

Akademia Górniczo-Hutnicza im. Stanisława Staszica w Krakowie  
Wydział Informatyki, Elektroniki i Telekomunikacji  
Katedra Telekomunikacji

Autoreferat Rozprawy Doktorskiej  
**Piotr Boryło**

# **Provisioning of Energy-Aware Cloud Services Over Optical Networks**

Świadczenie usług w chmurze za pośrednictwem sieci optycznych  
z uwzględnieniem pochodzenia wykorzystywanej energii

*Promotor:*

***Prof. dr hab. inż. Andrzej Jajszczyk***

*Promotor pomocniczy:*

***Dr inż. Artur Lason***

*Kraków 2015*



Liczba oraz atrakcyjność usług oferowanych za pośrednictwem sieci ciągle rośnie. W kontekście użytkowników indywidualnych najpopularniejsze usługi dotyczą przechowywania, wymiany i współdzielenia plików oraz wideo na żądanie. Osobnym zagadnieniem są usługi w chmurze oferowane firmom. W pierwszej kolejności należy wskazać kolokację zasobów. Począwszy od zasobów potrzebnych do uruchomienia najprostszych serwisów internetowych, przez rozbudowane wielowarstwowe aplikacje, a zakończywszy na zadaniach obliczeniowych wielkiej skali. Często stosowanym rozwiązaniem jest także wynoszenie usług, jak np. poczta elektroniczna, do chmury. Wszelkie usługi oferowane za pośrednictwem sieci w zewnętrznej infrastrukturze niosą firmom szereg korzyści. Obniżają barierę wejścia na rynek, pozwalają zredukować koszty operacyjne oraz eliminują potrzebę utrzymywania własnej infrastruktury IT. Dodatkowo, powstaje możliwość niezależnienia pracowników od konkretnego sprzętu fizycznego przez umożliwienie pracy zdalnej z wykorzystaniem maszyn wirtualnych udostępnianych przez sieć. Co więcej, większość danych w przedsiębiorstwach jest przesyłana i przechowywana w formie elektronicznej co wymusza potrzebę tworzenia i przechowywania kopii zapasowych. Infrastruktura chmury obliczeniowej świetnie nadaje się do tych celów.

Rozważając tematykę świadczenia usług w chmurze nie można pominąć aspektu sieciowego, który jest niezbędny aby zapewnić dostęp do zasobów obliczeniowych. Efekt potęgowany jest przez fakt, że infrastruktura chmury przyjmuje coraz to bardziej globalny charakter. Należy więc położyć nacisk nie tylko na sieci wewnątrz centrów danych ale także na sieci rozległe. W literaturze oraz działaniach operatorów można zaobserwować trend integracji sieci z zasobami IT w celu wydajnego świadczenia usług. Taka integracja jest jednak utrudniona przez fakt, że obie składowe infrastruktury są własnością różnych instytucji.

Sieci optyczne są wskazywane jako najwłaściwsze rozwiązanie w przypad-

ku świadczenia nowoczesnych usług. Optyczne techniki transmisji zapewniają odpowiednio dużą przepustowość i małe opóźnienia przy dużej efektywności energetycznej. Jednocześnie, trend w kierunku przeniesienia warstwy sterowania siecią do centralnego sterownika (*Software Defined Networking*) kreuje nowe możliwości dla sieci transportowych. Jedną z opcji, znajdującą swoje odzwierciedlenie w proponowanej tematyce badawczej, jest optymalizacja sieci transportowych pod kątem świadczenia usług w chmurze przez komunikację centralnego sterownika sieci z narzędziami do organizacji pracy chmury (*cloud orchestration software*).

Centra danych wskazywane są jako jeden z głównych udziałowców w poborze energii przez sektor ICT (*Information and Communication Technologies*). Wynika to z rosnącej liczby dostępnych usług oraz faktu, że istniejące usługi oferowane za pośrednictwem sieci wykazują zwiększone zapotrzebowanie na zasoby sieciowe oraz obliczeniowe. Dlatego też, sektor ICT ma coraz większy wpływ na interakcję człowieka ze środowiskiem naturalnym. Coraz więcej czynności może odbywać się za pośrednictwem sieci, np. praca zdalna z miejsca zamieszkania, zakupy w sklepach internetowych, wideokonferencje. W ten sposób ograniczana jest liczba podróży, maleje więc zużycie paliw czy też emisja dwutlenku węgla. Jednak z drugiej strony, infrastruktura sieciowa i obliczeniowa potrzebna do świadczenia wspomnianych usług przejawia bardzo duże zapotrzebowanie na energię elektryczną. Według [17] sektor ICT w 2008 roku w skali globalnej zużywał 8% energii elektrycznej, a udział ten wzrośnie do 20% w 2020 roku. Dodatkowo, od 2% do 4% emitowanego dwutlenku węgla pochodzi z sektora ICT [16]. Znajduje to swoje odzwierciedlenie w wysiłkach wkładanych w minimalizację zużycia energii przez pojedyncze centra danych. Następnym krokiem jest optymalizacja rozkładu obciążenia pomiędzy wieloma centrami danych dostarczającymi usługi równolegle. Jest to uzasadnione ponieważ infrastruktura złożona z wielu rozproszonych i redundantnych centrów danych połączonych siecią szkieletową staje się normą dla globalnych dostawców usług. Zmniejszenie poboru energii elektrycznej przez ten sektor może nie tylko zredukować negatywny wpływ na środowisko naturalne, ale również zmniejszyć koszty operacyjne ponoszone przez operatorów infrastruktury teleinformatycznej. Zagadnienia związane z ograniczeniem zużycia energii elektrycznej w sektorze ICT są zatem istotnym tematem badawczym, który może przynieść wymierne korzyści dla gospodarki oraz dla operatorów infrastruktury [10].

Koncepcja poprawy efektywności energetycznej oraz pomniejszania negatywnego wpływu na środowisko naturalne jest popularnym zagadnieniem w niemalże każdej dziedzinie naszego życia. W odniesieniu do sektora ICT może być motywowane wieloma czynnikami. Najbardziej oczywistym i naturalnym jest dbałość o otaczające nas środowisko. Jednakże, nie bez znaczenia pozostają takie czynniki jak poprawa wizerunku przedsiębiorstwa czy marketing. Dostawca usług w chmurze promujący się jako przyjazny środowisku może przyciągnąć zainteresowanie dodatkowych klientów. Należy jednak pamiętać, że najistotniejsze pozostają czyn-

niki ekonomiczne. Ogromne zużycie energii przekłada się na koszty operacyjne. Z jednej strony obniżenie całkowitego poboru energii prowadzi bezpośrednio do redukcji kosztów. Ponadto, w wielu krajach (np. Wielka Brytania lub Chiny) rozważane są regulacje prawne pozwalające obniżyć podatki na energię pochodzącą ze źródeł odnawialnych [1]. Jednocześnie, operatorzy sieci mogą zachęcać dostawców usług w chmurze przez gotowość do wdrażania mechanizmów poprawy efektywności energetycznej. Taka sposobność jest atrakcyjna szczególnie w perspektywie malejących przychodów za usługi związane ze standardowym przynoszeniem ruchu. Innowacyjne koncepcje, jak np. sieci programowalne o centralnym sterowaniu, stanowią szansę dla operatorów sieci na zwiększenie zysków przez oferowanie nowych funkcjonalności, jak np. uwzględnianie rodzaju źródła wykorzystywanej energii.

Rozważana infrastruktura złożona jest z centrów danych (DCs) połączonych siecią optyczną. Centra danych powiązane są z wybranymi węzłami sieci. Niektóre z centrów danych zasilane są z odnawialnych źródeł energii podczas gdy reszta zasilana jest ze źródeł konwencjonalnych. Sieć przełącza kanały optyczne (wavelength division multiplexing, WDM). Topologia fizyczna jest zadana i nie zmienia się w czasie. Konwertery długości fali nie są używane oraz nie ma możliwości zmiany trasy już zestawionych ścieżek optycznych. Jako, że jedną z głównych cech chmury jest świadczenie usług w odpowiedzi na napływające żądania, należy przyjąć, że żądania te napływają losowo i zadane są jedynie określonym rozkładem prawdopodobieństwa. W związku z tym, zarówno zasoby IT jak i sieciowe muszą być przydzielane bezpośrednio po przyjęciu żądania w warunkach aktualnego stanu infrastruktury. Kolejne założenie wynika z faktu, że w chmurze obliczeniowej te same usługi są oferowane jednocześnie w różnych centrach danych. W związku z tym możliwy jest wybór jednego spośród zbioru dostępnych centrów danych w celu dostarczenia usługi klientom. Odpowiadający takiej sytuacji schemat routingu nosi nazwę jeden do jednego z wielu (*anycast*) [7]. Podczas wyboru centrum danych można kierować się wieloma czynnikami, np. wykorzystaniem zasobów sieciowych, dostępnością energii odnawialnej czy też zasobami IT dostępnymi w konkretnych lokalizacjach

Niniejsza rozprawa doktorska jest poświęcona świadczeniu usług w chmurze obliczeniowej z uwzględnieniem pochodzenia wykorzystywanej energii. Zbadano możliwość redukcji emisji dwutlenku węgla przy założeniu, że chmura obliczeniowa jest złożeniem dwóch infrastruktur utrzymywanych odpowiednio przez operatora sieci optycznej oraz dostawcę zasobów obliczeniowych w centrach przetwarzania danych. Przedstawiono rozwiązanie problemu dynamicznego routingu w komutowanej sieci optycznej z jednoczesnym świadczeniem usług w centrach przetwarzania danych. Zarówno architektura chmury jak i sposób świadczenia usług pociągają za sobą szereg wymagań jakie to rozwiązanie powinno spełnić. Oczekuje się, że proponowane rozwiązanie będzie wybierać docelowe centrum danych opierając

się na dostępności energii odnawialnej. Przy czym powinna istnieć możliwość dopasowania preferencji do aktualnych warunków panujących w sieci. Strategie routingu typu *anycast* spełniają te wymagania. Proponowane rozwiązanie powinno ponadto uwzględniać fakt oferowania różnych typów usług w chmurze w celu efektywnego wykorzystania energii pochodzącej ze źródeł odnawialnych. Trzy typy usług w chmurze są rozważane. W rozprawie doktorskiej zaproponowano schemat dopasowywania strategii routingu *anycast* do cech charakterystycznych tych typów usług. Badania potwierdziły możliwość osiągnięcia istotnej poprawy przez wykorzystanie zaproponowanego schematu dopasowywania. Od zaproponowanego rozwiązania oczekuje się wreszcie aby brało pod uwagę fakt, że infrastruktura sieci i chmury zarządzane są przez oddzielne organizacje. Rozważa się różne poziomy współpracy i styków służących wymianie danych pomiędzy operatorami. W rozprawie zaproponowano model takiej współpracy oraz poddano go szczegółowym badaniom. Opierając się na analizie przeprowadzonej w rozprawie, założono że sieć sterowana jest przez centralny sterownik zgodny z ideą SDN podczas gdy do automatycznego sterowania zasobami obliczeniowymi użyto narzędzi organizacji pracy chmury (cloud orchestration software). Wydajność zaproponowanych mechanizmów i strategii została przebadana za pomocą symulacji komputerowej w dwóch topologiach sieciowych. Szczegółowym badaniom poddano wszystkie proponowane mechanizmy. Uzyskane wyniki pozwalają na stwierdzenie, że zaproponowane rozwiązanie spełnia narzucone wymagania i pozwala na obniżenie ilości emitowanego dwutlenku węgla bez istotnego pogorszenia wydajności sieci.

