RZECZPOSPOLITA POLSKA

Urząd Patentowy Rzeczypospolitej Polskiej

(12) **TŁUMACZENIE PATENTU EUROPEJSKIEGO**

(19) **PL**

(11) **PL/EP 3531622**

(13) **T3**

(51) Int.Cl.
H04L 12/24 (2006.01)

(96) Data i numer zgłoszenia patentu europejskiego:
23.02.2018 18461520.1

(97) O udzieleniu patentu europejskiego ogłoszono:
08.01.2020 Europejski Biuletyn Patentowy 2020/02
EP 3531622 B1

(54) Tytuł wynalazku:

Sposób obsługi przepływu pakietów w programowalnej sieci komputerowej SDN, produkt programu komputerowego oraz programowalna sieć komputerowa SDN

(30) Pierwszeństwo:

(43) Zgłoszenie ogłoszono:

28.08.2019 w Europejskim Biuletynie Patentowym nr 2019/35

(45) O złożeniu tłumaczenia patentu ogłoszono:

29.06.2020 Wiadomości Urzędu Patentowego 2020/08

(73) Uprawniony z patentu:

Akademia Górniczo-Hutnicza im. Stanisława Staszica w Krakowie, Kraków, PL

(72) Twórca(y) wynalazku:

PIOTR BORYŁO, Kraków, PL
ARTUR LASOŃ, Kraków, PL
MICHAŁ RZEPKA, Kraków, PL

(74) Pełnomocnik:
rzecz. pat. Marek Bury
BURY & BURY KANCELARIA PATENTOWA SP. Z O.O.
ul. Juliusza Słowackiego 5/13 / 111
01-592 Warszawa

PL/EP 3531622 T3

Uwaga:

W ciągu dziewięciu miesięcy od publikacji informacji o udzieleniu patentu europejskiego, każda osoba może wnieść do Europejskiego Urzędu Patentowego sprzeciw dotyczący udzielonego patentu europejskiego. Sprzeciw wnosi się w formie uzasadnionego na piśmie oświadczenia. Uważa się go za wniesiony dopiero z chwilą wniesienia opłaty za sprzeciw (Art. 99 (1) Konwencji o udzielaniu patentów europejskich).

Sposób obsługi przepływu pakietów w programowalnej sieci komputerowej SDN, produkt programu komputerowego oraz programowalna sieć komputerowa SDN

[0001] Przedmiotem wynalazku jest sposób obsługi przepływu pakietów w programowalnej sieci komputerowej SDN, produkt programu komputerowego oraz programowalna sieć komputerowa SDN.

[0002] Koncepcja działania sieci sterowanych programowo (ang. *Software Defined Networking*) polega na rozdzieleniu warstwy sterującej siecią od warstwy przełączania danych. Warstwę sterującą stanowi tzw. sterownik (ang. *controller*), czyli logicznie scentralizowane oprogramowanie, którego zadaniem jest podejmowanie decyzji związanych z regułami przełączania ruchu w sieci. Reguły te są następnie przekazywane do przełączników sieci SDN, które tworzą warstwę przełączania danych. Podczas gdy logika obsługi ruchu implementowana jest w sterowniku, jedynym zadaniem przełączników pozostaje wdrożenie opracowanych reguł i fizyczne przeniesienie pakietów pomiędzy portem wejściowym, a wynikającym z reguły portem wyjściowym. W kontekście sieci SDN należy także zdefiniować pojęcie przepływu, czyli ciąg danych identyfikowanych na podstawie wybranych pół nagłówków w przenoszonych danych. W klasycznym podejściu przepływ jest identyfikowany przez źródłowy i docelowy adres IP, źródłowy i docelowy port, oraz wykorzystywany protokół warstwy transportowej. Z kolei sposób trasowania pakietu tj. określania jego drogi pomiędzy punktami w sieci określa się jako politykę routingu.

[0003] Znanym protokołem komunikacji pomiędzy warstwą sterującą i warstwą przełączania danych w sieci SDN jest OpenFlow.

[0004] Protokół OpenFlow jest obecnie jednym z najpopularniejszych przykładów zastosowania koncepcji SDN. Kluczowym elementem specyfikacji protokołu są definicje

formatów komunikatów oraz wymagania stawiane kompatybilnym urządzeniom. Specyfikacja ta definiuje najważniejsze elementy sieci SDN oraz powiązane z nimi terminy stosowane w opisie niniejszego przykładu wykonania wynalazku. „Sterownik SDN” stanowi centralny punkt sieci podejmujący decyzje na temat sposobu obsługi konkretnych przepływów i przekazujący je przełącznikom SDN przy użyciu protokołu OpenFlow. „Przełącznik SDN” – urządzenie sieciowe zgodne z protokołem OpenFlow, zdolne do przetwarzania, przechowywania oraz realizowania instrukcji otrzymywanych od sterownika SDN.

[0005] Utrzymanie zadowalającej jakości usług napotyka pewne wyzwania w sieciach o dynamicznie zmieniających się politykach obsługi ruchu. W określonych przypadkach występuje konieczność zmiany polityki dla nowych przepływów o określonej charakterystyce z jednoczesnym zachowaniem sposobu obsługi dotychczasowych przepływów tego samego typu. Utrzymywanie wpisów dla każdego przepływu jest kosztowne w związku z ich przechowywaniem i przeszukiwaniem, z kolei stosowanie agregowanych wpisów nie pozwala na zachowanie szczegółowej kontroli nad ruchem i utrudnia monitorowanie zdarzeń sieciowych, gdyż wszelkie akcje i obserwacje są możliwe tylko na poziomie wpisów zagregowanych. Obsługa ruchu przez sterownik w czasie rzeczywistym ma negatywny wpływ na wydajność. W szczególności dotyczy to momentu pojawienia się nowego przepływu, gdzie przy domyślnym zachowaniu, przed obsługą przepływu konieczne jest oczekiwanie na odpowiedź od sterownika. Problem ma również związek ze sposobem przechowywania wpisów w tabelach przepływów przełączników SDN i sposobem ich przeszukiwania. Zgodnie ze standardem OpenFlow mogą być one rozmieszczone w osobnych tabelach – jest to możliwość dostępna w większości przełączników programowych SDN. Na rynku dostępne są rozwiązania programowe o wydajności porównywalnej do typowych przełączników sprzętowych.

[0006] Zgodnie z podstawowymi założeniami, sieci SDN mają na celu eliminację niektórych kluczowych problemów spotykanych podczas administracji współczesnymi sieciami komputerowymi. Trudności w zarządzaniu systemami teleinformatycznymi wynikają między innymi ze stale rosnącej liczby węzłów o autonomicznym charakterze oraz skomplikowanej struktury połączeń. Tak zbudowane sieci są mało elastyczne, a ich konfiguracja staje się procesem pracochłonnym oraz wymagającym pod względem zasobów. Utrudnione jest także monitorowanie stanu poszczególnych elementów systemu oraz efektywne reagowanie na zmiany zapotrzebowań użytkowników i występujące awarie.

[0007] W patencie amerykańskim nr US8964751 udzielonym na wynalazek pt. SYSTEM AND METHOD FOR STORING FLOW ENTRIES IN HARDWARE TABLES, ujawniono sposób obsługi przepływu pakietów w sieci SDN wyposażonej w sterownik oraz przełącznik z tablicą przepływów której wpisy określają sposób obsługi pakietów i w którym stosuje się podział na wpisy szczegółowe i wpisy zagregowane utworzone przy zastosowaniu symboli wieloznacznych (ang. *wildcard*), przepływy obsługiwane według ustalonych zasad zapisanych we wpisie zagregowanym nazywa się agregatem ruchu. Sterownik sieci SDN stosuje mechanizmy reaktywne do obsługi pakietów z nowych przepływów. Zbliżone rozwiązanie ujawniono także w patencie europejskim nr EP3175582B1 udzielonym na wynalazek pt. AUTOMATED FLOW DEVOLVEMENT IN AN AGGREGATE FLOW ENVIRONMENT. W artykule TROIS CELIO i in. pt "A Survey on SDN Programming Languages: Toward a Taxonomy", IEEE COMMUNICATIONS SURVEYS & TUTORIALS, vol. 18, no. 4 , pages 2687-2712, XP011634925, DOI:10.1109/COMST.2016.2553778 ujawniono architekturę SDN z kontrolerem SDN zarządzającym w sposób hybrydowy aktywny reaktywny przełączaniem tablic przepływów przez dodawania, zmienia lub usuwanie wpisów, w tym z wykorzystaniem pola określającego priorytet oraz symboli wieloznacznych w agregacji. Kontroler ten pozyskuje informacje statystyczne dotyczące ruchu w sieci zapytaniami w oknach

czasowych z minutową rozdzielczością. Patent Europejski nr EP 3 175 582 ujawnia przyporządkowywanie wpisom odpowiadającym mikroprzepływom wyższych priorytetów niż wpisom zagregowanym.

