia różnicowanie ścieżek transmisji poszczególnych klientów lub grup klientów. Dzięki różnicowaniu ścieżek możliwa jest transmisja z różnymi parametrami jakościowymi. W przypadku klasycznego

rutingu decyzje o wyborze konkretnej ścieżki podyktowane są celem transmisji – w tablicy rutingu istnieje wpis, na podstawie którego, dla danego celu transmisji, ruter kieruje pakiet do kolejnego urządzenia (rutera lub celu transmisji). W koncepcji P&R przyjęto, iż rutingu odbywa się nie tylko na podstawie celu transmisji ale również źródła, dzięki czemu pakiety podróżujące do tego samego celu lecz od różnych źródeł, mogą podróżować zupełnie różnymi ścieżkami. Różne ścieżki mogą oznaczać zupełnie różną jakość transmisji. Istotne jest, aby stwierdzić jaka jest jakość transmisji dla każdej ścieżki i czy w trakcie działania sieci jakość transmisji nie uległa zmianie. Jest to jedno z zadań agentów – odpowiadają one za monitorowanie jakości i rekonfigurację sieci w przypadku, gdy jakość ulegnie zmianie. Przez zmianę rozumieć można zarówno pogorszenie jakości, jak i transmisję z jakością wyższą niż oczekiwana przez klienta. Monitorowanie musi odbywać się w czasie, który można uznać za rzeczywisty tak, aby uniknąć sytuacji, w której jakość transmisji użytkownika jest na innym poziomie niż oczekiwany. W przypadku, gdy agenty stwierdzą zmianę jakości przeprowadzony zostanie proces rekonfiguracji sieci. W procesie tym wyznaczone zostaną nowe ścieżki transmisji. Aby tak się stało poszczególne agenty najpierw wyznaczają wszystkie możliwe ścieżki, dokonują ich oceny w oparciu o jakość poszczególnych łączy, a następnie dokonują wyboru ścieżek rutingu zgodnych z oczekiwaniami klientów. Ocena powinna zostać dokonana w jak najbardziej wiarygodny sposób, aby uzyskane wyniki były rzeczywiste. Dobrym rozwiązaniem jest transmisja pewnej porcji danych poprzez poszczególne łącza i monitorowanie pakietów tak, aby uzyskać parametry świadczące o jakości. W oparciu o uzyskane wyniki dla poszczególnych łączy określić można jakość całej ścieżki. Możliwe są dwa podejścia tzn. przyjęcie wyniku średniego jako oceny całej ścieżki lub działanie zgodnie z zasadą tzw. wąskiego gardła. W przypadku wąskiego gardła najniższa ocena ze wszystkich składowych (poszczególnych łączy) będzie oceną całej ścieżki. Ocena średnia wydaje się nie być miarodajna ponieważ nie przekłada się na rzeczywiste parametry transmisji w ramach całej ścieżki. W związku z tym, zdecydowano o wyborze metody wąskiego gardła do wyznaczania jakości ścieżki w oparciu o parametry łączy.

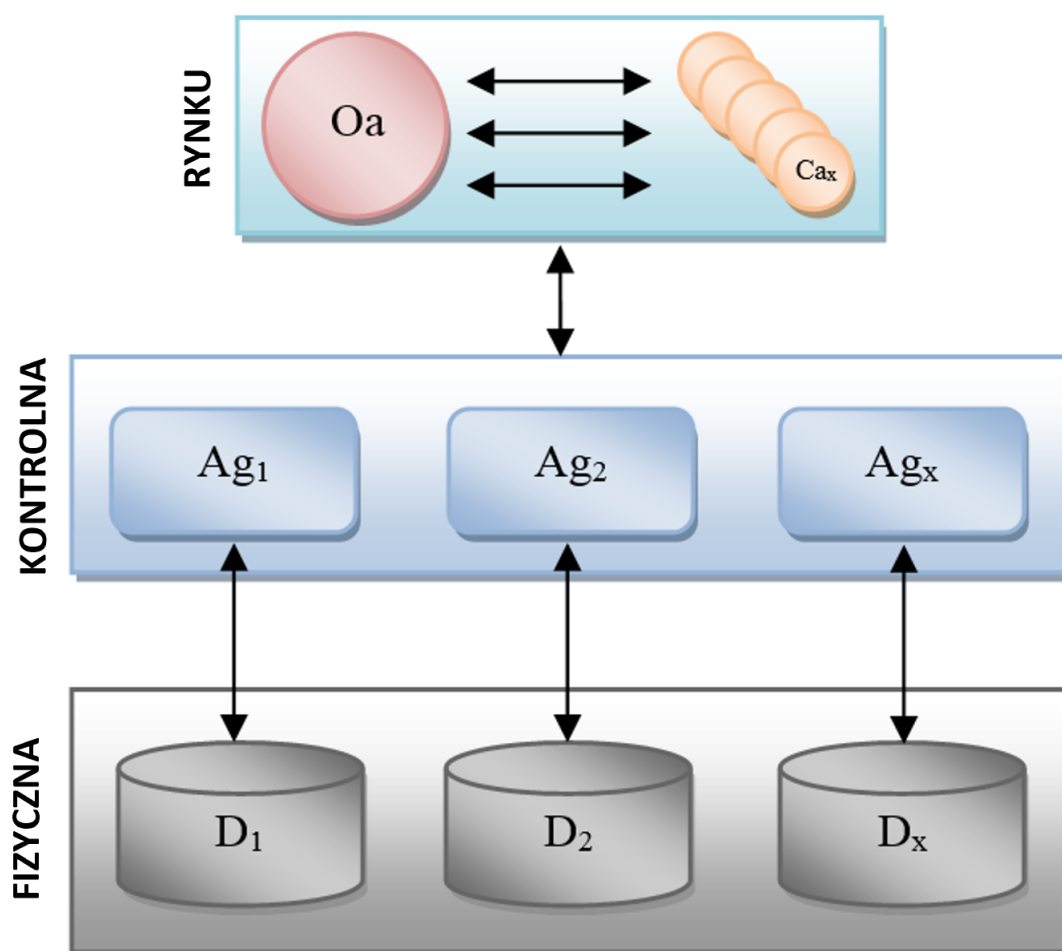
W momencie, gdy wyznaczone zostaną oceny poszczególnych ścieżek następuje weryfikacja wymagań klientów – na tej podstawie wybierane są odpowiednie ścieżki. Po wyborze ścieżek następuje zmiana tablic rutingu. W tym celu agenty przesyłają do urządzeń im podległych odpowiednie komunikaty sterujące. Po przeprowadzeniu rekonfiguracji transmisja odbywa się nowo wybranymi ścieżkami. Dzięki takiemu rozwiązaniu jakość świadczonej usługi jest cały czas na takim samym poziomie. Może zdarzyć się sytuacja, w której w ramach sieci nie jest dostępna jakość zgodna z oczekiwaniami użytkownika, tj.:

- użytkownik zapłacił za niższą jakość niż dostępna – wówczas jakość usługi świadczona będzie na poziomie, który jest dostępny,
- użytkownik zapłacił za wyższą jakość niż dostępna – jakość usługi będzie niższa niż oczekiwana, a użytkownik otrzyma zwrot pieniędzy.

Koncepcja P&R została zdecentralizowana poprzez separację warstwy fizycznej i warstwy kontrolnej. W warstwie fizycznej P&R działają urządzenia sieciowe (routery), z kolei w warstwie zarządzającej działają agenty. Agenty monitorujące parametry odpowiedzialne są za okresowe sprawdzenie aktualnej jakości transmisji. W momencie, gdy parametry uległy zmianie agenty przekażą tę informację do agentów odpowiedzialnych za rekonfigurację. Ten typ agenta w oparciu o jakość transmisji i wymagania klientów określi nowe ścieżki, a następnie przeprowadzi rekonfigurację urządzeń działających w warstwie fizycznej P&R. Można rozważyć dwa poziomy rozwiązania problemu decentralizacji:

- Poziom 1 - każdy agent posiada pełną wiedzę na temat topologii sieci. Tego typu podejście oznacza, że każdy agent udostępnia wszystkim agentom w całej sieci informacje dotyczące sieci podłączonych do routera na którym rezyduje tak, że w efekcie każdy agent posiada pełną informację o wszystkich routerach. Aby osiągnąć spójność takiej informacji niezbędna jest wymiana sporej ilości danych przy każdej zmianie w sieci. Z kolei w przypadku awarii nie ma potrzeby wymiany dodatkowych informacji – agent może natychmiast samodzielnie zaktualizować odpowiednie dane, czyli przeprowadzić rekonfigurację sieci.
- Poziom 2 - agenty posiadają tylko niezbędne informacje tj. na temat sieci przyłączonych do danego routera. Podejście to zakłada, że agent przechowuje w swojej lokalnej bazie tylko te informacje, które są niezbędne z punktu widzenia jego funkcjonowania. Dzięki temu po uruchomieniu agent może w dość krótkim czasie rozpocząć pracę. W przypadku awarii lub zmiany topologii sieci może istnieć konieczność uzyskania dodatkowych informacji. Istotnym rozszerzeniem może być takie rozwiązanie, w którym każdy agent rezydujący na danym routerze posiada częściową informację o całej sieci i pełną informację o najbliższym swoim sąsiedztwie.

Odrębną warstwę stanowią agenty odpowiedzialne za zakup jakości (warstwa rynku). Stanowią one pośredników pomiędzy klientem, a środowiskiem, w którym prowadzone są aukcje czy negocjacje. Również dostawca usługi reprezentowany jest przez agenta. Schemat zdecentralizowanego systemu przedstawiony został na rysunku 3.1.



RYSUNEK 3.1: Model P&amp;R

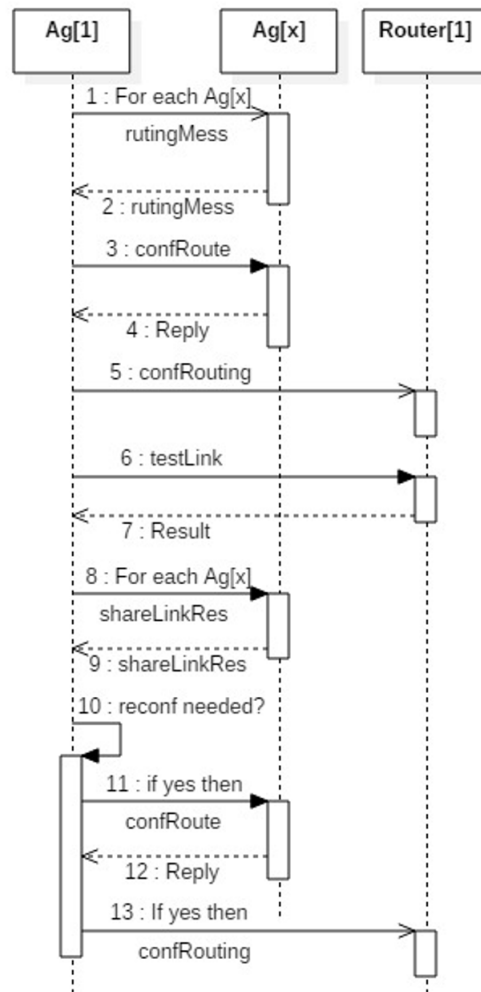


Prezentowane rozwiązanie nie posiada centralnego kontrolera, dzięki czemu nie jest tak podatne na awarie, jak środowisko z centralnym kontrolerem. Agenty poza komunikacją z warstwą fizyczną P&R wymieniają się również niezbędnymi informacjami.

W koncepcji P&R wykorzystano kilka rodzajów agentów. Pierwszym z nich jest agent odpowiedzialny za monitorowanie jakości. W ramach każdego urządzenia działającego w sieci działa przynajmniej jedna instancja agenta monitorującego. Agent tego typu może monitorować całe urządzenie lub poszczególne jego interfejsy – w tym przypadku, w ramach jednego urządzenia działać może wiele agentów monitorujących. Każdy agent tego typu odpowiada za monitorowanie parametrów poszczególnych łączy. Agent może podejmować decyzje w oparciu o z góry ustalone zasady lub w oparciu o własne doświadczenia nabyte w trakcie działania. Agent ten może zostać zaliczony do agentów reaktywnych, czyli takich, których działanie odbywa się na zasadzie mapowania sytuacji w środowisku na konkretną akcję. Decyzja o tym, czy jakość uległa zmianie i konieczna jest określona reakcja nie jest prosta do podjęcia ponieważ parametry świadczące o jakości ciężko jest mapować na zasadzie 1:1. Parametry jakościowe mogą ulec minimalnej zmianie, która nie wpływa w sposób odczuwalny na całkowitą jakość świadczonej usługi transmisji danych. W związku z tym agent powinien działać w oparciu o przedziały wartości parametrów, a w trakcie swojego istnienia powinien mieć możliwość uczenia się zachowań. Innym podejściem jest wykorzystanie uczenia maszynowego do translacji parametrów opisujących jakość transmisji w pewnej skali.

Drugi agent wykorzystywany w koncepcji P&R odpowiada za rekonfigurację. W momencie, gdy agent monitorujący stwierdzi, że parametry nadmiernie odbiegają od przyjętych wartości zgłosi potrzebę rekonfiguracji. Agent odpowiedzialny za rekonfigurację rozpocznie proces wyznaczania nowych ścieżek – w oparciu o jakość każdego z łączy w sieci oraz na podstawie informacji o klientach wyznaczona zostanie ścieżka pomiędzy każdym użytkownikiem. Ścieżka musi spełniać wymagania klienta – jakość nie może być gorsza i nie powinna być lepsza. Następnie po wyznaczeniu nowych ścieżek agent prześle nową konfigurację do urządzenia, które jemu podlega. Agent może nadzorować jedno urządzenie lub grupę urządzeń.

Agenty monitorujący i odpowiadający za rekonfigurację mogą stanowić odrębne byty lub zostać zrealizowane w formie jednego agenta posiadającego obie funkcjonalności. W ramach koncepcji P&R działania agenta odpowiedzialnego za monitoring i rekonfigurację podzielić można na kilka etapów. Na rysunku 3.2 zaprezentowano model agenta Ag wraz z wyszczególnieniem komunikacji pomiędzy agentami tego samego typu oraz ruterem (warstwą fizyczną P&R). Etapem pierwszym jest



RYSUNEK 3.2: Diagram sekwencji dla agenta Ag

wymiana informacji dotyczących sieci podłączonych bezpośrednio do rutera nadzorowanego przez agenta. Każdy agent Ag wysyła do najbliższych sąsiadów informację dotyczącą bezpośrednio podłączonych sieci wraz z wyszczególnieniem jakości łączy. Następnie agenty przesyłają między sobą informacje o sieciach, o których dowiedzieli się od sąsiadów. Wymiana komunikatów kończy się w momencie, gdy wszystkie agenty posiadają informację na temat topologii sieci. Agenty nadzorujące urządzenia do których podłączeni są klienci posiadają informację o jakości wymaganej przez poszczególnych klientów. Następnie określone są wszystkie możliwe ścieżki, a w oparciu o informację dotyczącą jakości wymaganej przez klienta wybierana jest ścieżka transmisji, która będzie wykorzystywana na potrzeby danego klienta. Po określeniu ścieżki następuje etap rozesłania zdefiniowanej konfiguracji do sąsiednich urządzeń, a urządzenia te roześlą ją dalej. W oparciu o otrzymane informacje agenty dokonają konfiguracji rutera lub ruterów, które nadzorują. W tym celu przesłane zostaną odpowiednie komendy konfiguracyjne. Od tego momentu sieć jest już w pełni skonfigurowana i gotowa do transmisji.

Co pewien okres czasu agent weryfikuje parametry łączy urządzenia, które nadzoruje. Test polega na przesłaniu strumienia informacji przez poszczególne łącza. Określane są parametry, takie jak przepływność, opóźnienie transmisji, fluktuacja opóźnienia i współczynnik utraty pakietów. Informacja o zmierzonych parametrach przesyłana jest do sąsiednich urządzeń. Agent nadzorujący urządzenie, do którego podłączeni są klienci, w oparciu o otrzymane informacje dokonuje oceny ścieżek wykorzystywanych do transmisji przez klientów. W momencie gdy agent stwierdzi, iż parametry ścieżki odbiegają od tych wymaganych przez klienta, podejmuje decyzję o rekonfiguracji sieci. Informacja o konieczności przeprowadzenia rekonfiguracji ścieżek wysyłana jest do sąsiadów, a sąsiedzi przesyłają informację dalej. Następnie agent dokonuje rekonfiguracji rutera, który nadzoruje. Aby proces rekonfiguracji nie powodował obniżenia jakości trwających transmisji należy zsynchronizować go poprzez ustalenie pewnego momentu, w którym wszystkie agenty dokonają rekonfiguracji. W przypadku gdy agent działa na zasadzie przeprowadzania rekonfiguracji zaraz po otrzymaniu stosownej informacji, jakość transmisji może się pogorszyć co wynika z niespójności tablic routingu prowadzących do utraty pakietów. W związku z tym istotne jest, aby rekonfiguracja przeprowadzana była w tym samym czasie dla wszystkich urządzeń w sieci. Monitorowanie łączy ponawiane jest co określony czas. Równocześnie podejmowana jest decyzja dotycząca rekonfiguracji.

Ostatni typ agenta odpowiada za handel jakością transmisji, przy czym agent tego typu może reprezentować operatora lub klienta. W najprostszym przypadku użytkownik płaci określoną kwotę za pewną jakość. Wysokość opłaty jest stała i taka sama dla każdego użytkownika. Opłata ta może być wnoszona co określony czas (np. jeden miesiąc). Nie jest to jedyne rozwiązanie, ponieważ zdarzyć może się tak, że użytkownik potrzebuje jakości wyłącznie na czas trwania określonej transmisji lub na potrzeby konkretnej usługi. Wówczas system umożliwia użytkownikowi tymczasową zmianę jakości transmisji. W najprostszym ujęciu system agentowy będzie działał na zasadzie pośrednika przy zakupie. Wysokość opłat nie zawsze musi być stała – możliwe jest wykorzystanie metod rynkowych do kształtowania cen.

Zakup jakości transmisji można uznać za złożony proces – stałe przełożenie ceny na jakość może być dobrym rozwiązaniem, lecz możliwe jest również dynamiczne kształtowanie cen przy użyciu metod rynkowych. Negocjacje to proces mający na celu osiągnięcie porozumienia w kwestiach dotyczących wymiany towarów i usług. Każdy z parametrów dotyczących jakości może stanowić odrębne dobro – wówczas negocjacje dotyczą wielu dóbr równocześnie, których kombinacja skutkuje satysfakcją klienta. Możliwa jest również translacja kombinacji parametrów jakościowych z wykorzystaniem pewnej skali. Translacja taka, możliwa jest do wykonania z wykorzystaniem uczenia maszynowego, takie rozwiązanie zaproponowano w

dalszej części pracy. Następnie jakość określona w danej skali podlega negocjacom. Wówczas oferowane jest wyłącznie jedno dobro. Negocjacje wielu dóbr są bardziej skomplikowane, ponieważ dotyczą kilku parametrów równocześnie, których kombinacja jest istotna. Stronami w negocjacjach mogą być sprzedający (operator) oraz kupujący. Taka sytuacja stanowi najprostszy przypadek negocjacji. Nie jest to jedyna możliwość, ponieważ o daną jakość może równocześnie zabiegać wielu klientów. Przewidzenie wyniku tego typu negocjacji jest trudne, ponieważ każdy klient może mieć różną strategię udziału w negocjacjach. Poza tym pojawiają się również kwestie związane ze zmową kupujących mającą na celu osiągnięcie wyniku jak najlepszego dla pewnej grupy klientów.

W kwestii alokacji dóbr skorzystać można z mechanizmu aukcji. Ten typ alokacji stosowany jest wtedy, gdy danym dobrem zainteresowany jest więcej niż jeden podmiot. W sytuacji, gdy dobro pożądane jest wyłącznie przez jeden podmiot, wykorzystanie aukcji nie jest uzasadnione – wystarczą negocjacje dwustronne. Celem aukcji jest alokacja dóbr na rzecz jednego z uczestników. Strategia każdego z uczestników jest jego indywidualną decyzją, tak jak i próg kwoty maksymalnej. Uczestnik nie powinien ujawniać swojej strategii. Aukcje przebiegają zgodnie z określonymi zasadami – protokołem aukcji. Istotne jest zdefiniowanie kto po zakończeniu aukcji otrzyma dobro. W przypadku stosowania koncepcji P&R można wykorzystać dowolny mechanizm prowadzenia aukcji, co jest wynikiem separacji warstw. Za przeprowadzenie aukcji odpowiada warstwa rynku w której działają agenci reprezentujące klientów i operatora (lub operatorów). Z warstwy tej pochodzi informacja, którą otrzymają agenci działające w warstwie kontrolnej, dotycząca jakości transmisji zakupionej przez klienta. Z kolei z warstwy kontrolnej pochodzi informacja o dostępnych jakościach – informacja ta zostanie wykorzystana przez warstwę rynku do przeprowadzenia aukcji. Jeśli możliwe jest zaoferowanie tej samej jakości więcej niż jednemu klientowi wówczas aukcja zostanie przeprowadzona tyle razy, ile jest ofert.

## 3.2 Projekt i realizacja emulatora

Na potrzeby projektu przygotowany został emulator, który składa się z oprogramowania realizującego funkcje warstwy kontrolnej oraz ruterów działających w warstwie fizycznej P&R. Do trasowania wykorzystany został programowy ruter Vyatta (aktualnie VyOS) [206] – wybór ten wynika z konieczności zastosowania dobrze zoptymalizowanego rutera programowego. W wykorzystanej sieci emulowano ruch łańcuchowy - w tym celu wykorzystano Trex [207] będący generatorem ruchu sieciowego. Trex umożliwia generowanie ruchu zbliżonego do tego, który występuje

| Ocena | Przepływność [Mb/s] |
|-------|---------------------|
| 5     | 100                 |
| 4     | 50                  |
| 3     | 10                  |
| 2     | 5                   |
| 1     | 1                   |
| -1    | łącze nieaktywne    |

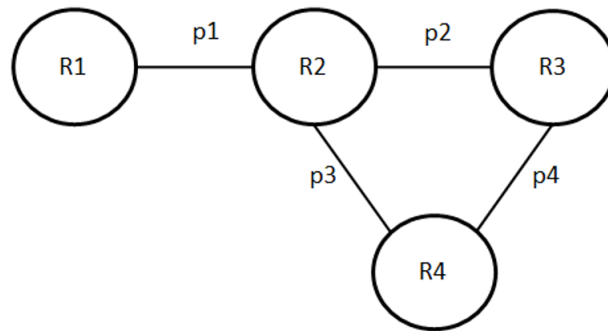
TABELA 3.1: Przykładowe cechy łącza dla poszczególnych poziomów usługi

w rzeczywistych sieciach komputerowych. Emulator został zrealizowany zgodnie z założeniami koncepcji P&R i w związku z tym pierwszą czynnością, którą wykonuje warstwa sterowania jest pobranie wstępnej konfiguracji ruterów. Konfiguracja dotyczy każdego interfejsu rutera, a pobranie konfiguracji odbywa się dla każdego rutera z osobna. Gdy oprogramowanie będzie posiadało konfigurację wszystkich ruterów, nastąpi analiza otrzymanych informacji, polegająca na wyszukaniu aktywnych połączeń pomiędzy poszczególnymi ruterami.

Założono, iż pomiędzy ruterami występują łącza punkt-punkt. Oprogramowanie przechowuje informację o wszystkich połączeniach pomiędzy ruterami. W kolejnym kroku ustalana jest przepływność poszczególnych łączy - aby tego dokonać poprzez każde pojedyncze łącze przesyłany jest plik o określonej wielkości, wypełniony losowymi danymi. Rozmiar pliku może być dowolny. Doświadczalnie stwierdzono, iż plik o wielkości 10 MB jest wystarczający. Przeprowadzono eksperyment dla kilku rozmiarów pliku, dla pliku o wielkości 1 MB wynik nie był miarodajny, z kolei dla pliku o wielkości 100 MB emulacja trwała zbyt długo, a uzyskane wyniki nie odbiegały od tych uzyskanych dla pliku o wielkości 10 MB. Dzięki transmisji danych testowych uzyskano informację dotyczącą czasu transmisji i średniej przepływności podawanej w b/s. Informacja ta zapisywana jest dla każdego łącza. Następnie w oparciu o otrzymane wyniki pomiarów wyznaczana jest ocena trasy. Na potrzeby emulacji zdefiniowana została skala ocen wynikająca ze zmierzonych parametrów. Skala ta może być dowolnie modyfikowana przez administratora. W tabeli 3.1 zaprezentowano najprostszą skalę jakości transmisji wyrażoną przy pomocy przydzielonej przepływności. Maksymalna przepływność 100 Mb/s wynika z ograniczeń oprogramowania wirtualizującego wykorzystywanego w emulacji.

Kolejnym krokiem jest wyznaczenie ścieżek. Ścieżki prowadzą z rutera źródłowego do docelowego, za pośrednictwem innych ruterów. W celu wyznaczenia wszystkich możliwych ścieżek wykorzystano następujący algorytm:

1. Weź pierwszą ścieżkę o długości 1 (punkt-punkt, łącząca tylko dwa routery).



RYSUNEK 3.3: Przykładowa sieć - charakterystyka algorytmu wyznaczania tras

2. Wyznacz wszystkie niepowtarzające się ścieżki o długości +1 (nie mogą powstawać pętle, łącze punkt-punkt może zostać wykorzystane tylko raz). Wykonuj krok 2 tak długo, aż nie będzie dłuższej ścieżki do wyboru – nie można postępować dalej, ponieważ wszystkie dostępne łącza punkt-punkt zostały wykorzystane.
3. Powtórz krok od 1-2 dla wszystkich łączy punkt-punkt.

Na rysunku 3.3 przedstawiono przykładową sieć wykorzystywaną do emulacji. Niech  $R$  będzie zbiorem ruterów,  $R = \{R1, R2, R3, R4\}$ , niech  $P$  będzie zbiorem połączeń między ruterami,  $P = \{p1, p2, p3, p4\}$  gdzie  $p1 = \{R1, R2\}$ ,  $p2 = \{R2, R3\}$ ,  $p3 = \{R2, R4\}$ ,  $p4 = \{R3, R4\}$ . W kolejnych etapach działania algorytmu tablica ścieżek ma następującą postać:

1.  $PR1 = \{\{p1\}\}$
2.  $PR1 = \{\{p1\}, \{p1, p2\}, \{p1, p3\}\}$
3.  $PR1 = \{\{p1\}, \{p1, p2\}, \{p1, p3\}, \{p1, p2, p4\}, \{p1, p3, p4\}\}$
4.  $PR1 = \{\{p1\}, \{p1, p2\}, \{p1, p3\}, \{p1, p2, p4\}, \{p1, p3, p4\}, \{p1, p2, p4, p3\}, \{p1, p3, p4, p2\}\}$

$PR1$  – zbiór wszystkich ścieżek z rutera  $R1$ .

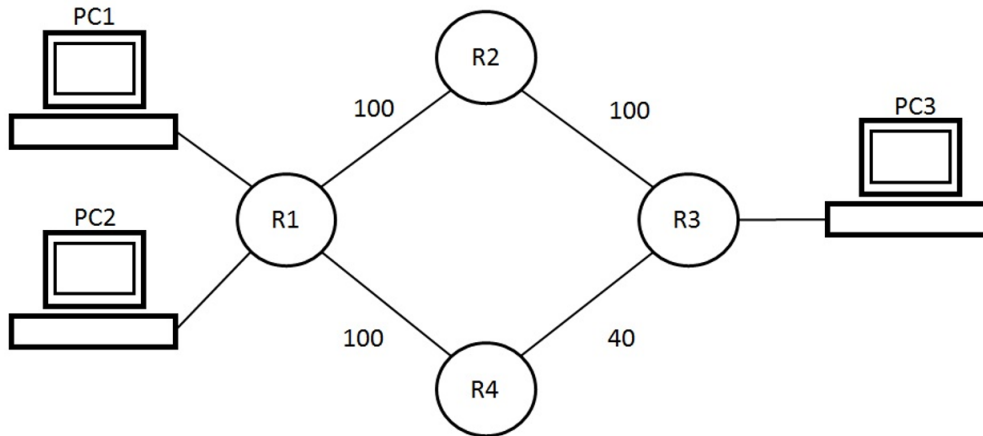
Powyższy algorytm został wykorzystany w emulatorze. Nie jest to jedyna metoda wyznaczania wszystkich możliwych ścieżek. Druga metoda to wykorzystanie specjalnego pakietu, który przesyłany jest między ruterami. Dla tej metody algorytm wygląda następująco:

1. Utwórz specjalny pakiet testujący ścieżki (zawierający informację o źródle transmisji, sąsiedzie, do którego pakiet ma trafić i celu, czyli kliencie docelowym), prześlij go do sąsiada.

2. Gdy odebrany zostanie pakiet testowy przeprowadzona zostanie weryfikacja, czy odbierający jest celem transmisji, a jeśli tak to przejdź do kroku 4. Jeśli nie to w oparciu o tablicę sąsiadów sprawdź, czy któryś z sąsiadów nie został już odwiedzony, a następnie powiel pakiet tyle razy do ilu sąsiadów go prześlesz, dodaj do każdego pakietu adres sąsiada. Prześlij pakiet dalej.
3. Krok 2 powtarzany jest na każdym kolejnym urządzeniu.
4. Sprawdź, czy posiadasz daną ścieżkę źródło-cel w swojej tablicy, jeśli nie to dopisz ją. Następnie zaznacz w pakiecie, że dotarł do celu i przekaz go do sąsiada zgodnie z informacją dotyczącą ścieżki zapisaną w pakiecie.
5. Gdy odebrany zostanie pakiet testujący z zaznaczoną opcją że dotarł do celu, zweryfikuj czy jesteś jego źródłem, a jeśli nie to przekaz go dalej zgodnie z informacją dotyczącą ścieżki zapisaną w pakiecie. Jeśli tak to przejdź do punktu 7.
6. Krok 5 powtarzany jest na każdym kolejnym urządzeniu.
7. Zapisz informację dotyczącą ścieżki pomiędzy klientem źródłowym, a docelowym w tablicy ścieżek.

Powyższe kroki powtarzane są dla każdego klienta/grupy klientów tak, aby każdy agent nadzorujący ruter do którego podłączeni są klienci posiadał informację o wszystkich możliwych ścieżkach pomiędzy klientami. Wybór konkretnego algorytmu zależy od konfiguracji środowiska – jeśli agent działający w warstwie kontrolnej posiada pełną wiedzę na temat całej topologii sieci, wówczas możliwe jest wykorzystanie pierwszego algorytmu. Z kolei, gdy agent posiada tylko częściową informację ograniczoną do najbliższych sąsiadów, algorytm pierwszy nie może być stosowany, należy zastosować algorytm drugi. W przypadku pełnej topologii konieczna jest wymiana większej liczby komunikatów tak, aby każdy agent posiadał spójny zestaw informacji.

Do określenia oceny ścieżki wykorzystano podejście, w ramach którego całkowitą ocenę ścieżki stanowi ocena łącza o najgorszych parametrach należącego do danej ścieżki. Takie podejście wydaje się być korzystniejsze, ponieważ łącze o najniższej przepływności zredukuje szybkość transmisji na całej trasie źródło-cel. Oceny ścieżek zapisywane są wraz z informacją o liczbie ruterów przez które musi przejść pakiet, aby dotrzeć do celu transmisji w danej ścieżce. Informacja ta posłuży w późniejszym czasie do podjęcia decyzji o wyborze ścieżki w przypadku, gdy kilka ścieżek ma taką samą ocenę.



RYSUNEK 3.4: Przykładowa sieć - porównanie klasycznego routingu z P&R

Następnie wybrana zostaje ścieżka, którą przesyłane będą dane klienta. W aplikacji przechowywana jest informacja o identyfikatorze klienta oraz oczekiwanym poziomie usługi (zgodnym z przyjętą skalą ocen). W pierwszej kolejności następuje weryfikacja do których routerów podłączeni są poszczególni klienci (konfiguracja sieci polega na umożliwieniu transmisji pomiędzy poszczególnymi klientami), wyznaczone są wszystkie możliwe ścieżki od jednego klienta do wszystkich pozostałych klientów. O wyborze danej trasy decyduje jej ocena. Jeśli jest zgodna z oczekiwaniami klienta lub wyższa (gdy nie ma oczekiwanej) to wówczas trasa zostaje wybrana. Jeśli istnieje więcej niż jedna trasa o tej samej ocenie i liczbie skoków to wybrana zostanie pierwsza z listy. Gdy ustalone zostaną wszystkie ścieżki dla wszystkich klientów nastąpi konfiguracja warstwy fizycznej P&R (routerów).

W sieciach komputerowych trasowanie odbywa się na podstawie informacji niesionej w nagłówku pakietu, tj. celu transmisji. W przypadku klasycznego routingu urządzenie porównuje adres sieci do której ma trafić pakiet, z adresami zawartymi we własnej tablicy routingu. W oparciu o tę informację przekierowuje pakiet do następnego rutera lub urządzenia docelowego. Na rysunku 3.4 przedstawiono schemat sieci komputerowej, która posłuży do opisanego działania. W przypadku prezentowanej sieci pomiędzy komputerami PC1 i PC2, a komputerem PC3 istnieją dwie ścieżki transmisji. Pierwsza ścieżka prowadzi przez R1-R2-R3, a druga przez R1-R4-R3. Poszczególnym łączom przypisano metryki oraz założono, że im metryka ma niższą wartość tym dane łącze jest lepsze. W związku z tym zgodnie z zasadami routingu klasycznego transmisja pomiędzy PC1-PC3 i PC2-PC3 będzie przebiegała dokładnie tą samą ścieżką (R1-R4-R3). W sieciach komputerowych, poza klasycznym podejściem do routingu, zdefiniowano również pojęcie PBR. Jest to technika, która pozwala administratorowi definiować złożone reguły routingu. Decyzja odnośnie skierowania pakietu do urządzenia następnego skoku może być podejmowana nie



tylko na podstawie adresu docelowego, lecz również na podstawie innych informacji znajdujących się w nagłówku pakietu, czyli np. źródle transmisji, numerze portu. PBR pozwala definiować wiele tablic routingu. Dla każdego użytkownika lub grupy użytkowników administrator może zdefiniować, którą ścieżką będzie odbywała się transmisja. Oczywiście istotną kwestią jest to, że powinna istnieć więcej niż jedna ścieżka pomiędzy źródłem i celem – tylko wtedy stosowanie PBR jest uzasadnione. Analizując przypadek przedstawiony na rysunku 3.4 administrator może tak skonfigurować rutery, że pakiety pochodzące z komputera PC1 przesyłane do PC3 będą podróżowały inną trasą (np. R1-R2-R3), niż pakiety pochodzące z PC2 przesyłane do PC3 (np. R1-R4-R3). W związku z tym PBR daje możliwość różnicowania ścieżek transmisji w zależności od źródła. W emulatorze do routingu wykorzystano technikę PBR. W momencie, gdy agenty posiadają już wszystkie niezbędne informacje i ścieżki zostały wyznaczone następuje konfiguracja warstwy fizycznej P&R (tablic routingu). Po przeprowadzeniu konfiguracji sieć zaczyna funkcjonować zgodnie z oczekiwaniami klientów. Co określony czas (parametr konfigurowalny) przeprowadzana jest weryfikacja aktualnych parametrów jakościowych poszczególnych ścieżek. Jeśli odbiegają od przyjętych przedziałów, to wówczas nastąpi proces rekonfiguracji sieci (wyznaczenie i wybór nowych ścieżek transmisji).

### 3.3 Symulacja z wykorzystaniem NS3

Kolejnym etapem badań było przeprowadzenie symulacji. Zdecydowano się na wykorzystanie NS3 [208] - symulatora sieci komputerowych i zdarzeń dyskretnych w sieciach. Rozwiązanie przygotowane w NS3 to kompletna implementacja koncepcji P&R, tożsama z rozwiązaniem wykorzystanym w procesie emulacji. Poza konfiguracją sieci (topologia, protokoły) na potrzeby symulacji zaimplementowano również agentów działających w warstwie sterowania. Agenty są reprezentowane poprzez odpowiednie klasy i metody. Za inicjalizację agentów i ich działanie odpowiada symulator. Etapy działania agentów można podzielić w następujący sposób:

1. Inicjalizacja (wykonywana tylko raz w trakcie trwania całej symulacji).  
Dla każdego agenta nadzorującego ruter wywoływana jest metoda *init\_tables*, której zadaniem jest zainicjowanie tablic wykorzystywanych w czasie działania agenta. Tablice wypełniane są wartością -1. Następnie dla każdego agenta, który nadzoruje ruter będący bramą domyślną wywoływana jest metoda *add\_Customer*, której zadaniem jest dodanie informacji o kliencie do tablicy klientów agenta. Informacja zawiera id klienta, wymaganą jakość oraz id interfejsu rutera, do którego podłączony jest klient. W kolejnym kroku wywoływana jest metoda *set\_customer\_comm*, której zadaniem jest umieszczenie w

tablicy agentów (rutery-bramy) identyfikatorów klientów, którzy mogą się ze sobą komunikować. Jest to celowe działanie, ponieważ nie w każdym przypadku komunikacja jest pożądana (scenariusze symulacji). Następnie dla każdego agenta wywoływana jest metoda *add\_config*, której zadaniem jest umieszczenie w odpowiedniej tablicy informacji dotyczących konfiguracji poszczególnych interfejsów rutera, takich jak id sieci, jakość (na tym etapie zawsze -1), id interfejsu, id sąsiada (rutera, do którego można dotrzeć za pośrednictwem tego interfejsu/sieci). Ostatni krok to wywołanie metody *change\_Quality* której zadaniem jest skonfigurowanie wstępnej jakości wszystkich łączy danego rutera, który nadzoruje agent. Przekazywane parametry to przepływność i opóźnienie transmisji.

2. Rozgłoszenie klientów (wykonywane raz chyba, że pojawi się nowy klient lub któryś z dotychczas skonfigurowany przestanie funkcjonować).

Zadaniem tego etapu jest poinformowanie agentów nadzorujących rutery będące bramami domyślnymi o wszystkich klientach w danej sieci. Pozostałe rutery nie posiadają takiej informacji. Dla każdego agenta, który nadzoruje ruter będący bramą domyślną, wywoływana jest metoda *advertise\_customers*. Każdy agent przekazuje informację dalej w oparciu o tablicę sąsiadów. Jeśli agent, który odebrał tego typu informację nadzoruje ruter-bramę to doda informację o klientach do odpowiedniej tablicy. Następnie informacja przekazywana jest dalej do kolejnych sąsiadów o ile nie dotarła do nich wcześniej. Identyfikator ruterów, które już otrzymały daną informację jest zapisywany w specjalnym nagłówku.

3. Ustalenie ścieżek.

Zadaniem tego etapu jest wyznaczenie wszystkich możliwych ścieżek pomiędzy źródłem, a dopuszczalnymi celami oraz ocena jakości ścieżek. Dla każdego agenta nadzorującego ruter-bramę wywoływana jest metoda *prepare\_routing*. W ramach niej tworzony jest specjalny pakiet zawierający informację o ruterze źródłowym i ruterze docelowym. Tego typu pakiet wysyłany jest do wszystkich sąsiadów. Agent, który odbierze tego typu pakiet sprawdza czy jest celem transmisji:

- Jeśli nie jest celem transmisji to przekaże informację do swoich sąsiadów, ale tylko tych, którzy nie zostali jeszcze odwiedzeni. W pakiecie zapisywana jest informacja o agentach już odwiedzonych. Czynność ta wykonywana jest w pętli tak długo, aż osiągnięty zostanie cel.

- Jeśli jest celem to wywoływana jest metoda *send\_info\_back*, a pakiet przekazywany jest dokładnie tą samą ścieżką, którą trafił do celu. W trakcie przekazywania uzupełniana jest informacja o jakości poszczególnych łączy punkt-punkt (pomiędzy ruterami). Gdy pakiet trafi do źródła, to uzupełniana jest tablica możliwych ścieżek (zawierająca wszystkie ścieżki źródło-cel). Jakość ścieżki określana jest zgodnie z zasadą „bottleneck” – sumaryczną oceną ścieżki to najniższa z jakości łączy.

Kolejnym krokiem jest wybranie ścieżki dla klienta spośród wszystkich dostępnych. W tym celu agent weryfikuje potrzeby klienta oraz możliwe ścieżki. Wybierana jest ścieżka zgodna z oczekiwaniami klienta, a jeśli taka ścieżka nie jest dostępna, to wybierana jest najgorsza z lepszych.

#### 4. Rozgłoszenie informacji o rutingu.

Dla każdego agenta nadzorującego ruter-bramę wywoływana jest metoda *send\_routing\_to\_neighbour* której zadaniem jest przekazanie informacji o wybranej ścieżce do każdego agenta, który jest jej uczestnikiem. Jeśli agent otrzymujący tego typu informację nie jest celem, to przekazuje informację do kolejnego sąsiada (zgodnie z wybraną ścieżką).

#### 5. Skonfigurowanie rutingu.

Dla każdego agenta wywoływana jest metoda *set\_router\_routing\_tab*. W metodzie tej usuwane są wszystkie wpisy z tablicy rutingu rutera nadzorowanego przez danego agenta. Następnie dodawane są reguły zgodne z tablicą rutingu agenta.

#### 6. Rekonfiguracja sieci.

Agenty monitorują parametry poszczególnych łączy. Gdy stwierdzą, iż parametry nie są zgodne z wymaganiami klienta to zainicjują proces rekonfiguracji sieci. Informacja o tym fakcie przekazywana jest do nadzorującego agenta (w rzeczywistym systemie agenty same testują łącza i stwierdzają, czy parametry uległy zmianie, w symulatorze nie ma takiej możliwości). Najpierw wywoływana jest metoda *change\_Quality* dla każdego agenta, którego dotyczy zmiana (pogorszenie jednego łącza powoduje zmianę u dwóch agentów). Informacja o zmianie przekazywana jest do rutera będącego źródłem (ruter-brama) zgodnie ze skonfigurowaną ścieżką (pośrednictwo innych agentów, ale tylko tych będących częścią ścieżki). Gdy agent nadzorujący ruter-bramę odbierze tego typu informację weryfikuje czy konieczna jest rekonfiguracja (czy parametry są nadal zgodne z wymaganiami klienta). Proces rekonfiguracji wymaga wykonania metod opisanych w punkcie 3, 4 i 5.

Przeprowadzenie symulacji wymagało również odpowiedniego skonfigurowania sieci. NS3 daje możliwość swobodnego budowania sieci. Na konfigurację składają się etapy, takie jak utworzenie węzłów (zarówno reprezentujących routery, jak i klientów), utworzenie tzw. kontenerów, które można uznać za reprezentację łącza oraz np. grup routerów czy użytkowników (jeśli mają mieć takie same parametry). W przypadku sieci, w ramach przyjętych założeń, pomiędzy routerami występują łącza punkt-punkt i w związku z tym, tak też skonfigurowano kontenery. Kolejny element to konfiguracja parametrów łącza – przepływność i opóźnienie transmisji, a następnie uruchamianie jest stos protokołów sieciowych na wszystkich urządzeniach. Ostatni etap to utworzenie poszczególnych podsieci, zaadresowanie urządzeń i skonfigurowanie tras domyślnych na klientach, czyli wpisu w tablicy routingu, który mówi o tym, gdzie klient ma kierować pakiety w momencie, gdy cel transmisji jest poza jego podsiecią.

### 3.4 Zastosowanie agentów

Zgodnie z przyjętą koncepcją w stworzonych rozwiązaniach wykorzystano agenty – monitorujące, rekonfiguracji tras, handlujące. W przypadku agenta-handlarza przyjęto kilka strategii działania:

1. Agent posiada informację, o jakości poszukiwanej przez klienta oraz maksymalnej kwocie, którą klient może zapłacić. Oferta (Of) składana jest wyłącznie do określonego maksimum. Na nierówności 3.1 zaprezentowano sposób składania oferty dla tej strategii, przy czym  $q$  oznacza jakość, a  $\max(c_1(q))$  to maksymalna kwota, jaką klient ( $c_1$ ) jest w stanie przeznaczyć na zakup jakości  $q$ .

$$Of(q) \leq \max(c_1(q)) \quad (3.1)$$

2. Agent posiada informację o maksymalnej kwocie, oczekiwanej jakości oraz o możliwości odstępstwa od jakości w określonym przedziale. Sytuacja taka może być korzystna dla klienta, któremu zależy na transmisji za optymalną cenę przy zachowaniu akceptowalnej jakości. Metoda złożenia oferty zaprezentowana została na nierówności 3.2.  $Q_m$  to margines odstępstwa od założonej jakości akceptowanej przez klienta.

$$Of(q \pm Q_m) \leq \max(c_1(q + Q_m)) \quad (3.2)$$

3. W sytuacji, gdy ta sama jakość oferowana jest po kolei na kilku różnych aukcjach (operator jest w stanie zapewnić określony poziom kilku klientom równocześnie), a agent posiada informację o liczbie takich aukcji, możliwe jest przyjęcie strategii, w której użytkownik obserwuje wyniki kolejnych gier. W ostatniej rundzie agent bierze udział w grze przewidując, jaki będzie jej wynik w oparciu o wcześniejsze gry. Informacja, którą otrzymuje agent może być częściowa (wyłącznie końcowy wynik aukcji) lub pełna (wynik końcowy oraz oferty złożone przez każdego agenta). W przypadku wiedzy częściowej, agent jest w stanie dokonać predykcji wyłącznie na podstawie wyników dotychczasowych gier. W związku z tym może założyć, iż w przypadku złożenia pewnej oferty wygra aukcję. Bardziej złożonym modelem jest ten, w którym agent posiada pełną informację na temat przebiegu wszystkich gier oraz ofert złożonych przez pozostałych uczestników. Dzięki takiemu podejściu agent może uczyć się zachowań innych graczy i poznawać ich strategię. W oparciu o to, dla każdego konkurenta może zostać stworzony model zachowań. Model ten zostanie wykorzystany na etapie składania oferty takiej, że agent z pewnym prawdopodobieństwem wygra aukcję płacąc przy tym optymalną kwotę (nie przepłacając). Agent posiadając wiedzę na temat konkurentów może porównywać ich modele, a następnie przyjmie taką strategię, która pozwoli mu zwyciężyć. To czy agent może posiadać pełną informację dotyczącą wcześniejszych aukcji czy tylko częściową, uzależnione jest od konfiguracji systemu i przyjętego protokołu prowadzenia aukcji.

### **3.5 Klasyfikacja jakości transmisji z wykorzystaniem uczenia maszynowego**

W P&R trudnym zadaniem okazało się przekształcanie parametrów opisujących jakość transmisji do pewnej przyjętej skali ocen. Parametry opisujące jakość to opóźnienie transmisji, przepływność, fluktuacja opóźnienia oraz współczynnik utraty pakietów. Te parametry są kluczowe, jak chodzi o wiarygodną determinację jakości transmisji. Tego typu parametry mogą być niezrozumiałe dla klienta oraz mogą być trudne do wykorzystania w systemie. W związku z tym podjęto próby statycznej translacji parametrów do pewnej skali ocen (1-5). Statyczne tablice okazały się trudne w zdefiniowaniu. Zdecydowano się na wykorzystanie uczenia maszynowego (z ang. Machine Learning, ML). Przeprowadzono liczne badania pod kątem wykorzystania różnych klasyfikatorów oraz różnych algorytmów klasyfikacji.

