Załącznik nr 3
Autoreferat

22 czerwca 2024

1 Imię i nazwisko

Piotr Arabas

2 Uzyskane dyplomy i stopnie naukowe

2004 r. – stopień doktora nauk technicznych w specjalności automatyka uzyskany na Wydziale Elektroniki i Technik Informacyjnych Politechniki Warszawskiej za rozprawę pt. *Hierarchiczna struktura w systemie obrony przeciwrakietowej; mechanizmy decyzyjne i badania symulacyjne*.

1996 r. – tytuł magistra inżyniera w specjalności systemy sterowania i wspomagania decyzji uzyskany w Instytucie Automatyki i Informatyki Stosowanej Politechniki Warszawskiej za pracę pt. *Rozproszona implementacja algorytmu wyznaczania optymalnych sterowań w przypadku prognoz wielowariantowych*.

3 Informacja o dotychczasowym zatrudnieniu w jednostkach naukowych

- **Instytut Automatyki i Informatyki Stosowanej,Wydział Elektroniki i Technik Informacyjnych Politechniki Warszawskiej**

  2002–2004 – asystent,

  2004–obecnie – adiunkt.

- **Naukowa i Akademicka Sieć Komputerowa Państwowy Instytut Badawczy**

  2002–2004 – asystent w Pionie Naukowym,

  2004–obecnie – adiunkt w Centrum Badań i Transferu Technologii.

4 Omówienie osiągnięć

Prezentowane osiągnięcie naukowe to cykl powiązanych tematycznie artykułów naukowych, zgodnie z art. 219 ust. 1 pkt 2 ustawy z dn. 30 lipca 2018 r. Prawo o szkolnictwie wyższym i nauce. Tytuł osiągnięcia naukowego:

Sterowanie przydziałem zasobów w energooszczędnych sieciach i rozproszonych systemach przetwarzania danych

4.1 Wykaz publikacji dotyczących opisywanego osiągnięcia naukowego

[**H1**] **Arabas, P.**, *Modeling and simulation of hierarchical task allocation system for energy-aware HPC clouds*, Simulation Modelling Practice and Theory, Vol. 107, 2021, 102221, DOI:10.1016/j.simpat.2020.102221.

- sformułowanie zadania sterowania energooszczędnym przydziałem zasobów do zadań obliczeniowych systemu chmurowego,
- heurystyczny, iteracyjny algorytm przydziału zasobów,
- **PA: 100%**, **IF21=4,199, 100 pkt.** MNiSW.

[**H2**] **Arabas**, P., Niewiadomska-Szynkiewicz E., *Energy-Efficient Workload Allocation in Distributed HPC System*, w: The 2019 International Conference on High Performance Computing & Simulation HPCS 2019,2020, ss. 470-477, DOI:10.1109/HPCS48598.2019.9188240

- propozycja hierarchicznej struktury energooszczędnego sterowania w jednoskach chmury obliczeniowej wraz z iteracyjnym algorytmem alokacji zasobów i wynikami badań symulacyjnych.
- **PA: 70%**, ENS: 30%, **CORE2018 B, 70 pkt.** MNiSW.

[**H3**] **Arabas P.**, *Energy Aware Data Centers and Networks: a Survey*, Journal of Telecommunications and Information Technology, nr 4/2018, 2018, s. 26-36, DOI:10.26636/jtit.2018.129818

- artykuł przeglądowy, moim autorskim wkładem jest definicja zadania koordynatora w hierarchicznym systemie sterowania chmurą obliczeniową,
- **PA: 100%**, **12 pkt. (obecnie 40 pkt.)** MNiSW.

[**H4**] Niewiadomska-Szynkiewicz, E., Sikora, A, **Arabas, P.**, Kołodziej, J., *Control System for Reducing Energy Consumption in Backbone Computer Network*, Concurrency and Computation – Practice & Experience, Dec, 25, 2013, ss. 1738-1754, DOI:10.1002/cpe.2964

- opis propozycji schematów energooszczędnego sterowania siecią oraz ocena ich złożoności obliczeniowej,
- ENS: 35%, AS: 35%, **PA: 20%**, JK: 10%, **IF13=0,784, 25 pkt. (obecnie 100 pkt.)** MNiSW.

[**H5**] Niewiadomska-Szynkiewicz, E., Sikora, A., **Arabas, P.**, Kamola, M., Mincer, M., Kołodziej, J., *Dynamic Power Management in Energy-aware Computer Networks and data intensive computing systems*, Future Generation Computer Systems – The International Journal of Grid Computing and Escience, Jul, 2014, 37, ss. 284-296, DOI:10.1016/j.future.2013.10.002

- sformułowanie czterech zadań optymalizacji dla energooszczędnego sterowania siecią,
- porównania w środowisku laboratoryjnym dwóch algorytmów energooszczędnego sterowania siecią,
- projekt stanowiska laboratoryjnego,
- ENS: 25%, AS: 25% **PA: 20%**, MK: 10%, MM: 10%, JK: 10%, **IF14=2,786, 40 pkt. (obecnie 140 pkt.)** MNiSW.

[**H6**] **Arabas P.**, Jóźwik T., Niewiadomska-Szynkiewicz, E., *Router Activation Heuristics for Energy-Saving ECMP and Valiant Routing in Data Center Networks*, Energies, 2023, Vol. 16, ss. 1–20, DOI:10.3390/en16104136,

- propozycja heurystyk dla energooszczędnego rutingu w centrach danych i ich ocena eksperymentalna,
- **PA: 50%**, TJ: 40%, ENS: 10%, **IF22=3,2, 140 pkt.** MNiSW.

[**H7**] Karpowicz, M., **Arabas, P.**, Niewiadomska-Szynkiewicz, E., *Energy-Aware Multilevel Control System for a Network of Linux Software Routers: Design and Implementation*, IEEE Systems Journal, 2018, 12, 1, ss. 571-582, DOI:10.1109/JSYST.2015.2489244.

- wyniki eksperymentalnej weryfikacji algorytmów energooszczędnego sterowania ruchem w sieci zbudowanej z ruterów programowych, opis konstrukcji stanowiska pomiarowego,
- MK: 70%, **PA: 20%**, ENS: 10%, **IF18=4,463, 35 pkt. (obecnie 140 pkt.)** MNiSW.

[**H8**] Jaskóła, P.; **Arabas, P.**; Karbowski, A., *Simultaneous Routing and Flow Rate Optimization in Energy-aware Computer Networks*, International Journal of Applied Mathematics and Computer Science, Mar 2016, 26, 1, ss. 231-243, DOI:10.1515/amcs-2016-0016.

- model poboru mocy przez ruter wraz z wynikami identyfikacji na stanowisku pomiarowym,

– wyniki weryfikacji zadania energooszczędnego sterowania siecią uwzględniającego użyteczność przepływów,
– PJ: 45%, **PA: 35%**, AK: 20%, **25 pkt. (obecnie 100 pkt.)** MNiSW.

[**H9**] Karpowicz M., **Arabas P.**, *Server Workload Model Identification: Monitoring and Control Tools for Linux*, Journal of Telecommunications and Information Technology vol. 2, ss. 5-12, 2016.

– propozycja wykorzystania interfejsów IPMI i RAPL do określenia obciążenia procesora i pobieranej przez niego mocy,
– przeprowadzenie eksperymentów pomiarowych,
– MK: 50%, **PA: 50%**, **12 pkt. (obecnie 40 pkt.)** MNiSW.

[**H10**] **Arabas P.**, Karpowicz M., *Wykorzystanie informacji z rejestrów procesora do identyfikacji modelu poboru mocy przez serwer*, Przegląd Elektrotechniczny, nr 3, ss. 34-41, 2016.

– identyfikacja modelu poboru mocy przez system komputerowy pracujący w różnych warunkach,
– **PA: 80%**, MK: 20%, **14 pkt. (obecnie 70 pkt.)** MNiSW.

[**H11**] Karpowicz M., **Arabas P.**, Niewiadomska-Szynkiewicz E., *Design and implementation of energy-aware application-specific CPU frequency governors for the heterogeneous distributed computing systems*, Future Generation Computer Systems – The International Journal of eScience, 2018, vol. 78, ss. 302–315, DOI:10.1016/j.future.2016.05.011,

– modele poboru mocy przez system komputerowy i ich wykorzystanie w konstrukcji sterownika procesora, którego celem jest oszczędzanie energii przy zachowaniu QoS,
– MK: 70%, **PA: 20%**, ENS 10%, **IF18=5,768, 40 pkt. (obecnie 140 pkt.)** MNiSW.

[**H12**] Niewiadomska-Szynkiewicz E., Marks M., **Arabas P.**, Sikora A., *Bezprzewodowe sieci czujników w internecie rzeczy Modele - Algorytmy - Protokoły*, 292 str., PWN, 2022.

– monografia zawierająca m.in. przegląd i klasyfikację algorytmów sterowania oraz analizę podejść i problemów występujących w bezprzewodowych sieciach sensorowych,
– ENS: 30%, MM: 30%, **PA: 20%**, AS: 20%, **80 pkt.** MNiSW.

[**H13**] **Arabas P.**, Sikora A., Szynkiewicz W., *Energy-Aware Activity Control for Wireless Sensing Infrastructure Using Periodic Communication and Mixed-Integer Programming*, Energies, 2021, t.14, s. 1–17, DOI:10.3390/en14164828

– propozycja schematu koordynacji aktywności czujników tworzących sieć bezprzewodową wraz z symulacyjną oceną efektywności,
– **PA: 70%**, AS: 20%, WS: 10%, **IF21=3,252 140 pkt.** MNiSW.

4.2 Wprowadzenie

W przeciągu ostatnich kilkunastu lat wiele uwagi poświęcono ograniczeniu zużycia energii przez sprzęt telekomunikacyjny i obliczeniowy. Zasadnicze znaczenie miał postęp technologiczny w dziedzinie elektroniki pozwalający na budowę bardzo wydajnych, także w sensie energetycznym, urządzeń. Dostrzeżono też potrzebę odpowiedniego zarządzania zużyciem energii, stopniowo wprowadzano coraz bardziej zaawansowane mechanizmy pozwalające na monitorowanie jej zużycia i sterowanie aktywnością urządzeń. Jako kluczowe techniki można wskazać [36]: okresowe usypianie (*smart standby*) oraz skalowanie napięcia i częstotliwości (*dynamic voltage and frequency scaling*). Pierwsza z nich polega na okresowym wyłączaniu części obwodów urządzenia. Oczywiste jest, że im więcej obwodów jest nieaktywne, tym większa oszczędność energii. Niestety z wyłączeniem wiąże się ograniczenie dostępnych funkcji i dłuższy czas potrzebny na ich aktywację. Dobierając napięcie zasilające i częstotliwość taktowania, można ograniczać zużycie energii i dopasować wydajność do zapotrzebowania bez wyłączenia systemu. Powszechnym standardem opisującym mechanizmy z tej grupy jest ACPI [26]. Podobną rolę dla sieci pełni standard IEEE 802.3az [1] dopuszczający okresowe usypianie urządzeń Ethernet. Inne rozwiązania pozwalają na stopniowe ograniczanie wydajności łączy i urządzeń o redundantnej lub hierarchicznej budowie, jak ma to miejsce np. w przełącznikach Mellanox InfiniBand [41].

Wraz z mechanizmami pozwalającymi ograniczać zużycie energii opracowano szereg standardów umożliwiających monitorowanie urządzeń. Dla procesorów firmy Intel rozwiązaniem takim jest RAPL (*Running Average Power Limit*) [27] udostępniający dość precyzyjne estymaty zużycia energii przez poszczególne elementy procesora. Wśród bardziej ogólnych można wskazać standardy PAPI [17] i IPMI [28] pozwalające na nadzorowanie zużycia energii przez serwery.

Wspomniane mechanizmy byłyby bezużyteczne, gdyby nie powstały odpowiednie algorytmy sterujące. Znanym przykładem są, ze względu na implementację o otwartym kodzie, sterowniki procesora w systemie Linux tzw. *frequency governor* i *idle governor* [35, 34] wykorzystywane nie tylko w typowych komputerach osobistych i serwerach, ale również w wielu urządzeniach, w tym w sprzęcie sieciowym. Funkcjonują one komplementarnie, dobierając częstotliwości taktowania i wprowadzając procesor w jeden ze stanów uśpienia, w zależności od obserwowanego obciążenia.

Działanie sieci komputerowej jest podporządkowane wspólnym celom związanym z efektywnym realizowaniem świadczonych usług. Stąd lokalne, tj. implementowane w poszczególnych urządzeniach, mechanizmy oszczędzania energii są w wielu przypadkach niewystarczające. Podobna sytuacja ma miejsce w przypadku systemów obliczeniowych, szczególnie złożonych z rozproszonych geograficznie komponentów. W istocie można je traktować jako rodzaj sieci, w której oprócz węzłów służących do transmisji danych występują węzły odpowiedzialne za przetwarzanie i magazynowanie danych. Zarządzanie takim systemem sprowadza się w najogólniejszej postaci do alokacji zadań do jednostek obliczeniowych i zapewnienia zasobów niezbędnych do ich wykonania oraz wyznaczenia odpowiednich tras pozwalających na transmisję danych między węzłami przetwarzającymi dane.

4.3 Elementy składowe osiągnięcia

Mechanizmy lokalnego oszczędzania energii są stosowane w większości używanych obecnie urządzeń – w serwerach i komputerach osobistych, jak również w sprzęcie sieciowym. Wnioski z przeprowadzonych przeze mnie studiów literaturowych oraz analizy dostępnych rozwiązań zamieszczone w pracy [H3] pokazują, że tego typu techniki nie wystarczają jednak do optymalnego zarządzania energią w złożonych systemach obliczeniowych, gdzie konieczna jest koordynacja pracy wszystkich komponentów. Próby rozwiązania tego problemu zostały zaproponowane w systemach zarządzających klastrami obliczeniowymi[1]. Niemniej, brakuje rozwiązań kompleksowych, uwzględniających sterowanie wszystkimi komponentami systemu, tj. jednostkami obliczeniowymi i łączącą je siecią. Wynika to w dużej mierze z faktu, że rozwiązanie zadania koordynacji dużego, rozproszonego systemu IT jest w ogólności trudne, ze względu na konieczność uwzględnienia wielu kryteriów, złożoność obliczeniową jak i samą implementację.

Prowadząc prace badawcze postawiłem tezę, że efektywne, tak w sensie energetycznym jak i jakości obsługi działanie systemu wymaga koordynacji wszystkich jego elementów. W takim podejściu mechanizmy lokalne zapewniają nadrzędnemu układowi sterowania możliwość wpływania na stan poszczególnych urządzeń, zwalniając go jednocześnie z konieczności częstej ingerencji. Zastosowanie algorytmów lokalnych, np. w sterownikach procesora czy kart sieciowych, nie pozwala osiągnąć wymaganej efektywności. Wynika to m.in. z dążenia do nadmiernego rozrzucania zadań między pracujące z ograniczoną prędkością procesory, czy też nieuwzględniania poboru mocy przez węzły sieciowe znajdujące się na dalszych etapach aktywnych tras. Dodatkowo, wykorzystanie metod koordynacji, a więc unikanie bezpośredniego wskazywania przez węzeł nadrzędny nastaw podsystemom, pozwala zmniejszyć nakład obliczeniowy, oraz pozostawia jednostkom lokalnym pewien zakres samodzielności. Ta ostatnia cecha ma znaczenie, tak ze względu na możliwość zapewnienia szybszej reakcji na zmiany warunków pracy oraz zwiększenie odporności na awarie, jak i na organizacyjne czy wręcz psychologiczne aspekty współpracy odrębnych jednostek. Przykładem może być budowa infrastruktury badawczej przez kilka instytucji dysponujących różnorodnym sprzętem: systemami zbierania i magazynowania danych, klastrami obliczeniowych, dedykowanymi instalacjami laboratoryjnymi. Każda z nich wykorzystuje wspomniany sprzęt w swojej codziennej działalności, z czego wynika konieczność optymalizacji jego wykorzystaniem. Z drugiej strony, w celu przeprowadzenia wspólnego eksperymentu, należy skoordynować użycie wszystkich potrzebnych elementów zarządzanych przez współpracujące strony. Rozwiązaniem jest umożliwienie instytucjom zadeklarowania dostępnych dla wspólnego przedsięwzięcia zasobów i wykorzystanie ich bez nadmiernego ingerowania w funkcjonujące w instytucji strategie zarządzania.

W dalszej części autoreferatu przestawię swój wkład w rozwój wspomnianej tematyki. Prace, których rezultaty są prezentowane w niniejszym opracowaniu, obejmowały następujące zagadnienia:

1. efektywne struktury sterowania dla energooszczędnego, rozproszonego systemu obliczeniowego,

2. zwiększenie wydajności energetycznej sieci przewodowych i bezprzewodowych poprzez energooszczędną inżynierię ruchu oraz specjalizowane metody harmonogra-

[1] np. mechanizm DRS (Distributed Resource Scheduler) w oprogramowaniu do zarządzania klastrem vSphere firmy VMware https://www.vmware.com/products/vsphere/drs-dpm.html

mowania komunikacji węzłów,

3. modelowanie poboru mocy przez procesor i system komputerowy z wykorzystaniem wyników eksperymentów laboratoryjnych.

Pierwsze z wymienionych zagadnień jest najobszerniejsze, w istocie proponowana struktura zawiera w sobie elementy systemu z punktu drugiego tzn. energooszczędne sterowanie siecią przesyłową łączącą wchodzące w skład systemu obliczeniowego klastry. Zagadnienie trzecie pełni funkcję pomocniczą dostarczając modeli sterowanych podsystemów.

Podsumowując, prowadzone przeze mnie badania obejmowały zarówno rozwiązania formalne, prace konstrukcyjne, jak też badania eksperymentalne. Prace koncepcyjne, w tym teoretyczne, dotyczyły zagadnień modelowania matematycznego, opracowania struktur sterowania oraz sformułowania, analizy i oceny złożoności zadań programowania matematycznego. Efektem prac konstrukcyjnych są odpowiednio dobrane i dostosowane algorytmy optymalizacji numerycznej oraz środowiska sprzętowo-programowe do identyfikacji modeli i weryfikacji poprawności wyznaczonych decyzji sterujących. Ten wątek badawczy był szczególnie ważny i wymagał największego wysiłku. Prowadząc badania, zwracałem szczególną uwagę na walidację wyników. Stąd zaproponowane rozwiązania były najpierw sprawdzane w środowisku symulacyjnym, a następnie w laboratoriach, na rzeczywistym sprzęcie. Konieczne więc było zbudowanie odpowiednich stanowisk badawczych i przygotowanie zestawów danych testowych.

Badania eksperymentalne obejmowały identyfikację parametryczną modeli opisujących procesy przebiegające w urządzeniach tworzących systemy przetwarzania danych oraz weryfikację poprawności i ocenę stosowalności i wydajności opracowanych przeze mnie schematów rozdziału zasobów. Ze względu na złożoność rozważanych systemów i różnorodność sprzętu oraz proponowanych metod, algorytmów i protokołów były to prace mocno absorbujące.

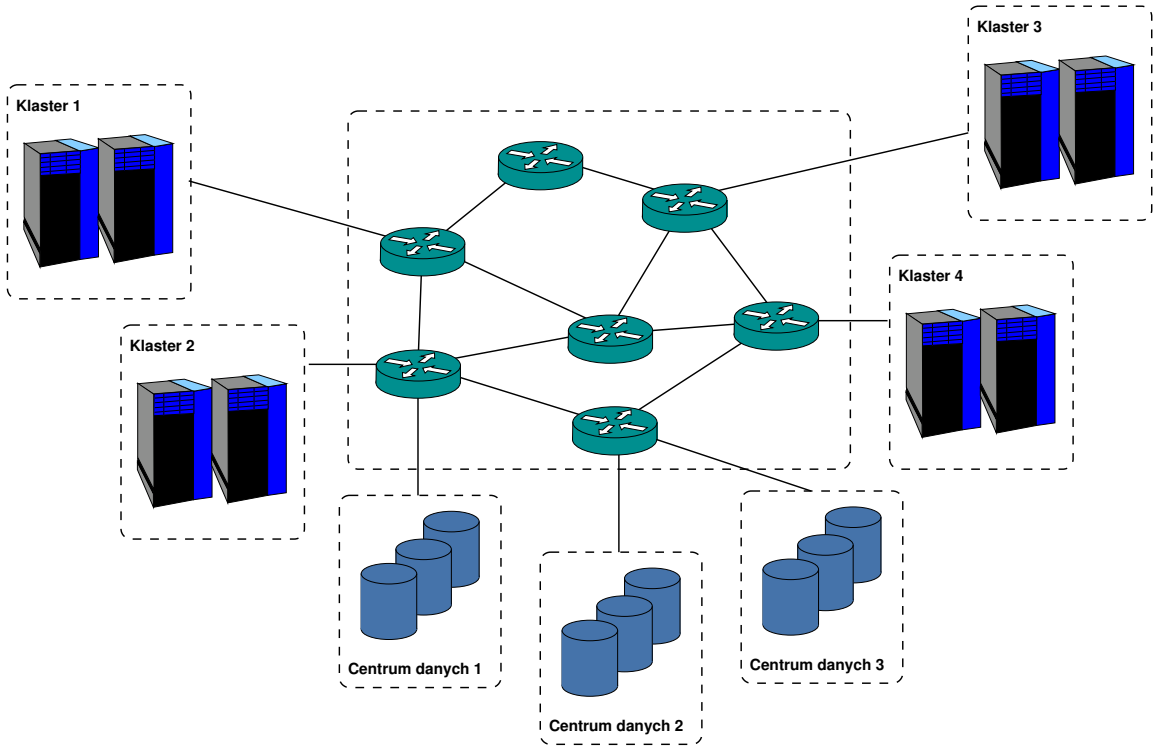
Uwaga: W autoreferacie jako równoważne używane będą następujące pojęcia: rozproszony system przetwarzania danych, rozproszony system obliczeniowy, chmura obliczeniowa. Używając tych określeń, mam na myśli system służący do przetwarzania znacznych ilości danych, zbudowany z klastrów obliczeniowych połączonych siecią. Zgłaszane osiągnięcie naukowe obejmuje badania prowadzone przeze mnie przy wsparciu dyplomantów, ale też prace wykonywane w projektach realizowanych przez krajowe i międzynarodowe zespoły naukowców i inżynierów, których byłem członkiem. Stąd większa liczba autorów w przypadku niektórych publikacji. W autoreferacie koncentruję się na prezentacji wyników badań, które są moim osiągnięciem i stanowiły wkład w wspomniane projekty i prace rozwojowe.