[0008] Wadą rozwiązań ze stanu techniki jest to, że pierwsze pakiety w nowym przepływie pojawiającym się w przełączniku są obarczone dodatkowym opóźnieniem w związku z koniecznością ustalenia nowych wpisów odpowiadających nowemu przepływowi. W efekcie powstaje istotne opóźnienie w obsłudze pierwszych pakietów nowego przepływu, co przykładowo może niekorzystnie wpływać na parametry zestawianych sesji TCP lub straty pakietów w przypadku transmisji UDP.

[0009] Celem wynalazku jest rozwiązanie ww. problemów z rozwiązaniami znanymi w stanie techniki i przyspieszenie obsługi nowych pakietów w sieci SDN.

[0010] Sposób obsługi przepływu pakietów w sieci komputerowej SDN zawierającej serwer ze sterownikiem SDN oraz przełącznik SDN z tabelą przepływów zawierającą wpisy szczegółowe określające parametry potrzebne do przyporządkowania pakietu do przepływu danych, a także pole określające priorytet, przy czym wpisy szczegółowe zawierają się w odpowiadających im wpisach zagregowanych które są przygotowywane z wyprzedzeniem dla określenia polityki routingu w zakresie obsługi przepływów w oknie czasowym o określonej długości, zgodnie z wynalazkiem cechuje się tym, że długość okna czasowego wynosi przynajmniej dwie godziny, zaś sterownik SDN z przełącznika SDN uzyskuje dane dotyczące statystyk przepływów dla wpisów zainstalowanych w tablicach przypływów. Przychodzący pakiet należący do nowego przepływu wysyła się do sterownika obsługując go jednocześnie zgodnie z odpowiadającym mu wpisem zagregowanym, zaś sterownik za pomocą mechanizmu reaktywnego definiuje nowy wpis szczegółowy odpowiadający temu przepływowi. Ponadto sterownik proaktywnie wprowadza zmiany do wpisów zagregowanych przewidzianych do zastosowania w określaniu polityki routingu w następnym oknie czasowym stosując optymalizację statyczną

funkcji celu z wykorzystaniem przynajmniej jednego kryterium wybranego z grupy obejmującej zużycie energii, suma opóźnień na wszystkich łączach w sieci, maksymalne opóźnienie spośród wszystkich łączy w sieci, sumaryczne wykorzystanie zasobów na wszystkich łączach sieci, maksymalne wykorzystanie zasobów spośród wszystkich łączy w sieci, na podstawie danych dotyczących statystyk przepływów, przy czym wpisom zagregowanym przyporządkowuje się niższe priorytety niż wpisom szczegółowym. Takie rozwiązanie pozwala na eliminacje opóźnienia pierwszych pakietów w nowym przepływie dzięki temu, że jednocześnie się je obsługuje i definiuje dla nich szczegółowe wpisy. Jednocześnie ogranicza się rozmiar pamięci potrzebnej dla przechowywania wpisów szczegółowych.

[0011] Korzystnie przełącznik SDN zawiera pierwszą tabelę przepływów zawierającą wpisy zagregowane oraz przynajmniej jedną pomocniczą tabelę przepływów zawierającą wpisy szczegółowe. Zastosowanie wielu tabel ułatwia organizację wpisów i pozwala na zastosowanie różnych rodzajów pamięci dla wpisów zagregowanych i różnych dla szczegółowych. Jest to korzystne, ponieważ częstotliwość zapisu bardzo się różni w przypadku tych wpisów.

[0012] Korzystnie wpisy szczegółowe odpowiadające temu danemu wpisowi zagregowanemu w tabeli przepływów znajdują się w odpowiadającej temu wpisowi zagregowanemu osobnej tabeli pomocniczej przy czym ta tabela pomocnicza zawiera ponadto wpis domyślny, dzięki temu uzyskuje się możliwość przyspieszenia przeszukiwania tabel przy przyporządkowywaniu pakietów do przepływu.

[0013] Korzystnie w pamięci serwera zapisuje się historyczne dane dotyczące statystyk przepływów w kolejnych oknach czasowych a długość okna czasowego poddaje się optymalizacji w osobnym procesie, z wykorzystaniem tych historycznych danych. Takie rozwiązanie pozwala na dostosowanie długości okna do specyfiki sieci, która może się zmieniać sezonowo.

[0014] Optymalizację statyczną tablicy przepływów dla okna o określonej długości korzystnie prowadzi się w najbliższym poprzedzającym je oknie o tej samej długości. Dzięki temu ogranicza się ryzyko, że czas trwania krótkiego okna poprzedzającego długie okno okaże się niewystarczający dla przeprowadzenia optymalizacji.

[0015] Programowalna sieć komputerowa czyli sieć SDN (SDN – ang. *Software Defined Network*) zawierająca serwer ze sterownikiem oraz przynajmniej jeden przełącznik SDN z tabelą przepływów zgodnie z wynalazkiem ma sterownik przystosowany do wykonywania sposobu według wynalazku.

[0016] Korzystnie tabela przepływów w przełączniku jest zapisana w pamięci TCAM (ang. *ternary content-addressable memory*) zapewniającej szybki dostęp.

[0017] Produkt programu komputerowego przystosowany do pełnienia funkcji sterownika sieci SDN zawiera zestaw instrukcji, które po załadowaniu do pamięci serwera powodują wykonywanie sposobu według wynalazku.

[0018] Przedmiot wynalazku objaśniono w przykładach realizacji na rysunku, na którym Fig. 1 przedstawia schematycznie prostą sieć SDN, Fig. 2 przedstawia schematycznie sposób obsługi pakietu w pierwszym przykładzie realizacji sposobu według wynalazku, Fig. 3 przedstawia strukturę programu realizującego sposób wynalazku, natomiast Fig. 4 przedstawia schematycznie sposób obsługi pakietu w alternatywnym przykładzie realizacji sposobu według wynalazku.

[0019] Podstawą architektury sieci SDN jest oddzielenie płaszczyzny sterowania 01 od płaszczyzny danych 02 oraz przeniesienie wszystkich mechanizmów związanych z płaszczyzną sterowania 01 do centralnego punktu sieci - sterownika 11a uruchomionego na serwerze 11. Prostą sieć zaprojektowaną zgodnie z tą doktryną ukazano na Fig. 1. Zawiera ona sterownik SDN 11 oraz przełącznik SDN 12 z tabelą przepływów 12a konfigurowaną przez sterownik SDN 11a. Przełącznik SDN 12

komunikuje się z innymi przełącznikami 13, 14 w sieci. Komunikacja pomiędzy sterownikiem a przełącznikami odbywa się zgodnie z protokołem OpenFlow, który został wykorzystany do ilustracji przykładów realizacji przedmiotowego wynalazku. Ściślej przykłady realizacji zostały przygotowane w wersji OpenFlow 1.3.

[0020] W płaszczyźnie sterowania są podejmowane decyzje związane z obsługą ruchu i dostarczane do węzłów sieciowych, takich jak przełącznik 12, których rola zredukowana zostaje do wykonywania instrukcji otrzymanych od sterownika SDN. Istotnym elementem płaszczyzny sterowania są algorytmy określające polityki obsługi przetwarzanych przepływów oraz interfejsy komunikacyjne służące wymianie danych z urządzeniami sieciowymi i zewnętrznymi aplikacjami oraz źródłami informacji. Na płaszczyznę danych składają się elementy urządzenia sieciowego odpowiedzialne za transmisję danych takie jak interfejsy sieciowe, kolejki, bufory, a także mechanizmy analizy i modyfikacji nagłówków pakietów.

[0021] Komunikacja pomiędzy przełącznikiem SDN 12, a sterownikiem SDN 11a wymaga zestawienia sesji OpenFlow. Domyślnie pierwszym krokiem jest nawiązanie połączenia TCP, wiadomości mogą być dodatkowo chronione przez zastosowanie protokołu TLS. Następnie dochodzi do wymiany komunikatów OFPT_HELLO, które pomagają obu stronom ustalić wykorzystywaną wersję OpenFlow oraz zweryfikować poprawność kanału transmisyjnego. Zakończenie negocjacji pozwala na swobodną wymianę wiadomości innego typu. Uzyskanie przez sterownik dodatkowych informacji na temat wspieranych przez przełącznik elementów standardu możliwe jest dzięki wymianie wiadomości OFPT_FEATURES_REQUEST i OFPT_FEATURES_REPLY. W celu ułatwienia współpracy każdego z urządzeń ze sterownikiem, przełącznik posiada unikalny parametr Datapath ID, który umożliwia jego jednoznaczną identyfikację.

[0022] Rolą przełącznika SDN 12 jest przechowywanie w tabeli przepływów 12a wpisów zainstalowanych przez sterownik oraz zastosowanie właściwych akcji w przypadku odebrania pakietu odpowiadającego jednemu z wpisów. Tabela przepływów 12a składa się z wpisów zawierających m.in. priorytet, dane statystyczne oraz listę akcji, np. operacji modyfikacji pola nagłówka i przekazania pakietu do portu wyjściowego. Kluczowym elementem wpisu jest dopasowanie (ang. *match*) określające parametry pakietu należącego do definiowanego przepływu. Są to zwykle wartości wybranych pól nagłówków, np. adresy docelowe MAC lub wartość pola ToS pakietu IP. W momencie odebrania pakietu, w tabeli przepływów wyszukiwany jest pasujący wpis o najwyższym priorytecie, którego akcje zostają zastosowane do pakietu. W niektórych przełącznikach pakiet może zostać przekazany do przetwarzania w kolejnej tabeli zawierającej inny zestaw wpisów. W przypadku braku dopasowania, zastosowana zostaje domyślna akcja, którą najczęściej jest wysłanie pakietu nowego przepływu do sterownika. Uproszczony schemat przedstawiający obsługę wielu przepływów, przeszukiwanie tabel przełącznika i zastosowanie akcji kierującej pakiet do wyjściowego portu umieszczony został na Fig. 2.