Postawiono następującą tezę:

**It is possible to reduce carbon dioxide emissions of a cloud infrastructure comprising data centers interconnected via optical networks using energy-aware resource provisioning strategies.**

W tłumaczeniu na język polski teza brzmi następująco:

**Możliwa jest redukcja ilości dwutlenku węgla emitowanego przez centra danych tworzące chmurę obliczeniową połączoną siecią optyczną przez zastosowanie strategii rozkładu obciążenia uwzględniającego pochodzenie wykorzystywanej energii.**

Rozważany problem świadczenia usług w chmurze na żądanie z wykorzystaniem sieci optycznych oraz z uwzględnieniem źródła pochodzenia energii elektrycznej jest aktualnym i istotnym tematem. Proponowane rozwiązanie zostało w sposób przemyślany umiejscowione w architekturze chmury zakładającej koegzystencję infrastruktury obliczeniowej i sieciowej. W związku z powyższym proponowane rozwiązanie może być rozważane do wdrożenia w rzeczywistych systemach bez istotnego zwiększenia złożoności warstwy sterującej.

---

Część wyników przedstawionych w niniejszej pracy została wcześniej opublikowana w postaci trzech artykułów w czasopismach oraz dwóch referatów konferencyjnych:

- [2] P. Boryło, A. Lason, J. Rzasa, A. Szymanski, and A. Jajszczyk. Usługi w chmurze korzystające z sieci optycznych o minimalnym zużyciu energii. *Przegląd Telekomunikacyjny, Wiadomości Telekomunikacyjne*, 86(10):1323–1330, 2013.
- [3] P. Boryło, A. Lason, J. Rzasa, A. Szymanski, and A. Jajszczyk. Anycast Routing for Carbon Footprint Reduction in WDM Hybrid Power Networks with Data Centers. In *Proc. IEEE International Conference on Communications (ICC 2014)*, pages 3720–3726, Sydney, Australia, June 2014.
- [4] P. Boryło, A. Lason, J. Rzasa, A. Szymanski, and A. Jajszczyk. Assessment of green anycast strategies in hybrid power networks having regular topologies. *Przegląd Telekomunikacyjny, Wiadomości Telekomunikacyjne*, 83(8-9):748–754, 2014.
- [5] P. Boryło, A. Lason, J. Rzasa, A. Szymanski, and A. Jajszczyk. Fitting Green Anycast Strategies to Cloud Services in WDM Hybrid Power Networks. In *Proc. IEEE Global Communications Conference (GLOBECOM 2014)*, pages 2633–2639, Austin, TX, USA, Dec. 2014.
- [6] P. Boryło, A. Lason, J. Rzasa, A. Szymanski, and A. Jajszczyk. Green Cloud Provisioning Throughout Cooperation of a WDM Wide Area Network and a Hybrid Power IT Infrastructure. *Journal of Grid Computing (Springer)*, 1(1), 2015. doi: 10.1007/s10723-015-9354-7.

Wszystkie publikacje były przedmiotem szczegółowej recenzji przed przyjęciem do druku.





Rozwiązanie zaproponowane w rozprawie złożone jest z trzech mechanizmów składowych. Pierwszym z nich są strategie rutingu typu *anycast*. Przy podejmowaniu decyzji dotyczącej docelowego miejsca obsługi żądania kierują się one dostępnością energii pochodzącej ze źródeł odnawialnych. Zgodnie z przedstawionymi założeniami każde żądanie związane z użyciem zasobów chmury może zostać sklasyfikowane jako jeden z trzech typów usług: przetwarzanie jako usługa (Processing as a Service, PaaS), przechowywanie jako usługa (Storage as a Service, StaaS) oraz oprogramowanie jako usługa (Software as a Service, – SaaS). W rozprawie zawarto szczegółowy opis tych typów usług. Uwzględniając wymagania na zasoby sieciowe oraz średni pobór energii przez każdy z typów usług rozważane są różne schematy dopasowywania strategii *anycast* do tych typów. Efektywne wykorzystanie zasobów powiązanych z energią odnawialną jest nadrzędnym celem włączenia schematów dopasowywania do kompletnego rozwiązania. Problem rozwiązywany w rozprawie jest także poszerzony o założenie, że infrastruktura chmury i infrastruktura sieci optycznej są administrowane przez różne podmioty. Współpraca pomiędzy tymi podmiotami, czyli operatorem sieci i dostawcą usług, bywa utrudniona. Dzieje się tak głównie z uwagi na występujące często konflikty interesów oraz niechęć do ujawniania informacji o wewnętrznej strukturze i topologii systemu. Efektywne wykorzystanie dostępnych zasobów oraz zmniejszanie emisji dwutlenku węgla w takiej architekturze jest możliwe tylko dzięki opracowaniu modeli współpracy pomiędzy wspomnianymi podmiotami.

W celu formalizacji zapisu oraz dla uproszczenia sformułowań wprowadzono szereg symboli. Graf  $G(V, E)$  reprezentuje fizyczną topologię, gdzie zbiór  $V$  jest zbiorem węzłów a zbiór  $E$  zbiorem łączy. Zbiór  $V_{DC}$  opisuje węzły sieciowe powiązane z centrami danych. Centra danych zasilane ze źródeł odnawialnych nazywane będą *zielonymi*, a węzły sieci z nimi powiązane oznaczane będą jako zbiór  $V_{gDC}$ . Analogicznie, centra danych zasilane z konwencjonalnych źródeł

energii nazywane będą *brązowymi*, a węzły sieci z nimi powiązane oznaczane jako zbiór  $V_{bDC}$ . Indeksy dolne  $g$  i  $b$  będą we wszystkich symbolach oznaczały odpowiednio *zieloność* i *brązowość* centrów danych (od angielskich słów *green* i *brown*). Dla każdego z centrów danych  $d \in V_{DC}$ ,  $c_d$  i  $r_d$  oznaczają odpowiednio całkowitą ilość zasobów oraz dostępną obecnie ilość zasobów. Zbiory  $C_{DC}$  i  $R_{DC}$  zawierają odpowiednio  $c_d$  i  $r_d$  dla wszystkich  $d \in V_{DC}$ . Zbiór centrów danych, które posiadają wystarczającą ilość zasobów potrzebnych do obsługi obecnie przetwarzanego zadania oznaczono jako  $V_{rDC}$ . Także ten zbiór można podzielić na dwa podzbiory:  $V_{rgDC}$  i  $V_{rbDC}$  odpowiednio dla *zielonych* i *brązowych* centrów danych. Dynamiczny algorytm przydziału zasobów sieciowych służy do obsługi jednego z dwóch żądań:

1. Żądanie typu *unicast* od węzła źródłowego  $s \in V$  do węzła docelowego  $d \in V$ .
2. Żądanie typu *anycast* od węzła źródłowego  $s \in V_C$  do jednego z potencjalnych węzłów docelowych  $d \in D \subseteq V_{DC}$ . Żądania typu *anycast* dotyczą ruchu kierowanego do centrów danych. Żądania tego typu napływają najpierw do narzędzia organizacji pracy chmury i są dalej przekazywane do centralnego sterownika sieci programowalnej.

Żądania napływające do infrastruktury chmury oraz przekazywane do sterownika sieci oznaczane będą odpowiednio  $CR$  (cloud request) i  $LR$  (lightpath request). Natomiast symbol  $LR_r$  określa zapotrzebowanie żądania  $LR$  na zasoby obliczeniowe w centrum danych. Każde żądanie typu *anycast* powiązane jest z jednym typem usługi oferowanej w chmurze. Dla uproszczenia żądania te będzie nazywać żądaniami *PaaS*, *StaaS* lub *SaaS*.

## 2.1 Strategie routingu typu jeden do jednego z wielu

Strategie *randomGreen*, *closestGreen* i *closestGreenWithPenalty* są oryginalnymi propozycjami przedstawionymi w rozprawie. Dla celów porównania rozważono także strategie referencyjne. Strategia *single*, jest najprostszą i powszechnie znaną w literaturze strategią. Podstawową wadą strategii *single* jest fakt, że rozważa ona tylko jeden z węzłów jako docelowy podczas gdy głównym celem stosowania routingu typu *anycast* jest wykorzystanie nadmiarowości. Dlatego, aby zapewnić miarodajną ocenę, w rozprawie sformułowano dodatkowe strategie referencyjne: *random* i *closest*.

W pseudokodach opisujących strategie wykorzystywana jest operacja  $LAC(s, d)$ , która oznacza sprawdzenie dostępności zasobów sieciowych pomiędzy węzłem źródłowym  $s$ , a węzłem docelowym  $d$ . Operacja  $LAC(s, d)$  zwraca długość najkrótszej

dostępnej ścieżki liczoną w skokach sieciowych. Jeśli nie ma wolnych zasobów łączących oba węzły operacja zwraca nieskończoność. Jako parametr wejściowy wszystkie strategie przyjmują zbiór  $D$  będący podzbiorem  $V_{DC}$  ( $D \subseteq V_{DC}$ ). W skrajnym przypadku  $D = V_{DC}$ , co oznacza, że strategia działa na zbiorze wszystkich centrów danych. W niektórych przypadkach argumentami wejściowymi jest para  $(s, D) = (s, \{D_g, D_b\})$ . Oznacza to, że zbiór  $D$  zawiera dwa podzbiory  $D_g$  i  $D_b$ , odpowiednio *zielone* i *brązowe* centra danych w zbiorze  $D$ . Spełnione są następujące zależności:  $D_g \cap D_b = \emptyset$  i  $D_g \cup D_b = D$ . W przypadku gdy  $D = V_{DC}$  wówczas  $D_g = V_{gDC}$  i  $D_b = V_{bDC}$ .

### 2.1.1 Strategia *single*

Algorytm 1 jest formalnym zapisem strategii *single*. Pojedynczy węzeł docelowy  $d$  jest wybierany losowo ze zbioru  $D$ . Następnie sprawdzana jest dostępność zasobów sieciowych pomiędzy  $s$  i  $d$ . Jeśli takie zasoby są dostępne zestawiana jest odpowiednia ścieżka optyczna, a zwracany jest identyfikator węzła docelowego. Jeśli zasoby sieciowe są niedostępne, wówczas żądanie jest odrzucane a strategia zwraca *null*.

---

#### Algorytm 1 Strategia *single* [3]

---

**Input:**  $(s, D)$

- 1:  $d \leftarrow$  losowe DC ze zbioru  $D$
- 2: **if**  $LAC(s, d) < \infty$  **then**
- 3:     Zestaw ścieżkę optyczną pomiędzy  $s$  i  $d$
- 4:     **return**  $d$
- 5: **else**
- 6:     **return** *null*
- 7: **end if**

---

### 2.1.2 Strategia *random*

Algorytm 2 jest formalnym zapisem strategii *random*. Sterownik sieciowy wykonuje operację  $LAC(s, d)$  dla kolejnych węzłów docelowych  $d$  wybieranych losowo ze zbioru  $D$ . Strategia kończy działanie z sukcesem po znalezieniu pierwszego osiągalnego węzła  $d$ , w efekcie ścieżka jest zestawiana i zwracany jest identyfikator węzła docelowego. Jeśli żaden z potencjalnych węzłów docelowych  $d \in D$  nie jest osiągalny żądanie jest odrzucane a strategia zwraca *null*.