4.4 Struktury sterowania i algorytmy przydziału zasobów energooszczędnego, rozproszonego systemu przetwarzania danych

Współczesne systemy gromadzące i przetwarzające dane charakteryzują się zazwyczaj wielką skalą i rozproszeniem geograficznym wynikającym z udziału wielu podmiotów. O jakości ich działania, a tym samym efektywności, decyduje w znacznym stopniu odpowiednie zarządzanie zasobami. Skuteczne rozwiązanie tego problemu wymaga uwzględnienia szeregu zagadnień, takich jak:

1. spełnienie wymagań jakościowych odnośnie przetwarzania danych,
2. ograniczenie całkowitego zużycia energii przez system przetwarzania danych,
3. koordynacja działania odległych podsystemów, w tym zlokalizowanych w różnych podmiotach.

Należy zaznaczyć, że wymienione powyżej problemy są wzajemnie powiązanie i zazwyczaj powinno się je rozważać łącznie. W szczególności konieczność spełnienia wymagań jakościowych praktycznie zawsze ogranicza możliwość oszczędzania energii. Podobnie, koordynacja działań wielu podmiotów utrudnia częstą i bezpośrednią ingerencję w zarządzane przez nie systemy na rzecz podejmowanych, w dłuższej skali czasowej, działań, których celem jest optymalizacja kryteriów dla całego systemu, pozostawiając stosowny margines samodzielności lokalnym decydentom.



Rysunek 1: Architektura rozważanego systemu przetwarzania danych (chmury obliczeniowej).

Systemy klastrowe i chmurowe mają wiele zastosowań. Mogą one świadczyć zróżnicowane usługi przetwarzania danych dla wielu, często zewnętrznych podmiotów. Mogą też być budowane w ramach nadzorującej je instytucji w konkretnym celu, np. dostarczenia infrastruktury dla określonego systemu informatycznego. W obu przypadkach można mówić o świadczeniu usług, choć ich definicja będzie różna, a co się z tym wiąże, różne będą wymagania na jakość obsługi. Zgodnie z literaturą najczęściej są to:

1. wymóg zapewnienia określonego czasu wykonania zadań,
2. wymóg zagwarantowania określonej zdolności przetwarzania.

Pierwszy wariant jest typowy dla wsadowego trybu przetwarzania, gdzie zlecający zadanie oczekuje, że wynik będzie dostępny we wskazanym momencie, bądź jest gotowy ponieść koszty szybszego wykonania obliczeń. Tak zdefiniowany cel implikuje problem matematyczny w postaci zadania harmonogramowania, którego dokładne rozwiązanie jest dla bardziej złożonych zadań trudne. Drugi przypadek dotyczy głównie zastosowań związanych z przetwarzaniem strumienia danych pojawiających się w sposób ciągły, z pewną, możliwą zazwyczaj do scharakteryzowania, intensywnością. W kategorii tej mieszczą się zarówno zadania polegające na obsłudze żądań pochodzących od klientów, w przypadku np. sklepu czy portalu internetowego, jak też przetwarzanie danych zbieranych podczas wielkich eksperymentów naukowych, czy też przez systemy wspierające zapewnienie bezpieczeństwa sieci teleinformatycznych (np. rozproszone systemy IDS[2]). Znajomość charakterystyki obciążenia pozwala zlecającemu określić, w mniej lub bardziej precyzyjny sposób, wymagania odnośnie zasobów obliczeniowych. Z formalnego punktu widzenia jest to więc zadanie alokacji zasobów. W pewnych przypadkach jest ono prostsze od zagadnienia harmonogramowania. Dzieje się tak, gdy obciążenie jest względnie stałe, co umożliwia eliminację czasu w sposób analogiczny jak podczas rozwiązywania zadania sterowania na horyzoncie nieskończonym a przez to zmniejszenie wymiarowości problemu.

Większość moich doświadczeń wynikających z pracy zawodowej dotyczy budowy klastrów dedykowanych systemom informatycznym przetwarzającym masowe dane spływające z podsystemów analizujących bezpieczeństwo sieci. Z tego też powodu bliższe jest mi podejście drugie, gdzie zarządzający infrastrukturą jest odpowiedzialny za zapewnienie odpowiedniej mocy obliczeniowej dla maszyn wykonujących poszczególne zadania. Konsekwentnie większość prezentowanych w autoreferacie prac skupiała się na rozwiązaniu tego problemu.

Zdefiniujmy system, przetwarzania i analizy danych, którego dotyczyły prowadzone prace badawcze. Jest to chmura obliczeniowa składająca się z szeregu klastrów połączonych siecią przedstawiona na rysunku 1. Dodatkowo w infrastrukturze występują węzły pełniące rolę źródeł, czy też magazynów danych. Mogą to być zarówno bazy danych umieszczone w osobnych lokalizacjach, tzw. centrach danych (*data center*), ale też instalacje zbierające dane, np. sieci czujników rozrzucone w terenie czy sieci sond próbkujących ruch sieciowy. Sterowanie tak złożoną infrastrukturą wiąże się z koniecznością uwzględnienia szeregu problemów:

1. elementy infrastruktury znajdują się często w odległych lokalizacjach,
2. elementy infrastruktury mogą być zarządzane przez różne podmioty,
3. infrastruktura może być niejednorodna technologicznie,
4. koordynując działania, należy uwzględnić transmisję danych przez sieć.

Celem sterowania jest, jak zostało wspomniane wcześniej, ograniczenie zużycia energii przez całą infrastrukturę, przy spełnieniu wymagań na jakość świadczonych przez nią

[2] *Intrusion Detection System* – systemy wykrywające niepożądane działania poprzez m.in. analizę ruchu sieciowego. Skuteczne działanie takich systemów wymaga ciągłego działania z wydajnością pozwalającą na identyfikację zagrożeń i generowanie alarmów w czasie pozwalającym na podjęcie działań ograniczających zakres szkód.

usług. Problemy wymienione w pierwszych trzech punktach skłaniają do zastosowania lokalnego sterowania w każdym z klastrów. Przemawiają za tym zarówno korzyści organizacyjne, tj. umożliwienie podejmowania decyzji podmiotom bezpośrednio odpowiedzialnym za klastry, jak też czysto techniczne, związane z koniecznością dostosowania modeli matematycznych, algorytmów i oprogramowania do technologii, w jakiej zrealizowany jest dany klaster. Dodatkowo podejście takie pozwala na umieszczenie elementów oprogramowania sterującego w tej samej lokalizacji, czy wręcz na tym samym sprzęcie, co skutkuje zwiększeniem prędkości działania i niezawodności. Z drugiej strony, osiągnięcie wspólnego celu, jakim jest minimalizacja zużycia energii, w systemie jako całości, wymaga podejścia globalnego, obejmującego cały system, tym bardziej że konieczność przesyłania danych przez sieć narzuca dodatkowe ograniczenia wynikające z przepustowości. Co więcej, transmisja danych wpływa na zużycie energii, co powinno być również uwzględnione przy alokacji zadań.

4.4.1 Centralny przydział zasobów energooszczędnej chmury obliczeniowej

W pracy [H1] zdefiniowałem zadanie sterowania rozdziałem zasobów chmury obliczeniowej, przy założeniu, że wszystkie decyzje są podejmowane przez jednostkę centralnego zarządcy chmury. Istotnym aspektem odróżniającym zaproponowane podejście od wariantów znanych z literatury (np. [6, 29, 43, 9, 22, 31]) jest uwzględnienie, przy alokacji zadań, poboru mocy przez sieć łączącą klastry. Alokowane do maszyn wirtualnych zadania charakteryzowane są przez zasoby potrzebne do ich wykonania, tj. moc obliczeniową procesora, wyrażoną np. w MIPS oraz wielkość pamięci operacyjnej w MB. Wynika to z kluczowego wymagania narzuconego na rozważany system, jakim jest zagwarantowanie ciągłości przetwarzania danych. Celem jest wyznaczenie alokacji zadań i odpowiadających jej stanów energetycznych urządzeń tak, aby zminimalizować zużycie energii przez cały system. Typowo jest to osiągane przez skupienie zadań na wybranych węzłach i wyłączenie, względnie wprowadzenie w niski stan energetyczny, pozostałych. Należy jednak pamiętać, że uwzględnienie zużycia energii i przepustowości sieci transmisyjnej wprowadza dodatkowe ograniczenia. W przedstawionym dalej zadaniu programowania matematycznego funkcja celu opisuje pobór mocy przez wszystkie składniki systemu obliczeniowego, w tym maszyny wchodzące w skład klastrów oraz urządzenia tworzące łączącą je sieć. Przyjęty model energetyczny urządzeń zakłada możliwość wprowadzania procesorów maszyn wchodzących w skład klastra w jeden z K stanów energetycznych. W przypadku ruterów tworzących sieć przesyłową dopuszcza się wyłączenie (czy też wprowadzenie w stan uśpienia) całych urządzeń, jak też znajdujących się w nich kart liniowych i umieszczonych na nich portów. Rozważana sieć składa się więc z R ruterów z C kartami zawierającymi P portów. Maszyny rozmieszczone są w F klastrach, każdy z nich zaś jest wyposażony w S_f procesorów będących najmniejszą jednostką podlegającą alokacji. Poniżej przedstawiono formalny zapis zadania optymalizacji:

$$\min_{\substack{x^k_{fs},\vartheta_{fsj},x_r,\\ x_c,x_l,v_{ljd}}} \left[\sum_{f=1}^{F}\sum_{s=1}^{S_f}\sum_{k=1}^{K} P^k_{fs}x^k_{fs} + \sum_{r=1}^{R} P_r x_r + \sum_{c=1}^{C} P_c x_c + \sum_{l=1}^{L} P_l x_l \right], \tag{1}$$

$$\forall_{\substack{f=1,\ldots,F\\ s=1,\ldots,S_f}} \sum_{k=1}^{K} x_{fs}^{k} = 1, \tag{2}$$

$$\forall_{\substack{f=1,\ldots,F\\ s=1,\ldots,S_f}} \sum_{j=1}^{J} W_j \vartheta_{fsj} \leqslant \sum_{k=1}^{K} \Theta_{fs}^{k} x_{fs}^{k}, \tag{3}$$

$$\forall_{\substack{f=1,\ldots,F\\ s=1,\ldots,S_f}} \sum_{j=1}^{J} M_j \vartheta_{fsj} \leqslant \Psi_{fs}, \tag{4}$$

$$\forall_{j=1,\ldots,J} \sum_{f=1}^{F} \sum_{s=1}^{S_f} \vartheta_{fsj} = 1, \tag{5}$$

$$\forall_{\substack{j=1,\ldots,J,\\ d=1,\ldots,D_j,\\ c=1,\ldots,C}} \sum_{p=1}^{P} u_{cp} \sum_{l=1}^{L} a_{lp} v_{ljd} \leqslant x_c, \tag{6}$$

$$\forall_{\substack{j=1,\ldots,J,\\ d=1,\ldots,D_j,\\ c=1,\ldots,C}} \sum_{p=1}^{P} u_{cp} \sum_{l=1}^{L} b_{lp} v_{ljd} \leqslant x_c, \tag{7}$$

$$\forall_{\substack{r=1,\ldots,R,\\ c=1,\ldots,C}} y_{rc} x_c \leqslant x_r, \tag{8}$$

$$\forall_{\substack{l=1,\ldots,L,\\ j=1,\ldots,J,\\ d=1,\ldots,D_j}} v_{ljd} \leqslant x_l, \tag{9}$$

$$\forall_{\substack{r\in\bar{\Omega},\\ j=1,\ldots,J,\\ d=1,\ldots,D_j}} \sum_{c=1}^{C} y_{rc} \left(\sum_{p=1}^{P} u_{cp} \sum_{l=1}^{L} a_{lp} v_{ljd} - \sum_{p=1}^{P} u_{cp} \sum_{l=1}^{L} b_{lp} v_{ljd} \right) = g_{rjd}, \tag{10}$$

$$\forall_{\substack{r\in\bar{\Omega},\\ j=1,\ldots,J,\\ d=1,\ldots,D_j}} \sum_{c=1}^{C} y_{rc} \left(\sum_{p=1}^{P} u_{cp} \sum_{l=1}^{L} a_{lp} v_{ljd} - \sum_{p=1}^{P} u_{cp} \sum_{l=1}^{L} b_{lp} v_{ljd} \right) = -\sum_{f=1}^{F} \sum_{s=1}^{S_f} \vartheta_{fsj} h_{rf}, \tag{11}$$

$$\forall_{\substack{r\in\Omega,\\ j=1,\ldots,J,\\ d=1,\ldots,D_j}} \sum_{c=1}^{C} y_{rc} \left(\sum_{p=1}^{P} u_{cp} \sum_{l=1}^{L} a_{lp} v_{ljd} - \sum_{p=1}^{P} u_{cp} \sum_{l=1}^{L} b_{lp} v_{ljd} \right) = 0, \tag{12}$$

$$\forall_{l=1,\ldots,L} \sum_{j=1}^{J} \sum_{d=1}^{D_j} \beta_{jd} v_{ljd} \leqslant B_l, \tag{13}$$

$$\forall_{l=1,\ldots,L} \sum_{l'=1}^{L} \gamma_{ll'} x_l = \sum_{l'=1}^{L} \gamma_{l'l} x_{l'}. \tag{14}$$

Alokację procesorów do zadań opisuje binarna zmienna $\vartheta_{fsj} = 1$, jeśli procesor s z klastra f jest alokowany do zadania j (0 w przeciwnym przypadku), aktualny stan procesora $x_{fs}^{k} = 1$, gdy procesor s z klastra f pracuje w stanie energetycznym k (0 w przeciwnym przypadku). Stan urządzeń sieciowych opisują zmienne binarne $x_r = 1$, $x_c = 1$ i $x_l = 1$ oznaczające, że odpowiednio: ruter r, karta liniowa c i łącze l są włączone (0 oznacza wyłączenie, bądź stan uśpienia), zmienna $v_{ljd} = 1$ wskazuje, że łącze l przesyła dane d zadania j (0 w przeciwnym przypadku). Topologia połączeń jest ustalona za pomocą stałych o wartościach binarnych: $a_{lp} = 1$ jeśli łącze l zaczyna się w porcie p (0 w przeciwnym przypadku), $b_{lp} = 1$ jeśli łącze l prowadzi do portu p (0 w przeciwnym przypadku), $u_{cp} = 1$ jeśli port p znajduje się na karcie c (0 w przeciwnym przypadku), $y_{rc} = 1$ jeśli karta c jest zainstalowana w ruterze r (0 w przeciwnym przypadku). Dodatkowo stała g_{rjd} =1,

gdy centrum danych połączone poprzez ruter r może dostarczyć dane d zadaniu j, h_{rf}=1 wskazuje ruter brzegowy r, poprzez który jest podłączony klaster f (0 w przeciwnym przypadku).

Stałe W_j i Mj określają odpowiednio zapotrzebowanie na moc obliczeniową i pamięć operacyjną zadania j, Θ_{fs}^k i Ψ_{fs} to moc obliczeniowa w stanie energetycznym k i pamięć operacyjna związane z procesorem s z klastra f, β_{jd} to zapotrzebowanie na pasmo przepływu d, zaś B_l oznacza pasmo dostępne na łączu l. Połączeniu w pary łącz jednokierunkowych służy stała $\gamma_{l'l} = 1$, gdy łącza l oraz l' obsługują dwa kierunki tej samej relacji.

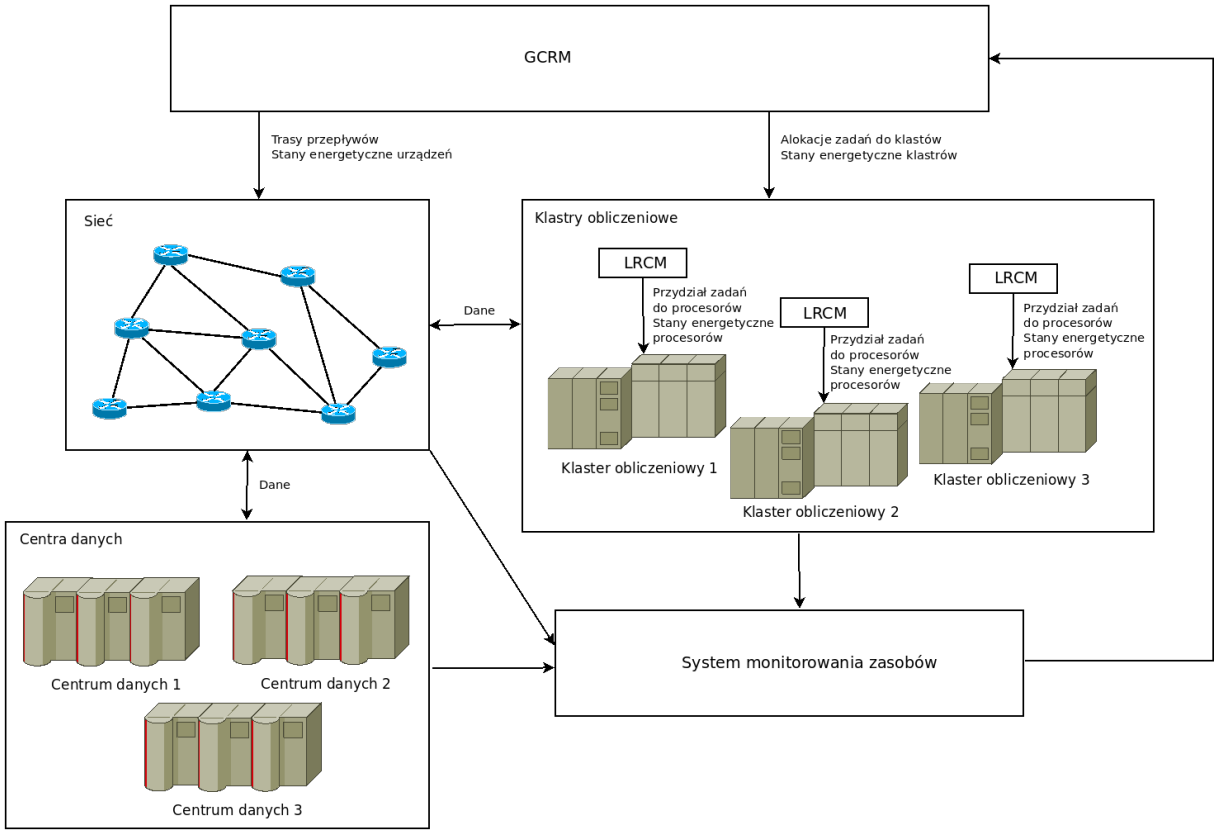
Ograniczenie (2) wymusza warunek, że każdy procesor działa w jednym stanie energetycznym, (3) i (4) narzucają warunki na zasoby odpowiednie do wykonania alokowanych zadań, (5) gwarantuje, że wszystkie zadania zostaną rozdzielone między klastry. Za stan energetyczny łączy, ruterów i kart liniowych uczestniczących w transmisji danych odpowiadają ograniczenia (6)-(9). Ciągłość przepływów w węzłach źródłowych, docelowych i pośrednich zapewniają nierówności (10)-(12), zaś (13) zabezpiecza wymaganą pojemność łączy. Wszystkie łącza modelowane są jako pary łączy jednokierunkowych, z tego powodu (14) zapewnia równoczesne włączenie obu z nich w przypadku transmisji.

Zapisane w ten sposób zadanie programowania matematycznego (1)-(14) może być rozwiązywane z użyciem standardowych metod optymalizacji dla zadań mieszanych z liniowymi ograniczeniami i liniową funkcją celu. Należy jednak zwrócić uwagę, że ze wzrostem jego wymiarowości czas potrzebny do jego rozwiązania może szybko stać się nieakceptowalny. Szczególnie istotną rolę odgrywają tu zmienne x_{fs}^k i ϑ_{fsj}, których wymiar jest uzależniony od liczby zadań i zainstalowanych w klastrach procesorów. Ze zrozumiałych względów liczby te rosną razem, przyczyniając się do eksplozji wymiarowości. W związku z tym powyższe zadanie należy traktować przede wszystkim jako punkt wyjścia do opisanej dalej hierarchicznej struktury decyzyjnej.

4.4.2 Dwupoziomowa struktura decyzyjna przydziału zasobów chmury obliczeniowej

W przypadku złożonych systemów, a takim jest rozproszony system przetwarzania danych, naturalnym rozwiązaniem jest jego dekompozycja na podsystemy i wskazanie jednostki nadzorującej pracę wyodrębnionych komponentów [20]. Zarządzanie jest realizowane w strukturze hierarchicznej, gdzie decyzje dotyczące podsystemów są podejmowane przez lokalnych decydentów, a ich działania koordynuje jednostka centralna. W literaturze są szeroko omawiane zalety takiego rozwiązania [20, 33, 5]. W pracach [H2, U1, U2] zaproponowałem dwupoziomowy schemat wyznaczania decyzji dotyczących przydziału zasobów chmury obliczeniowej. Przedstawione rozwiązanie polega na konstrukcji zadań optymalnego rozdziału zasobów w poszczególnych klastrach obliczeniowych (jednostkach lokalnych) i sformułowaniu zadania nadrzędnego rozwiązywanego przez zarządcę całej chmury. Motywacją dla takiego podejścia było przede wszystkim:

1. umożliwienie współdziałania podmiotów zarządzających poszczególnymi klastrami przez wyznaczenie zakresu ich odpowiedzialności,
2. określenie sposobu komunikacji między podmiotami,
3. dekompozycja złożonego zadania (1)-(14) na mniejsze, łatwiejsze do rozwiązania podzadania.



Rysunek 2: Architektura chmury.