[0023] Sterownik SDN 11a instaluje w tabeli przepływów 12a przełącznika 12 wpisy definiujące docelowe akcje dla przetwarzanych przepływów. Modyfikacja tabeli przepływów odbywa się przez wysłanie wiadomości typu OFPT_FLOW_MOD. Proces ten w niniejszym przykładzie wykonania jest inicjowany przez sam sterownik w przypadku wpisów zagregowanych - proaktywna instalacja wpisów wyprzedzająca pojawienie się przepływów lub w odpowiedzi na wiadomość OFPT_PACKET_IN wysyłaną przez przełącznik w wypadku odebrania pakietu nie pasującego do żadnego z aktualnie przechowywanych wpisów - reaktywna instalacja wpisów po pojawieniu się konkretnego przepływu. Sterownik może również komunikatem OFPT_PACKET_OUT

zażądać wysłania przez przełącznik pakietu o dowolnej zawartości z wybranego portu wyjściowego.

[0024] Protokół OpenFlow pozwala na jednoczesne definiowanie i wykorzystywanie wpisów o różnej zawartości pola MATCH określającego charakterystykę pasującego przepływu. W przypadku pakietu odpowiadającego dopasowaniom obecnym w kilku różnych wpisach o wyniku klasyfikacji decydują ich priorytety. Ten mechanizm umożliwia tworzenie wpisów agregujących wiele przepływów, a jednocześnie, przy zastosowaniu właściwie dobranych wartości priorytetów oraz opcjonalnym rozproszeniu wpisów pomiędzy tabelami, na zaprojektowanie wielopoziomowego procesu przetwarzania pakietów.

[0025] Instrukcja powodująca przesłanie pakietu do sterownika, zgodnie z założeniami jest wykorzystywana w sytuacji braku wpisów odpowiadających odebranemu pakietowi i konieczna jest dalsza analiza w celu podjęcia decyzji o sposobie obsługi nowego przepływu. W niniejszym przykładzie wykonania taka akcja jest umieszczana jest we wpisie typu TABLE MISS posiadającym najniższy priorytet w tabeli, ale może być też przypisana do innych wpisów. Odpowiednie wykorzystanie tych funkcji protokołu OpenFlow pozwala na efektywne zarządzanie ruchem w sieci, zapewnienie niezawodności usług oraz możliwość reagowania na zmiany zapotrzebowań użytkowników.

[0026] Możliwe jest agregowanie przynajmniej części istniejących przepływów stosując np. wspólną trasę dla strumieni o jednakowym docelowym adresie IP. Zgodnie z wynalazkiem takie wpisy przygotowuje się z wyprzedzeniem przed rozpoczęciem obsługi przepływów w oknie czasowym o predefiniowanej długości – w niniejszym przykładzie 4 godziny. Twórcy wynalazku zauważyli, że okna czasowe o długości mieszczącej się między 3 a 8 godzin, z reguły około połowy dnia pracy pozwalają przy długim czasie obserwacji ruchu na skuteczną optymalizację przygotowywanych na kolejne okna czasowe wpisów zagregowanych. Optymalizacja ta zapewnia

proaktywny mechanizm przygotowywania wpisów zagregowanych. Wpisy agregowane przygotowywane są w oparciu o mechanizmy i algorytmy optymalizacji. Te algorytmy optymalizacji działają cyklicznie, tzn. w czasie trwania jednego okna czasowego wyznaczają zalecany sposoby obsługi przepływów w nadchodzących oknach czasowych. To znaczy, że w rozważanej sieci możliwe są okresowe zmiany polityki ruchu dotyczące nowych przepływów pasujących do znanych wpisów zagregowanych. Taka optymalizacja odbywa się na podstawie aktualnego stanu sieci, przewidywanego stanu sieci w nadchodzącym oknie czasowym stanowiącego wynik procesu predykcji, oraz informacji pochodzących z aplikacji korzystających z sieci np. dotyczących spodziewanego wydarzenia masowego, które może zaburzyć normalne funkcjonowanie sieci. Jednocześnie, wskazane jest utrzymanie sposobu obsługi dla już istniejących przepływów. Dotyczy to zarówno momentu okresowej zmiany polityk routingu jak i ewentualnej modyfikacji polityk dla przepływów już obsługiwanych w sieci zgodnie z wpisami szczegółowymi. Pożądana jest również minimalizacja udziału sterownika w podejmowaniu decyzji dotyczących akcji dla przepływów w celu uniknięcia przeciążenia ruchem sterującym oraz eliminacji opóźnienia związanego z obsługą pierwszych pakietów nowego przepływu.

[0027] Nowo-pojawiające się przepływy niepasujące do zoptymalizowanych wpisów zagregowanych obsługuje się reaktywnie tworząc wpisy szczegółowe – mające wyższy priorytet.

[0028] W dalszym opisie zastosowano następujące oznaczenia

P1 - pierwszy pakiet należący do przepływu;

P2, P3, ... - kolejne pakiety należące do przepływu;

A, B, ... - wpis zagregowany w tabeli przepływów odpowiadający zagregowanej grupie przepływów;

A.1, A.2, ... - szczegółowe wpisy w tabeli przepływów pasujące do wpisu zagregowanego A.

[0029] Rozważmy czysto reaktywną obsługę przepływów. W takiej obsłudze dla każdego zaobserwowanego przepływu byłby tworzony nowy wpis, a wpisy byłyby w pojedynczej tabeli. Przychodzący pakiet P1 należący do nowego przepływu nie zostaje dopasowany do żadnego z istniejących wpisów o wyższym priorytecie i zgodnie z akcją zapisaną we wpisie TABLE MISS zostaje przekazany do sterownika w wiadomości OFPT_PACKET_IN. Sterownik podejmuje decyzję o akcji odpowiedniej dla nowego przepływu i przy użyciu komunikatu OFPT_FLOW_MOD dodaje nowy wpis do tabeli przepływów. Pakiety P2, P3 oraz kolejne zostają dopasowane do nowo utworzonego wpisu i obsłużone zgodnie z przypisaną akcją bez udziału sterownika. Tego typu reaktywne podejście do instalacji wpisów gwarantuje zarówno łatwą zmianę polityk dla przychodzących przepływów (akcja definiowana przez sterownik na bieżąco podczas analizy nowego przepływu) oraz zachowanie polityk dla już zdefiniowanych przepływów (akcje zapisane na stałe we wpisach do czasu ich wygaśnięcia). Wymagana jest jednak komunikacja ze sterownikiem przed utworzeniem wpisów dla każdego z zaobserwowanych przepływów, co może powodować znaczne opóźnienie zestawiania transmisji, szczególnie w przypadku dużego obciążenia sterownika.

[0030] Rozważmy czysto proaktywną obsługę przepływów. W tabeli przepływów utrzymuje się jedynie wpisy tworzone proaktywnie i określające akcje dla większych wpisów zagregowanych. W tym przypadku odebrany przez przełącznik pakiet P1 należący do nowego przepływu zostaje na podstawie pól nagłówków dopasowany do wpisu zagregowanego A określającego wspólną akcję i natychmiastowo obsłużony zgodnie z zapisanymi instrukcjami. W ten sam sposób obsłużone zostają pakiety P2, P3 i kolejne należące do tego samego przepływu. Zastosowanie czysto proaktywnej obsługi pozwala na zminimalizowanie opóźnienia związanego z wymianą wiadomości OpenFlow przed zestawieniem każdego przepływu. Umieszczanie agregowanych wpisów w tabeli przepływów pozwala również na znaczne ograniczenie rozmiaru

tabeli, co w niektórych sytuacjach ułatwia zarządzanie i monitorowanie ruchu. Jednak w przypadku zmiany polityki obsługi dla wybranego wpisu zagregowanego, nie jest możliwe zachowanie akcji stosowanych do już istniejących strumieni danych, co może prowadzić do chwilowych zakłóceń transmisji. Całkowite pominięcie sterownika w obsłudze nowych przepływów wymaga również proaktywnego instalowania w tabeli przepływów wpisów obsługujących wszystkie przewidywane agregaty ruchu, co nie zawsze jest realizowalne i wiąże się z ryzykiem odmowy obsługi jeżeli przyjmie się ograniczenia pamięci lub znacznym rozrostem tablic przepływów w przeciwnym wypadku.