Pierwszą istotną kwestią było pozyskanie próbek wykorzystanych w procesie szkolenia klasyfikatorów. Zdecydowano się na pozyskanie rzeczywistych próbek z wykorzystaniem QoE. Próbki oparto na empirycznych doznaniach użytkowników testowych. Każda próbka to inny przypadek kombinacji parametrów opisujących jakość transmisji. Na potrzeby ewaluacji stworzono 100 różnych próbek. Próbki uzyskano dzięki wykorzystaniu oprogramowania Trex, które umożliwiło emulowanie ruchu tła w sieci. Następnie mierzono parametry opisujące jakość transmisji. Jednakże same parametry nie nadawały się do ewaluacji przez użytkowników testowych. W związku z tym przygotowano dwa przypadki tj. (1) strumieniowanie filmu, (2) wczytanie strony internetowej - są to dwie sytuacje, z którymi użytkownicy bardzo często się spotykają i w związku z tym, na podstawie nich są w stanie ocenić jakość. Te dwa przypadki zaprezentowano użytkownikom w celu oceny jakości transmisji. W przypadku QoE wyróżnić można dwa podejścia:

- Zaprezentowanie każdej próbki osobno i poproszenie użytkownika o ocenę pojedynczej próbki w skali 1-5, gdzie 1 oznacza najgorszą jakość, a 5 to jakość najlepsza,
- Zaprezentowanie próbki referencyjnej, a następnie poproszenie użytkownika o ocenę pozostałych próbek porównując je z próbką referencyjną. W takim przypadku możliwe jest zastosowanie skali -2 do +2, gdzie -2 oznacza znaczne pogorszenie jakości w stosunku do próbki referencyjnej, a +2 oznacza dużo lepszą jakość.

Zdecydowano się na wykorzystanie podejścia pierwszego, czyli niezależnej oceny próbek. Jedna próbka to oba przypadki - (1) oraz (2). Przygotowano 100 próbek, każda została niezależnie oceniona przez użytkowników testowych. Szczegóły dotyczące zbioru danych przedstawiono w tabeli 3.2. O ocenę poproszono czterech użytkowników. W ten sposób dla jednej próbki uzyskano cztery oceny. Aby zapewnić dane wykorzystane w procesie uczenia maszynowego, zweryfikowano czy w ocenie próbek nie występują znaczne różnice. Nie stwierdzono tego typu problemu i w związku z tym zdecydowano, że ostateczną oceną próbki będzie zaokrąglona średnia z uzyskanych ocen.

| Liczba użytkowników testowych: 4 |               |
|----------------------------------|---------------|
| Liczba próbek: 100               |               |
| Parametry                        |               |
| Wejściowe                        | Wyjściowe     |
| Przepustowość [Mb/s]             | Jakość [1-5]  |
| Opóźnienie transmisji [ms]       |               |
| Fluktuacja opóźnienia [ms]       |               |
| Współczynnik utraty pakietów [%] |               |
| Klasy                            |               |
| Klasa (jakość)                   | Liczba próbek |
| 1                                | 15            |
| 2                                | 20            |
| 3                                | 26            |
| 4                                | 20            |
| 5                                | 19            |
| Razem                            | 100           |

TABELA 3.2: Szczegóły zbioru danych





## Rozdział 4

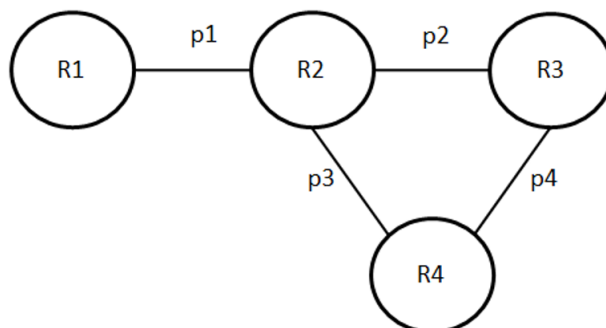
# Eksperymenty

W rozdziale tym przedstawiono eksperymenty wykonane pod kątem weryfikacji stosowalności prezentowanej koncepcji. Eksperymenty przeprowadzono dla sieci komputerowych LAN o różnej liczbie węzłów. W niniejszym rozdziale zaprezentowano uzyskane wyniki oraz wnioski z przeprowadzonych eksperymentów.

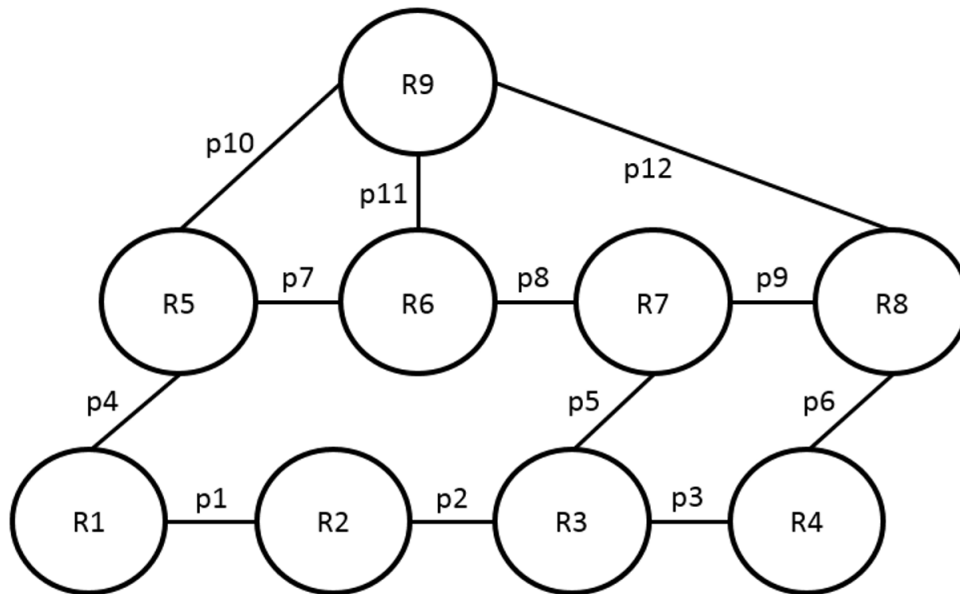
### 4.1 Emulacja

Emulację przeprowadzono dla sieci składającej się z 4 ruterów (rysunek 4.1) oraz sieci składającej się z 9 ruterów (rysunek 4.2). Pierwsza sieć posłużyła do oceny algorytmu i stworzonego rozwiązania. Druga z kolei została wykorzystana do zweryfikowania skalowalności rozwiązania.

Sieć zaprezentowana na rysunku 4.1. składa się z 4 ruterów oraz 4 użytkowników. Użytkownicy zostali podłączeni do ruterów R1 i R3, przy czym dwóch użytkowników potrzebuje wysokiej jakości transmisji, a pozostali akceptują niską jakość transmisji. Do rutera R1 i R3 podłączono po jednym użytkowniku każdego typu. Celem emulacji tej sieci było wykazanie zmian w wyborze ścieżek w zależności od parametrów poszczególnych łączy. Dzięki temu, że w sieci występują tylko 4 łącza



RYSUNEK 4.1: Schemat sieci wykorzystanej w badaniach (4 rutery)

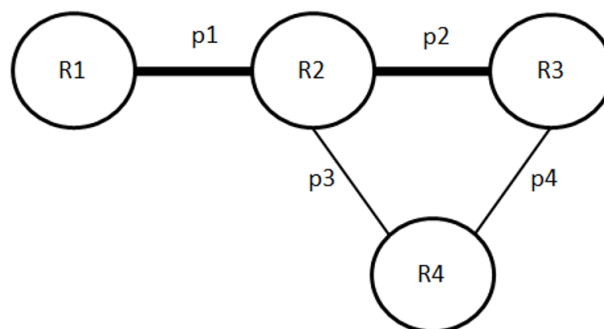


RYSUNEK 4.2: Schemat sieci wykorzystanej w badaniach (9 ruterów)

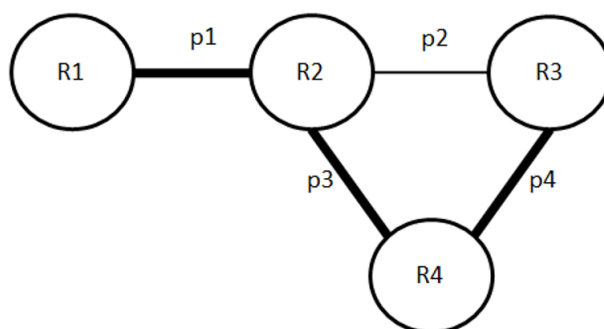
można wiarygodnie stwierdzić, jakie są możliwości wyboru ścieżek. W sieci emulowano również ruch tła odzwierciedlający sytuację, która występuje w rzeczywistych sieciach. Sieć ta ze względu na rozmiar nie pozwalała na uzyskanie całkowitej separacji ścieżek transmisji – wąskie gardło stanowi łączy p1, nie ma łączy alternatywnego. W emulacji wzięto pod uwagę ścieżki wyznaczone między użytkownikami podłączonymi do rutera R1 i R3. Emulację rozpoczęto od sprawdzenia ścieżek wybranych w momencie, gdy w sieci nie występuje żadna usterka ani nadmierne obciążenie.

Na rysunku 4.3 przedstawiono sieć z zaznaczoną ścieżką (pogrubiona linia) wybraną przez oprogramowanie dla przypadku, w którym sieć działa z maksymalnymi parametrami. W tym przypadku dla obu klientów wybrana została ta sama ścieżka co wynika z faktu, iż w sieci nie występuje ścieżka o niskiej jakości. W związku z tym klient akceptujący niską jakość, otrzymał usługę na wyższym poziomie niż oczekiwany. Wybrana trasa (R1-p1-R2-p2-R3) nie jest jedyną możliwą. Z punktu widzenia oceny trasy możliwy byłby również wybór ścieżki R1-p1-R2-p3-R4-p4-R3. Jednakże w przypadku, gdy ścieżki uzyskają taką samą ocenę wybór konkretnej będzie uzależniony od liczby urządzeń pośredniczących w transmisji (ruterów).

W kolejnym etapie emulacji założono pogorszenie się parametrów łączy p1 (R1-R2). Wybrana została ścieżka taka, jak dla poprzedniego przypadku (rysunek 4.3) co wynika z faktu, iż nie ma łączy alternatywnego dla p1. W sytuacji tej użytkownik akceptujący niską jakość otrzymał usługę na oczekiwanym poziomie, natomiast w sieci nie ma możliwości zagwarantowania usługi na poziomie wysokim. Wiąże



RYSUNEK 4.3: Wybrana ścieżka dla sieci działającej z maksymalnymi parametrami

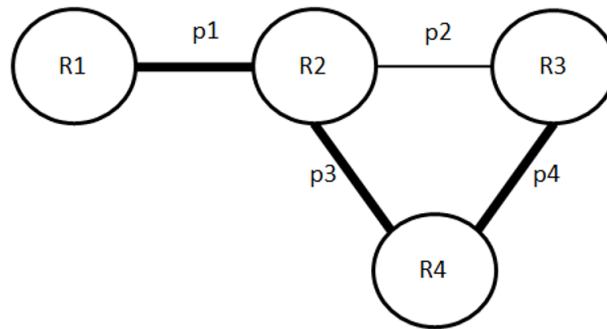


RYSUNEK 4.4: Wybrana ścieżka dla klienta akceptującego niską jakość dla sieci, w której pogorszeniu uległy parametry łącza p3

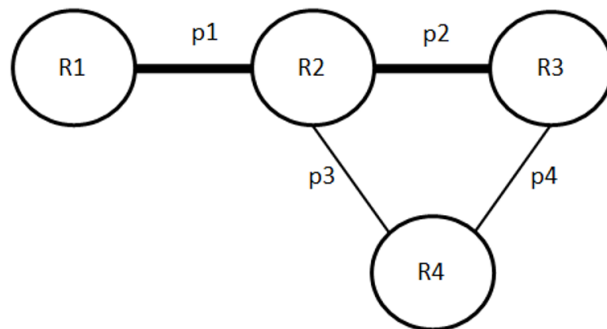
się to z koniecznością poinformowania użytkownika o tym fakcie i zwrotu poniesionych przez niego kosztów. Na podstawie tego przykładu zauważyć można, jak bardzo istotne jest występowanie tras alternatywnych – bez tego koncepcja P&R może być stosowana, lecz nie można uzyskać oczekiwanych efektów.

Następnie wykonano emulację dla sieci, w której obniżeniu uległa wyłącznie jakość transmisji łącza p3 (R2-R4). Na rysunku 4.4 zaprezentowano trasę wybraną dla użytkownika akceptującego niską jakość transmisji. Zauważyć można, iż w stosunku do poprzedniego przypadku nastąpiła zmiana trasy. Zmiana ta wynika z faktu, iż poprzez pogorszenie parametrów łącza p3, jakość ścieżki R1-p1-R2-p3-R4-p4-R3 jest zgodna z oczekiwaniami klienta akceptującego niską jakość transmisji. Z kolei dla użytkownika oczekującego wysokiej jakości transmisji, ścieżka nie uległa zmianie i nadal transmisja odbywa się ścieżką R1-p1-R2-p2-R3 (rysunek 4.3). Wynika to z faktu, iż parametry tej ścieżki nie uległy zmianie i są zgodne z oczekiwaniami klienta.

Ostatnia emulacja dotyczyła przypadku, w którym pogorszeniu uległa jakość łącza p2 (R2-R3). Dla użytkownika oczekującego wysokiej jakości wybrana została ścieżka R1-p1-R2-p3-R4-p4-R3. (rysunek 4.5). Wybór ten wynika z faktu, że mimo



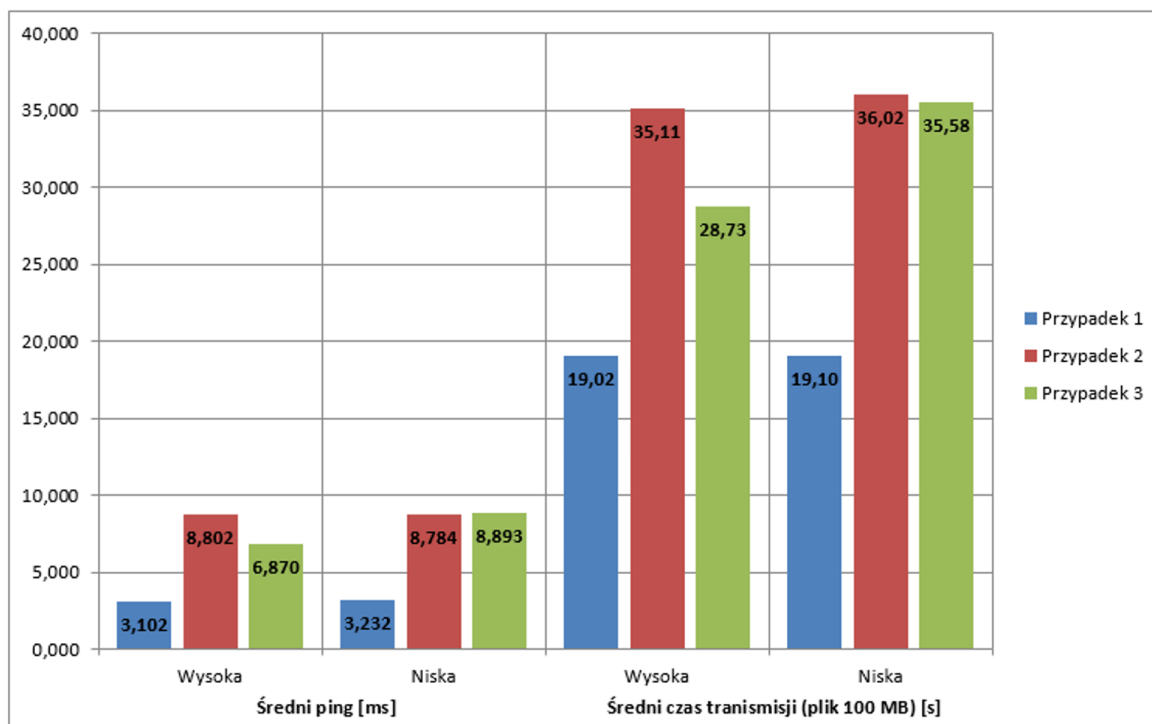
RYSUNEK 4.5: Ścieżka wybrana dla klienta wymagającego wysokiej jakości dla sieci, w której pogorszeniu uległy parametry łącza p2



RYSUNEK 4.6: Ścieżka wybrana dla klienta wymagającego niskiej jakości dla sieci, w której pogorszeniu uległy parametry łącza p2

tego, iż ścieżka ta jest dłuższa od alternatywnej to jej parametry jakościowe są zgodne z wymaganiami klienta. Z kolei dla klienta akceptującego niską jakość wybrana została ścieżka R1-p1-R2-p2-R3 (rysunek 4.6). Wybór ten podyktowany jest tym, że w momencie, gdy pogorszeniu uległy parametry łącza p2 ścieżka R1-p1-R2-p2-R3 może zostać wykorzystana na potrzeby transmisji klienta akceptującego niską jakość.

W ramach emulacji, poza weryfikacją ścieżek transmisji, dokonano również pomiarów opóźnień transmisji i średnich czasów transmisji. Do określenia opóźnień transmisji jako narzędzie pomiarowe, wykorzystano program Ping. Z kolei na potrzeby weryfikacji czasu transmisji wykorzystano plik o ustalonej wielkości 100 MB wypełniony losowymi danymi. Ping wykonywany był 1000 razy dla każdego przypadku, z kolei transmisja pliku wykonywana była 30 razy. Uśrednione wyniki przedstawione zostały w tabeli 4.1 oraz na rysunku 4.7. W celu stwierdzenia referencyjnych czasów dokonano pomiarów w momencie, gdy sieć działała z maksymalnymi parametrami i wybrana została ta sama ścieżka dla obu klientów (przypadek 1 na rysunku 4.7). Wynik w zależności od wymaganej jakości jest zbliżony, a zaobserwowane różnice są nieznaczne.



RYSUNEK 4.7: Wykres ping i czasów transmisji dla rutingu klasycznego i P&R

| Metodologia   | Jakość | Średni ping [ms] | Przedział ufności | Średni czas transmisji (plik 100MB) [s] | Przedział ufności |
|---|--------|------------------|-------------------|---|-------------------|
| Brak obciążenia, sieć działa z maksymalnymi parametrami, ta sama ścieżka dla obu jakości (przypadek 1)                      | Wysoka | 3,102            | 4,64% (0,144)     | 19,02                                   | 0,90% (0,173)     |
|   | Niska  | 3,232            | 4,49% (0,145)     | 19,10                                   | 0,92% (0,176)     |
| Sieć znacząco obciążona, duże natężenie ruchu, ta sama ścieżka dla obu jakości (przypadek 2)                                | Wysoka | 8,802            | 3,29% (0,290)     | 35,11                                   | 2,49% (0,876)     |
|   | Niska  | 8,784            | 3,30% (0,290)     | 36,02                                   | 2,60% (0,936)     |
| Sieć znacząco obciążona, duże natężenie ruchu, wykorzystano Pay&Require - ścieżka zależy od wymaganej jakości (przypadek 3) | Wysoka | 6,870            | 4,25% (0,292)     | 28,73                                   | 1,22% (0,350)     |
|   | Niska  | 8,893            | 3,58% (0,318)     | 35,58                                   | 2,51% (0,893)     |

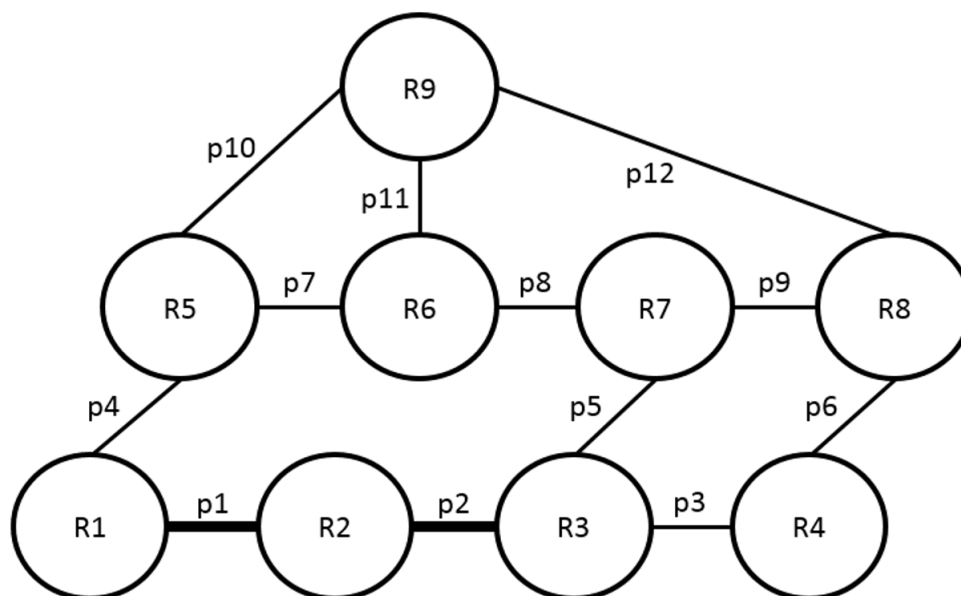
TABELA 4.1: Czasy transmisji dla rutingu klasycznego i P&R wraz z estymacją dla poziomu ufności  $\beta=95\%$

W kolejnym kroku należało stwierdzić, jak zmieniają się czasy transmisji w przypadku znaczącego obciążenia sieci (przypadek 2 na rysunku 4.7). Sieć obciążono poprzez zainicjowanie wielu równoczesnych transmisji dużych jednostek danych pomiędzy urządzeniami podłączonymi do R1 i R3. Równocześnie dokonano pomiarów czasów transmisji w przypadku klientów wymagających wysokiej i niskiej jakości. Zaobserwować można duże spowolnienie transmisji - w przypadku ping opóźnienie wzrosło prawie trzykrotnie. Z kolei średni czas transmisji pliku o wielkości 100 MB wypełnionego losowymi danymi wzrósł prawie dwukrotnie. Dzięki powyższym dwóm pomiarom uzyskano punkt odniesienia dla ostatniego etapu badań.

W ostatnim etapie dokonano pomiaru czasów dla sieci obciążonej tak samo, jak w poprzednim etapie z tym, że tym razem wykorzystano P&R (przypadek 3 na rysunku 4.7). Klient wymagający wysokiej jakości transmisji otrzymał inną ścieżkę niż ten akceptujący niską jakość. Ścieżka dla niskiej jakości jest taka sama, jak ta, którą przesyłane są pakiety obciążające sieć. Zaobserwować można, że czas transmisji dla niskiej jakości w stosunku do poprzedniego przypadku praktycznie się nie zmienił, natomiast w przypadku wysokiej jakości uzyskano znaczną poprawę. Czas transmisji uległ zmniejszeniu. Niestety łącze pomiędzy R1 i R2 stanowi tzw. wąskie gardło znacząco wpływające na jakość. Wynika to z faktu, iż każda transmisja między R1-R3 musi przez nie przejść. W związku z tym podejrzewać można, że w przypadku, gdyby istniała alternatywa dla łącza R1-R2, to czas transmisji dla wysokiej jakości zmniejszyłby się i zbliżył do czasu transmisji w przypadku nieobciążonej sieci.

Kolejna emulacja dotyczyła sieci składającej się z 9 ruterów (rysunek 4.2). Również i w tym przypadku do emulacji ruchu tła wykorzystano Trex. Do rutera R1 i R8 podłączonych zostało po 6 klientów – 3 wymagających wysokiej jakości i 3 akceptujących niską jakość. Do ruterów R3, R6 i R9 podłączono po dwóch klientów, z których jeden akceptuje niską jakość transmisji, a drugi wymaga wysokiej jakości transmisji. W przypadku tej emulacji celem było stwierdzenie czy algorytm wyboru tras działa poprawnie, oraz czy koncepcja P&R może zostać zastosowana w przypadku większych sieci. Wyniki zaprezentowano dla klientów podłączonych do rutera R1 – klienta wymagającego wysokiej jakości transmisji i akceptującego niską jakość transmisji. W pierwszej kolejności emulację przeprowadzono dla sieci działającej z maksymalnymi parametrami. Uzyskano następujące wyniki:

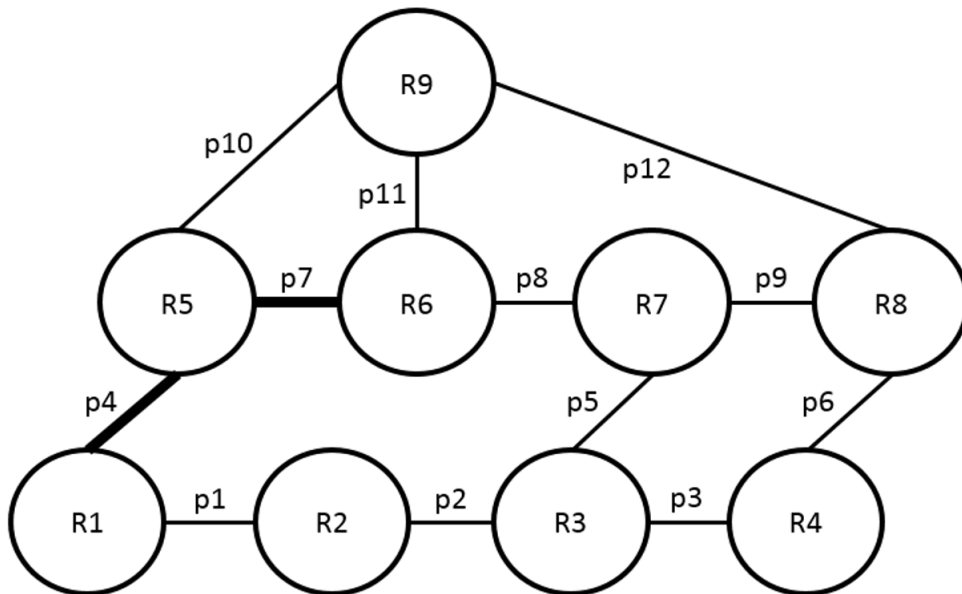
1. W przypadku transmisji pomiędzy klientem podłączonym do rutera R1 (zarówno wymagającego wysokiej jakości transmisji, jak i akceptującego niską jakość transmisji), a klientem podłączonym do rutera R3 wybrana została ścieżka R1-R2-R3 (rysunek 4.8),



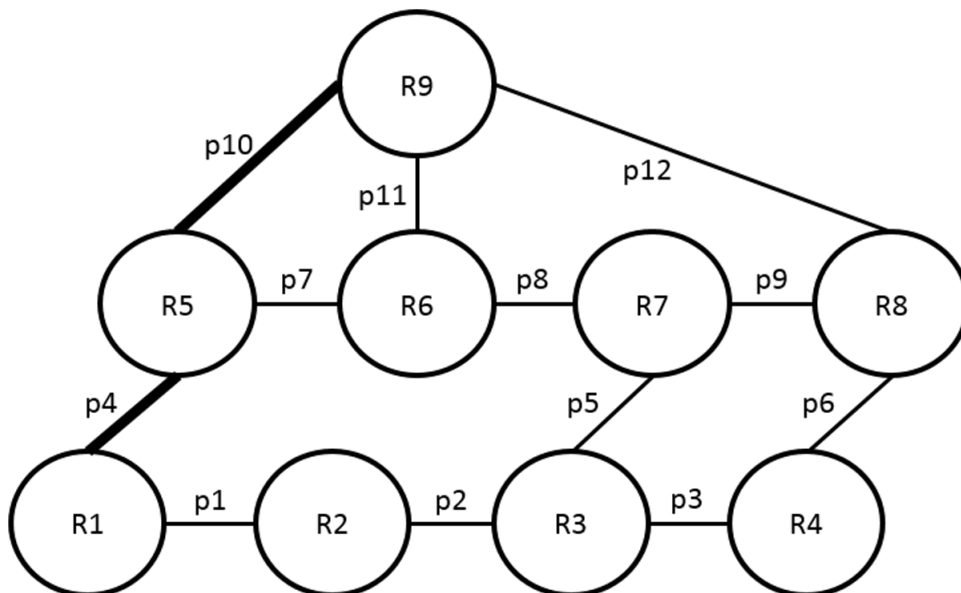
RYSUNEK 4.8: Wybrana ścieżka pomiędzy klientami podłączonymi do R1 i R3 dla sieci działającej z maksymalnymi parametrami

2. Na potrzeby transmisji pomiędzy klientami podłączonymi do rutera R1 (obie jakości transmisji), a klientem podłączonym do R6 wybrana została ścieżka R1-R5-R6 (rysunek 4.9),
3. Dla transmisji pomiędzy klientami podłączonymi do rutera R1 (jakość transmisji wysoka i niska) i klientem podłączonym do R9 wybrana została ścieżka R1-R5-R9 (rysunek 4.10),
4. W przypadku transmisji pomiędzy klientami podłączonymi do rutera R1 i klientami podłączonymi do rutera R8 wybrana została ścieżka R1-R5-R9-R8 (rysunek 4.11).

Powyższe decyzje zgodne są z zasadą działania algorytmu wyznaczania i wyboru tras. Kolejna emulacja przeprowadzona została dla sieci, w której pogorszeniu uległy parametry łącza pomiędzy R1 i R2 (obciążenie transmisją). W przypadku klienta wymagającego wysokiej jakości transmisji podłączonego do R1 ścieżka transmisji do klientów podłączonych do R3 uległa zmianie (rysunek 4.12, linia ciągła, szara). Wynika to z faktu, iż pogorszenie jakości łącza R1-R2 spowodowało, że transmisja dotychczas wykorzystywaną ścieżką nie spełnia wymagań klienta. Dla klienta akceptującego niską jakość transmisji trasa nie uległa zmianie - po zmianie parametrów łącza R1-R2 ścieżka R1-R2-R3 jest zgodna z oczekiwaniami klienta (rysunek 4.12, linia przerywana).

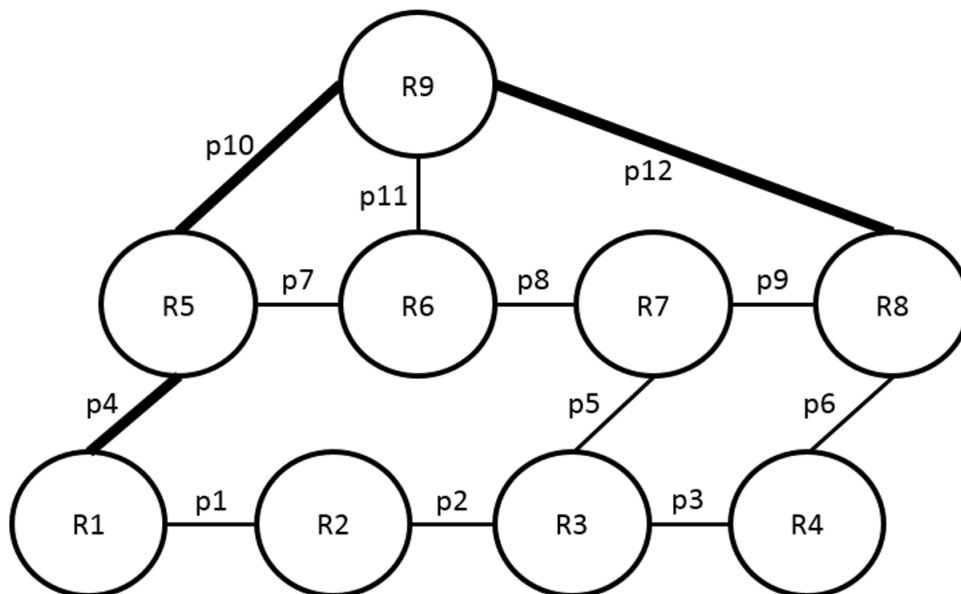


RYSUNEK 4.9: Wybrana ścieżka pomiędzy klientami podłączonymi do R1 i R6 dla sieci działającej z maksymalnymi parametrami

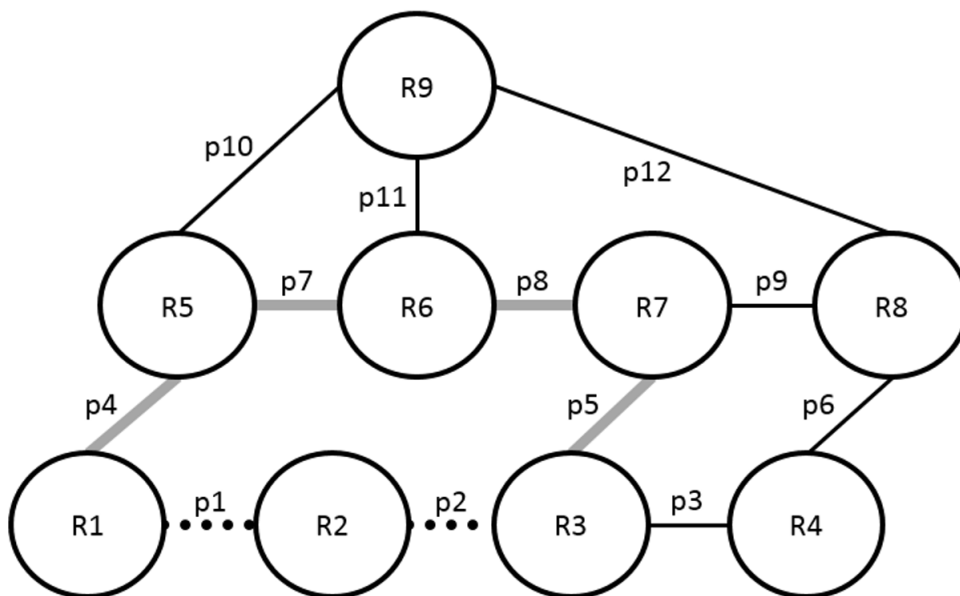


RYSUNEK 4.10: Wybrana ścieżka pomiędzy klientami podłączonymi do R1 i R9 dla sieci działającej z maksymalnymi parametrami

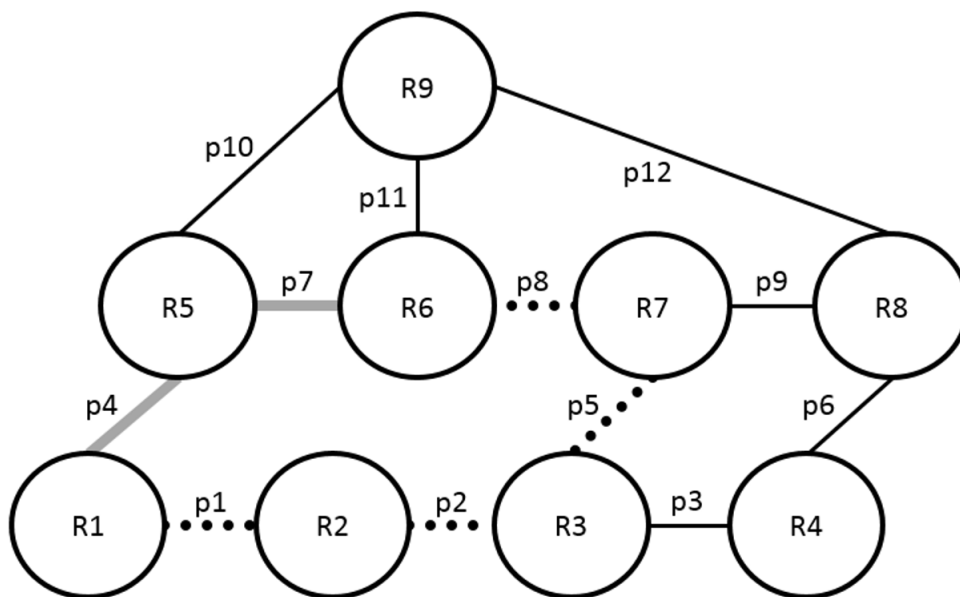




RYSUNEK 4.11: Wybrana ścieżka pomiędzy klientami podłączonymi do R1 i R8 dla sieci działającej z maksymalnymi parametrami



RYSUNEK 4.12: Wybrana ścieżka pomiędzy klientami podłączonymi do R1 i R3 dla sieci, w której pogorszeniu uległy parametry łącza R1-R2

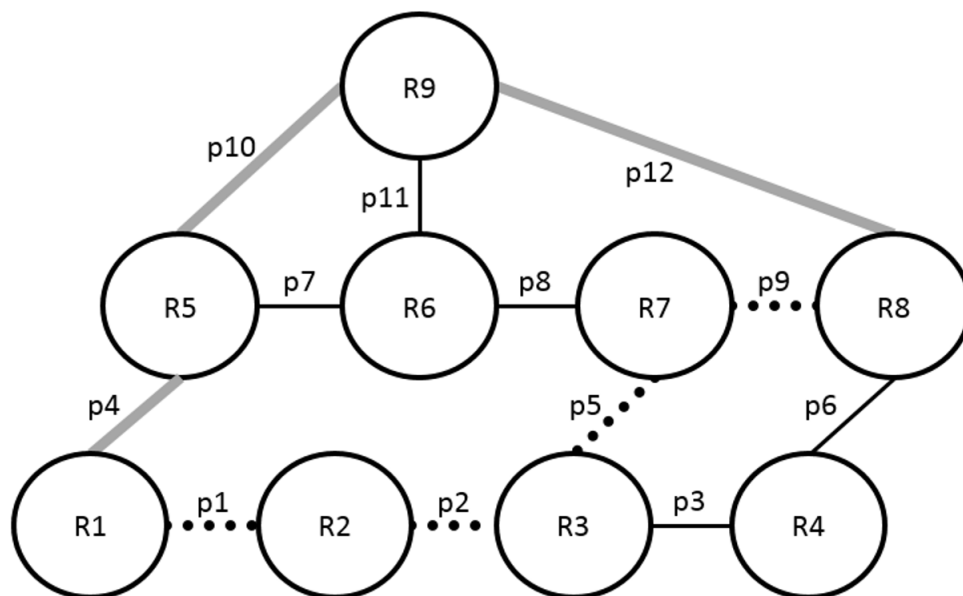


RYSUNEK 4.13: Wybrana ścieżka pomiędzy klientami podłączonymi do R1 i R6 dla sieci, w której pogorszeniu uległy parametry łącza R1-R2

W przypadku transmisji pomiędzy klientami podłączonymi do R1 i R6 trasa dla klienta wymagającego wysokiej jakości transmisji ścieżka nie uległa zmianie (rysunek 4.13, linia ciągła, szara). Zmianie uległa ścieżka wykorzystywana na potrzeby transmisji klienta akceptującego niską jakość (rysunek 4.13, linia przerywana). W tym przypadku wybrana została ścieżka R1-R2-R3-R7-R6, a wybór ten wynika z faktu, iż poprzez pogorszenie parametrów łącza R1-R2 transmisja tą ścieżką jest zgodna z oczekiwaniami klienta.

Z kolei w przypadku komunikacji pomiędzy klientem akceptującym niską jakość transmisji podłączonym do rutera R1 i klientami podłączonymi do rutera R8 ścieżka po rekonfiguracji ma postać R1-R2-R3-R7-R8 (rysunek 4.14, linia przerywana). Ścieżka ta wybrana została zgodnie z zasadą działania algorytmu wyboru ścieżek tzn. posiada najmniejszą liczbę skoków pomiędzy źródłem, a celem. Nie jest to jedyna możliwa ścieżka zgodna z przyjętą zasadą – alternatywę stanowi ścieżka R1-R2-R3-R4-R8. Decyzja wynika z pozycji danej ścieżki na liście agenta – wybrana ścieżka znajdowała się wyżej na liście ścieżek. Z kolei dla klienta wymagającego wysokiej jakości transmisji ponownie ścieżka nie uległa zmianie (rysunek 4.14, linia ciągła, szara).

Gdy klient akceptujący niską jakość będzie się komunikował z klientami podłączonymi do rutera R9 transmisja będzie się odbywała ścieżką R1-R2-R3-R7-R8-R9. Ponownie nie jest to jedyna możliwa trasa o tej samej, najmniejszej liczbie skoków.



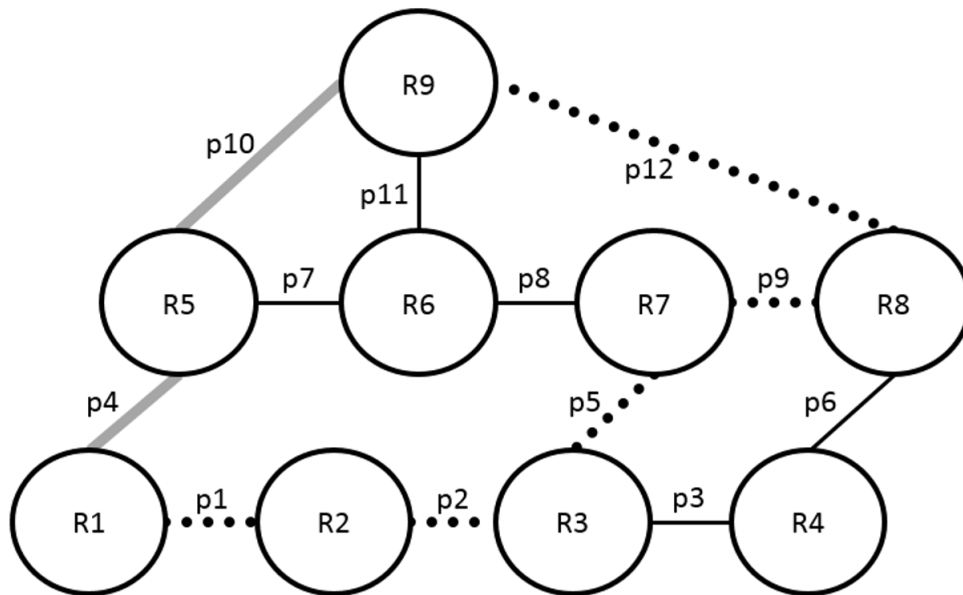
RYSUNEK 4.14: Wybrana ścieżka pomiędzy klientami podłączonymi do R1 i R8 dla sieci, w której pogorszeniu uległy parametry łącza R1-R2

Trasa dla tego przypadku zaprezentowana została na rysunku 4.15 (linia przerywana). Dla klienta wymagającego wysokiej jakości w przypadku transmisji z klientami podłączonymi do R9 zmiana trasy nie wystąpiła (rysunek 4.15, linia ciągła, szara).

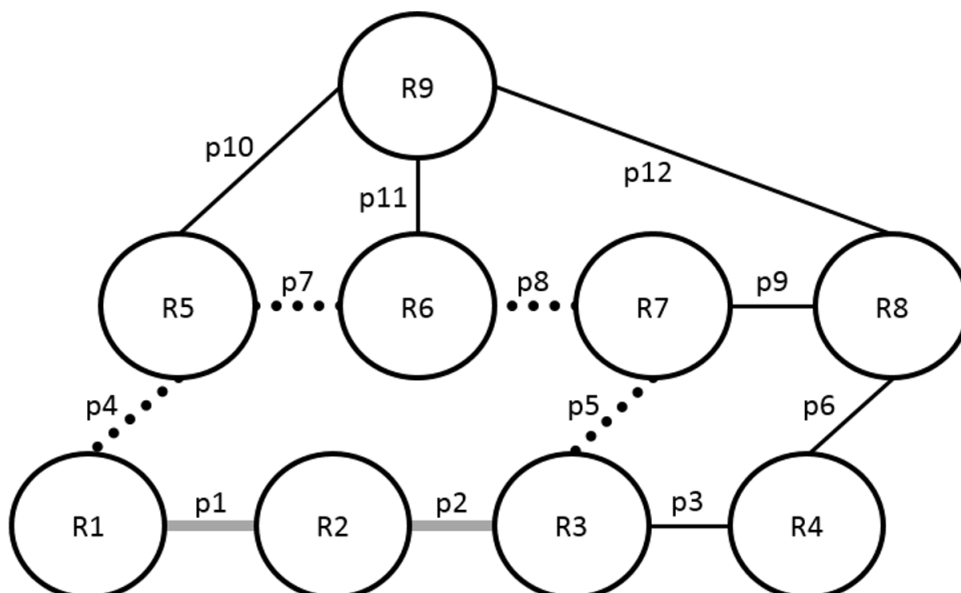
W przypadku kolejnej emulacji pogorszeniu uległy parametry łącza R5-R6. Pozostałe łącza działają z maksymalnymi parametrami. Pogorszenie parametrów uzyskano poprzez równoczesną transmisję wielu porcji danych przez łącze. Dla transmisji pomiędzy klientem wymagającym wysokiej jakości transmisji, a klientami podłączonymi do R3 wybrana została ścieżka R1-R2-R3 (rysunek 4.16, linia ciągła, szara) co oznacza, iż rekonfiguracja ścieżki nie była konieczna. Z kolei dla klienta akceptującego niską jakość dla transmisji z klientami podłączonymi do R3 wybrana została ścieżka R1-R5-R6-R7-R3 (rysunek 4.16, linia przerywana). Wybór ten wynika z faktu, iż w momencie, gdy pogorszeniu uległy parametry łącza R5-R6 ścieżka wybrana przez algorytm spełnia oczekiwania klienta.

Dla obu typów klientów w przypadku transmisji z klientami podłączonymi do R6 ścieżka jest taka sama na odcinku R1-R5 (rysunek 4.17, linia ciągła, pogrubiona, czarna). Od routera R5 transmisja klienta wymagającego wysokiej jakości odbywa się ścieżką R5-R9-R6 (rysunek 4.17, linia ciągła, szara), a transmisja klienta akceptującego niską jakość transmisji ścieżką R5-R6 (rysunek 4.17, linia przerywana). Oznacza to, że ścieżka krótsza w tym wypadku nie jest wystarczająca na potrzeby transmisji z wysoką jakością.

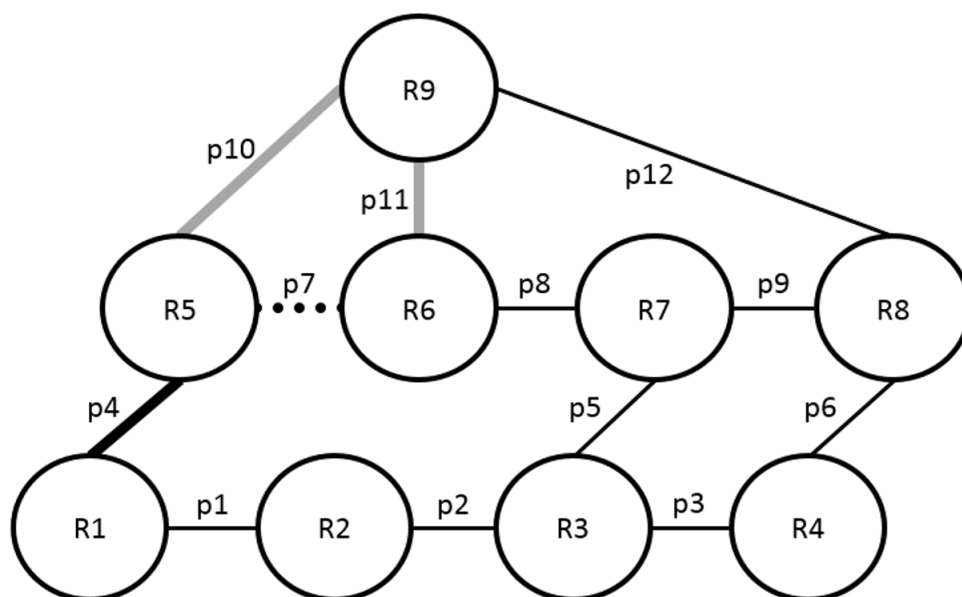
Dla transmisji pomiędzy klientami podłączonymi do R1 i R9 ponownie ścieżka



RYSUNEK 4.15: Wybrana ścieżka pomiędzy klientami podłączonymi do R1 i R9 dla sieci, w której pogorszeniu uległy parametry łącza R1-R2



RYSUNEK 4.16: Wybrana ścieżka pomiędzy klientami podłączonymi do R1 i R3 dla sieci, w której pogorszeniu uległy parametry łącza R5-R6

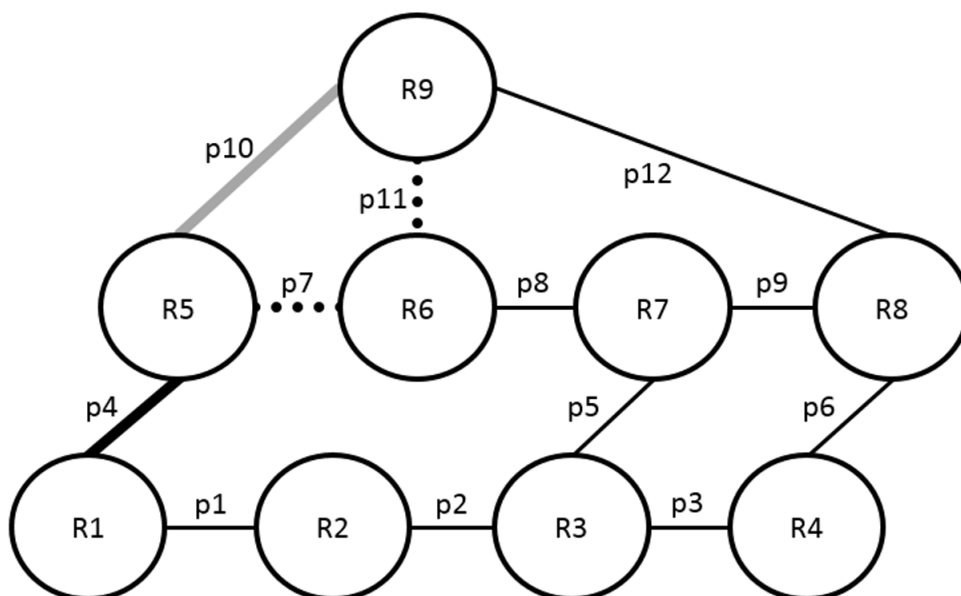


RYSUNEK 4.17: Wybrana ścieżka pomiędzy klientami podłączonymi do R1 i R6 dla sieci, w której pogorszeniu uległy parametry łącza R5-R6

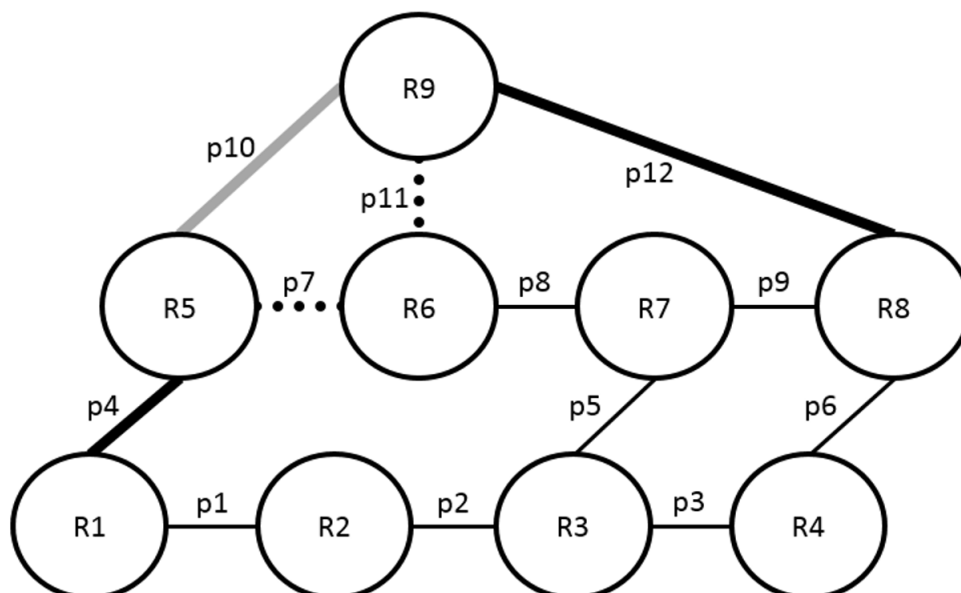
transmisji jest taka sama na odcinku R1-R5 (rysunek 4.18, linia ciągła, pogrubiona, czarna). Następnie w przypadku klienta akceptującego niską jakość, transmisja odbywa się ścieżką R5-R6-R9 (rysunek 4.18, linia przerywana). Z kolei ścieżka wykorzystywana przez klienta wymagającego wysokiej jakości transmisji nie uległa zmianie w stosunku do ścieżki w sieci działającej z maksymalnymi parametrami (rysunek 4.18, linia ciągła, szara).