Często obserwowanym problemem we współpracy między ośrodkami jest niechęć do wzajemnego udostępniania sprzętu. Przyczyną takiej sytuacji bywa brak odpowiedniego zabezpieczenia, które zapewniałoby dostateczną kontrolę nad zasobami danej jednostki, a jednocześnie umożliwiałoby jego wykorzystanie przez jednostki współpracujące. W proponowanej strukturze decyzyjnej koordynator (GCRM – *Global Computing Resource Manager*) odpowiada za przydział zadań obliczeniowych do poszczególnych klastrów i sterowanie łączącą je siecią. Co istotne, klastry zgłaszając koordynatorowi gotowość do współpracy, nie muszą oferować swoich zasobów w całości, a jedynie wydzielić ich część tak, aby zachować zdolność do wykonania innych zadań. Takie rozwiązanie ma szereg zalet, przede wszystkim, oprócz wspomnianej możliwości wykorzystania klastra do własnych celów, daje szansę zapewnienia pewnej elastyczności, tj. możliwości wykonania zadań o specyfikacji nieco przekraczającej zadeklarowane zasoby (np. w okresach, gdy własne zadania nie są wykonywane z pełną intensywnością). Zasoby klastra są określane w sposób możliwie prosty, tj. jako zestaw procesorów o mocy obliczeniowej wyrażonej w ogólnie przyjętych jednostkach (np. MIPS, MFLOPS), wyposażonych w pamięć operacyjną o podanej pojemności (w GB). Takie rozwiązanie upraszcza zadanie alokacji rozwiązywane przez GCRM, pozwala też uniknąć udostępniania szczegółowych informacji o konstrukcji klastra, co dla niektórych jednostek może również mieć znaczenie. Architektura chmury obliczeniowej oraz struktura systemu są przedstawione na rysunku 2.

Sformułowanie zadania programowania matematycznego koordynatora. Zadanie koordynatora (GCRM) sprowadza się do wstępnej alokacji zadań do klastrów obliczeniowych. Ze względu na konieczność uwzględnienia wszystkich klastrów tworzących

chmurę obliczeniową musi być ono stosunkowo proste. Podejście, które zaprezentowałem w [H3] ogranicza sterowanie stanem maszyn tworzących klaster do ich włączania i wyłączania i wprowadza pojęcie stanu energetycznego klastra. Przyjąłem, że system monitorowania zasobów przekazuje koordynatorowi informację o liczbie procesorów aktualnie udostępnianych przez każdy z f klastrów oraz średni koszt energetyczny $\bar{P}_s^f$ związany z ich działaniem. W przypadku, gdy zarządca klastra udostępnia pełne zasoby aktualnie działających maszyn, koszt energetyczny jest wyznaczany jako $\bar{P}_s^f = (\sum_{s=1}^{S_f} P_s^{max} x_s^f)/S_f$, a odpowiadająca mu moc obliczeniowa jako $\bar{\Theta}_s^f = (\sum_{s=1}^{S_f} \Theta_s^{max} x_s^f)/S_f$. Zmienna x_s^f opisuje decyzje zarządcy klastra (LCRM – *Local Computing resource Manager*), co oznacza, że koordynator nie ma na nią bezpośredniego wpływu. Przyjmuje ona wartości 0 lub 1: $x_s^f = 1$, jeżeli procesor s jest w stanie aktywnym, $x_s^f = 0$ gdy, procesor s jest w stanie uśpionym. Przy takim podejściu stan energetyczny k klastra oznacza średnią liczbę włączonych procesorów. Pozwala to wyznaczyć średnie zużycie energii P_f^k przez klaster oraz odpowiadającą mu moc obliczeniową Θ_f^k:

$$\forall_{k=1,\ldots,S_f} \quad P_f^k = \sum_{s=1}^{k} \bar{P}_s^f x_s^f. \tag{15}$$

$$\forall_{k=1,\ldots,S_f} \quad \Theta_f^k = \sum_{s=1}^{k} \bar{\Theta}_s^f x_s^f. \tag{16}$$

Taki opis stanu energetycznego klastra jest znacznym uproszczeniem w stosunku do modelu energetycznego stosowanego w podejściu scentralizowanym (1)-(14). Pozwala to zmniejszyć wymiar zmiennych decyzyjnych, a przez to ułatwić rozwiązanie zadania. Jest to zgodne z opisanym wcześniej schematem hierarchicznego zarządzania chmurą obliczeniową. GCRM wymusza pewne ograniczenia, a każdy z zarządców lokalnych (LCRM) dobiera stany energetyczne procesorów zgodnie ze swoimi lokalnymi celami. Przy tak zdefiniowanym modelu podejmowania decyzji zadanie koordynatora przyjmuje następującą postać.

$$\min_{x_f,\vartheta_f,x_r,x_c,x_l,v_{ld}} \left[\sum_{f=1}^{F} \sum_{k=1}^{K_f} P_f^k x_f^k + \sum_{r=1}^{R} P_r x_r + \sum_{c=1}^{C} P_c x_c + \sum_{l=1}^{L} P_l x_l \right], \tag{17}$$

$$\forall_{f=1,\ldots,F} \quad \sum_{k=1}^{K_f} x_f^k \leqslant 1, \tag{18}$$

$$\forall_{f=1,\ldots,F} \quad \sum_{j=1}^{J} W_j \vartheta_{fj} \leqslant \sum_{k=1}^{K_f} \Theta_f^k x_f^k, \tag{19}$$

$$\forall_{f=1,\ldots,F} \quad \sum_{j=1}^{J} M_j \vartheta_{fj} \leqslant \sum_{k=1}^{K_f} \Psi_f x_f^k, \tag{20}$$

$$\forall_{\substack{d=1,\ldots,D_j,\\ c=1,\ldots,C}} \quad \sum_{p=1}^{P} u_{cp} \sum_{l=1}^{L} a_{lp} v_{ld} \leqslant x_c, \tag{21}$$

$$\forall_{\substack{d=1,\ldots,D_j,\\ c=1,\ldots,C}} \quad \sum_{p=1}^{P} u_{cp} \sum_{l=1}^{L} b_{lp} v_{ld} \leqslant x_c, \tag{22}$$

$$\forall_{\substack{r=1,\ldots,R,\\ c=1,\ldots,C}} \quad y_{rc}x_c \leqslant x_r, \tag{23}$$

$$\forall_{\substack{j=1,\ldots,J,\\ d=1,\ldots,D_j,\\ r=1,\ldots,R,\\ p=e_d}} \sum_{c=1}^{C} y_{rc}u_{cp} \sum_{l=1}^{L} a_{lp}v_{ld} - \sum_{c=1}^{C} y_{rc}u_{cp} \sum_{l=1}^{L} b_{lp}v_{ld} = 1, \tag{24}$$

$$\forall_{\substack{j=1,\ldots,J,\\ d=1,\ldots,D_j,\\ r=1,\ldots,R\\ p\neq g_d, p\neq e_d}} \sum_{c=1}^{C} y_{rc} \sum_{p=1}^{P} u_{cp} \sum_{l=1}^{L} a_{lp}v_{ld} - \sum_{c=1}^{C} y_{rc} \sum_{p=1}^{P} u_{cp} \sum_{l=1}^{L} b_{lp}v_{ld} = 0, \tag{25}$$

$$\forall_{\substack{j=1,\ldots,J,\\ d=1,\ldots,D_j,\\ r=1,\ldots,R,\\ p=g_d}} \sum_{c=1}^{C} y_{rc}u_{cp} \sum_{l=1}^{L} a_{lp}v_{ld} - \sum_{c=1}^{C} y_{rc}u_{cp} \sum_{l=1}^{L} b_{lp}v_{ld} = -1, \tag{26}$$

$$\forall_{l=1,\ldots,L} \quad \sum_{d=1}^{D} V_d v_{ld} \leqslant \Phi_l x_l, \tag{27}$$

Stałe występujące w powyższym sformułowaniu to W_j i M_j, oznaczające odpowiednio zapotrzebowanie zadania j na moc obliczeniową i pamięć, V_d to wielkość zapotrzebowania d na pojemność łącza wiążącego dwa porty e_d i g_d, przez które są przekazywane dane ze zdalnego centrum danych. Źródło danych to e_d zaś g_d port docelowy klastra odbierającego dane. D_j oznacza liczbę przepływów w sieci ($d = 1, \ldots, D_j$) związanych z zapotrzebowaniem na dane niezbędne do wykonania zadania j. Szczególnym przypadkiem jest $D_j = 0$ co oznacza, że zadanie j nie wymaga zdalnej transmisji i nie generuje związanych z nią kosztów. Pojemność łącza l jest określona przez Φ_l. Stała Θ_f^k oznacza moc obliczeniową klastra f w stanie k, a Ψ_f całkowitą pamięć klastra f. Za powiązanie kart liniowych z ruterami odpowiada zmienna y_{rc}. Karta c należy do rutera r, gdy $y_{rc} = 1$, podobnie zmienna $u_{cp} = 1$, jeżeli port p należy do karty c. Skierowane łącze l wychodzi z portu p, jeśli zmienna $a_{lp} = 1$, $b_{lp} = 1$, jeżeli skierowane łącze l wchodzi do portu p. Zmienne stanu przyjmują wartości 0 lub 1. Zmienna $x_f^k = 1$, gdy klaster f jest w stanie k, $\vartheta_{fj} = 1$, gdy zadanie j jest alokowane do klastra f, $x_r = 1$, $x_c = 1$ i $x_l = 1$ odpowiednio, jeśli ruter r, karta c i link l są aktywne i uczestniczą w transmisji danych między klastrem obliczeniowym i centrum danych, stała $v_{ld} = 1$, jeżeli przepływ d jest realizowany na łączu l.

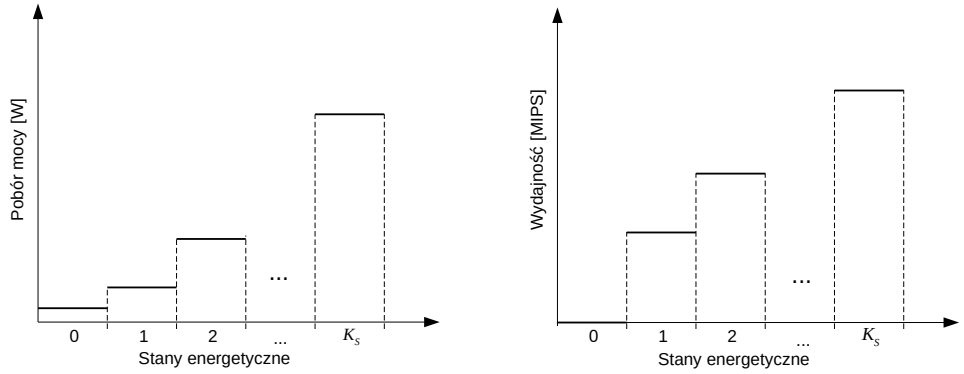
W przedstawionym sformułowaniu zadania pomijane są koszty energetyczne występujące w stanie, gdy urządzenia są nieaktywne.

Ograniczenia występujące w zadaniu mają następującą interpretację. Ograniczenie (18) gwarantuje, że każdy klaster może pracować w danej chwili tylko w jednym stanie energetycznym, ograniczenia (19) i (20) wymuszają taką alokację zadań, w której zasoby wybranych klastrów (CPU i pamięć) są wystarczające do wykonania alokowanych zadań. Pozostałe ograniczenia dotyczą sieci szkieletowej. Ograniczenia (21)-(23) determinują liczbę ruterów i kart wykorzystanych do transmisji danych, (24)-(26) gwarantują ciągłość przepływów dla źródłowego, pośredniczącego i docelowego rutera, (27) gwarantuje, że przepływ nie przekroczy dostępnej pojemności łącza.

Zadanie (17)-(27) jest całkowitoliczbowym zadaniem liniowym. Do jego rozwiązania można zastosować metodę podziału i ograniczeń. Niestety jest to zadanie NP-trudne, a więc mogą wystąpić problemy z jego rozwiązaniem w przypadku dużych centrów przetwarzania danych. Często stosowanym rozwiązaniem, w przypadku tego typu zadań, jest ich relaksacja i transformacja do przestrzeni ciągłej.

Zadanie lokalne (LRCM). Rozwiązaniem zadania koordynatora jest alokacja J blo-

ków zadań do klastrów obliczeniowych. Rolą lokalnych zarządców klastrów (LCRM) jest przydzielenie otrzymanych zadań do konkretnych procesorów obliczeniowych funkcjonujących w obrębie klastra. W pracach [H1, U2] przyjąłem, że koszty transmisji w wewnętrznej sieci klastra są stałe, co pozwala nie uwzględniać ich w lokalnym zadaniu optymalizacji. Założenie to nie musi oznaczać rezygnacji z wykorzystania algorytmów oszczędzania energii w przełącznikach tworzących sieć klastra, jest jedynie wynikiem obserwacji, pokazującej, że, przynajmniej w przypadku małych i średnich klastrów, pobór mocy przez urządzenia sieciowe jest niewielką częścią mocy pobieranej przez jednostki obliczeniowe. Oznacza to, że potencjalne oszczędności nie powinny mieć decydującego znaczenia[3]. Podejście takie pozwala skoncentrować się jedynie na kosztach energetycznych związanych z pracą jednostek obliczeniowych, a przez to uprościć znacznie model matematyczny.

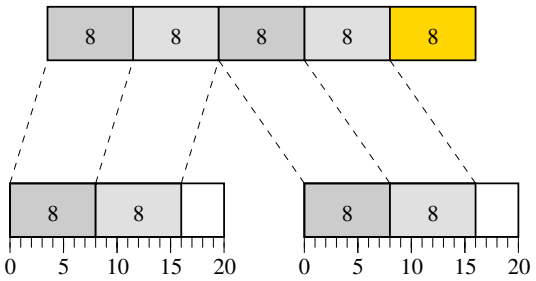


Rysunek 3: Zależność poboru mocy (lewy wykres) i wydajności obliczeniowej (prawy wykres) procesora w kolejnych stanach energetycznych. Stan oznaczony numerem 0 odpowiada uśpieniu procesora, zaś ostatni, K_s-ty stan maksymalnej częstotliwości taktowania.

Tak postawiony problem sprowadza się do alokowania N_j zadań z danego bloku j, $(n = 1, \ldots, N_j)$ do procesorów obliczeniowych klastra f. Zgodnie z przyjętym modelem energetycznym, przedstawionym na rys. 3 każdy procesor może pracować w różnych stanach energetycznych. Optymalna alokacja powinna minimalizować pobór mocy przez klaster, przy założeniu spełnienia warunków na jakość usług, których realizacja wymaga zapewnienia niezbędnej mocy obliczeniowej i pamięci operacyjnej wszystkim wykonywanym zadaniom. Zgodnie z wcześniejszą klasyfikacją, takie rozumienie jakości obsługi odpowiada sytuacji ciągłego świadczenia usług o określonych wymaganiach. Matematyczny zapis zadania jest przedstawiony poniżej:

$$\min_{x_s, \vartheta_s} \left[\sum_{s=1}^{S_f} \sum_{k=1}^{K_s} P_s^k x_s^k \right], \tag{28}$$

[3]Przykładowy szacunek: klaster złożony z 32 dwuprocesorowych serwerów pobiera w warunkach pełnego obciążenia około 16,6 kW (2 przełączniki QFX5100 – pobór maksymalny 335 W, 32 serwery Dell R740 w średniej specyfikacji – typowy, obserwowany pobór mocy pod pełnym obciążeniem wynosi około 500 W). W przypadku zmniejszonego obciążenia możliwe jest okresowe usypianie portów przełączników prowadzących do wybranych maszyn, co może skutkować oszczędnością rzędu 2-4 W na port, czyli przy 32 maszynach i redundantnych połączeniach do obu przełączników jest to nie więcej niż 256 W (źródło – pomiary własne [U7], specyfikacje: https://www.juniper.net/us/en/products/switches/qfx-series/qfx5100-ethernet-switch-datasheet.html, https://i.dell.com/sites/csdocuments/Shared-Content_data-Sheets_Documents/en/poweredge-r740-spec-sheet.pdf).



Rysunek 4: Ilustracja problemu niepodzielności zasobów: górny wiersz pokazuje zadania i wymagane przez nie zasoby, dolny zasoby dostępne na poszczególnych procesorach. Mimo iż suma zasobów jest wystarczająca, zadanie oznaczone kolorem żółtym nie może być wykonane.

$$\forall_{s=1,\ldots,S_f} \quad \sum_{k=1}^{K_s} x_s^k = 1, \tag{29}$$

$$\forall_{n=1\ldots,N_j} \quad \sum_{s=1}^{S_f} \vartheta_{sn} = 1, \tag{30}$$

$$\forall_{s=1,\ldots,S_f} \quad \sum_{n=1}^{N_j} W_n \vartheta_{sn} \leqslant \sum_{k=1}^{K_s} \Theta_s^k x_s^k, \tag{31}$$

$$\forall_{s=1,\ldots,S_f} \quad \sum_{n=1}^{N_j} M_n \vartheta_{sn} \leqslant \sum_{k=1}^{K_s} \Psi_s x_s^k, \tag{32}$$

gdzie zmienne x_s^k i ϑ_{sn} przyjmują wartość 0 lub 1. Zmienna $x_s^k = 1$ oznacza, że procesor s działa w stanie energetycznym k, a $\vartheta_{sn} = 1$ oznacza, że zadanie n jest wykonywane przez procesor s. Zapotrzebowanie na moc obliczeniową i pamięć operacyjną zadania n definiują stałe W_n i M_n, zaś Θ_s^k i Ψ_s to moc obliczeniowa w stanie energetycznym k i pamięć operacyjna związane z procesorem s.

Ograniczenie (29) gwarantuje, że każdy procesor może w danej chwili działać tylko w jednym stanie energetycznym, (30), że każde zadanie jest przypisane do procesora, a (31) i (32) zapewniają, że procesory, do których są przydzielone zadania, mają wystarczające zasoby do ich wykonania.

4.4.3 Algorytm alokacji zasobów

W pracy [H2] przedstawiłem schemat iteracyjnego algorytmu alokacji, czyli sposobu, w jaki GCRM koordynuje działania lokalnych zarządców klastrów. Należy zwrócić uwagę na fakt, że koordynacja nie oznacza bezpośredniego ingerowania w działanie klastrów, a jedynie zlecanie im do wykonania pewnych zbiorów zadań, których przypisanie wynika z określonych wcześniej i zgłoszonych do GCRM zasobów. Zgodnie z tym, co wspomniałem w poprzednich punktach, wiedza GCRM na temat budowy i sposobu działania klastrów może nie być pełna. Stosowane są tu uproszczone modele, takie jak np. (17)-(27), w którym liczba stanów energetycznych została zredukowana do dwóch (aktywny – wyłączony), a zapotrzebowanie na zasoby (moc obliczeniowa i pamięć) jest rozumiane jako suma zasobów wymaganych przez poszczególne zadania. W wyniku tego GCRM podejmuje decyzję o wykorzystaniu pewnej liczby procesorów klastra, nie zaś o przypisaniu zadań do konkret-

nych procesorów. Podejście takie jest atrakcyjne, tak ze względu na uproszczenie zadania obliczeniowego, jak również pozostawienie pewnej swobody nadzorcy klastra, który może dokonać alokacji zgodnie ze swoimi celami, znając aktualną sytuację w klastrze. Operuje on dokładniejszym modelem (patrz (28)-(32)), który uwzględnia możliwość przełączania procesorów w stany energetyczne obniżające pobór mocy. LCRM musi także przypisać zadania do konkretnych procesorów, co pomijał koordynator. W pewnych przypadkach może prowadzić to do niezgodności ograniczeń zasobowych rozważanych na poziomie koordynatora (19)-(20) z występującymi w zadaniu lokalnym (31-32). Ze względu na fakt, że zastosowany schemat koordynacji jest przykładem metody bezpośredniej [20], może to uniemożliwić rozwiązanie zadania LCRM, a co się z tym wiąże uniemożliwić działanie systemu. Innymi słowy, koordynator, posługując się przybliżonym modelem klastrów, może narzucić wartości zmiennych koordynujących (czyli alokacje zadań do klastrów), dla których nie istnieje rozwiązanie zadania lokalnego. Problem ten jest znany z literatury [40, 20]. Typowym rozwiązaniem, które chroni przed jego wystąpieniem, jest wykorzystanie iteracyjnego algorytmu gradientowego uwzględniającego sprzężenie zwrotne od zrelaksowanych zadań lokalnych.

W omawianym przypadku przyczyną komplikacji jest dyskretna natura zasobów powiązanych z konkretnymi procesorami obliczeniowymi, takich jak pamięć operacyjna i moc obliczeniowa, czego prostą ilustrację przedstawia rysunek 4. Z powodu dyskretnej natury zadania nie jest możliwe zastosowanie wspomnianych wcześniej, klasycznych metod optymalizacji. Korzystając z zaprezentowanej interpretacji, można jednak zaproponować metodę heurystyczną, co prawda suboptymalną, ale pozwalającą w racjonalny sposób zorganizować współpracę zarządców klastrów i koordynatora. Metoda ta jest bezpośrednią konsekwencją faktu, że z punktu widzenia zarządcy klastra problem występuje, gdy zasoby przeznaczone do wykonania zleconych przez koordynatora zadań są niewystarczające. W takiej sytuacji zarządca klastra może podjąć jedną z trzech decyzji:

1. wykonać zadania pomimo niewystarczających zasobów, licząc się z niedotrzymaniem wymagań na jakość,

2. przeznaczyć więcej zasobów na wykonanie zadań,

3. zgłosić koordynatorowi konieczność realokacji zadań.