[0031] Zgodnie z wynalazkiem łączy się podejście proaktywne i reaktywne w sposób pozwalający zaoszczędzić pamięć, zredukować czas przeszukiwania pamięci oraz przyspieszyć obsługę nowych pakietów. W celu eliminacji opóźnienia związanego z zestawianiem transmisji, stosuje się zarówno wpisy agregujące przepływy przygotowywane w sposób proaktywny, jak i wpisy dopasowujące przepływy szczegółowe przygotowywane reaktywnie. Rozmieszczenie wpisów oraz etapy przetwarzania odbieranych pakietów przedstawione zostały na Fig. 2. W pojedynczej tabeli przepływów 12a wszystkie wpisy A, A.1, A.2, ... są rozmieszczone w grupach odpowiadających kolejnym wpisom zagregowanym w kolejności ustalonej za pomocą priorytetów. W każdej z grup znajdują się posiadające wyższy priorytet szczegółowe wpisy A.1, A.2, ... dotyczące szczegółowych przepływów, poprzedzające domyślny dla całego agregatu ruchu wpis zagregowany A. Wpis zagregowany A oprócz podstawowych akcji zdefiniowanych przez sterownik 11a dla całego agregatu ruchu, jak np. przekazanie pakietu na port wyjściowy określony w sposób proaktywny i w efekcie działania algorytmów optymalizacji statycznej zawiera dodatkową akcję przekazującą pakiet do portu OFPP_CONTROLLER powodującą wysłanie dopasowanego pakietu do sterownika 11a.

[0032] W momencie odebrania, należący do nowego przepływu pakiet P1 jest porównywany kolejno z wpisami A.1, A.2, ... w kolejności priorytetów – począwszy od najwyższego. W związku z brakiem dopasowania dokonywane jest kolejne, zakończone powodzeniem, porównanie z wpisem A. Pakiet zostaje natychmiastowo obsłużony zgodnie z zestawem akcji przypisanych do wpisu. Jednocześnie stosowane są instrukcja powodująca przekazanie pakietu w wiadomości OFPT_PACKET_IN do sterownika 11 oraz instrukcje domyślne dla agregatu, takie jak wysłanie pakietu do jednego z portów wyjściowych, który to port został wyznaczony na podstawie mechanizmów optymalizacji statycznej biorącej pod uwagę aktualne stan sieci, predykcje dotyczące przyszłości oraz informacje pochodzące od aplikacji sieciowych. Sterownik 11a podejmuje decyzję dotyczącą sposobu obsługi nowego przepływu w sposób reaktywny na podstawie mechanizmów optymalizacji dynamicznej. Za pomocą wiadomości OFPT_FLOW_MOD sterownik dodaje do tabeli przepływów w odpowiednim miejscu wpis szczegółowy A.3. P2, P3 oraz kolejne pakiety przepływu przetwarzane w analogiczny sposób zostają dopasowane do nowego wpisu A.3 i obsłużone bez dalszego udziału sterownika. Akcje przypisane do nowego wpisu mogą wskazywać dowolny port wyjściowy wyznaczony reaktywnie na podstawie optymalizacji dynamicznej. W ten sposób możliwa jest modyfikacja domyślnego sposobu obsługi nowego przepływu bez ingerencji w przepływy już obsługiwane.

[0033] Dzięki rozszerzeniu listy akcji we wpisie dla agregowanego przepływu, opóźnienie obsługi pierwszych pakietów zostaje zmniejszone dzięki eliminacji opóźnienia powodowanego przez komunikację ze sterownikiem 11a. W przypadku zmiany polityki obsługi dla nowych przepływów należących do agregatu, wymagana jest jedynie modyfikacja wpisu A. Istniejące już strumienie danych obsługiwane przez wpisy A.1, A.2, ..., które są bardziej szczegółowe względem wpisu A i posiadają wyższy priorytet. W efekcie zachowują dotychczasowe akcje, co

zapobiega zmianom w sposobie obsługi i zakłóceniom ruchu. Do realizacji niniejszego sposobu w sterowniku 11a zaimplementowano mechanizmy przydzielające odpowiednie priorytety wpisom za każdym razem gdy dodawany jest nowy wpis szczegółowy, tak, że podział wpisów na grupy jest zachowany.

[0034] Implementacja sposobu stanowi rozszerzenie sterownika Ryu zastosowanego jako sterownik SDN 11a.Sterownik Ryu został zaimplementowany w języku Python. Implementacja rozszerzenia wiąże się ze stworzeniem w osobnym module klasy, nazywanej aplikacją sterownika. Klasa ta, będąca rozszerzeniem klasy ryu.base.app_manager.RyuApp jest odpowiedzialna za wstępną konfigurację systemu oraz obsługę zdarzeń sieciowych:

```
class CustomApp(app_manager.RyuApp):
OFP_VERSIONS = [ofproto_v1_3.OFP_VERSION]
def __init__(self, *args, **kwargs):
super(CustomApp, self).__init__(*args, **kwargs)
```

[0035] Kod aplikacji zarządzającej rozmieszczeniem wpisów w tabelach przepływów składa się z elementów odpowiadających stosowanym funkcjom protokołu OpenFlow i przełącznika oraz interfejsowi komunikacyjnemu. Zaimplementowano klasy odpowiadające konfigurowanym agregatom ruchu i zestawowi wpisów obecnych w tabeli przełącznika. Strukturę aplikacji przedstawiono na Fig. 3. Najważniejszymi komponentami aplikacji są:

- FlowDistributionController - główna klasa aplikacji, odpowiedzialna za obsługę zdarzeń sieciowych oraz komunikację z przełącznikiem;
- FlowDistributionControllerApi - klasa implementująca interfejs REST API służący do konfiguracji aplikacji;
- FlowAggregate - klasa reprezentująca zdefiniowany przez użytkownika agregat ruchu wraz z jego dopasowaniem i akcjami;
- FlowPipeline - klasa reprezentująca zestaw reguł obecnych w tabeli przełącznika.

[0036] W aplikacji sterownika Ryu zdarzenia obsługiwane są przez dedykowane do tego metody. Są one opakowane odpowiednimi dekoratorami zdefiniowanymi w modułach ryu.topology.event oraz ryu.controller.ofp_event. Implementacja takiej metody pozwala na stworzenie dowolnych mechanizmów postępowania w przypadku wystąpienia w sieci określonych zdarzeń. Przyjmują one jako argumenty obiekty przechowujące dane związane ze zdarzeniem, np. pola nagłówków odebranego pakietu lub statystyki przepływu. Zaimplementowano obsługę dwóch zdarzeń:

- @set_ev_cls(ryu.controller.ofp_event.EventOFPPacketIn, MAIN_DISPATCHER) - obsługa zdarzenia PACKET IN wywołanego przekazaniem pakietu przez przełącznik do sterownika.
- @set_ev_cls(ryu.controller.ofp_event.EventOFPSwitchFeatures, CONFIG_DISPATCHER) - obsługa odebranej wiadomości SWITCH FEATURES zawierającej informacje nt. parametrów przełącznika, końcowy etap nawiązywania łączności pomiędzy przełącznikiem i sterownikiem.

[0037] Modyfikacja zawartości tabeli przepływów 12a to jedna z podstawowych operacji wykonywanych przez aplikację. Wymaga to wysłania wiadomości protokołu OpenFlow typu OFPT_FLOW_MOD do przełącznika. Może ona zawierać parametry takie jak m.in. priorytet wpisu, dopasowanie przepływu i akcje a także typ operacji np. wstawianie lub usuwanie. Wygenerowanie wiadomości OFPT_FLOW_MOD w przełączniku Ryu polega na stworzeniu obiektów klas OFPInstructionActions i OFPMatch odpowiadających odpowiednio żądanym instrukcjom dla danego przepływu oraz jego dopasowaniu. Są one następnie używane podczas tworzenia obiektu klasy OFPFlowMod, który reprezentuje gotową do wysłania wiadomość. W celu ułatwienia procesu zaimplementowano w aplikacji metodę add_flow odpowiedzialną za stworzenie prawidłowej wiadomości i wysłanie jej do przełącznika.

[0038] W aplikacji stosowane jest pojęcie agregatu oznaczające zbiór przepływów definiowany przy użyciu wspólnego parametru,

np. tego samego docelowego adresu IP. Do jednego agregatu można dopasować kilka przepływów, które przykładowo charakteryzują się wspólnym adresem docelowym IP, ale różnymi docelowymi numerami portów TCP. Rolą agregatu w zaimplementowanej aplikacji oraz opracowanych algorytmach jest opisanie sposobu obsługi wielu przepływów przy użyciu zaledwie jednego wpisu. W utworzonej aplikacji agregat ruchu reprezentowany jest przez klasę FlowAggregate. Posiada ona atrybuty określające dopasowanie, akcje i unikalny identyfikator stosowany podczas rozmieszczania wpisów w tabeli. Zarówno akcje, jak i dopasowanie zapisywane są w sposób zgodny z formatem używanym przez moduł ryu.app.ofctl_rest:

```
{
        "ipv4_src": "192.168.0.1",
        "eth_type": 0x0800
}
[
        {
                "type":"OUTPUT",
                "port": 1
        }
]
```

co pozwala wykorzystać zaimplementowane w sterowniku 11a mechanizmy przetwarzania oraz weryfikacji danych wejściowych od użytkownika.