### 2.1.3 Strategia *closest*

Algorytm 3 jest formalnym zapisem strategii *closest*. Kontroler sieciowy wykonuje operację  $LAC(s, d)$  dla kolejnych węzłów docelowych  $d$ . Następnie porównywane

---

**Algorytm 2** Strategia *random* [3]

---

**Input:**  $(s, D)$

```

1: while  $D \neq \emptyset$  do
2:    $d \leftarrow$  losowe DC ze zbioru  $D$ 
3:   if  $LAC(s, d) < \infty$  then
4:      $D \leftarrow \emptyset$ 
5:     Zestaw ścieżkę optyczną pomiędzy  $s$  i  $d$ 
6:     return  $d$ 
7:   else
8:      $D \leftarrow D \setminus d$ 
9:     if  $D = \emptyset$  then
10:      return null
11:     end if
12:   end if
13: end while

```

---



---

**Algorytm 3** Strategia *closest* [3]

---

**Input:**  $(s, D)$

```

1:  $d \leftarrow$  null
2:  $dist \leftarrow \infty$ 
3: for all  $i \in D$  do
4:   if  $LAC(s, i) < dist$  then
5:      $d \leftarrow i$ 
6:      $dist \leftarrow LAC(s, i)$ 
7:   end if
8: end for
9: if  $d \neq$  null then
10:  Zestaw ścieżkę optyczną pomiędzy  $s$  i  $d$ 
11:  return  $d$ 
12: else
13:  return null
14: end if

```

---

są odległości pomiędzy tymi węzłami docelowym a węzłem źródłowym. W ten sposób wybierany jest najbliższy osiągalny węzeł docelowy, zestawiana jest ścieżka i zwracany jest identyfikator węzła docelowego. Jeśli żaden z potencjalnych węzłów docelowych  $d \in D$  nie jest osiągalny żądanie jest odrzucane a strategia zwraca *null*.

### 2.1.4 Strategia *randomGreen*

Algorytm 4 jest formalnym zapisem strategii *randomGreen*. Kontroler sieciowy wykonuje operację  $LAC(s, d)$  dla kolejnych węzłów docelowych  $d$  wybieranych losowo ze zbioru  $D_g$ . Strategia kończy działanie z sukcesem po znalezieniu pierwszego osiągalnego *zielonego* węzła  $d$ , w efekcie ścieżka jest zestawiana i zwracany jest identyfikator węzła docelowego. Jeśli żaden z potencjalnych węzłów docelowych

$d \in D_g$  nie jest osiągalny, ta sama procedura wykonywana jest dla węzłów *brązowych*,  $d \in D_b$ . Po raz kolejny strategia kończy działanie z sukcesem po znalezieniu pierwszego osiągalnego *brązowego* węzła  $d$ ; w efekcie ścieżka jest zestawiana i zwracany jest identyfikator węzła docelowego. Jeśli żaden z potencjalnych *brązowych* węzłów docelowych  $d \in D_b$  nie jest osiągalny żądanie jest odrzucane a strategia zwraca *null*.

---

**Algorytm 4** Strategia *randomGreen* [3]
 

---

**Input:**  $(s, D) = (s, \{D_g, D_b\})$

- 1: **while**  $D_g \neq \emptyset$  **do**
- 2:      $d \leftarrow$  losowe DC ze zbioru  $D_g$
- 3:     **if**  $LAC(s, d) < \infty$  **then**
- 4:          $D_g \leftarrow \emptyset$
- 5:          $D_b \leftarrow \emptyset$
- 6:         Zestaw ścieżkę optyczną pomiędzy  $s$  i  $d$
- 7:         **return**  $d$
- 8:     **else**
- 9:          $D_g \leftarrow D_g \setminus d$
- 10:    **end if**
- 11: **end while**
- 12: **while**  $D_b \neq \emptyset$  **do**
- 13:      $d \leftarrow$  losowe DC ze zbioru  $D_b$
- 14:     **if**  $LAC(s, d) < \infty$  **then**
- 15:          $D_b \leftarrow \emptyset$
- 16:         Zestaw ścieżkę optyczną pomiędzy  $s$  i  $d$
- 17:         **return**  $d$
- 18:     **else**
- 19:          $D_b \leftarrow D_b \setminus d$
- 20:         **if**  $D_b = \emptyset$  **then**
- 21:             **return** *null*
- 22:         **end if**
- 23:     **end if**
- 24: **end while**

---

### 2.1.5 Strategia *closestGreen*

Algorytm 5 jest formalnym zapisem strategii *closestGreen*. Kontroler sieciowy wykonuje operację  $LAC(s, d)$  dla kolejnych węzłów docelowych  $d \in D_g$ . Następnie porównywane są odległości pomiędzy tymi węzłami docelowym a węzłem źródłowym. W ten sposób wybierany jest najbliższy osiągalny *zielony* węzeł docelowy, zestawiana jest ścieżka i zwracany jest identyfikator węzła docelowego. Jeśli żaden z potencjalnych węzłów docelowych  $d \in D_g$  nie jest osiągalny ta sama procedura wykonywana jest dla węzłów *brązowych*,  $d \in D_b$ . Po raz kolejny porównywane są odległości pomiędzy tymi węzłami docelowym a węzłem źródłowym. W ten sposób wybierany jest najbliższy osiągalny *brązowy* węzeł docelowy, zestawiana

jest ścieżka i zwracany jest identyfikator węzła docelowego. Jeśli żaden z potencjalnych *brązowych* węzłów docelowych  $d \in D_b$  nie jest osiągalny, żądanie jest odrzucane a strategia zwraca *null*.

---

### Algorytm 5 Strategia *closestGreen* [3]

---

**Input:**  $(s, D) = (s, \{D_g, D_b\})$

- 1:  $d \leftarrow null$
- 2:  $dist \leftarrow \infty$
- 3: **for all**  $i \in D_g$  **do**
- 4:     **if**  $LAC(s, i) < dist$  **then**
- 5:          $d \leftarrow i$
- 6:          $dist \leftarrow LAC(s, i)$
- 7:     **end if**
- 8: **end for**
- 9: **if**  $d \neq null$  **then**
- 10:     Zestaw ścieżkę optyczną pomiędzy  $s$  i  $d$
- 11:     **return**  $d$
- 12: **else**
- 13:     **for all**  $i \in D_b$  **do**
- 14:         **if**  $LAC(s, i) < dist$  **then**
- 15:              $d \leftarrow i$
- 16:              $dist \leftarrow LAC(s, i)$
- 17:         **end if**
- 18:     **end for**
- 19:     **if**  $d \neq null$  **then**
- 20:         Zestaw ścieżkę optyczną pomiędzy  $s$  i  $d$
- 21:         **return**  $d$
- 22:     **else**
- 23:         **return** *null*
- 24:     **end if**
- 25: **end if**

---

#### 2.1.6 Strategia *closestGreenWithPenalty*

Algorytm 6 jest formalnym zapisem strategii *closestGreenWithPenalty*, której działanie jest analogiczne do działania strategii *closest*. Kontroler sieciowy wykonuje operację  $LAC(s, d)$  dla kolejnych węzłów docelowych  $d \in D$ . Następnie porównywane są odległości pomiędzy tymi węzłami docelowym a węzłem źródłowym. Podstawowa różnica pomiędzy strategią *closestGreenWithPenalty* a *closest* wynika z faktu, że odległości do węzłów *brązowych* (pomiędzy  $s$  a  $d \in D_b$ ) mnożone są przez skalujący parametr kary (*penalty*). W ten sposób wybierane jest preferowane centrum danych, zestawiana jest ścieżka i zwracany jest identyfikator węzła docelowego. Jeśli żaden z potencjalnych węzłów docelowych  $d \in D$  nie jest osiągalny żądanie jest odrzucane a strategia zwraca *null*.

Podsumowując, trzy zaproponowane strategie (*randomGreen*, *closestGreen* i *closestGreenWithPenalty*) stwarzają możliwość obniżenia emisji dwutlenku węgla

**Algorytm 6** Strategia `closestGreenWithPenalty` [3]

---

**Input:**  $(s, D) = (s, \{D_g, D_b\})$

```

1:  $d \leftarrow null$ 
2:  $dist \leftarrow \infty$ 
3: for all  $i \in D$  do
4:   if  $i \in D_b$  then
5:     if  $LAC(s, i) \cdot Penalty < dist$  then
6:        $d \leftarrow i$ 
7:        $dist \leftarrow LAC(s, i) \cdot Penalty$ 
8:     end if
9:   else if  $i \in D_g$  then
10:    if  $LAC(s, i) < dist$  then
11:       $d \leftarrow i$ 
12:       $dist \leftarrow LAC(s, i)$ 
13:    end if
14:  end if
15: end for
16: if  $d \neq null$  then
17:   Zestaw ścieżkę optyczną pomiędzy  $s$  i  $d$ 
18:   return  $d$ 
19: else
20:   return  $null$ 
21: end if

```

---

przez preferencję *zielonych* centrów danych względem *brązowych*. Efektem ubocznym może być wzrost średniej długości ścieżki optycznej w sieci, a co za tym idzie, także wykorzystanie zasobów sieciowych. Ten efekt może być szczególnie widoczny w przypadku strategii *randomGreen* i *closestGreen*, w których węzły *zielone* są ściśle preferowane. Z tych względów strategia *closestGreenWithPenalty* została zaproponowana w celu równoważenia zysków w dziedzinie emisji dwutlenku węgla i strat wynikających z pogorszenia wydajności sieci. Im większa wartość zostanie przypisana do parametru *penalty* tym bardziej węzły *zielone* są preferowane i tym większa jest średnia długość ścieżki.

## 2.2 Schematy dopasowywania strategii routingu do typów usług oferowanych w chmurze

Uwzględniając wstępne wyniki badań, w rozprawie zaproponowano schemat *compound* dopasowujący strategię *anycast* do typów usług oferowanych w chmurze. Do celów porównania zdefiniowano także trzy referencyjne schematy dopasowywania (*closest*, *closestGreen* i *closestGreenWithPenalty*). W tych schematach wszystkie typy usług obsługiwane są przez tę samą strategię *anycast*.

### 2.2.1 Schemat dopasowywania *compound*

Uwzględniając właściwości typów usług oraz strategii *anycast* zaproponowano schemat *compound*. Żądania typu *PaaS* charakteryzują się największym poborem energii wewnątrz centrów danych. Z tego względu do obsługi tych żądań wybrano strategię *closestGreen*, która ściśle preferuje *zielone* centra danych względem *brązowych*. Oczekuje się, że obniży to poziom emisji dwutlenku węgla, jednak najprawdopodobniej dojdzie do tego kosztem pogorszenia wydajności sieci. W przypadku żądań *StaaS* zapotrzebowanie na energię jest pomijalne. Dlatego żądania te obsługiwane będą przez strategię *closest*. Nie wprowadzana ona żadnej preferencji *zielonych* centrów danych, zapewniając jednocześnie wykorzystanie najkrótszej dostępnej ścieżki optycznej, a tym samym najmniejsze zużycie zasobów sieciowych. Ostatni typ żądań, *SaaS* cechuje wydłużenie czasu zajmowania zasobów sieciowych względem dwóch poprzednich typów usług. Ponadto zapotrzebowanie na energię może zmieniać się w zakresie ograniczonym przez zapotrzebowanie żądań typu *PaaS* i *StaaS*. Taka zmienność usług typu *SaaS* znajduje swoje odzwierciedlenie w ich obsłudze przez strategię *closestGreenWithPenalty*. Strategia ta wykorzystuje parametr *penalty* aby dopasować stopień preferencji węzłów *zielonych* do właściwości żądań *SaaS*. Dzięki schematowi *compound* dla żądań o największych wymaganiach energetycznych *zielone* centra danych byłyby ściśle preferowane, podczas gdy dla innych usług ta priorytetyzacja byłaby ważona względem pogorszenia wydajności sieci.

### 2.2.2 Referencyjne schematy dopasowywania

Schemat referencyjny *closest* stosuje strategię *closest* do obsługi wszystkich typów usług. Jest to schemat bazowy do oceny wpływu na wydajność sieci ponieważ dla każdego typu usług przynoszony jest najkrótszą dostępną ścieżką. Jednocześnie, schemat *closest* nie wprowadza żadnej preferencji węzłów *zielonych* dla któregośkolwiek typu usług.

Schemat *closestGreen* reprezentuje dokładnie odwrotne podejście. *Zielone* centra danych są ściśle preferowane przez wykorzystanie strategii *closestGreen* do obsługi wszystkich typów usług. W efekcie, każde żądanie dotyczące usług w chmurze jest kierowane do *zielonych* centrów danych nawet jeśli są one odległe od źródła. Minimalizacja emisji dwutlenku węgla może być w tym przypadku okupiona degradacją wydajności sieci.