Z zastosowaniem pierwszych dwóch podejść wiąże się pewne ograniczenie. Pogorszenie jakości nie zawsze jest dopuszczalne, zaś powiększenie zasobów często niemożliwe. Trzeba jednak pamiętać, że koordynator nie dysponuje dokładnym modelem klastra, zaś zarządca klastra nie musi przeznaczać wszystkich zasobów do wykonania zlecanych zdań. Stąd możliwe jest występowanie pewnych rezerw. Z punktu widzenia niezawodności i dostępności systemu obliczeniowego najlepszym rozwiązaniem jest trzeci wariant. W pracy [H1] zaproponowałem dwuetapowy algorytm. W pierwszym etapie dokonuje się estymacji brakujących zasobów, w drugim korekty alokacji. Faza pierwsza jest wykonywana przez LCRM i sprowadza się do iteracyjnego rozwiązania zrelaksowanego zadania lokalnego w celu określenia stopnia przekroczenia ograniczeń na zasoby. W tym celu w zadaniu

(28)-(32) ograniczenia (31) i (32) należy zastąpić następująca parą ograniczeń:

$$\forall_{s=1,\ldots,S_f} \quad \sum_{n=1}^{N} W_n \vartheta_{sn} \leqslant \sum_{k=1}^{K_s} \Theta_s^k \bar{W}^k x_s^k, \tag{33}$$

$$\forall_{s=1,\ldots,S_f} \quad \sum_{n=1}^{N} M_n \vartheta_{sn} \leqslant \sum_{k=1}^{K_s} \bar{M} \Psi_s, \tag{34}$$

przy czym współczynniki $\bar{W}^k$ i $\bar{M}$ początkowo są równe 1. Iteracyjna relaksacja polega na stopniowym zwiększaniu wspomnianych współczynników, aż do znalezienia rozwiązania dopuszczalnego. Następnie sprawdzany jest poziom przekroczenia zasobów:

$$E_f^{\Theta} = \sum_{s=1}^{S_f} \sum_{n=1}^{N} W_n \vartheta_{sn} - \sum_{s=1}^{S_f} \sum_{k=1}^{K_s} \Theta_s^k x_s^k, \tag{35}$$

$$E_f^{M} = \sum_{s=1}^{S_f} \sum_{n=1}^{N} M_n \vartheta_{sn} - \sum_{s=1}^{S_f} \Psi_s, \tag{36}$$

gdzie E_f^{Θ} to suma przekroczeń wszystkich ograniczeń dotyczących mocy obliczeniowej (czyli ograniczeń odpowiadających kolejnym procesorom). E_f^{M} to sumaryczne przekroczenie ograniczeń dotyczących pamięci operacyjnej. W przypadku, gdy wykonanie zadań nie jest możliwe, czyli nie ma dostępnej rezerwy zasobów, a niedotrzymanie deklarowanej jakości obsługi jest niedopuszczalne wyznaczone wartości E_f^{Θ} i E_f^{M} są raportowane do zarządcy chmury, gdzie są wykorzystywane do modyfikacji ograniczeń zasobowych, tj. zmniejszenia dostępnych zasobów klastra występujących po prawej stronie ograniczeń (19) i (20). Należy zwrócić uwagę, że opisywany proces jest iteracyjny, tzn. zmniejszenie zasobów dla jednego z klastrów powoduje konieczność przeniesienia części z poprzednio przypisanych mu zadań, co może powodować konieczność korekcji w pozostałych klastrach. Innymi słowy, w kolejnych iteracjach korekta jest wykonywana lokalnie i dotyczy klastrów zgłaszających koordynatorowi brak zasobów. Z tego też powodu opisany algorytm nie gwarantuje uzyskania optymalnego rozwiązania, jednakże wyniki badań symulacyjnych pokazują istotną poprawę efektywności energetycznej przy znacznym zmniejszeniu czasu obliczeń, co uzasadnia stosowanie rozwiązania suboptymalnego.

Heurystyczny algorytm alokacji zasobów klastra. Opisany wcześniej schemat koordynacji wymaga wielokrotnego rozwiązania przez LCRM zadania lokalnego. Z tego powodu jednym ze sposobów przyspieszenia działania systemu jest zastosowanie uproszczonego sposobu wyznaczania alokacji zadań do procesorów klastra. Metoda heurystyczna, którą zaproponowałem w pracy [H1] jest rodzajem algorytmu zachłannego. W algorytmie tym zadania są sortowane względem ich zapotrzebowania na zasoby, a następnie przypisywane do kolejnych procesorów klastra. Postępowanie takie ma na celu upakowanie zadań na, w miarę możliwości, jak najmniejszej liczbie procesorów, co powinno pozwolić na wyłączenie pozostałych maszyn. Ze względu na fakt, że zgodnie z przyjętym modelem zadania, brane są pod uwagę dwa rodzaje zasobów, tj. pamięć operacyjna i moc obliczeniowa procesora, na wstępie szacowane jest sumaryczne zapotrzebowanie wszystkich

zadań na zasoby

$$U^{\Theta} = \sum_{n=1}^{N} W_n, \tag{37}$$

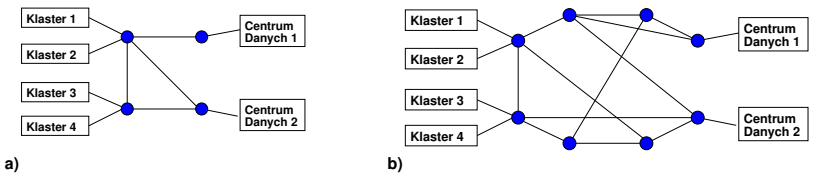
$$U^{M} = \sum_{n=1}^{N} M_n, \tag{38}$$

gdzie w sytuacji, gdy $U^{\Theta} > U^{M}$ zadania są sortowane ze względu na zapotrzebowanie na moc obliczeniową. W przeciwnym przypadku sortowanie odbywa się według, deklarowanej przez zadania, wielkości pamięci. Następnie zadania są przypisywane kolejno do procesorów klastra. Ostatnim krokiem algorytmu jest dostosowanie stanów energetycznych procesorów do nałożonego na nie obciążenia. Procesory, do których nie alokowano żadnych zadań (o ile takie występują), są wyłączane, zaś procesory obciążone poniżej maksymalnej mocy obliczeniowej wprowadzane w najniższy, dopuszczalny (tj. zapewniający wystarczającą zdolność przetwarzania) stan energetyczny.

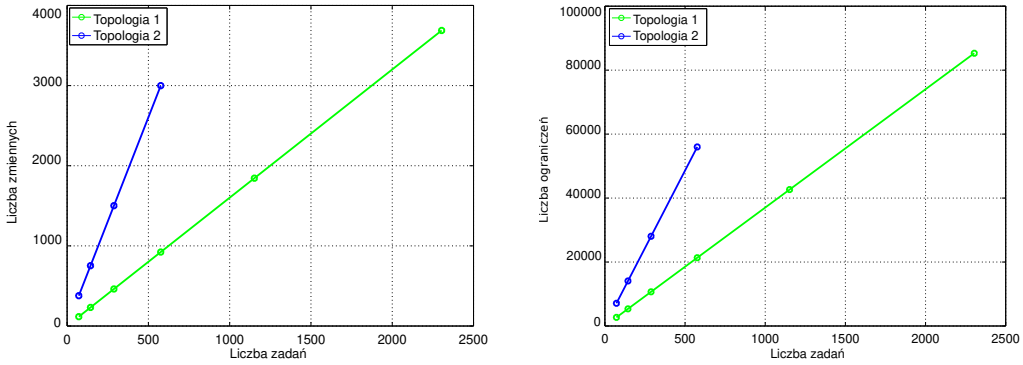
Jest to algorytm suboptymalny, jednakże jego złożoność obliczeniowa jest zazwyczaj znacznie mniejsza niż rozwiązanie zadania programowania mieszanego (28)-(32). Szacując złożoność należy uwzględnić sortowanie zadań – typowo $O(ln(N))$, liczbę operacji potrzebnych do wyznaczenia sumarycznych zapotrzebowań na zasoby (37) i (38) oraz obciążenia i stanów energetycznych procesorów w trakcie alokacji. Ostatecznie złożoność obliczeniowa to $O(N^2 S_f + NKS_f)$. Opisane dalej eksperymenty numeryczne potwierdzają celowość stosowania algorytmu zachłannego dla wybranych przypadków.

4.4.4 Eksperymentalna ocena rozwiązań

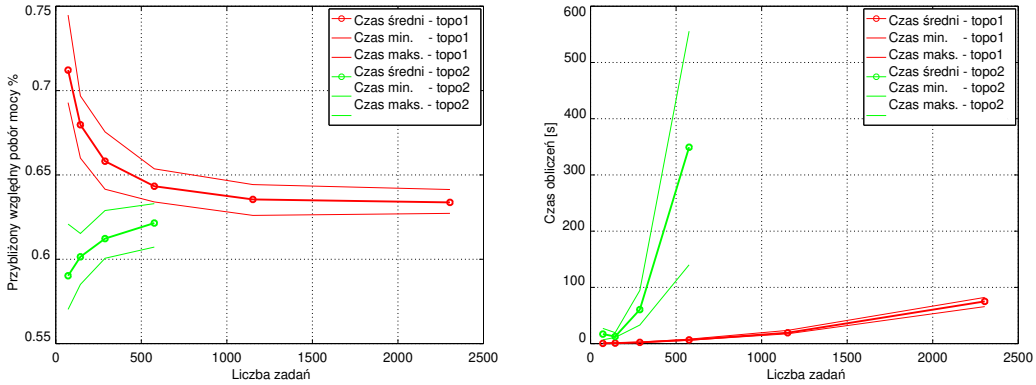
Efektywność i przydatność opracowanych algorytmów została sprawdzona przez wykonanie licznych eksperymentów, wyniki, których przedstawiłem w pracach [H1, H2]. Problem (17)-(27) rozwiązywany przez koordynatora stanowi poważne wyzwanie dla metod optymalizacji. Jak pokazałem w pracy [H2] wymiar tego zadania może być znaczny już dla problemów średniej wielkości. Pozytywną cechą jest natomiast liniowe skalowanie wymiaru ze wzrostem liczby zadań. Wykresy pokazane na rys. 6 obrazują zmianę liczby zmiennych oraz liczby ograniczeń wraz ze wzrostem liczby rozdzielanych między klastry zadań obliczeniowych dla dwóch topologii pokazanych na rys. 5. Z tego powodu istotne było sprawdzenie efektywności i przydatności tak uzyskanych rozwiązań. Miarą efektywności była osiągana redukcja zapotrzebowania na moc. W badanym przypadku porównywałem zapotrzebowanie na moc osiąganą przy realizacji wyznaczonej alokacji z zapotrzebowaniem wynikającym z utrzymania sieci klastrów w maksymalnych stanach energetycznych, czyli przygotowanej na przetwarzanie dowolnego zestawu danych. Miarą przydatności w tym przypadku jest przede wszystkim czas obliczeń. Wykresy przedstawione na rys. 7 wskazują, że czas ten silne zależy od topologii sieci łączącej klastry. W istocie to topologia sieci



Rysunek 5: Analizowane topologie chmury obliczeniowej: a) topologia 1, b) topologia 2.



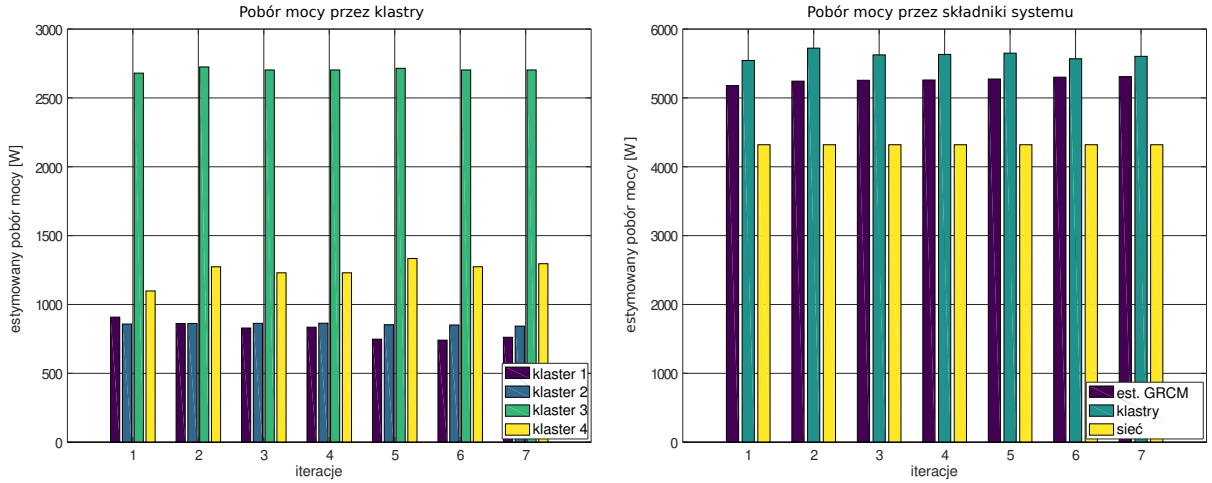
Rysunek 6: Zależność liczby zmiennych i ograniczeń od liczby zadań dla analizowanych topologii.



Rysunek 7: Estymaty poboru mocy w zależności od liczby alokowanych zadań (wykres lewy) oraz czasy obliczeń (wykres prawy) dla dwóch analizowanych topologii.

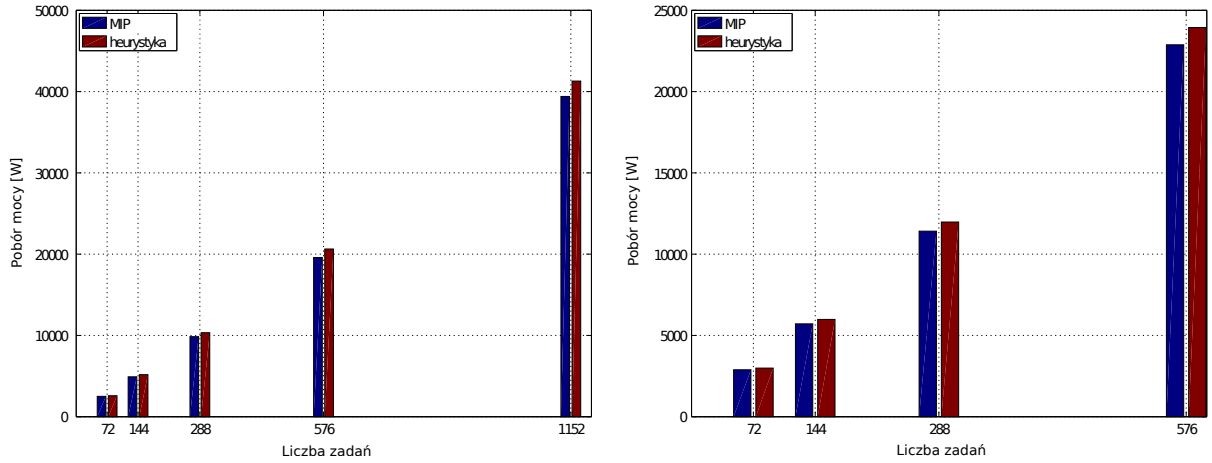
decyduje o stopniu komplikacji zadania obliczeniowego, co widać na rysunku 6. Oznacza to, że stosowalność proponowanego podejścia jest ograniczona do prostych topologii lub też niezbyt dużej liczby alokowanych zadań. Wykres przedstawiający estymowany pobór mocy – wartość względną w stosunku do wspomnianego wcześniej scenariusza bez energooszczędnego sterowania – pokazuje inną istotną cechę zadania i samego systemu. Otóż w przypadku mało licznych zadań oszczędności osiągane dla dwóch badanych topologi są zasadniczo różne. Mniej korzystna, i pozornie zaskakująca, sytuacja obserwowana dla sieci o złożonej topologi jest oczywistą konsekwencją większego, w tym przypadku, udziału infrastruktury sieciowej w całkowitym poborze mocy. Wynik ten można interpretować jako zalecenie, aby małe zadania, o ile tylko dostępność danych na to pozwala, realizować w prostych, możliwie skupionych konfiguracjach, albo korzystać ze współdzielonej, nie zaś dedykowanej, infrastruktury sieciowej, aby maksymalizować jej wykorzystanie.

Iteracyjny algorytm alokacji. Opisane wcześniej badania dotyczyły zadania alokacji rozwiązywanego przez GCRM. Należy jednak pamiętać, że proponowany rozdział zadań do klastrów jest wykonywany w sposób iteracyjny. Wynika to z konieczności określenia dopuszczalnego zbioru zmiennych koordynacyjnych. Z tego powodu przedstawione wcześniej wyniki są jedynie optymistycznymi oszacowaniami potencjalnych oszczędności energetycz-



Rysunek 8: Pobór mocy przez poszczególne klastry (z lewej) oraz składniki systemu (z prawej) podczas iteracji algorytmu rozdziału zadań.

nych, zaś pełna ocena skuteczności wymaga przeprowadzenia symulacji działania całego systemu. Interesujący jest tu przede wszystkim sam przebieg procesu uzgadniania alokacji, a konkretnie ile iteracji trzeba wykonać dla osiągnięcia dopuszczalnej alokacji i jak liczne korekty są niezbędne. Trzeba bowiem pamiętać, że każda korekta jest potencjalnym odstępstwem od wyznaczonej pierwotnie, optymalnej alokacji. Wyniki przedstawione w [H1], wskazują, że proces ten nie musi być bardzo kosztowny, tak w sensie liczby iteracji, jak też dodatkowej mocy wynikającej z konieczności uruchomienia dodatkowych maszyn. Przykład zilustrowany wykresami z rysunku 8 wskazuje, że konieczne relokacje zadań nie

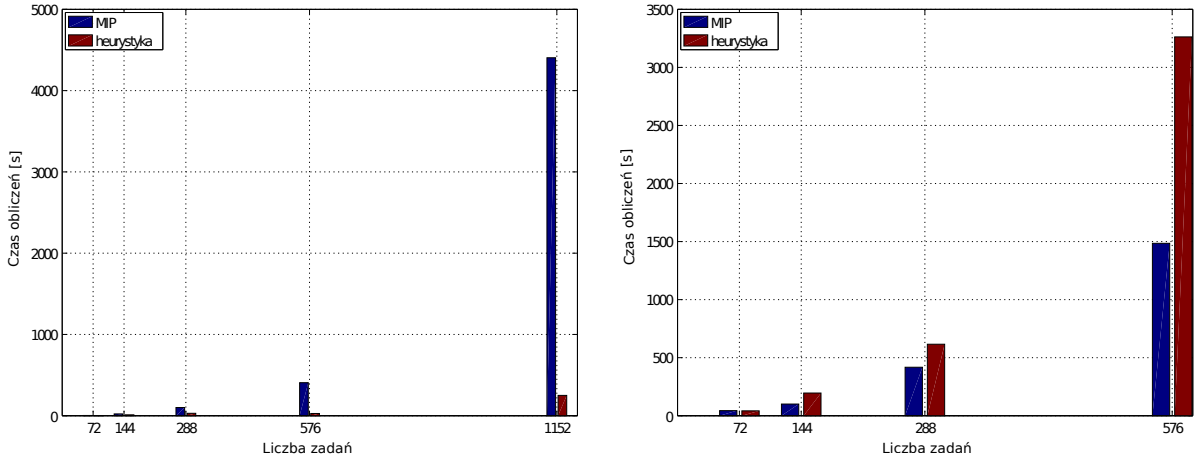


Rysunek 9: Całkowity pobór mocy przez system chmurowy dla dwóch topologii sieci i zmieniającej się liczby alokowanych zadań.

muszą powodować dużych zmian poboru mocy, gdyż zwiększenie obciążenia części z klastrów jest rekompensowanie przez spadek obciążenia innych. Oznacza to, że ewentualny przyrost całkowitego poboru mocy wynika zazwyczaj z mniejszej efektywności klastrów, na które muszą być przeniesione zadania. Co więcej, pobierana moc jest około 15% więk-

sza niż wynosiła początkowa estymata wynikająca z rozwiązania zadania koordynatora, a pobór mocy przez sieć pozostaje na niezmienionym poziomie. Ostatnia z obserwacji jest o tyle istotna, że pozwala myśleć o zaprojektowaniu algorytmów heurystycznych, w których realokacja zadań mogłaby być oddzielona od bardzo czasochłonnego zadania wyznaczenia rozpływu w sieci łączącej klastry.

Całkowity pobór mocy dla zmieniającej się liczby zadań przedstawiają wykresy widoczne na rysunku 9, zaś średni czas obliczeń ilustrują wykresy z rysunku 10. Analizowane były oba algorytmy rozwiązywania zadania LCRM.



Rysunek 10: Średni czas obliczeń dla dwóch topologii sieci i zmieniającej się liczby alokowanych zadań.

4.5 Techniki zwiększenia wydajności energetycznej sieci komputerowych

Nowoczesne sieci komputerowe umożliwiają przesyłanie danych z wielkimi prędkościami, gwarantują przy tym spełnienie wymagań jakościowych określonych w umowie zawartej z klientem (tzw. SLA – *service level agreement*). Podstawowym środkiem technicznym dla zapewnienia tych wymagań jest odpowiedni zapas dostępnego pasma na wykorzystywanych łączach i przygotowanie struktury sieci charakteryzującej się dostatecznym poziomem redundancji. Redundancja może być realizowana na poziomie łączy, przez zwielokrotnienie linii pomiędzy tymi samymi lokalizacjami, ale również na poziomie sieci, dzięki drogom obejściowym prowadzącym przez różne węzły pośrednie. Wadą wspomnianych technik jest zazwyczaj niepełne wykorzystanie przepustowości sieci, co wprowadza dodatkowe koszty związane m.in. z nadmiernym poborem mocy. Podstawową metodą zmniejszenia poboru mocy jest wykorzystanie nowoczesnych urządzeń sieciowych wyposażonych w całą gamę lokalnych mechanizmów zwiększających wydajność energetyczną, takich jak np. IEEE 803.2az, czyli *Energy Efficient Ethernet*. Mechanizmy te pozwalają uzyskać znaczące oszczędności, ale nie wykorzystują one całkowitego potencjału wynikającego przede wszystkim z istnienia redundancji.

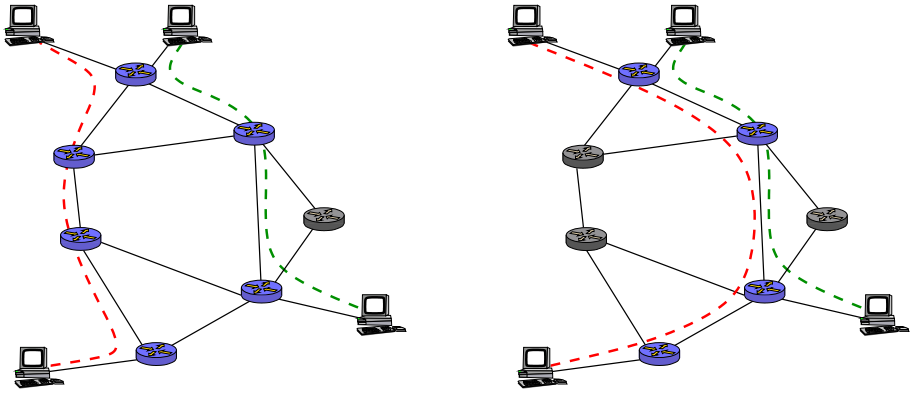
Prowadzone przeze mnie prace w zakresie energooszczędnych sieci komputerowych obejmowały dwa kierunki działań:

- Optymalizację stanów energetycznych urządzeń sieciowych (ang. activity control).
- Propozycję heurystycznych algorytmów trasowania dla sieci łączących jednostki obliczeniowe.