[0039] Aktywny tryb pracy aplikacji w momencie zdefiniowania przez użytkownika nowego agregatu może wymagać dodania do tabeli przepływów 12a przełącznika wpisu utworzonego na podstawie przekazanych parametrów. W tym celu wywołana zostaje, opisana uprzednio, metoda add_flow. Zgodnie z wymogami specyfikacji OpenFlow wykorzystanie w dopasowaniu niektórych pól wymaga wcześniejszego zdefiniowania wartości innych pól. Najczęstszym przypadkiem jest konieczność definiowania wartości pola EtherType protokołu Ethernet przy

korzystaniu z pól warstwy trzeciej oraz wartości pola Protocol protokołu IPv4 podczas korzystania z pól warstwy czwartej.

[0040] Zbiór zdefiniowanych przez użytkownika agregatów ruchu reprezentowany jest klasę FlowPipeline. Jej zadaniem jest utrzymywanie listy aktywnych agregatów wraz z odpowiadającymi im identyfikatorami. W ramach klasy zaimplementowano metody przeznaczone do wyszukiwania na podstawie identyfikatora lub dopasowywanych pól nagłówków. Pozwala to na dopasowanie pakietu odebranego przez sterownik do właściwego agregatu.

[0041] Wszystkie wpisy są przechowywane jednej tabeli przełącznika. Podczas definiowania nowego agregatu w tabeli przepływów zostaje dodany, w sposób proaktywny, domyślny wpis agregatu. Port wyjściowy wskazany w akcji dla tworzonego agregatu zostaje wyznaczony na skutek działania mechanizmów optymalizacji statycznej z wyprzedzeniem względem okien czasowych, dla których polityki routingu są przygotowywane. Tworzony wpis posiada parametry „cookie” i „priority” generowane na podstawie unikalnego identyfikatora agregatu:

priority = aggregate_id * 100

cookie = aggregate_id * 100

[0042] Pakiety analizowane w metodzie obsługującej zdarzenia OFPT_PACKET_IN przypisywane są do agregatu na podstawie wartości parametru cookie, który pozwala jednoznacznie określić poszukiwany identyfikator agregatu. Na podstawie znalezionego dopasowania oraz listy skonfigurowanych akcji do tabeli przepływów zostaje, w sposób reaktywny, dodany szczegółowy wpis dla nowego przepływu. Port wyjściowy wskazany w akcji dla tworzonego wpisu szczegółowego zostaje wyznaczony na skutek działania mechanizmów optymalizacji dynamicznej. Możliwe jest także przechowanie w sterowniku więcej niż jednego pakietu i podjęcie decyzji o porcie wyjściowym na podstawie wielkości i rozkładu tych pakietów. Parametry cookie oraz priority tego wpisu szczegółowego zwiększane są o 10 w stosunku do wartości obecnych w domyślnym wpisie agregatu. W ten sposób,

z wykorzystaniem priorytetów, osiągnięte zostaje rozmieszczenie wpisów w grupach odpowiadających przynależności do poszczególnych agregatów.

[0043] Dodatkowym usprawnieniem obecnym w przełącznikach sprzętowych jest możliwość przechowywania wpisów w pamięci TCAM, która pozwala na niezwykle szybką klasyfikację pakietów. Najczęściej jest ona stosowana do obsługi pierwszej z dostępnych tabel, w której można zapisać od kilkuset do kilkunastu tysięcy wpisów. Dostępne na rynku moduły pamięci TCAM są w stanie przeprowadzić ponad milion wyszukiwań w ciągu sekundy, a efektywny czas klasyfikacji pakietu osiąga wartości rzędu nanosekund lub poniżej nanosekundy, co czyni go nieistotnym w porównaniu z czasem działania pozostałych mechanizmów, do których zaliczają się m.in. kolejkowanie, analiza nagłówków. Ze względu na duży przyrost wydajności przy wykorzystaniu tabeli przechowywanej w pamięci TCAM, a zarazem jej ograniczoną pojemność, istotną kwestią staje się wybór pomiędzy umieszczeniem wpisu w TCAM lub mniej wydajnej pamięci innego typu. Rozmiar pamięci TCAM jest ograniczony, a wraz z jego wzrostem rośnie ryzyko awarii. Z tego względu oszczędność pamięci związana z zastosowaniem sposobu według wynalazku przekłada się dodatkowo na wzrost szybkości działania sieci, gdyż mniejsze tabele przepływów łatwiej mieszczą się w pamięci.

[0044] Przeszukiwanie wpisów można przyspieszyć dzieląc tabelę przepływów na większą liczbę tabel 12b, 12c, jak pokazano na Fig. 4. Każdy pakiet obsługiwany jest w co najmniej dwóch tabelach. Rozmieszczenie wpisów oraz etapy przetwarzania odbieranych pakietów zostały schematycznie przedstawione na Fig. 4. W chwili odebrania, pakiet P1 jest porównywany z wpisami agregatów A, B, ... znajdującymi się w domyślnej tabeli 12b przełącznika. Każdy z tych wpisów zawiera instrukcję OFPIT_GOTO_TABLE powodującą przeniesienie pakietu do przetwarzania w tabeli 12c zawierającej wpisy szczegółowe mieszczące się w danym wpisie zagregowanym zgodnie ze sposobem

według wynalazku. Jest to tabela pomocnicza dla wpisu agregowanego. W wyniku dopasowania do agregatu A pakiet zostaje porównany z wpisami A.1, A.2,... znajdującymi się w tabeli pomocniczej 12c dla danego wpisu agregowanego. Z powodu braku pasujących wpisów, pakiet P1 zostaje obsłużony zgodnie z domyślną akcją określoną we wpisie o najniższym priorytecie w tabeli pomocniczej 12c dla danego agregatu tj. TABLE MISS – w tym przykładzie zostaje on wysłany do portu wyjściowego, który został wyznaczony w sposób proaktywny na skutek działania algorytmów optymalizacji statycznej, a jednocześnie przekazany do sterownika 11a. Stosowana jest więc metoda jednoczesnej obsługi pakietu oraz przekazania go do warstwy sterującej. Sterownik 11a dobiera akcję dla nowego przepływu i dodaje szczegółowy wpis A.3 w odpowiedniej tabeli w wyniku działania reaktywnego bazującego na algorytmach optymalizacji dynamicznej. Pakiety P2, P3 oraz kolejne przetwarzane są w analogiczny sposób i dopasowane do nowego wpisu A.3. Powiązana z nimi akcja zostaje natychmiast zastosowana do pakietu bez potrzeby dodatkowej komunikacji ze sterownikiem. Akcje przypisane do nowego wpisu mogą wskazywać dowolny port wyjściowy wyznaczony reaktywnie na podstawie optymalizacji dynamicznej. W ten sposób możliwa jest modyfikacja domyślnego sposobu obsługi nowego przepływu bez ingerencji we wpisy już obsługiwane. Tabelę 12b zawierająca wpisy zagregowane można przy dopasowywaniu pakietu do wpisu traktować jako indeks do tabeli 12c zawierającej wpisy szczegółowe i tym samym przyspieszyć przeszukiwanie. Podział wpisów na grupy umieszczone w osobnych tabelach dodatkowo pomaga w zarządzaniu nimi.

[0045] W implementacji tego przykładu wykonania sposobu według wynalazku na potrzeby dodawania agregatu generuje się unikalny identyfikator, będący jednocześnie numerem tabeli 12c zawierającej szczegółowe wpisy dotyczące przepływów z danej grupy odpowiadającej wpisowi zagregowanemu. W tabeli 12b

przełącznika dodany zostaje wpis kierujący wszystkie pakiety należące do danego agregatu do właściwej tabeli pomocniczej 12b. W niej umieszczony zostaje wpis typu TABLE MISS o najniższym priorytecie stosujący do pakietu domyślną akcję agregatu oraz jednocześnie przekazujący go do sterownika 11a. Domyślna akcja wpisu zagregowanego wyznaczana jest więc w sposób proaktywny na podstawie mechanizmów optymalizacji statycznej. Po odebraniu pakietu w metodzie obsługującej zdarzenia OFPT_PACKET_IN, z wiadomości odczytany zostaje numer tabeli, do której pierwotnie trafił pakiet. Na jego podstawie znajdowany jest właściwy dla pakietu agregat, a w tabeli o tym samym numerze wstawiony zostaje szczegółowy wpis dotyczący nowego przepływu. Port wyjściowy wskazany w akcji dla tworzonego wpisu szczegółowego zostaje wyznaczony na skutek działania mechanizmów optymalizacji dynamicznej. Możliwe jest także przechowanie w sterowniku 11a więcej niż jednego pakietu i podjęcie decyzji o porcie wyjściowym na podstawie wielkości i rozkładu tych pakietów.