Ostatni przypadek w ramach tej emulacji to transmisja pomiędzy klientami podłączonymi do R1 i R8. Dla klienta wymagającego wysokiej jakości transmisji ścieżka nie uległa zmianie w stosunku do ścieżki wybranej w przypadku sieci działającej z maksymalnymi parametrami (rysunek 4.19, linie pogrubione czarna i szara). Z kolei ścieżka klienta akceptującego niską jakość transmisji uległa zmianie co wynika z faktu, iż przy pogorszeniu parametrów łącza R5-R6 ścieżka R1-R5-R6-R9-R8 jest zgodna z oczekiwaniami klienta (rysunek 4.19, linie pogrubiona czarna i przerywana). Dla obu typów klientów ścieżki są takie same na odcinku R1-R5 oraz R9-R8.

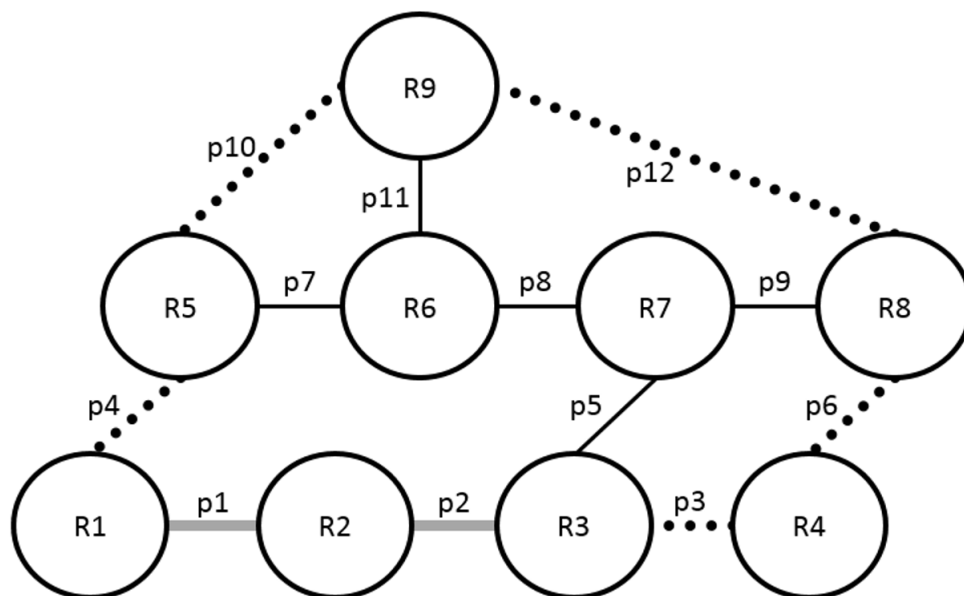
Ostatnia emulacja dla tej sieci przeprowadzona została dla przypadku, w którym pogorszeniu uległy parametry łącza p10 (R5-R9). Ponownie przeprowadzono badania dotyczące wyboru ścieżki w przypadku transmisji pomiędzy klientem podłączonym do rutera R1 (niska i wysoka jakość), a klientami podłączonymi do rutera R3, R6, R8 i R9. Na rysunku 4.20 zaznaczono ścieżki, które zostały wybrane dla poszczególnych klientów. Dla klienta wymagającego wysokiej jakości transmisji ścieżka nie uległa zmianie w stosunku do ścieżki wybranej w przypadku sieci działającej z maksymalnymi parametrami (rysunek 4.20, linia ciągła, szara). Spowodowane jest



RYSUNEK 4.18: Wybrana ścieżka pomiędzy klientami podłączonymi do R1 i R9 dla sieci, w której pogorszeniu uległy parametry łącza R5-R6



RYSUNEK 4.19: Wybrana ścieżka pomiędzy klientami podłączonymi do R1 i R8 dla sieci, w której pogorszeniu uległy parametry łącza R5-R6

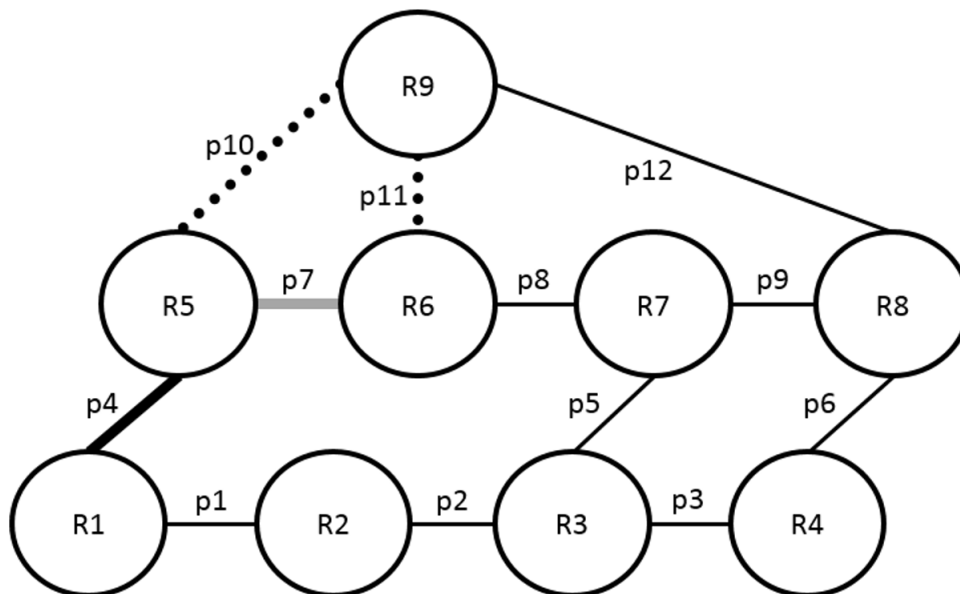


RYSUNEK 4.20: Wybrana ścieżka pomiędzy klientami podłączonymi do R1 i R3 dla sieci, w której pogorszeniu uległy parametry łącza R5-R9

to tym, iż transmisja tą ścieżką jest zgodna z wymaganiami klienta. Z kolei ścieżka transmisji pomiędzy klientem akceptującym niską jakość transmisji podłączonym do R1, a klientami podłączonymi do R3 uległa zmianie – po rekonfiguracji wykorzystana zostanie ścieżka R1-R5-R9-R8-R4-R3 (rysunek 4.20, linia przerywana). Analizując wszystkie możliwe ścieżki można dojść do wniosku, iż wybrana nie jest najkrótsza. Jednakże biorąc pod uwagę, że pogorszeniu uległy parametry łącza p10 i fakt, iż łącze to stanowi wąskie gardło mające bezpośredni wpływ na ocenę całej ścieżki, to wybór tej ścieżki jest uzasadniony. Obniżenie parametrów spowodowało, że transmisja daną ścieżką odbywa się z jakością oczekiwaną przez klienta.

Z kolei na potrzeby transmisji pomiędzy użytkownikiem akceptującym niską jakość, a użytkownikami podłączonymi do rutera R6 ścieżka uległa zmianie. Wybrana została ścieżka R1-R5-R9-R6 (rysunek 4.21, linia przerywana). Ścieżka w przypadku transmisji użytkownika wymagającego wysokiej jakości nie uległa zmianie (rysunek 4.21, linia ciągła, szara) w stosunku do ścieżki wybranej, gdy sieć działa z maksymalnymi parametrami (rysunek 4.9). Niezależnie czy transmisja dotyczy klienta wymagającego wysokiej jakości czy akceptującego niską jakość elementem wspólnym ścieżek jest wykorzystanie łącza p4 (rysunek 4.21, linia ciągła, czarna).

Kolejny przypadek dotyczył wyboru ścieżki w sytuacji, w której transmisja ma



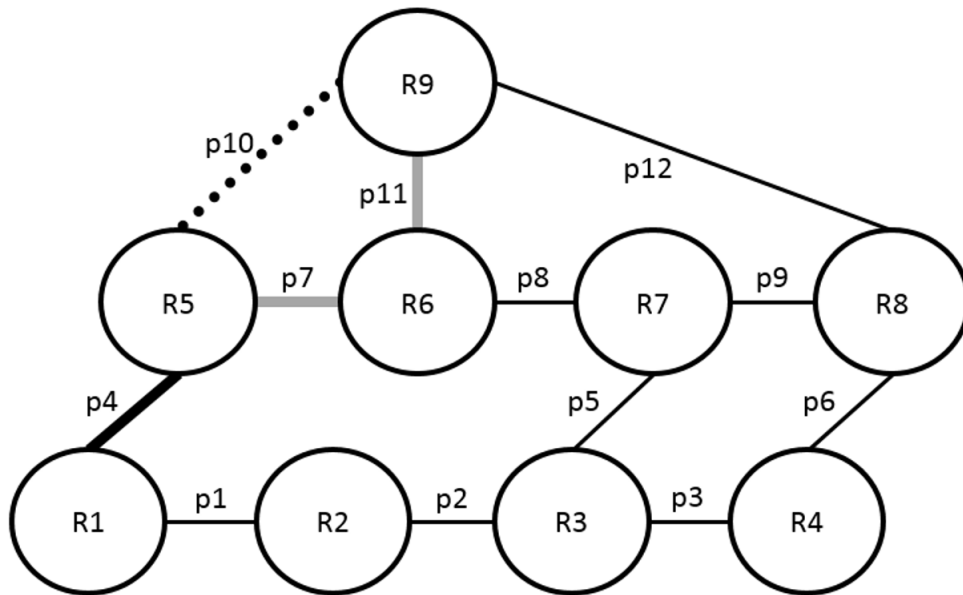
RYSUNEK 4.21: Wybrana ścieżka pomiędzy klientami podłączonymi do R1 i R6 dla sieci, w której pogorszeniu uległy parametry łącza R5-R9

się odbywać pomiędzy klientami podłączonymi do R1 i R9. Na rysunku 4.22 przedstawione zostały wybrane ścieżki. Elementem wspólnym obu ścieżek (dla obu typów klientów) jest łącze p4 (rysunek 4.22, linia ciągła, czarna). Dla klienta wymagającego wysokiej jakości transmisji wybrana została ścieżka R1-R5-R6-R9 (rysunek 4.22, linia ciągła, szara). Oznacza to zmianę ścieżki w stosunku do tej która wybrana została w przypadku sieci działającej z maksymalnymi parametrami (rysunek 4.10). Wynika to z faktu, iż parametry łącza R5-R9 uległy pogorszeniu, a co za tym idzie parametry dotychczasowej ścieżki nie są zgodne z wymaganiami klienta. Z kolei ścieżka wybrana na potrzeby transmisji klienta akceptującego niską jakość nie uległa zmianie (rysunek 4.22, linia przerywana) co oznacza, że po pogorszeniu się parametrów łącza p10 ścieżka spełnia oczekiwania klienta, jak chodzi o jakość transmisji.

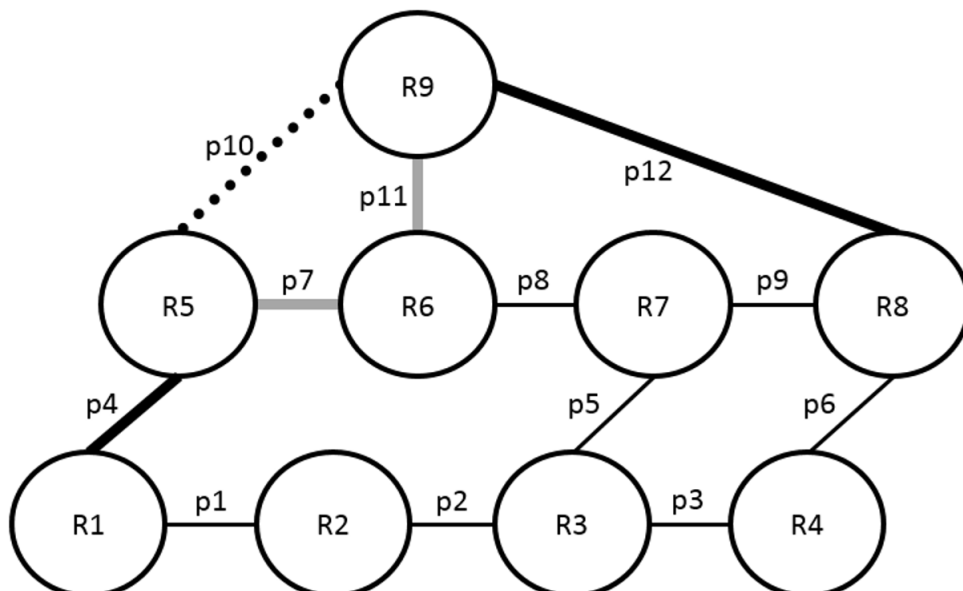
Ostatni wybór dotyczył ścieżek pomiędzy klientami podłączonymi do R1 i klientami podłączonymi do R8. W przypadku klienta akceptującego niską jakość ścieżka transmisji (rysunek 4.23, linia czarna, ciągła i przerywana) nie uległa zmianie w stosunku do tej wybranej w przypadku sieci działającej z maksymalnymi parametrami (rysunek 4.11). Na potrzeby transmisji klienta wymagającego wysokiej jakości, ścieżka została zmieniona na R1-R5-R6-R9-R8 (rysunek 4.23, linia pogrubiona szara i czarna). Dla obu ścieżek elementami wspólnymi, czyli wykorzystywanymi dla transmisji w obu jakościach są łącza p4 i p12 (rysunek 4.23, linia pogrubiona czarna).

W przypadku obu emulacji (sieć składająca się z 4 i 9 ruterów) wykazano, że





RYSUNEK 4.22: Wybrana ścieżka pomiędzy klientami podłączonymi do R1 i R9 dla sieci, w której pogorszeniu uległy parametry łącza R5-R9



RYSUNEK 4.23: Wybrana ścieżka pomiędzy klientami podłączonymi do R1 i R8 dla sieci, w której pogorszeniu uległy parametry łącza R5-R9

algorytm wyboru ścieżek działa prawidłowo. Oznacza to, że ścieżki transmisji wybierane są zgodnie z oczekiwaniami klientów. Klient otrzymuje dokładnie taką jakość transmisji jaką oczekuje. Nie byłoby to możliwe, gdyby w sieci wykorzystano tradycyjne metody routingu, tzn. takie, które nie pozwalają na różnicowanie ścieżek transmisji w oparciu o inny parametr niż cel transmisji.

## 4.2 Symulacja

Symulacja przeprowadzona została dla dwóch topologii sieci. Pierwsza z nich jest zgodna z topologią wykorzystaną w przypadku emulacji (rysunek 4.1). Sieć składa się z 4 ruterów, każde łącze pomiędzy ruterami posiada przepływność 1000 Mb/s i opóźnienie transmisji 1 ms. Do rutera R1 oraz R3 podłączonych zostało po 2 użytkowników. przepływność każdego łącza użytkownik-ruter wynosi 100 Mb/s, a opóźnienie 2 ms. Zarówno w przypadku rutera R1, jak i R3 jeden użytkownik wymaga wysokiej jakości transmisji, a drugi akceptuje niską jakość transmisji. Na potrzeby badań wykorzystano program ping (opóźnienia transmisji) oraz transmisję pliku o wielkości 100 MB wypełnionego losowymi danymi. Symulacja dla tej topologii miała na celu umożliwienie porównania sposobu działania algorytmu użytego w przypadku symulatora i emulatora. Istotne było wykazanie zmian w czasie i opóźnieniu transmisji oraz obserwacja modyfikacji ścieżek routingu. Pomiarów dokonano dla trzech przypadków - sieć nieobciążona, sieć znacząco obciążona (duży ruch na łączu R2-R3), sieć znacząco obciążona (łącze R2-R3) z wykorzystaniem P&R.

Wyniki zaprezentowano w tabeli 4.2. Zauważyć można, iż w przypadku symulacji z wykorzystaniem NS3 (S) uzyskano taką samą zależność, jak w przypadku wcześniejszych badań - emulacji (E), tzn. w przypadku sieci obciążonej, w której wykorzystano P&R dla klienta wymagającego wysokiej jakości transmisji uzyskano wynik zbliżony do referencyjnego (sieć nieobciążona), a dla klienta akceptującego niską jakość transmisji wynik zbliżony jest do tego, który uzyskany został dla sieci obciążonej. Dla klienta wymagającego wysokiej jakości nastąpiła rekonfiguracja trasy (zmiana parametrów) co doprowadziło do separacji ścieżek obu klientów (niska jakość transmisji ścieżka R1-R2-R3, wysoka jakość transmisji R1-R2-R4-R3) i różnicy w uzyskanych czasach. Zauważyć można różnice w czasach uzyskanych dla symulacji i emulacji - wynikają one z różnic w działaniu warstwy fizycznej P&R w symulacji oraz emulacji. W warstwie fizycznej P&R wykorzystano algorytmy dostępne w NS3 (symulacja) i Vyatta (emulacja).

Na rysunku 4.24 i rysunku 4.25 zaprezentowano wyniki w formie wykresów. Wyniki dotyczą następujących przypadków:

1 – Brak obciążenia, jakość wysoka,

| Metodologia   | Jakość | E/S | Średnie opóźnienie[ms] | Przedział ufności | Średni czas transmisji (plik 100 MB) [s] | Przedział ufności |
|---|--------|-----|------------------------|-------------------|--|-------------------|
| Brak obciążenia, sieć działa z maksymalnymi parametrami, ta sama ścieżka dla obu jakości                      | Wysoka | E   | 3,102                  | 4,64% (0,144)     | 19,020                                   | 0,90% (0,173)     |
|   |        | S   | 4,332                  | 4,53% (0,196)     | 6,514                                    | 0,04% (0,002)     |
|   | Niska  | E   | 3,232                  | 4,49% (0,145)     | 19,100                                   | 0,92% (0,176)     |
|   |        | S   | 4,361                  | 4,77% (0,208)     | 6,513                                    | 0,07% (0,005)     |
| Sieć znacząco obciążona, duże natężenie ruchu, ta sama ścieżka dla obu jakości                                | Wysoka | E   | 8,802                  | 3,29% (0,290)     | 35,110                                   | 2,49% (0,876)     |
|   |        | S   | 14,479                 | 2,32% (0,335)     | 23,293                                   | 0,07% (0,016)     |
|   | Niska  | E   | 8,784                  | 3,30% (0,290)     | 36,020                                   | 2,60% (0,936)     |
|   |        | S   | 14,509                 | 2,11% (0,307)     | 23,292                                   | 0,05% (0,012)     |
| Sieć znacząco obciążona, duże natężenie ruchu, wykorzystano Pay&Require - ścieżka zależy od wymaganej jakości | Wysoka | E   | 6,870                  | 4,25% (0,292)     | 28,730                                   | 1,22% (0,350)     |
|   |        | S   | 4,262                  | 3,83% (0,163)     | 6,513                                    | 0,04% (0,003)     |
|   | Niska  | E   | 8,893                  | 3,58% (0,318)     | 35,580                                   | 2,51% (0,893)     |
|   |        | S   | 14,679                 | 2,99% (0,439)     | 23,293                                   | 0,02% (0,005)     |

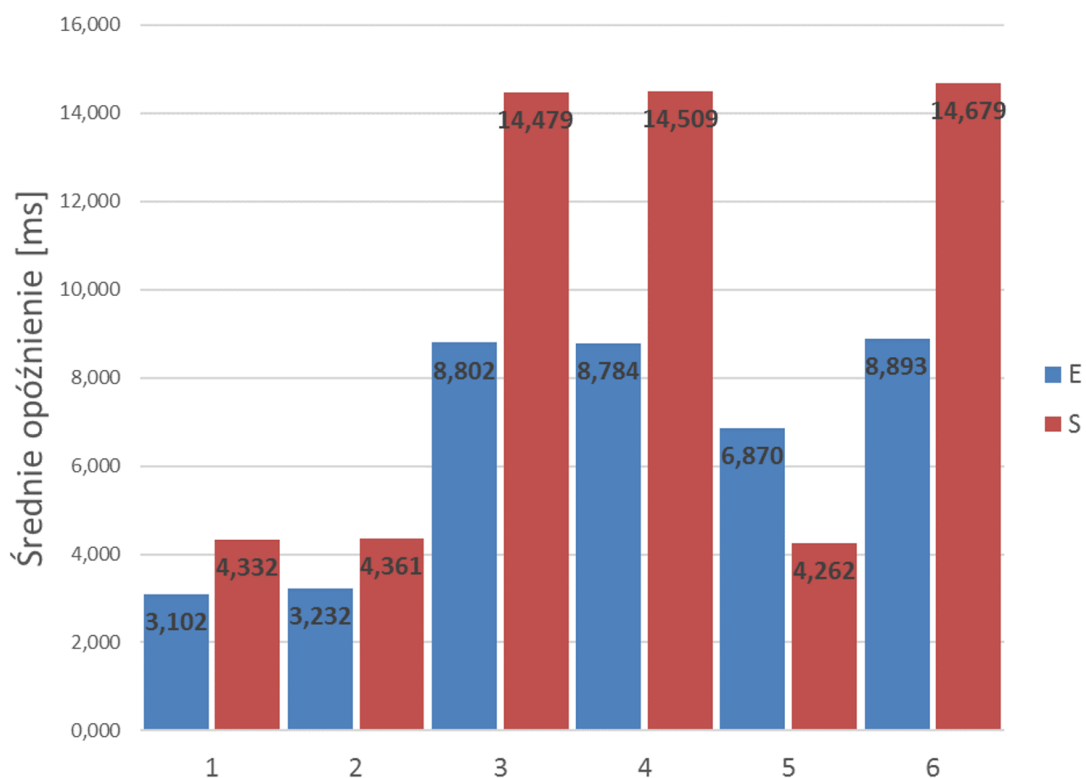
TABELA 4.2: Wyniki symulacji wraz z wynikami emulacji i estymacją dla przyjętego poziomu ufności  $\beta=95\%$

- 2 – Brak obciążenia, jakość niska,
- 3 – Sieć obciążona, jakość wysoka,
- 4 – Sieć obciążona, jakość niska,
- 5 – Sieć obciążona, P&R, jakość wysoka,
- 6 – Sieć obciążona, P&R, jakość niska.

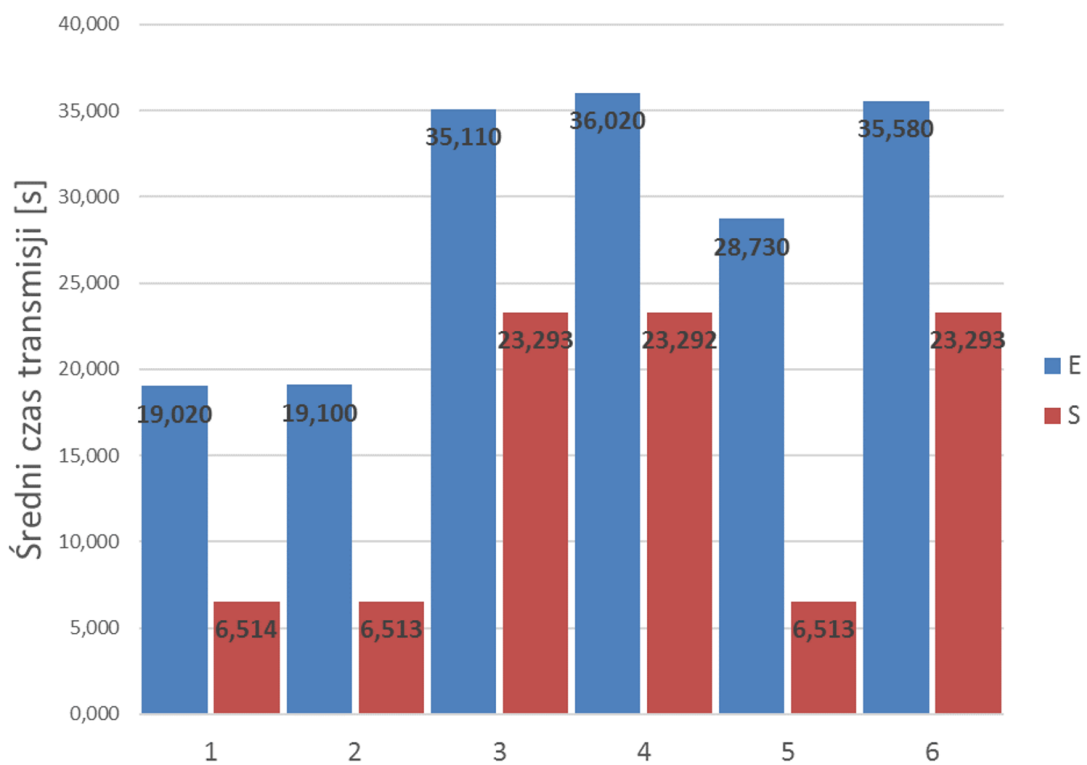
Kolejna symulacja dotyczyła większej sieci (rysunek 4.26), składającej się z 12 węzłów. Każde łącze pomiędzy ruterami posiada przepływność 1000 Mb/s i opóźnienie transmisji 1 ms. Do rutera Szczecin i Rzeszów podłączono po 2 klientów – jeden wymaga wysokiej jakości transmisji, drugi akceptuje niską jakość transmisji. Przepływność każdego łącza użytkownik-ruter wynosi 100 Mb/s, a opóźnienie transmisji 2 ms.

Istotne w przypadku tej sieci było to, jakie ścieżki zostaną wybrane. W przypadku sieci nieobciążonej wybrana została ścieżka Szczecin-Kołobrzeg-Gdańsk-Białystok-Rzeszów, która spełnia wymagania obu klientów (rysunek 4.27). Jest to działanie zgodne z przyjętymi założeniami, ponieważ w sieci wszystkie ścieżki posiadają taką samą ocenę, a co za tym idzie nie ma możliwości zróżnicowania ścieżek w oparciu o wymagania klientów. W tym przypadku wybór podyktowany był długością ścieżki.

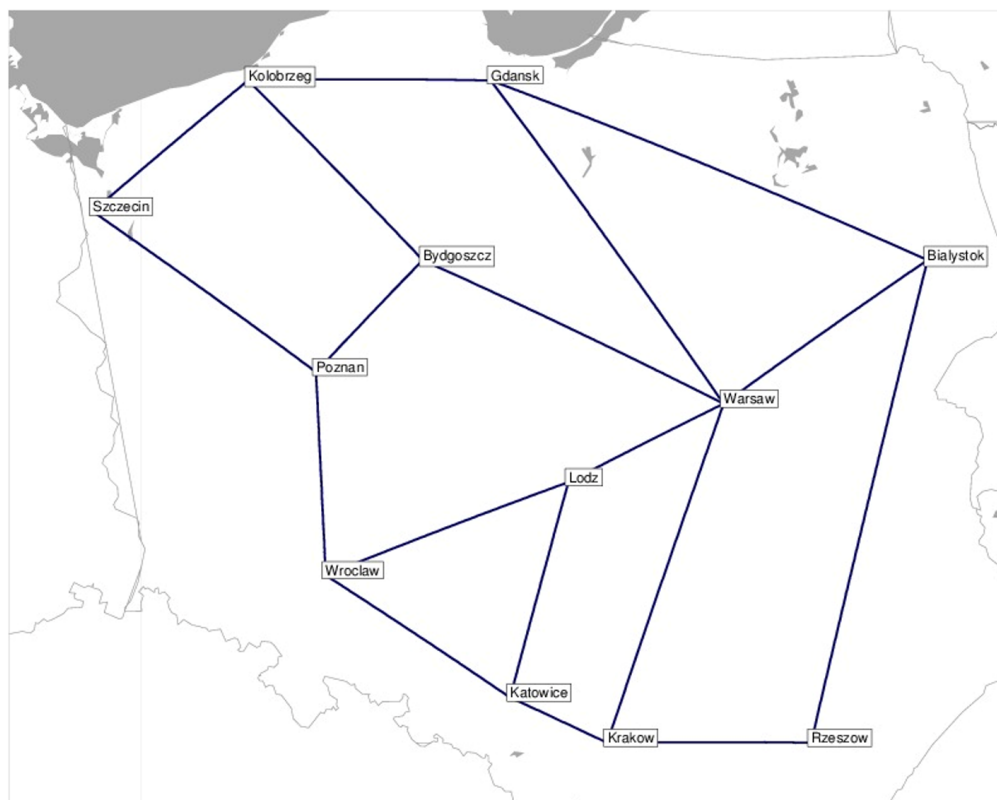
Następnie obciążone zostało łącze Gdańsk-Białystok. Gdy zastosowano klasyczne podejście nie nastąpiła rekonfiguracja (rysunek 4.27). Z kolei w momencie, gdy wykorzystano podejście P&R dla klienta wymagającego wysokiej jakości transmisji



RYSUNEK 4.24: Wyniki uzyskane dla symulacji (S) i emulacji (E) – średnie opóźnienie transmisji



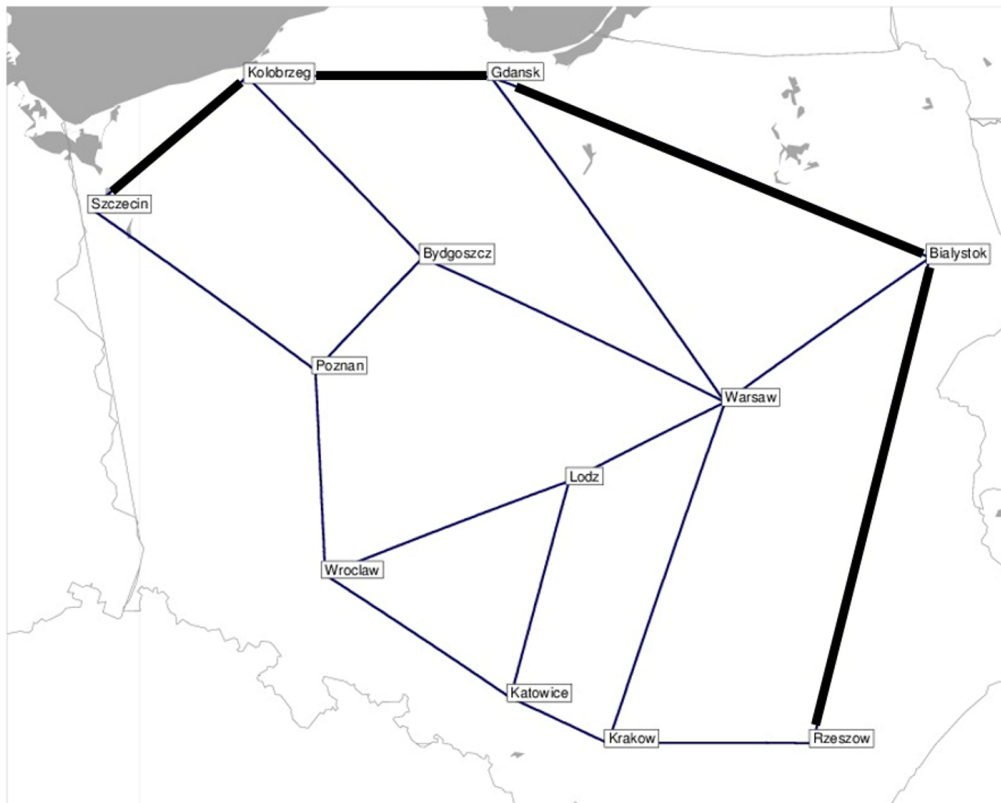
RYSUNEK 4.25: Wyniki uzyskane dla symulacji (S) i emulacji (E) – średni czas transmisji



RYSUNEK 4.26: Schemat sieci wykorzystanej w symulacji  
Źródło: SNDLib

zmodyfikowana została ścieżka na Szczecin-Kolobrzeg-Gdańsk-Warszawa-Białystok-Rzeszów (rysunek 4.28, linia ciągła, czarna i szara). Dla klienta akceptującego niską jakość transmisji ścieżka pozostała bez zmian (rysunek 4.28, linia czarna, ciągła i przerywana). Zachowanie to jest zgodne z założeniami koncepcji P&R, ponieważ w momencie pogorszenia parametrów łącza Gdańsk-Białystok ocena całej ścieżki została obniżona do poziomu niskiego, w tym przypadku zgodnego z oczekiwaniami klienta akceptującego niską jakość transmisji. W przypadku obu ścieżek występują elementy wspólne, czyli łącza przez które odbywa się transmisja zarówno klienta wymagającego wysokiej jakości, jak i akceptującego niską jakość (rysunek 4.28, linia ciągła, czarna)

Poza wyborem ścieżek istotne były również pomiary dotyczące czasu transmisji i opóźnień. Wyniki dotyczące tych parametrów zaprezentowano w tabeli 4.3. Zauważyć można, iż w przypadku sieci obciążonej, w której wykorzystano P&R czas transmisji i wartość opóźnienia zbliżone są do wyników uzyskanych dla sieci nieobciążonej. Z kolei dla klienta akceptującego niską jakość transmisji (rekonfiguracja nie była konieczna) wyniki są zbliżone do tych uzyskanych dla sieci obciążonej, w



RYSUNEK 4.27: Wybrana ścieżka pomiędzy klientami dla sieci działającej z maksymalnymi parametrami

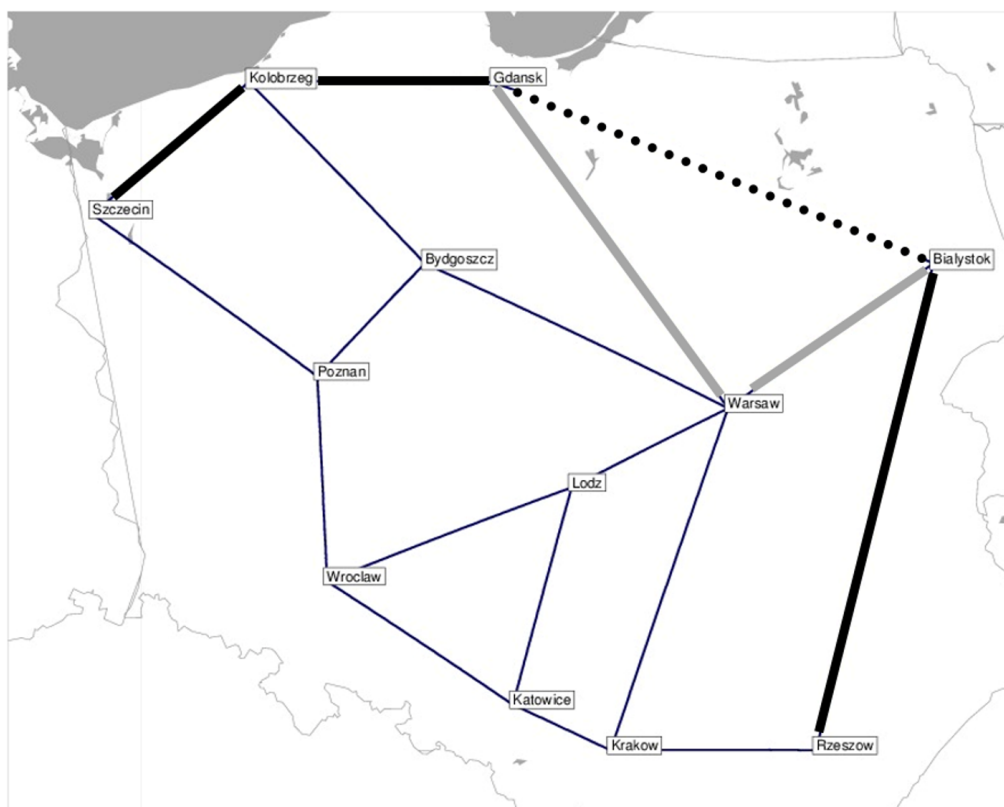
której nie wykorzystano P&R. Na rysunku 4.29 i 4.30 uzyskane wyniki przedstawiono w formie wykresów. Przypadek 1 to sieć, działająca z maksymalnymi parametrami, przypadek 2 to sieć obciążona, a przypadek 3 to wyniki uzyskane dla sieci obciążonej w której wykorzystano koncepcję P&R.

## 4.3 Aukcje

Badania przeprowadzono dla aukcji jednego dobra oraz kombinatorycznych (wielu dóbr). W aukcjach jednego dobra koniecznym jest przekształcenie jakości (na którą wpływ ma wiele parametrów) w oparciu o pewną regułę mapowania parametrów na jakość. Dzięki takiemu rozwiązaniu z problemu, który dotyczy wielu dóbr (np. przepływność, opóźnienie transmisji, współczynnik utraty pakietów) można uzyskać mniej złożony problem decyzyjny.

### 4.3.1 Jednokrotne aukcje jednego dobra

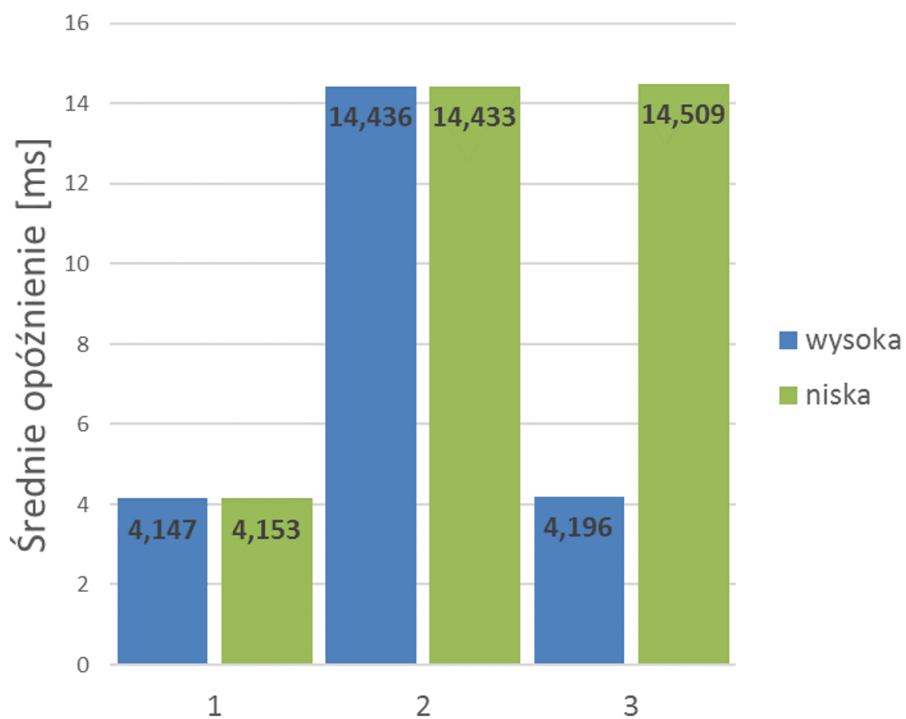
Badania przeprowadzono dla agenta, którego zadaniem jest dokonanie zakupu na rzecz klienta. Jakość usługi, jaką jest transmisja danych została zdefiniowana



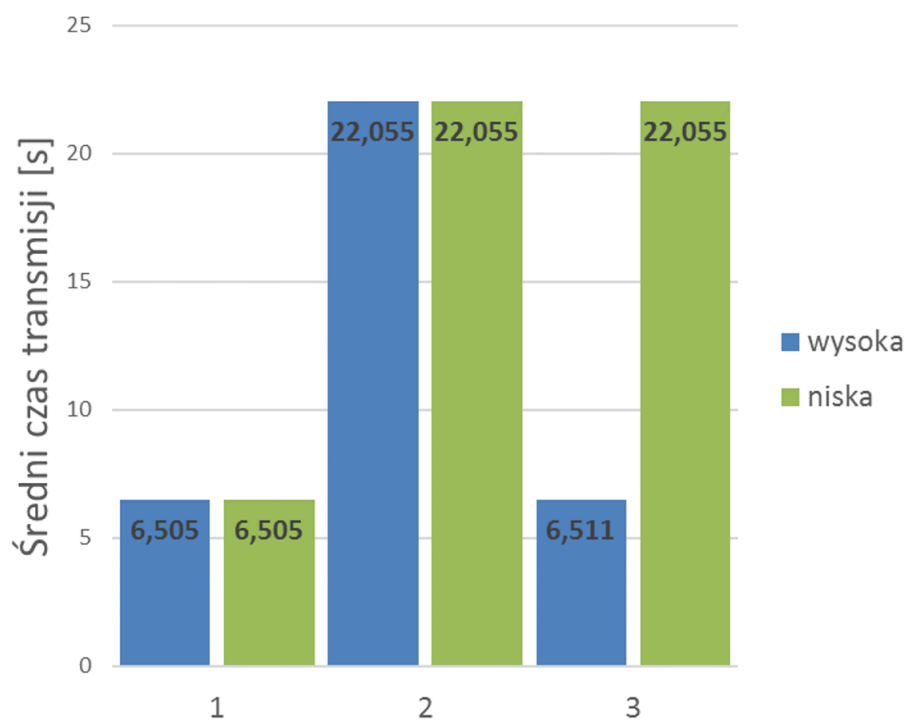
RYSUNEK 4.28: Wybrana ścieżka pomiędzy klientami dla sieci działającej z obniżonymi parametrami łącza Gdańsk-Białystok

| Metodologia   | Jakość | Średni ping [ms] | Przedział ufności | Średni czas transmisji (plik 100MB) [min] | Przedział ufności |
|---|--------|------------------|-------------------|---|-------------------|
| Brak obciążenia, sieć działa z maksymalnymi parametrami, ta sama ścieżka dla obu jakości                      | Wysoka | 4,147            | 2,39% (0,099)     | 6,505                                     | 0,00% (0,000)     |
|   | Niska  | 4,153            | 2,33% (0,097)     | 6,505                                     | 0,00% (0,000)     |
| Sieć znacząco obciążona, duże natężenie ruchu, ta sama ścieżka dla obu jakości                                | Wysoka | 14,436           | 2,14% (0,309)     | 22,055                                    | 0,00% (0,000)     |
|   | Niska  | 14,433           | 2,12% (0,306)     | 22,055                                    | 0,00% (0,000)     |
| Sieć znacząco obciążona, duże natężenie ruchu, wykorzystano Pay&Require - ścieżka zależy od wymaganej jakości | Wysoka | 4,196            | 4,39% (0,184)     | 6,505                                     | 0,00% (0,000)     |
|   | Niska  | 14,509           | 2,10% (0,304)     | 22,055                                    | 0,00% (0,000)     |

TABELA 4.3: Wyniki symulacji dla sieci Poland wraz z estymacją dla poziomu ufności  $\beta=95\%$



RYSUNEK 4.29: Wyniki uzyskane dla sieci Poland – średnie opóźnienie transmisji



RYSUNEK 4.30: Wyniki uzyskane dla sieci Poland – średni czas transmisji



| Klient | Kwota | Jakość |
|--------|-------|--------|
| K2     | 4     | 3      |
| K3     | 4     | 3      |
| K4     | 4     | 3      |
| K5     | 14    | 3      |
| K6     | 14    | 3      |
| K7     | 14    | 3      |
| K8     | 24    | 3      |
| K9     | 24    | 3      |
| K10    | 24    | 3      |

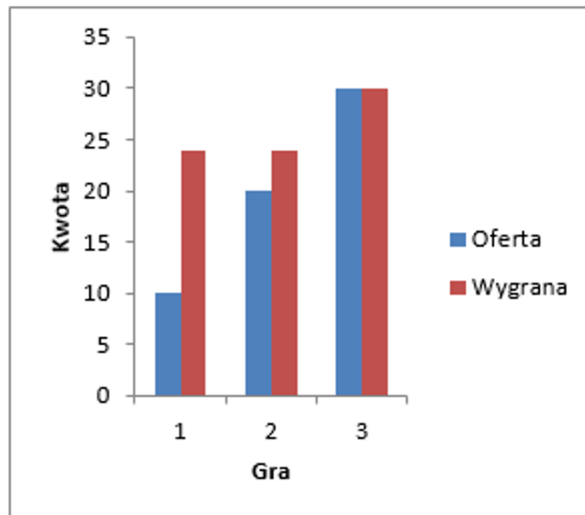
TABELA 4.4: Oponenci w przypadku strategii 1

w przedziale od 1 do 5. Badania przeprowadzono dla aukcji *first-price* oraz *second-price*. Aukcje przeprowadzane są w jednej rundzie tzn. agenci równocześnie składają oferty i robią to tylko raz. W obu przypadkach wygrywa najwyższa oferta, przy czym w *second-price* końcowa kwota to druga co do wielkości ze złożonych ofert. Na potrzeby symulacji aukcji zdefiniowano 10 klientów. Klient numer 1 analizowany jest pod kątem osiągniętego wyniku gry. Pozostali klienci stanowią jego przeciwników, przy czym nie wymieniają się informacjami dotyczącymi przyjętej strategii. Dla strategii 1 wszyscy klienci zainteresowani są taką samą jakością. W tabeli 4.4 przedstawiono przyjęte przez klientów maksymalne kwoty.