Znaczna część prac była prowadzona w większym zespole realizującym projekt ECONET. Uczestniczyłem w opracowaniu i eksperymentalnej weryfikacji rozwiązań. Mój główny wkład był związany z tworzeniem środowisk sprzętowo-programowych do badań, opracowaniem scenariuszy testowych i prowadzeniem eksperymentów mających na celu ocenę stosowalności i wydajności algorytmów opracowanych w ramach projektu.

4.5.1 Scentralizowana architektura sterowania przepływami w sieci

Ze względu na sposób wykorzystania urządzeń sieciowych charakterystyki poboru mocy przez urządzenia (prezentowane w sekcji 4.7) są zazwyczaj funkcjami wklęsłymi, odznaczającymi się wyraźnym skokiem związanym z włączeniem. Oznacza to, że konsolidacja przepływów na ograniczonym podzbiorze łączy (zilustrowana na rysunku 11) może prowadzić do istotniejszego zmniejszenia poboru mocy niż w przypadku, gdy wykorzystane są jedynie wbudowane mechanizmy oszczędzania energii. Warunkiem jest jednak możliwość wyłączenia, czy też wprowadzenia w stan niskiego poboru mocy, części łączy i urządzeń w sposób niepogarszający jakości obsługi poniżej gwarantowanego w umowie poziomu.



Rysunek 11: Rozpływ ruchu możliwy do otrzymania za pomocą algorytmów lokalnych i centralnych. Konsolidacja przepływów pozwala wyłączyć więcej urządzeń (oznaczone kolorem szarym).

Wynikiem powyższych obserwacji jest propozycja dwupoziomowej struktury sterowania aktywnością urządzeń sieciowych, opracowanej i rozwijanej w ramach projektu ECONET. W jej skład wchodzą:

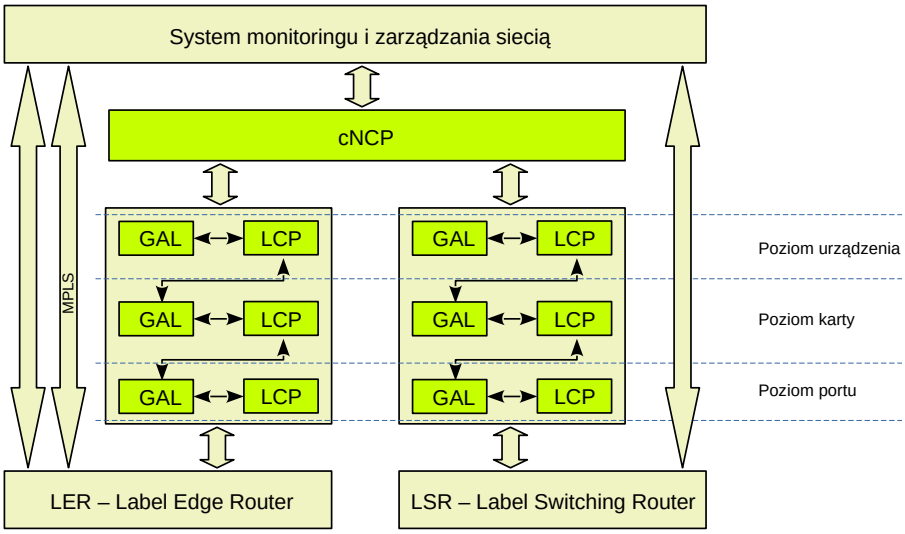
- algorytmy lokalne – sterujące aktywnością urządzeń, zaimplementowane bezpośrednio w urządzeniach sieciowych,
- strategie centralne – pozwalające na koordynację działań urządzeń i gwarantujące zachowanie wymaganej jakości usług.

W podejściu, zaprezentowanym w [H4, U3], wykorzystałem istniejące technologie, narzędzia i algorytmy pozwalające zbudować scentralizowany system sterujący siecią w sposób zapewniający minimalizację zużycia energii. Wśród nich można wymienić:

- system zarządzania i monitoringu,
- protokół MPLS[4] (ang. Multiprotocol Label Switching).

Są to rozwiązania dostępne również w sieciach, które nie są wyposażone w funkcje związane z energooszczędnym sterowaniem. Oprócz tego niezbędne są mechanizmy wspierające sterowanie zużyciem energii, takie jak:

- lokalne mechanizmy sterowania urządzeniami sieciowymi,
- mechanizm pozyskiwania charakterystyk urządzeń, w tym wypadku opracowany w ramach projektu ECONET interfejs GAL (Green Abstraction Layer) [3, 12, 13].



Rysunek 12: Scentralizowana architektura sterowania przepływami w energooszczędnej sieci komputerowej.

Dzięki wykorzystaniu wymienionych mechanizmów możliwe jest wpływanie na wydajność i pobór mocy poszczególnych urządzeń. Należy podkreślić rolę zunifikowanego interfejsu dostarczającego informacji o energooszczędnych funkcjach urządzeń, jakim jest GAL. Zawiera on pełny opis charakterystyk sprzętu w rozbiciu na jego poszczególne składniki. Uwzględnia dzięki temu złożoną, zazwyczaj hierarchiczną, budowę urządzeń, co jest podstawą opisanego dalej modelu energetycznego węzła sieciowego. Drugą ważną funkcją interfejsu GAL jest sterowanie stanami energetycznymi urządzeń. Funkcja ta jest wykorzystywana przez sterowniki zlokalizowane w urządzeniach.

Zespół Politechniki Warszawskiej, którego byłem członkiem, opracował dwupoziomową strukturę sterowania aktywnością urządzeń i przepływem danych w sieci. Jest ona opisana w pracy [H4] i zaprezentowana na rysunku 12. Przyjmuje się, że jednostka nadrzędna sieci cNCP (central Netwok Control Policy) wyznacza optymalne energetycznie trasy, którymi przesyłany jest ruch sieciowy oraz ustawia odpowiadające im, optymalne

[4]MPLS jest techniką pozwalająca na wydajną inżynierię ruchu dzięki zastosowaniu etykiet umożliwiających zbudowanie wirtualnej topologii ponad fizyczną siecią. Jest ona opisana w RFC 3031 i 3032 (https://datatracker.ietf.org/doc/html/rfc3031, https://datatracker.ietf.org/doc/html/rfc3032)

stany energetyczne urządzeń sieciowych. Wejściem sterownika są dane ruchowe odbierane z systemu monitoringu i zarządzania siecią, wykorzystywane następnie do wyznaczenia prognoz zapotrzebowań pasma na poszczególnych relacjach w sieci. Jako mechanizm inżynierii ruchu pozwalający na zestawienie tras przyjęto MPLS. Należy podkreślić, że wykorzystanie pośrednictwa systemu zarządzania siecią zapewnia abstrakcję od konkretnej technologii, o ile tylko oferuje ona wystarczającą funkcjonalność. Z tego powodu nie istnieją przeszkody, aby wykorzystać inne metody inżynierii ruchu, w szczególności właściwe dla sieci SDN (ang. Software Defined Network – programowalna sieć komputerowa[5]). Wykorzystanie MPLS uproszcza konstrukcję sterownika, który komunikuje się tylko z ruterami brzegowymi (LER – Label Edge Router), zaś rutery wewnątrz sieci (LSR – Label Switching Router) pozyskują informacje przez protokół MPLS od sąsiadów. Rola sterowników urządzeń LCP (Local Control Policy) sprowadza się w tym podejściu przede wszystkim do raportowania stanu komponentów na wszystkich poziomach urządzenia i egzekwowania sterowań otrzymanych od cNCP, z zachowaniem ograniczeń narzuconych przez sprzęt.

Należy podkreślić, że dodanie scentralizowanego sterownika nie zaburza podstawowego funkcjonowania sieci, opartego o rozproszone algorytmy, takie jak ruting czy sterowanie przepływem TCP, a co się z tym wiąże, nie obniża jej niezawodności. Prezentowaną architekturę można traktować jako rozwinięcie koncepcji będących podstawą konstrukcji takich rozwiązań jak brokery pasma [32, 46, 37] czy inżynieria ruchu.

4.5.2 Hierarchiczna architektura sterowania przepływami w sieci

Hierarchiczna architektura prezentowana w pracach [H4, H5, U4] jest rozwinięciem wersji scentralizowanej. Struktura rozważanej sieci nie uległa zmianie, nadal dostępny jest system zarządzania i monitoringu dostarczający statystyk i pozwalający zestawiać trasy z wykorzystaniem protokołu MPLS. Zadaniem jednostki nadrzędnej sieci (hNCP – hierarchical Network Control Policy) jest wyznaczenie optymalnych, w sensie poboru mocy, tras. Oznacza to, że hNCP nie steruje stanami energetycznymi urządzeń, pozostawiając tę decyzję sterownikom lokalnym. Oczywiście nadrzędny zarządca nadal musi być świadomy modelu energetycznego urządzeń, w tym stanów energetycznych, stąd konieczność pozyskiwania tych danych. Takie podejście pozwala zmniejszyć złożoność obliczeniową zadania sterowania realizowanego przez jednostkę nadrzędną dzięki zmniejszeniu częstotliwości interwencji i uproszczeniu modelu. Zwiększa również odporność na opóźnienia i błędy komunikacji.

Mniejsza częstotliwość interwencji wynika z założenia, że trasy, którymi przesyłany jest ruch, zmieniają się rzadziej niż stany energetyczne urządzeń. Mają na to wpływ dwa fakty: fizyczne ograniczenie, które czyni nieefektywnym zestawianie tras częściej niż co kilka-kilkanaście minut[6], oraz możliwość dostosowania wydajności urządzeń do aktualnego poziomu ruchu przez zmianę stanu energetycznego przez sterownik lokalny. Sterownik lokalny rozwiązuje zadanie dotyczące pojedynczego urządzenia, a więc znacznie prostsze od zadania sterowania siecią, przez co decyzje mogą być częściej aktualizowane. Dodatko-

[5]SDN jest architekturą pozwalającą, dzięki wyizolowaniu warstwy sterowania (Control Plane), na zarządzanie urządzeniami sieciowymi w sposób zunifikowany w celu m.in. budowy wirtualnych topologii umożliwiających przesył danych po ustalonych trasach. Rozwiązanie to jest promowane przez Open Networking Fundation (https://opennetworking.org/) i w wybranych fragmentach implementowane w wielu urządzeniach, systemach operacyjnych i klastrowych np. OpenStack.

[6]Zestawienie trasy przy pomocy systemu zarządzania siecią może trwać do kilku sekund, por. m.in. wyniki testów omawianego w rozdziale 5 systemu IPQoS, którego jestem współautorem [B7].

wo, z racji położenia w pobliżu sterowanego sprzętu, lub wręcz wbudowaniu, wyznaczone sterowania mogą być znacznie szybciej wdrożone. Tak zdefiniowany hierarchiczny schemat sterowania przepływami pozwala więc dostosować częstość interwencji sterowników do ograniczeń. Osobną korzyścią, wynikającą z lokalnego wyznaczania stanów energetycznych urządzeń, jest możliwość uproszczenia modelu energetycznego wykorzystywanego w zadaniu sterowania siecią poprzez zmniejszenie liczby rozpatrywanych stanów energetycznych[7], czy też ich aproksymację, np. uciąglenie.

Zadanie optymalizacji energooszczędnego sterowania siecią. Zadanie programowania matematycznego rozwiązywane przez sterownik sieci (NCP) może, w ogólnej postaci, być niezwykle złożone obliczeniowo. Wynika to z konieczności wyznaczenia tras minimalizujących zużycie energii i zapewniających transmisje danych na wymaganych relacjach. Dla praktycznego wykorzystania konieczne jest, aby czas rozwiązania tego zadania był krótszy niż czas wyznaczania sterowania. Propozycja uproszczenia, a przez to zwiększenia efektywności obliczeniowej, przedstawiona w [H4] sprowadza się, do zastąpienia procesu wyznaczania optymalnych tras wyborem z wcześniej dostarczonego, predefiniowanego zbioru tras. Zaletą takiego postępowania jest znaczne uproszczenie zadania obliczeniowego, uzyskane dzięki zmniejszeniu jego wymiarowości. Zbiór tras może być wyznaczony wcześniej, typowo z użyciem metod heurystycznych. Otrzymane w ten sposób rozwiązanie jest w oczywisty sposób suboptymalne, a odległość od optimum zależy w dużym stopniu od sposobu wyznaczenia tras i ich liczby. Należy jednak pamiętać, że topologie sieci są zazwyczaj projektowane zgodnie ze znanymi schematami. W szczególności sieci łączące maszyny w centrach obliczeniowych mają zazwyczaj hierarchiczną i redundantną topologię. Co więcej, zazwyczaj są dostępne dane historyczne z systemu monitoringu sieci, co ułatwia wyznaczenie zbioru prawdopodobnych tras. Pozwala to wzorować się na obserwowanym dla typowych przypadków rozkładzie ruchu i modyfikować wyznaczane heurystycznie trasy, tak aby zrealizować założone cele.

Formalnie, zadanie to można zapisać następująco:

$$\min_{x_c, y_{ek}, z_r, u_{dh}} \left\{ F_{LP} = \sum_{e=1}^{E} \sum_{k=1}^{K} \xi_{ek} y_{ek} + \sum_{c=1}^{C} W_c x_c + \sum_{r=1}^{R} T_r z_r \right\}, \tag{39}$$

przy ograniczeniach:

$$\forall_{e=1,\ldots,E} \quad \sum_{k=1}^{K} y_{ek} \leqslant 1, \tag{40}$$

$$\forall_{\substack{d=1,\ldots,D,\\ c=1,\ldots,C}} \quad \sum_{h \in P(d)} l_{ch} u_{dh} \leqslant x_c, \tag{41}$$

$$\forall_{\substack{r=1,\ldots,R,\\ c=1,\ldots,C}} \quad g_{rc} x_c \leqslant z_r, \tag{42}$$

$$\forall_{d=1,\ldots,D} \quad \sum_{h \in P(d)} u_{dh} = 1, \tag{43}$$

$$\forall_{e=1,\ldots,E} \quad \sum_{d=1}^{D} \sum_{h \in P(d)} \delta_{edh} V_d u_{dh} \leqslant \sum_{k=1}^{K} M_{ek} y_{ek}, \tag{44}$$

[7] W rozważanym modelu przyjęto dwa stany energetyczne dla procesorów i macierzy przełączającej rutera, choć można się spodziewać, że podobnie jak w ruterze programowym rozwijanym w projekcie ECONET, co najmniej procesor może być wyposażony w szereg stanów energetycznych.

gdzie indeks $h \in P(d)$ oznacza predefiniowaną trasę obsługującą zapotrzebowanie d, a binarna zmienna $u_{dh} = 1$, gdy zapotrzebowanie d używa trasy h, (0 w przeciwnym przypadku), $\delta_{edh} = 1$ jeśli łącze e należące do trasy h jest użyte do przesyłania zapotrzebowania d, (0 w przeciwnym przypadku). Zmienna $y_{ek} = 1$ oznacza, że łącze e jest w stanie energetycznym k, w którym pobór mocy wynosi ξ_{ek}. Podobnie zmienne x_c i z_r oznaczają, że odpowiednio karta liniowa c i ruter r są włączone i pobierają moc, odpowiednio W_c i T_r. Stałe l_{ch} i g_{rc} odwzorowują przypisanie odpowiednio: kart liniowych do tras i kart liniowych do ruterów, zaś stała M_{ek} jest przepustowością łącza e w stanie energetycznym k.

Ograniczenie (40) zapewnia, że łącze może pracować tylko w jednym stanie energetycznym, nierówności (41) i (42) zapewniają włączenie ruterów i kart liniowych niezbędnych do transmisji. Dzięki uwzględnieniu ograniczenia (43) wszystkie zapotrzebowania na transmisje są obsługiwane, zaś (44) uwzględnia ograniczenia pasma łączy. Podsumowując, wynikiem zadania jest zestaw stanów energetycznych łączy oraz tras, minimalizujący pobór mocy przez sieć, przy jednoczesnym spełnieniu wymagań jakościowych. Wymagania te są odwzorowane w ograniczeniu (43) jako wymóg zestawienia wszystkich tras w sposób nienaruszający wydajności sieci oferowanej przy danym zestawie stanów energetycznych, co modeluje ograniczenie (44). Wskazane trasy mogą być następnie zestawione z użyciem np. protokołu MPLS, zaś wykorzystanie wyznaczonych stanów energetycznych zależy od stosowanej wersji sterownika. W przypadku sterowania scentralizowanego (cNCP) są one przekazywane do sterowników lokalnych (LCP). Porównanie złożoności proponowanego podejścia i zadania z wyznaczaniem tras wymaga założenia topologii sieci. Zestawienie wyników oszacowania złożoności obliczeniowej zadania dla wybranych sieci przedstawia tabela 1. Pokazuje ona liczbę zmiennych i ograniczeń w zadaniach z predefiniowanymi trasami i pełnym rutingiem dla sieci złożonych z 20, 50 i 100 węzłów połączonych poprzez 40, 100 i 200 łączy składających się z 12 włókien światłowodowych, które mogą być niezależnie włączane i wyłączone, co zostało odwzorowane w postaci 13 stanów energetycznych.

Tabela 1: Estymowana złożoność dla zadań z predefiniowanymi trasami i pełnym rutingiem.

Liczba węzłów	Liczba	predefiniowane trasy	pełny ruting
20	zmiennych	7300	515300
	ograniczeń	67680	146912
50	zmiennych	18250	3208250
	ograniczeń	415200	913232
100	zmiennych	36500	12816500
	ograniczeń	1650400	3646432

Inne metody uproszczenia zadania energooszczędnego sterowania siecią. W pracach [H5, U5, U6] przedstawiłem szereg wariantów zadania, z których tutaj warto wspomnieć jeszcze dwa:

- dopuszczenie routingu wielościeżkowego,
- uciąglenie modelu energetycznego.

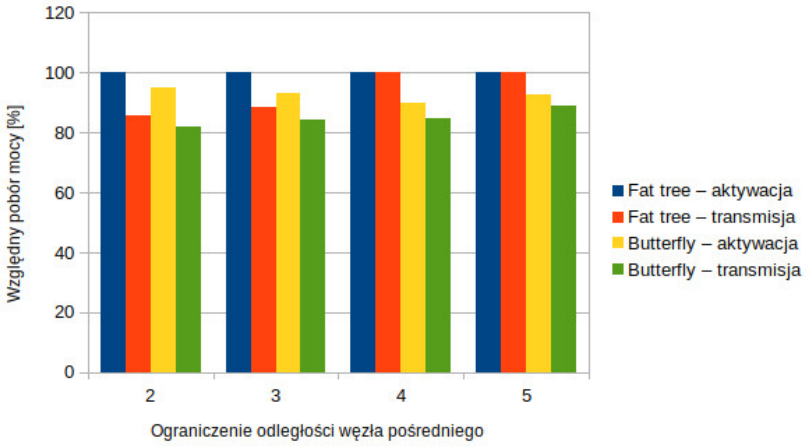
Wariant pierwszy nie był rozwijany w projekcie ECONET ze względu na przyjętą architekturę sieci. Było to zgodne z typową praktyką, dotyczącą w szczególności rozległych sieci TCP/IP, w których przesyłanie danych wieloma trasami postrzegane jest głównie jako źródło błędów wynikających z możliwej zamiany kolejności pakietów. Należy jednak zauważyć, że w architekturach sieciowych stosowanych w centrach danych często wykorzystuje się ruting wielościeżkowy. Możliwość taka wynika ze specyfiki protokołów, jak również z hierarchicznej budowy sieci i realizacji, przynajmniej części, funkcji w 2 warstwie ISO/OSI [18, 4, 44]. Pozwala to unikać błędów szeregowania pakietów, czyniąc znacznie prostsze zadanie wielościeżkowe, bardzo atrakcyjną propozycją. Uproszczenie takiego zadania wynika z możliwości zastosowania zmiennych ciągłych w miejsce binarnych zmiennych opisujących trasy. Wynikowe zadanie nadal wymaga użycia metod właściwych dla zadań mieszanych ze względu na obecność binarnych zmiennych opisujących stan energetyczny urządzeń, jednakże jego złożoność jest znacznie mniejsza.

Uciąglenie modelu energetycznego pozwala, przy zachowaniu trasowania jednościeżkowego, zmniejszyć wymiarowość zadania kosztem pominięcia wyznaczania precyzyjnych nastaw przekazywanych do urządzeń. Z tego powodu podejście to jest właściwe dla hierarchicznego sterownika sieci (hNCP), który model energetyczny wykorzystuje jedynie do oszacowania wydajności i poboru mocy urządzeń, a sterowanie stanami energetycznymi pozostawia sterownikom lokalnym (LCP). Szacunkowe porównanie złożoności obliczeniowej wspomnianych podejść zamieściłem w pracy [U6]. Dodatkowo zbudowałem środowisko badawcze, w którym porównałem efektywność energetyczną proponowanych struktur sterowania przy założonych wymaganiach na jakość usług. Wyniki przedstawiłem w pracy [H5].