[0046] Dużą zaletą niniejszego wynalazku jest możliwość modyfikacji sposobu obsługi przepływów bez ingerencji w przepływy już istniejące. Sposób według wynalazku pozwala na efektywne zarządzanie zawartością tabeli przepływów i eliminację opóźnienia obsługi pierwszych pakietów nowego przepływu związanego z oczekiwaniem na utworzenie nowego wpisu przez sterownik. Sieć SDN według wynalazku w typowych warunkach działa szybciej i bardziej niezawodnie niż sieci ze stanu techniki. Zdefiniowanie agregowanych przepływów o domyślnej akcji na podstawie algorytmów optymalizacji, wraz z jednoczesnym przekazywaniem nowych pakietów do sterownika 11a w celu reaktywnego zainstalowania szczegółowych wpisów na podstawie mechanizmów optymalizacji dynamicznej umożliwia czerpanie korzyści zarówno z proaktywnej jak i reaktywnej obsługi pakietów. Dzięki temu, pakiety zostają obsłużone w oczekiwany sposób z pominięciem oczekiwania na odpowiedź

sterownika 11a. Jest to możliwe dzięki zachowaniu cyklicznego działania, tzn. czas podzielony jest na przedziały – okna czasowe, a opracowywanie proaktywnych sposobów ruchu dla przyszłych okien czasowych odbywa się z wyprzedzeniem. Okresowe działanie umożliwia zastosowanie mechanizmów predykcji, do przywidywania obciążenia sieci w przyszłych oknach czasowych. Te przewidywania są następnie wykorzystywane przez algorytmy optymalizacji statycznej podczas przygotowywanie reguł proaktywnych. Dodatkowo jako dane wejściowe służy aktualne obciążenie sieci oraz informacje pochodzące z aplikacji zewnętrznych.

[0047] Informacje o obciążeniu sieci uzyskuje się wykorzystując mechanizmy wyspecyfikowane w protokole OpenFlow, tj. przez odpytywanie przełącznika o statystyki poszczególnych przepływów dla wpisów zainstalowanych w tablicach przypływów. Dodatkowo można wykorzystać narzędzie sFlow, w którym zewnętrzna względem sterownika SDN instancja monitora pobiera informacje o przepływach bazując na pakietach próbkowanych przez urządzenia sieciowe. Oba te mechanizmy pozyskiwania informacji o obciążeniu sieci realizuje się minimalizując związany z nimi natłok sygnalizacyjny.

[0048] Informacje o obciążeniu sieci są wykorzystywane przez predykcję i optymalizację – zarówno statyczną jak i dynamiczną. Predykcja przechowuje wyniki pomiarów w bazie danych i na ich podstawie szacuje obciążenie sieci w kolejnych oknach czasowych. Wyniki predykcji trafiają do optymalizacji statycznej, trafiają tam też informacje o aktualnym obciążeniu sieci. W ten sposób algorytm optymalizacji przygotowuje polityki routingu. Natomiast optymalizacja dynamiczna wykorzystuje informacje o aktualnym obciążeniu sieci podejmując decyzje o obsłudze nowych przepływów tj. akcji związanej z wpisami szczegółowymi. W razie detekcji natłoku w jakimś obszarze sieci optymalizacja dynamiczna może w związku

z tym zapewnić mechanizmy reakcji na sytuacje niepożądane kierując ruch przez inne obszary sieci.

[0049] W optymalizacji statycznej stosowanej w wynalazku potrzeba uwzględnienia kilku kluczowych wskaźników jakości (ang. *Key Performance Indicator – KPI*) implikuje sformułowanie wielokryterialnego problemu optymalizacyjnego. Dodatkowo, uwzględnienie zależności energetycznych wymusza zastosowania zmiennych binarnych. Oba te czynniki sprawiają, że problem jest trudny do rozwiązania metodami dokładnymi. Rozwiązania takich problemów są jednak znane w stanie techniki i można dopasować je do niniejszego zagadnienia.

[0050] Poniżej opisano podejście do problemu optymalizacji, które sprawdza się szczególnie dobrze i które dało rozwiązanie spełniające wymagania we wszystkich testowanych typach ruchu. Problem opisano za pomocą sformułowania typu węzeł-łącze (ang. *node-link*) – M. Pióro and D. Medhi, Routing, Flow, and Capacity Design in Communication and Computer Networks. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., July 2004. W funkcji celu zawarto pięć wskaźników i powiązano je z wagami po to by operator sieci SDN mógł uwzględniać własne priorytety i preferencje. Uwzględniane wskaźniki to: zużycie energii, suma opóźnień na wszystkich łączach w sieci, maksymalne opóźnienie spośród wszystkich łączy w sieci, sumaryczne wykorzystanie zasobów na wszystkich łączach sieci, maksymalne wykorzystanie zasobów spośród wszystkich łączy w sieci. Ważona suma tych wskaźników poddawana jest minimalizacji.

[0051] Sformułowanie problemu optymalizacyjnego obejmuje opisane poniżej ograniczenia. Podstawowe ograniczenie dla sformułowania problemu typu węzeł-łącze, wymusza żeby dla konkretnego żądania różnica pomiędzy ruchem wpływającym i wypływającym z każdego węzła była równa zero dla węzłów pośrednich, a dla węzłów końcowych żeby odpowiadała wielkości tego żądania, z dokładnością do znaku. Kolejne ograniczenie wymusza obciążenia na poszczególnych łączach sieci pozwalające

na realizację wszystkich żądań, wartości obciążeń wszystkich łączy są później sumowane w celu znalezienia wartości łącznej. Procentowe wykorzystanie łączy wyliczane jest jako ułamek użytej przepływności w stosunku do całkowitej dostępnej przepływności łącza, wyznaczana jest także maksymalna wartość procentowego wykorzystania łącza spośród wszystkich łączy sieci. Kolejne ograniczenie związane jest z wyliczaniem parametrów aproksymacji liniowej funkcji opóźnienia pakietów w postaci rosnącej wypukłej funkcji obciążenia łącza. Wykorzystano tutaj przybliżenie liniowe segmentami tej funkcji bazujące na modelu kolejkowym M/M/1. Dwa następne ograniczenia dotyczą sumarycznego i maksymalnego opóźnienia, wartości te wyliczane są z wykorzystaniem wcześniejszej funkcji. Kolejne ograniczenie wymusza uruchomienie łącza w przypadku gdy przesyłany jest przez nie jakikolwiek ruch, natomiast ostatnie ograniczenie zapewnia wyliczenie całkowitego kosztu związanego z wykorzystaną energią elektryczną.

[0052] Do rozwiązywania wielokryterialnych problemów ze zmiennymi binarnymi zastosowano metodę dekompozycji Lagrange'a opisaną w publikacji J. Karkazis, "Facilities location in a competitive environment: A promethee based multiple criteria analysis," European Journal of Operational Research, vol. 42, no. 3, pp. 294 – 304, Oct. 1989. Opcjonalne usprawnienie obejmuje wykorzystanie sekwencji ergodycznych, znanych także pod nazwą: algorytmu wolumenu (ang. *volume algorithm*) opisaną w publikacjach:

- E. Gustavsson, M. Patriksson, and A.-B. Strömberg, "Primal convergence from dual subgradient methods for convex optimization," Mathematical Programming, vol. 150, no. 2, pp. 365–390, May 2015.

oraz

- F. Barahona and F. A. Chudak, "Near-optimal solutions to large-scale facility location problems," Discrete Optimization, vol. 2, no. 1, pp. 35 – 50, Mar. 2005.

[0053] Dodatkową poprawę działania wynalazku można uzyskać dostosowując i poddając optymalizacji statycznej sam czas trwania okien czasowych, w ramach których cyklicznie przeprowadza się optymalizację statyczną przygotowując polityki routingu dla kolejnego okna. Przykładowo w przypadku predykcji ruchu w sieci komputerowej w przedsiębiorstwie o ściśle określonych godzinach pracy 8-godzinne okno rozpoczynające się wraz z początkiem zmiany jest najwłaściwsze, natomiast w sytuacji obowiązywania elastycznych godzin pracy należy czas pracy podzielić na trzy 4-godzinne okna: pierwsze związane z przychodzeniem pracowników na stanowiska, drugie związane z czasem gdy większość pracowników przebywa w biurze i trzecie związane z opuszczaniem biura przez pracowników.

[0054] Inny rytm zmian przepływów sieci dotyczy nocnej aktywności w obszarach sieci przyłączającej budynki biurowe, centra danych, ale i osiedla mieszkalne. Długość okien czasowych powinna uwzględniać istnienie nocnych zmian, z którymi lepiej sprawdzają się 4-godzinne okna lub ich brak, który sprawia, że można i należy zastosować 8-godzinne okna. Również w przypadku centrów danych powinno nastąpić skrócenie okien czasowych z 8 do 4 godzin w przypadku aktywności użytkowników centrów danych i do 2 godzin w przypadku występowania cyklicznych prac utrzymaniowych skutkujących wzmożoną aktywnością sieciową.

[0055] W przypadku gospodarstw domowych, różne typy skupisk ludzkich będą cechować się większą synchronizacją czasową w kontekście przerywania działalności w Internecie. W rejonach niezurbanizowanych sprawdzają się dobrze 6-godzinne okna, zaś w rejonach wysoce zurbanizowanych, 4-godzinne okna.

[0056] Adaptacyjne dostosowywanie właściwej długości okien czasowych zgodnie z wynalazkiem prowadzi się na podstawie prowadzonych pomiarów sieciowych w sprzężeniu z predykcją ruchu sieciowego. Dzięki temu w sposób dynamiczny sieć SDN

dopasowuje się do aktywności użytkowników przy wykorzystaniu danych historycznych.