Ostatni z referencyjnych schematów dopasowywania, schemat *closestGreenWithPenalty*, wprowadza bardziej umiarkowaną preferencję *zielonych* centrów danych. Wszystkie typy usług obsługiwane są przez strategię *closestGreenWithPenalty* o takim samym poziomie preferencji węzłów *zielonych*. Parametr *penalty* zastosowano w celu równoważenia zysków w dziedzinie emisji dwutlenku węgla i negatywnego wpływu na wydajność sieci.



Tabela 2.1 podsumowuje opisy schematów dopasowywania i objaśnia zależności pomiędzy schematami a strategiami *anycast*.

Tabela 2.1: Podsumowanie schematów dopasowywania

| Schemat dopasowywania          | Typ usługi  | Strategia <i>anycast</i>       |
|--------------------------------|-------------|--------------------------------|
| <i>closest</i>                 | <i>PaaS</i> | <i>closest</i>                 |
|                                | <i>SaaS</i> |                                |
|                                | <i>SaaS</i> |                                |
| <i>closestGreen</i>            | <i>PaaS</i> | <i>closestGreen</i>            |
|                                | <i>SaaS</i> |                                |
|                                | <i>SaaS</i> |                                |
| <i>closestGreenWithPenalty</i> | <i>PaaS</i> | <i>closestGreenWithPenalty</i> |
|                                | <i>SaaS</i> |                                |
|                                | <i>SaaS</i> |                                |
| <i>compound</i>                | <i>PaaS</i> | <i>closestGreen</i>            |
|                                | <i>SaaS</i> | <i>closest</i>                 |
|                                | <i>SaaS</i> | <i>closestGreenWithPenalty</i> |

## 2.3 Modele współpracy pomiędzy operatorem sieci a dostawcą usług w chmurze

Różnice pomiędzy modelami współpracy wynikają głównie z ilości informacji wymienianej pomiędzy operatorem sieci a dostawcą usług. Można przyjąć, że im mniej informacji podmioty muszą ujawnić tym lepiej. Z drugiej jednak strony operatorzy są świadomi, że współpraca jest niezbędna do efektywnego świadczenia usług w chmurze oraz zmniejszenia emisji dwutlenku węgla. W takiej sytuacji, kompromis polega na wymianie ściśle określonych informacji przez jasno zdefiniowane interfejsy. Umożliwia to pełną kontrolę nad wymienianą informacją co zwiększa atrakcyjność modeli w oczach operatorów.

Pewne ogólne zasady obsługi żądań znajdują zastosowanie we wszystkich modelach. Żądanie związane z usługą w chmurze dociera do narzędzi organizacji pracy chmury (*cloud orchestration software*), a następnie jest przekazywane do centralnego sterownika sieci programowalnej jako żądanie *anycast*. W niektórych przypadkach, przy przekazywaniu żądania dołączane są dodatkowe informacje. W ten sposób narzędzia organizacji pracy chmury dostarczają danych wejściowych sterownikowi sieci. Sterownik wybiera węzeł docelowy  $d$  do obsługi tego żądania oraz zestawia ścieżkę optyczną pomiędzy  $s$  i  $d$ . Węzeł  $d$  wybierany jest z wykorzystaniem strategii *anycast* przypisanej do konkretnego typu usługi. Następnie, narzędzie organizacji pracy chmury otrzymuje informacje o wybranym centrum

danych i próbuje przydzielić niezbędne zasoby obliczeniowe w tym centrum. W przypadku gdy nie ma wystarczającej ilości zasobów żądanie jest odrzucane, a przydzielone uprzednio zasoby sieciowe są zwalniane.

Każda operacja zawarta w pseudokodach opisujących modele współpracy wykonywana jest przez narzędzie organizacji pracy chmury, za wyjątkiem instrukcji zawartych wewnątrz bloków  $SDN(arguments)$ . Instrukcje te wykonywane przez sterownik sieci programowalnej. Zmienna  $arguments$  oznacza informacje przekazane sterownikowi przez narzędzie organizacji pracy chmury. Sterownik sieci wykonuje operację  $arwa(s, D)$  oznaczającą wywołanie strategii *anycast* powiązanej z usługą, której dotyczy przetwarzane żądanie. Węzeł źródłowy  $s$  i zbiór potencjalnych węzłów docelowych  $D$  są parametrami wejściowymi strategii *anycast*. Z kolei zmienna  $arwa$  jako wartość przyjmuje nazwę strategii *anycast* wywoływanej przez operację  $arwa(s, D)$ . Dodatkowe informacje specyficzne dla konkretnych modeli współpracy zostały wyjaśnione w tekście oraz w formie komentarzy do pseudokodów.

### 2.3.1 Model *overlay*

Algorytm 7 jest formalnym zapisem modelu *overlay*, w którym narzędzie organizacji pracy chmury przekazuje sterownikowi sieci żądanie bez żadnych dodatkowych informacji. W związku z tym, w przypadku modelu *overlay* współpraca pomiędzy operatorami ograniczona jest do absolutnego minimum. Sterownik sieci programowalnej nie posiada informacji o dostępności zasobów obliczeniowych w poszczególnych centrach danych. Z tego względu, jako docelowe centrum danych sterownik może wskazać lokalizację nie posiadającą wystarczającej ilości zasobów do obsługi tego żądania. Jest to główną wadą tego modelu współpracy. Może bowiem doprowadzić do istotnego zwiększenia prawdopodobieństwa blokady żądań, pomimo że w sieci znajdują się centra danych, do których można zestawić ścieżkę optyczną i które posiadają wystarczającą ilość zasobów obliczeniowych. Żądanie jest obsłużone tylko wtedy gdy węzeł docelowy  $d$  wskazany przez sterownik sieci posiada zasoby niezbędne do obsłużenia żądania.

---

#### Algorytm 7 Model *overlay* [6]

---

**Input:**  $CR$

- 1: **SDN** ( $LR$ )
  - 2:      $d \leftarrow arwa(s, V_{DC})$
  - 3: **end SDN**
  - 4: **if**  $d \neq null$  AND  $r_d \geq LR_r$  **then**
  - 5:     Zarezerwuj zasoby w  $d$
  - 6:      $r_d \leftarrow r_d - LR_r$
  - 7: **else**
  - 8:     Odrzuć żądanie
  - 9: **end if**
-

### 2.3.2 Model *augmented*

Algorytm 8 jest formalnym zapisem modelu *augmented*. W tym przypadku narzędzie organizacji pracy chmury informuje sterownik sieci, które z centrów danych posiadają zasoby wystarczające do obsługi żądania. W ten sposób powstaje zbiór  $V_{rDC}$ , który później wykorzystywany jest jako parametr wejściowy dla strategii *anycast*. Ponieważ centralny sterownik sieci rozważa centra danych posiadające wystarczającą ilość zasobów eliminuje to podstawową wadę modelu *overlay*. Żądanie odrzucane jest tylko wtedy gdy sterownik sieci nie jest w stanie znaleźć ścieżki optycznej od źródła do dowolnego  $d \in V_{rDC}$ . W związku z tym oczekuje się istotnej poprawy prawdopodobieństwa blokady względem modelu *overlay*. Należy jednak pamiętać, że decyzje co do wyboru węzła docelowego wynikają ze stanu topologii sieciowej, podczas gdy rozkład zasobów dostępnych w centrach danych ma znaczenie drugoplanowe.

---

#### Algorytm 8 Model *augmented* [6]

---

```

Input:  $CR$ 
1:  $V_{rDC} \leftarrow \emptyset$ 
2: for all  $d \in V_{DC}$  do
3:   if  $r_d \geq LR_r$  then
4:      $V_{rDC} \leftarrow V_{rDC} \cup \{d\}$ 
5:   end if
6: end for
7: SDN ( $LR, V_{rDC}$ )
8:    $d \leftarrow arwa(s, V_{rDC})$ 
9: end SDN
10: if  $d \neq null$  then
11:   Zarezerwuj zasoby w  $d$ 
12:    $r_d \leftarrow r_d - LR_r$ 
13: else
14:   Odrzuć żądanie
15: end if

```

---

### 2.3.3 Model *peer*

Algorytm 9 jest formalnym zapisem modelu *peer* zaproponowanego w rozprawie. Narzędzie organizacji pracy chmury dostarcza sterownikowi sieci następujące informacje:  $LR$ ,  $V_{rDC}$ ,  $preferenceMode$ ,  $dc_{th}$ ,  $C_{DC}$  i  $R_{DC}$ . Dzięki temu sterownik sieci programowalnej posiada informację o tym, które centra danych mają wystarczającą ilość zasobów do obsługi żądania ( $V_{rDC}$ ), preferencje operatora chmury ( $preferenceMode$  i  $dc_{th}$ , które zostaną wyjaśnione poniżej) oraz całkowitą ilość zasobów, a także aktualnie dostępną ilość zasobów w każdym centrum danych (odpowiednio  $C_{DC}$  i  $R_{DC}$ ). Sterownik sieci uruchamia strategię *anycast* dla zbioru dopuszczalnych węzłów docelowych  $D = V_{rDC}$  poszukując centrum

danych zdolnego do obsługi żądania. W ten sposób otrzymujemy identyfikator węzła  $d_{real}$  reprezentujący centrum danych, które wybrałby operator sieci przy braku jakichkolwiek preferencji ze strony dostawcy usług.

Celem kolejnych kroków algorytmu jest ograniczenie wydłużenia ścieżki optycznej wywołanego respektowaniem preferencji dostawcy usług. Różnice w długości ścieżki od węzła źródłowego do węzłów  $d \in V_{rDC}$  i  $d_{real}$  są wyznaczane i normalizowane względem średnicy grafu sieci. Te znormalizowane wartości porównywane są z progiem  $net_{th}$  wyrażającym akceptowalne przez operatora sieci wydłużenie ścieżki optycznej. Jeśli próg  $net_{th}$  nie zostanie przekroczony sterownik sieci bierze pod uwagę preferencje operatora chmury wyrażone przez następujące parametry:  $r_d$  (już wyjaśnione),  $dc_{th}$  i  $preferenceMode$ . Parametr  $dc_{th}$  jest wyrażony jako ułamek całkowitej ilości zasobów obliczeniowych ( $c_d$ ). Parametr  $preferenceMode$  może przyjmować jedną z dwóch wartości : *mostOccupied* lub *leastOccupied*. Wartość *mostOccupied* oznacza preferencję centrów danych posiadających mniej zasobów niż zadany próg. Oznacza to, że  $d \in V_{DC}$  musi spełnić zależność  $r_d \leq dc_{th} \cdot c_d$  aby być preferowane. Analogicznie, wartość *leastOccupied* oznacza preferencję centrów danych posiadających więcej zasobów niż zadany próg, a węzeł  $d \in V_{DC}$  aby uzyskać preferencję musi spełnić zależność:  $r_d \geq dc_{th} \cdot c_d$ .

W efekcie, w przypadku modelu *peer* możemy mówić o ścisłej współpracy pomiędzy operatorem sieci i dostawcą usług. Jeśli tylko próg  $net_{th}$  nie jest przekroczony znaczenie priorytetowe przy podejmowaniu decyzji mają preferencje dostawcy usług. To oznacza, że najmniej lub najbardziej zajęte centrum danych jest wybierane w zależności od wartości parametru  $preferenceMode$ . Niestety od dostawcy usług wymagane jest aby ujawnił informacje dotyczące jego preferencji oraz zasobów dostępnych w każdym z centrów danych.