Dla strategii 1 uzyskane wyniki przedstawiono na rysunku 4.31. Pomiar przeprowadzono dla trzech różnych kwot maksymalnych, akceptowalnych dla klienta 1 tj. 10, 20 i 30 (odpowiednio gra 1, 2 i 3). Na wykresie umieszczono dwie wartości – oferta, czyli kwotę jaką zaoferował klient 1 oraz wygrana, czyli kwotę, która okazała się najwyższa. W momencie, gdy oferta jest większa lub równa wartości wygranej oznacza to, że wygrał agent reprezentujący klienta 1. Biorąc pod uwagę strategię pozostałych graczy wygrana agenta 1 ma miejsce wyłącznie w przypadku, gdy jego maksimum jest wyższe niż maksimum pozostałych graczy. Sytuacja ta wynika z założenia, w którym każdy agent zagra kwotą maksymalną.

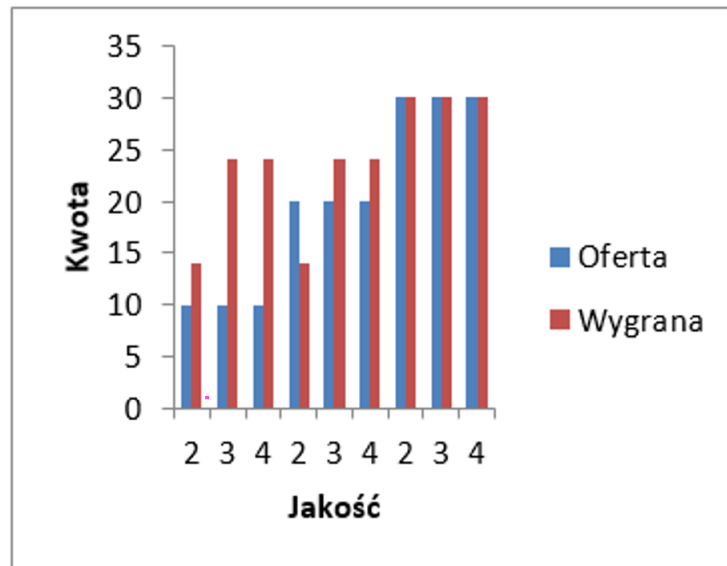
Dla strategii 2 przyjęto, że klienci poza maksymalnymi kwotami zainteresowani są również konkretną jakością transmisji, która nie musi być dla nich identyczna. Klient 1, dla którego rozważany jest wynik gry, w przypadku strategii 2 ma zdefiniowany próg akceptacji jakości transmisji  $\pm 1$ . Dla tych parametrów przyjęte strategię zaprezentowane zostały w tabeli 4.5.

Wyniki dla strategii 2 przedstawiono na rysunku 4.32. Również i w tym przypadku dokonano trzech pomiarów w zależności od maksymalnych kwot akceptowanych przez klienta 1. Dla każdego pomiaru dokonano analizy sytuacji w zależności od oczekiwanej jakości ( $3 \pm 1$ ). W momencie, gdy maksymalna kwota wyniesie

RYSUNEK 4.31: Wynik dla strategii 1, aukcja *first-price*

| Klient | Kwota | Jakość |
|--------|-------|--------|
| K2     | 4     | 1      |
| K3     | 4     | 2      |
| K5     | 14    | 2      |
| K4     | 4     | 3      |
| K6     | 14    | 3      |
| K8     | 24    | 3      |
| K7     | 14    | 4      |
| K9     | 24    | 4      |
| K10    | 24    | 5      |

TABELA 4.5: Oponenci w przypadku strategii 2

RYSUNEK 4.32: Wynik dla strategii 1, aukcja *first-price*

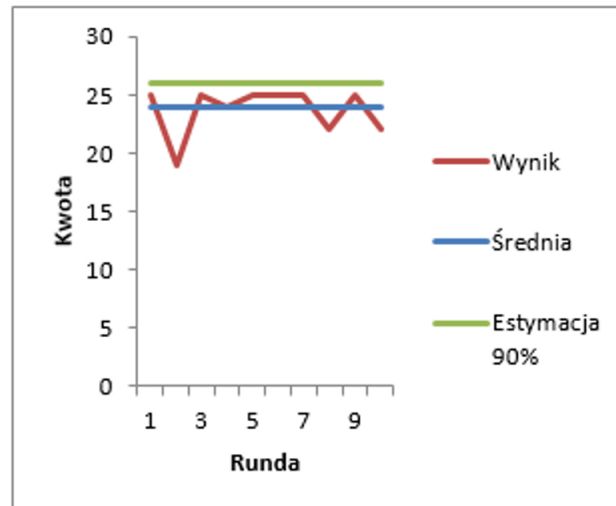
10, agent nie wygra aukcji mimo progu akceptacji jakości transmisji. Z kolei, gdy maksymalna kwota wyniesie 20 to dla jakości równej 2 agent wygra aukcję, dla pozostałych przegra. Gdy przyjęta zostanie maksymalna kwota 30 agent wygra dla każdej wartości jakości transmisji i zapłaci tą kwotę. W związku z tym agent grając kwotą wysoką, powinien starać się uzyskać jak najlepszą jakość.

### 4.3.2 Wielokrotne aukcje jednego dobra

Dla strategii 3 i przypadku posiadania częściowej wiedzy, dotyczącej przebiegu dotychczasowych aukcji przyjęto, iż zakup możliwy jest w 10 rundach, przy czym przez 9 rund agent 1 obserwuje oferty, a następnie w 10 rundzie zagra kwotą większą od stwierdzonej średniej, lecz mniejszą od maksimum ustalonego przez klienta (wariant 1) lub taką kwotą, dzięki której wygrałby 90% przeprowadzanych dotychczas aukcji (wariant 2).

$$\frac{1}{n} * \sum_{i=1}^n x_i < Of(q) \leq \max(c_1(q)) \quad (4.1)$$

Metoda działania agenta 1 w przypadku stosowania wariantu 1 przedstawiona została na nierówności 4.1. Oferta uzależniona jest od jakości –  $Of(q)$ . Jest większa od obliczonej średniej i mniejsza lub równa maksimum akceptowanego przez klienta dla danej jakości transmisji.

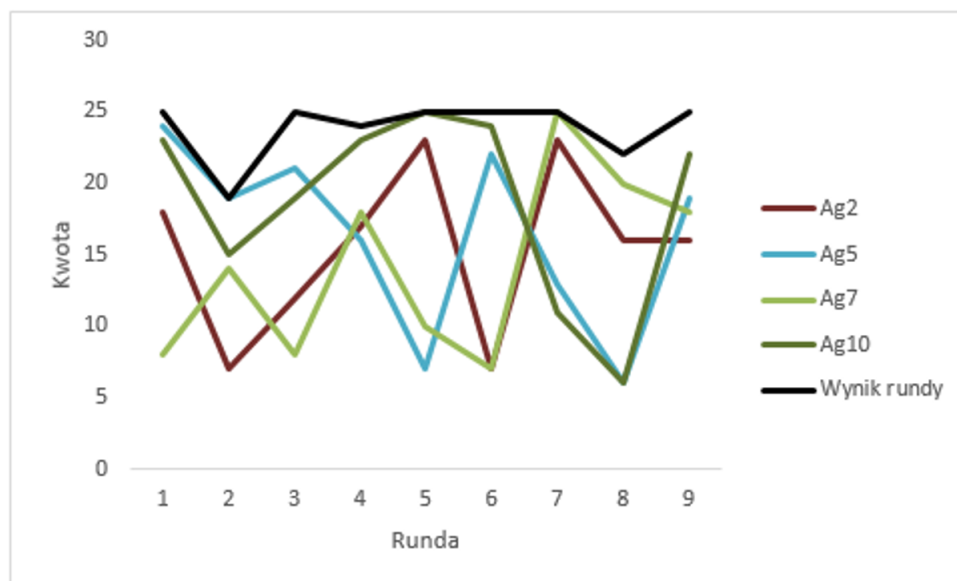


RYSUNEK 4.33: Wyniki dla strategii 3, *first-price* - częściowa informacja

Przebieg poszczególnych rund wraz z naniesioną wartością średnią, zaobserwowaną przez agenta po 9 rundach oraz estymacją wartości, dzięki której agent wygrałby 90% przeprowadzonych aukcji przedstawiono na rysunku 4.33. Dla strategii, w której agent zagra wartością nie mniejszą niż średnia z dotychczasowych gier zauważyć można, iż dla przyjętych strategii pozostałych graczy, gdy agent 1 w rundzie 10 zagra co najmniej kwotą równą stwierdzonej średniej wartości, to wówczas wygra tę rundę. Wynika to z faktu, iż średnia wynosi 23,89, a maksymalna oferta w rundzie 10 wyniosła 22. Jednakże agent 1 nie ma pewności, że jego oferta będzie najwyższą, ponieważ strategie pozostałych graczy mogą zmieniać się w kolejnych rundach. W związku z tym innym rozwiązaniem jest złożenie takiej oferty, dzięki której agent 1 wygrałby 90% przeprowadzonych do tej pory aukcji. Zgodnie z tym podejściem, agent 1 zagrałby kwotą 26. Przyjmując znany przebieg rundy 10 (wynik 22) agent 1 składając ofertę 26 wygrałby ją, lecz zapłaciłby więcej niż w przypadku strategii z wartością średnią. Z kolei, gdyby wynik rundy 10 był większy od wartości średniej to agent wygrałby grając wartością z wariantu 2.

Dla strategii 3 z założeniem, iż agent posiada pełną wiedzę dotyczącą wszystkich przeprowadzonych dotychczas aukcji i ofert składanych przez konkurentów założono, że w 9 rundach udział brały wszystkie agenty. W rundzie 10 udział, poza agentem 1, wezmą również agenty 2, 5, 7 i 10. Dzięki pełnej wiedzy dotyczącej składanych ofert agent 1 jest w stanie określić, z pewnym prawdopodobieństwem, jaka oferta złożona zostanie przez poszczególne agenty w kolejnej rundzie. Na rysunku 4.34 przedstawiono oferty składane przez agenty w kolejnych rundach oraz wynik każdej aukcji.

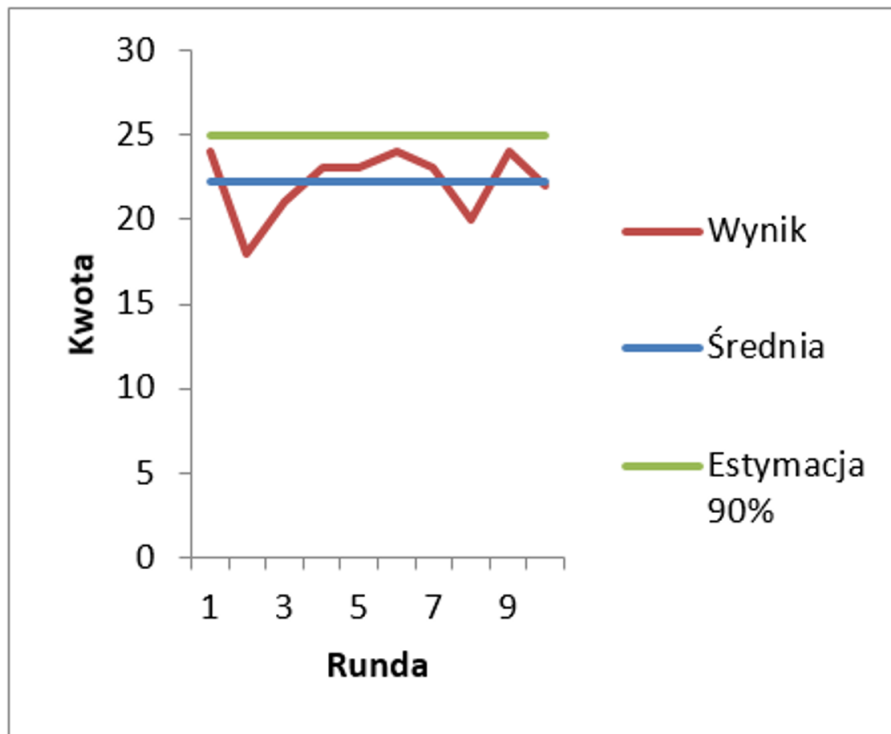
W oparciu o posiadaną wiedzę, po 9 rundach zauważyć można, iż agent 10 złożył ofertę maksymalną 25 i minimalną 6, agent 2 maksymalną 23 i minimalną 7,

RYSUNEK 4.34: Wyniki dla strategii 3, *first-price* - pełna informacja

agent 5 maksimum 24 i minimum 6, agent 7 maksimum 25 i minimum 7. Na tej podstawie można przyjąć, że z dużym prawdopodobieństwem wartości maksymalne ze złożonych ofert stanowią progi wartości dla poszczególnych agentów. W związku z tym, jeśli agent 1 w rundzie 10 chce wygrać, powinien złożyć ofertę nie mniejszą niż maksimum z maksymalnych ofert poszczególnych konkurentów. W tym przypadku maksimum to 25 i w związku z tym agent 1 powinien złożyć ofertę wyższą od tej wartości, lecz nie powinna być ona dużo wyższa, aby nie przepłacił. W ten sposób agent 1 jest w stanie uczyć się zachowań pozostałych graczy oraz, w oparciu o tą wiedzę, podjąć decyzję, która pozwoli uzyskać oczekiwany rezultat. Im większą liczbę aukcji obserwuje agent tym bardziej dokładna będzie jego wiedza. Rozwinięcie tej koncepcji może stanowić analiza częstości składania ofert o określonej wysokości oraz określenie prawdopodobieństwa złożenia danej oferty w kolejnej rundzie.

### 4.3.3 Aukcje drugiej ceny

Przeprowadzono szereg eksperymentów dla rozważanych strategii dla aukcji drugiej ceny. W szczególności w przypadku strategii 3 i posiadania częściowej wiedzy dotyczącej przebiegu dotychczasowych aukcji, wynik średni z 9 gier wyniesie 22,22 i w związku z tym agent 1 w rundzie 10 powinien zaoferować nie mniej niż tą wartość. Ponieważ zapłaci drugą co do wartości kwotę to powinien zagrać możliwie wysoko. W rundzie 10 drugi co do wartości wynik to 22, a co za tym idzie klient 1 niezależnie o ile wyżej zagra zapłaci 22. Wyniki dla tej strategii oraz przyjętych założeń przedstawiono na rysunku 4.35. Dla wariantu 2 agent powinien zagrać nie



RYSUNEK 4.35: Wyniki dla strategii 3, *second-price* - pełna informacja

mniej niż 25, aby wygrać w 90% aukcji. Wartość ta stanowi wartość minimalną, ponieważ dokładnie nie wiadomo, jakie oferty złożyły pozostałe agenty w poszczególnych rundach – znana jest tylko końcowa kwota stanowiąca drugą co do wartości. W związku z tym agent 1 powinien zagrać możliwie wysoko i z dużym prawdopodobieństwem nie zapłaci więcej niż 25 w rundzie, w której weźmie udział.

#### 4.3.4 Aukcje kombinatoryczne

Aukcje tego typu dotyczą więcej niż jednego dobra równocześnie. W przypadku usługi, jaką jest transmisja danych mowa o parametrach dotyczących jakości, takimi jak przepływność, opóźnienia transmisji, fluktuacja opóźnienia, współczynnik utraty pakietów. Każdy z tych parametrów rozumiany jest jako oddzielne dobro, a ich kombinacja jest podstawą do złożenia oferty. Ogólna zasada dla aukcji kombinatorycznych (dotyczących jakości transmisji) z wykorzystaniem operacji logicznych przedstawiona została na równaniu 4.2.  $Op$  oznacza operator logiczny – na potrzeby badań XOR lub OR,  $q$  to oznaczenie parametru jakościowego, a  $amt_m$  oznacza ofertę dla danej kombinacji jakości.

$$Of = (\{q_1, q_2, \dots, q_n\}, amt_1)OP \dots OP(\{q_1, q_2, \dots, q_n\}, amt_m) \quad (4.2)$$

W badaniach przyjęto, że parametr  $q_1$  oznacza przepływność,  $q_2$  opóźnienie transmisji, a  $q_3$  to współczynnik utraty pakietów. Klient 1 jest w stanie zapłacić 15 w momencie, gdy parametry przyjmą wartości  $\{50 \text{ Mb/s}, 4\text{ms}\}$ , 25 dla  $\{100 \text{ Mb/s}, 2\text{ms}, 0\%\}$  i 10, gdy gwarantowana jest wyłącznie przepływność na poziomie  $\{50 \text{ Mb/s}\}$ . Strategia ta została zaprezentowana na równaniu 4.3. Dla przyjętych założeń operatorem może być OR lub XOR. W przypadku operatora OR dana strategia nie ma większego uzasadnienia, ponieważ oznaczałoby to, że użytkownik zapłaci 50 w momencie, gdy otrzyma wszystkie parametry równocześnie. Jeśli użytkownik otrzyma parametry  $\{100, 2, 0\}$  to nie jest uzasadnionym płacenie za gorsze parametry. W związku z tym odpowiednim jest wykorzystanie operatora XOR. W takiej sytuacji użytkownik zapłaci maksymalną kwotę, jeśli spełnione są jego wymagania.

$$Of = (\{50, 4\}, 15)OP(\{100, 2, 0\}, 25)OP(\{50\}, 10) \quad (4.3)$$

Niestety parametry jakościowe ciężko jest zapewnić na stałym poziomie. Zawsze pojawiają się zmiany, które mogą nie mieć większego wpływu na końcową jakość transmisji. W związku z tym dla każdego parametru powinien zostać zdefiniowany próg akceptacji. Oznacza to, że parametry mieszczące się w danym przedziale są nadal akceptowalne dla klienta. W najprostszym przypadku klient zdefiniuje próg akceptacji dla każdego parametru. W bardziej złożonym przypadku możliwym jest zdefiniowanie przedziałów akceptacji dla każdej kwoty z osobna. Kolejnym możliwym rozwiązaniem jest zdefiniowanie, iż poszczególne wartości parametrów stanowią wartości minimalne lub maksymalne. Dla powyższego przykładu przyjęć można, iż klient określa minimalną przepływność, maksymalne opóźnienie transmisji oraz maksymalny współczynnik utraty pakietów. Podejście to może być mało zrozumiałe dla klienta więc rozważa się zastosowanie innych metod upraszczających translację parametrów, np. uczenie maszynowe.

## 4.4 **Uczenie maszynowe**

Badania podzielono na trzy etapy:

1. Etap I - zastosowanie klasycznych metod klasyfikacji,
2. Etap II - własne algorytmy z wykorzystaniem uczenia zespołowego,
3. Etap III - zastosowanie fuzji sieci neuronowej i algorytmu genetycznego.

We wszystkich etapach jako miarę porównania jakości procesu klasyfikacji przyjęto:

- ewaluację liczby błędów w poszczególnych klasach (macierze pomyłek) [209],

- ogólny współczynnik dokładności (z ang. sensitivity, SEN) [209]. Ponieważ problem jest wieloklasowy koniecznym było wykorzystanie SEN, czyli ogólnego współczynnika dokładności ( $O_{ACC}$ ). SEN wyliczono w następujący sposób:

$$SEN = \frac{\sum TP}{\sum TP + \sum FN} \quad (4.4)$$

gdzie:

- SEN - ogólny współczynnik dokładności,
- TP - liczba przypadków prawdziwie pozytywnych,
- FN - liczba przypadków fałszywie negatywnych.

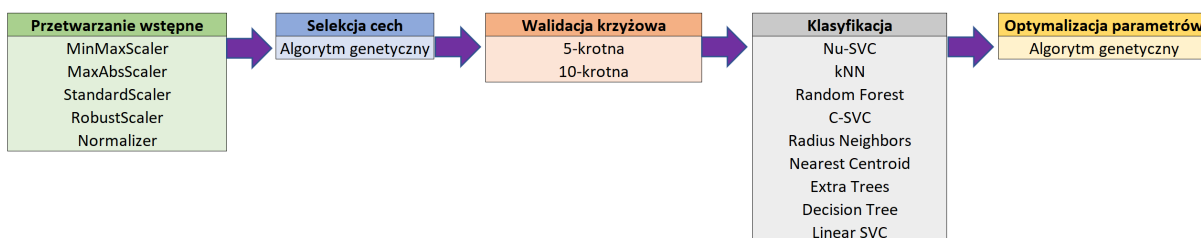
Dodatkowo w przypadku etapu II wykorzystano F1\_Score jako funkcję adaptacji [210, 211]. Z kolei w etapie III zweryfikowano zmienność błędu średniokwadratowego dla różnych epok algorytmu genetycznego. Na podstawie tych parametrów dokonano oceny poszczególnych metod klasyfikacji.

#### 4.4.1 Etap I

Pierwsze badania przeprowadzono z wykorzystaniem klasycznych metod klasyfikacji oraz *Stacking Classifier*. Wykorzystano bibliotekę Sklearn oraz Deap. Jako język programowania wykorzystano Python. Proces uczenia maszynowego podstawowych klasyfikatorów przedstawiony został na rysunku 4.36, a *Stacking Classifier* na rysunku 4.37. W obu przypadkach proces podzielony został na pięć kroków:

1. Przetwarzanie wstępne - wykorzystywane, aby uzyskać dane w określonych przedziałach wartości. Zastosowano następujące metody: *MinMaxScaler*, *MaxAbsScaler*, *StandardScaler*, *RobustScaler* oraz *Normalizer*.
2. Selekcja cech - cechy wybrane zostały przy użyciu algorytmu genetycznego (z ang. genetic algorithm, GA). Cechy to parametry opisujące jakość transmisji tj. opóźnienie transmisji, przepływność, fluktuację opóźnienia oraz współczynnik utraty pakietów. Z całego zbioru parametrów GA wybrał te, które są istotne w procesie ML. Etap ten jest ważny, ponieważ umożliwia eliminację cech, które nie są istotne z punktu widzenia uczenia klasyfikatorów. Geny w przeprowadzonych badaniach wykorzystują wartości 0 i 1, gdzie 0 oznacza, że cecha może zostać pominięta, a 1 oznacza cechę, która musi zostać wykorzystana.
3. Walidacja krzyżowa - wykorzystano dwa rodzaje walidacji krzyżowej (z ang. cross validation, CV) - 10-krotną i 5-krotną. W przypadku 10-krotnej CV utworzono 10 kombinacji danych testowych i treningowych, z kolei w przypadku





RYSUNEK 4.36: Proces ML - Etap I

5-krotnej CV dostępnych było 5 kombinacji. Obie metody walidacji krzyżowej przetestowane zostały w kombinacji z wszystkimi możliwymi metodami przetwarzania wstępnego oraz klasyfikacji. W przypadku *Stacking Classifier* wykorzystano tylko 10-krotną CV.

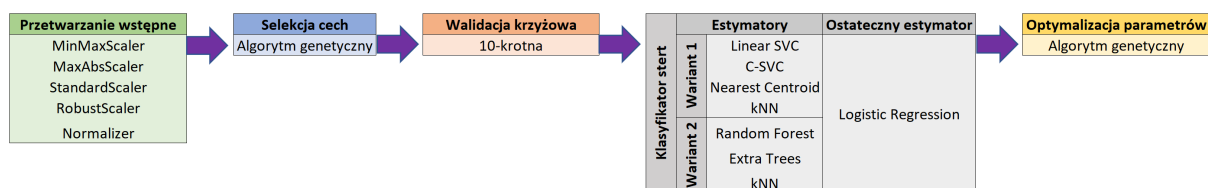
#### 4. Klasyfikacja - przetestowano 9 typów klasyfikatorów:

- *Nu-SVC* [212, 213, 214],
- *kNN* [215],
- *Random Forest* [216],
- *Radius Neighbors* [215],
- *Nearest Centroid* [217],
- *Extra Trees* [218],
- *Linear SVC* [219],
- *C-SVC* [212, 213, 214],
- *Decision Tree* [220].

W tabeli 4.6 przedstawiono przedziały wartości parametrów wykorzystanych w przypadku poszczególnych klasyfikatorów. Na potrzeby prezentowanych badania wykorzystano również *Stacking Classifier* [221]. Ten typ klasyfikatora przetestowany został z dwoma wariantami kombinacji estymatorów. Ostatecznym estymatorem było *Logistic Regression* [222] we obu wariantach. Estymatory oraz ich parametry zaprezentowane zostały w tabeli 4.7.

#### 5. Optymalizacja parametrów - do optymalizacji parametrów wykorzystano algorytm genetyczny. Parametry algorytmu genetycznego przedstawiono w tabeli 4.8.

Wszystkie wyniki przedstawione zostały dla testowego zbioru danych. Badania podzielono na dwa przypadki: (1) każdy z dziewięciu klasyfikatorów opisanych w tabeli 4.6 przetestowany został dla dwóch rozmiarów populacji i dwóch wielkości generacji (100 i 1000), pięciu algorytmów przetwarzania wstępnego oraz dwóch

RYSUNEK 4.37: Proces ML - *Stacking Classifier*

| Klasyfikator     | Parametr 1                          | Parametr 2                                  | Parametr 3                                  | Parametr 4                 |
|------------------|-------------------------------------|---|---|----------------------------|
| Nu-SVC           | kernel=(linear, rbf, poly, sigmoid) | nu=(0.001 - 0.5)                            | degree=(1 - 5)                              | gamma=(0.001 - 2)          |
| kNN              | neighbors=(1 - 7)                   | weights=(uniform, distance)                 | algorithm=(ball_tree, kd_tree, brute, auto) | leaf_size=(15 - 45)        |
| Random Forest    | n_estimators=(50 - 200)             | max_depth=(1 - 20)                          | random_state=(0 - 20)                       | max_samples=(1 - 80)       |
| C-SVC            | kernel=(linear, rbf, poly, sigmoid) | coef0=(0 - 0.5)                             | degree=(1 - 6)                              | gamma=(0.001 - 2)          |
| Radius Neighbors | radius=(5 - 15)                     | algorithm=(auto, ball_tree, kd_tree, brute) | leaf_size=(15 - 45)                         | wights=(uniform, distance) |
| Nearest Centroid | -                                   | -   | -   | -                          |
| Extra Trees      | n_estimators=(50 - 200)             | -   | -   | -                          |
| Decision Tree    | -                                   | -   | -   | -                          |
| Linear SVC       | random_state=(0 - 50)               | tol=(1e-6 - 1e-4)                           | c=(0.5 - 2.0)                               | -                          |

TABELA 4.6: Parametry klasyfikatorów etapu I

|                  | Estymator        | Parametr 1                          | Parametr 2                  | Parametr 3                                  | Parametr 4           |
|------------------|------------------|-------------------------------------|-----------------------------|---|----------------------|
| <b>Wariant 1</b> | Linear SVC       | random_state(0 - 50)                | tol=(1e-6 - 1e-4)           | c=(0.5 - 2.0)                               |                      |
|                  | C-SVC            | kernel=(linear, rbf, poly, sigmoid) | coef0=(0 - 0.5)             | degree=(1 - 6)                              | gamma=(0.001 - 2)    |
|                  | Nearest Centroid | -                                   | -                           | -   | -                    |
|                  | kNN              | neighbors=(1 - 7)                   | weights=(uniform, distance) | algorithm=(ball_tree, kd_tree, brute, auto) | leaf_size=(15 - 45)  |
| <b>Wariant 2</b> | Random Forest    | n_estimators=(50 - 200)             | max_depth=(1 - 20)          | random_state=(0 - 20)                       | max_samples=(1 - 64) |
|                  | Extra Trees      | n_estimators=(50 - 200)             | -                           | -   | -                    |
|                  | kNN              | neighbors=(1 - 7)                   | weights=(uniform, distance) | algorithm=(ball_tree, kd_tree, brute, auto) | leaf_size=(15 - 45)  |

TABELA 4.7: Parametry *Stacking Classifier*

| Parametr                       | Wartość                |
|--------------------------------|------------------------|
| Rozmiar populacji              | 100 lub 1000           |
| Liczba generacji               | 100 lub 1000           |
| Prawdopodobieństwo krzyżowania | 0,8                    |
| Prawdopodobieństwo mutacji     | 0,2                    |
| Algorytm selekcji              | tournament (rozmiar=3) |
| Funkcja dopasowania            | Accuracy               |

TABELA 4.8: Parametry algorytmu genetycznego (GA) - etap I

metod walidacji krzyżowej. Takie podejście pozwoliło uzyskać 180 różnych kombinacji; (2) *Stacking Classifier* przetestowany został dla dwóch wariantów. Dla każdego wariantu różna kombinacja estymatorów została wykorzystana (tabela 4.7). Każdy z wariantów *Stacking Classifier* przetestowany został dla rozmiaru populacji i wielkości generacji 100, pięciu metod przetwarzania wstępnego oraz 10-krotnej walidacji krzyżowej. Takie podejście pozwoliło na uzyskanie 10 różnych kombinacji. Zaprezentowano trzy najlepsze uzyskane wyniki.

Wyniki dla klasyfikatora *Nu-SVC* zaprezentowane zostały w tabeli 4.9. Najwyższa uzyskana wartość SEN to 87,00%. Wynik ten uzyskany został po wyeliminowaniu jednej cechy - współczynnika utraty pakietów. Cecha ta została uznana przez GA za nieistotną. Jest to bardzo ciekawe działanie. Pozostałe parametry to 10-krotna walidacja krzyżowa oraz *MinMaxScaler* jako metoda przetwarzania wstępnego. Trzy najlepsze wyniki dla klasyfikatora *Random Forest* przedstawiono w tabeli 4.10. Uzyskano SEN o wartości 87,00% przy uwzględnieniu wszystkich cech, 10-krotnej walidacji krzyżowej oraz przetwarzaniu wstępnym *MaxAbsScaler*. Kolejnym klasyfikatorem było *C-SVC* (tabela 4.11). W przypadku tego klasyfikatora najlepszym wynikiem było SEN=83,00%. Wynik ten uzyskano dla trzech cech - opóźnienie transmisji, fluktuacja opóźnienia oraz przepływność. Także w tym przypadku GA wyeliminował współczynnik utraty pakietów jako nieistotną cechę. Wynik ten uzyskano dla 10-krotnej walidacji krzyżowej oraz przetwarzania wstępnego *StandardScaler*. Algorytm *kNN* uzyskał najlepszy wynik SEN=84,00% (tabela 4.12). W tym przypadku GA nie wyeliminował żadnej cechy. Wspomniany wynik uzyskany został dla 10-krotnej walidacji krzyżowej. Wynik był taki sam dla algorytmów przetwarzania wstępnego *MaxAbsScaler* oraz *MinMaxScaler*. W tabeli 4.13 przedstawiono trzy najlepsze wyniki dla klasyfikatora *Decision Tree*. Dla wszystkich SEN było takie samo tj. 84,00%. Bardzo ciekawa była eliminacja dwóch cech dokonana przez GA - opóźnienie transmisji oraz współczynnik utraty pakietów. Kolejnym klasyfikatorem był *Nearest Centroid*, który uzyskał SEN=81,00%. Wynik ten uzyskany został dla 10-krotnej walidacji krzyżowej, algorytmu przetwarzania wstępnego *Normalizer* i przy eliminacji jednej cechy - opóźnienia transmisji, jako nieistotnej (tabela 4.14). W tabeli 4.15 przedstawiono wyniki klasyfikatora *Radius Neighbors*. Najwyższa uzyskana wartość SEN to 83,00%. Uzyskano ją przy uznaniu wszystkich cech jako istotne, 10-krotnej walidacji krzyżowej oraz *Normalizer* jako metody przetwarzania wstępnego. Wyniki klasyfikatora *Linear SVC* były najgorszymi jakie uzyskano - SEN=75,00% (tabela 4.16). Ostatnim klasyfikatorem był *Extra Trees*. Wyniki zaprezentowano w tabeli 4.17. Klasyfikator ten uzyskał najlepszy wynik SEN=85,00% przy uznaniu wszystkich cech za istotne, 10-krotnej walidacji krzyżowej. Dokładnie taki sam wynik uzyskano przy zastosowaniu algorytmu przetwarzania wstępnego *Normalizer*,

| Walidacja | Przeskalowanie | Kernel | Nu          | Degree | Gamm        | Opóźnienie | Fluktuacja | Przepustowość | Strata pakietów | O_ACC = SEN |
|-----------|----------------|--------|-------------|--------|-------------|------------|------------|---------------|-----------------|-------------|
| 10        | MinMaxScaler   | rbf    | 0,167114060 | 4      | 0,016614809 | 1          | 1          | 1             | 0               | 87,00%      |
| 10        | StandardScaler | poly   | 0,219859266 | 1      | 0,746559689 | 1          | 1          | 1             | 1               | 85,00%      |
| 5         | MinMaxScaler   | poly   | 0,068178494 | 1      | 1,197786714 | 1          | 1          | 1             | 0               | 85,00%      |

TABELA 4.9: Trzy najlepsze wyniki klasyfikatora *Nu-SVC*

| Walidacja | Przeskalowanie | N estimators | Max depth | Random state | Max samples | Opóźnienie | Fluktuacja | Przepustowość | Strata pakietów | O_ACC = SEN |
|-----------|----------------|--------------|-----------|--------------|-------------|------------|------------|---------------|-----------------|-------------|
| 10        | MaxAbsScaler   | 149          | 4         | 0            | 79          | 1          | 1          | 1             | 1               | 87,00%      |
| 10        | MinMaxScaler   | 164          | 5         | 14           | 72          | 1          | 1          | 1             | 1               | 86,00%      |
| 5         | MaxAbsScaler   | 78           | 17        | 16           | 18          | 1          | 1          | 1             | 1               | 84,00%      |

TABELA 4.10: Trzy najlepsze wyniki klasyfikatora *Random Forest*

*StandardScaler* i *MinMaxScaler*.

Kolejnym krokiem badań było przetestowanie *Stacking Classifier* w obu przyjętych wariantach. W tabeli 4.18 zaprezentowano wyniki uzyskane dla wariantu 1. Najwyższa uzyskana wartość SEN to 89,00%. Jest to najlepszy wynik spośród wszystkich do tej pory przetestowanych klasyfikatorów. Wynik ten uzyskano przy eliminacji jednej cechy, uznanej za nieistotną - współczynnika utraty pakietów. Zastosowano algorytm przetwarzania wstępnego *StandardScaler*. Wyniki dla wariantu 2 *Stacking Classifier* przedstawiono w tabeli 4.19. W tym wypadku najwyższa wartość SEN to 87,00% uzyskana przy zastosowaniu *StandardScaler* jako algorytmu przetwarzania wstępnego.

Bardzo istotną kwestią było zweryfikowanie jak w poszczególnych klasach poradziły sobie poszczególne klasyfikatory. Ponieważ problem jest bardziej złożony niż prosty przypadek binarny, należało przygotować ogólne macierze pomyłek oraz rozszerzoną informację dotyczącą każdej klasy (ocena jakości transmisji). Na rysunkach 4.38-4.48 przedstawiono miary: prawdziwie pozytywną (z ang. True Positive, TP), fałszywie pozytywną (z ang. False Positive, FP), fałszywie negatywną (z ang. False Negative, FN) oraz prawdziwie negatywną (z ang. True Negative, TN). Dzięki zaprezentowanym informacjom możliwe było zaobserwowanie dla których klas klasyfikatory radziły sobie lepiej, a dla których gorzej. Wnioski są następujące:

1. *Nu-SVC* uzyskało najwyższe SEN dla klasy 2 (95,00%), a najniższe dla klasy 5 (78,95%), również dla klasy 1 SEN było niższe niż ogólny współczynnik dokładności (rysunek 4.38).
2. Klasyfikator *Random Forest* wykazał się najwyższą skutecznością działania dla klasy 3 (SEN=96,15%). Najgorszy wynik uzyskał dla klasy 4, to jest 75,00% (rysunek 4.39).

| Walidacja | Przeskalowanie | Kernel | Coef0       | Degree | Gamma       | Opóźnienie | Fluktuacja | Przepustowość | Strata pakietów | O_ACC = SEN |
|-----------|----------------|--------|-------------|--------|-------------|------------|------------|---------------|-----------------|-------------|
| 10        | StandardScaler | poly   | 0,115410283 | 6      | 1,910356008 | 0          | 1          | 1             | 0               | 83,00%      |
| 5         | Normalizer     | poly   | 0,190109674 | 6      | 1,730517548 | 0          | 1          | 1             | 1               | 82,00%      |
| 10        | Normalizer     | poly   | 0,374051492 | 3      | 2,376059635 | 1          | 1          | 1             | 1               | 82,00%      |

TABELA 4.11: Trzy najlepsze wyniki klasyfikatora *C-SVC*

| Walidacja | Przeskalowanie | Neighbors | Weights | Algorithm | Leaf Size | Opóźnienie | Fluktuacja | Przepustowość | Strata pakietów | O_ACC = SEN |
|-----------|----------------|-----------|---------|-----------|-----------|------------|------------|---------------|-----------------|-------------|
| 10        | MaxAbsScaler   | 3         | uniform | brute     | 15        | 1          | 1          | 1             | 1               | 84,00%      |
| 10        | MinMaxScaler   | 3         | uniform | brute     | 34        | 1          | 1          | 1             | 1               | 84,00%      |
| 10        | RobustScaler   | 4         | uniform | brute     | 44        | 1          | 1          | 1             | 0               | 82,00%      |

TABELA 4.12: Trzy najlepsze wyniki algorytmu *kNN*

| Walidacja | Przeskalowanie | Opóźnienie | Fluktuacja | Przepustowość | Strata pakietów | O_ACC = SEN |
|-----------|----------------|------------|------------|---------------|-----------------|-------------|
| 10        | StandardScaler | 0          | 1          | 1             | 0               | 84,00%      |
| 10        | MinMaxScaler   | 0          | 1          | 1             | 0               | 84,00%      |
| 10        | MaxAbsScaler   | 0          | 1          | 1             | 0               | 84,00%      |

TABELA 4.13: Trzy najlepsze wyniki klasyfikatora *Decision Tree*

| Walidacja | Przeskalowanie | Opóźnienie | Fluktuacja | Przepustowość | Strata pakietów | O_ACC = SEN |
|-----------|----------------|------------|------------|---------------|-----------------|-------------|
| 10        | Normalizer     | 0          | 1          | 1             | 1               | 81,00%      |
| 5         | Normalizer     | 0          | 1          | 1             | 1               | 81,00%      |
| 10        | MinMaxScaler   | 1          | 1          | 1             | 1               | 70,00%      |

TABELA 4.14: Trzy najlepsze wyniki klasyfikatora *Nearest Centroid*

| Walidacja | Przeskalowanie | Radius       | Algorithm | Leaf size | Weights  | Opóźnienie | Fluktuacja | Przepustowość | Strata pakietów | O_ACC = SEN |
|-----------|----------------|--------------|-----------|-----------|----------|------------|------------|---------------|-----------------|-------------|
| 10        | Normalizer     | 9,222858104  | ball_tree | 43        | distance | 1          | 1          | 1             | 1               | 83,00%      |
| 5         | Normalizer     | 7,781229291  | auto      | 15        | distance | 1          | 1          | 1             | 1               | 80,00%      |
| 10        | StandardScaler | 12,609341221 | ball_tree | 42        | distance | 0          | 1          | 1             | 0               | 77,00%      |

TABELA 4.15: Trzy najlepsze wyniki klasyfikatora *Radius Neighbors*

| Walidacja | Przeskalowanie | Random state | Tol         | C           | Opóźnienie | Fluktuacja | Przepustowość | Strata pakietów | O_ACC = SEN |
|-----------|----------------|--------------|-------------|-------------|------------|------------|---------------|-----------------|-------------|
| 10        | Normalizer     | 2            | 8,70323E-05 | 1,697960635 | 0          | 1          | 1             | 1               | 75,00%      |
| 10        | Normalizer     | 28           | 1,17533E-05 | 1,838197728 | 0          | 1          | 1             | 1               | 75,00%      |
| 5         | Normalizer     | 50           | 6,45777E-05 | 1,371446259 | 0          | 1          | 1             | 1               | 73,00%      |

TABELA 4.16: Trzy najlepsze wyniki klasyfikatora *Linear SVC*

| Walidacja | Przeskalowanie | N estimators | Opóźnienie | Fluktuacja | Przepustowość | Strata pakietów | O_ACC = SEN |
|-----------|----------------|--------------|------------|------------|---------------|-----------------|-------------|
| 10        | Normalizer     | 195          | 1          | 1          | 1             | 1               | 85,00%      |
| 10        | StandardScaler | 94           | 1          | 1          | 1             | 1               | 85,00%      |
| 10        | MinMaxScaler   | 93           | 1          | 1          | 1             | 1               | 85,00%      |

TABELA 4.17: Trzy najlepsze wyniki klasyfikatora *Extra Trees*

| Walidacja | Przeskalowanie | Estymator        | Parametr 1      | Parametr 2        | Parametr 3          | Parametr 4        | Opóźnienie | Fluktuacja | Przepustowość | Strata pakietów | O_ACC = SEN |
|-----------|----------------|------------------|-----------------|-------------------|---------------------|-------------------|------------|------------|---------------|-----------------|-------------|
| 10        | StandardScaler | Linear SVC       | random_state=44 | tol=2,49343E-05   | C=1,104396883       | -                 | 1          | 1          | 1             | 0               | 89,00%      |
|           |                | C-SVC            | kernel='poly'   | coef0=0,363466800 | degree=4            | gamma=1,148169838 |            |            |               |                 |             |
|           |                | Nearest Centroid | -               | -                 | -                   | leaf size=39      |            |            |               |                 |             |
| 10        | MinMaxScaler   | KNN              | neighbors=6     | weights='uniform' | algorithm='brute'   | -                 | 1          | 1          | 1             | 1               | 87,00%      |
|           |                | Linear SVC       | random_state=28 | tol=8,47317E-05   | C=1,192405108       | -                 |            |            |               |                 |             |
|           |                | C-SVC            | kernel='poly'   | coef0=0,151722215 | degree=5            | gamma=1,315051318 |            |            |               |                 |             |
| 10        | RobustScaler   | Nearest Centroid | -               | -                 | -                   | -                 | 1          | 1          | 1             | 1               | 86,00%      |
|           |                | KNN              | neighbors=2     | weights='uniform' | algorithm='auto'    | leaf size=45      |            |            |               |                 |             |
|           |                | Linear SVC       | random_state=3  | tol=3,77272E-05   | C=0,857679318       | -                 |            |            |               |                 |             |
| 10        | RobustScaler   | C-SVC            | kernel='rbf'    | coef0=0,288858698 | degree=3            | gamma=0,285172185 | 1          | 1          | 1             | 1               | 86,00%      |
|           |                | Nearest Centroid | -               | -                 | -                   | -                 |            |            |               |                 |             |
|           |                | KNN              | neighbors=5     | weights='uniform' | algorithm='kd_tree' | leaf size=25      |            |            |               |                 |             |

TABELA 4.18: Trzy najlepsze wyniki klasyfikatora *Stacking Classifier* w wariancie 1

| Walidacja | Przeskalowanie | Estymator     | Parametr 1     | Parametr 2         | Parametr 3          | Parametr 4     | Opóźnienie | Fluktuacja | Przepustowość | Strata pakietów | O_ACC = SEN |
|-----------|----------------|---------------|----------------|--------------------|---------------------|----------------|------------|------------|---------------|-----------------|-------------|
| 10        | StandardScaler | Random Forest | estimators=99  | max depth=12       | random state=16     | max samples=54 | 1          | 1          | 1             | 1               | 87,00%      |
|           |                | Extra Trees   | estimators=92  | -                  | -                   | -              |            |            |               |                 |             |
|           |                | KNN           | neighbors=4    | weights='uniform'  | algorithm='kd_tree' | leaf size=34   |            |            |               |                 |             |
| 10        | MinMaxScaler   | Random Forest | estimators=100 | max depth=19       | random state=0      | max samples=45 | 1          | 1          | 1             | 0               | 86,00%      |
|           |                | Extra Trees   | estimators=50  | -                  | -                   | -              |            |            |               |                 |             |
|           |                | KNN           | neighbors=3    | weights='uniform'  | algorithm='auto'    | leaf size=37   |            |            |               |                 |             |
| 10        | Normalizer     | Random Forest | estimators=156 | max depth=10       | random state=13     | max samples=59 | 1          | 1          | 1             | 1               | 85,00%      |
|           |                | Extra Trees   | estimators=170 | -                  | -                   | -              |            |            |               |                 |             |
|           |                | KNN           | neighbors=4    | weights='distance' | algorithm='auto'    | leaf size=43   |            |            |               |                 |             |

TABELA 4.19: Trzy najlepsze wyniki klasyfikatora *Stacking Classifier* w wariancie 2

3. Klasyfikator *C-SVC* poradził sobie najlepiej z klasą 3, uzyskując  $SEN=92,31\%$ , a najgorzej z klasą 4 ( $70,00\%$ ). Macierz pomyłek przedstawiono na rysunku 4.40.
4. Algorytm *kNN* uzyskał najlepszy wynik dla klasy 1 ( $100,00\%$ ), a najgorszy dla klasy 4 tj.  $60,00\%$  (rysunek 4.41). W pozostałych klasach poradził sobie w miarę dobrze.
5. Klasyfikator *Decision Tree* uzyskał  $SEN$  powyżej  $90,00\%$  dla klasy 1 oraz 3. Najgorszym wynikiem było  $75,00\%$ , który uzyskany został dla klasy 2 i 4 (rysunek 4.42).
6. Klasyfikator *Nearest Centroid* uzyskał  $SEN=100,00\%$  dla klasy 5 i tylko  $55,00\%$  dla klasy 4 (rysunek 4.43).
7. *Radius Neighbors* uzyskało najlepszy wynik dla klasy 5 tj.  $100,00\%$ . Najgorszy wynik uzyskany został dla klasy 4 i było to zaledwie  $45,00\%$  (rysunek 4.44).
8. Klasyfikator *Linear SVC* uzyskał  $100,00\%$   $SEN$  dla klasy 5 i tylko  $15,00\%$  dla klasy 4 (rysunek 4.45).
9. *Extra Trees* uzyskało najwyższą wartość  $SEN$   $94,74\%$  dla klasy 5. Wynik uzyskany dla klas 1-3 był w przedziale  $84,00\% - 90,00\%$ . Najgorszym uzyskanym wynikiem było  $70,00\%$  dla klasy 4 (rysunek 4.46).
10. *Stacking Classifier* w wariancie 1 najlepszy wynik uzyskał w klasie 1 -  $100,00\%$ , dla klas 2, 3 i 5 było to ponad  $90,00\%$ . Najgorszy wynik uzyskał dla klasy 4 -  $70,00\%$  (rysunek 4.47).
11. *Stacking Classifier* w wariancie 2 najlepiej poradził sobie z klasyfikacją w klasie 3 ( $SEN=92,31\%$ ), a najgorzej z klasą 4 ( $SEN=75,00\%$ ). Macierz pomyłek przedstawiona została na rysunku 4.48.

Podsumowując uzyskane wyniki zaobserwowano, że najgorsze wyniki zazwyczaj uzyskiwane były dla klasy 4, a klasa ta najczęściej mylona była z klasą 3 i 5. Wniosek jaki można wyciągnąć jest taki, że parametry opisujące jakość transmisji danych w tych trzech klasach nie różnią się w znacznym stopniu. Na rysunku 4.49 przedstawiono zestawienie wyników uzyskanych przez poszczególne klasyfikatory dla poszczególnych klas.