[0057] Takie podejście umożliwia wykrycie okresowych zależności, powtarzających się w określonych dniach tygodnia czy miesiąca. Możliwe jest wykorzystanie okien czasowych różnej długości wydzielonych w ramach doby.

[0058] W sytuacji gdy wydłużone okno występować będzie po krótkim może to doprowadzić do sytuacji, w której nie wystarczy czasu na przeprowadzenie optymalizacji. Należy więc podzielić okna na pary o możliwie zbliżonym czasie trwania i z uwzględnieniem by te pary następowały w możliwie najkrótszych odstępach czasu po sobie. Następnie dokonując w ramach tych par optymalizacji minimalizuje się ryzyko zaistnienia niedoboru czasu do przeprowadzenia procesu optymalizacji.

[0059] Należy podkreślić, że przykłady realizacji opisane powyżej stanowią ilustrację wynalazku zdefiniowanego w załączonych zastrzeżeniach patentowych, którego zakres nie ogranicza się do tych przykładów realizacji. Znawca bez trudu i rutynowo może zaproponować liczne inne przykłady realizacji z wykorzystaniem różnych komponentów softwarowych i hardwarowych. Znawca znający literaturę przedmiotu również bez trudu zaproponuje inne metody definiowania problemu optymalizacji jak i stosowanych metod jego rozwiązania.

[0060] Serwer 11 może stanowić dowolna maszyna cyfrowa, w szczególności komputer, komputer osobisty, mikrokomputer, lub procesor sygnałowy, zaś sterownik 11a może stanowić program z zestawem instrukcji w których zaimplementowano sterowanie siecią SDN oraz wykonywanie sposobu według wynalazku. Program ten może stanowić funkcja, procedura, zespół obiektów z metodami, aplet, servlet, kod źródłowy, biblioteka lub sekwencja instrukcji. Program ten może być zapisany na nośniku danych takim jak dysk optyczny, dysk twardy, dysk magnetyczny lub półprzewodnikowy lub w postaci zakodowanych danych

ładowalnych do pamięci komputera. Nośnikiem też może być sygnał w sieci komputerowej przewodowej lub bezprzewodowej.

[0061] Zastosowanie w zastrzeżeniach terminu „zawiera” lub jego synonimów, po którym następuje lista elementów oznacza, że listę tę należy interpretować niezamykająco, w sposób niewykluczający zastosowania innych, dodatkowych elementów. Wymienienie elementu w liczbie pojedynczej nie oznacza ograniczenia do jednego elementu w zakresie wynalazku mieszczą się też rozwiązania zawierające zbiory tych elementów.

Zastrzeżenia patentowe

1. Sposób obsługi przepływu pakietów w sieci w programowalnej sieci komputerowej SDN zawierającej serwer (11) ze sterownikiem SDN (11a) oraz przełącznik SDN (12) z tabelą przepływów (12a,12b,12c) zawierającą wpisy szczegółowe (A.1, A.2, A.3 B.1, B.2, B.3, C.1) określające parametry potrzebne do przyporządkowania pakietu do przepływu danych, a także pole określające priorytet, przy czym wpisy szczegółowe zawierają się w odpowiadających im wpisach zagregowanych (A,B,C), które są przygotowywane z wyprzedzeniem dla określenia polityki routingu przed rozpoczęciem obsługi przepływów w oknie czasowym o predefiniowanej długości, **znamienny tym, że**

Predefiniowana długość okna wynosi przynajmniej dwie godziny,

sterownik SDN (11a) z przełącznika SDN (12) uzyskuje dane dotyczące statystyk przepływów dla wpisów zainstalowanych w tablicach przypływów, natomiast

przychodzący pakiet (P1) należący do nowego przepływu wysyła się do sterownika SDN (11a) obsługując go jednocześnie zgodnie z odpowiadającym mu wpisem zagregowanym, zaś sterownik SDN (11a) za pomocą mechanizmu reaktywnego definiuje nowy wpis szczegółowy odpowiadający temu przepływowi oraz

proaktywnie wprowadza zmiany do wpisów zagregowanych (A,B,C) przewidzianych do zastosowania do określenia polityki routingu w następnym oknie czasowym stosując optymalizację statyczną

funkcji celu sformułowanej z wykorzystaniem przynajmniej jednego kryterium wybranego z grupy obejmującej zużycie energii, sumaryczne opóźnienie na wszystkich łączach w sieci, maksymalne opóźnienie spośród wszystkich łączy w sieci, sumaryczne wykorzystanie zasobów na wszystkich łączach sieci, maksymalne

wykorzystanie zasobów spośród wszystkich łączy w sieci,

przy użyciu danych dotyczących statystyk przepływów, przy czym wpisom zagregowanym (A,B,C) przyporządkowuje się niższe priorytety niż wpisom szczegółowym (A.1, A.2, A.3 B.1, B.2, B.3, C.1).

2. Sposób według zastrz. 1, znamienny tym, że przełącznik SDN (12) zawiera pierwszą tabelę przepływów (12b) zawierającą wpisy zagregowane (A,B,C) oraz przynajmniej jedną pomocniczą tabelę przepływów (12c) zawierającą wpisy szczegółowe (A.1, A.2, A.3).

3. Sposób według zastrz. 2, znamienny tym, że wpisy szczegółowe (A.1, A.2, A.3) odpowiadające temu danemu wpisowi zagregowanemu (A) w tabeli przepływów (12b) znajdują się w odpowiadającej temu wpisowi zagregowanemu osobnej tabeli pomocniczej (12c) przy czym ta tabela pomocnicza zawiera ponadto wpis domyślny (TABLE_MISS).

4. Sposób według zastrz. 1 albo 2 albo 3, znamienny tym, zapisuje się w pamięci serwera (11) historyczne dane dotyczące statystyk przepływów w kolejnych oknach czasowych a długość okna czasowego poddaje się optymalizacji w osobnym procesie, z wykorzystaniem tych historycznych danych.

5. Sposób według zastrz. 4, znamienny tym, że optymalizację statyczną tablicy przepływów dla okna o określonej długości prowadzi się w najbliższym poprzedzającym je oknie o tej samej długości.

6. Programowalna sieć komputerowa SDN zawierająca serwer (11) ze sterownikiem (11a) oraz przynajmniej jeden przełącznik SDN (12) z tabelą przepływów (12a, 12b, 12c), znamienna tym, że sterownik SDN (11a) jest przystosowany do wykonywania sposobu według dowolnego z zastrz. od 1 do 5.

7. Programowalna sieć komputerowa SDN według zastrz. 6, znamienna tym, że tabela przepływów w przełączniku SDN (12) jest zapisana w pamięci TCAM.

8. Produkt programu komputerowego przystosowany do pełnienia funkcji sterownika programowalnej sieci komputerowej SDN (11a), znamienny tym, że zawiera zestaw instrukcji, które po załadowaniu do pamięci serwera (11) powodują wykonywanie sposobu według dowolnego z zastrz. od 1 do 5.