4.5.3 Heurystyczne algorytmy trasowania dla sieci centrów danych

Sieci stosowane do połączenia maszyn tworzących klastry obliczeniowe (ang. DCN – data center networks) odznaczają się specyficznymi, regularnymi i symetrycznymi topologiami skutkującymi występowaniem między węzłami wielu tras o tej samej długości. Z tego powodu stosowanie rutingu wielościeżkowego jest w nich łatwiejsze i powszechnie akceptowane, a nawet zalecane, tak ze względu na zwiększenie dostępnej przepustowości, jak również zapewnienie wysokiej niezawodności (ang. HA – high availability). Przykładami takich topologii mogą być Fat Tree oraz Butterfly [4, 30]. W obu z nich występują redundantne połączenia zwiększające niezawodność, ale także pobór mocy. Stosowane w tych sieciach algorytmy rutingu starają się zwiększać niezawodność i równoważyć obciążenie łączy przez np. losowy wybór węzła pośredniego, jak ma to miejsce w algorytmie Valiant routing [42]. Sytuacja taka daje możliwość osiągnięcia znaczących oszczędności energetycznych w okresach niepełnego obciążenia sieci. W pracy [H6] zaproponowałem podejście bazujące na modyfikacji algorytmu Valiant routing w wariancie z ograniczeniem zbioru, z którego wybierane są węzły pośrednie [10]. W klasycznym algorytmie węzeł pośredni jest wybierany losowo i może znajdować się w dowolnym miejscu sieci. Podejście takie nie sprawdza się zbyt dobrze w sieciach o topologii wielokrotnego drzewa (np. Fat tree), ze względu na możliwość wylosowania węzła położonego w innej gałęzi sieci niż węzeł początkowy i końcowy trasy. Z tego powodu w wariancie zmodyfikowanym zbiór kandydatów węzłów pośrednich jest ograniczany do węzłów znajdujących się w pewnej odległości od węzła początkowego. Dzięki temu ogranicza się liczbę tras wykorzystujących węzły położone wysoko w hierarchii. W algorytmie opisanym w pracy [H6] zaproponono-

wałem dwie heurystyki uwzględniające pobór mocy przez sieć. Pierwsza z nich wybiera następne węzły, kierując się ich efektywnością energetyczną. Oznacza to, że w przypadku, gdy dostępne jest kilka węzłów, wybrany zostanie ten, którego efektywność energetyczna jest największa, pod warunkiem że nie spowoduje to przeciążenia łącza. Druga heurystyka umożliwia wyłączanie nieużywanych węzłów po upływie określonego czasu bezczynności. Skuteczność obu heurystyk została sprawdzona z wykorzystaniem symulacji przeprowadzonej z użyciem zmodyfikowanego pakietu NetBench[8]. Symulacje wykazały, że zastosowanie obu heurystyk pozwala obniżyć zużycie mocy i przez to zbliżyć efektywność Valiant routing do klasycznego rutingu ECMP[9], przy zachowaniu nieco lepszego rozrzucenia tras między węzłami, a więc potencjalnie większej niezawodności. Ilustruje to rysunek 13 prezentujący pobór mocy względem algorytmu bazowego tj. niewykorzystującego heurystyk i ograniczenia długości ścieżki. Przedstawiono na nim pobór mocy w rozbiciu na składniki związane z aktywacją urządzeń i transmisją danych. Pozwala on zauważyć znaczną, sięgająca niemal 20% redukcję poboru mocy związanej z transmisją danych dla topologii Butterfly. Dla topologii Fat tree oszczędność jest nieco mniejsza i osiągana dla krótszych ścieżek, przy czym w badanym przykładzie nie było możliwe wyłączenie dodatkowych urządzeń. Wynika to z mniejszej liczby redundantnych węzłów i ścieżek w tej topologii.



Rysunek 13: Pobór mocy względem rozwiązania bez ograniczenia długości ścieżki dla dwóch przykładowych topologii.

4.5.4 Stanowisko testowe i demonstrator systemu energooszczędnego sterowania siecią

Efektem końcowym prac dotyczących energooszczędnych sieci była eksperymentalna weryfikacja opracowanych rozwiązań w środowisku laboratoryjnym. W tym celu zbudowałem instalację laboratoryjną opisaną w pracach [H5, U3, U7], która posłużyła do:

- prowadzenia pomiarów dla pojedynczych urządzeń, m.in. identyfikacji charakterystyk poboru mocy rutera programowego,

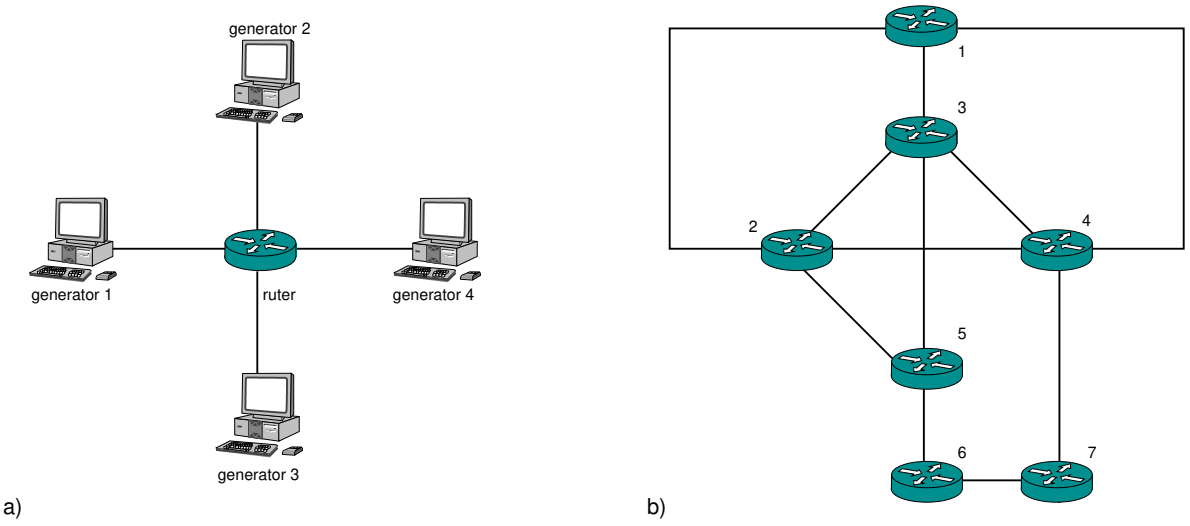
[8]NetBench–Packet Simulator for Data Center Network Topologies, Routing, and Congestion Control. https://github.com/ndal-eth/netbench

[9]ang. Equal Cost Multi-Path – trasowanie wielościeżkowe polegające na równoważeniu obciążenia między trasami o tej samej metryce, dostępne m.in. w OSPF (Open Shortest Path First).

- demonstracji i weryfikacji efektywności systemu przydziału zasobów energooszczędnej sieci.

Wymagania na instalację laboratoryjną sprowadzają się do:

- przygotowania zestawu urządzeń – w tym wypadku komputerów klasy PC,
- przygotowania sieci łączącej urządzenia umożliwiającej zestawianie różnorodnych topologii,
- zapewnienia systemu pomiarowego pozwalającego na częsty pomiar pobieranej przez urządzenia mocy,
- dostarczenia zestawu generatorów ruchu i programów symulujących obciążenie maszyn.



Rysunek 14: Przykładowe topologie stanowiska testowego. Pomiar charakterystyki rutera obciążonego wywołaniami pochodzącymi z wielu generatorów (a), uproszczona topologia sieci WARMAN używana przez demonstrator systemu (b).

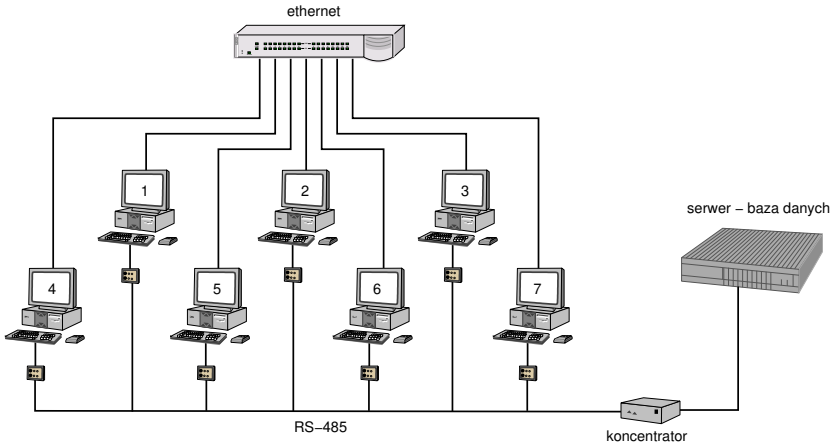
Środowisko badawcze było wykorzystane w ramach projektu ECONET. Wybór maszyn klasy PC wynikał z podstawowego celu projektu, jakim była budowa rutera programowego. Bazą takiego urządzenia jest w istocie komputer klasy PC o wydajności odpowiedniej do obsługi ruchu transmitowanego przez interfejsy. W przypadku pomiarów charakterystyk pojedynczego urządzenia sieć potrzebna jest w scenariuszach, gdzie obciążeniem jest ruch sieciowy generowany przez zewnętrzne maszyny. Scenariusze te charakteryzują się stosunkowo prostą topologią sieci (np. topologią gwiazdy, przedstawioną na rysunku 14a). Dla demonstracji działania systemu przydziału zasobów w sieci konieczne jest zestawienie topologii przypominających, w miarę możliwości, rzeczywiste sieci. W części eksperymentów wykorzystywałem sieć będącą uproszczeniem sieci metropolitalnej WARMAN, przedstawionej na rysunku 14b. Realizacja obu celów jest możliwa przez zapewnienie szybkiej rekonfiguracji. Rozwiązaniem pozwalającym to osiągnąć, przy stosunkowo niskich nakładach, jest wyposażenie maszyn w odpowiedni zestaw interfejsów sieciowych, oraz połączenie ich z wykorzystaniem przełącznika sieciowego wspierającego technologię

sieci wirtualnych VLAN, IEEE 802.1q. W omawianym przypadku każdy z komputerów był wyposażony w czteroportowe karty 1 Gb/s oraz karty 100 Mb/s. Dzięki temu można było w dowolny sposób, kreować połączenia między maszynami, a także w razie potrzeby, zwielokrotniać przepustowość na wybranych relacjach przez zastosowanie agregacji łączy. Co więcej, karty o przepustowości 1 Gb/s można przełączać na mniejsze prędkości transmisji, np. 100 Mb/s i 10 Mb/s, a przez to także wpływać na ich pobór mocy.

Tak zaprojektowane stanowisko składało się z następujących elementów:

- 7 komputerów PC wyposażonych w procesor i7-3770, karty Ethernet 4x1 Gb/s, jedną lub dwie karty Ethernet 100 Mb/s,
- liczników energii wyposażonych w interfejs RS-485,
- koncentratora RS-485 z interfejsem Ethernet,
- serwera bazy danych z zainstalowanym oprogramowaniem służącym do komunikacji z koncentratorem.

Wykorzystanie typowych liczników energetycznych wynikało z chęci obniżenia nakładów przy zachowaniu dobrej dokładności. Wadą rozwiązania jest ograniczona częstotliwość, z jaką można zbierać pomiary, co wynika z konstrukcji liczników, jak i użytego protokołu RS-485. W wyniku tego w większości eksperymentów pobór mocy próbkowano co 15 s. Dzięki wykorzystaniu bazy danych (mySQL) dane zbierane w czasie eksperymentów były utrwalane i mogły być następnie przeszukiwane z wykorzystaniem przygotowanych skryptów. Schemat stanowiska testowego przedstawia rys. 15. Ze względu na to,



Rysunek 15: Schemat stanowiska testowego.

że w chwili, gdy były prowadzone eksperymenty, moduł jądra Linuxa realizujący protokół MPLS był poważnie niedopracowany[10] zaproponowałem rozwiązanie wykorzystujące możliwości filtra pakietów (iptables) oraz rutingu źródłowego (PBR – Policy Based Routing) udostępniane przez standardowy pakiet iproute2. Opis zawiera praca [H7]. Takie podejście wpisuje się w koncepcję systemu sterowania siecią przez zapewnienie zestawu skryptów i tekstowej bazy danych pozwalających na sterowanie zawartością tablic tras

[10]Dostępne były wyłącznie nieoficjalne pakiety wymagające samodzielnej kompilacji i współdziałające z przestarzałymi wersjami jądra, a przez to niestabilne.

i kierujących do nich ruch polityk. Filtr pakietów wykorzystywany jest do nadpisywania adresów przez mechanizm NAT. Stanowi to w pewnym sensie ekwiwalent przepisywania etykiet przez protokół MPLS.

4.5.5 Eksperymenty laboratoryjne weryfikujące jakość polityk przydziału zasobów w sieci

Opisaną instalację laboratoryjną wykorzystałem do szeregu eksperymentów weryfikujących opracowane struktury i algorytmy sterowania. Opracowałem odpowiednie scenariusze, przygotowałem skrypty służące do zestawiania tras i zbierania danych, przeprowadziłem eksperymenty i przeanalizowałem wyniki.

Uzyskane wyniki eksperymentów pozwoliły określić nie tylko osiągniętą dzięki odpowiedniej alokacji zasobów redukcję poboru mocy, lecz także pogorszenie jakości obsługi, wynikające m.in. z narzutów wprowadzanych przez mechanizmy rutingu. Tabele 2 i 3 z pracy [H5] pokazują przykładowe wyniki porównania dwóch algorytmów, tj. algorytmu z pełnym wyznaczaniem tras oraz algorytmu heurystycznego zakładającego relaksację zadania (39)-(44).

Ruch			Moc	
oferowany [Mb/s]	zmierzony [Mb/s]	rozbieżność	pobór [W]	redukcja [%]
850	852	+2,4%	239,7	12,8
1000	1004,7	+0,5%	240,9	12,1
1150	1130,7	-1,7%	242,1	11,6
1300	1274,7	-1,9%	243,5	11,1
1450	1400,7	-3,4%	245,5	10,4

Tabela 2: Wyniki eksperymentów dla rozwiązania zadania dokładnego (binarnego), redukcja mocy podana względem rutingu wykorzystującego algorytm najkrótszej ścieżki.

Ruch			Moc	
oferowany [Mb/s]	zmierzony [Mb/s]	rozbieżność	pobór [W]	redukcja [%]
850	852	+2,4%	235,9	13,9
1000	999,4	-0,06%	239,5	12,7
1150	1111,4	-3,4%	239,3	12,6
1300	1239,4	-4,7%	241	12
1450	1351,4	-6,8%	241,6	11,8

Tabela 3: Wyniki eksperymentów dla algorytmu heurystycznego, redukcja mocy podana względem rutingu wykorzystującego algorytm najkrótszej ścieżki.

W pracach [H8, U8] pokazuję weryfikację wyników rozwiązania zadania, które w przeciwieństwie do przedstawianych w poprzednich punktach sformułowań, dążyło do optymalizacji funkcji celu będącej sumą ważoną składnika związanego z zużyciem energii i użyteczności przesyłanych przepływów. Przy takim wskaźniku jakości możliwa jest, w zależności od przyjętych wag składników, wymiana pomiędzy jakością transmisji, tj. osiągnięta przepływnością a kosztem zużytej energii. Z tego też powodu wykonanie pomiarów w środowisku laboratoryjnym, pozwoliło ocenić, w jakim stopniu uproszczone modele wpływają na

skuteczność wyznaczonych sterowań. Podobne eksperymenty, ale dla kompletnego, hierarchicznego systemu sterowania siecią zawiera praca [H7]. Oprócz polityki sterowania siecią wykorzystano w nich algorytmy lokalne, a dokładniej specjalizowane wersje algorytmu sterowania częstotliwością taktowania procesora (power governor). Należy podkreślić, że z racji zastosowania zmiennego obciążenia generowanym ruchem algorytm sterujący siecią zmieniał w trakcie działania systemu trasy, po których wykonywana była transmisja. Zebranie danych pomiarowych w tym przypadku pozwoliło przede wszystkim oszacować poziom strat pakietów wynikający z przełączania tras.

4.6 Model poboru mocy przez urządzenie sieciowe

W projektowaniu energooszczędnych sieci kluczowe jest stosowanie dokładnych modeli poboru mocy przez urządzenia sieciowe. Modele powinny dobrze oddawać rzeczywistość. Wykorzystanie uproszczonych modeli może skutkować podjęciem błędnych decyzji. Prowadząc badania zwracałem uwagę na dokładny opis procesów zachodzących w urządzeniach sieciowych. Parametry opracowanych przeze mnie modeli były identyfikowane na rzeczywistym sprzęcie. Należy podkreślić, że współcześnie stosowane urządzenia sieciowe odznaczają się złożoną strukturą. Wynika to z jednej strony z modułowej budowy tych urządzeń, z drugiej zaś z konieczności zwielokrotnienia przynajmniej niektórych ich elementów, tak w celu zapewnienia niezawodności, jak też dla zwiększenia wydajności. Typowym przykładem może być przełącznik wyposażony w szereg kart liniowych z wieloma interfejsami sieciowymi[11], a także zwielokrotnioną macierz przełączającą czy zasilacze sieciowe. Co istotne, dzięki zastosowaniu agregacji łączy, możliwe jest zwiększenie przepustowości na wybranych relacjach przez wykorzystanie pewnej liczby (np. 4, 8) interfejsów w obu łączonych urządzeniach. Zgodnie z obserwacją przedstawioną w pracy [21] wykorzystanie zwielokrotnionych łączy daje możliwość zmniejszenia poboru mocy w okresach niepełnego ich wykorzystania przez wprowadzenie części z nich w stan czuwania, czy wręcz wyłączenia. Uogólniając, można zaproponować podobną metodę w przypadku innych komponentów węzła sieciowego, takich jak karty liniowe czy np. wentylatory. Oznacza to, że oszczędności można uzyskać nie tylko dzięki zmianie wydajności konkretnego komponentu (o ile pozwala na to jego konstrukcja), ale również dzięki wyłączaniu części zwielokrotnionych komponentów. Z tego powodu, koncepcja modelu energetycznego powinna z jednej strony oddawać hierarchiczną budowę urządzenia sieciowego, z drugiej zaś nie istnieją powody, które wymagałyby odróżniania stanów energetycznych łączy wynikających z zastosowania wewnętrznych mechanizmów energooszczędnych (np. okresowego usypiania) i wyłączania części zwielokrotnionych czy redundantnych komponentów. W pracach [H4, U9, U10] przedstawiłem koncepcję takiej dekompozycji urządzenia sieciowego, w której możliwe jest sterowanie stanami energetycznymi na trzech poziomach:

- najniższym, tworzonym przez interfejsy sieciowe,
- pośrednim, w skład którego wchodzą karty liniowe z interfejsami,
- najwyższym, odpowiadającym całemu urządzeniu.

Uporządkowanie poziomów odpowiada zależnościom między tworzącymi je komponentami. Wyłączenie karty liniowej wymaga wyłączenia wszystkich znajdujących się na niej

[11] Por. np. przełącznik Juniper QFX10016 *https://www.juniper.net/us/en/products/switches/qfx-series/qfx10000-qfx10008-qfx10016-spine-and-core-switches.html*.

interfejsów. Podobnie, wprowadzenie całego rutera w stan czuwania pociąga za sobą wygaszenie wszystkich kart liniowych i portów. Pobór mocy tak zdefiniowanego urządzenia można opisać następującym wzorem

$$P(y_{ek}, x_c, z) = \sum_{e=1}^{E} \sum_{k=1}^{K} \xi_{ek} y_{ek} + \sum_{c=1}^{C} W_c x_c + Tz + P_0, \tag{45}$$

gdzie ξ_{ek} oznacza pobór mocy interfejsu e w stanie energetycznym k, W_c jest poborem mocy karty liniowej c w stanie włączenia, T odpowiada poborowi mocy przez pozostałe komponenty rutera w stanie włączenia, zaś P_0 jest składnikiem stałym, pobieranym niezależnie od stanu energetycznego. Dla uproszczenia modelu, a przede wszystkim dla uproszczenia rozwiązywanego zadania optymalizacji, przyjąłem, że zarówno ruter jak i karta liniowa mogą znajdować się tylko w jednym z dwóch stanów energetycznych. Pierwszy z nich odpowiada wyłączeniu, czy raczej uśpieniu. Drugi to tryb normalnego działania, przy czym na odpowiadający temu stanowi pobór mocy składają się te części karty liniowej czy rutera[12], które są wspólne dla komponentów modelowanych na niższym poziomie hierarchii. Zmienne binarne y_{ek}, x_c i z opisują stan energetyczny interfejsu e, karty liniowej c i rutera. Zmienna y_{ek} jest w istocie wektorem K zmiennych binarnych, z których tylko jedna, odpowiadająca aktualnemu stanowi energetycznemu, może być równa 1. Na wyjaśnienie zasługuje też kwestia poboru mocy w stanie wyłączenia (uśpienia) rutera P_0. Wartość ta odpowiada poborowi mocy przez wszystkie komponenty niepracującego rutera i stanowi czynnik stały. Z tego powodu w zapisie zadań optymalizacji może być pomijana.

Przedstawiony model był wykorzystywany w prezentowanych wcześniej algorytmach energooszczędnego sterowania siecią opisanych w pracach [H4, H5, U3, U4, U6]. Przyjęte uproszczenia, w tym ograniczenie stanów energetycznych rutera i kart linowych do włączenia i uśpienia, wynikają z konieczności zmniejszenia wymiaru zmiennych decyzyjnych w zadaniach optymalizacji. W innych przypadkach, szczególnie gdy rozważane jest lokalne sterowanie urządzenia, może być celowe zwiększenie dokładności modelu poprzez przede wszystkim uwzględnienie stanów energetycznych komponentów związanych z przekazywaniem ruchu między interfejsami tj. macierzy przełączającej czy procesora w przypadku rutera programowego.

4.7 Modelowanie poboru mocy przez komputerowy system przetwarzania danych

Istotnym elementem omawianych wcześniej struktur i algorytmów sterowania jest model pozwalający wyznaczyć pobór mocy przez procesor, urządzenie sieciowe, lub cały system komputerowy na podstawie dostępnych pomiarów. Należy zauważyć, że ostatecznym celem sterowania powinno być ograniczenie poboru mocy przez cały system komputerowy, nie zaś sam procesor, czy urządzenia peryferyjne jak pamięć, czy karty sieciowe. Jednakże, z drugiej strony, wykonywanie wiarygodnych i częstych pomiarów całkowitego poboru mocy jest utrudnione, gdyż wymaga instalacji odpowiednich układów w części zasilającej. Obecnie odczyt takich wartości zapewnia przede wszystkim standard IPMI[13] dostępny

[12] Będą to zasilacze, wentylatory, ale także procesory i macierz przełączająca.