**Algorytm 9** Model *peer* [6]

---

```

Input:  $CR$ 
1:  $V_{rDC} \leftarrow \emptyset$ 
2: for all  $d \in V_{DC}$  do
3:   if  $r_d \geq LR_r$  then
4:      $V_{rDC} \leftarrow V_{rDC} \cup \{d\}$ 
5:   end if
6: end for
7: SDN ( $LR, V_{rDC}, preferenceMode, dc_{th}, C_{DC}, R_{DC}$ )
8:    $d \leftarrow null$ 
9:    $d_{real} \leftarrow arwa(s, V_{rDC})$  ▷ Brak preferencji
10:  if  $d_{real} \neq null$  then
11:     $dist_{real} \leftarrow LAC(s, d_{real})$ 
12:    if  $arwa = closestGreenWithPenalty$  AND  $d_{real} \in V_{bDC}$  then
13:       $dist_{real} \leftarrow dist_{real} \cdot Penalty$  ▷ Odległości do węzłów brązowych obciążone są karą
14:    end if
15:    for all  $i \in V_{rDC}$  do
16:       $dist_{tmp} \leftarrow LAC(s, i)$ 
17:      if  $arwa = closestGreenWithPenalty$  AND  $i \in V_{bDC}$  then
18:         $dist_{tmp} \leftarrow dist_{tmp} \cdot Penalty$  ▷ Odległości do węzłów brązowych obciążone są karą
19:      end if
20:      if  $(dist_{tmp} - dist_{real}) / diameter \leq net_{th}$  then
21:        if  $arwa = closestOR$  ▷ Jedynie zielone DC mogą wykluczyć zielone DC
22:           $d_{real} \in V_{bDC}$  OR
23:           $(d_{real} \in V_{gDC}$  AND  $i \in V_{gDC})$  then
24:            if  $preferenceMode = mostOccupied$  then
25:              if  $r_d \leq dc_{th} \cdot c_d$  then
26:                if  $d = null$  OR  $r_i \leq r_d$  then
27:                   $d \leftarrow i$ 
28:                end if
29:              end if
30:            end if
31:            if  $preferenceMode = leastOccupied$  then
32:              if  $r_d \geq dc_{th} \cdot c_d$  then
33:                if  $d = null$  OR  $r_i \geq r_d$  then
34:                   $d \leftarrow i$ 
35:                end if
36:              end if
37:            end if
38:          end if
39:        end if
40:      end for
41:      if  $d = null$  then
42:         $d \leftarrow d_{real}$ 
43:      end if
44:    end if
45:  end SDN
46: if  $d \neq null$  then
47:   Zarezerwuj zasoby w  $d$ 
48:    $r_d \leftarrow r_d - LR_r$ 
49: else
50:   Odrzuć żądanie
51: end if

```

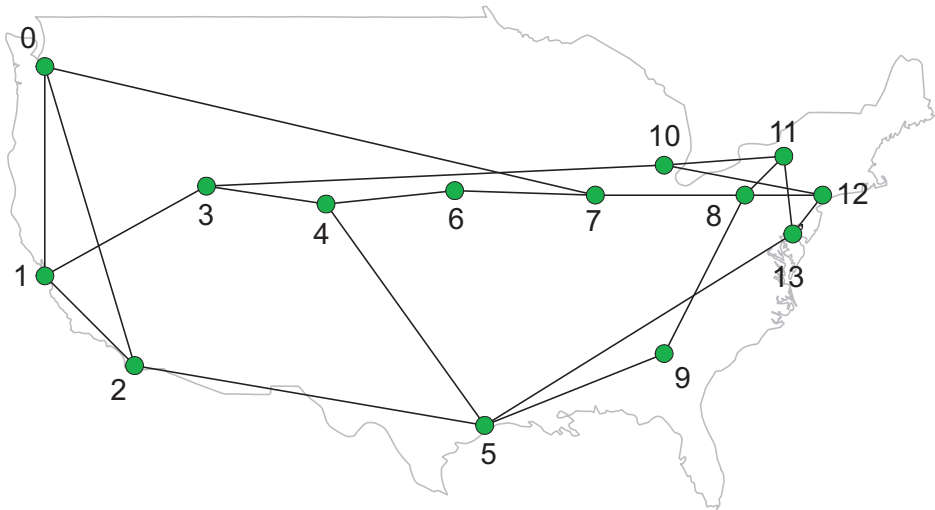
---



Do oceny przedstawionych mechanizmów użyto komputerowej symulacji zdarzeń dyskretnych. Celem przeprowadzonych symulacji była statystycznie wiarygodna ocena prawdopodobieństwa blokady zgłoszeń oraz poziom emisji dwutlenku węgla przez centra danych. Całkowite prawdopodobieństwo blokady wyliczane jest jako stosunek wszystkich odrzuconych żądań do wszystkich żądań napływających do sieci. Obejmuje więc zarówno żądania *unicast* jak i *anycast*. Wskaźnikiem wyrażającym emisję dwutlenku węgla jest stosunek mocy pobranej z nieodnawialnych źródeł energii do ilości ruchu przełączonego we wszystkich centrach danych *brązowe kW/(Gb/s)*. Porównywanie wyników osiąganych przy różnych prawdopodobieństwach blokady jest miarodajne dzięki wprowadzonej normalizacji. Na potrzeby oszacowania emisji dwutlenku węgla założono, że jedynie *brązowe* centra danych mają udział w tym wskaźniku, wydzielając  $CO_2$  proporcjonalnie do zużytej energii. Założono jednocześnie, że *zielone* centra danych nie emitują dwutlenku węgla. [12]. Energia elektryczna pobierana przez węzły sieciowe zawsze pochodzi ze źródeł nieodnawialnych. Przyjęto jednak, że pobór energii przez przełącznice optyczne jest pomijalnie mały w porównaniu do zapotrzebowania energetycznego dużych centrów danych [11].

Do badań użyto symulacji przedziałowej z nienakładającymi się przedziałami, co zmniejsza narzut obliczeniowy związany z fazą robiegową symulacji. Do obliczeń przedziałów ufności pokazanych na wykresach założono poziom ufności 95%. Szczegółowy opis przeprowadzonej oceny wiarygodności statystycznej, metod gromadzenia danych i sposobu prowadzenia symulacji znajduje się w rozprawie.

Środowisko symulacyjne zostało oparte na symulatorze OMNeT++ [15]. Kod związany z symulacją środowiska chmury został samodzielnie napisany przez autora w językach C++ oraz NED. Obliczenia statystyczne były prowadzone w pakiecie statystycznym R [14]. W celu prezentacji wyników wykorzystano bibliotekę *ggplots* [13]. Przygotowane przez autora skrypty języka powłoki posłużyły do



Rys. 3.1: The topology of the NSF network

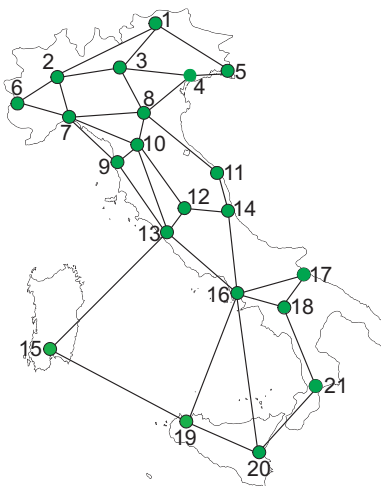
automatycznego uruchamiania wielu scenariuszy symulacyjnych ze zmiennymi parametrami wejściowymi, a także zapisywania wyników w hierarchicznej strukturze katalogów oraz pogładowej prezentacji wyników.

Symulacje zostały przeprowadzone w dwóch sieciach referencyjnych: 14-węzłowej sieci NSF pokazanej na Rys. 3.1 oraz 21-węzłowej sieci *ItalyNet* pokazanej na Rys. 3.2. W obu sieciach założono, że łącze składa się z dwóch światłowodów przenoszących dane w przeciwnych kierunkach a każdy światłowód przenosi 80 długości fali.

W sieci NSF założono istnienie pięciu centrów danych, których lokalizacja została wybrana na podstawie wyników zaprezentowanych w [8] ( $V_{DC} \in \{2, 4, 7, 9, 11\}$ ). W kolejnych scenariuszach symulacyjnych inne pary centrów danych były uznawane za *zielone*. Analogicznie, w sieci *ItalyNet* założono istnienie sześciu centrów danych, tym razem przy wyborze lokalizacji kierowano się wynikami przedstawionymi w [9] ( $V_{DC} \in \{1, 7, 11, 13, 15, 20\}$ ). Trzy spośród centrów danych w kolejnych scenariuszach symulacyjnych uznawane były za *zielone*.

Na ruch oferowany składał się ruch tła (żądania typu *unicast*) oraz ruch kierowany do centrów danych (żądania typu *anycast*). Do opisu czasu życia ścieżek optycznych użyto rozkładu wykładniczego o średniej zależnej od rodzaju ruchu. Dla ruchu tła wynosiła ona 10 sekund, podobnie jak dla żądań związanych z usługami *PaaS* i *StaaS*. Dla żądań związanych z usługą *SaaS* ten parametr był zmienny i przyjmował wartości 100, 360 i 1800 sekund. W dalszych rozważaniach dotyczących intensywności napływu żądań przyjęto, że średni czas życia żądania





Rys. 3.2: The topology of the *ItalyNet*

*SaaS* wynosi 360 sekund. Średnia intensywność napływu żądań w sieci NSF wynosi: 0,3795 (ruch tła), 1,0956 (usługi *PaaS* i *StaaS*) oraz 0,0297 (usługa *SaaS*) zgłoszenia na sekundę. Z kolei w sieci *ItalyNet* średnia intensywność żądań wynosi: 0,132 (ruch tła), 1,1778 (usługi *PaaS* i *StaaS*) oraz 0,0319 (usługa *SaaS*) zgłoszenia na sekundę. Dla potrzeb analizy zachowania sieci przy różnym obciążeniu skalowano ruch *anycast* i w odniesieniu do tego parametru prezentowano znaczną część wyników. W przedstawionych rozważaniach założono, że poszczególne typy usług uczestniczą w ruchu *anycast* w stosunku 1:1:1 (*PaaS*:*StaaS*:*SaaS*). Przebadało także dwie inne wersje tych proporcji: 1:2:4 and 1:5:25. Efekt ten osiągnięto modyfikując średnią intensywność żądań odpowiadającą poszczególnym typom usług.



W niniejszym rozdziale zawarto i omówiono przykładowe wyniki, których kompletny zestaw zaprezentowano w rozprawie. Dla każdego analizowanego mechanizmu ograniczono się do jednego scenariusza symulacyjnego w sieci NSF. Przez scenariusz rozumiany jest wybór określonych centrów danych jako *zielone*, a także zadany średni czas trwania żądania *SaaS*, pobór energii przez żądanie *SaaS* oraz stosunek w jakim żądania odpowiadające różnym typom usług uczestniczą w ruchu *anycast*. Pominięto także szczegółowe rozważania dotyczące doboru parametrów wejściowych dla badanych mechanizmów (*penalty*, *net<sub>th</sub>*, *dc<sub>th</sub>* i *preferenceMode*) oraz ocenę jakości działania mechanizmów w zależności od umiejscowienia *zielonych* centrów danych. Pełny zestaw osiągniętych rezultatów wraz ze szczegółową analizą znajduje się w rozprawie.

W kolejnych sekcjach dokonano analizy strategii *anycast*, schematów dopasowywania oraz modeli współpracy. Wszystkie zaproponowane mechanizmy porównywano z podejściami referencyjnymi pod kątem emisji dwutlenku węgla oraz wpływu na wydajność infrastruktury. Zastosowano podejście przyrostowe, w którym mechanizmy składowe oceniane są przy dodatkowych założeniach upraszczających. Pozwala to wyciągnąć jasne i miarodajne wnioski co do zasady działania tych mechanizmów. Rozszerzając proponowane rozwiązanie o kolejne mechanizmy, równocześnie znoszone są kolejne uproszczenia. W szczególności, ocenę modeli współpracy należy potraktować jako ocenę kompletnego rozwiązania spełniającego wszystkie wymagania i przeprowadzoną bez dodatkowych założeń upraszczających.