W tradycyjnych sieciach komputerowych cały ruch przesyłany jest zazwyczaj tą samą ścieżką. W metodzie P&R ścieżki transmisji można różnicować w celu spełnienia oczekiwań klienta. Klientowi może być trudno zrozumieć, w jaki sposób parametry transmisji wpływają na ogólną jakość transmisji. Dlatego koniecznym było

|             |   | Przewidywane |    |        |        |    |    |        |        |        |        |        |        |        |        |        |        |        |        |        |        |       |        |
|-------------|---|--------------|----|--------|--------|----|----|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|-------|--------|
|             |   | 1            | 2  | 3      | 4      | 5  |    |        |        |        |        |        |        |        |        |        |        |        |        |        |        |       |        |
| Rzeczywiste | 1 | 12           | 3  | 80,00% | 20,00% | 19 | 1  | 95,00% | 5,00%  | 23     | 3      | 88,46% | 11,54% | 18     | 2      | 90,00% | 10,00% | 15     | 4      | 78,95% | 21,05% |       |        |
|             | 2 | 1            | 19 | 0      | 0      | 4  | 76 | 95,00% | 5,00%  | 1      | 73     | 98,65% | 1,35%  | 6      | 74     | 92,50% | 7,50%  | 1      | 80     | 98,77% | 1,23%  |       |        |
|             | 3 | 0            | 1  | 23     | 2      | 0  | 0  | 1,18%  | 82,61% | 98,70% | 95,00% | 95,83% | 96,05% | 88,46% | 75,00% | 97,37% | 90,00% | 93,75% | 95,24% | 78,95% | 21,05% |       |        |
|             | 4 | 0            | 0  | 1      | 18     | 1  | 0  | 0      | 7,69%  | 3,45%  | 20,00% | 17,39% | 1,30%  | 5,00%  | 4,17%  | 3,95%  | 11,54% | 25,00% | 2,63%  | 10,00% | 6,25%  | 4,76% | 21,05% |
|             | 5 | 0            | 0  | 0      | 4      | 15 | 0  | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0     | 0      |

RYSUNEK 4.38: Macierz pomyłek dla najlepszego wyniku klasyfikacji klasyfikatora *Nu-SVC* (SEN=87,00%). W pomarańczowych komórkach SEN zostało przedstawione dla pojedynczych klas.

|             |   | Przewidywane |    |        |       |    |   |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |
|-------------|---|--------------|----|--------|-------|----|---|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
|             |   | 1            | 2  | 3      | 4     | 5  |   |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |
| Rzeczywiste | 1 | 14           | 1  | 93,33% | 6,67% | 16 | 4 | 80,00% | 20,00% | 25     | 1      | 96,15% | 3,85%  | 15     | 5      | 75,00% | 25,00% | 17     | 2      | 89,47% | 10,53% |        |        |
|             | 2 | 2            | 16 | 2      | 0     | 0  | 2 | 83     | 97,65% | 2,35%  | 1      | 79     | 98,75% | 1,25%  | 3      | 77     | 96,25% | 3,75%  | 2      | 79     | 97,53% | 2,47%  |        |
|             | 3 | 0            | 0  | 25     | 1     | 0  | 0 | 0      | 87,50% | 98,81% | 93,33% | 94,12% | 95,18% | 80,00% | 83,33% | 98,57% | 96,15% | 83,33% | 93,90% | 75,00% | 89,47% | 97,53% | 89,47% |
|             | 4 | 0            | 0  | 3      | 15    | 2  | 0 | 0      | 12,50% | 1,19%  | 6,67%  | 5,88%  | 4,82%  | 20,00% | 16,67% | 1,43%  | 3,85%  | 16,67% | 6,10%  | 25,00% | 10,53% | 2,47%  | 10,53% |
|             | 5 | 0            | 0  | 0      | 2     | 17 | 0 | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      |

RYSUNEK 4.39: Macierz pomyłek dla najlepszego wyniku klasyfikacji klasyfikatora *Random Forest* (SEN=87,00%). W pomarańczowych komórkach SEN zostało przedstawione dla pojedynczych klas.

zdefiniowanie jakości transmisji danych w bardziej zrozumiały sposób. Zdecydowano się zastosować skalę ocen od 1 do 5. Definiowanie statycznej tabeli translacji parametrów opisujących jakość transmisji nie było wystarczająco elastyczne. Z tego powodu zastosowano ML i wykonano pierwszą serię badań. Różne klasyfikatory zostały przeszkolone z wykorzystaniem danych testowych. Dane testowe uzyskano za pomocą QoE. Jakość transmisji oceniono na podstawie doświadczenia użytkownika. Dane uzyskane od użytkowników zostały zastosowane w procesie ML. Przetestowano dziewięć klasyfikatorów o różnych parametrach, a także zastosowano dwa warianty *Stacking Classifier*. Najlepszy wynik, SEN=89,00%, osiągnięto dla wariantu 1 *Stacking Classifier*. W tym wypadku zastosowano cztery estymatory, 10-krotną walidacją krzyżową, algorytm przetwarzania wstępnego (przeskalowania) *StandardScaler* i pozostałymi parametrami:

- *Linear SVC*:  $random\ state=44$ ,  $tol=2.49343E-05$ ,  $C=1.104396883$ ,
- *C-SVC*:  $kernel='poly'$ ,  $coef0=0.363466800$ ,  $degree=4$ ,  $gamma=1.148169838$ ,
- *Nearest Centroid*,
- *kNN*:  $neighbors=6$ ,  $weights='uniform'$ ,  $algorithm='brute'$ ,  $leaf\ size=39$ .

*Logistic Regression* zastosowano jako ostateczny estymator. Bardzo interesujące było to, że najwyższa wartość SEN została osiągnięta przy zastosowaniu tylko trzech cech - współczynnik utraty pakietów został wyeliminowany przez GA jako cecha nieznacząca.



|             |   | Przewidywane     |                 |                  |                  |                 |                  |                  |                 |                 |                  |                 |                  |                  |                 |                  |
|-------------|---|------------------|-----------------|------------------|------------------|-----------------|------------------|------------------|-----------------|-----------------|------------------|-----------------|------------------|------------------|-----------------|------------------|
|             |   | 1                | 2               | 3                | 4                | 5               |                  |                  |                 |                 |                  |                 |                  |                  |                 |                  |
| Rzeczywiste | 1 | 13               | 2               | 86,67%<br>13,33% | 15               | 5               | 75,00%<br>25,00% | 24               | 2               | 92,31%<br>7,69% | 14               | 6               | 70,00%<br>30,00% | 17               | 2               | 89,47%<br>10,53% |
|             | 2 | 3                | 82              | 96,47%<br>3,53%  | 3                | 77              | 96,25%<br>3,75%  | 6                | 68              | 91,89%<br>8,11% | 3                | 77              | 96,25%<br>3,75%  | 2                | 79              | 97,53%<br>2,47%  |
|             | 3 | 81,25%<br>18,75% | 97,62%<br>2,38% | 86,67%<br>13,33% | 83,33%<br>16,67% | 93,90%<br>6,10% | 75,00%<br>25,00% | 80,00%<br>20,00% | 97,14%<br>2,86% | 92,31%<br>7,69% | 82,35%<br>17,65% | 92,77%<br>7,23% | 70,00%<br>30,00% | 89,47%<br>10,53% | 97,53%<br>2,47% | 89,47%<br>10,53% |
|             | 4 | 0                | 0               | 4                | 14               | 2               | 0                | 0                | 4               | 14              | 2                | 0               | 0                | 4                | 14              | 2                |
|             | 5 | 0                | 0               | 0                | 2                | 17              | 0                | 0                | 0               | 2               | 17               | 0               | 0                | 0                | 2               | 17               |

RYSUNEK 4.40: Macierz pomyłek dla najlepszego wyniku klasyfikacji klasyfikatora C-SVC (SEN=83,00%). W pomarańczowych komórkach SEN zostało przedstawione dla pojedynczych klas.

|             |   | Przewidywane     |                  |                  |                  |                 |                  |                  |                 |                 |                  |                 |                  |                  |                 |                  |
|-------------|---|------------------|------------------|------------------|------------------|-----------------|------------------|------------------|-----------------|-----------------|------------------|-----------------|------------------|------------------|-----------------|------------------|
|             |   | 1                | 2                | 3                | 4                | 5               |                  |                  |                 |                 |                  |                 |                  |                  |                 |                  |
| Rzeczywiste | 1 | 15               | 0                | 100,00%<br>0,00% | 15               | 5               | 75,00%<br>25,00% | 25               | 1               | 96,15%<br>3,85% | 12               | 8               | 60,00%<br>40,00% | 17               | 2               | 89,47%<br>10,53% |
|             | 2 | 3                | 82               | 96,47%<br>3,53%  | 0                | 80              | 100,00%<br>0,00% | 6                | 68              | 91,89%<br>8,11% | 3                | 77              | 96,25%<br>3,75%  | 4                | 77              | 95,06%<br>4,94%  |
|             | 3 | 83,33%<br>16,67% | 100,00%<br>0,00% | 100,00%<br>0,00% | 100,00%<br>0,00% | 94,12%<br>5,88% | 75,00%<br>25,00% | 80,65%<br>19,35% | 98,55%<br>1,45% | 96,15%<br>3,85% | 80,00%<br>20,00% | 90,59%<br>9,41% | 60,00%<br>40,00% | 80,95%<br>19,05% | 97,47%<br>2,53% | 89,47%<br>10,53% |
|             | 4 | 0                | 0                | 25               | 1                | 0               | 0                | 0                | 25              | 1               | 0                | 0               | 0                | 25               | 1               |                  |
|             | 5 | 0                | 0                | 0                | 2                | 17              | 0                | 0                | 0               | 2               | 17               | 0               | 0                | 0                | 2               | 17               |

RYSUNEK 4.41: Macierz pomyłek dla najlepszego wyniku klasyfikacji algorytmu kNN (SEN=84,00%). W pomarańczowych komórkach SEN zostało przedstawione dla pojedynczych klas.

|             |   | Przewidywane     |                 |                 |                 |                 |                  |                  |                 |                 |                  |                 |                  |                  |                 |                  |
|-------------|---|------------------|-----------------|-----------------|-----------------|-----------------|------------------|------------------|-----------------|-----------------|------------------|-----------------|------------------|------------------|-----------------|------------------|
|             |   | 1                | 2               | 3               | 4               | 5               |                  |                  |                 |                 |                  |                 |                  |                  |                 |                  |
| Rzeczywiste | 1 | 14               | 1               | 93,33%<br>6,67% | 15              | 5               | 75,00%<br>25,00% | 24               | 2               | 92,31%<br>7,69% | 15               | 5               | 75,00%<br>25,00% | 16               | 3               | 84,21%<br>15,79% |
|             | 2 | 3                | 82              | 96,47%<br>3,53% | 1               | 79              | 98,75%<br>1,25%  | 5                | 69              | 93,24%<br>6,76% | 5                | 75              | 93,75%<br>6,25%  | 2                | 79              | 97,53%<br>2,47%  |
|             | 3 | 82,35%<br>17,65% | 98,80%<br>1,20% | 93,33%<br>6,67% | 93,75%<br>6,25% | 94,05%<br>5,95% | 75,00%<br>25,00% | 82,76%<br>17,24% | 97,18%<br>2,82% | 92,31%<br>7,69% | 75,00%<br>25,00% | 93,75%<br>6,25% | 75,00%<br>25,00% | 88,89%<br>11,11% | 96,34%<br>3,66% | 84,21%<br>15,79% |
|             | 4 | 0                | 0               | 3               | 15              | 2               | 0                | 0                | 3               | 15              | 2                | 0               | 0                | 3                | 15              | 2                |
|             | 5 | 0                | 0               | 0               | 3               | 16              | 0                | 0                | 0               | 3               | 16               | 0               | 0                | 0                | 3               | 16               |

RYSUNEK 4.42: Macierz pomyłek dla najlepszego wyniku klasyfikacji klasyfikatora Decision Tree (SEN=84,00%). W pomarańczowych komórkach SEN zostało przedstawione dla pojedynczych klas.

|             |   | Przewidywane     |                 |                 |                  |                 |                  |                  |                 |                  |                  |                  |                  |                  |                  |                  |
|-------------|---|------------------|-----------------|-----------------|------------------|-----------------|------------------|------------------|-----------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|
|             |   | 1                | 2               | 3               | 4                | 5               |                  |                  |                 |                  |                  |                  |                  |                  |                  |                  |
| Rzeczywiste | 1 | 14               | 1               | 93,33%<br>6,67% | 16               | 4               | 80,00%<br>20,00% | 21               | 5               | 80,77%<br>19,23% | 11               | 9                | 55,00%<br>45,00% | 19               | 0                | 100,00%<br>0,00% |
|             | 2 | 3                | 82              | 96,47%<br>3,53% | 2                | 78              | 97,50%<br>2,50%  | 4                | 70              | 94,59%<br>5,41%  | 4                | 76               | 95,00%<br>5,00%  | 6                | 75               | 92,59%<br>7,41%  |
|             | 3 | 82,35%<br>17,65% | 98,80%<br>1,20% | 93,33%<br>6,67% | 88,89%<br>11,11% | 95,12%<br>4,88% | 80,00%<br>20,00% | 84,00%<br>16,00% | 93,33%<br>6,67% | 80,77%<br>19,23% | 73,33%<br>26,67% | 89,41%<br>10,59% | 55,00%<br>45,00% | 76,00%<br>24,00% | 100,00%<br>0,00% | 100,00%<br>0,00% |
|             | 4 | 0                | 0               | 1               | 21               | 4               | 0                | 0                | 1               | 21               | 4                | 0                | 0                | 1                | 21               | 4                |
|             | 5 | 0                | 0               | 0               | 1                | 19              | 0                | 0                | 0               | 1                | 19               | 0                | 0                | 0                | 1                | 19               |

RYSUNEK 4.43: Macierz pomyłek dla najlepszego wyniku klasyfikacji klasyfikatora Nearest Centroid (SEN=81,00%). W pomarańczowych komórkach SEN zostało przedstawione dla pojedynczych klas.

|             |   | Przewidywane    |                 |                  |                  |                 |                  |                  |                 |                 |                  |                  |                  |                  |                  |                  |
|-------------|---|-----------------|-----------------|------------------|------------------|-----------------|------------------|------------------|-----------------|-----------------|------------------|------------------|------------------|------------------|------------------|------------------|
|             |   | 1               | 2               | 3                | 4                | 5               |                  |                  |                 |                 |                  |                  |                  |                  |                  |                  |
| Rzeczywiste | 1 | 13              | 2               | 86,67%<br>13,33% | 18               | 2               | 90,00%<br>10,00% | 24               | 2               | 92,31%<br>7,69% | 9                | 11               | 45,00%<br>55,00% | 19               | 0                | 100,00%<br>0,00% |
|             | 2 | 1               | 84              | 98,82%<br>1,18%  | 2                | 78              | 97,50%<br>2,50%  | 6                | 68              | 91,89%<br>8,11% | 2                | 78               | 97,50%<br>2,50%  | 6                | 75               | 92,59%<br>7,41%  |
|             | 3 | 92,86%<br>7,14% | 97,67%<br>2,33% | 86,67%<br>13,33% | 90,00%<br>10,00% | 97,50%<br>2,50% | 90,00%<br>10,00% | 80,00%<br>20,00% | 97,14%<br>2,86% | 92,31%<br>7,69% | 81,82%<br>18,18% | 87,64%<br>12,36% | 45,00%<br>55,00% | 76,00%<br>24,00% | 100,00%<br>0,00% | 100,00%<br>0,00% |
|             | 4 | 0               | 0               | 5                | 9                | 6               | 0                | 0                | 5               | 9               | 6                | 0                | 0                | 5                | 9                | 6                |
|             | 5 | 0               | 0               | 0                | 5                | 19              | 0                | 0                | 0               | 5               | 19               | 0                | 0                | 0                | 5                | 19               |

RYSUNEK 4.44: Macierz pomyłek dla najlepszego wyniku klasyfikacji klasyfikatora Radius Neighbors (SEN=83,00%). W pomarańczowych komórkach SEN zostało przedstawione dla pojedynczych klas.



|             |   | Przewidywane |    |    |   |        | 1      |        | 2      |        | 3       |         | 4       |         | 5       |         |
|-------------|---|--------------|----|----|---|--------|--------|--------|--------|--------|---------|---------|---------|---------|---------|---------|
|             |   | 1            | 2  | 3  | 4 | 5      |        |        |        |        |         |         |         |         |         |         |
| Rzeczywiste | 1 | 14           | 1  | 0  | 0 | 0      | 14     | 1      | 14     | 6      | 25      | 1       | 3       | 17      | 19      | 0       |
|             | 2 | 3            | 14 | 3  | 0 | 0      | 1      | 79     | 10     | 64     | 0       | 80      | 11      | 70      | 11      | 70      |
|             | 3 | 0            | 0  | 25 | 0 | 1      | 3      | 82     | 3      | 82     | 0       | 80      | 0       | 80      | 0       | 80      |
|             | 4 | 0            | 0  | 7  | 3 | 10     | 82,35% | 98,80% | 82,35% | 98,80% | 100,00% | 82,47%  | 100,00% | 82,47%  | 63,33%  | 100,00% |
|             | 5 | 0            | 0  | 0  | 0 | 19     | 17,65% | 1,20%  | 6,67%  | 6,67%  | 28,57%  | 1,54%   | 0,00%   | 17,53%  | 36,67%  | 0,00%   |
|             |   |              |    |    |   | 93,33% | 6,67%  | 93,33% | 6,67%  | 96,15% | 3,85%   | 15,00%  | 85,00%  | 100,00% | 0,00%   |         |
|             |   |              |    |    |   | 96,47% | 3,53%  | 98,75% | 1,25%  | 86,49% | 13,51%  | 100,00% | 0,00%   | 86,42%  | 13,58%  |         |
|             |   |              |    |    |   | 82,35% | 98,80% | 93,33% | 92,94% | 71,43% | 98,46%  | 100,00% | 82,47%  | 63,33%  | 100,00% |         |
|             |   |              |    |    |   | 17,65% | 1,20%  | 6,67%  | 7,06%  | 28,57% | 1,54%   | 0,00%   | 17,53%  | 36,67%  | 0,00%   |         |

RYSUNEK 4.45: Macierz pomyłek dla najlepszego wyniku klasyfikacji klasyfikatora *Linear SVC* (SEN=75,00%). W pomarańczowych komórkach SEN zostało przedstawione dla pojedynczych klas.

|             |   | Przewidywane |    |    |    |         | 1       |         | 2      |         | 3      |         | 4      |         | 5      |        |
|-------------|---|--------------|----|----|----|---------|---------|---------|--------|---------|--------|---------|--------|---------|--------|--------|
|             |   | 1            | 2  | 3  | 4  | 5       |         |         |        |         |        |         |        |         |        |        |
| Rzeczywiste | 1 | 13           | 2  | 0  | 0  | 0       | 13      | 2       | 18     | 2       | 22     | 4       | 14     | 6       | 18     | 1      |
|             | 2 | 1            | 18 | 1  | 0  | 0       | 1       | 84      | 3      | 77      | 4      | 70      | 4      | 76      | 3      | 78     |
|             | 3 | 0            | 1  | 22 | 3  | 0       | 0       | 84      | 0      | 77      | 0      | 70      | 0      | 76      | 0      | 78     |
|             | 4 | 0            | 0  | 3  | 14 | 3       | 92,86%  | 97,67%  | 85,71% | 97,47%  | 84,62% | 94,59%  | 77,78% | 92,68%  | 85,71% | 98,73% |
|             | 5 | 0            | 0  | 0  | 1  | 18      | 7,14%   | 2,33%   | 14,29% | 2,53%   | 15,38% | 5,41%   | 22,22% | 7,32%   | 14,29% | 1,27%  |
|             |   |              |    |    |    | 86,67%  | 13,33%  | 85,71%  | 97,47% | 84,62%  | 94,59% | 77,78%  | 92,68% | 85,71%  | 98,73% |        |
|             |   |              |    |    |    | 98,82%  | 1,18%   | 96,25%  | 3,75%  | 94,59%  | 5,41%  | 95,00%  | 5,00%  | 96,30%  | 3,70%  |        |
|             |   |              |    |    |    | 100,00% | 100,00% | 100,00% | 90,00% | 100,00% | 84,62% | 100,00% | 70,00% | 100,00% | 94,74% |        |
|             |   |              |    |    |    | 0,00%   | 0,00%   | 0,00%   | 10,00% | 0,00%   | 15,38% | 0,00%   | 30,00% | 0,00%   | 5,26%  |        |

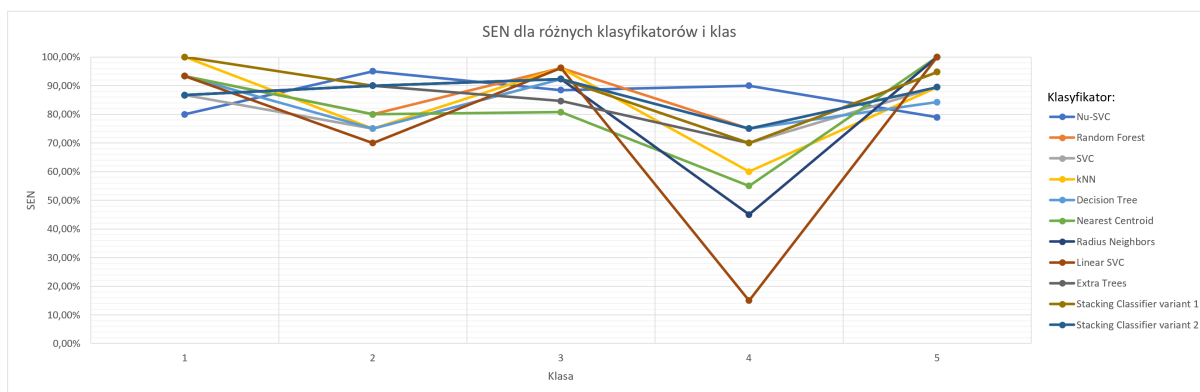
RYSUNEK 4.46: Macierz pomyłek dla najlepszego wyniku klasyfikacji klasyfikatora *Extra Trees* (SEN=85,00%). W pomarańczowych komórkach SEN zostało przedstawione dla pojedynczych klas.

|             |   | Przewidywane |    |    |    |         | 1       |         | 2      |        | 3      |        | 4      |        | 5      |        |
|-------------|---|--------------|----|----|----|---------|---------|---------|--------|--------|--------|--------|--------|--------|--------|--------|
|             |   | 1            | 2  | 3  | 4  | 5       |         |         |        |        |        |        |        |        |        |        |
| Rzeczywiste | 1 | 15           | 0  | 0  | 0  | 0       | 15      | 0       | 18     | 2      | 24     | 2      | 14     | 6      | 18     | 1      |
|             | 2 | 0            | 18 | 2  | 0  | 0       | 0       | 85      | 1      | 79     | 6      | 68     | 2      | 78     | 2      | 79     |
|             | 3 | 0            | 1  | 24 | 1  | 0       | 0       | 85      | 0      | 79     | 0      | 68     | 0      | 78     | 0      | 79     |
|             | 4 | 0            | 0  | 4  | 14 | 2       | 100,00% | 100,00% | 94,74% | 97,53% | 80,00% | 97,14% | 87,50% | 92,86% | 90,00% | 98,75% |
|             | 5 | 0            | 0  | 0  | 1  | 18      | 0,00%   | 0,00%   | 5,26%  | 2,47%  | 20,00% | 2,86%  | 12,50% | 7,14%  | 10,00% | 1,25%  |
|             |   |              |    |    |    | 100,00% | 100,00% | 94,74%  | 97,53% | 80,00% | 97,14% | 87,50% | 92,86% | 90,00% | 98,75% |        |
|             |   |              |    |    |    | 0,00%   | 0,00%   | 5,26%   | 2,47%  | 20,00% | 2,86%  | 12,50% | 7,14%  | 10,00% | 1,25%  |        |
|             |   |              |    |    |    | 100,00% | 100,00% | 94,74%  | 97,53% | 80,00% | 97,14% | 87,50% | 92,86% | 90,00% | 98,75% |        |
|             |   |              |    |    |    | 0,00%   | 0,00%   | 5,26%   | 2,47%  | 20,00% | 2,86%  | 12,50% | 7,14%  | 10,00% | 1,25%  |        |

RYSUNEK 4.47: Macierz pomyłek dla najlepszego wyniku klasyfikacji *Stacking Classifier* w wariancie 1 (SEN=89,00%). W pomarańczowych komórkach SEN zostało przedstawione dla pojedynczych klas.

|             |   | Przewidywane |    |    |    |         | 1       |        | 2      |        | 3      |        | 4      |        | 5      |        |
|-------------|---|--------------|----|----|----|---------|---------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
|             |   | 1            | 2  | 3  | 4  | 5       |         |        |        |        |        |        |        |        |        |        |
| Rzeczywiste | 1 | 13           | 2  | 0  | 0  | 0       | 13      | 2      | 18     | 2      | 24     | 2      | 15     | 5      | 17     | 2      |
|             | 2 | 0            | 18 | 2  | 0  | 0       | 0       | 85     | 3      | 77     | 5      | 69     | 3      | 77     | 2      | 79     |
|             | 3 | 0            | 1  | 24 | 1  | 0       | 0       | 85     | 0      | 77     | 0      | 69     | 0      | 77     | 0      | 79     |
|             | 4 | 0            | 0  | 3  | 15 | 2       | 100,00% | 97,70% | 85,71% | 97,47% | 82,76% | 97,18% | 83,33% | 93,90% | 89,47% | 97,53% |
|             | 5 | 0            | 0  | 0  | 2  | 17      | 0,00%   | 2,30%  | 14,29% | 2,53%  | 17,24% | 2,82%  | 16,67% | 6,10%  | 10,53% | 2,47%  |
|             |   |              |    |    |    | 100,00% | 97,70%  | 85,71% | 97,47% | 82,76% | 97,18% | 83,33% | 93,90% | 89,47% | 97,53% |        |
|             |   |              |    |    |    | 0,00%   | 2,30%   | 14,29% | 2,53%  | 17,24% | 2,82%  | 16,67% | 6,10%  | 10,53% | 2,47%  |        |
|             |   |              |    |    |    | 100,00% | 97,70%  | 85,71% | 97,47% | 82,76% | 97,18% | 83,33% | 93,90% | 89,47% | 97,53% |        |
|             |   |              |    |    |    | 0,00%   | 2,30%   | 14,29% | 2,53%  | 17,24% | 2,82%  | 16,67% | 6,10%  | 10,53% | 2,47%  |        |

RYSUNEK 4.48: Macierz pomyłek dla najlepszego wyniku klasyfikacji *Stacking Classifier* w wariancie 2 (SEN=87,00%). W pomarańczowych komórkach SEN zostało przedstawione dla pojedynczych klas.



RYSUNEK 4.49: SEN dla poszczególnych klasyfikatorów (etap I) i klas

| Parametr                       | Wartość                |
|--------------------------------|------------------------|
| Rozmiar populacji              | 100                    |
| Liczba generacji               | 100                    |
| Prawdopodobieństwo krzyżowania | 0,4                    |
| Prawdopodobieństwo mutacji     | 0,9                    |
| Algorytm selekcji              | tournament (rozmiar=3) |
| Funkcja dopasowania            | Accuracy lub F1 score  |

TABELA 4.20: Parametry algorytmu genetycznego (GA) - etap II

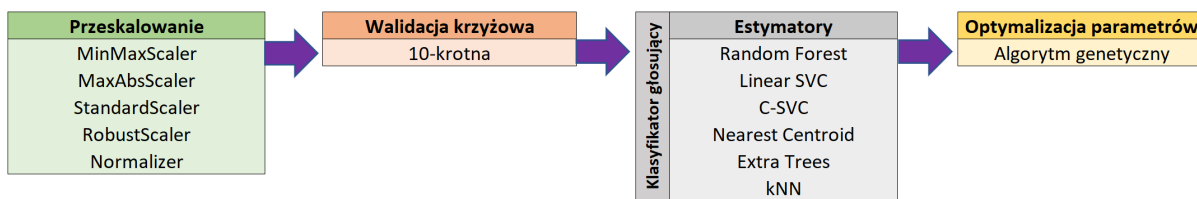
#### 4.4.2 Etap II

W kolejnych badaniach zdecydowano się na stworzenie własnych algorytmów wykorzystujących uczenie zespołowe. Do przeprowadzenia tych badań również wykorzystano bibliotekę Sklearn i Deap oraz język programowania Python. Wykorzystano klasyfikatory, które uzyskały najlepsze wyniki w pierwszym etapie badań. Parametry oraz zakresy wartości były takie same, jak w poprzednich badaniach (tabela 4.6). W tabeli 4.20 przedstawiono parametry algorytmów genetycznych wykorzystanych w badaniach. Badania przeprowadzono dla testowego zestawu danych. Dla każdego algorytmu zastosowano 10-krotną walidację krzyżową, każdy z algorytmów uczony był z wykorzystaniem wszystkich czterech cech, tj. opóźnienia transmisji, fluktuacji opóźnienia, przepływności i współczynnik utraty pakietów. Zaproponowano kilka różnych algorytmów:

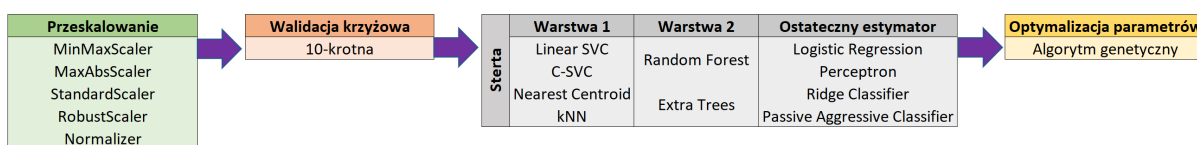
##### 1. Algorytm 1.

Pierwszym algorytmem w prezentowanych badaniach był *Voting Classifier* [223, 224] dostępny w bibliotece Sklearn. Przetestowano go w kombinacji z kilkoma klasyfikatorami. Proces uczenia przedstawiony został na rysunku 4.50, podzielony został na kilka etapów:

- (a) Przetwarzanie wstępne (przeskalowanie) - wykorzystano kilka różnych metod tj. *MinMaxScaler*, *MaxAbsScaler*, *StandardScaler*, *RobustScaler* i *Normalizer*.
- (b) Walidacja krzyżowa - zastosowano 10-krotną walidację krzyżową.
- (c) Klasyfikacja - zastosowano *Voting Classifier* w kombinacji z estymatorami *Random Forest*, *Linear SVC*, *C-SVC*, *Nearest Centroid*, *Extra Trees* i *kNN*.
- (d) Optymalizacja parametrów - wykorzystano GA do optymalizacji parametrów.



RYSUNEK 4.50: Proces ML, algorytm 1



RYSUNEK 4.51: Proces ML, algorytm 2

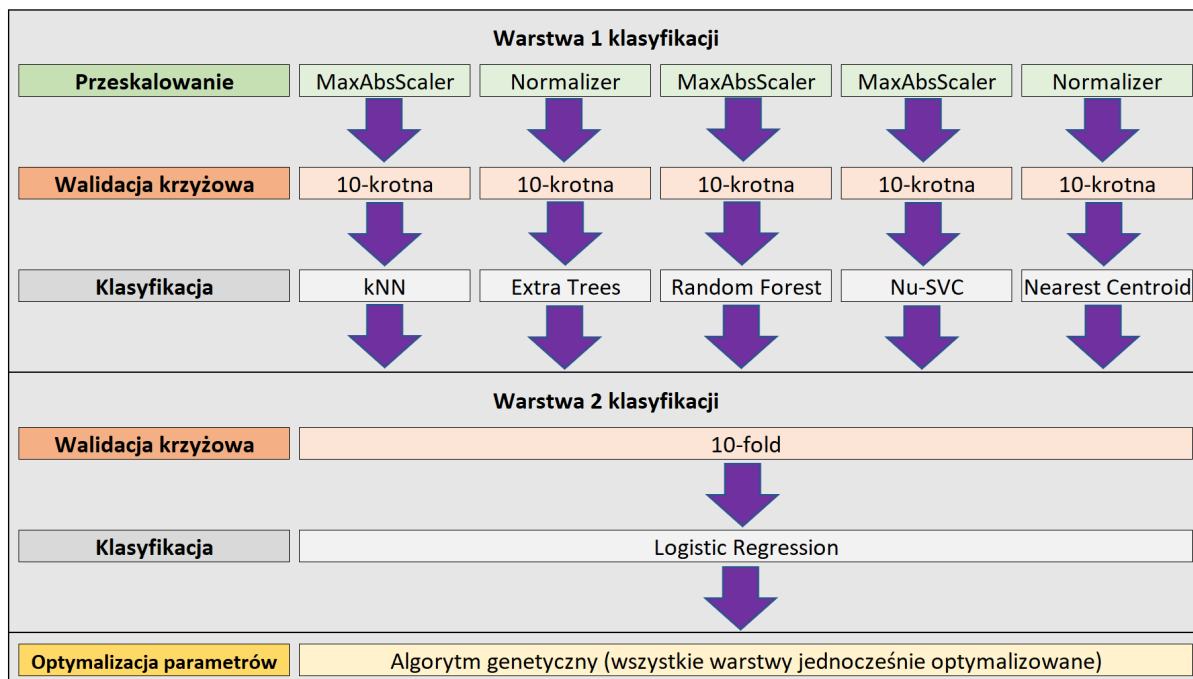
## 2. Algorytm 2.

W pierwszych badaniach *Stacking Classifier* uzyskał dobry wynik, zdecydowano się na rozszerzenie tej metody poprzez dodanie kolejnej warstwy estymacji. Na rysunku 4.51 przedstawiono przebieg procesu uczenia, składa się on z następujących etapów:

- Przetwarzanie wstępne (przeskalowanie) - przetestowano kilka różnych metod.
- Walidacja krzyżowa - zastosowano 10-krotną walidację krzyżową.
- Klasyfikacja - wykorzystano dwie warstwy estymacji. W warstwie 1 wykorzystano estymatory *Linear SVC*, *C-SVC*, *Nearest Centroid* oraz *kNN*. Z kolei w warstwie 2 wykorzystano *Random Forest* i *Extra Trees*. Do ostatecznej estymacji wykorzystano jedną metodę, a na potrzeby badań zweryfikowano wynik uzyskany przez każdą z czterech możliwych metod.
- Optymalizacja parametrów - wykorzystano GA.

## 3. Algorytm 3.

Algorytm 3 posiadał trzy warianty - w zależności od liczby warstw klasyfikacji. Wariant 1 (rysunek 4.52) posiada dwie warstwy klasyfikacji - warstwa 1 składa się z 5 klasyfikatorów. Przetwarzanie wstępne wykonywane jest osobno dla każdego klasyfikatora, konkretne metody dobrano na podstawie wcześniejszych badań - wybrano metody w przypadku stosowania których, klasyfikatory uzyskały najlepsze wyniki. Po przetworzeniu wstępnym wykonywana była 10-krotna walidacja krzyżowa. Tak przygotowane dane podawane były na wejściu klasyfikatorów. Warstwa 2 to ostateczna klasyfikacja, na wejściu tej warstwy podawane są wyniki klasyfikacji poszczególnych klasyfikatorów

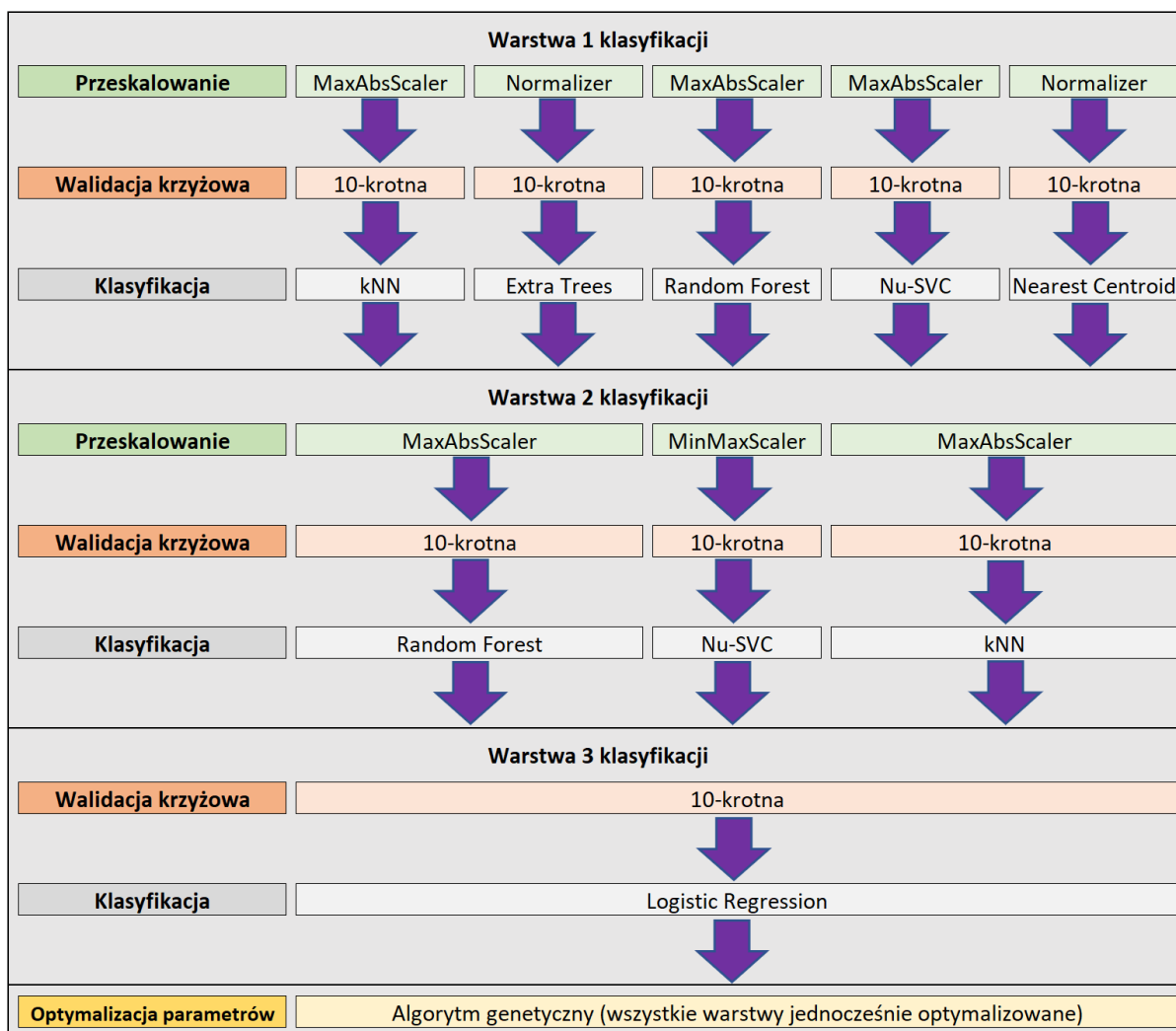


RYSUNEK 4.52: Proces ML, algorytm 3 wariant 1

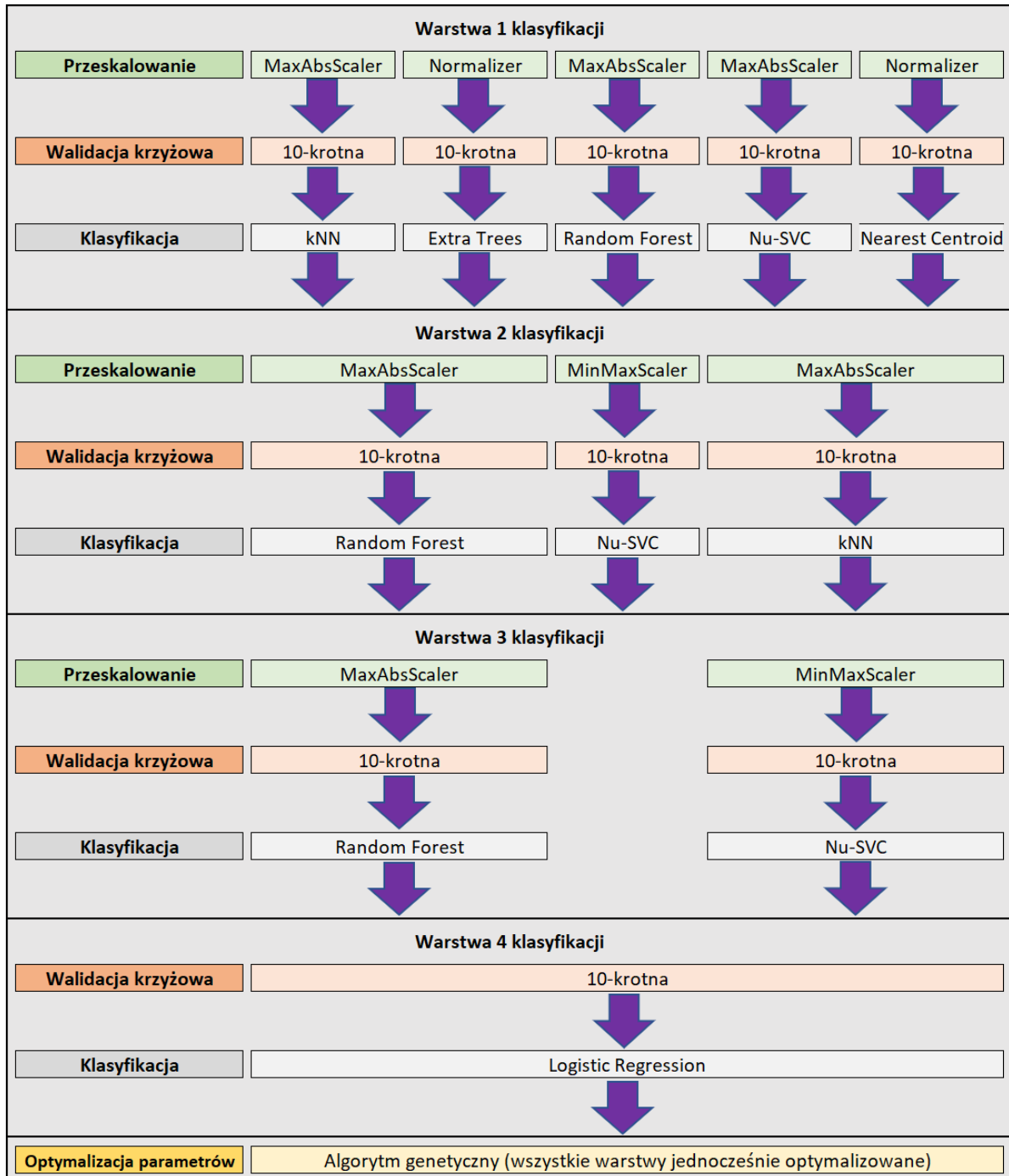
warstwy 1. W warstwie 2 również wykorzystano 10-krotną walidację krzyżową, jako algorytm ostatecznej klasyfikacji wykorzystano *Logistic Regression*. Ostatnim krokiem była optymalizacja parametrów wykonywana przez GA. Co istotne GA równocześnie dokonuje optymalizacji parametrów wszystkich klasyfikatorów, tj. zarówno warstwy 1, jak i 2.

Wariant 2 różni się od wariantu 1 liczbą warstw klasyfikacji - dodano jedną dodatkową warstwę klasyfikacji (rysunek 4.53). Zasada działania warstwy 1 pozostała bez zmian w stosunku do wariantu 1, z kolei warstwa 3 działa na takiej samej zasadzie jak warstwa 2 w wariacie 1 - ostateczna klasyfikacja. W wariacie 2 w warstwie 2 wykorzystano trzy klasyfikatory. Na wejście warstwy 2 podawane są wyniki klasyfikacji warstwy 1. Następnie dokonywane jest przetwarzanie wstępne. Również w tym przypadku dobór konkretnych metod dokonany został na podstawie wcześniejszych badań. Ostatnim działaniem w tej warstwie jest klasyfikacja. Wyniki otrzymane w tej warstwie podawane są na wejściu warstwy kolejnej.

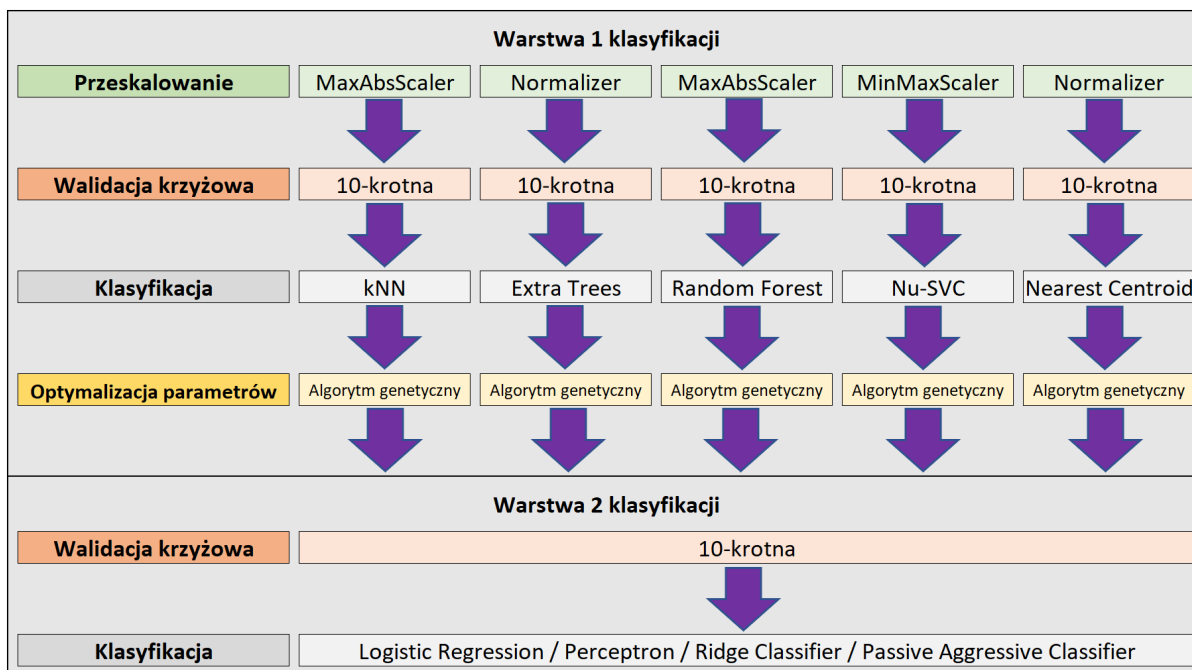
Kolejny wariant, czyli wariant 3 (rysunek 4.54), jest rozszerzeniem wariantu 2 o kolejną warstwę klasyfikacji. Kolejność kroków wykonywanych w dodatkowej warstwie jest taka sama jak w poprzednich warstwach, tj. przetwarzanie wstępne, walidacja krzyżowa i klasyfikacja. Wyniki klasyfikacji dokonanej przez tą warstwę podawane są na wejściu warstwy kolejnej.



RYSUNEK 4.53: Proces ML, algorytm 3 wariant 2



RYSUNEK 4.54: Proces ML, algorytm 3 wariant 3



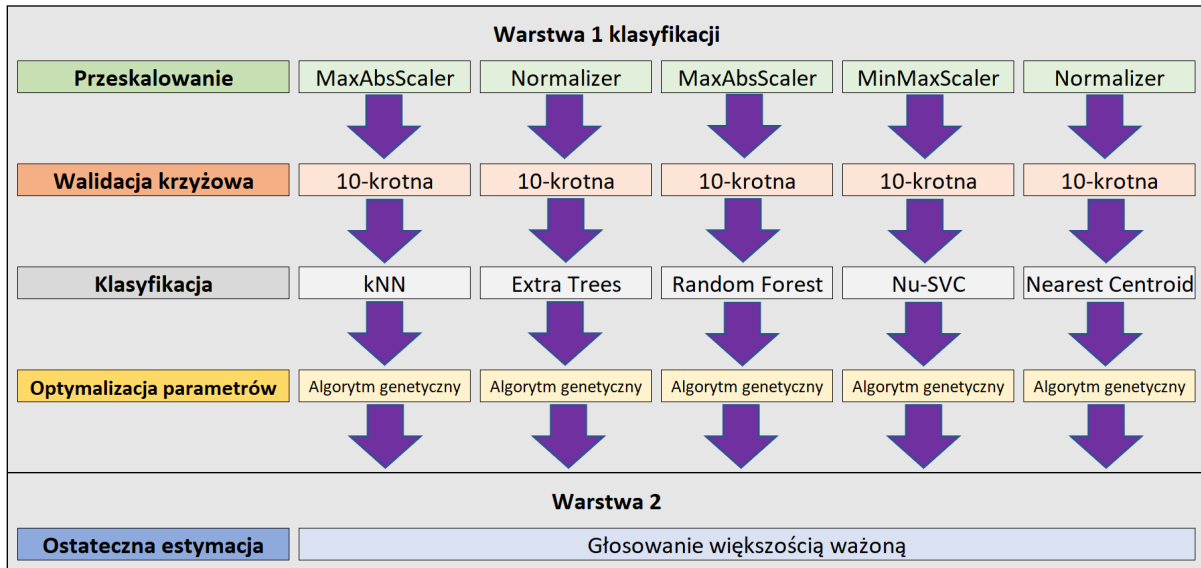
RYSUNEK 4.55: Proces ML, algorytm 4

## 4. Algorytm 4.

W algorytmie tym wykorzystano dwie warstwy klasyfikacji (rysunek 4.55). Warstwa 1 wykorzystuje 5 różnych klasyfikatorów. Pierwszym krokiem jest przetwarzanie wstępne, dobór metod oparto na wynikach poprzednich badań. Kolejny krok to walidacja krzyżowa. Następnie dokonywana jest klasyfikacja. Skuteczność poszczególnych klasyfikatorów ewaluowana jest na podstawie *F1 Score*. Po przeprowadzeniu klasyfikacji GA weryfikuje uzyskany wynik i dokonuje optymalizacji parametrów każdego klasyfikatora osobno. Warstwa 2 otrzymuje wyniki klasyfikacji przeprowadzonej przez warstwę 1. Wykonywana jest walidacja krzyżowa, a następnie dokonywana jest ostateczna klasyfikacja. Przeprowadzono badania z wykorzystaniem kilku różnych klasyfikatorów w warstwie 2, tj. *Logistic Regression*, *Perceptron*, *Ridge Classifier* oraz *Passive Aggressive Classifier*. Ostateczny wynik klasyfikacji oceniony został na podstawie SEN.