[13] IPMI (Intelligent Platform Management Interface) jest interfejsem pozwalającym na monitoring i zarządzanie serwerem za pośrednictwem wbudowanego w niego sterownika, tzw. BMC (Baseboard management controller).

w sprzęcie klasy serwerów, ale częstotliwość odczytów jest ograniczona. Podobnie śledzenie stanu baterii w urządzeniach przenośnych nie może być traktowane jako rozwiązanie zapewniające dokładność i rozdzielczość czasową potrzebną dla konstrukcji szybkich algorytmów lokalnych. Z tego powodu wskazane jest skonstruowanie modeli poboru mocy wykorzystujących jako zmienne wejściowe stosunkowo łatwo dostępne wartości takie jak:

1. obciążenie procesora,
2. estymaty poboru mocy przez procesor dostępne przez rejestry RAPL[14],
3. ruch obserwowany na interfejsach sieciowych.

Przez łatwą dostępność rozumie się możliwość stosunkowo częstego odczytu (z okresem rzędu 100 ms) bez potrzeby instalacji dodatkowego oprzyrządowania. Przyjęte próbkowanie jest w tym przypadku zgodne z okresem repetycji algorytmów sterujących częstotliwością pracy procesora w systemie Linux (*frequency governor*) [34].

4.7.1 Budowa oprzyrządowania pomiarowego do identyfikacji modeli poboru mocy

Skonstruowanie, identyfikacja i weryfikacja dokładnych modeli wymaga przeprowadzenia szeregu eksperymentów pomiarowych. Większość dostępnych opracowań związanych z tą tematyką skupia się na bardzo dokładnym określeniu poboru mocy przez poszczególne elementy systemu komputerowego. Należy tu wskazać przede wszystkim prace [24, 23] weryfikujące dokładność odczytów mocy uzyskiwanych poprzez ogólnie dostępne interfejsy jak IPMI i RAPL. Przedstawione tam wyniki są o tyle istotne, że jak już zostało wspomniane, wartości dostępne za pomocą RAPL są estymatami poboru mocy przez poszczególne komponenty, stąd konieczność ich weryfikacji.

Identyfikacja modelu całkowitego poboru mocy przez urządzenie wymaga innego podejścia. Przede wszystkim potrzebne jest stanowisko pomiarowe wyposażone w mierniki pozwalające na pomiar poboru na liniach zasilających sprzęt, oprogramowanie umożliwiające określenie obciążenia procesora i ew. innych elementów systemu oraz odpowiednie generatory obciążenia. Przez generatory obciążenia należy rozumieć uruchamiane na będącym obiektem pomiarów komputerze aplikacje symulujące obliczenia, odczyt i przetwarzanie danych oraz inne czynności angażujące procesor, pamięć i pozostałe komponenty. Drugą istotną klasą generatorów są urządzenia służące do generowania ruchu sieciowego o zadanej charakterystyce i natężeniu. Ruch ten jest odbierany, przetwarzany lub też transmitowany dalej przez opomiarowany komputer. Kwestia generowania ruchu sieciowego była dla mnie o tyle istotna, że jednym z celów, dla których przygotowywane były modele poboru mocy, było wykorzystanie ich w algorytmach sterujących energooszczędnego rutera programowego zbudowanego na bazie komputera klasy PC. Na potrzeby tych prac zostało wykorzystane stanowisko pomiarowe opisane w rozdziale 4.5.4.

4.7.2 Model poboru mocy wykorzystujący dane z rejestrów procesora

Obciążenie procesora można obserwować analizując statystyki systemowe, w systemie Linux dostępne poprzez interfejs `proc`, lub z większą dokładnością przez bezpośredni odczyt

[14]RAPL (Running Average Power Limit) zestaw specjalizowanych rejestrów procesorów Intel służących m.in. do nadzorowania i zarządzania poborem mocy.

zawartosci rejestrów RAPL procesora. Należy przy tym zwrócić uwagę, że rejestry te nie zawierają pomiarów zużycia energii, lecz estymaty wynikające z zliczania aktywnych, tj. takich, w których wykonywane są instrukcje programu i bezczynnych cykli procesora. Długość cyklu zależy od ustawionej częstotliwości taktowania, zaś liczba wykonanych cykli może ulec zmniejszeniu w przypadku wprowadzenia procesora w stan uśpienia. Modelowanie rzeczywistego obciążenia procesora jest w związku z tym zadaniem złożonym, a szczegóły zidentyfikowanego przez producenta procesora modelu nie są do końca znane [24, 39]. Mimo to rejestry RAPL mogą być z powodzeniem wykorzystane do sterowania poborem mocy przez procesor. Sprzyja temu, wspomniana wcześniej możliwość programowego odczytu z częstotliwością dostateczną dla algorytmów pracujących w jądrze systemu operacyjnego, co sprawdziłem eksperymentalnie w trakcie prac opisanych m.in. w artykule [H9]. W pracach [H10, U11] opisałem szereg eksperymentów, których wynikiem było zidentyfikowanie modelu uzależniającego pobór mocy przez komputer od poboru mocy odczytywanego z rejestrów RAPL. Istotną obserwacją był fakt, że charakterystyki poboru mocy w istotny sposób zależą od charakteru obciążenia, któremu poddany jest komputer. W związku z tym badałem scenariusze, w których komputer pełnił kolejno rolę:

1. serwera obliczeniowego,
2. rutera programowego,
3. serwera transkodującego strumień wideo.

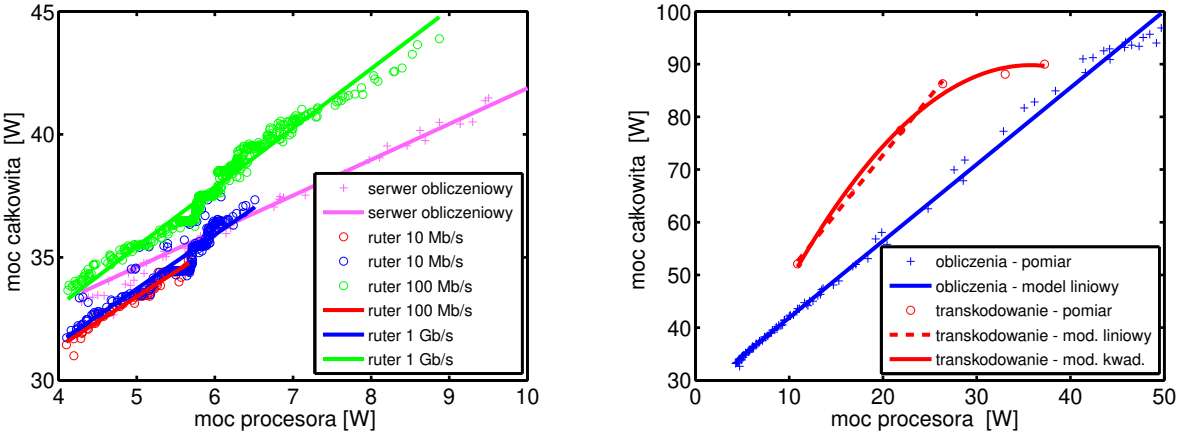
W pierwszym scenariuszu obciążeniem był program wykonujący obliczenia arytmetyczne. W drugim ruter programowy przesyłał ruch między swoimi interfejsami. Ostatni przypadek był w pewnym sensie połączeniem obu poprzednich, gdyż zadaniem serwera była zmiana formatu strumienia wideo odbieranego na jednym z interfejsów sieciowych i wysłanie go do odbiorcy przez drugi z interfejsów. W ogólnej postaci proponowany model można przedstawić jako wielomian drugiego stopnia:

$$p(w) = \alpha_0 + \alpha_1 w + \alpha_2 w^2, \tag{46}$$

gdzie w to pobór mocy przez procesor odczytany z rejestrów RAPL, $p(w)$ całkowity pobór mocy przez komputer, zaś α_0, α_1 i α_2 są identyfikowanymi eksperymentalnie współczynnikami modelu. Należy przy tym dodać, że w przypadku pierwszych dwu scenariuszy wystarczająco dokładny jest model liniowy, tzn. można przyjąć $\alpha_2 = 0$. Jedynie w przypadku serwera transkodującego strumień wideo konieczne jest uwzględnienie części kwadratowej, przy czym wartość współczynnika α_2 jest stosunkowo niewielka. Wystąpienie nieliniowości w ostatnim przypadku można interpretować jako dowód złożoności zadania przekodowywania strumienia wideo, wymagającego pełniejszego niż w pozostałych przypadkach zaangażowania układów komputera, w tym pamięci i jednostki zmiennoprzecinkowej. Przykłady danych pomiarowych i dopasowanych modeli ilustruje rysunek 16. Modele zostały wykorzystane w konstrukcji sterowników częstotliwości procesora opisanych w pracy [H11].

4.7.3 Model poboru mocy wykorzystujący statystyki ruchu sieciowego

Natężenie ruchu sieciowego można odczytać za pomocą funkcji systemu operacyjnego na podobnych zasadach jak w przypadku statystyk procesora, m.in. przez interfejs `proc`



Rysunek 16: Identyfikacja modeli dla rutera programowego, serwera obliczeniowego (z lewej) oraz serwera transkodującego wideo (z prawej); punkty i linie odpowiadają pomiarom i dopasowanym modelom.

czy wewnętrzne mechanizmy jądra. Pozyskane w ten sposób informacje są o tyle istotne, że dostarczają wiedzy o obciążeniu niejako w momencie jego powstania, tj. w chwili, gdy żądanie obsługi zostaje odebrane przez serwer. Uwzględnienie takich danych pozwala na szybszą reakcję na zmianę obciążenia, pod warunkiem że znany jest model wiążący obserwowany ruch sieciowy z poborem mocy przez procesor i całe urządzenie.

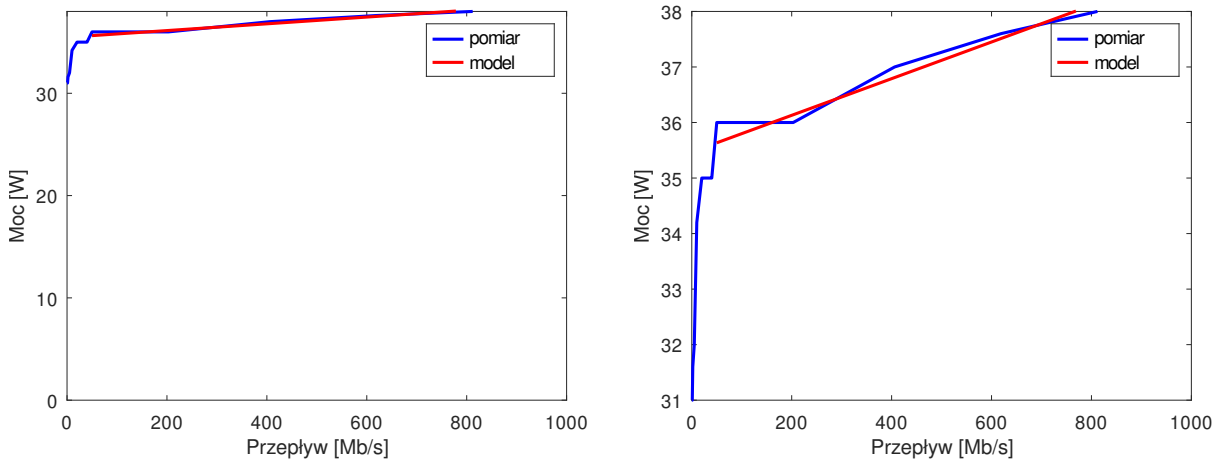
Po przeprowadzeniu pomiarów, których wyniki prezentuje [H8, U7] został zdefiniowany model poboru mocy przez ruter programowy

$$P_{total} = P_0 + P_{var}(x). \tag{47}$$

Podstawowym założeniem jest możliwość przedstawienia całkowitej mocy pobieranej przez ruter programowy P_{total} jako sumy mocy pobieranej niezależnie od obciążenia P_0 oraz składnika zależnego od całkowitego ruchu x przesyłanego przez wszystkie interfejsy rutera, tj. $P_{var}(x)$. Część stałą P_0 należy rozumieć jako pobór mocy w stanie bezczynności, czyli w sytuacji, gdy układy, w tym procesor, pamięć i karty sieciowe, działają w trybach obniżających pobór mocy do minimum przez zmniejszenie częstotliwości taktowania, lub są wyłączone. Należy zwrócić uwagę, że całkowite wyłączenie, czyli stan, w którym urządzenie nie pobiera energii, nie jest zazwyczaj możliwe, gdyż uniemożliwiało by to ponowne włączenie. Z tego powodu, nawet przy braku zadań, część obwodów musi pozostać w trybie czuwania, aby możliwa była aktywacja[15]. Podczas przekazywania ruchu między interfejsami wszystkie podzespoły komputera wykorzystywanego jako ruter programowy obciążane są w stopniu zależnym od natężenia ruchu x. W związku z tym mogą być wprowadzane w stany energetyczne charakteryzujące się wyższą wydajnością, ale też większym poborem mocy. Zmiany te odzwierciedla funkcja $P_{var}(x)$. Model (47) jest zgodny z typowym podejściem prezentowanym w wielu pracach, m.in. [12, 14, 38, 16]. Moim autorskim rozwiązaniem jest sposób identyfikacji i modelowania części zmiennej $P_{var}(x)$, który jest powiązany bezpośrednio z wykorzystaniem modelu do wyznaczania optymalnych stanów

[15]W przypadku rutera programowego rozważane było przede wszystkim wykorzystanie technologii Wake-on-Lan pozwalającej uruchomić maszynę poprzez wysłanie komunikatu na odpowiednio skonfigurowany interfejs sieciowy.

energetycznych rutera programowego, w tym przede wszystkim sterowania poborem mocy przez karty sieciowe. W prowadzonych eksperymentach wykorzystałem karty sieciowe Ethernet o przepustowości 1 Gb/s, które mogą być przełączane w tryby pracy z mniejszą przepustowością, tj. 100 Mb/s oraz 10 Mb/s. Badałem typową konfigurację taniego rutera programowalnego bazującego na komputerze PC ogólnego przeznaczenia, który może być stosowany w niewielkich sieciach. Należy jednak zwrócić uwagę, że rozwiązania sprzętowe stosowane w wysokowydajnych ruterach nie odbiegają znacząco od badanej architektury. W istocie większość budowanych w ostatnich czasach ruterów ma konstrukcję modułową, w której można wyróżnić karty interfejsów sieciowych, macierz przełączającą i procesory nadzorujące ich pracę [41, 16]. Taka architektura oferuje różnorodne możliwości sterowania wydajnością a przez to poborem mocy komponentów. Podstawą jest, dobór częstotliwości taktowania i przepustowości interfejsów. Oprócz tego możliwe jest wyłączanie zwielokrotnionych komponentów, np. interfejsów obsługujących zagregowane łącza, zduplikowanych dla zwiększenia niezawodności i wydajności procesorów, bądź też elementów hierarchicznej macierzy przełączającej.



Rysunek 17: Krzywa poboru mocy i zidentyfikowany model w liniowej części przebiegu dla rutera programowego z interfejsami w trybie 1Gb/s. Dla większej czytelności na prawym wykresie zawężono zakres osi rzędnych.

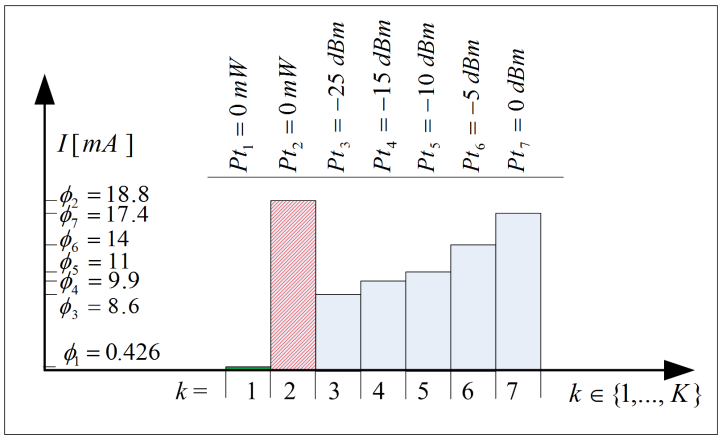
Pomiary wykonywałem przy obciążeniu ruchem sieciowym o zmiennym natężeniu, przy czym karty sieciowe pracowały w jednym z wymienionych trybów. Analiza wyników pomiarów wykazała, że w przybliżeniu kształt charakterystyki poboru mocy można określić jako wklęsły, przy czym największa nieliniowość występuje w początkowej części. Pokazuje to rysunek 17. Co więcej, tryby o mniejszej przepustowości pozwalają, dla niektórych poziomów natężenia obsługiwanego ruchu, zmniejszyć nieco pobór mocy. Na przykład dla przepustowości 1 Mb/s użycie trybu 10 Mb/s pozwala obniżyć całkowity pobór mocy z 31 W na 30,4 W. Uwzględnienie przełączania trybów umożliwia modelowanie z zadowalającą dokładnością poboru mocy przy użyciu funkcji liniowej

$$P_{var}(x) = \begin{cases} \alpha + \beta x & \text{jeśli } x \geqslant x_0, \\ P_0 & \text{w przeciwnym przypadku.} \end{cases} \tag{48}$$

Taka postać funkcji poboru mocy upraszcza konstrukcję mechanizmów sterujących, w szczególności pozwala korzystać z odznaczających się wysoką wydajnością solwerów liniowych.

4.8 Energooszczędne sieci bezprzewodowych czujników

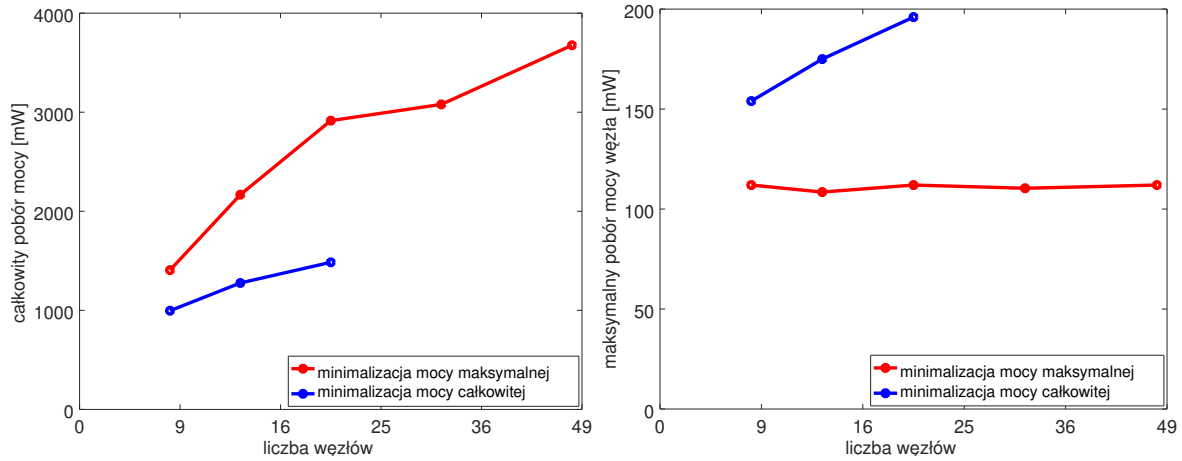
Wśród zastosowań złożonych systemów obliczeniowych można znaleźć ostatnio coraz więcej takich, gdzie dane przetwarzane przez system zbierane są za pomocą osobnej sieci niewielkich urządzeń, bezprzewodowych czujników, rozlokowanych w terenie, a przez to wykorzystujących własne źródła zasilania. Sytuacja taka wynika z jednej strony z coraz większego zapotrzebowania na rozproszone systemy monitorujące stan środowiska, instalacji przemysłowych czy komunalnych, z drugiej zaś strony, z łatwej dostępności i stosunkowo niskich kosztów czujników. Typowe scenariusze to zbieranie danych o skażeniach w przypadku katastrofy kolejowej i rozszczelnienia cystern z chemikaliami, monitorowanie migracji zwierząt w obszarze chronionym, czy też nadzorowanie ruchu drogowego i zanieczyszczeń powietrza jako element wejściowy systemu zarządzania tzw. inteligentnym miastem. Sieci czujników w istotny sposób różnią się od standardowych sieci komputerowych. Przede wszystkim są to sieci bezprzewodowe zbudowane z urządzeń o niewielkim zasięgu, a przez to zmuszonych zazwyczaj przekazywać dane przez wiele węzłów pośrednich (ang. multi-hop). Metoda wieloskokowej transmisji pozwala zapewnić funkcjonowanie sieci na rozległym obszarze, a przy tym ograniczyć zużycie energii dzięki redukcji mocy nadawania. Urządzenia mogą zazwyczaj nadawać z kilkoma poziomami mocy, i co istotne zazwyczaj pracują w trybie *simpleks*, czyli nie mogą odbierać transmisji, gdy same nadają. Co więcej, pobór mocy w trybie odbioru jest typowo zbliżony, czy wręcz wyższy niż podczas nadawania. Pobór mocy przykładowego urządzenia dla różnych trybów przedstawia rysunek 18. Biorąc pod uwagę niewielkie wymiary, a więc i pojemność baterii zasilającej



Rysunek 18: Pobór prądu przez układ CC2420 [2] w różnych trybach: k=1 - uśpienia, k=2 - odbioru, k=3-7 nadawania.

czujnik, konieczne jest bardzo ostrożne korzystanie z możliwości jego układów, w tym toru radiowego. Typowym podejściem jest wykorzystanie algorytmów sterowania aktywnością sprowadzających się, w pewnym uproszczeniu, do okresowego budzenia uśpionych czujników, tak aby mogły one wykonać pomiary i przesłać zebrane dane. W zależności od zastosowania czujniki mogą być aktywowane asynchronicznie, w tym w odpowiedzi na zewnętrzne zdarzenie lub synchronicznie zgodnie z określonym harmonogramem. Wybrane problemy i zagadnienia związane z projektowaniem sieci bezprzewodowych czujników prezentuje współautorska monografia [H12]. Ważną jej częścią jest przedstawienie algorytmów służących do gospodarowania zasobami energetycznymi węzłów sieci będące głównie moim wkładem. Zaprezentowałem zróżnicowane podejścia, włącznie z ich krytyczną ana-

lizą, oceną i rekomendacją zastosowań. Przedstawione w monografii wyniki prowadzonych przez zespół autorów badań eksperymentalnych, w tym z użyciem rzeczywistego sprzętu, pokazują własności różnych podejść. Należy podkreślić, że klasyfikacja algorytmów wykorzystywanych w sieciach czujników jest dość złożona, szczególnie gdy celem jest oszczędność energii i wydłużenie czasu życia sieci. W istocie, realizacji tego celu mogą służyć również odpowiednio skonstruowane algorytmy trasowania, które opisałem w rozdziale 11 monografii, jak również algorytmy sterowaniem dostępem do medium transmisyjnego (rozdział 14), czy też techniki grupowania węzłów (rozdział 10). Integracja algorytmów działających na poziomie medium transmisyjnego oraz wysokopoziomowego zarządzania strukturą sieci, jak grupowanie węzłów i sterowanie aktywnością, jest z jednej strony konsekwencją konieczności postrzegania sieci jako całości z drugiej zaś względnej prostoty rozwiązań wynikającej z ograniczonych zasobów. Sieć czujników jest systemem realizującym wspólny, dobrze zdefiniowany cel, co sprzyja silnemu powiązaniu algorytmów. Wymaganie prostoty implementacji skutkuje zaś często powiązaniem funkcji warstw oprogramowania, czy wręcz ich redukcji.