### 4.1 Ocena strategii routingu

Podczas oceny strategii routingu wprowadzono dwa założenia upraszczające. Centrum danych przyznano nieograniczoną ilość zasobów obliczeniowych, a jedyną usługą oferowaną przez infrastrukturę chmury była usługa *PaaS*. Oczekuje się, że

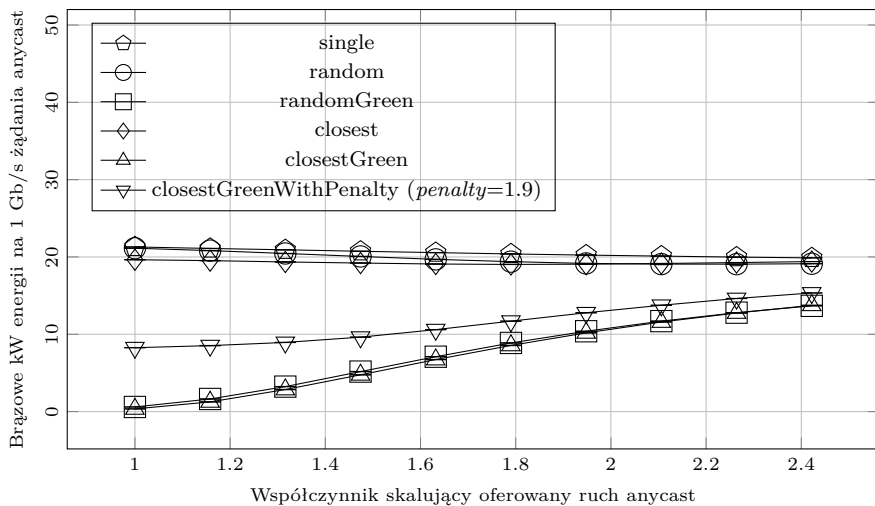
każda z zaproponowanych strategii obniży prawdopodobieństwo blokady względem strategii *single*. Na szczególną uwagę, w kontekście prawdopodobieństwa blokady, zasługuje strategia *closest* zapewniająca dla żądań *anycast* najkrótszą dostępną ścieżkę optyczną minimalizując w ten sposób zajętość zasobów sieciowych.

Rys. 4.1 przedstawia redukcję emisji dwutlenku węgla osiągniętą przez analizowane strategie. Strategie *randomGreen* i *closestGreen* ściśle preferują *zielone* centra danych względem *brązowych*. Oferują dzięki temu największą redukcję emisji  $CO_2$ . Strategia *closestGreenWithPenalty* osiąga mniejszą redukcję niż *randomGreen* i *closestGreen*, z uwagi na słabszą preferencję węzłów *zielonych*. Jednakże, w porównaniu do strategii referencyjnych poprawa jest wysoce zadowalająca. Wyniki osiągane przez strategie referencyjne, *single*, *closest* i *random* są mocno zależne od lokalizacji węzłów *zielonych* w sieci. Wynika to faktu, że wzajemne położenie *zielonych* i *brązowych* centrów danych ma wpływ na kierowanie żądań do tych centrów.

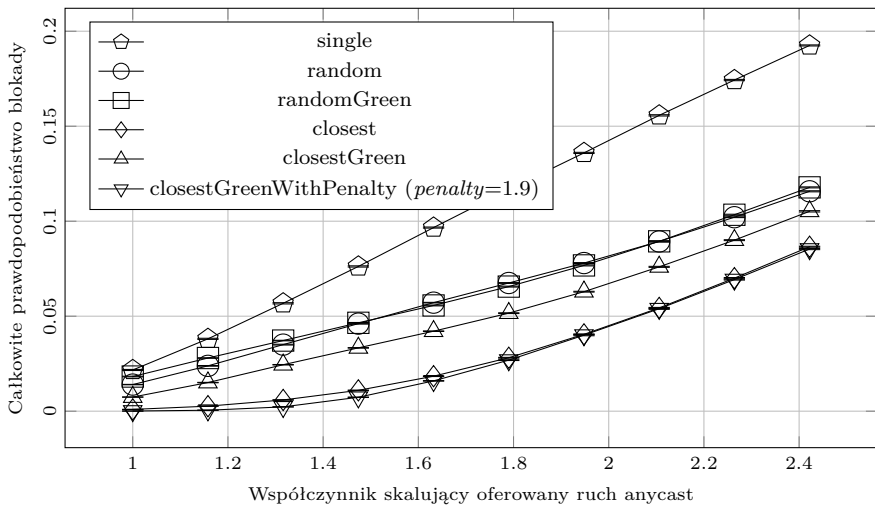
Rys. 4.2 przedstawia całkowite prawdopodobieństwo blokady. Wszystkie *zielone* strategie osiągają istotną poprawę względem strategii *single*, która osiąga najgorsze wyniki także spośród strategii referencyjnych. Wyniki porównania pomiędzy strategiami *random* i *randomGreen* są zależne od umiejscowienia węzłów *zielonych* w sieci. Natomiast porównując strategie *closest* i *closestGreen* można wnioskować, że wykorzystanie strategii *closestGreen* będzie zawsze skutkowało zwiększeniem prawdopodobieństwa blokady. Jednakże, we wszystkich scenariuszach symulacyjnych wyniki osiągane przez strategię *closestGreen* są lepsze od tych osiągniętych przez strategię *randomGreen*.

Najbardziej interesujące wyniki osiąga strategia *closestGreenWithPenalty*, która niezależnie od scenariusza symulacyjnego osiąga najniższe prawdopodobieństwo blokady spośród trzech zaproponowanych strategii. Co więcej, strategia *closestGreenWithPenalty*, w niektórych przypadkach, obniża prawdopodobieństwo blokady także względem strategii *closest*. Wynika to z faktu umiejscowienia *zielonych* centrów danych w obszarach sieci, które sprzyjają równomiernemu rozłożeniu ruchu. Dodatkowo, strategia *closestGreenWithPenalty* nie próbuje za wszelką cenę kierować ruchu do *zielonych* centrów danych jeśli są one odległe.

Podsumowując należy zaznaczyć, że wszystkie zaproponowane strategie mogą zostać z powodzeniem zaimplementowane w centralnym sterowniku sieci programowalnej. Do efektywnego działania wymagają bowiem niewielkiej ilości dodatkowych danych wejściowych. Wyniki przedstawione w rozprawie pozwalają stwierdzić, że wszystkie trzy propozycje mogą skutecznie obniżyć emisję dwutlenku węgla w porównaniu do strategii referencyjnych oraz poprawić prawdopodobieństwo blokady względem strategii *single*. Strategia *closestGreenWithPenalty* zapewnia ponadto prawdopodobieństwo blokady porównywalne ze strategią *closest*. Parametr *penalty* pozwala strategii *closestGreenWithPenalty* dodatkowo równoważyć nacisk pomiędzy uzyskiwaną redukcją emisji dwutlenku węgla, a negatywnym wpływem na



Rys. 4.1: Brązowe kilowaty potrzebne do obsługi 1 Gb/s żądania anycast w sieci NSF przy  $V_{gDC} \in \{4, 11\}$



Rys. 4.2: Całkowite prawdopodobieństwo blokady dla dwóch rodzajów ruchu w sieci NSF przy  $V_{gDC} \in \{4, 11\}$

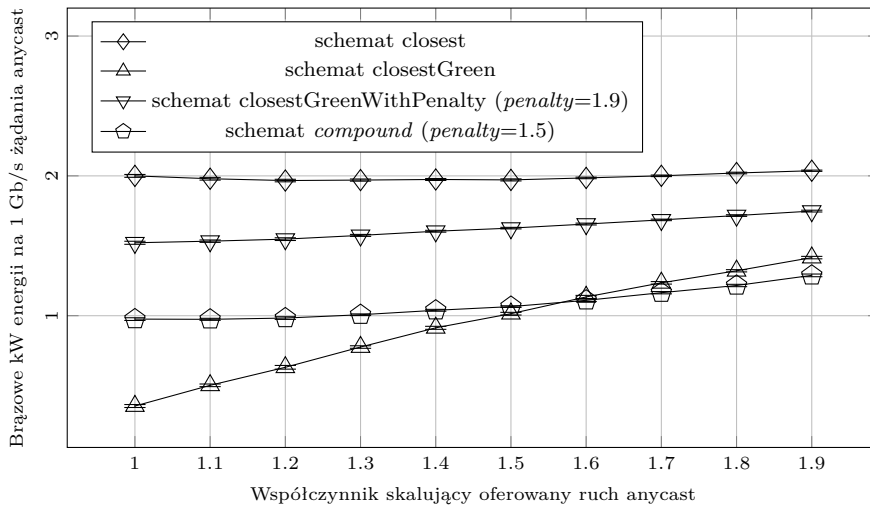
wydajność sieci. Strategie *anycast* zostały przeanalizowane jako część rozwiązania proponowanego w rozprawie. Pozwalają one na wybór docelowego centrum danych z uwzględnieniem dostępności energii pochodzącej ze źródeł odnawialnych.

## 4.2 Ocena schematów dopasowywania

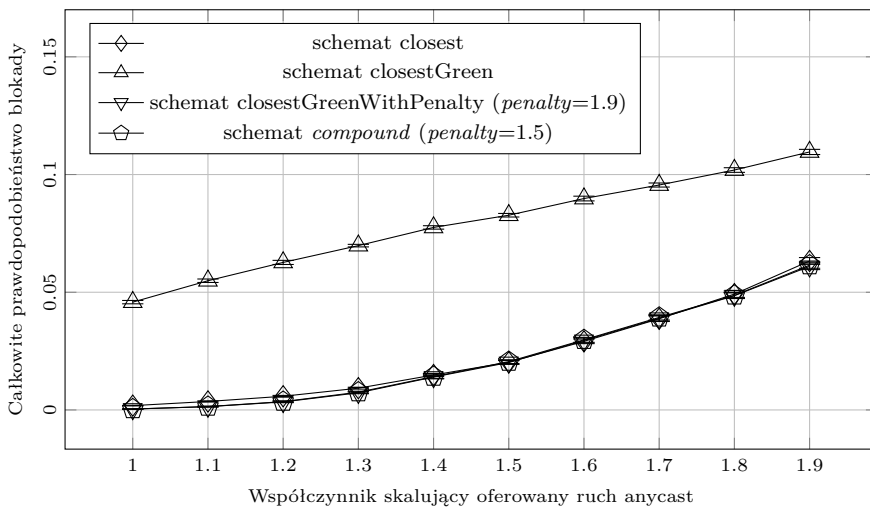
Podczas oceny schematów dopasowywania zniesiono jedno z założeń upraszczających. Mianowicie, wszystkie trzy typy usług są oferowane przez infrastrukturę chmury. W mocy pozostało natomiast założenie dotyczące nieograniczonych zasobów obliczeniowych w centrach danych.