## 5. Algorytm 5.

Algorytm 5 składa się z dwóch warstw klasyfikacji (rysunek 4.56). W warstwie 1 klasyfikatory zostały wyszkolone do klasyfikacji wszystkich klas, lecz z założeniem, że specjalizują się w klasyfikacji jednej klasy. Algorytm *kNN* specjalizuje się w klasyfikacji klasy 1, *Extra Trees* w klasie 2, *Random Forest* w klasie 3. Z kolei *Nu-SVC* specjalizuje się w klasyfikowaniu klasy 4, a *Nearest Centroid* klasy 5. Przetwarzanie wstępne wykonywane jest dla każdego klasyfikatora



RYSUNEK 4.56: Proces ML, algorytm 5

osobno. Następnie wykonywana jest 10-krotna walidacja krzyżowa. Parametry każdego klasyfikatora optymalizowane są osobno przez GA. W ten sposób uzyskano sytuację, w której klasyfikatory specjalizują się w klasyfikowaniu konkretnej klasy. Ewaluacja dokładności klasyfikacji poszczególnych klasyfikatorów wykonywana jest przy użyciu ważonego  $F1$  Score. Waga próbek należącej do klasy, w której klasyfikacji specjalizuje się klasyfikator, jest dużo wyższa niż w przypadku próbek należących do pozostałych klas.

Warstwa 2 dokonuje ostatecznej estymacji na podstawie wyników uzyskanych w warstwie 1. Estymacja wykonywana jest przy użyciu głosowania większością ważoną, z uwzględnieniem specjalizacji poszczególnych klasyfikatorów warstwy 1. Przykładowo, jeśli algorytm *kNN* (wyspecjalizowany w klasyfikacji klasy 1) wskaże, że dana próbka należy do klasy 1 to takie wskazanie ma dużo wyższą wagę niż wskazanie w innych klasach. Jeśli klasyfikator *Nearest Centroid* wskaże, że próbka należy do klasy 5, to taka klasyfikacja jest bardziej wiarygodna niż pozostałe. Aby uzyskać taki efekt zastosowano różne wagi klasyfikacji. Jeśli klasyfikator specjalizujący się w klasyfikacji klasy 5 wskaże inną klasę dla pewnej próbki, to waga takiego wskazania będzie taka sama jak pozostałych klasyfikatorów nie będących wyspecjalizowanymi w danej klasie. Na podstawie uzyskanych wyników - wag klasyfikacji, dokonywana jest ostateczna estymacja i wskazanie klasy, w której próbka uzyskała w warstwie 1 najwyższą wagę. Całościowej oceny klasyfikacji dokonano z wykorzystaniem SEN.

## 6. Algorytm 6.



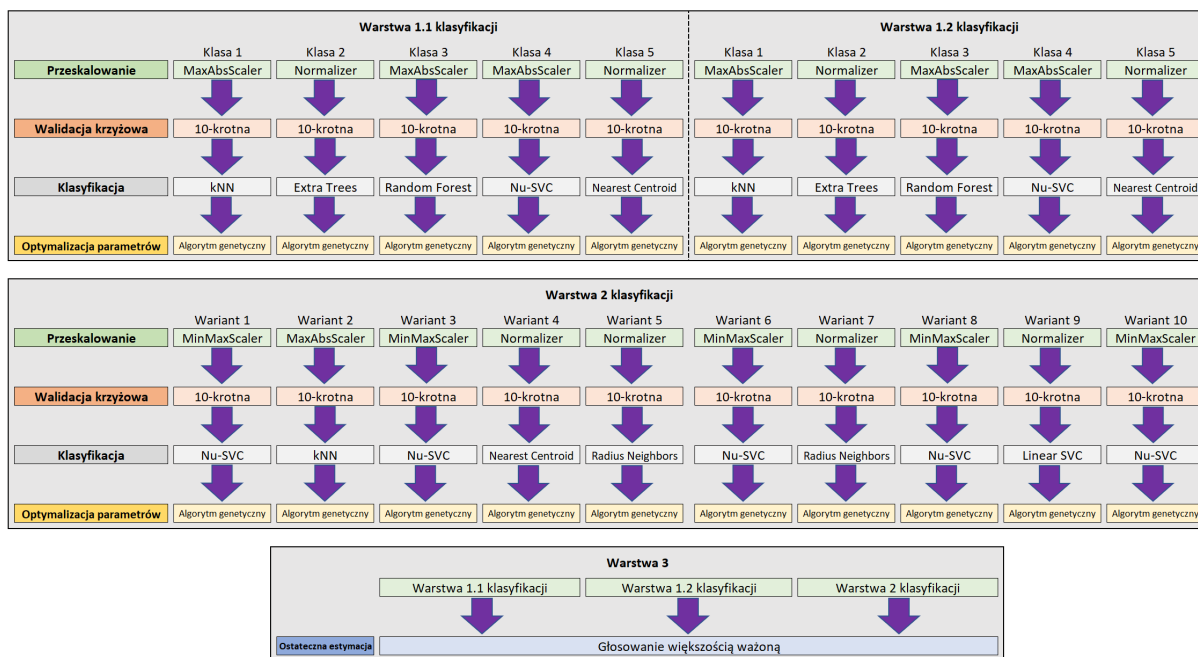
Algorytm ten jest najbardziej skomplikowanym, jak chodzi o złożoność i trudność implementacji. Wykorzystuje 3 warstwy (rysunek 4.57). Warstwa 1 podzielona została na dwie części, w każdej części wykorzystywanych jest pięć klasyfikatorów. Każdy klasyfikator specjalizuje się w klasyfikacji jednej konkretnej klasy. Oznacza to, że w warstwie 1 do klasyfikacji klasy 1 wyspecjalizowano dwa klasyfikatory. Pierwszym krokiem jest wykonanie przetwarzania wstępnego (dla każdego klasyfikatora osobno), następnie wykonywana jest walidacja krzyżowa. Ewaluacja dokładności klasyfikacji wykonywana jest z wykorzystaniem ważonego *F1 Score*. Wagi zależne są od klasy w klasyfikacji której specjalizuje się dany klasyfikator. Uzyskany wynik weryfikowany jest przez GA, parametry są optymalizowane.

W tym samym czasie szkolona jest warstwa 2. W warstwie 2 klasyfikatory specjalizują się w różnych wariantach klasyfikacji. Pojedynczy wariant oznacza specjalizację w klasyfikacji dwóch różnych klas. Warianty są następujące:

- 1: Klasa 1 i 2,
- 2: Klasa 1 i 3,
- 3: Klasa 1 i 4,
- 4: Klasa 1 i 5,
- 5: Klasa 2 i 3,
- 6: Klasa 2 i 4,
- 7: Klasa 2 i 5,
- 8: Klasa 3 i 4,
- 9: Klasa 3 i 5,
- 10: Klasa 4 i 5.

Celem zastosowania wariantów klasyfikacji jest uzyskanie specjalizacji klasyfikatorów, jak chodzi o wykrywanie różnic pomiędzy dwoma klasami. Proces uczenia każdego wariantu składa się z przetwarzania wstępnego, walidacji krzyżowej, klasyfikacji oraz optymalizacji parametrów. Ocena dokładności klasyfikacji dokonywana jest z wykorzystaniem ważonego *F1 Score*.

Warstwa 3 dokonuje ostatecznej estymacji. Wykorzystano algorytm, który na wejściu otrzymuje wynik klasyfikacji dokonanej w warstwie 1 i 2. Jeśli wskazania klasyfikatorów warstwy 1.1 i 1.2, wyspecjalizowanych w klasyfikacji danej klasy, są identyczne wówczas ostateczną klasą jest ta wskazania przez warstwę 1. Jeśli klasyfikatory wyspecjalizowane w danej klasie wskażą różne klasy, wówczas ostateczna klasyfikacja dokonywana jest w oparciu o wskazanie



RYSUNEK 4.57: Proces ML, algorytm 6

| Przeskalowanie | Estymator        | Parametr 1      | Parametr 2         | Parametr 3        | Parametr 4        | O_ACC = SEN |
|----------------|------------------|-----------------|--------------------|-------------------|-------------------|-------------|
| Normalizer     | Random Forest    | n_estimators=64 | max_depth=20       | random_state=2    | max_samples=56    | 85,00%      |
|                | Linear SVC       | random_state=29 | tol=6.53463e-06    | c=1.378117402     | -                 |             |
|                | C-SVC            | kernel='rbf'    | coef0=0.256643826  | degree=4          | gamma=0.358487459 |             |
|                | Nearest Centroid | -               | -                  | -                 | -                 |             |
|                | Extra Trees      | n_estimators=67 | -                  | -                 | -                 |             |
|                | kNN              | n_neighbors=1   | weights='distance' | algorithm='brute' | leaf_size=33      |             |

TABELA 4.21: Parametry wykorzystane w przypadku, w którym używano najlepszą dokładność klasyfikacji - algorytm 1

warstwy 2. Przykładowo jeśli klasyfikator wyspecjalizowany w klasyfikacji klasy 1 wskaże dla danej próbki klasę 1, a klasyfikator wyspecjalizowany w klasyfikacji klasy 2 wskaże dla tej samej próbki klasę 2, wówczas ostateczna klasa określona zostanie na podstawie wariantu 1 drugiej warstwy klasyfikacji. Jeśli warstwa 1 nie dokona klasyfikacji zgodnej ze specjalizacją, wówczas ostateczna klasyfikacja wykonywana jest w oparciu o większościową wagę klasyfikacji dokonanej przez warstwę 1.

Wyniki klasyfikacji uzyskane z wykorzystaniem przedstawionych algorytmów zostały zaprezentowane w niniejszej części pracy. W tabeli 4.21 zaprezentowano najlepszy wynik SEN uzyskany z wykorzystaniem algorytmu 1. W tabeli przedstawiono również zastosowane parametry. Macierz pomyłek dla algorytmu 1 zaprezentowana została na rysunku 4.58. Zauważyć można, że ten algorytm najlepiej poradził sobie z klasą 1 (93,33%), a najgorzej z klasą 4 (60,00%).

|             |   | Przewidywane |    |   |    |    | 1      |        | 2      |       | 3      |        | 4      |        | 5      |        |        |        |        |        |        |       |         |        |        |        |        |       |        |        |        |        |         |       |         |
|-------------|---|--------------|----|---|----|----|--------|--------|--------|-------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|-------|---------|--------|--------|--------|--------|-------|--------|--------|--------|--------|---------|-------|---------|
|             |   | 1            | 2  | 3 | 4  | 5  |        |        |        |       |        |        |        |        |        |        |        |        |        |        |        |       |         |        |        |        |        |       |        |        |        |        |         |       |         |
| Rzeczywiste | 1 | 14           | 1  | 0 | 0  | 0  | 93,33% | 6,67%  | 17     | 3     | 85,00% | 15,00% | 23     | 3      | 88,46% | 11,54% | 12     | 8      | 60,00% | 40,00% | 19     | 0     | 100,00% | 0,00%  |        |        |        |       |        |        |        |        |         |       |         |
|             | 2 | 2            | 17 | 1 | 0  | 0  | 97,65% | 2,35%  | 2      | 78    | 97,50% | 2,50%  | 6      | 68     | 91,89% | 8,11%  | 2      | 78     | 97,50% | 2,50%  | 3      | 78    | 96,30%  | 3,70%  |        |        |        |       |        |        |        |        |         |       |         |
|             | 3 | 0            | 0  | 5 | 12 | 3  |        |        |        |       |        |        |        |        |        |        |        |        |        |        |        |       |         |        |        |        |        |       |        |        |        |        |         |       |         |
|             | 4 | 0            | 0  | 0 | 0  | 19 |        |        |        |       |        |        |        |        |        |        |        |        |        |        |        |       |         |        |        |        |        |       |        |        |        |        |         |       |         |
|             | 5 | 0            | 0  | 0 | 0  | 0  | 87,50% | 12,50% | 98,81% | 1,19% | 93,33% | 6,67%  | 89,47% | 10,53% | 96,30% | 3,70%  | 85,00% | 15,00% | 79,31% | 20,69% | 95,77% | 4,23% | 88,46%  | 11,54% | 85,71% | 14,29% | 90,70% | 9,30% | 60,00% | 40,00% | 86,36% | 13,64% | 100,00% | 0,00% | 100,00% |

RYSUNEK 4.58: Macierz pomyłek dla najlepszego wyniku klasyfikacji z wykorzystaniem algorytmu 1 (SEN=85,00%). W ciemnoniebieskich komórkach SEN zostało przedstawione dla pojedynczych klas

| Przeskalowanie | Warstwa | Estymator        | Parametr 1       | Parametr 2        | Parametr 3          | Parametr 4        | O_ACC = SEN |
|----------------|---------|------------------|------------------|-------------------|---------------------|-------------------|-------------|
| MaxAbsScaler   | 1       | Linear SVC       | random_state=21  | tol=4.83839e-05   | c=0.982870391       | -                 | 90,00%      |
|                |         | C-SVC            | kernel='poly'    | coef0=0.006357180 | degree=1            | gamma=1.331759915 |             |
|                |         | Nearest Centroid | -                | -                 | -                   | -                 |             |
|                | 2       | kNN              | n_neighbors=3    | weights='uniform' | algorithm='kd_tree' | leaf_size=25      |             |
|                |         | Random Forest    | n_estimators=75  | max_depth=17      | random_state=14     | max_samples=32    |             |
|                |         | Extra Trees      | n_estimators=154 | -                 | -                   | -                 |             |

TABELA 4.22: Parametry wykorzystane w przypadku, w którym uzyskano najlepszą dokładność klasyfikacji - algorytm 2

Z kolei parametry w przypadku najlepszego wyniku uzyskanego z wykorzystaniem algorytmu 2 zaprezentowano w tabeli 4.22. Uzyskano SEN=90,00%. W algorytmie tym wykorzystano dwie warstwy klasyfikacji. Rysunek 4.59 przedstawia macierz pomyłek dla tego algorytmu. Najlepszym rezultatem klasyfikacji było SEN=100,00%, osiągnięte dla klasy 1. Zadawalające wyniki uzyskano również dla klasy 3 i 5. Najgorszy wynik uzyskano dla klasy 4, tj. 70,00%.

Kolejnym algorytmem był algorytm 3. Algorytm ten posiadał 3 warianty. Parametry dla wariantu 1, tj. dwóch warstw klasyfikacji przedstawiono w tabeli 4.23. Uzyskano SEN=91,00%. Na rysunku 4.60 przedstawiono macierz pomyłek. Zauważyć można, że wariant 1 algorytmu 3 dobrze zadziałał dla klasy 1 i 5, tj. SEN=100,00%. Dla klasy 2 i 3 uzyskano wynik powyżej 90,00%. Najgorszy wynik uzyskano dla klasy 4 (75,00%).

Parametry wariantu 2 algorytmu 3 przedstawione zostały w tabeli 4.24. W tym wypadku wykorzystano trzy warstwy klasyfikacji, uzyskano SEN=94,00%. Macierz pomyłek przedstawiono na rysunku 4.61. Najlepszy wynik, tj. 100,00% uzyskano dla klasy 1 i 5. Z kolei dla klasy 2 i 3 uzyskano SEN powyżej 95,00%. Również i w tym przypadku najgorszy wynik uzyskano dla klasy 4 (80,00%).

|             |   | Przewidywane |    |    |    |    | 1       |       | 2       |       | 3       |        | 4      |        | 5      |        |        |       |        |        |        |        |        |       |        |        |        |        |        |       |        |
|-------------|---|--------------|----|----|----|----|---------|-------|---------|-------|---------|--------|--------|--------|--------|--------|--------|-------|--------|--------|--------|--------|--------|-------|--------|--------|--------|--------|--------|-------|--------|
|             |   | 1            | 2  | 3  | 4  | 5  |         |       |         |       |         |        |        |        |        |        |        |       |        |        |        |        |        |       |        |        |        |        |        |       |        |
| Rzeczywiste | 1 | 15           | 0  | 0  | 0  | 0  | 100,00% | 0,00% | 18      | 2     | 90,00%  | 10,00% | 25     | 1      | 96,15% | 3,85%  | 14     | 6     | 70,00% | 30,00% | 18     | 1      | 94,74% | 5,26% |        |        |        |        |        |       |        |
|             | 2 | 0            | 18 | 2  | 0  | 0  | 100,00% | 0,00% | 0       | 80    | 100,00% | 0,00%  | 5      | 69     | 93,24% | 6,76%  | 2      | 78    | 97,50% | 2,50%  | 3      | 78     | 96,30% | 3,70% |        |        |        |        |        |       |        |
|             | 3 | 0            | 0  | 25 | 1  | 0  |         |       |         |       |         |        |        |        |        |        |        |       |        |        |        |        |        |       |        |        |        |        |        |       |        |
|             | 4 | 0            | 0  | 3  | 14 | 3  |         |       |         |       |         |        |        |        |        |        |        |       |        |        |        |        |        |       |        |        |        |        |        |       |        |
|             | 5 | 0            | 0  | 0  | 1  | 18 | 100,00% | 0,00% | 100,00% | 0,00% | 97,56%  | 2,44%  | 90,00% | 10,00% | 83,33% | 16,67% | 98,57% | 1,43% | 96,15% | 3,85%  | 87,50% | 12,50% | 92,86% | 7,14% | 70,00% | 30,00% | 85,71% | 14,29% | 98,73% | 1,27% | 94,74% |

RYSUNEK 4.59: Macierz pomyłek dla najlepszego wyniku klasyfikacji z wykorzystaniem algorytmu 2 (SEN=90,00%). W ciemnoniebieskich komórkach SEN zostało przedstawione dla pojedynczych klas

| Warstwa | Estymator          | Przeskalowanie | Parametr 1       | Parametr 2        | Parametr 3        | Parametr 4        | O_ACC = SEN |
|---------|--------------------|----------------|------------------|-------------------|-------------------|-------------------|-------------|
| 1       | kNN                | MaxAbsScaler   | n_neighbors=3    | weights='uniform' | algorithm='brute' | leaf_size=45      | 91,00%      |
|         | Extra Trees        | Normalizer     | n_estimators=154 | -                 | -                 | -                 |             |
|         | Random Forest      | MaxAbsScaler   | n_estimators=174 | max_depth=12      | random_state=15   | max_samples=18    |             |
|         | Nu-SVC             | MaxAbsScaler   | kernel='rbf'     | nu=0.270444645    | degree=5          | gamma=2.619916864 |             |
| 2       | Nearest Centroid   | Normalizer     | -                | -                 | -                 | -                 |             |
|         | LogisticRegression | -              | -                | -                 | -                 | -                 |             |

TABELA 4.23: Parametry wykorzystane w przypadku, w którym użyto najlepszą dokładność klasyfikacji - algorytm 3 wariant 1

|             |   | Przewidywane |    |         |    |    |        |    |    |        |    |    |        |    |    |         |
|-------------|---|--------------|----|---------|----|----|--------|----|----|--------|----|----|--------|----|----|---------|
|             |   | 1            | 2  | 3       | 4  | 5  |        |    |    |        |    |    |        |    |    |         |
| Rzeczywiste | 1 | 15           | 0  | 100,00% | 18 | 2  | 90,00% | 24 | 2  | 92,31% | 15 | 5  | 75,00% | 19 | 0  | 100,00% |
|             | 2 | 1            | 84 | 98,82%  | 1  | 79 | 98,75% | 4  | 70 | 94,59% | 1  | 79 | 98,75% | 2  | 79 | 97,53%  |
|             | 3 | 0            | 1  | 1,18%   | 0  | 0  | 0,00%  | 0  | 0  | 0,00%  | 0  | 0  | 0,00%  | 0  | 0  | 0,00%   |
|             | 4 | 0            | 0  | 0,00%   | 0  | 0  | 0,00%  | 0  | 0  | 0,00%  | 0  | 0  | 0,00%  | 0  | 0  | 0,00%   |
|             | 5 | 0            | 0  | 0,00%   | 0  | 0  | 0,00%  | 0  | 0  | 0,00%  | 0  | 0  | 0,00%  | 0  | 0  | 0,00%   |

RYSUNEK 4.60: Macierz pomyłek dla najlepszego wyniku klasyfikacji z wykorzystaniem algorytmu 3 wariant 1 (SEN=91,00%). W ciemnoniebieskich komórkach SEN zostało przedstawione dla pojedynczych klas

| Warstwa | Estymator          | Przeskalowanie | Parametr 1       | Parametr 2         | Parametr 3          | Parametr 4        | O_ACC = SEN |
|---------|--------------------|----------------|------------------|--------------------|---------------------|-------------------|-------------|
| 1       | kNN                | MaxAbsScaler   | n_neighbors=3    | weights='uniform'  | algorithm='kd_tree' | leaf_size=36      | 94,00%      |
|         | Extra Trees        | Normalizer     | n_estimators=172 | -                  | -                   | -                 |             |
|         | Random Forest      | MaxAbsScaler   | n_estimators=111 | max_depth=13       | random_state=14     | max_samples=29    |             |
|         | Nu-SVC             | MaxAbsScaler   | kernel='poly'    | nu=0.435555251     | degree=3            | gamma=2.384872936 |             |
| 2       | Nearest Centroid   | Normalizer     | -                | -                  | -                   | -                 |             |
| 3       | Random Forest      | MaxAbsScaler   | n_estimators=193 | max_depth=4        | random_state=14     | max_samples=9     |             |
|         | Nu-SVC             | MinMaxScaler   | kernel='poly'    | nu=0.069852823     | degree=1            | gamma=2.295198665 |             |
|         | kNN                | MaxAbsScaler   | n_neighbors=3    | weights='distance' | algorithm='kd_tree' | leaf_size=33      |             |
|         | LogisticRegression | -              | -                | -                  | -                   | -                 |             |

TABELA 4.24: Parametry wykorzystane w przypadku, w którym użyto najlepszą dokładność klasyfikacji - algorytm 3 wariant 2

|             |   | Przewidywane |    |         |    |    |        |    |    |        |    |    |         |    |    |         |
|-------------|---|--------------|----|---------|----|----|--------|----|----|--------|----|----|---------|----|----|---------|
|             |   | 1            | 2  | 3       | 4  | 5  |        |    |    |        |    |    |         |    |    |         |
| Rzeczywiste | 1 | 15           | 0  | 100,00% | 19 | 1  | 95,00% | 25 | 1  | 96,15% | 16 | 4  | 80,00%  | 19 | 0  | 100,00% |
|             | 2 | 1            | 84 | 98,82%  | 1  | 79 | 98,75% | 3  | 71 | 95,95% | 0  | 80 | 100,00% | 1  | 80 | 98,77%  |
|             | 3 | 0            | 1  | 1,18%   | 0  | 0  | 0,00%  | 0  | 0  | 0,00%  | 0  | 0  | 0,00%   | 0  | 0  | 0,00%   |
|             | 4 | 0            | 0  | 0,00%   | 0  | 0  | 0,00%  | 0  | 0  | 0,00%  | 0  | 0  | 0,00%   | 0  | 0  | 0,00%   |
|             | 5 | 0            | 0  | 0,00%   | 0  | 0  | 0,00%  | 0  | 0  | 0,00%  | 0  | 0  | 0,00%   | 0  | 0  | 0,00%   |

RYSUNEK 4.61: Macierz pomyłek dla najlepszego wyniku klasyfikacji z wykorzystaniem algorytmu 3 wariant 2 (SEN=94,00%). W ciemnoniebieskich komórkach SEN zostało przedstawione dla pojedynczych klas

| Warstwa | Estymator          | Przeskalowanie | Parametr 1       | Parametr 2         | Parametr 3          | Parametr 4        | O_ACC = SEN |
|---------|--------------------|----------------|------------------|--------------------|---------------------|-------------------|-------------|
| 1       | kNN                | MaxAbsScaler   | n_neighbors=4    | weights='distance' | algorithm='kd_tree' | leaf_size=43      | 94,00%      |
|         | Extra Trees        | Normalizer     | n_estimators=126 | -                  | -                   | -                 |             |
|         | Random Forest      | MaxAbsScaler   | n_estimators=175 | max_depth=8        | random_state=20     | max_samples=63    |             |
|         | Nu-SVC             | MaxAbsScaler   | kernel='rbf'     | nu=0.228686519     | degree=4            | gamma=0.110472750 |             |
|         | Nearest Centroid   | Normalizer     | -                | -                  | -                   | -                 |             |
| 2       | Random Forest      | MaxAbsScaler   | n_estimators=57  | max_depth=9        | random_state=10     | max_samples=14    |             |
|         | Nu-SVC             | MinMaxScaler   | kernel='rbf'     | nu=0.084774299     | 3                   | gamma=0.944700151 |             |
| 3       | kNN                | MaxAbsScaler   | n_neighbors=7    | weights='distance' | algorithm='brute'   | leaf_size=17      |             |
|         | Random Forest      | MaxAbsScaler   | n_estimators=196 | max_depth=7        | random_state=7      | max_samples=39    |             |
| 4       | Nu-SVC             | MinMaxScaler   | kernel='rbf'     | nu=0.467701715     | degree=3            | gamma=0.312846365 |             |
|         | LogisticRegression | -              | -                | -                  | -                   | -                 |             |

TABELA 4.25: Parametry wykorzystane w przypadku, w którym uzyskano najlepszą dokładność klasyfikacji - algorytm 3 wariant 3

| Przewidywane |   | 1      |         |         |        |        | 2      |         |         |        |        | 3      |         |         |        |        | 4      |         |         |        |        | 5      |         |         |        |        |
|--------------|---|--------|---------|---------|--------|--------|--------|---------|---------|--------|--------|--------|---------|---------|--------|--------|--------|---------|---------|--------|--------|--------|---------|---------|--------|--------|
|              |   | 1      | 2       | 3       | 4      | 5      | 1      | 2       | 3       | 4      | 5      | 1      | 2       | 3       | 4      | 5      | 1      | 2       | 3       | 4      | 5      | 1      | 2       | 3       | 4      | 5      |
| Rzeczywiste  | 1 | 15     | 0       | 0       | 0      | 0      | 15     | 0       | 0       | 0      | 0      | 15     | 0       | 0       | 0      | 0      | 15     | 0       | 0       | 0      | 0      | 15     | 0       | 0       | 0      | 0      |
|              | 2 | 1      | 19      | 0       | 0      | 0      | 1      | 19      | 0       | 0      | 0      | 1      | 19      | 0       | 0      | 0      | 1      | 19      | 0       | 0      | 0      | 1      | 19      | 0       | 0      | 0      |
|              | 3 | 0      | 1       | 25      | 0      | 0      | 0      | 1       | 25      | 0      | 0      | 0      | 1       | 25      | 0      | 0      | 0      | 1       | 25      | 0      | 0      | 0      | 1       | 25      | 0      | 0      |
|              | 4 | 0      | 0       | 3       | 16     | 1      | 0      | 0       | 3       | 16     | 1      | 0      | 0       | 3       | 16     | 1      | 0      | 0       | 3       | 16     | 1      | 0      | 0       | 3       | 16     | 1      |
|              | 5 | 0      | 0       | 0       | 0      | 19     | 0      | 0       | 0       | 0      | 19     | 0      | 0       | 0       | 0      | 19     | 0      | 0       | 0       | 0      | 19     | 0      | 0       | 0       | 0      | 19     |
|              |   | 93,75% | 100,00% | 100,00% | 95,00% | 98,75% | 93,75% | 100,00% | 100,00% | 95,00% | 98,75% | 93,75% | 100,00% | 100,00% | 95,00% | 98,75% | 93,75% | 100,00% | 100,00% | 95,00% | 98,75% | 93,75% | 100,00% | 100,00% | 95,00% | 98,75% |
|              |   | 6,25%  | 0,00%   | 0,00%   | 5,00%  | 1,25%  | 6,25%  | 0,00%   | 0,00%   | 5,00%  | 1,25%  | 6,25%  | 0,00%   | 0,00%   | 5,00%  | 1,25%  | 6,25%  | 0,00%   | 0,00%   | 5,00%  | 1,25%  | 6,25%  | 0,00%   | 0,00%   | 5,00%  | 1,25%  |

RYСУNEK 4.62: Macierz pomyłek dla najlepszego wyniku klasyfikacji z wykorzystaniem algorytmu 3 wariant 3 (SEN=94,00%). W ciemnoniebieskich komórkach SEN zostało przedstawione dla pojedynczych klas

W tabeli 4.25 przedstawiono parametry wykorzystane dla najlepszego wyniku wariantu 3 algorytmu 3. W tym wariantcie wykorzystano cztery warstwy klasyfikacji i uzyskano SEN=94,00%, co jest takim samym wynikiem jak ten uzyskany w przypadku wykorzystania wariantu 2 algorytmu 3. Rysunek 4.62 przedstawia macierz pomyłek. Najwyższą wartość SEN uzyskano dla klasy 1 i 5, tj. 100,00%, najniższą dla klasy 4, tj. 80,00%.

Parametry kolejnego algorytmu, tj. algorytmu 4 przedstawiono w tabeli 4.26. Dla klasy 1, 2 i 3 uzyskano wynik powyżej 90,00% co pokazano na rysunku 4.63. Najgorszym wynikiem było 65,00% uzyskane dla klasy 4.

W przypadku algorytmu 5, SEN=87,00% uzyskano przy wykorzystaniu parametrów przedstawionych w tabeli 4.27. Macierz pomyłek zaprezentowano na rysunku 4.64. Zauważono, że algorytm ten najlepiej poradził sobie z klasyfikacją próbek klasy 1 i 5 (100,00%), a najgorzej dla klasy 4 (65,00%).

Ostatnim badanym algorytmem był algorytm 6. Dla najlepszego przypadku, tj. SEN=86,00% parametry przedstawiono w tabeli 4.28. Z kolei macierz pomyłek zaprezentowano na rysunku 4.65. Algorytm poradził sobie najlepiej z klasą 3 i 5

| Estymator        | Przeskalowanie | Parametr 1       | Parametr 2        | Parametr 3        | Parametr 4        | O_ACC = SEN |
|------------------|----------------|------------------|-------------------|-------------------|-------------------|-------------|
| kNN              | MaxAbsScaler   | n_neighbors=3    | weights='uniform' | algorithm='brute' | leaf_size=15      | 85,00%      |
| Extra Trees      | Normalizer     | n_estimators=74  | -                 | -                 | -                 |             |
| Random Forest    | MaxAbsScaler   | n_estimators=113 | max_depth=2       | random_state=13   | max_samples=27    |             |
| Nu-SVC           | MinMaxScaler   | kernel='sigmoid' | nu=0.196159703    | degree=7          | gamma=2.363264651 |             |
| Nearest Centroid | Normalizer     | -                | -                 | -                 | -                 |             |

TABELA 4.26: Parametry wykorzystane w przypadku, w którym uzyskano najlepszą dokładność klasyfikacji - algorytm 4

|             |   | Przewidywane |    |   |    |    | 1      |        | 2      |        | 3      |        | 4      |        | 5      |        |        |        |        |        |        |        |        |        |
|-------------|---|--------------|----|---|----|----|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
|             |   | 1            | 2  | 3 | 4  | 5  |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |
| Rzeczywiste | 1 | 14           | 1  | 0 | 0  | 0  | 93,33% | 6,67%  | 18     | 2      | 90,00% | 10,00% | 24     | 2      | 92,31% | 7,69%  | 13     | 7      | 65,00% | 35,00% | 16     | 3      | 84,21% | 15,79% |
|             | 2 | 1            | 18 | 1 | 0  | 0  | 98,82% | 1,18%  | 2      | 78     | 97,50% | 2,50%  | 4      | 70     | 94,59% | 5,41%  | 4      | 76     | 95,00% | 5,00%  | 4      | 77     | 95,06% | 4,94%  |
|             | 3 | 0            | 0  | 3 | 13 | 4  | 93,33% | 98,82% | 90,00% | 97,50% | 90,00% | 85,71% | 97,22% | 92,31% | 76,47% | 91,57% | 65,00% | 80,00% | 96,25% | 84,21% | 20,00% | 96,25% | 3,75%  | 15,79% |
|             | 4 | 0            | 0  | 0 | 3  | 16 | 6,67%  | 1,18%  | 6,67%  | 10,00% | 2,50%  | 10,00% | 14,29% | 2,78%  | 7,69%  | 23,53% | 8,43%  | 35,00% | 20,00% | 3,75%  | 15,79% | 15,79% |        |        |
|             | 5 | 0            | 0  | 0 | 0  | 16 |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |

RYSUNEK 4.63: Macierz pomyłek dla najlepszego wyniku klasyfikacji z wykorzystaniem algorytmu 4 (SEN=85,00%). W ciemnoniebieskich komórkach SEN zostało przedstawione dla pojedynczych klas

| Estymator        | Przeskalowanie | Parametr 1       | Parametr 2        | Parametr 3        | Parametr 4        | O_ACC = SEN |
|------------------|----------------|------------------|-------------------|-------------------|-------------------|-------------|
| kNN              | MaxAbsScaler   | n_neighbors=3    | weights='uniform' | algorithm='brute' | leaf_size=21      | 87,00%      |
| Extra Trees      | Normalizer     | n_estimators=102 | -                 | -                 | -                 |             |
| Random Forest    | MaxAbsScaler   | n_estimators=72  | max_depth=2       | random_state=16   | max_samples=14    |             |
| Nu-SVC           | MinMaxScaler   | kernel='poly'    | nu=0.303701374    | degree=4          | gamma=2.388222064 |             |
| Nearest Centroid | Normalizer     | -                | -                 | -                 | -                 |             |

TABELA 4.27: Parametry wykorzystane w przypadku, w którym uzyskano najlepszą dokładność klasyfikacji - algorytm 5

(100,00%), dla klas 1 i 2 uzyskany wynik wyniósł około 85,00%, a najgorszy rezultat uzyskano dla klasy 4 (55,00%).

Na rysunku 4.66 przedstawiono SEN uzyskane dla poszczególnych algorytmów etapu II.

Zestawienie wszystkich uzyskanych wyników (oba badania) przedstawiono na rysunku 4.67. W przypadku etapu II zaproponowano 6 nowych metod uczenia. Uzyskane wyniki są w większości przypadków lepsze niż w poprzednich badaniach. W poprzednich badaniach najlepszy uzyskany wynik to SEN = 89,00%. W etapie II algorytmy 1, 4, 5 i 6 uzyskały gorsze wyniki w porównaniu do poprzednich badań. Algorytmy, które wypadły lepiej niż najlepszy pojedynczy klasyfikator w poprzednich badaniach, to algorytmy 2 i 3 (we wszystkich wariantach).

### 4.4.3 Etap III

Na tym etapie badań zaproponowano nową metodę klasyfikacji, która stanowi fuzję sieci neuronowej (NN) i GA. Proponowany algorytm został przedstawiony na rysunku 4.68. Został podzielony na pięć kroków:

|             |   | Przewidywane |    |   |    |    | 1       |         | 2      |        | 3      |        | 4      |        | 5      |        |        |        |         |         |        |       |         |       |
|-------------|---|--------------|----|---|----|----|---------|---------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|---------|---------|--------|-------|---------|-------|
|             |   | 1            | 2  | 3 | 4  | 5  |         |         |        |        |        |        |        |        |        |        |        |        |         |         |        |       |         |       |
| Rzeczywiste | 1 | 15           | 0  | 0 | 0  | 0  | 100,00% | 0,00%   | 16     | 4      | 80,00% | 20,00% | 24     | 2      | 92,31% | 7,69%  | 13     | 7      | 65,00%  | 35,00%  | 19     | 0     | 100,00% | 0,00% |
|             | 2 | 2            | 16 | 2 | 0  | 0  | 97,65%  | 2,35%   | 1      | 79     | 98,75% | 1,25%  | 5      | 69     | 93,24% | 6,76%  | 1      | 79     | 98,75%  | 1,25%   | 4      | 77    | 95,06%  | 4,94% |
|             | 3 | 0            | 0  | 3 | 13 | 4  | 88,24%  | 100,00% | 94,12% | 95,18% | 80,00% | 82,76% | 97,18% | 92,31% | 92,86% | 91,86% | 65,00% | 82,61% | 100,00% | 100,00% | 11,76% | 0,00% | 0,00%   | 0,00% |
|             | 4 | 0            | 0  | 0 | 3  | 19 | 11,76%  | 0,00%   | 5,88%  | 4,82%  | 20,00% | 17,24% | 2,82%  | 7,69%  | 7,14%  | 8,14%  | 35,00% | 17,39% | 0,00%   | 0,00%   | 0,00%  | 0,00% |         |       |
|             | 5 | 0            | 0  | 0 | 0  | 19 |         |         |        |        |        |        |        |        |        |        |        |        |         |         |        |       |         |       |

RYSUNEK 4.64: Macierz pomyłek dla najlepszego wyniku klasyfikacji z wykorzystaniem algorytmu 5 (SEN=87,00%). W ciemnoniebieskich komórkach SEN zostało przedstawione dla pojedynczych klas



| Warstwa | Estymator        | Przeskalowanie | Parametr 1         | Parametr 2            | Parametr 3          | Parametr 4         | O_ACC = SEN |
|---------|------------------|----------------|--------------------|-----------------------|---------------------|--------------------|-------------|
| L1.1    | kNN              | MaxAbsScaler   | n_neighbors=3      | weights='uniform'     | algorithm='brute'   | leaf_size=17       | 86,00%      |
|         | Extra Trees      | Normalizer     | n_estimators=99    | -                     | -                   | -                  |             |
|         | Random Forest    | MaxAbsScaler   | n_estimators=158   | max_depth=2           | random_state=3      | max_samples=44     |             |
|         | Nu-SVC           | MinMaxScaler   | kernel='sigmoid'   | nu=0.241815544        | degree=2            | gamma=2.124638721  |             |
| L1.2    | Nearest Centroid | Normalizer     | -                  | -                     | -                   | -                  |             |
|         | Linear SVC       | Normalizer     | random_state=29    | tol=6.10873e-05       | c=1.290368551       | -                  |             |
|         | Nu-SVC           | MinMaxScaler   | kernel='linear'    | nu=0.178280458        | degree=4            | gamma=2.048582813  |             |
|         | kNN              | MaxAbsScaler   | n_neighbors=4      | weights='uniform'     | algorithm='brute'   | leaf_size=15       |             |
|         | Random Forest    | MaxAbsScaler   | n_estimators=92    | max_depth=4           | random_state=10     | max_samples=43     |             |
|         | Radius Neighbors | Normalizer     | radius=6.872890495 | algorithm='ball_tree' | leaf_size=30        | weights='distance' |             |
| L2.1    | Nu-SVC           | MinMaxScaler   | kernel='linear'    | nu=0.176140558        | degree=5            | gamma=3.438696991  |             |
| L2.2    | kNN              | MaxAbsScaler   | n_neighbors=4      | weights='uniform'     | algorithm='kd_tree' | leaf_size=27       |             |
| L2.3    | Nu-SVC           | MinMaxScaler   | kernel='linear'    | nu=0.178148738        | degree=1            | gamma=1.161476259  |             |
| L2.4    | Nearest Centroid | Normalizer     | -                  | -                     | -                   | -                  |             |
| L2.5    | Radius Neighbors | Normalizer     | radius=7.179069875 | algorithm='ball_tree' | leaf_size=36        | weights='distance' |             |
| L2.6    | Nu-SVC           | MinMaxScaler   | kernel='rbf'       | nu=0.009248084        | degree=7            | gamma=0.051971845  |             |
| L2.7    | Radius Neighbors | Normalizer     | radius=5.109086106 | algorithm='brute'     | leaf_size=45        | weights='distance' |             |
| L2.8    | Nu-SVC           | MinMaxScaler   | kernel='sigmoid'   | nu=0.302767450        | degree=4            | gamma=2.735249768  |             |
| L2.9    | Linear SVC       | Normalizer     | random_state=0     | tol=4.40873e-05       | c=1.802774393       | -                  |             |
| L2.10   | Nu-SVC           | MinMaxScaler   | kernel='poly'      | nu=0.375384705        | degree=4            | gamma=0.645935745  |             |

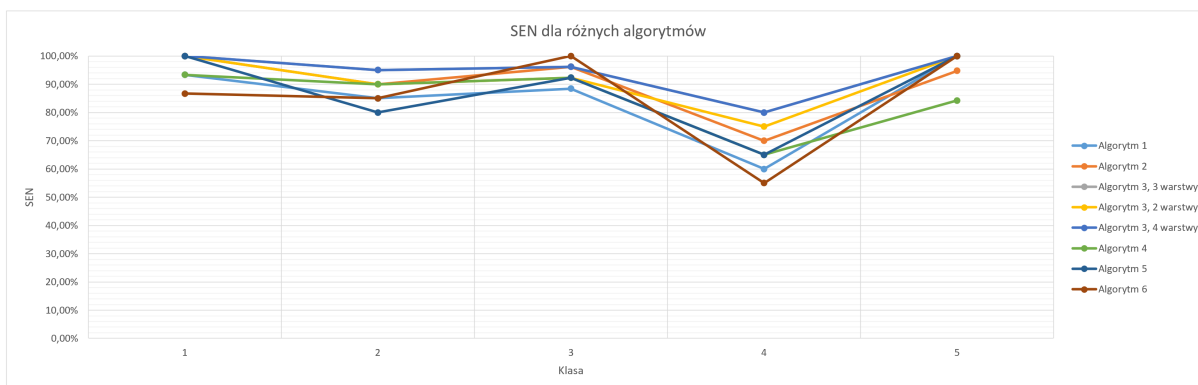
TABELA 4.28: Parametry wykorzystane w przypadku, w którym użyto najlepszą dokładność klasyfikacji - algorytm 6

|             |   | Przewidywane |    |    |    |    |
|-------------|---|--------------|----|----|----|----|
|             |   | 1            | 2  | 3  | 4  | 5  |
| Rzeczywiste | 1 | 13           | 2  | 17 | 0  | 0  |
|             | 2 | 1            | 84 | 2  | 78 | 0  |
|             | 3 | 0            | 0  | 3  | 11 | 6  |
|             | 4 | 0            | 0  | 0  | 19 | 0  |
|             | 5 | 0            | 0  | 0  | 0  | 19 |

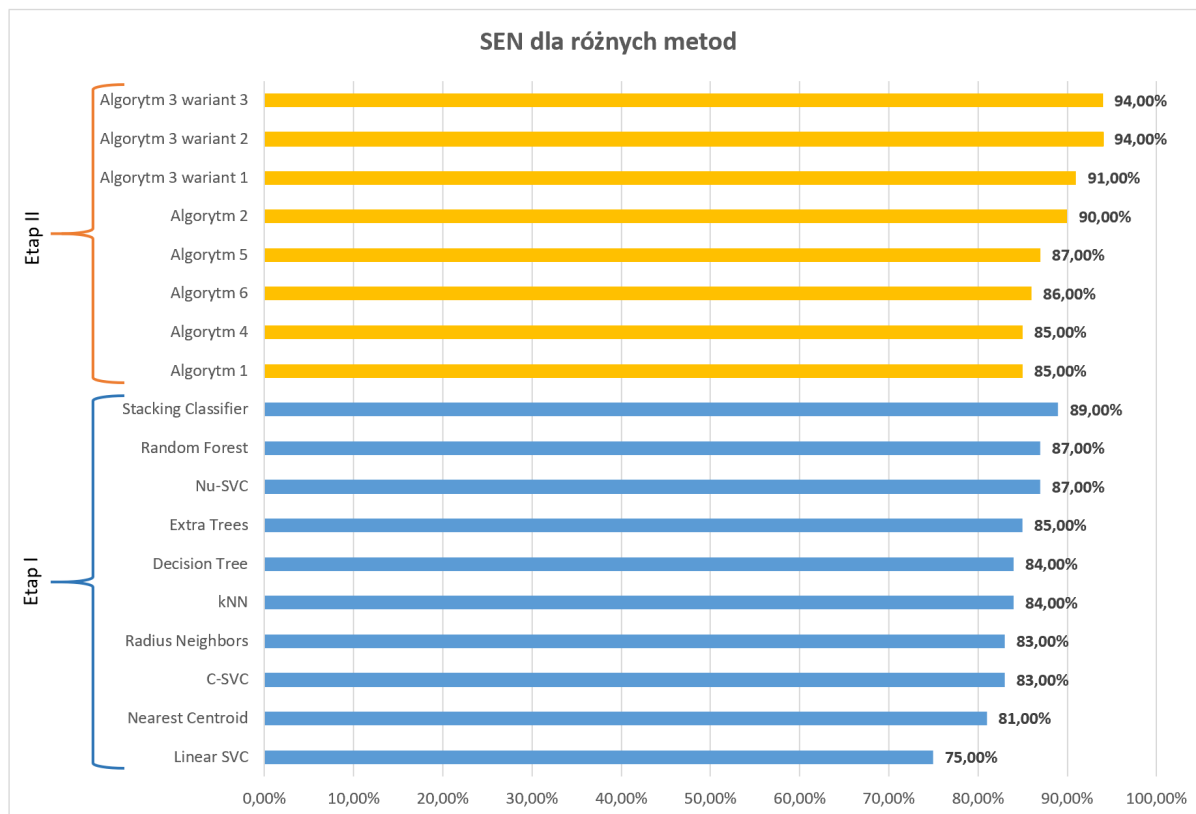
  

|        |        | 1      | 2      | 3       | 4       | 5       |
|--------|--------|--------|--------|---------|---------|---------|
| 86,67% | 13,33% | 85,00% | 15,00% | 100,00% | 0,00%   | 55,00%  |
| 98,82% | 1,18%  | 97,50% | 2,50%  | 93,24%  | 6,76%   | 100,00% |
| 92,86% | 97,67% | 86,67% | 89,47% | 96,30%  | 85,00%  | 83,87%  |
| 7,14%  | 2,33%  | 13,33% | 10,53% | 3,70%   | 15,00%  | 16,13%  |
| 0,00%  | 0,00%  | 0,00%  | 0,00%  | 0,00%   | 100,00% | 100,00% |
| 0,00%  | 0,00%  | 0,00%  | 0,00%  | 0,00%   | 0,00%   | 100,00% |
| 0,00%  | 0,00%  | 0,00%  | 0,00%  | 0,00%   | 0,00%   | 100,00% |
| 0,00%  | 0,00%  | 0,00%  | 0,00%  | 0,00%   | 0,00%   | 100,00% |
| 0,00%  | 0,00%  | 0,00%  | 0,00%  | 0,00%   | 0,00%   | 100,00% |

RYSUNEK 4.65: Macierz pomyłek dla najlepszego wyniku klasyfikacji z wykorzystaniem algorytmu 6 (SEN=86,00%). W ciemnoniebieskich komórkach SEN zostało przedstawione dla pojedynczych klas



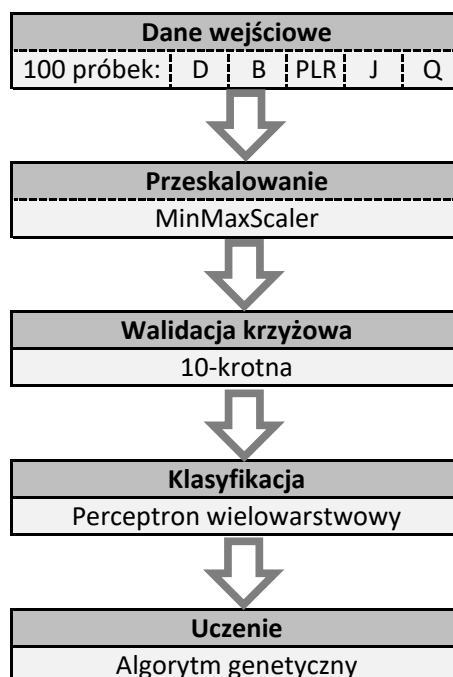
RYSUNEK 4.66: SEN dla poszczególnych algorytmów (etap II) i klas



RYSUNEK 4.67: SEN dla różnych metod (etap I i etap II)

1. *Odczyt danych wejściowych*: Dane wejściowe składały się ze 100 próbek - takich samych jak w przypadku poprzednich etapów badań. Jedna próbka składała się z wartości opóźnienia transmisji (D), przepływności (B), współczynnika utraty pakietów (PLR), fluktuacji opóźnienia (J) i etykiety jakości (Q).
2. *Przeskalowanie*: Zastosowano jeden algorytm przeskalowania, tj. *min-max scaler*. Dzięki zastosowaniu tego algorytmu uzyskano dane w określonym przedziale wartości. Prezentowana metoda wykorzystuje skalowanie do wartości z zakresu 0–1.
3. *Walidacja krzyżowa*: Zastosowano 10-krotną stratyfikowaną walidację krzyżową. Oznacza to, że cały zbiór próbek został podzielony na dziesięć podzbiorów. Dziewięć z tych podzbiorów wykorzystano na etapie uczenia NN i optymalizacji parametrów za pomocą GA. Do weryfikacji precyzji klasyfikacji wykorzystano jeden podzbiór.
4. *Klasyfikacja*: Na potrzeby klasyfikacji zastosowano wielowarstwowy perceptron (z ang. multilayer perceptron, MLP).
5. *Uczenie*: Do optymalizacji parametrów wykorzystano GA.