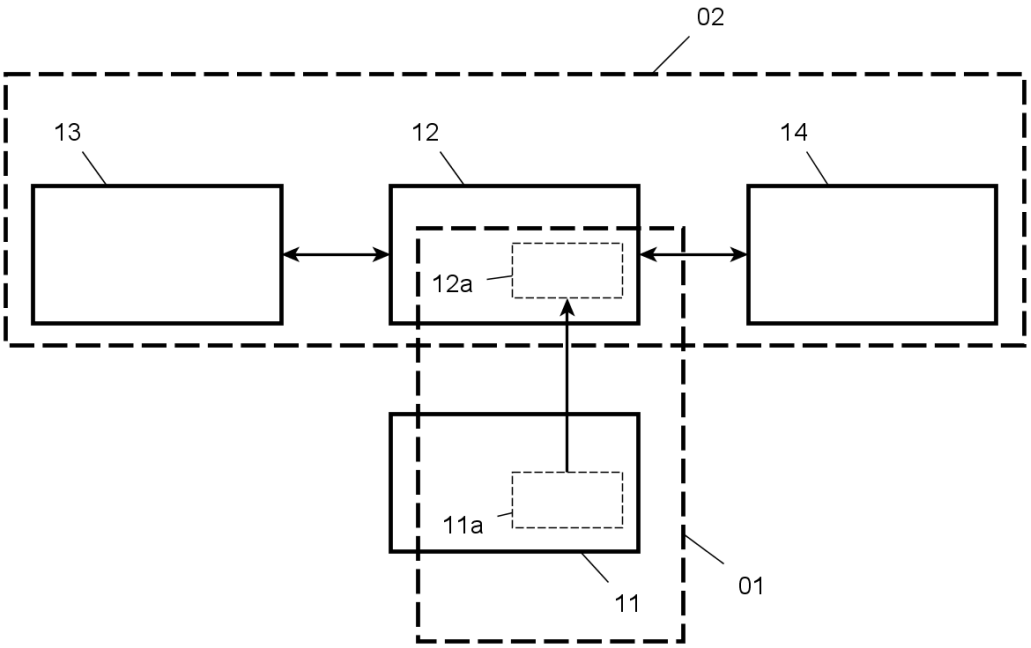
02
13
12
14
12a
11a
11
01

Fig. 1

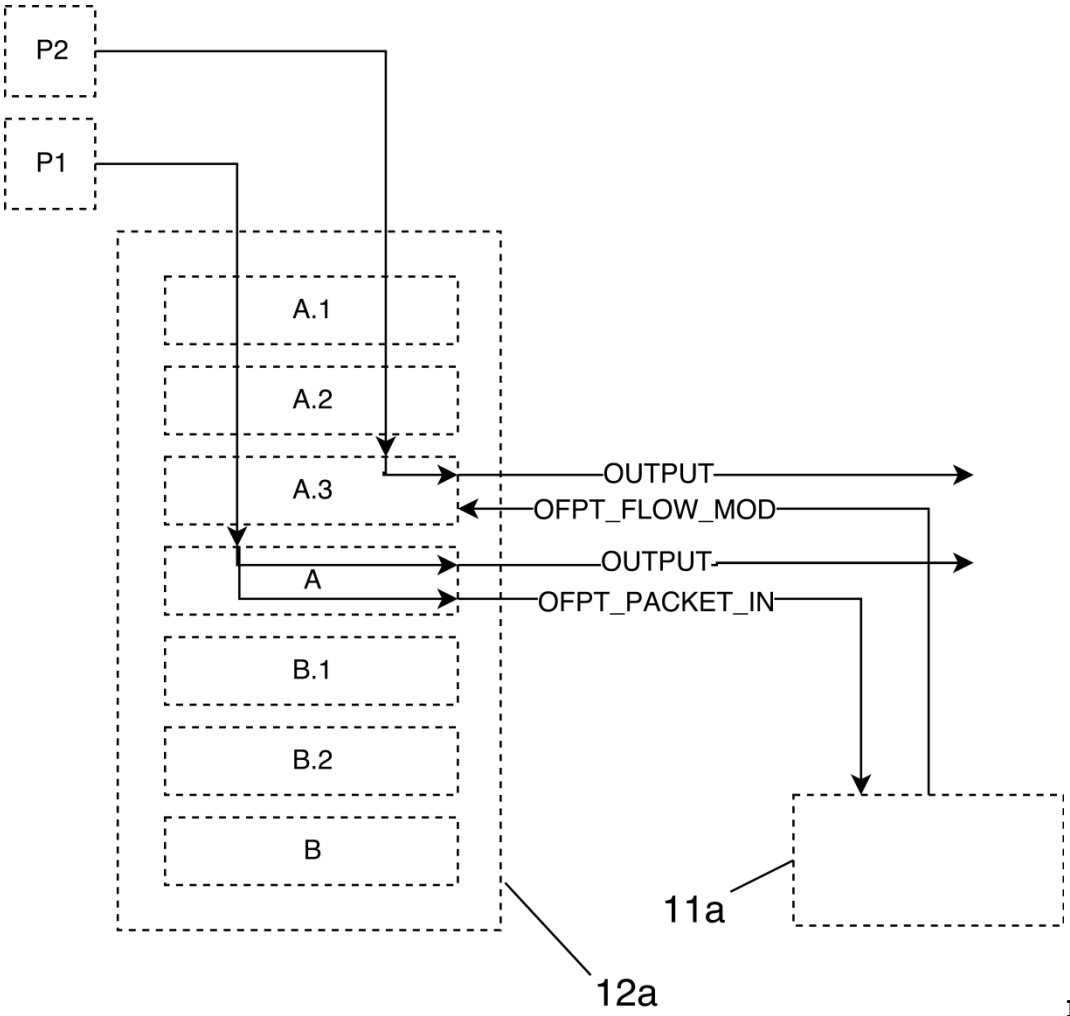
P2
P1
A.1
A.2
A.3
A
B.1
B.2
B
OUTPUT
OFPT_FLOW_MOD
OUTPUT
OFPT_PACKET_IN
11a
12a

Fig. 2

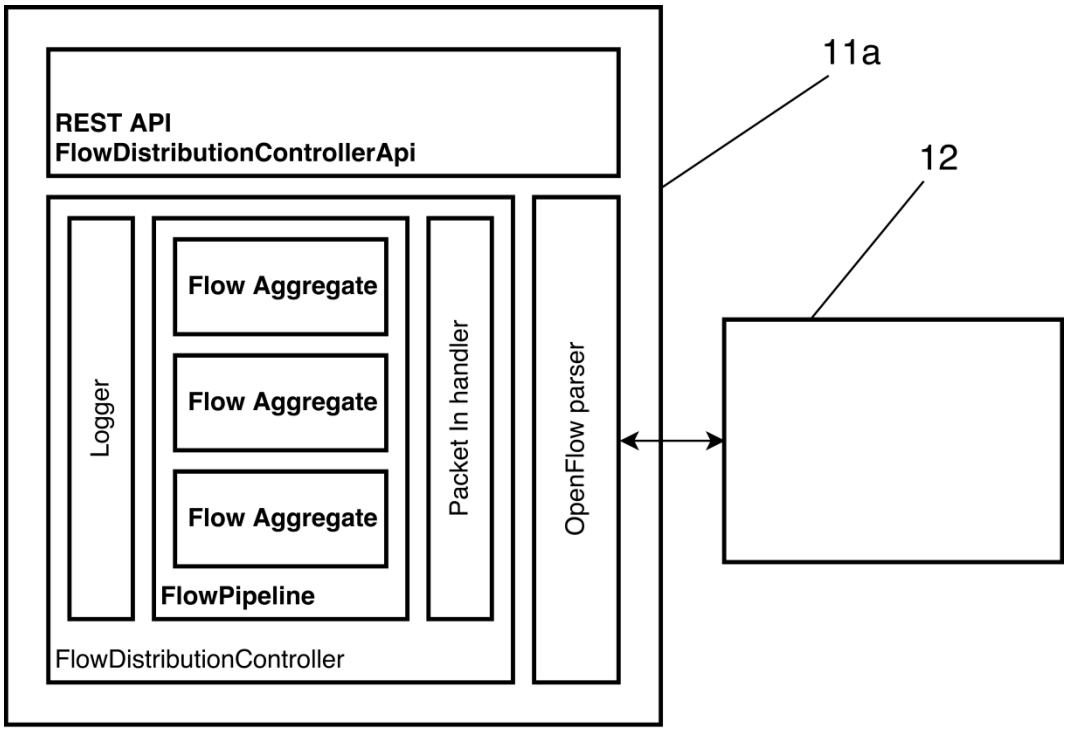
REST API
FlowDistributionControllerApi
Logger
Flow Aggregate
Flow Aggregate
Flow Aggregate
FlowPipeline
Packet In handler
OpenFlow parser
FlowDistributionController
11a
12

Fig. 3

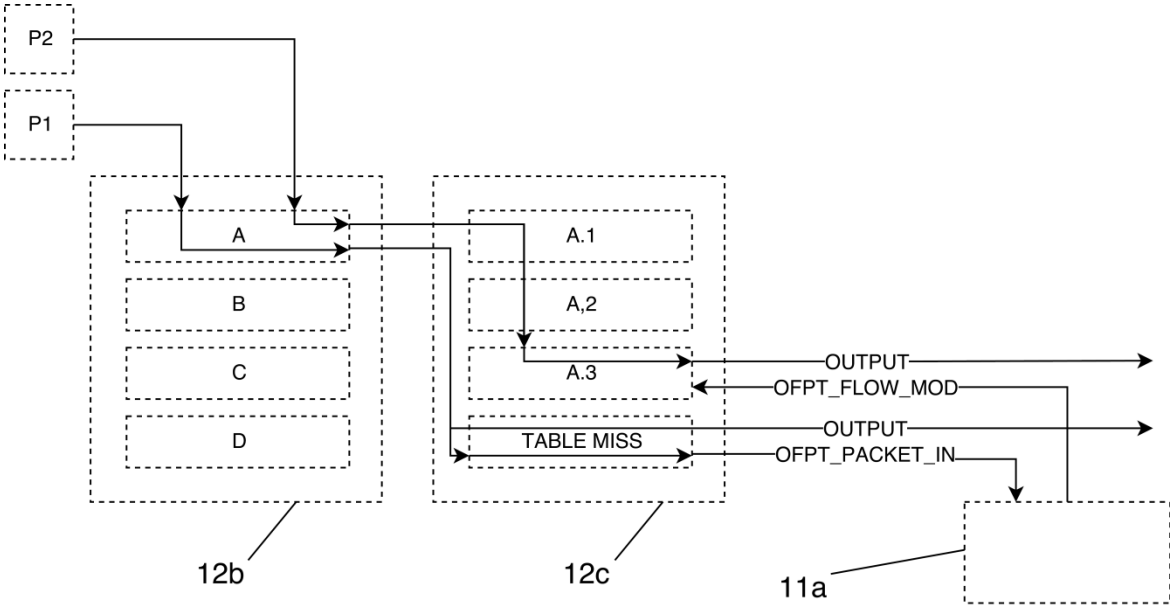
P2
P1
A
B
C
D
A.1
A,2
A.3
TABLE MISS
OUTPUT
OFPT_FLOW_MOD
OUTPUT
OFPT_PACKET_IN
12b
12c
11a

Fig. 4