Rys. 4.3 przedstawia redukcję emisji dwutlenku węgla w przykładowym scenariuszu symulacyjnym, podczas gdy Rys. 4.4 prezentuje całkowite prawdopodobieństwo blokady dla tego scenariusza. Schemat dopasowywania *closest* niesie największą emisję dwutlenku węgla spośród wszystkich analizowanych schematów. Jest to wynik braku jakiegokolwiek priorytetyzacji *zielonych* centrów danych względem *brązowych*. Jednocześnie, prawdopodobieństwo blokady zapewniane przez schemat *closest* powinno być traktowane jako punkt odniesienia, ponieważ schemat ten zapewnia przenoszenie całego ruchu w sieci najkrótszymi dostępnymi ścieżkami. Schemat *closestGreen* wprowadza ścisłą preferencję *zielonych* centrów danych dla każdego typu usług. Skutkuje to znaczącą redukcją emisji dwutlenku węgla w porównaniu do schematu *closest*. Ten zysk jest jednak okupiony nieakceptowalną degradacją wydajności sieci, która wynika z faktu, że wszystkie żądania *anycast* kierowane są w pierwszej kolejności do *zielonych* centrów danych, nawet jeśli węzły te są odległe od węzła źródłowego. W niektórych scenariuszach symulacyjnych żądania o niskim zapotrzebowaniu na energię mogą zajmować zasoby sieciowe w pobliżu *zielonych* centrów danych. Może to wymusić przeniesienie obsługi żądań o dużym zapotrzebowaniu energetycznym do *brązowych* centrów danych. Skutkuje to nieefektywnym wykorzystaniem energii pochodzącej ze źródeł odnawialnych, w szczególności w odniesieniu do schematu *compound*, który dopasowuje poziom preferencji węzłów *zielonych* do różnych typów usług. Schemat *closestGreenWithPenalty* osiąga gorsze wyniki w redukcji emisji dwutlenku węgla niż schemat *closestGreen* z uwagi na słabszą preferencję węzłów *zielonych*. Z drugiej strony schemat *closestGreenWithPenalty* utrzymuje prawdopodobieństwo blokady na poziomie porównywalnym ze schematem *closest*.

Zaproponowany schemat *compound* znacznie obniża emisję dwutlenku węgla w porównaniu do schematu *closest* utrzymując jednocześnie porównywalne prawdopodobieństwo blokady. Jest to możliwe dzięki ważeniu stopnia preferencji *zielonych* centrów danych dla żądań typu *SaaS* i ścisłej preferencji węzłów najbliższych dla żądań typu *StaaS*. Ponadto schemat *compound*, w niektórych scenariuszach symulacyjnych, zapewnia równocześnie najniższą emisję dwutlenku węgla oraz prawdopodobieństwo blokady. Dodatkowa poprawa w kontekście



Rys. 4.3: Brązowe kilowaty potrzebne do obsługi 1 Gb/s żądania anycast w sieci NSF przy  $V_{gDC} \in \{2, 9\}$ , proporcjach usług 1:5:25 oraz poborze energii przez żądanie *SaaS* na poziomie 2.7 kW/(Gb/s)



Rys. 4.4: Całkowite prawdopodobieństwo blokady dla dwóch rodzajów ruchu w sieci NSF przy  $V_{gDC} \in \{2, 9\}$  i proporcjach usług 1:5:25

prawdopodobieństwa blokady wywołana jest umiejscowieniem *zielonych* centrów danych w obszarach sieci przełączających mniej ruchu tła. W efekcie osiągamy bardziej równomierne obciążenie sieci co bezpośrednio prowadzi do obniżenia prawdopodobieństwa blokady.

Wyniki przedstawione w rozprawie dowodzą, że zaproponowany schemat dopasowywania *compound* pozwala istotnie obniżyć ilość dwutlenku węgla emitowanego przez centra danych połączone siecią optyczną bez istotnego pogorszenia wydajności pracy tej sieci. Ponadto schemat *compound* jest w stanie elastycznie dopasować się do aktualnych warunków w sieci, a dzięki małemu zapotrzebowaniu na dodatkowe informacje sterujące, może zostać z łatwością zaimplementowany w centralnym sterowniku sieci optycznej. Schematy dopasowywania zostały przeanalizowane jako składowa rozwiązania proponowanego w rozprawie. Głównym zadaniem schematu *compound* jest efektywne wykorzystanie zasobów energii odnawialnej. Cel ten realizowany jest przez odzwierciedlenie właściwości różnych typów usług w sposobie obsługi żądań odpowiadającym tym usługom.

### 4.3 Ocena modeli współpracy

Podczas oceny modeli współpracy zniesiono wszystkie założenia upraszczające. Oznacza to, że wszystkie trzy typy usług są oferowane przez infrastrukturę chmury podczas gdy zasoby obliczeniowe dostępne w centrach danych zostały ograniczone. Aby obsłużyć żądanie związane z usługą w chmurze konieczne jest zapewnienie zarówno zasobów sieciowych jak i obliczeniowych. Należy zwrócić uwagę, że modele współpracy wykorzystują schematy dopasowywania strategii *anycast* do typów usług. Analizie w tym przypadku podlega więc kompletne rozwiązanie proponowane w rozprawie. Efektywność tego rozwiązania jest zależna od wszystkich elementów składowych. Należy także zaznaczyć, że w celu miarodajnego porównania modeli współpracy należy zdefiniować schematy dopasowywania z jakim poszczególne modele są zintegrowane. Szczegółowe rozważania na ten temat, wraz z wynikami symulacji, zostały przedstawione w rozprawie. Na podstawie tych rozważań przyjęto następujące pary: model *overlay* wykorzystuje schemat *closest*, podczas gdy modele *augmented* oraz *peer* wykorzystują schemat *compound*.

Rys. 4.5 przedstawia redukcję dwutlenku węgla emitowanego przez centra danych w przykładowym scenariuszu symulacyjnym podczas gdy Rys. 4.6 prezentuje całkowite prawdopodobieństwo blokady dla tego scenariusza. Model *overlay* niesie ze sobą największe prawdopodobieństwo blokady spośród wszystkich analizowanych modeli. Jest to bezpośrednia konsekwencja najniższego poziomu współpracy pomiędzy operatorem sieci i dostawcą usług. Docelowe centrum danych jest bowiem wybierane jedynie w oparciu o dostępność zasobów sieciowych podczas gdy zasoby obliczeniowe są pomijane. Może to skutkować wyborem centrum danych, które nie posiada wystarczających zasobów obliczeniowych do obsłużenia żądania.

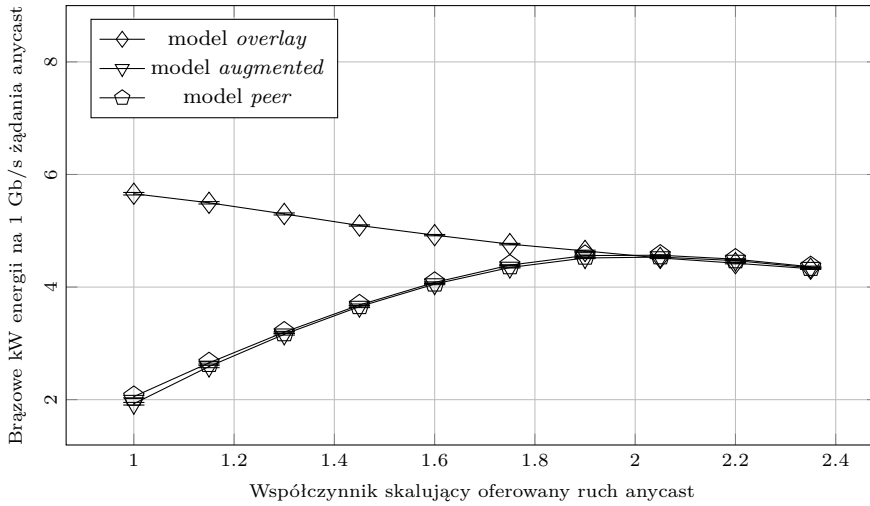


Przy zastosowaniu modelu *overlay* może dojść do odrzucenia żądań pomimo, że istnieją w sieci ścieżki do centrów danych posiadających wystarczającą ilość zasobów obliczeniowych. Model *overlay* w wielu scenariuszach testów osiąga też najgorsze wyniki w kontekście emisji dwutlenku węgla, gdyż wykorzystanie schematu *closest* nie wprowadza żadnej preferencji *zielonych* centrów danych względem *brązowych*. Wykorzystanie tego schematu jest uzasadnione z uwagi na zakładany brak współpracy pomiędzy podmiotami.

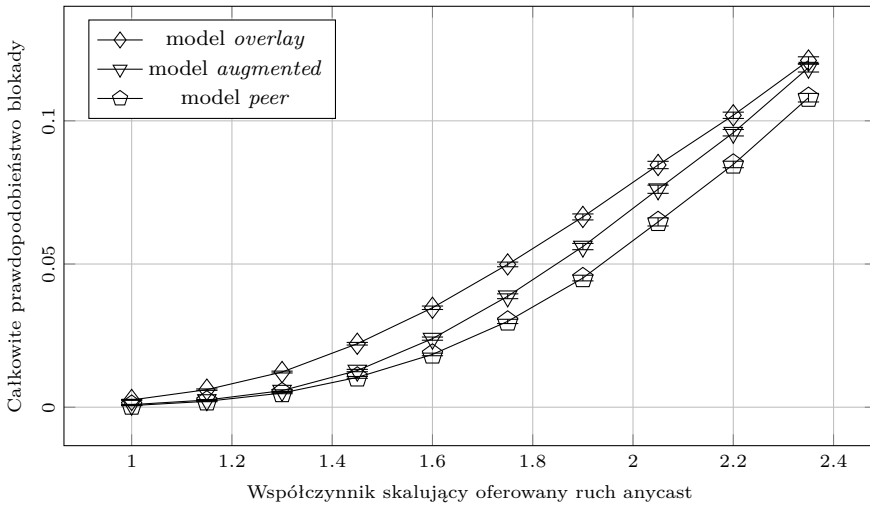
Zastosowanie modelu *augmented* prowadzi do obniżenia prawdopodobieństwa blokady względem modelu *overlay*. W tym modelu centralny sterownik sieci próbuje kierować żądania jedynie do tych centrów danych, które posiadają zasoby wystarczające do obsługi żądania. Takie podejście eliminuje główną wadę modelu *overlay* polegającą na niewykorzystywaniu dostępnych zasobów. Jednakże, w niektórych scenariuszach symulacyjnych model *overlay* zapewnia mniejsze prawdopodobieństwo blokady niż model *augmented*, w szczególności dla dużych wartości ruchu w sieci. Dzieje się tak dlatego, że model *augmented* próbuje kierować żądania do centrów danych z wystarczającą ilością zasobów obliczeniowych, nawet jeśli te centra danych są odległe. Prowadzi to do obniżenia prawdopodobieństwa blokady ruchu *anycast* ale dzieje się to kosztem zwiększenia zajętości zasobów sieciowych co prowadzi do zwiększonego prawdopodobieństwa blokady ruchu tła, ponieważ na całkowite prawdopodobieństwo blokady składają się prawdopodobieństwa blokady ruchu tła oraz ruchu *anycast*.

Zaproponowany model współpracy *peer* zapewnia najniższe prawdopodobieństwo blokady we wszystkich badanych scenariuszach symulacyjnych. Dodatkowe informacje dostarczane przez narzędzie organizacji pracy chmury ograniczają liczbę sytuacji, w których zasoby obliczeniowe pozostają nieosiągalne z uwagi na zajętość sieci. Model *peer* zapewnia, dzięki temu, efektywne wykorzystanie zasobów obliczeniowych, przy jednoczesnym obniżeniu emisji dwutlenku węgla do poziomu osiąganego przez model *augmented*.

Modele współpracy przedstawione w rozprawie są dobrze umiejscowione w architekturze chmury zakładającej współistnienie operatora sieci i dostawcy usług. Ponadto interfejsy wymiany danych pomiędzy podmiotami są jasno zdefiniowane co zwiększa potencjał wykorzystania tych modeli w praktyce. Wyniki zaprezentowane w rozprawie pozwalają stwierdzić, że zaproponowany model *peer* skutecznie podnosi wydajność pracy infrastruktury, a dzięki połączeniu z *zielonymi* strategiami *anycast* i schematem *compound* prowadzi także do istotnego obniżenia emisji dwutlenku węgla. Modele współpracy jako składowa kompletnego rozwiązania zwiększają jego potencjał wdrożeniowy w nowoczesnej architekturze chmury.