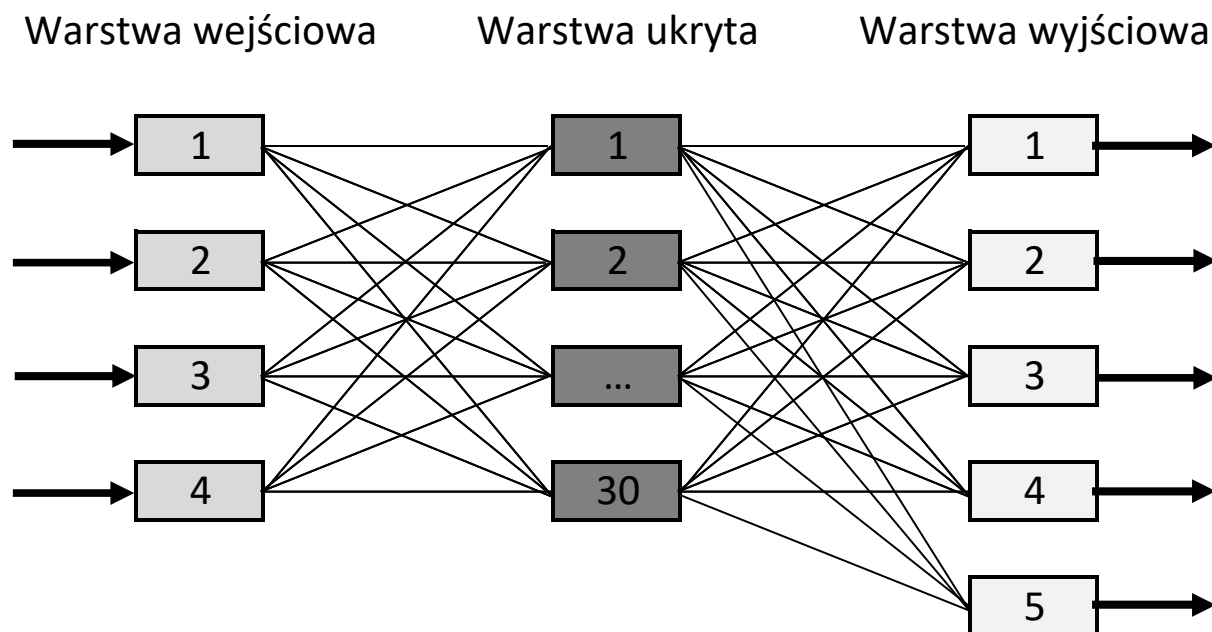




RYSUNEK 4.68: Proces ML

Sztuczna sieć neuronowa (z ang. artificial neural network, ANN) składa się z warstwy wejściowej, określonej liczby warstw ukrytych i warstwy wyjściowej. Podstawowym elementem sieci neuronowej jest sztuczny neuron. W badaniach użyto standardowej jednokierunkowej sieci neuronowej (z ang. feed-forward neural network). Neurony warstwy wejściowej połączone są z neuronami warstwy ukrytej, neurony warstwy ukrytej z neuronami warstwy wyjściowej. Każde z tych połączeń ma określoną wagę —najczęściej w zakresie od  $-1$  do  $1$ . Celem uczenia sieci neuronowej jest określenie wag na zbiorze uczącym w taki sposób, aby sieć skutecznie klasyfikowała próbki na zestawie testowym. Najpopularniejszą metodą wyboru wag jest metoda wstecznej propagacji błędów oparta na metodach gradientowych. Przykładami stosowanych algorytmów są *lbfgs*, *sgd*, *adam*. Błąd średniokwadratowy (z ang. mean square error, MSE) jest najczęściej używany jako miara oceny sieci. MSE można obliczyć tak, jak pokazano na równaniu 4.5. Każdy z tych algorytmów ma dodatkowe parametry, takie jak *momentum*, szybkość i liczbę epok uczenia się, które należy dodatkowo skonfigurować. Ponadto należy wybrać architekturę sieci, czyli w szczególności liczbę warstw ukrytych i liczbę neuronów w warstwach ukrytych. To klasyczne podejście nie zawsze daje dobre wyniki, ponieważ algorytmy gradientowe mają skłonność do utknięcia w lokalnych minimach. W rezultacie sieć neuronowa nie jest w stanie dokładnie się uczyć.

$$MSE = \frac{1}{2} \sum_{i=1}^n (z_i - y_i)^2 \quad (4.5)$$



RYSUNEK 4.69: Model jednokierunkowej sieci neuronowej

gdzie:

- $n$ —liczba zbiorów testowych,
- $z_i$ —oczekiwana odpowiedź dla próbki  $i$ ,
- $y_i$ —odpowiedź perceptronu dla próbki  $i$ .

Na tym etapie badań zaproponowano metodę opartą na fuzji NN i GA. Wykorzystanie połączenia NN i GA jest oryginalnym rozwiązaniem w kontekście klasyfikacji jakości transmisji danych. Istnieją trzy sposoby łączenia GA i NN razem:

1. Wybór architektury NN poprzez obliczenia ewolucyjne.
2. Znalezienie odpowiedniego zestawu wag NN za pomocą GA.
3. Kombinacja podejścia 1 i 2.

W badaniach zastosowano podejście (2). W pierwszym kroku eksperymentalnie wybrano architekturę NN, a następnie korzystając z obliczeń ewolucyjnych, wybrano najlepszy zestaw wag.

Wykorzystano język programowania Python, biblioteki Sklearn i Deap. Wyniki przedstawiono dla zbioru danych testowych. W eksperymencie przetestowano różne architektury sieci NN. Położono nacisk na modyfikację liczby neuronów w warstwie ukrytej. Przetestowano 5, 10, 15, 20, 25, 30, 40 i 50 neuronów w warstwie ukrytej. Najlepsze rezultaty uzyskano stosując 30 neuronów w warstwie ukrytej.

Warstwa wejściowa składała się z czterech neuronów, a warstwa wyjściowa z pięciu neuronów (rysunek 4.69). Cała sieć wymagała optymalizacji 305 parametrów (wagi i biasy). To pokazuje stopień zaawansowania problemu optymalizacji. Prze-testowano również różne warianty konfiguracji GA. W testach zmieniono:

- Rozmiar populacji - [100, 200, 300, 400, 500, 1000, 2000, 3000];
- Liczbę epok - [100, 200, 300, 400, 500, 1000, 2000, 3000];
- Typ krzyżowania - *two-point, uniform, arithmetic*;
- Typ mutacji - jedno, dwu i trzy punktowa.

Najlepszą konfiguracją była:

- Rozmiar populacji - 300;
- Liczba epok - 2000;
- Krzyżowanie *uniform*;
- Mutacja dwu punktowa;
- *Elitism* - jeden najlepszy osobnik przechodzi do następnej epoki;
- Selekcja turniejowa z rozmiarem 3.

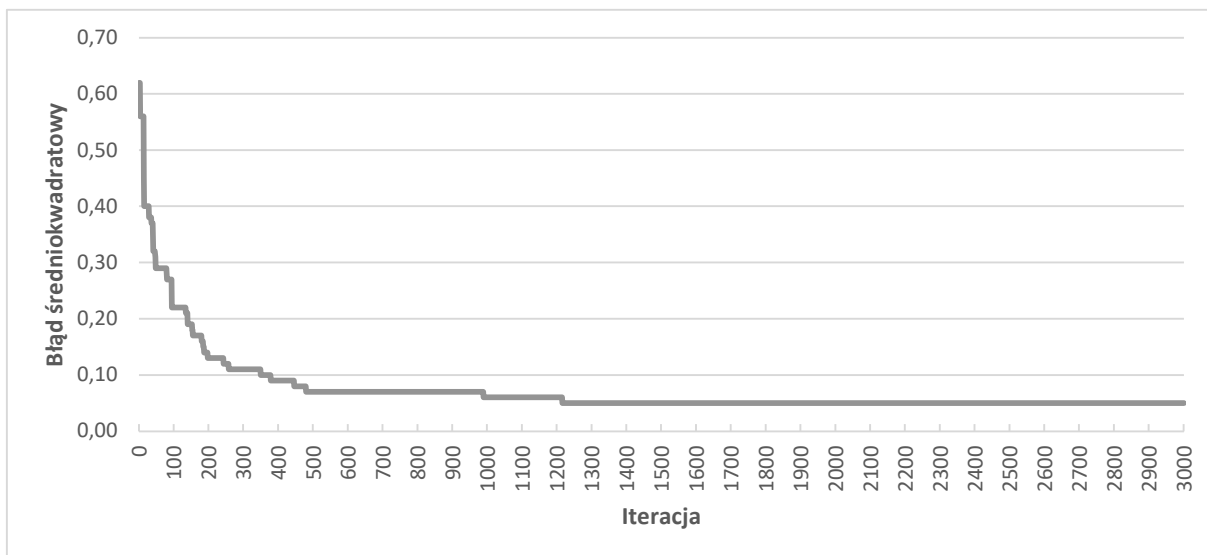
Metodę nazwano GA-MLP. Najlepszy uzyskany wynik to SEN=95%. Rysunek 4.70 przedstawia macierz pomyłek dla proponowanego algorytmu oraz wyniki klasyfikacji uzyskane w każdej klasie. Można zauważyć, że w przypadku dwóch klas, tj. 1 i 3, algorytm osiągnął SEN=100%. Z kolei w klasie 2 wynik wyniósł 95%. W klasie 5 było to nieco mniej, tj. 94,74%. Algorytm sprawdził się najgorzej w klasie 4 uzyskując wynik równy 85%. Z macierzy pomyłek można wyczytać, że występujące błędy klasyfikacji zawsze dotyczyły błędu rzędu  $\pm 1$  klasy w stosunku do wartości oczekiwanej. Oznacza to, że w przypadku klasyfikacji jakości transmisji danych, klient może w najgorszym przypadku uzyskać jakość gorszą lub lepszą o 1 od rzeczywistej. Uzyskany SEN wydaje się być zadowalający.

Rysunek 4.71 przedstawia wykres zbieżności. Można zauważyć zadowalającą zależność, tj. w kolejnych epokach błąd klasyfikacji maleje. Jest to bardzo dobra zależność, dowodząca, że algorytm działa poprawnie zarówno w zakresie klasyfikacji, jak i uczenia się.

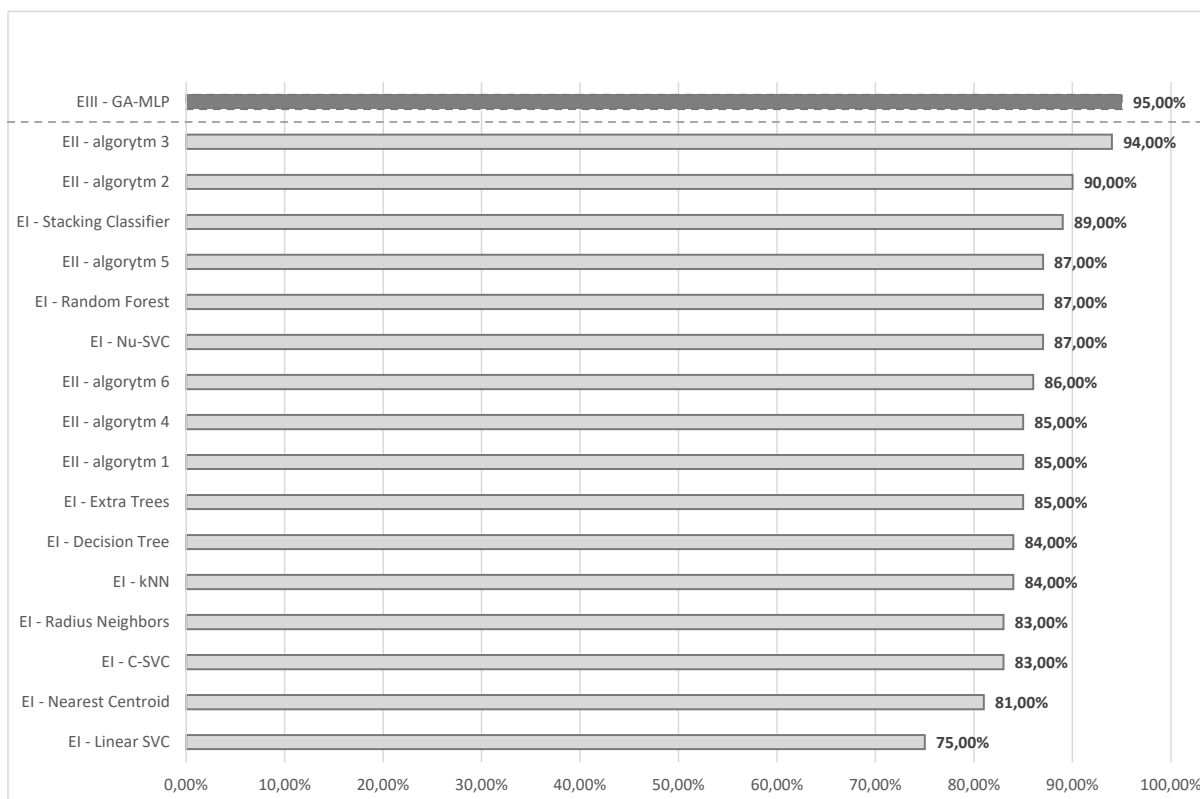
Na rysunku 4.72 i 4.73 podsumowano badania etapu I, II i III. Rysunek 4.72 stanowi zestawienie wszystkich wartości SEN we wszystkich dotychczas przeprowadzonych badaniach. Można zauważyć, że metoda zaproponowana w etapie III uzyskała najwyższą wartość SEN. Metody z wcześniejszych etapów, uzyskały gorsze

|             |   | Przewidywane |         |         |         |        |        |        |         |         |        |         |        |        |        |        |        |        |
|-------------|---|--------------|---------|---------|---------|--------|--------|--------|---------|---------|--------|---------|--------|--------|--------|--------|--------|--------|
|             |   | 1            | 2       | 3       | 4       | 5      | 1      | 2      | 3       | 4       | 5      | 1       | 2      | 3      | 4      | 5      |        |        |
| Rzeczywiste | 1 | 15           | 0       | 0       | 0       | 0      | 15     | 0      | 0       | 0       | 0      | 100,00% | 0,00%  | 0,00%  | 0,00%  | 0,00%  | 0,00%  |        |
|             | 2 | 0            | 19      | 1       | 0       | 0      | 0      | 19     | 1       | 0       | 0      | 100,00% | 5,00%  | 5,00%  | 0,00%  | 0,00%  | 0,00%  |        |
|             | 3 | 0            | 0       | 26      | 0       | 0      | 0      | 0      | 26      | 0       | 0      | 100,00% | 0,00%  | 0,00%  | 0,00%  | 0,00%  | 0,00%  |        |
|             | 4 | 0            | 0       | 2       | 17      | 1      | 0      | 0      | 3       | 71      | 1      | 95,95%  | 4,05%  | 4,05%  | 1,25%  | 1,25%  | 1,23%  |        |
|             | 5 | 0            | 0       | 0       | 1       | 18     | 0      | 0      | 0       | 0       | 18     | 100,00% | 0,00%  | 0,00%  | 0,00%  | 0,00%  | 0,00%  | 0,00%  |
|             |   | 100,00%      | 100,00% | 100,00% | 100,00% | 98,77% | 95,00% | 89,66% | 100,00% | 100,00% | 94,44% | 96,34%  | 85,00% | 94,74% | 98,77% | 94,74% | 98,77% | 94,74% |
|             |   | 0,00%        | 0,00%   | 0,00%   | 0,00%   | 1,23%  | 5,00%  | 10,34% | 0,00%   | 0,00%   | 5,56%  | 3,66%   | 15,00% | 5,26%  | 1,23%  | 5,26%  | 5,26%  | 5,26%  |

RYSUNEK 4.70: Macierz pomyłek dla najlepszego wyniku klasyfikacji (SEN=95,00%). Jasnoszara komórka – SEN obliczone dla pojedynczej klasy



RYSUNEK 4.71: Zmienność błędu średniokwadratowego dla różnych iteracji (epok) GA



RYSUNEK 4.72: SEN dla różnych metod, EI - etap I, EII - etap II, EIII - etap III

wyniki niż GA-MLP. Oznacza to, że biorąc pod uwagę tylko SEN, metoda zaproponowana w etapie III jest najlepsza. Jednak aby uzyskać pełny obraz uzyskanych wyników na rysunku 4.73 przedstawiono wartości SEN poszczególnych algorytmów w każdej klasie. Pogrubiona linia obrazuje najlepsze wyniki dla różnych klas. Można zauważyć, że GA-MLP uzyskał najlepsze wyniki w trzech klasach. W klasie, która okazała się najbardziej problematyczna (biorąc pod uwagę wszystkie wyniki), czyli w klasie 4 algorytm uzyskał drugi najlepszy wynik. Niestety w klasie 5 wynik nie jest zadowalający. Znaczna część algorytmów z wcześniejszych etapów badań uzyskała w tej klasie SEN=100%. Biorąc pod uwagę wyniki uzyskane we wszystkich klasach, można stwierdzić, że algorytm GA-MLP uzyskał najlepszy wynik klasyfikacji.

|                          | Klasa 1        | Klasa 2       | Klasa 3        | Klasa 4       | Klasa 5        | Średnia       |
|--------------------------|----------------|---------------|----------------|---------------|----------------|---------------|
| <b>EIII - GA-MLP</b>     | <b>100,00%</b> | <b>95,00%</b> | <b>100,00%</b> | 85,00%        | 94,74%         | <b>94,95%</b> |
| EII - algorytm 1         | 93,33%         | 85,00%        | 88,46%         | 60,00%        | <b>100,00%</b> | 85,36%        |
| EII - algorytm 4         | 93,33%         | 90,00%        | 92,31%         | 65,00%        | 84,21%         | 84,97%        |
| EII - algorytm 6         | 86,67%         | 85,00%        | <b>100,00%</b> | 55,00%        | <b>100,00%</b> | 85,33%        |
| EII - algorytm 5         | <b>100,00%</b> | 80,00%        | 92,31%         | 65,00%        | <b>100,00%</b> | 87,46%        |
| EII - algorytm 2         | <b>100,00%</b> | 90,00%        | 96,15%         | 70,00%        | 94,74%         | 90,18%        |
| EII - algorytm 3         | <b>100,00%</b> | <b>95,00%</b> | 96,15%         | 80,00%        | <b>100,00%</b> | 94,23%        |
| EI - Linear SVC          | 93,33%         | 70,00%        | 96,15%         | 15,00%        | <b>100,00%</b> | 74,90%        |
| EI - Nearest Centroid    | 93,33%         | 80,00%        | 80,77%         | 55,00%        | <b>100,00%</b> | 81,82%        |
| EI - C-SVC               | 86,67%         | 75,00%        | 92,31%         | 70,00%        | 89,47%         | 82,69%        |
| EI - Radius Neighbors    | 86,67%         | 90,00%        | 92,31%         | 45,00%        | <b>100,00%</b> | 82,80%        |
| EI - kNN                 | <b>100,00%</b> | 75,00%        | 96,15%         | 60,00%        | 89,47%         | 84,12%        |
| EI - Decision Tree       | 93,33%         | 75,00%        | 92,31%         | 75,00%        | 84,21%         | 83,97%        |
| EI - Extra Trees         | 86,67%         | 90,00%        | 84,62%         | 70,00%        | 94,74%         | 85,21%        |
| EI - Nu-SVC              | 80,00%         | <b>95,00%</b> | 88,46%         | <b>90,00%</b> | 78,95%         | 86,48%        |
| EI - Random Forest       | 93,33%         | 80,00%        | 96,15%         | 75,00%        | 89,47%         | 86,79%        |
| EI - Stacking Classifier | <b>100,00%</b> | 90,00%        | 92,31%         | 70,00%        | 94,74%         | 89,41%        |

RYSUNEK 4.73: Zestawienie SEN dla pojedynczych klas i różnych metod, EI - etap I, EII - etap II, EIII - etap III

## Podsumowanie

Tematyka zaprezentowana w niniejszej pracy jest istotna ze względu na ciągły wzrost wykorzystania sieci komputerowych, a co za tym idzie konieczność zapewnienia usługi transmisji danych na określonym poziomie. Istotnym jest także racjonalne gospodarowanie zasobami sieci komputerowej. Mimo definiowania różnych pojęć związanych z jakością transmisji nadal w wielu przypadkach użytkownicy płacą za maksymalną możliwą do uzyskania przepływność (bandwidth), która z dużym prawdopodobieństwem nigdy nie zostanie osiągnięta. Takie podejście jest negatywne z punktu widzenia gospodarowania zasobami sieci i relacji dostawca-klient.

W niniejszej pracy zaprezentowano rozwiązanie, w którym głównym założeniem jest to, iż usługa transmisji danych świadczona jest z określonymi parametrami i na zagwarantowanym poziomie. Proponowane rozwiązanie stanowi kombinację routingu statycznego i dynamicznego w połączeniu z systemem wieloagentowym. Wykorzystanie agentów do monitorowania parametrów sieci i w celu decentralizacji zarządzania siecią okazało się być dobrym podejściem. Nie jest to jedyna płaszczyzna, na której zastosowano podejście agentowe. Agenty zostały wykorzystane również w procesie dynamicznego kształtowania cen jakości transmisji danych. Stworzono kompletny system, a sama koncepcja została sprawdzona z wykorzystaniem emulatora i symulatora. Zarówno emulacja, jak i symulacja przeprowadzone zostały dla różnych wielkości sieci komputerowych. W badaniach zweryfikowano poprawność wyboru ścieżek transmisji w zależności od oczekiwań klienta w kontekście jakości transmisji danych. Wykonano również pomiary wartości opóźnienia transmisji oraz czasu transmisji w celu zweryfikowania czy różnicowanie ścieżek będzie miało wpływ na uzyskane wyniki. Kolejne badania dotyczyły możliwości wykorzystania aukcji w procesie dynamicznego kształtowania cen. Ostatni z eksperymentów dotyczył wykorzystania uczenia maszynowego w procesie przekształcania parametrów opisujących jakość transmisji w pewnej skali. Uzyskane wyniki są zadowalające.

Następująca teza postawiona została w niniejszej pracy:

*Możliwe jest różnicowanie jakości usługi, jaką jest transmisja danych, poprzez zastosowanie systemu wieloagentowego do sterowania siecią tak, aby usługa realizowana była na poziomie oczekiwanym przez użytkownika.*

W niniejszej pracy zaproponowano koncepcję Pay&Require zakładającą wykorzystanie systemu wieloagentowego do monitorowania i rekonfigurowania sieci. Wykorzystano kilka różnych agentów, do których zadań należało:

- monitorowanie parametrów łączy w sieci,
- podejmowanie decyzji odnośnie rekonfiguracji sieci w sytuacji, w której jakość transmisji nie była zgodna z oczekiwaniami klienta,
- determinacja ścieżek transmisji spełniających oczekiwania klienta w kontekście jakości transmisji,
- definiowanie tablic rutingu,
- sterowanie urządzeniami sieciowymi,
- obsługa procesu zakupu jakości usługi transmisji danych.

Dzięki wykorzystaniu systemu wieloagentowego uzyskano zdecentralizowane rozwiązanie - agenty nadzorują poszczególne urządzenia i komunikują się między sobą wymieniając niezbędne informacje. Decentralizacja jest dobrym rozwiązaniem ze względów bezpieczeństwa. W badaniach opisanych w podrozdziałach 4.1 i 4.2 wykorzystano podejście agentowe. Zarówno w przypadku emulacji, jak i symulacji nie było to łatwe zadanie. Z przeprowadzonych badań wynika, że system wieloagentowy dobrze poradził sobie ze sformułowanymi zadaniami. Rozwiązanie zostało zdecentralizowane i stwierdzono, że agenty mogą zostać wykorzystane do sterowania siecią komputerową. Przeprowadzenie emulacji i symulacji miało na celu zweryfikowanie czy różnicowanie ścieżek transmisji danych przekłada się na zapewnienie jakości usługi na oczekiwanym przez klienta poziomie. Stworzenie emulatora było bardzo skomplikowanym procesem, ponieważ wymagało przygotowania kompletnego rozwiązania implementującego koncepcję Pay&Require.

Kolejnym aspektem poruszonym w niniejszej pracy było wykorzystanie agentów w procesie zakupu jakości usługi transmisji danych. Nowatorskim jest podejście, w którym zakup dokonywany jest przy wykorzystaniu aukcji. W podrozdziale 3.4 przedstawiono koncepcję wykorzystania agentów w procesie zakupu jakości transmisji danych. Zaproponowano kilka strategii działania. Z kolei w podrozdziale 4.3 przedstawiono wyniki badań przeprowadzonych dla:

- Jednokrotnych aukcji jednego dobra,



- Wielokrotnych aukcji jednego dobra,
- Aukcji drugiej ceny,
- Aukcji kombinatorycznych.

Uzyskane wyniki pozwoliły stwierdzić, że możliwym jest zastosowanie aukcji w procesie zakupu jakości transmisji. Stwierdzono, że musi to być proces mający miejsce niezależnie od normalnego funkcjonowania sieci. Wynika to z faktu, iż aukcje wymagają poświęcenia czasu na przeprowadzenie niezbędnych czynności. Lecz niezależnie od tego aukcje mogą zostać wykorzystane do dynamicznego kształtowania cen - wielu klientów może brać udział w licytacji, co może pozytywnie wpłynąć na cenę usługi zwłaszcza z punktu widzenia maksymalizacji zysku przez operatora. Istotny jest przyjęty protokół prowadzenia aukcji oraz strategie stosowane przez poszczególnych uczestników.

Nowatorskim podejściem, zaprezentowanym w niniejszej pracy, było wykorzystanie uczenia maszynowego do przekształcania parametrów opisujących jakość transmisji w pewnej skali. Parametry opisujące jakość transmisji danych to przepływność, opóźnienie transmisji, fluktuacja opóźnień oraz współczynnik utraty pakietów. Parametry te mogą być trudne do zrozumienia dla klienta chcącego dokonać transmisji danych z określoną jakością. Również w przypadku agentów, których zadaniem jest zweryfikowanie czy ścieżka transmisji spełnia oczekiwania jakościowe klienta, trudną kwestią okazało się określenie jak kombinacja poszczególnych parametrów wpłynie na całościową jakość transmisji. W związku z tym, zdecydowano się na przekształcenie parametrów w pewnej skali - zgodnej z tą wykorzystaną w Pay&Require, czyli 1-5. Każda ocena stanowiła odrębną klasę w procesie klasyfikacji. W pierwszej kolejności próbowano wykorzystać statyczne tablice translacji, jednakże takie rozwiązanie okazało się trudne w zdefiniowaniu. Innym podejściem było wykorzystanie uczenia maszynowego. Aby przeprowadzić proces uczenia przygotowano dane testowe, które ocenione zostały przez użytkowników. Tak pozyskane próbki wykorzystane zostały w procesie uczenia maszynowego. Przeprowadzono trzy serie eksperymentów:

- z wykorzystaniem klasycznych klasyfikatorów,
- z wykorzystaniem uczenia zespołowego,
- z zastosowaniem fuzji sieci neuronowej i algorytmu genetycznego.

Przeprowadzone eksperymenty opisane zostały w podrozdziale 3.5, a wyniki badań zaprezentowano w podrozdziale 4.4. W pierwszej serii eksperymentów wykorzystano 9 typów klasyfikatorów, 5 metod przetwarzania wstępnego, 2

rodzaje walidacji krzyżowej oraz dwie wielkości populacji i generacji. W ten sposób uzyskano 180 różnych kombinacji. Zaprezentowano trzy najlepsze wyniki uzyskane dla każdego klasyfikatora. W tej serii przeprowadzono także eksperymenty, w których wykorzystano *Stacking Classifier* w dwóch wariantach kombinacji estymatorów i z wykorzystaniem 5 metod przetwarzania wstępnego oraz 10-krotnej walidacji krzyżowej. Do porównania jakości procesu klasyfikacji wykorzystano macierze pomyłek (ewaluacja liczby błędów) oraz ogólny współczynnik dokładności (SEN). Uzyskane wyniki zostały w obszerny sposób zaprezentowane w pracy. Najlepszy wynik uzyskany na tym etapie to  $SEN=89,00\%$  dla klasyfikatora *Stacking Classifier* w wariacie 1. Wynik ten nie był satysfakcjonujący i w związku z tym zdecydowano się na rozszerzenie badań.

Druga seria badań przeprowadzona została pod kątem wykorzystania uczenia zespołowego. Zaproponowano sześć różnych algorytmów, każdy stanowił inne podejście do procesu klasyfikacji - różna liczba warstw klasyfikacji, różne estymatory. Również w tym wypadku wykorzystano macierze pomyłek do ewaluacji liczby błędów, ogólny współczynnik dokładności klasyfikacji (SEN) oraz *F1 Score*, jako funkcję akceptacji w przypadku części zaproponowanych algorytmów. Zaprezentowano najlepszy wynik uzyskany przez każdy algorytm. Poza tym zaprezentowano macierze pomyłek dla każdego algorytmu co pozwoliło ocenić, jak algorytmy poradziły sobie w poszczególnych klasach. W tej serii badań najlepszy wynik uzyskał algorytm 3 w dwóch wariantach. Wynik ten to  $SEN=94,00\%$ .

Ostatni etap badań dotyczył wykorzystania fuzji sieci neuronowej i algorytmu genetycznego. Zaproponowano algorytm łączący w sobie oba rozwiązania. Uzyskany wynik to  $SEN=95,00\%$ .

W ramach całej pracy przeprowadzono liczne eksperymenty, a uzyskane wyniki pozwoliły stwierdzić, że teza sformułowana została prawidłowo. Zastosowanie systemu wieloagentowego do sterowania siecią w połączeniu z uczeniem maszynowym wykorzystanym w przypadku przekształcania parametrów opisujących jakość transmisji oraz przy wykorzystaniu agentów do dynamicznego kształtowania cen, pozwoliły na uzyskanie rozwiązania, które może zostać wykorzystane do gwarantowania jakości usługi transmisji danych. To nowatorskie rozwiązanie, istotne z punktu widzenia funkcjonowania sieci komputerowych. Kombinacja zastosowanych rozwiązań stanowi element nowatorski we współczesnych sieciach komputerowych.

## Oryginalne elementy pracy

Na podstawie przeprowadzonego przeglądu literatury (rozdział 1) stwierdzić można, iż do oryginalnych elementów pracy należą:

- **Zapewnianie jakości transmisji danych realizowane w oparciu o różnicowanie ścieżek transmisji.**

W klasycznych sieciach komputerowych w których wykorzystano standard Ethernet i protokół IP sposób działania protokołów routingu powoduje, że ścieżki transmisji uzależnione są wyłącznie od celu transmisji. W zaproponowanym rozwiązaniu wykorzystano podejście takie, w którym ścieżka transmisji zależna jest zarówno od celu, jak i źródła. Oznacza to, że możliwym jest różnicowanie ścieżek w zależności od nadawcy, a co za tym idzie, jeśli w sieci poszczególne ścieżki posiadają różne parametry jakościowe to dane klientów mogą być przesyłane ścieżkami spełniającymi ich wymagania odnośnie jakości transmisji.

- **Wykorzystanie systemu wieloagentowego do zarządzania siecią.**

Różnicowanie ścieżek transmisji w sposób statyczny nie ma uzasadnienia. W związku z tym, zdecydowano się na wykorzystanie rozwiązania dynamicznego. Aby możliwe było dynamiczne sterowanie siecią wykorzystano system wieloagentowy. Agenty odpowiedzialne są za monitorowanie parametrów opisujących jakość transmisji oraz, jeśli zajdzie taka potrzeba, za przeprowadzenie procesu rekonfiguracji sieci. Rekonfiguracja ma miejsce w momencie, gdy jakość transmisji odbiega od oczekiwań klienta. System agentowy weryfikuje parametry poszczególnych ścieżek i przeprowadza rekonfigurację tablic routingu w sposób automatyczny.

- **Zastosowanie aukcji do kształtowania cen usługi transmisji danych.**

Zazwyczaj użytkownicy ponoszą koszt świadczonej usługi jako opłatę za pewien okres użytkowania sieci. W niniejszej pracy zaproponowano wykorzystanie aukcji do dynamicznego kształtowania cen. Wpływ na cenę usługi ma zapotrzebowanie tzn. liczba klientów, którzy są zainteresowani zakupem konkretnej jakości transmisji danych. Zakup może zostać dokonany na różny okres, a cena uzależniona jest od wyniku aukcji. Do obsługi procesu zakupu wykorzystano system wieloagentowy w którym agenty stanowią reprezentantów klientów i operatora.

- **Wykorzystanie uczenia maszynowego do przekształcania parametrów opisujących jakość transmisji w pewnej skali.**

W niniejszej pracy istotnym okazało się wykorzystanie mechanizmu przekształcania parametrów determinujących jakość transmisji, tj. przepływności, opóźnienia transmisji, fluktuacji opóźnienia oraz współczynnika utraty pakietów w pewnej ustalonej skali. Proces ten jest niezbędny, aby dostarczyć klientowi zrozumiałą informację odnośnie tego co kupuje. Takie przekształcenie ułatwia też weryfikację jakości ścieżek przez system wieloagentowy. Przekształcenie okazało się skomplikowanym procesem i w związku z tym zdecydowano się na wykorzystanie uczenia maszynowego. Oryginalne są:

- stworzenie zbioru danych wykorzystanych w procesie uczenia klasyfikatorów,
- przetestowanie kilku różnych klasyfikatorów na potrzeby klasyfikacji jakości transmisji,
- zaproponowanie nowych algorytmów uczenia zespołowego,
- zaproponowanie algorytmu stanowiącego fuzję sieci neuronowej i algorytmu genetycznego.

## Kierunki dalszych badań

Uzyskane wyniki są zadowalające i potwierdzają możliwość wykorzystania zaprezentowanej koncepcji do zapewniania jakości transmisji danych w sieciach komputerowych. Zauważono kilka możliwych kierunków dalszych badań:

- Weryfikacja skalowalności zaproponowanego rozwiązania - sieci o większej liczbie urządzeń i użytkowników,
- Zastosowanie uczenia maszynowego do przewidywania przebiegu aukcji,
- Nowe algorytmy wykorzystywane na potrzeby przekształcania parametrów jakościowych w pewnej skali,
- Pozyskanie większej liczby próbek, które wykorzystane zostaną do uczenia klasyfikatorów, a co za tym idzie próba zwiększenia precyzji klasyfikacji,
- Wykorzystanie zaprezentowanego rozwiązania w rzeczywistej sieci komputerowej.

## Publikacje autora

Publikacje naukowe związane z tematyką rozprawy doktorskiej:

1. **D. Żelasko**, W. Książek, P. Pławiak; "[Transmission quality classification with use of fusion of neural network and genetic algorithm in Pay&Require multi-agent managed network](#)". *Sensors*, Vol. 21, Iss. 12, Spec. Iss. (2021): [1-14]. IF = 3.576, MEiN = 100 pkt.
2. **D. Żelasko**, P. Pławiak; "[Ensemble learning techniques for transmission quality classification in a Pay&Require multi-layer network](#)". *International Journal of Applied Mathematics and Computer Science*, Vol. 31, No. 1 (2021): 135-153. IF = 1.417, MEiN = 100 pkt.
3. **D. Żelasko**, P. Pławiak, J. Kołodziej; "[Machine learning techniques for transmission parameters classification in multi-agent managed network](#)". *CCGRID 2020 : 20th IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing*, 11-14 May 2020 Melbourne, Australia (pp. 699-707). IEEE, 2020. MEiN = 140 pkt.
4. **D. Żelasko**; "[Simulation of transmission quality classification in Pay&Require multi-agent managed network by means of Machine Learning techniques](#)". *Simulation Modelling Practice and Theory*, Vol. 103 (2020): [1-24]. IF = 3.272, MEiN = 100 pkt.
5. **D. Żelasko**; "[Ensuring the QoS in computer networks through the use of the Pay&Require multi-agent system and electronic auctions](#)". *Technical Transactions = Czasopismo Techniczne*, Vol. 115, Iss. 10 (2018): 71-83. MNiSW = 13 pkt.
6. **D. Żelasko**, J. Koźlak, K. Wajda; "Koncepcja zapewniania jakości Pay&Require – badania wydajności z zastosowaniem symulatora NS3". *Przegląd Telekomunikacyjny. Wiadomości Telekomunikacyjne*, no.nr 8-9 (2017): 769-772. MNiSW = 9 pkt.

7. **D. Żelasko**, K. Cetnarowicz, K. Wajda, J. Koźlak; "Pay&Require as concept of variable cost routing in dynamically reconfigured networks". Technical Transactions. Fundamental Sciences = Czasopismo Techniczne. Nauki Podstawowe, Iss. 1-NP (2016): 201-214. MNiSW = 13 pkt.
8. **D. Żelasko**, K. Cetnarowicz, J. Koźlak, K. Wajda; "Mechanizm zdecentralizowanego routingu Pay&Require". Przegląd Telekomunikacyjny. Wiadomości Telekomunikacyjne, nr 8-9 (2016): 879-886. MNiSW = 9 pkt.
9. **D. Żelasko**, K. Cetnarowicz, K. Wajda; "Koncepcja trasowania ze zmiennym kosztem dla sieci o sterowaniu rozproszonym". Przegląd Telekomunikacyjny. Wiadomości Telekomunikacyjne, nr 8-9 (2015): 724-729. MNiSW = 9 pkt.
10. **D. Żelasko**; "Policy-Based Routing na przykładzie systemu Vyatta". Przegląd Elektrotechniczny, Nr 8 (2014): 46-49. MNiSW = 10 pkt.
11. **D. Żelasko**; "Współczesne technologie zapewniania jakości w sieciach komputerowych". Przegląd Elektrotechniczny, R. 89, Nr 8 (2013): 183-186. MNiSW = 10 pkt.

Publikacje naukowe niezwiązane z tematyką rozprawy doktorskiej:

1. R.K. Nowicki, R. Seliga, **D. Żelasko**, Y. Hayashi; "[Performance analysis of rough set-based hybrid classification systems in the case of missing values](#)". Journal of Artificial Intelligence and Soft Computing Research, Vol. 11, No. 4: 307-318. IF = 2.500, MEiN = 140 pkt.

# Spis rysunków

|      |   |    |
|------|---|----|
| 1.1  | SDN - separacja warstw . . . . .  | 15 |
| 3.1  | Model P&R . . . . .   | 40 |
| 3.2  | Diagram sekwencji dla agenta Ag . . . . .   | 42 |
| 3.3  | Przykładowa sieć - charakterystyka algorytmu wyznaczania tras . . . . .   | 46 |
| 3.4  | Przykładowa sieć - porównanie klasycznego routingu z P&R . . . . .  | 48 |
| 4.1  | Schemat sieci wykorzystanej w badaniach (4 rutery) . . . . .  | 57 |
| 4.2  | Schemat sieci wykorzystanej w badaniach (9 ruterów) . . . . .   | 58 |
| 4.3  | Wybrana ścieżka dla sieci działającej z maksymalnymi parametrami . . . . .  | 59 |
| 4.4  | Wybrana ścieżka dla klienta akceptującego niską jakość dla sieci, w której pogorszeniu uległy parametry łącza p3 . . . . .        | 59 |
| 4.5  | Ścieżka wybrana dla klienta wymagającego wysokiej jakości dla sieci, w której pogorszeniu uległy parametry łącza p2 . . . . .     | 60 |
| 4.6  | Ścieżka wybrana dla klienta wymagającego niskiej jakości dla sieci, w której pogorszeniu uległy parametry łącza p2 . . . . .      | 60 |
| 4.7  | Wykres ping i czasów transmisji dla routingu klasycznego i P&R . . . . .  | 61 |
| 4.8  | Wybrana ścieżka pomiędzy klientami podłączonymi do R1 i R3 dla sieci działającej z maksymalnymi parametrami . . . . .             | 63 |
| 4.9  | Wybrana ścieżka pomiędzy klientami podłączonymi do R1 i R6 dla sieci działającej z maksymalnymi parametrami . . . . .             | 64 |
| 4.10 | Wybrana ścieżka pomiędzy klientami podłączonymi do R1 i R9 dla sieci działającej z maksymalnymi parametrami . . . . .             | 64 |
| 4.11 | Wybrana ścieżka pomiędzy klientami podłączonymi do R1 i R8 dla sieci działającej z maksymalnymi parametrami . . . . .             | 65 |
| 4.12 | Wybrana ścieżka pomiędzy klientami podłączonymi do R1 i R3 dla sieci, w której pogorszeniu uległy parametry łącza R1-R2 . . . . . | 65 |
| 4.13 | Wybrana ścieżka pomiędzy klientami podłączonymi do R1 i R6 dla sieci, w której pogorszeniu uległy parametry łącza R1-R2 . . . . . | 66 |
| 4.14 | Wybrana ścieżka pomiędzy klientami podłączonymi do R1 i R8 dla sieci, w której pogorszeniu uległy parametry łącza R1-R2 . . . . . | 67 |

|      |   |    |
|------|---|----|
| 4.15 | Wybrana ścieżka pomiędzy klientami podłączonymi do R1 i R9 dla sieci, w której pogorszeniu uległy parametry łącza R1-R2 . . . . .   | 68 |
| 4.16 | Wybrana ścieżka pomiędzy klientami podłączonymi do R1 i R3 dla sieci, w której pogorszeniu uległy parametry łącza R5-R6 . . . . .   | 68 |
| 4.17 | Wybrana ścieżka pomiędzy klientami podłączonymi do R1 i R6 dla sieci, w której pogorszeniu uległy parametry łącza R5-R6 . . . . .   | 69 |
| 4.18 | Wybrana ścieżka pomiędzy klientami podłączonymi do R1 i R9 dla sieci, w której pogorszeniu uległy parametry łącza R5-R6 . . . . .   | 70 |
| 4.19 | Wybrana ścieżka pomiędzy klientami podłączonymi do R1 i R8 dla sieci, w której pogorszeniu uległy parametry łącza R5-R6 . . . . .   | 70 |
| 4.20 | Wybrana ścieżka pomiędzy klientami podłączonymi do R1 i R3 dla sieci, w której pogorszeniu uległy parametry łącza R5-R9 . . . . .   | 71 |
| 4.21 | Wybrana ścieżka pomiędzy klientami podłączonymi do R1 i R6 dla sieci, w której pogorszeniu uległy parametry łącza R5-R9 . . . . .   | 72 |
| 4.22 | Wybrana ścieżka pomiędzy klientami podłączonymi do R1 i R9 dla sieci, w której pogorszeniu uległy parametry łącza R5-R9 . . . . .   | 73 |
| 4.23 | Wybrana ścieżka pomiędzy klientami podłączonymi do R1 i R8 dla sieci, w której pogorszeniu uległy parametry łącza R5-R9 . . . . .   | 73 |
| 4.24 | Wyniki uzyskane dla symulacji (S) i emulacji (E) – średnie opóźnienie transmisji . . . . .  | 76 |
| 4.25 | Wyniki uzyskane dla symulacji (S) i emulacji (E) – średni czas transmisji   | 76 |
| 4.26 | Schemat sieci wykorzystanej w symulacji Źródło: SNDLib . . . . .  | 77 |
| 4.27 | Wybrana ścieżka pomiędzy klientami dla sieci działającej z maksymalnymi parametrami . . . . .   | 78 |
| 4.28 | Wybrana ścieżka pomiędzy klientami dla sieci działającej z obniżonymi parametrami łącza Gdańsk-Białystok . . . . .  | 79 |
| 4.29 | Wyniki uzyskane dla sieci Poland – średnie opóźnienie transmisji . .  | 80 |
| 4.30 | Wyniki uzyskane dla sieci Poland – średni czas transmisji . . . . .   | 80 |
| 4.31 | Wynik dla strategii 1, aukcja <i>first-price</i> . . . . .  | 82 |
| 4.32 | Wynik dla strategii 1, aukcja <i>first-price</i> . . . . .  | 83 |
| 4.33 | Wyniki dla strategii 3, <i>first-price</i> - częściowa informacja . . . . .   | 84 |
| 4.34 | Wyniki dla strategii 3, <i>first-price</i> - pełna informacja . . . . .   | 85 |
| 4.35 | Wyniki dla strategii 3, <i>second-price</i> - pełna informacja . . . . .  | 86 |
| 4.36 | Proces ML - Etap I . . . . .  | 89 |
| 4.37 | Proces ML - <i>Stacking Classifier</i> . . . . .  | 90 |
| 4.38 | Macierz pomyłek dla najlepszego wyniku klasyfikacji klasyfikatora <i>Nu-SVC</i> (SEN=87,00%). W pomarańczowych komórkach SEN zostało przedstawione dla pojedynczych klas. . . . . | 95 |



|      |  |     |
|------|--|-----|
| 4.39 | Macierz pomyłek dla najlepszego wyniku klasyfikacji klasyfikatora <i>Random Forest</i> (SEN=87,00%). W pomarańczowych komórkach SEN zostało przedstawione dla pojedynczych klas. . . . .       | 95  |
| 4.40 | Macierz pomyłek dla najlepszego wyniku klasyfikacji klasyfikatora <i>C-SVC</i> (SEN=83,00%). W pomarańczowych komórkach SEN zostało przedstawione dla pojedynczych klas. . . . .               | 96  |
| 4.41 | Macierz pomyłek dla najlepszego wyniku klasyfikacji algorytmu <i>kNN</i> (SEN=84,00%). W pomarańczowych komórkach SEN zostało przedstawione dla pojedynczych klas. . . . .                     | 96  |
| 4.42 | Macierz pomyłek dla najlepszego wyniku klasyfikacji klasyfikatora <i>Decision Tree</i> (SEN=84,00%). W pomarańczowych komórkach SEN zostało przedstawione dla pojedynczych klas. . . . .       | 96  |
| 4.43 | Macierz pomyłek dla najlepszego wyniku klasyfikacji klasyfikatora <i>Nearest Centroid</i> (SEN=81,00%). W pomarańczowych komórkach SEN zostało przedstawione dla pojedynczych klas. . . . .    | 96  |
| 4.44 | Macierz pomyłek dla najlepszego wyniku klasyfikacji klasyfikatora <i>Radius Neighbors</i> (SEN=83,00%). W pomarańczowych komórkach SEN zostało przedstawione dla pojedynczych klas. . . . .    | 96  |
| 4.45 | Macierz pomyłek dla najlepszego wyniku klasyfikacji klasyfikatora <i>Linear SVC</i> (SEN=75,00%). W pomarańczowych komórkach SEN zostało przedstawione dla pojedynczych klas. . . . .          | 97  |
| 4.46 | Macierz pomyłek dla najlepszego wyniku klasyfikacji klasyfikatora <i>Extra Trees</i> (SEN=85,00%). W pomarańczowych komórkach SEN zostało przedstawione dla pojedynczych klas. . . . .         | 97  |
| 4.47 | Macierz pomyłek dla najlepszego wyniku klasyfikacji <i>Stacking Classifier</i> w wariancie 1 (SEN=89,00%). W pomarańczowych komórkach SEN zostało przedstawione dla pojedynczych klas. . . . . | 97  |
| 4.48 | Macierz pomyłek dla najlepszego wyniku klasyfikacji <i>Stacking Classifier</i> w wariancie 2 (SEN=87,00%). W pomarańczowych komórkach SEN zostało przedstawione dla pojedynczych klas. . . . . | 97  |
| 4.49 | SEN dla poszczególnych klasyfikatorów (etap I) i klas . . . . .  | 97  |
| 4.50 | Proces ML, algorytm 1 . . . . .  | 99  |
| 4.51 | Proces ML, algorytm 2 . . . . .  | 99  |
| 4.52 | Proces ML, algorytm 3 wariant 1 . . . . .  | 100 |
| 4.53 | Proces ML, algorytm 3 wariant 2 . . . . .  | 101 |
| 4.54 | Proces ML, algorytm 3 wariant 3 . . . . .  | 102 |
| 4.55 | Proces ML, algorytm 4 . . . . .  | 103 |
| 4.56 | Proces ML, algorytm 5 . . . . .  | 104 |

|      |  |     |
|------|--|-----|
| 4.57 | Proces ML, algorytm 6 . . . . .  | 106 |
| 4.58 | Macierz pomyłek dla najlepszego wyniku klasyfikacji z wykorzystaniem algorytmu 1 (SEN=85,00%). W ciemnoniebieskich komórkach SEN zostało przedstawione dla pojedynczych klas . . . . .           | 107 |
| 4.59 | Macierz pomyłek dla najlepszego wyniku klasyfikacji z wykorzystaniem algorytmu 2 (SEN=90,00%). W ciemnoniebieskich komórkach SEN zostało przedstawione dla pojedynczych klas . . . . .           | 107 |
| 4.60 | Macierz pomyłek dla najlepszego wyniku klasyfikacji z wykorzystaniem algorytmu 3 wariant 1 (SEN=91,00%). W ciemnoniebieskich komórkach SEN zostało przedstawione dla pojedynczych klas . . . . . | 108 |
| 4.61 | Macierz pomyłek dla najlepszego wyniku klasyfikacji z wykorzystaniem algorytmu 3 wariant 2 (SEN=94,00%). W ciemnoniebieskich komórkach SEN zostało przedstawione dla pojedynczych klas . . . . . | 108 |
| 4.62 | Macierz pomyłek dla najlepszego wyniku klasyfikacji z wykorzystaniem algorytmu 3 wariant 3 (SEN=94,00%). W ciemnoniebieskich komórkach SEN zostało przedstawione dla pojedynczych klas . . . . . | 109 |
| 4.63 | Macierz pomyłek dla najlepszego wyniku klasyfikacji z wykorzystaniem algorytmu 4 (SEN=85,00%). W ciemnoniebieskich komórkach SEN zostało przedstawione dla pojedynczych klas . . . . .           | 110 |
| 4.64 | Macierz pomyłek dla najlepszego wyniku klasyfikacji z wykorzystaniem algorytmu 5 (SEN=87,00%). W ciemnoniebieskich komórkach SEN zostało przedstawione dla pojedynczych klas . . . . .           | 110 |
| 4.65 | Macierz pomyłek dla najlepszego wyniku klasyfikacji z wykorzystaniem algorytmu 6 (SEN=86,00%). W ciemnoniebieskich komórkach SEN zostało przedstawione dla pojedynczych klas . . . . .           | 111 |
| 4.66 | SEN dla poszczególnych algorytmów (etap II) i klas . . . . .   | 111 |
| 4.67 | SEN dla różnych metod (etap I i etap II) . . . . .   | 112 |
| 4.68 | Proces ML . . . . .  | 113 |
| 4.69 | Model jednokierunkowej sieci neuronowej . . . . .  | 114 |
| 4.70 | Macierz pomyłek dla najlepszego wyniku klasyfikacji (SEN=95,00%). Jasnoszara komórka – SEN obliczone dla pojedynczej klasy . . . . .   | 116 |
| 4.71 | Zmienność błędu średniokwadratowego dla różnych iteracji (epok) GA116  |     |
| 4.72 | SEN dla różnych metod, EI - etap I, EII - etap II, EIII - etap III . . . . .   | 117 |
| 4.73 | Zestawienie SEN dla pojedynczych klas i różnych metod, EI - etap I, EII - etap II, EIII - etap III . . . . .   | 118 |

# Spis tabel

|      |  |     |
|------|--|-----|
| 1.1  | Skala MOS . . . . .  | 18  |
| 1.2  | Skala porównawcza . . . . .  | 18  |
| 3.1  | Przykładowe cechy łącza dla poszczególnych poziomów usługi . . . . .   | 45  |
| 3.2  | Szczegóły zbioru danych . . . . .  | 55  |
| 4.1  | Czasy transmisji dla routingu klasycznego i P&R wraz z estymacją dla poziomu ufności $\beta=95\%$ . . . . .    | 61  |
| 4.2  | Wyniki symulacji wraz z wynikami emulacji i estymacją dla przyjętego poziomu ufności $\beta=95\%$ . . . . .    | 75  |
| 4.3  | Wyniki symulacji dla sieci Poland wraz z estymacją dla poziomu ufności $\beta=95\%$ . . . . .                  | 79  |
| 4.4  | Oponenty w przypadku strategii 1 . . . . .   | 81  |
| 4.5  | Oponenty w przypadku strategii 2 . . . . .   | 82  |
| 4.6  | Parametry klasyfikatorów etapu I . . . . .   | 90  |
| 4.7  | Parametry <i>Stacking Classifier</i> . . . . .   | 90  |
| 4.8  | Parametry algorytmu genetycznego (GA) - etap I . . . . .   | 90  |
| 4.9  | Trzy najlepsze wyniki klasyfikatora <i>Nu-SVC</i> . . . . .  | 92  |
| 4.10 | Trzy najlepsze wyniki klasyfikatora <i>Random Forest</i> . . . . .   | 92  |
| 4.11 | Trzy najlepsze wyniki klasyfikatora <i>C-SVC</i> . . . . .   | 92  |
| 4.12 | Trzy najlepsze wyniki algorytmu <i>kNN</i> . . . . .   | 93  |
| 4.13 | Trzy najlepsze wyniki klasyfikatora <i>Decision Tree</i> . . . . .   | 93  |
| 4.14 | Trzy najlepsze wyniki klasyfikatora <i>Nearest Centroid</i> . . . . .  | 93  |
| 4.15 | Trzy najlepsze wyniki klasyfikatora <i>Radius Neighbors</i> . . . . .  | 93  |
| 4.16 | Trzy najlepsze wyniki klasyfikatora <i>Linear SVC</i> . . . . .  | 93  |
| 4.17 | Trzy najlepsze wyniki klasyfikatora <i>Extra Trees</i> . . . . .   | 93  |
| 4.18 | Trzy najlepsze wyniki klasyfikatora <i>Stacking Classifier</i> w wariancie 1 . . . . .                         | 93  |
| 4.19 | Trzy najlepsze wyniki klasyfikatora <i>Stacking Classifier</i> w wariancie 2 . . . . .                         | 93  |
| 4.20 | Parametry algorytmu genetycznego (GA) - etap II . . . . .  | 98  |
| 4.21 | Parametry wykorzystane w przypadku, w którym uzyskano najlepszą dokładność klasyfikacji - algorytm 1 . . . . . | 106 |

---

|      |  |     |
|------|--|-----|
| 4.22 | Parametry wykorzystane w przypadku, w którym uzyskano najlepszą dokładność klasyfikacji - algorytm 2 . . . . .           | 107 |
| 4.23 | Parametry wykorzystane w przypadku, w którym uzyskano najlepszą dokładność klasyfikacji - algorytm 3 wariant 1 . . . . . | 108 |
| 4.24 | Parametry wykorzystane w przypadku, w którym uzyskano najlepszą dokładność klasyfikacji - algorytm 3 wariant 2 . . . . . | 108 |
| 4.25 | Parametry wykorzystane w przypadku, w którym uzyskano najlepszą dokładność klasyfikacji - algorytm 3 wariant 3 . . . . . | 109 |
| 4.26 | Parametry wykorzystane w przypadku, w którym uzyskano najlepszą dokładność klasyfikacji - algorytm 4 . . . . .           | 109 |
| 4.27 | Parametry wykorzystane w przypadku, w którym uzyskano najlepszą dokładność klasyfikacji - algorytm 5 . . . . .           | 110 |
| 4.28 | Parametry wykorzystane w przypadku, w którym uzyskano najlepszą dokładność klasyfikacji - algorytm 6 . . . . .           | 111 |

# Bibliografia

- [1] J. Ray. "Asynchronous Transfer Mode (ATM) Technology and Applications". In: *SSRN Electronic Journal* (June 1998).
- [2] *Multiprotocol Label Switching Architecture (RFC3031)*. <https://www.ietf.org/rfc/rfc3031.txt>.
- [3] *Software-Defined Networking: A Perspective from within a Service Provider Environment (RFC7149)*. <https://tools.ietf.org/html/rfc7149>.
- [4] *Software-Defined Networking (SDN): Layers and Architecture Terminology (RFC7426)*. <https://tools.ietf.org/html/rfc7426>.
- [5] *E.800 : Definitions of terms related to quality of service*. <https://www.itu.int/rec/T-REC-E.800-200809-I/en>.
- [6] J. Gozdecki, A. Jajszczyk, and R. Stankiewicz. "Quality of service terminology in IP networks". In: *IEEE Communications Magazine* 41.3 (2003), pp. 153–159.
- [7] R. Stankiewicz, P. Chołda, and A. Jajszczyk. "QoX: What is it really". In: *IEEE Communications Magazine* 49.4 (2011).
- [8] A. Iwata and N. Fujita. "A hierarchical multilayer QoS routing system with dynamic SLA management". In: *IEEE Journal on Selected Areas in Communications* 18.12 (2001), pp. 2603–2616.
- [9] J. A. Perez, V. H. Zarate, and C. Cebreira. "A Network and Data Link Layer QoS Model to Improve Traffic Performance". In: *International Conference on Embedded and Ubiquitous Computing* (2006).
- [10] P. Nanda. "Supporting QoS Guarantees Using Traffic Engineering and Policy Based Routing". In: *International Conference on Computer Science and Software Engineering* (2008).
- [11] M. T. Z. Win, Y. Ishibashi, and K. T. Mya. "QoS-aware Traffic Engineering in Software Defined Networks". In: *2019 25th Asia-Pacific Conference on Communications (APCC)* (2019), pp. 171–176.

- [12] S. Tomovic and I. Radusinovic. "Toward a Scalable, Robust, and QoS-Aware Virtual-Link Provisioning in SDN-Based ISP Networks". In: *IEEE Transactions on Network and Service Management* 16.3 (2019), pp. 1032–1045.
- [13] M. Bagaa et al. "On SDN-Driven Network Optimization and QoS Aware Routing Using Multiple Paths". In: *IEEE Transactions on Wireless Communications* 19.7 (2020), pp. 4700–4714.
- [14] J. Y. Le Boudec. *Rate adaptation, Congestion Control and Fairness: A Tutorial*. [https://icalwww.epfl.ch/PS\\_files/LEB3132.pdf](https://icalwww.epfl.ch/PS_files/LEB3132.pdf). Nov. 2020.
- [15] A. Verma and N. Bhardwaj. "A Review on Routing Information Protocol (RIP) and Open Shortest Path First (OSPF) Routing Protocol". In: *International Journal of Future Generation Communication and Networking* 9 (Apr. 2016), pp. 161–170.
- [16] A. Dumka et al. "Implementation of QoS algorithm in integrated services (IntServ) MPLS network". In: *2nd International Conference on Computing for Sustainable Global Development (INDIACom)* (2015).
- [17] *New Terminology and Clarifications for Diffserv (RFC3260)*. <https://tools.ietf.org/html/rfc3260>.
- [18] *The Differentiated Services Model (DiffServ)*. [http://www.cisco.com/en/US/products/ps6610/products\\_data\\_sheet09186a00800a3e30.html](http://www.cisco.com/en/US/products/ps6610/products_data_sheet09186a00800a3e30.html).
- [19] *DiffServ - The Scalable End-to-End QoS Model*. [http://www.cisco.com/en/US/technologies/tk543/tk766/technologies\\_white\\_paper09186a00800a3e2f.html](http://www.cisco.com/en/US/technologies/tk543/tk766/technologies_white_paper09186a00800a3e2f.html).
- [20] X. Xiao, X. Zheng, and T. Jie. "Dynamic resource allocation algorithm of virtual networks in edge computing networks". In: *Personal and Ubiquitous Computing* (Aug. 2019).
- [21] S. Xu et al. "Load-Balancing and QoS Based Dynamic Resource Allocation Method for Smart Grid Fiber-Wireless Networks". In: *Chinese Journal of Electronics* 28.6 (2019), pp. 1234–1243.
- [22] M. Wooldridge. *An Introduction to MultiAgent Systems*. John Wiley & Sons Ltd, 2002.
- [23] Z. Jing. "Centralized routing and distributed routing protocol for dynamic routing". In: *World Automation Congress* (2012).
- [24] *Comments on the Usefulness of Simple Best-Effort Traffic (RFC5290)*. <https://www.ietf.org/rfc/rfc5290.txt>.

- [25] *Support of IP-based services using IP transfer capabilities*. <https://www.itu.int/rec/T-REC-Y.1241/en>.
- [26] *A Framework for QoS-based Routing in the Internet*. <https://tools.ietf.org/html/rfc2386>.
- [27] *Policy Quality of Service (QoS) Information Model*. <https://tools.ietf.org/html/rfc3644>.
- [28] *Integrated Services in the Internet Architecture: an Overview (RFC1633)*. <https://tools.ietf.org/html/rfc1633>.
- [29] *Resource ReSerVation Protocol (RFC2005)*. <https://tools.ietf.org/html/rfc2205>.
- [30] *An Architecture for Differentiated Services (RFC2475)*. <https://tools.ietf.org/html/rfc2475>.
- [31] *Signaled QoS (Using RSVP)*. [http://www.cisco.com/en/US/products/ps6611/products\\_white\\_paper09186a00800ade1a.shtml](http://www.cisco.com/en/US/products/ps6611/products_white_paper09186a00800ade1a.shtml).
- [32] *Inter-Domain MPLS and GMPLS Traffic Engineering - Resource Reservation Protocol-Traffic Engineering (RSVP-TE) Extensions (RFC5151)*. <https://tools.ietf.org/html/rfc5151>.
- [33] *RSVP-TE: Extensions to RSVP for LSP Tunnels (RFC3209)*. <https://tools.ietf.org/html/rfc3209>.
- [34] *Overview and Principles of Internet Traffic Engineering (RFC3272)*. <https://tools.ietf.org/html/rfc3272>.
- [35] M. A. S. Santos et al. "Decentralizing SDN's control plane". In: *39th Annual IEEE Conference on Local Computer Networks* (2014).
- [36] K. Theviyanthan et al. "BlockFlow: A decentralized SDN controller using blockchain". In: *International Journal of Scientific and Research Publications (IJ-SRP)* 10 (2020).
- [37] B. Bhattacharya and D. Das. "QoS Enhancement using Embedded Agent in OpenFlow based SDN Node". In: *IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT)* (2015).
- [38] *Definitions of terms related to quality of service*. <https://www.itu.int/rec/T-REC-E.800-200809-I>.
- [39] A. M. Elnaka and Q. H. Mahmoud. "Real-time traffic classification for unified communication networks". In: *International Conference on Selected Topics in Mobile and Wireless Networking (MoWNeT)* (2013).

- [40] *How QoS Works*. <https://technet.microsoft.com/en-us/library/cc728211%28v=ws.10%29.aspx>.
- [41] J. Tapolcai et al. "Quality of resilience (QoR): NOBEL approach to the multi-service resilience characterization". In: *2nd International Conference on Broadband Networks* (2005).
- [42] R. Prabakaran and J. B. Evans. "Experiences with class of service (CoS) translations in IP/MPLS networks". In: *Proceedings LCN 2001. 26th Annual IEEE Conference on Local Computer Networks* (2001).
- [43] *Definition of Quality of Experience (QoE)*. <https://www.itu.int/rec/T-REC-P.10-200701-S!Amd1>.
- [44] R. Stankiewicz and A. Jajszczyk. "A survey of QoE assurance in converged networks". In: *Computer Networks* 55.7 (2011), pp. 1459–1473.
- [45] S. Balasubramaniam et al. "An evaluation of parameterized gradient based routing with QoE monitoring for multiple IPTV providers". In: *IEEE Transactions on Broadcasting, tom 57, nr 2, pp. 183-194, 2011*. 57.2 (2011), pp. 183–194.
- [46] *Methods for subjective determination of transmission quality*. <https://www.itu.int/rec/T-REC-P.800-199608-I/en>.
- [47] H. G. Msakni and H. Youssef. "Is QoE estimation based on QoS parameters sufficient for video quality assessment?" In: *9th International Wireless Communications and Mobile Computing Conference (IWCMC)* (2013).
- [48] R. Stankiewicz et al. "Influence of traffic management solutions on Quality of Experience for prevailing overlay applications". In: *7th EURO-NGI Conference on Next Generation Internet Networks* (2011).
- [49] O. B. Maia, H. C. Yehia, and L. de Errico. "A concise review of the quality of experience assessment for video streaming". In: *Computer Communications* 57 (2015), pp. 1–12.
- [50] S. Tomovic and I. Radusinovic. "Fast and efficient bandwidth-delay constrained routing algorithm for SDN networks". In: 2016, pp. 303–311.
- [51] *Requirements for IP Version 4 Routers (RFC1812)*. <https://www.ietf.org/rfc/rfc1812.txt>.
- [52] *Cisco's Enhanced Interior Gateway Routing Protocol (RFC7868)*. <https://tools.ietf.org/html/rfc7868>.
- [53] *A Border Gateway Protocol 4 (RFC4271)*. <https://tools.ietf.org/html/rfc4271>.



- [54] M. Soliman et al. "Exploring source routed forwarding in SDN-based WANs". In: *IEEE International Conference on Communications (ICC)* (2014).
- [55] P. Amaral, L. Bernardo, and P. Pinto. "Multipath policy routing using destination based hop-by-hop forwarding". In: *21st IEEE International Conference on Network Protocols (ICNP)* (2013).
- [56] *Simple Mail Transfer Protocol (RFC1104)*. <https://tools.ietf.org/html/rfc1104>.
- [57] P. Purkayastha, P. Hovareshti, and J. S. Baras. "A policy-based routing scheme for multi-robot systems". In: *16th Mediterranean Conference on Control and Automation* (2008).
- [58] G. Katsikogiannis, S. Mitropoulos, and C. Douligeris. "Policy-based QoS management for SLA-driven adaptive routing". In: *Journal of Communications and Networks* 15.3 (2013), pp. 301–311.
- [59] B. Statovci-Halimi et al. "A dynamic SLA management approach for MPLS networks". In: *Proceedings of the Eighth IEEE Symposium on Computers and Communications. ISCC 2003* (2003).
- [60] L. Bin et al. "Design and Experiments of Multiple Service Transport with Multi-Agent Negotiations". In: *IEEE-RIVF International Conference on Computing and Communication Technologies* (2009).
- [61] Y. Chung, M. H. Park, and E. H. Paik. "A QoS negotiable service framework for multimedia services connected through subscriber networks". In: *IEEE International Symposium on Consumer Electronics* (2006).
- [62] V. Fineberg, C. Chen, and X. Xiao. "An end-to-end QoS architecture with the MPLS-based core". In: *IEEE Workshop on IP Operations and Management* (2002).
- [63] D. Huijun, Q. Hua, and Z. Jihong. "QoS routing algorithm with multi-dimensions for overlay networks". In: *China Communications* 10.100 (2013), pp. 167–176.
- [64] V. Fineberg. "A practical architecture for implementing end-to-end QoS in an IP network". In: *IEEE Communications Magazine* 40.1 (2002), pp. 122–130.
- [65] B. Sanso, C. Awad, and A. Girard. "Can DiffServ guarantee IP QoS under failures?" In: *IEEE Network* 20.4 (2006), pp. 32–40.
- [66] C. Wijaya. "Performance Analysis of Dynamic Routing Protocol EIGRP and OSPF in IPv4 and IPv6 Networ". In: *First International Conference on Informatics and Computational Intelligence* (2011).

- [67] A. M. Hailan, T. Nadia, and T. V. Vavenko. "Design of dynamic routing scheme in telecommunication networks". In: *The Experience of Designing and Application of CAD Systems in Microelectronics* (2015).
- [68] J. Zhang et al. "Dynamic hybrid routing: Achieve load balancing for changing traffic demands". In: *IEEE 22nd International Symposium of Quality of Service (IWQoS)* (2014).
- [69] G. P. S. Kalyan and D. V. V. Prasad. "Optimal selection of Dynamic Routing Protocol with real time case studies". In: *International Conference on Recent Advances in Computing and Software Systems* (2012).
- [70] L. Volker, M. Scholler, and M. Zitterbart. "Introducing QoS mechanisms into the IPsec packet processing". In: *32nd IEEE Conference on Local Computer Networks (LCN 2007)* (2007).
- [71] R. Suryasaputra, A. A. Kist, and R. J. Harris. "Verification of MPLS traffic engineering techniques". In: *13th IEEE International Conference on Networks* (2005).
- [72] J. M. Chung. "Analysis of MPLS traffic engineering". In: *Proceedings of the 43rd IEEE Midwest Symposium on Circuits and Systems* (2000).
- [73] W. Sun, P. Bhaniramka, and R. Jain. "Quality of service using traffic engineering over MPLS: an analysis". In: *Proceedings 25th Annual IEEE Conference on Local Computer Networks. LCN 2000* (2000).
- [74] M. Lv and W. Ji. "Research on GMPLS traffic engineering mechanism". In: *IEEE 13th International Conference on Communication Technology* (2011).
- [75] B. Berde, D. Papadimitriou, and M. Jager. "Traffic Engineering Element for GMPLS Networks". In: *IEEE/IFIP Network Operations and Management Symposium NOMS* (2006).
- [76] M. Vigoureux et al. "Multilayer traffic engineering for GMPLS-enabled networks". In: *IEEE Communications Magazine* 43.7 (2005), pp. 44–50.
- [77] I. Ramadza, J. Ozegovic, and V. Pekic. "Network performance monitoring within MPLS traffic engineering enabled networks". In: *23rd International Conference on Software, Telecommunications and Computer Networks (SoftCOM)* (2005).
- [78] E. Kim et al. "Traffic measurements supporting end-to-end QoS requirements in MPLS networks". In: *The 6th International Conference on Advanced Communication Technology* (2004).

- [79] S. H. Lim et al. "Traffic engineering enhancement to QoS-OSPF in DiffServ and MPLS networks". In: *IEEE Proceedings - Communications* 151.1 (2004), pp. 101–106.
- [80] B. Dekeris and L. Narbutaite. "Traffic control mechanism within MPLS networks". In: *26th International Conference on Information Technology Interfaces* (2004).
- [81] A. M. Bongale, N. Nithin, and L. S. Jyothi. "Traffic prioritization in MPLS enabled OSPF network". In: *World Congress on Information and Communication Technologies* (2012).
- [82] D. Shimazaki et al. "Traffic-driven virtual network topology reconfiguration for GMPLS network". In: *Workshop on High Performance Switching and Routing* (2006).
- [83] L. Zhenyu and Z. Zhongzao. "Bandwidth constrained QoS routing scheme for MPLS traffic engineering". In: *Journal of Systems Engineering and Electronics* 15.4 (2012), pp. 735–740.
- [84] B. J. Oommen, S. Misra, and O.-C. Granmo. "Routing Bandwidth-Guaranteed Paths in MPLS Traffic Engineering: A Multiple Race Track Learning Approach". In: *IEEE Transactions on Computers* 56.7 (2007).
- [85] H. Y. Cho, J. Y. Lee, and B. C. Kim. "Multi-path constraint-based routing algorithms for MPLS traffic engineering". In: *IEEE International Conference on Communications* (2003).
- [86] P. H. Ho and H. T. Mouftah. "Capacity-balanced alternate routing for MPLS traffic engineering". In: *Proceedings ISCC 2002 Seventh International Symposium on Computers and Communications* (2002).
- [87] G. M. Lee and J. K. Choi. "Multipath Routing for Traffic Engineering with Flow Classification in GMPLS Network". In: *2006 Asia-Pacific Conference on Communications* (2006).
- [88] X. Li et al. "DiffServ-aware MPLS traffic engineering". In: *The 3rd International Conference on Information Sciences and Interaction Sciences* (2010).
- [89] S. Butenweg. "Two distributed reactive MPLS traffic engineering mechanisms for throughput optimization in best effort MPLS networks". In: *Proceedings of the Eighth IEEE Symposium on Computers and Communications. ISCC 2003* (2003).
- [90] C. Awad, B. Sanso, and A. Girard. "Differentiated reliability in traffic engineered MPLS and DiffServ-aware next generation networks". In: *7th International Workshop on Design of Reliable Communication Networks* (2009).

- [91] W. H. Hsu et al. "Multiple Paths Selection Algorithm for DiffServ-Aware MPLS Traffic Engineering". In: *5th International Conference on Wireless Communications, Networking and Mobile Computing* (2009).
- [92] O. V. Lemeshko, A. M. Hailan, and O. V. Starkova. "Multi-level traffic management in the MPLS-TE DiffServ network". In: *11th International Conference The Experience of Designing and Application of CAD Systems in Microelectronics* (2011).
- [93] W. Vallat and S. Ganti. "Aggregation of traffic classes in MPLS networks". In: *24th Canadian Conference on Electrical and Computer Engineering* (2011).
- [94] M. Moh, B. Wei, and J. H. Zhu. "Supporting differentiated services with per-class traffic engineering in MPLS". In: *Proceedings Tenth International Conference on Computer Communications and Networks* (2001).
- [95] R. Tomar et al. "Traffic management in MPLS network using GNS simulator using class for different services". In: *2nd International Conference on Computing for Sustainable Global Development* (2005).
- [96] X. Liu. "Load distribution models of the MPLS traffic". In: *GLOBECOM '05. IEEE Global Telecommunications Conference* (2005).
- [97] K. Long, Z. Zhang, and S. Cheng. "Load balancing algorithms in MPLS traffic engineering". In: *IEEE Workshop on High Performance Switching and Routing* (2001).
- [98] A. A. Adewale et al. "Improvement of Quality of Service (QoS) over a Wide Area Network (WAN) using Multiprotocol Label Switching Traffic Engineering (MPLS-TE)". In: *International Journal of Computer Applications* 88.10 (2014), pp. 14–18.
- [99] I. Bryskin. "From GMPLS to transport SDN". In: *Optical Fiber Communications Conference and Exhibition (OFC)* (2016).
- [100] M. Karakus and A. Durrezi. "Quality of Service (QoS) in Software Defined Networking (SDN): A survey". In: *Journal of Network and Computer Applications* 80 (2017), pp. 200–218.
- [101] M. J. Todorovic and N. D. Krajnovic. "Simulation analysis of SDN network capabilities". In: *21st Telecommunications Forum Telfor (TELFOR)* (2013).
- [102] H. Zhang and J. Yan. "Performance of SDN Routing in Comparison with Legacy Routing Protocols". In: *International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery* (2015).

- [103] M. Jarschel et al. "Interfaces, attributes, and use cases: A compass for SDN". In: *IEEE Communications Magazine* 52.6 (2014), pp. 210–217.
- [104] K. Bakshi. "Considerations for Software Defined Networking (SDN): Approaches and use cases". In: *IEEE Aerospace Conference* (2013).
- [105] S. Sasidharan and S. K. Chandra. "Defining future SDN based network management systems characterization and approach". In: *Fifth International Conference on Computing, Communications and Networking Technologies (ICCCNT)* (2014).
- [106] P. Danielis et al. "Emulation of SDN-supported automation networks". In: *IEEE 20th Conference on Emerging Technologies & Factory Automation (ETFA)* (2015).
- [107] Y. W. Ma et al. "SDN test cases development and implementation". In: *17th International Conference on Advanced Communication Technology (ICACT)* (2015).
- [108] B. Varga et al. "Hybrid-SDN for packet transport: The horizontal split". In: *17th International Conference on Transparent Optical Networks (ICTON)* (2015).
- [109] Y. Wang and I. Matta. "SDN Management Layer: Design Requirements and Future Direction". In: *IEEE 22nd International Conference on Network Protocols* (2014).
- [110] G. Poullos et al. "Autonomics and SDN for self-organizing networks". In: *11th International Symposium on Wireless Communications Systems (ISWCS)* (2014).
- [111] Y. Jimenez, C. Cervello-Pastor, and A. J. Garcia. "On the controller placement for designing a distributed SDN control layer". In: *IFIP Networking Conference* (2014).
- [112] K. Govindarajan et al. "Realizing the Quality of Service (QoS) in Software-Defined Networking (SDN) based Cloud infrastructure". In: *2nd International Conference on Information and Communication Technology (ICoICT)* (2014).
- [113] S. Baik et al. "Adaptive flow monitoring in SDN architecture". In: *17th Asia-Pacific Network Operations and Management Symposium (APNOMS)* (2015).
- [114] A. Gelberger, N. Yemini, and R. Giladi. "Performance Analysis of Software-Defined Networking (SDN)". In: *IEEE 21st International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems* (2013).
- [115] H. Li et al. "An autonomic management architecture for SDN-based multi-service network". In: *IEEE Globecom Workshops (GC Wkshps)* (2013).

- [116] H. Eghbali and V. W. S. Wong. "Bandwidth allocation and pricing for SDN-enabled home networks". In: *IEEE International Conference on Communications (ICC)* (2015).
- [117] R. Mijumbi et al. "Dynamic resource management in SDN-based virtualized networks". In: *10th International Conference on Network and Service Management (CNSM) and Workshop* (2014).
- [118] F. Tao, B. Jun, and W. Ke. "Allocation and scheduling of network resource for multiple control applications in SDN". In: *China Communications* 12.6 (2015), pp. 85–95.
- [119] P. Sharma et al. "Enhancing network management frameworks with SDN-like control". In: *IFIP/IEEE International Symposium on Integrated Network Management (IM 2013)* (2013).
- [120] M. Qilin and S. Weikang. "A Load Balancing Method Based on SDN". In: *Seventh International Conference on Measuring Technology and Mechatronics Automation* (2015).
- [121] A. Craig et al. "Load balancing for multicast traffic in SDN using real-time link cost modification". In: *IEEE International Conference on Communications (ICC)* (2015).
- [122] S. Baik, C. Hwang, and Y. Lee. "SDN-based architecture for end-to-end path provisioning in the mixed circuit and packet network environment". In: *16th Asia-Pacific Network Operations and Management Symposium* (2014).
- [123] Y. Jinyao et al. "HiQoS: An SDN-based multipath QoS solution". In: *China Communications* 12.5 (2015), pp. 123–133.
- [124] M. R. Celenlioglu and H. A. Mantar. "An SDN Based Intra-Domain Routing and Resource Management Model". In: *IEEE International Conference on Cloud Engineering* (2015).
- [125] J. M. Wang et al. "SDN-based multi-class QoS-guaranteed inter-data center traffic management". In: *IEEE 3rd International Conference on Cloud Networking (CloudNet)* (2014).
- [126] H. E. Egilmez and A. M. Tekalp. "Distributed QoS Architectures for Multimedia Streaming Over Software Defined Networks". In: *IEEE Transactions on Multimedia* 16.6 (2014), pp. 1597–1609.
- [127] M. A. Karaman et al. "Quality of service control and resource prioritization with Software Defined Networking". In: *Proceedings of the 2015 1st IEEE Conference on Network Softwarization (NetSoft)* (2015).

- [128] B. Gorkemli et al. "QoS control and prioritization with SDN". In: *23rd Signal Processing and Communications Applications Conference (SIU)* (2015).
- [129] K. Sood et al. "A General QoS Aware Flow-Balancing and Resource Management Scheme in Distributed Software-Defined Networks". In: *IEEE Access* 4 (2016), pp. 7176–7185.
- [130] S. Scott-Hayward, G. O'Callaghan, and S. Sezer. "Sdn Security: A Survey". In: *IEEE SDN for Future Networks and Services (SDN4FNS)* (2013).
- [131] S. Scott-Hayward. "Design and deployment of secure, robust, and resilient SDN controllers". In: *Proceedings of the 2015 1st IEEE Conference on Network Softwarization (NetSoft)* (2015).
- [132] C. Röpke and T. Holz. "Retaining control over SDN network services". In: *International Conference and Workshops on Networked Systems (NetSys)* (2015).
- [133] B. Villain, J. Ridoux, and G. Pujolle. "SDN based network access controller: A first milestone". In: *6th International Conference on the Network of the Future (NOF)* (2015).
- [134] S. Civanlar et al. "Distributed management of service-enabled flow-paths across multiple SDN domains". In: *European Conference on Networks and Communications (EuCNC)* (2015).
- [135] R. Bennesby et al. "Innovating on Interdomain Routing with an Inter-SDN Component". In: *IEEE 28th International Conference on Advanced Information Networking and Applications* (2014).
- [136] K. Phemius, M. Bouet, and J. Leguay. "DISCO: Distributed multi-domain SDN controllers". In: *IEEE Network Operations and Management Symposium (NOMS)* (2014).
- [137] N. Giroux. "A tutorial on ATM traffic management". In: *Canadian Journal of Electrical and Computer Engineering* 21.3 (1996), pp. 103–106.
- [138] Y.-C. Chen et al. "Traffic control through in-service monitoring for QoS improvement in ATM networks". In: *Journal of Communications and Networks* 1.4 (1999), pp. 286–294.
- [139] T. Sanguankotchakorn and B. Chiewthanakul. "Bandwidth allocation scheme in ATM networks based on virtual path for different quality of service (QoS) requirements". In: *IEEE International Conference on Networks, ICON 2001* (2001).

- [140] S. Pillalamarri and S. Ghosh. "The impact of source traffic distribution on quality of service (QoS) in ATM networks". In: *IEEE International Conference on Communications* (2001).
- [141] R. H. Hwang, M. X. Chen, and C. M. Hsu. "Routing in ATM networks with multiple classes of QoS". In: *Global Telecommunications Conference* (2000).
- [142] S. L. Tan, C. K. Tham, and L. H. Ngoh. "QoS-based connection set-up in ATM networks". In: *Proceedings IEEE International Conference on Networks 2000 (ICON 2000)* (2000).
- [143] J. Tanaka et al. "QoS interworking for Diffserv over ATM access networks". In: *Global Telecommunications Conference* (2000).
- [144] J. M. Tsai and C. Y. Lee. "Novel architecture for ATM QoS management". In: *IEEE Proceedings - Communications* 144.6 (1997), pp. 412–418.
- [145] F. Schlake and C. Ruland. "A security protocol providing QoS in ATM networks". In: *The 8th International Conference on Communication Systems* (2002).
- [146] J. L. Marzo et al. "Distributed Architecture for Dynamic Resource Management". In: *China Communications* (2005), pp. 61–72.
- [147] R. Wójcik and J. Domżał. "Performance evaluation of traffic management mechanisms for Peer-to-Peer networks". In: *AEU - International Journal of Electronics and Communications* 70.2 (2016), pp. 162–171.
- [148] A. C. Begen et al. "Multi-path selection for multiple description video streaming over overlay networks". In: *Signal Processing: Image Communication* 20.1 (2005), pp. 39–60.
- [149] T. Miyamura et al. "A multi-layer disjoint path selection algorithm for highly reliable carrier services". In: *Global Telecommunications Conference, GLOBECOM '04. IEEE* (2004).
- [150] H. Wang et al. "An End-to-End QoS Guarantee Scheme in Heterogeneous Networks". In: *Third International Conference on Instrumentation, Measurement, Computer, Communication and Control* (2013).
- [151] C. Bouchat, S. van den Bosch, and T. Pollet. "QoS in DSL access". In: *IEEE Communications Magazine* 41.9 (2003), pp. 108–114.
- [152] S. Wright and T. Anschutz. "QoS requirements in DSL networks". In: *Global Telecommunications Conference, GLOBECOM '03. IEEE* (2003).
- [153] J. L. Poza, J. L. Posadas, and J. E. Simo. "From the Queue to the Quality of Service Policy: A Middleware Implementation". In: *10th International Work-Conference on Artificial Neural Networks* (2009).



- [154] J. A. Perez et al. "A Network and Data Link Layer Infrastructure Design to Improve QoS for Real Time Collaborative Systems". In: *Advanced Int'l Conference on Telecommunications and Int'l Conference on Internet and Web Applications and Services (AICT-ICIW'06)* (2006).
- [155] M. Prangl, I. Kofler, and H. Hellwagner. "Towards QoS Improvements of TCP-Based Media Delivery". In: *Fourth International Conference on Networking and Services (icns 2008)* (2008).
- [156] N. Fujita and A. Iwata. "Adaptive and efficient multiple path pre-computation for QoS routing protocols". In: *Global Telecommunications Conference, GLOBECOM '01. IEEE* (2001).
- [157] F. A. Elgeldawy, G. M. Salama, and M. F. Abdelfattah. "Performance of QoS parameters for IPTV through NGN". In: *IEEE Student Conference on Research and Development (SCOReD)* (2016).
- [158] Z. Liu, P. Li, and T. Zhao. "The Study on QoS in the Networked Test System". In: *Fourth International Conference on Instrumentation and Measurement, Computer, Communication and Control* (2014).
- [159] B. Khoshnevis and B. H. Khalaj. "Optimum QoS classes in interworking of next generation networks". In: *Journal of Communications and Networks* 9.4 (2007), pp. 438–445.
- [160] S. Franklin and A. Graesser. "Is It an agent, or just a program?: A taxonomy for autonomous agents". In: *International Workshop on Agent Theories, Architectures, and Languages* (1996).
- [161] H. S. Nwana. "Software Agents: An Overview". In: *Knowledge Engineering Review* 11.3 (1996), pp. 1–40.
- [162] G. Weiss. *Multiagent system*. Massachusetts Institute of Technology, 2013.
- [163] S. Russell i P. Norvig. *Artificial Intelligence A Modern Approach*. Pearson Education Inc., 2010.
- [164] K. Cetnarowicz. *Paradygmat agentowy w informatyce. Koncepcje, podstawy i zastosowania*. Akademicka Oficyna Wydawnicza EXIT, 2012.
- [165] M. Mangili et al. "Bandwidth and Cache Leasing in Wireless Information-Centric Networks: A Game-Theoretic Study". In: *IEEE Transactions on Vehicular Technology* 66.1 (2017), pp. 679–695.
- [166] R. T. B. Ma. "Efficient Resource Allocation and Consolidation with Selfish Agents: An Adaptive Auction Approach". In: *IEEE 36th International Conference on Distributed Computing Systems (ICDCS)* (2016).

- [167] P. Klemperer. *Auctions: Theory and Practice*. Mar. 2004.
- [168] M. P. Wellman. *Trading agents*. Morgan & Claypool, 2011.
- [169] J. Sow, P. Anthony, and C. Mun Ho. "Competition among intelligent agents and standard bidders with different risk behaviors in simulated English auction marketplace". In: *IEEE International Conference on Systems, Man and Cybernetics* (2010).
- [170] E. Guerci, A. Kirman, and S. Moulet. "An agent-based model for sequential Dutch auctions". In: *Winter Simulations Conference (WSC)* (2013).
- [171] S. Arai and T. Miura. "An intelligent agent for combinatorial auction". In: *11th International Conference on Hybrid Intelligent Systems (HIS)* (2011).
- [172] G. Zachariadis and J. A. Barria. "Demand management for telecommunications services". In: *Computer Networks* 51.12 (2007), pp. 3507–3524.
- [173] V. Radonjic and M. Dogatovic. "A Novel Simulation Model for Pricing Different QoS Levels in IP Networks". In: *Advances in Electrical and Computer Engineering* 20 (Feb. 2020), pp. 27–34.
- [174] A. Gupta and L. Zhang. "Pricing for end-to-end assured bandwidth services". In: *International Journal of Information Technology & Decision Making* 7.2 (2008).
- [175] T. Li, Y. Iraqi, and R. Boutaba. "Tariff-Based Pricing and Admission Control for DiffServ Networks". In: *Integrated Network Management VIII, Springer US* (2003).
- [176] *Hypertext Transfer Protocol - HTTP/1.1 (RFC2616)*. <https://tools.ietf.org/html/rfc2616>.
- [177] *Simple Mail Transfer Protocol (RFC5321)*. <https://tools.ietf.org/html/rfc5321>.
- [178] S. S. Manvi and P. Venkataram. "Applications of agent technology in communications: a review". In: *Computer Communications* 27.15 (2004), pp. 1493–1508.
- [179] D. Bogdanovic. "Autonomic agent for transport networks". In: *Optical Fiber Communications Conference and Exhibition (OFC)* (2015).
- [180] R. Olfati-Saber, J. A. Fax, and R. M. Murray. "Consensus and Cooperation in Networked Multi-Agent Systems". In: *Proceedings of the IEEE* 95.1 (2007), pp. 215–233.

- [181] D. Mitrovic, M. Ivanovic, and Z. Geler. "Agent-Based Distributed Computing for Dynamic Networks". In: *Information technology and control* 43.1 (2014), pp. 88–97.
- [182] J. Marzo et al. "Automatic Self-Configuration of the Logical Network using Distributed Software Agents". In: *Global Telecommunications Conference, GLOBECOM '04. IEEE* (2004).
- [183] R. Valadas and P. Salvador. "A Distributed Scheme for Value-Based Bandwidth Reconfiguration". In: *First Euro-NF Workshop, FITraME*n (2008).
- [184] N. Goldberg et al. "Local Unicast Routing Control Agent". In: *IEEE Military Communications Conference* (2009).
- [185] S. Khurana et al. "Performance Evaluation of a Unicast Routing Control Agent for Proactive Diverse Link Selection". In: *IEEE Military Communications conference* (2006).
- [186] S. S. Manvi and P. Venkataram. "An Agent-Based Best Effort Routing Technique for Load Balancing". In: *Informatica* 17.3 (2006), pp. 407–426.
- [187] S. M. Dragos and R. C. Dragos. "Modern routing techniques for future QoS enabled networks". In: *Carpathian Journal of Mathematics* 21.1 (2005).
- [188] L. Wai-tun. *Mobile agent-based routing for the next-generation Intern.* Department of Computing, The Hong Kong Polytechnic University, 2008.
- [189] S. M. Dragos. *Scalable QoS routing in MPLS networks using mobile code.* Dublin City University, 2006.
- [190] S. Ladhe, S. Devane, and M. Chatterjee. "Improved Agent based Explicit Path Computation Algorithm for MPLS-TE". In: *16th IEEE International Conference on Networks* (2008).
- [191] B. Li, Y. Zu, and S. Huang. "Design and Experiments of Multiple Service Transport with Multi-Agent Negotiations". In: *IEEE-RIVF International Conference on Computing and Communication Technologies* (2009).
- [192] S. Makki and S. V. Wunnava. "Application of Mobile Agents in Managing the Traffic in the Network and Improving the Reliability and Quality of Service". In: *IAENG International Journal of Computer Science* 33.1 (2007), pp. 135–138.
- [193] F. Krief and Z. Jrad. "An Intelligent Policy-based networking environment for dynamic negotiation, provisioning and control of QoS". In: *Conference on Network Control and Engineering for QoS, Security and Mobility (Net-Con 2002)* (2002).

- [194] F. S. Tegueu et al. "Towards Application Driven Networking". In: *IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN)* (2016).
- [195] K. S. Lim, C. Adam, and R. Stadler. "Decentralizing Network Management". In: *IEEE electronic Transactions on Network and Service Management (eTNSM)* (2004).
- [196] P. Vila et al. "Multi-Agent System Co-ordination in a Distributed Network Resource Management Scenario". In: (2005).
- [197] T. C. Du, E. Y. Li, and A. P. Chang. "Mobile agents in distributed network management". In: *Communications of the ACM* 46.7 (2003), pp. 127–132.
- [198] D. Mitrovic et al. "Improving fault-tolerance of distributed multi-agent systems with mobile network-management agents". In: *Proceedings of the International Multiconference on Computer Science and Information Technology* (2010).
- [199] M. Ganzha et al. "Agent-based approaches to managing fault-tolerant networks of distributed multi-agent systems". In: *Multiagent and Grid Systems* 7.6 (2011), pp. 203–218.
- [200] D. Gavalas, G. E. Tsekouras, and C. Anagnostopoulos. "A mobile agent platform for distributed network and systems management". In: *Journal of Systems and Software* 82.2 (2009), pp. 355–371.
- [201] R. Stephan, P. Ray, and N. Paramesh. "Network management platform based on mobile agents". In: *International Journal of Network Management* 14.1 (2004), pp. 59–73.
- [202] C. Valliyammai et al. "Efficient Management of Network Resources Using Autonomous Agents in Grid". In: *IJCSNS International Journal of Computer Science and Network Security* 8.4 (2008), pp. 178–184.
- [203] J. Smith et al. "Overlay network resource allocation using a decentralized market-based approach". In: *Future Generation Computer Systems* 28.1 (2012), pp. 24–35.
- [204] H. He, K. Xu, and Y. Liu. "Internet resource pricing models, mechanisms, and methods". In: *Networking Science* 1.1 (2012), pp. 48–66.
- [205] R. Davoyan and J. Altmann. "Real-Time Market Model for Pricing Differentiated Services". In: *Fourth International Conference on Networking and Services (icns 2008)* (2008).
- [206] VyOS – Open source router and firewall platform. <https://vyos.io/>.
- [207] TRex Realistic Traffic Generator. <https://trex-tgn.cisco.com/>.

- [208] NS3 network simulator. <https://www.nsnam.org/>.
- [209] C. Sammut and I. G. Webb. *Encyclopedia of Machine Learning*. Springer, Boston, MA, 2010.
- [210] Y. Sasaki. "The truth of the F-measure". In: *Teach Tutor Mater* (Jan. 2007).
- [211] Sklearn F1\_score. [https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1\\_score.html](https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html).
- [212] C. Cortes and V. Vapnik. "Support-vector networks". In: *Machine Learning* 20 (1995), pp. 273–297.
- [213] P. Pławiak, M. Abdar, and U. R. Acharya. "Application of New Deep Genetic Cascade Ensemble of SVM Classifiers to Predict the Australian Credit Scoring". In: *Applied Soft Computing* 84 (2019).
- [214] C. C. Chang and C. J. Lin. "LIBSVM: A library for support vector machines". In: *ACMTrans. Intell. Syst. Technol. (TIST)* 2 (2011).
- [215] N.S. Altman. "An introduction to kernel and nearest neighbor non parametric regression". In: *Am Stat* 46 (1992), pp. 175–185.
- [216] L. Breiman. "Random Forests". In: *Machine Learning* 45 (2001), pp. 5–32.
- [217] R. Tibshirani et al. "Diagnosis of multiple cancer types by shrunken centroids of gene expression". In: *Proceedings of the National Academy of Sciences of the United States of America* 99 (2002).
- [218] P. Geurts, D. Ernst, and L. Wehenkel. "Extremely randomized trees". In: *Machine Learning* 63 (2006), pp. 3–42.
- [219] R. E. Fan et al. "LIBLINEAR: A library for large linear classification". In: *Journal of Machine Learning Research* 9 (2008), pp. 1871–1874.
- [220] L. Breiman et al. *Classification and Regression Trees*. Wadsworth, Belmont, CA, 1984.
- [221] D. H. Wolpert. "Stacked generalization". In: *Neural networks* 5 (1992), pp. 241–259.
- [222] Y. Hsiang-Fu, H. Fang-Lan, and L. Chih-Jen. "Dual coordinate descent methods for logistic regression and maximum entropy models". In: *Machine Learning* 85.1-2 (2011), pp. 41–75.
- [223] Z. Yong et al. "A Weighted Voting Classifier Based on Differential Evolution". In: *Artificial Intelligence and Data Mining 2014* (2014).
- [224] A. Onan, S. Korukoglu, and H. Bulut. "A multiobjective weighted voting ensemble classifier based on differential evolution algorithm for text sentiment classification". In: *Expert Systems with Applications* 62 (2016), pp. 1–16.