Rys. 4.5: Brązowe kilowaty potrzebne do obsługi 1 Gb/s żądania anycast w sieci NSF przy  $V_{gDC} \in \{9, 11\}$ , proporcjach usług 1:2:4 oraz poborze energii przez żądanie *SaaS* na poziomie 5.4 kW/(Gb/s)



Rys. 4.6: Całkowite prawdopodobieństwo blokady dla dwóch rodzajów ruchu w sieci NSF przy  $V_{gDC} \in \{9, 11\}$ , proporcjach usług 1:2:4 oraz poborze energii przez żądanie *SaaS* na poziomie 5.4 kW/(Gb/s)

Celem rozprawy była ocena możliwości obniżenia ilości dwutlenku węgla emitowanego przez infrastrukturę chmury obliczeniowej złożonej z centrów danych połączonych siecią optyczną. W rozprawie zaproponowano rozwiązanie przydziału zasobów na żądanie z uwzględnieniem źródła pochodzenia energii. Rozwiązanie składa się z trzech mechanizmów składowych, których zadaniem jest spełnienie wymagań narzucanych przez architekturę i właściwości chmury obliczeniowej. Strategie umożliwiają wybór docelowego centrum danych w sposób uwzględniający dostępność energii odnawialnej oraz aktualne warunki panujące w sieci. Schematy dopasowywania tych strategii do różnych typów usług pozwalają na efektywne wykorzystanie energii ze źródeł odnawialnych. Całości dopełniają modele współpracy pomiędzy operatorem sieci i dostawcą usług w chmurze. Modele te umożliwiają współpracę oddzielnych organizacji celem poprawy jakości świadczenia usług w chmurze.

Zaproponowane rozwiązanie łączące trzy wspomniane mechanizmy zostało poddane ocenie w zaimplementowanym środowisku symulacyjnym opartym na symulatorze OMNeT++. Symulacje przeprowadzono w dwóch sieciach referencyjnych NSF oraz *ItalyNet*. Badania przeprowadzono w wielu scenariuszach symulacyjnych modyfikując parametry wejściowe symulacji.

Ponieważ zaproponowane rozwiązanie złożone było z trzech składowych zdecydowano się aby dla właściwej oceny własności mechanizmów oraz zależności pomiędzy nimi zastosować podejście przyrostowe. Kolejne mechanizmy były dodawane do kompletnego rozwiązania, a po każdym kroku przeprowadzany był szereg badań. Takie podejście pozwoliło nie tylko sformułować wartościowe wnioski o każdym z mechanizmów ale także w pełni zrozumieć właściwości kompletnego rozwiązania. Przedstawione wyniki odnoszą się do prawdopodobieństwa blokady napływających żądań oraz emisji dwutlenku węgla związanego ze świadczeniem usług w chmurze obliczeniowej.

W wyniku przeprowadzonych badań dowiedziono, że jedna z kombinacji mechanizmów składowych pozwala osiągnąć zdecydowanie lepsze wyniki niż pozostałe podejścia. W tej kombinacji wykorzystano schemat dopasowywania *compound*, co oznacza obsługę żądań *PaaS*, *StaaS* i *SaaS* odpowiednio przez strategie *closestGreen*, *closest* i *closestGreenWithPenalty*. Schemat *compound* połączono z modelem współpracy *peer*. Taka kombinacja pozwala na redukcję emisji dwutlenku węgla bez istotnego pogorszenia wydajności sieci. Ponadto parametr *penalty* wykorzystywany przez strategię *closestGreenWithPenalty* oraz parametry  $dc_{th}$ ,  $net_{th}$  i *preferenceMode* stosowane w modelu *peer* pozwalają, w elastyczny sposób, dopasować się do zmiennych warunków pracy infrastruktury. W rozprawie przedstawiono wyniki dla różnych wartości tych parametrów aby przeanalizować ich wpływ na działanie mechanizmów.

Zaproponowane rozwiązanie powinno zostać przeanalizowane także pod kątem potencjalnych wdrożeń. Wszystkie mechanizmy są dobrze umiejscowione w nowoczesnej architekturze chmury obliczeniowej. Strategie *anycast* mogą zostać zaimplementowane w centralnym sterowniku sieci programowalnych. Podobnie schemat *compound*, który wykorzystywałby ujednolicony opis usług w jednym z powszechnie używanych języków opisu danych (np. XML lub JSON). Także wdrożenie modeli współpracy jako rozszerzenia do narzędzi organizacji pracy chmury jest uzasadnione. Komunikacja pomiędzy sterownikiem sieci programowalnej a wspomnianymi narzędziami organizacji pracy chmury byłaby definiowana przez tzw. północny interfejs programistyczny (northbound API) zdefiniowany na potrzeby komunikacji sterownika z aplikacjami.

## 5.1 Najważniejsze osiągnięcia

Najważniejsze osiągnięcia pracy można streścić w następujących punktach:

1. W rozprawie określono nowoczesną architekturę chmury, w której zasoby IT oraz zasoby sieciowe są zarządzane przez osobne organizacje. W związku z tym modele sieci optycznej i zasobów IT zostały omówione wraz z modelami trzech typów usług świadczonych w chmurze. Przepływ żądań zakładany w rozprawie odzwierciedla charakter chmury, gdzie żądania obsługiwane są po zgłoszeniu przez użytkownika.
2. Opierając się na danych dotyczących poboru energii przez centra danych oraz ich udziału w globalnej emisji dwutlenku węgla, przedstawiono motywację dla poszukiwania rozwiązań minimalizujących negatywny wpływ sektora ICT na środowisko naturalne.
3. Sformułowano problem dynamicznego routingu typu *jeden do jednego z wielu*

w sieciach optycznych połączony z problemem przydziału zasobów chmury obliczeniowej z uwzględnieniem źródła pochodzenia energii elektrycznej.

4. W rozprawie zdefiniowano wymagania jakie nakłada architektura chmury i sposób świadczenia przez nią usług na rozwiązanie zadanego problemu badawczego.
5. Zaproponowano trzy strategie routingu typu *jeden do jednego z wielu*: *randomGreen*, *closestGreen* i *closestGreenWithPenalty*. Strategie te zostały szczegółowo przebadane oraz porównane z rozwiązaniami referencyjnymi.
6. Zaproponowano schemat *compound* dopasowujący różne strategie *anycast* do różnych typów usług oferowanych w chmurze. Schemat ten porównano z rozwiązaniami, w których wszystkie usługi obsługiwane są przez tę samą strategię *anycast*.
7. Przedstawiono modele współpracy pomiędzy operatorem sieci a operatorem infrastruktury IT. Dwa z tych modeli, *overlay* i *augmented*, są formalnie opisanymi modelami intuicyjnymi. Zaproponowany model *peer* został porównany z tymi modelami.
8. Zaproponowano kompletne rozwiązanie sformułowanego problemu badawczego z uwzględnieniem opisanych wymagań. Rozwiązanie to jest wyjądną kombinacją mechanizmów składowych: strategii routingu typu *anycast*, schematów dopasowywania oraz modeli współpracy.
9. Korzystając z symulatora OMNeT++ zbudowano narzędzie symulacyjne.
10. Przeprowadzono szczegółowe badania i dokonano oceny wszystkich mechanizmów składowych. Podejście przyrostowe do tworzenia kompletnego rozwiązania ułatwiło zrozumienie zarówno poszczególnych mechanizmów, jak i zależności pomiędzy nimi. W celu dokładnej analizy przeanalizowano wiele scenariuszy symulacyjnych przy zmiennych parametrach wejściowych.

Na podstawie uzyskanych wyników można stwierdzić, że teza wskazana na początku rozprawy: *Możliwa jest redukcja ilości dwutlenku węgla emitowanego przez centra danych tworzące chmurę obliczeniową połączoną siecią optyczną przez zastosowanie strategii rozkładu obciążenia uwzględniającego pochodzenie wykorzystywanej energii* została udowodniona.



---

# Literatura

- [1] D. Alger. *Grow a Greener Data Center*. Cisco Press, Indianapolis, 2010. ISBN 978-1-58705-813-4.
- [2] P. Borylo, A. Lason, J. Rzasa, A. Szymanski, and A. Jajszczyk. Usługi w chmurze korzystające z sieci optycznych o minimalnym zużyciu energii. *Przegląd Telekomunikacyjny, Wiadomości Telekomunikacyjne*, 86(10):1323–1330, 2013.
- [3] P. Borylo, A. Lason, J. Rzasa, A. Szymanski, and A. Jajszczyk. Anycast Routing for Carbon Footprint Reduction in WDM Hybrid Power Networks with Data Centers. In *Proc. IEEE International Conference on Communications (ICC 2014)*, pages 3720–3726, Sydney, Australia, June 2014.
- [4] P. Borylo, A. Lason, J. Rzasa, A. Szymanski, and A. Jajszczyk. Assessment of green anycast strategies in hybrid power networks having regular topologies. *Przegląd Telekomunikacyjny, Wiadomości Telekomunikacyjne*, 83(8-9):748–754, 2014.
- [5] P. Borylo, A. Lason, J. Rzasa, A. Szymanski, and A. Jajszczyk. Fitting Green Anycast Strategies to Cloud Services in WDM Hybrid Power Networks. In *Proc. IEEE Global Communications Conference (GLOBECOM 2014)*, pages 2633–2639, Austin, TX, USA, Dec. 2014.
- [6] P. Borylo, A. Lason, J. Rzasa, A. Szymanski, and A. Jajszczyk. Green Cloud Provisioning Throughout Cooperation of a WDM Wide Area Network and a Hybrid Power IT Infrastructure. *Journal of Grid Computing (Springer)*, 1(1), 2015. doi: 10.1007/s10723-015-9354-7.
- [7] C. Develder, M. De Leenheer, B. Dhoedt, M. Pickavet, D. Colle, F. De Turck, and P. Demeester. Optical Networks for Grid and Cloud Computing Applications. *Proceedings of the IEEE*, 100(5):1149–1167, Jan. 2012.

- 
- [8] X. Dong, T. El-Gorashi, and J. M. H. Elmirghani. Green IP Over WDM Networks With Data Centers. *IEEE Journal of Lightwave Technology*, 29(12):1861–1880, June 2011.
- [9] M. Gattulli, M. Tornatore, R. Fiandra, and A. Pattavina. Low-Carbon Routing Algorithms For Cloud Computing Services in IP-over-WDM Networks. In *Proc. IEEE International Conference on Communications (ICC 2012)*, pages 2999–3003, Ottawa, Canada, June 2012.
- [10] M. Gupta and S. Singh. Greening of the Internet. In *Proc. ACM Special Interest Group on Data Communication (SIGCOMM 2003)*, pages 19–26, Karlsruhe, Germany, Aug. 2003.
- [11] W. Van Heddeghem, F. Idzikowski, W. Vereecken, D. Colle, M. Pickavet, and P. Demeester. Power consumption modeling in optical multilayer networks. *Springer Photonic Network Communications*, 24(2):86–102, Oct. 2012.
- [12] F. Kong and X. Liu. A Survey on Green-Energy-Aware Power Management for Datacenters. *ACM Computing Surveys*, 47(2):30:1–30:38, Nov. 2014.
- [13] PGFPlots. PGFPlots – A LaTeX Package to create normal/logarithmic plots in two and three dimensions. <http://pgfplots.sourceforge.net/>. Cited 19 Aug. 2015.
- [14] RProject. R: The R Project for Statistical Computing. <http://www.r-project.org/>. Cited 19 Aug. 2015.
- [15] A. Varga. Using the OMNeT++ Discrete Event Simulation System in Education. *IEEE Transactions on Education*, 42(4):11, Nov. 1997.
- [16] W. Vereecken, W. V. Heddeghem, M. Deruyck, B. Puype, B. Lannoo, W. Joseph, D. Colle, L. Martens, and P. Demeester. Power Consumption in Telecommunication Networks: Overview and Reduction Strategies. *IEEE Communications Magazine*, 49:62–69, June 2011.
- [17] Y. Zhang, P. Chowdhury, M. Tornatore, and B. Mukherjee. Energy Efficiency in Telecom Optical Networks. *IEEE Communications Surveys & Tutorials*, 12(4):441–458, 2010.