Rysunek 19: Średnie wartości całkowitego poboru mocy przez sieć (wykres lewy) i maksymalnego poboru mocy przez czujnik (wykres prawy) uzyskane w wyniku optymalizacji aktywności węzłów.

Wynikiem prowadzonych przeze mnie prac badawczych w zakresie projektowania energooszczędnych sieci bezprzewodowych czujników jest rozwiązanie opisane w pracy [H13]. Skoncentrowałem uwagę na technikach sterowania aktywnością węzłów, uznając je za najbardziej obiecujące. Opisane w [H13] rozwiązanie należy do grupy technik określanych jako PSN (ang. Periodic Sensor Network), i zakłada, że podzielone na J grup czujniki, są uruchamianie okresowo, tak aby mogły wysłać zebrane dane wykorzystując trasy wyznaczone w ramach swojej grupy. Do każdej z grup przypisana jest podzielona na szczeliny ramka. Czujniki należące do grupy transmitują dane w przyporządkowanych im szczelinach. Takie podejście pozwala ograniczyć interferencję między czujnikami, a także rozwiązywać zadanie alokacji szczelin czasowych dla każdej ramki oddzielnie, co skutkuje redukcją jego wymiarowości i czasu obliczeń. Wykorzystany model komunikacji bierze pod uwagę ubieganie się czujników o dostęp do medium transmisyjnego, tj. uwzględnia problemy ukrytego i eksponowanego węzła [H12] przez analizę zasięgu efektywnej transmisji i interferencji w różnych trybach nadawania. Model energetyczny odwzorowuje pobór mocy

w kolejnych stanach aktywności: uśpienia, odbioru i nadawania ze zmieniającą się mocą sygnału. Celem sterowania jest wyznaczenie optymalnego harmonogramu nadawania w ramach przypisanej grupie ramki, przy zapewnieniu przekazania danych z wszystkich czujników do węzła bazowego. Zdefiniowałem dwa warianty wskaźnika jakości. W pierwszym z nich minimalizowany jest pobór mocy przez wszystkie czujniki. Drugie sformułowanie ma postać mini-maksową, tj. minimalizowany jest pobór mocy przez najmniej efektywny czujnik. Pierwszy wariant wydaje się prostszy, jest on jednak właściwy dla sieci dużych, gdzie kolejne czujniki mogą przejmować funkcje tych, które w znaczącym stopniu wyczerpały swoje źródła zasilania transmitując dane. W przypadku, gdy nie ma takiej możliwości, sieć może przedwcześnie utracić spójność[16] w wyniku wyłączenia kluczowego węzła, np. węzła tworzącego połączenie między dwoma jej częściami. Drugie podejście ogranicza możliwość przeciążenia takich węzłów, a co się z tym wiąże, pomaga wydłużyć czas pracy sieci. Skuteczność obu wariantów algorytmu została zweryfikowana przez eksperymenty numeryczne. Wykazały one, zgodnie z przypuszczeniem, że jakkolwiek minimalizacja całkowitego zużycia energii przez sieć przyczynia się do ograniczenia sumy pobieranej przez czujniki energii, to nie zapobiega wyczerpywaniu baterii niektórych urządzeń. Widać to na rysunku 19. Podejście mini-maksowe skutkuje utrzymywaniem pobieranej przez czujniki mocy na podobnym poziomie mimo zmian rozmiaru sieci, co zapewnia dłuższe jej działanie. Oczywistym skutkiem jest w tym wypadku zwiększenie całkowitego poboru mocy wynikające z wydłużenia tras. Co ciekawe, tak zdefiniowane zadanie, mimo praktycznie identycznego rozmiaru, można rozwiązać w znacznie krótszym czasie. Pozwala to uzyskać dokładne rozwiązanie z użyciem typowych solwerów realizujących metodę podziału i oszacowań[17] dla sieci średnich rozmiarów, tj. zwierających około kilkudziesięciu węzłów.

5 Informacje o pozostałej istotnej aktywności naukowej

Zarówno w ramach realizowanych przeze mnie projektów rozwojowych jak i w badaniach naukowych zajmowałem się szeregiem zagadnień dotyczących ogólnie rozumianych sieci. Najważniejsze wątki badawcze, poza podstawowym opisanym w poprzednim rozdziale, obejmowały:

- analizę sieci społecznych i złożonych,
- bezpieczeństwo sieciowe,
- inżynierię ruchu w sieciach telekomunikacyjnych.

Lista wybranych publikacji

[P1] Kamola M., **Arabas P.**, *Sieci społeczne i technologiczne. Jak zrozumieć, jak wykorzystać*, 2018, Warszawa, Wydawnictwo Naukowe PWN, 254 str., ISBN 978-83-01-19917-3, **80 pkt. MNiSW**.

[16] Przez spójność sieci rozumiem w tym przypadku stan, w którym istnieje co najmniej jedna ścieżka między każdym z węzłów a stacją bazową.

[17] Np. solwer CPLEX wchodzący w skład pakietu AMPL (https://ampl.com/products/solvers/solvers-we-sell/cplex/).

[**P2**] Kamola M., **Arabas P.**, *Improving Time-Series Demand Modeling in Hospitality Business by Analytics of Public Event Datasets*, IEEE Access, 2020, vol. 8, ss. 53666-53677, **100 pkt. MNiSW, IF=3,367**.

[**P3**] **Arabas P.**, Karpowicz M., *Częstość występowania wybranych triad w sieci połączeń między systemami autonomicznymi jako wskaźnik niektórych typów anomalii ruchu*, Przegląd Telekomunikacyjny – Wiadomości Telekomunikacyjne, 2016, nr 8-9 2016, ss. 1179-1184, **9 (obecnie 20) pkt. MNiSW**.

[**P4**] Gruszczyński W., **Arabas P.**, *Application of Social Network Inferred Data to Churn Modeling in Telecoms*, Journal of Telecommunications and Information Technology, 2016, nr 2, ss. 77-86, **12 (obecnie 40) pkt. MNiSW**.

[**P5**] Kamola M., **Arabas P.**, *Network Resilience Analysis: Review of Concepts and a Country-Level Case Study*, Computer Science, 2014, vol. 15, nr 3, ss. 311-327, **12 (obecnie 40) pkt. MNiSW**.

[**P6**] Kamola M., **Arabas P.**, *Dynamically established transmission paths in the future Internet – proposal of a framework*, Bulletin of the Polish Academy of Sciences Technical Sciences, 2011, Vol. 59, nr 3, ss. 357-366, **obecnie 100 pkt. MNiSW**.

[**P7**] Tarasiuk H. , Śliwiński J., **Arabas P.**, Jaskóła P., Góralski W., *Performance Evaluation of Signaling in the IP QoS System*, Journal of Telecommunications and Information Technology, 2011, nr 3/2011, ss. 12-20, **7 (obecnie 40) pkt. MNiSW**.

[**P8**] Kamola M., **Arabas P.**, *Wykorzystanie technologii Vecta Star do przekazu audiowizualnego wysokiej rozdzielczości*, Przegląd Telekomunikacyjny – Wiadomości Telekomunikacyjne, 2009, nr 8-9 2009, ss. 1508-1513, **4 (obecnie 20) pkt. MNiSW**.

[**P9**] Skłodowski J., **Arabas P.**, *Wykorzystanie drzew sufiksowych do efektywnej prezentacji podobieństw sesji z systemu pułapek honeypot*, Cybersecurity and Law, nr 1 (9) 2023, ss. 298-315, **70 pkt. MNiSW**.

5.1 Sieci społeczne i złożone jako źródło danych w procesie modelowania

Szeroko rozumiane sieci, nie tylko telekomunikacyjne, ale również społeczne czy też złożone stanowią centralny punkt prowadzonych przeze mnie badań. Powiązanie tych zagadnień nie jest przypadkowe. Warto wspomnieć, że prace, które doprowadziły do powstania Internetu (np. [8]) zbiegły się w czasie z pierwszymi istotnymi osiągnięciami dotyczącymi sieci losowych – [19], które po latach przyczyniły się do powstania nowej dziedziny, związanej z badaniem sieci społecznych, czy podchodząc szerzej sieci złożonych [7][18]. W istocie propozycja nieregularnej i pozbawionej hierarchii topologii sieci była wynikiem analiz prowadzonych na sieci losowej.

[18]Sieć losowa, w szczególności opisana przez P. Erdősa i A. Rényiego i nazwana od ich nazwisk sieć ER, nie jest siecią złożoną. Decyduje o tym prostszy rozkład stopni wierzchołków i co za tym idzie brak bezskalowości. Jednakże, ze względu na wspólne cechy i występujące w niej procesy, np. perkolację, czy małe światy, jest ona chętnie wykorzystywana w badaniach jako uproszczony przykład.

Konsekwencją powszechnego wykorzystania technologii sieciowych jest dostępność różnorodnych danych, tak w postaci ustrukturyzowanych, celowo przygotowanych zbiorów, jak też tworzonych ad-hoc wpisów dostępnych w Internecie. Stwarza to szerokie możliwości ich wykorzystania do modelowania i analizy zjawisk zachodzących w sferze związanej bezpośrednio z procesami społecznymi, jak i technologicznymi. Istotną, wspólną, cechą samych danych, jak i badanych tutaj procesów, jest występowanie wzajemnych powiązań mających charakter sieci, a w wielu przypadkach sieci złożonej. Termin sieć złożona został celowo użyty zamiast popularniejszego terminu sieć społeczna ze względu na jego szersze znaczenie. Sieci społeczne wiąże się z pionierskimi pracami zespołu A-L. Barabásiego, w szczególności z pracą [7] A-L. Barabásiego i R. Albert, ale nawet w tej pracy badacze ci nie ograniczali się wyłącznie do sieci opisujących bezpośrednio interakcje między ludźmi. Wiele przykładów to tzw. sieci technologiczne, opisujące zależności między urządzeniami, infrastrukturą telekomunikacyjną czy informatyczną, a także złożone systemy biologiczne i oddziaływania chemiczne. Ich wspólną cechą jest złożoność, rozumiana nie tylko jako wielki rozmiar, ale właściwość bezskalowości, skutkująca występowaniem takich fenomenów jak anomalna liczba połączeń niektórych węzłów, np. istnienie tzw. hubów czy superwęzłów, albo występowanie stosunkowo krótkich, proporcjonalnych do logarytmu z liczby węzłów w sieci, tras. Istotą proponowanego przeze mnie podejścia do analizy takich sieci jest przekonanie o istnieniu związków przyczynowo-skutkowych wynikających zarówno z cech wykorzystanej technologii jak też uwarunkowań psychologicznych. Rozumiem przez to fakt, że złożoność sieci technologicznej, np. sieci połączeń lotniczych czy sieci komputerowej, jest wynikiem, czy wręcz odwzorowaniem, złożoności relacji łączących użytkowników tej sieci. Rzeczywiście, realizacja połączenia lotniczego jest zazwyczaj poprzedzona analizą jego dochodowości. Pośrednio jest to więc dowód na występowanie potrzeby przemieszczania się, czyli też kontaktu, między osobami znajdującymi się w połączonych w ten sposób miastach. Również wybór lotnisk przesiadkowych nie jest przypadkowy. Wynika on z dogodności połączeń lub innych korzyści związanych z odwiedzanym przy okazji miejscem. Podobnie w przypadku sieci komputerowej. O ile istnienie połączenia między dwoma węzłami może wynikać wyłącznie z przesłanek technologicznych, to zdefiniowanie i wykorzystanie trasy między węzłami końcowymi zależy już od występowania potrzeby transmisji informacji, których końcowym odbiorcą jest zazwyczaj człowiek. Stąd też możliwe jest wykorzystanie danych pochodzących z sieci technologicznych do rekonstrukcji sieci kontaktów między ich użytkownikami, jak również wykorzystanie danych o ich zachowaniach do modelowania procesów zachodzących na rynku. Należy tu podkreślić, że dane te mogą pochodzić z różnych źródeł: bezpośrednio z topologii połączeń sieci społecznej, z informacji w niej publikowanych, jak również z oficjalnych, ustrukturyzowanych rejestrów. Wszystkie te zagadnienia poruszamy wspólnie z Mariuszem Kamolą w monografii [P1]. Jest to druga wydana w języku polskim pozycja ujmująca całościowo tematykę sieci złożonych. Prezentujemy w niej szereg podejść do modelowania i analizy tego typu sieci. Przedstawiamy również wyniki naszych prac badawczych, w których stosowaliśmy techniki analizy sieci społecznych do rozwiązania rzeczywistych problemów.

Podsumowując, prowadząc badania w zakresie sieci złożonych kierowałem się chęcią potwierdzenia następujących hipotez:

- Uwzględnienie danych z więcej niż jednego źródła pozwala na odkrycie w nich nietrywialnych zależności, przy czym szczególnie istotne są dane odwzorowujące sieć zależności między podmiotami podlegającymi analizie.

- Spośród dostępnych źródeł danych szczególnie istotne są te, które zawierają dane o czynnikach zewnętrznych mających wpływ na analizowany proces.
- Zachowania użytkowników sieci technologicznej i zachodzące w niej procesy znajdują odzwierciedlenie w jej topologii.

Modelowanie wpływu wydarzeń na popyt na usługi hotelarskie. W pracy [P2], w której wspólnie z Mariuszem Kamolą podsumowaliśmy wyniki naszych prac w projekcie *POIR.01.01.01-00-0050/15, „Hotels' Management Optimizer (HMO) – Pricing, Forecasting, Distribution"* zaproponowałem ogólną koncepcję addytywnego modelu popytu. Uwzględnia on trzy składniki odwzorowujące: trend, sezonowość i wpływ czynników zewnętrznych. Przez trend rozumiałem stałą tendencję do wzrostu (bądź spadku) zapotrzebowania na usługi. Sezonowość ma tutaj szersze niż zazwyczaj znaczenie. Rozumiałem przez nią okresową zmienność popytu. Trzecim składnikiem był wpływ czynników zewnętrznych, identyfikowanych jako wydarzenia kulturalne, konferencje naukowe i branżowe, czy targi. Analiza danych wykazała, że dominuje okres długości tygodnia. Prowadzi to do konkluzji o istnieniu dwóch form korzystania z usług hotelarskich: biznesowej i rekreacyjnej. Model został zaimplementowany z użyciem pakietu Prophet[19], co pozwoliło wykorzystać krzywą logistyczną do modelowania trendu, a przez to odwzorować zjawisko nasycenia pojemności hotelu. Do wykrycia okresowości zastosowałem analizę fourierowską. Modelowanie wpływu zdarzeń zewnętrznych wykonałem rozszerzając koncepcję zdarzeń kalendarzowych modelu Prophet. W bazowym modelu pozwalały one na przypisanie wag do dni będących świętami. Zaproponowałem rozszerzenie tej kategorii na wskazane wcześniej wydarzenia. W podejściu tym tworzony był kalendarz wydarzeń należących do wygenerowanego wcześniej zestawu kategorii co pozwalało identyfikować związane z nimi wagi na podstawie danych historycznych. Zaimplementowany model został wykorzystany jako część modułu optymalizacji cen usług hotelarskich. Projekt ten był koordynowany przez firmę YieldPlanet.

Wykorzystanie właściwości topologicznych sieci społecznej do segmentacji użytkowników. Jednym z istotnych zagadnień, z którymi spotykają się firmy telekomunikacyjne, jest redukcja odejść klientów. Istnieją przesłanki [45, 25, 15] aby zbliżającą się decyzję o zmianie usługodawcy wiązać ze zmianą zachowania użytkownika, np. wykonywaniem połączeń na numery z innych sieci czy biura obsługi. Oznacza to, że można próbować przewidzieć taką decyzję wykorzystując dane ruchowe, tzw. CDR (Call Data Records). W publikacji [P4] pracy postawiłem hipotezę, że efektywność powszechnie stosowanych modeli regresyjnych można zwiększyć, przez segmentację klientów z wykorzystaniem informacji wydobytych z topologii tworzonej przez nich sieci połączeń. Zaletą takiego rozwiązania jest uniknięcie dostępu do bazy klientów, utrudnionego ze względu na ograniczenia prawne, przede wszystkim RODO. Wykorzystanie topologii sieci oznacza, że szczegółowe dane użytkowników nie są niezbędne i można działać na zanonimizowanych rekordach CDR. Co więcej, mimo użycia tego samego zbioru danych możliwe jest odkrycie w nim dodatkowych, nieuwzględnianych wcześniej informacji. Analizowane dane obejmujące około 130 milionów rekordów CDR dla 299 tysięcy indywidualnych użytkowników[20], dotyczyły sieci o lokalnym zasięgu. Sieć uzyskana wprost, tzn. poprzez połączenie

[19] https://facebook.github.io/prophet/

[20] Pozostałe 16 tysięcy to użytkownicy instytucjonalni, co było zaznaczone w rekordach CDR, względnie różnego rodzaju numery techniczne zapewniające obsługę i działanie sieci. Ze względu na stosunkowo małą

użytkowników sieci wykonujących między sobą połączenia, nie była zbyt gęsta i spójna. Udało się w niej wyróżnić siedem składowych spójnych zawierających razem około 281 tysięcy użytkowników. W związku z tym wykonałem próbę rekonstrukcji połączeń między użytkownikami przez zbudowanie sieci dwudzielnej z wykorzystaniem numerów nienależących do dostawcy usług i jej projekcję. W tak utworzonej sieci połączenie między dwoma użytkownikami oznaczało, że wykonywali oni rozmowy z tymi samymi użytkownikami zewnętrznych sieci telekomunikacyjnych. Podejście takie pozwoliło stworzyć ósmy segment, w którym znaleźli się użytkownicy połączeni wspomnianą siecią, a nie występujący w składowych spójnych pierwotnej sieci. Segment dziewiąty, zawierał użytkowników, dla których nie znaleziono żadnego połączenia.

Dla tak określonych segmentów zbudowałem hybrydowy model regresyjny będący sumą dziewięciu modeli wykorzystujących regresję logistyczną. Mimo względnej prostoty modeli składowych udało się osiągnąć poprawę predykcji odejść klientów. Wskaźnik F1 dla modelu hybrydowego wzrósł do 0,599, w porównaniu z wartością 0,558 dla pojedynczego modelu uczonego na całym zbiorze użytkowników. Co istotne wyniki dla niektórych modeli składowych były lepsze co świadczy, że w tych przypadkach tak wykonana segmentacja wykryła pewne cechy wspólne użytkowników mające odwzorowanie w ich sposobie użytkowania usług. Co więcej, analiza k-rdzeni w ramach wyznaczonych segmentów pokazała różnice między segmentami. W szczególności segmenty, dla których modele regresyjne wykazały się niższą skutecznością miały inną strukturę k-rdzeni od pozostałych. Można to interpretować jako kolejny argument za tezą, że sposób, w jaki użytkownicy korzystają z usług znajduje odbicie w topologii sieci.

Wykorzystanie deklaracji z bazy RIR i próbek tras BGP do rekonstrukcji grafu połączeń systemów autonomicznych. Niezawodność sieci telekomunikacyjnej zależy, w dużym stopniu, od istnienia redundantnych połączeń. Typowo, system autonomiczny[21] powinien dysponować co najmniej dwoma łączami do innych, sąsiadujących systemów autonomicznych. W praktyce liczba łączy zależy od skali i profilu działalności, a także roli pełnionej przez system w Internecie. Systemy dostawców usług zapewniają swoim klientom dostęp do Internetu. Jest to więc relacja klient-dostawca wiążąca się z ponoszeniem przez klienta kosztów korzystania z usług. W uzasadnionych wzajemną korzyścią przypadkach systemy autonomiczne mogą wymieniać się ruchem na zasadach partnerskich, bezpłatnie. Jest to tzw. relacja *peeringu*. Jako szczególną jego odmianę można traktować połączenie pomiędzy systemami autonomicznymi pod zarządem tej samej organizacji, tzw. systemami bliźniaczymi (ang. *syblings*). W pracy [P5] opisałem procedurę rekonstrukcji sieci połączeń między systemami autonomicznymi podmiotów zarejestrowanych w Polsce. Wykorzystałem do tego dane z dwóch źródeł: rejestru systemów autonomicznych dostępnego na stronach odpowiedzialnej organizacji, w tym wypadku RIPE[22] oraz próbkach tras BGP dostarczanych przez projekt CAIDA[23]. Należy zauważyć, że oba źródła zawierają dane, które są niekompletne, stąd konieczność umiejętnego ich połączenia. Dane w bazie RIPE mają charakter deklaratywny i są zmieniane stosunkowo rzadko. Oznacza to, że w wielu przypadkach, szczególnie dla dużych dostawców usług sieciowych,

liczbę i potencjalne nietypowe zachowanie numery te nie były uwzględniane w opisanej analizie.

[21]System autonomiczny, (AS - ang. *autonomous system*) jest zbiorem sieci IP (tzw. prefiksów) administrowanych przez jedną instytucję wprowadzającą spójną politykę trasowania przy pomocy protokołu trasowania zewnętrznego BGP (Border Gateway Protocol).

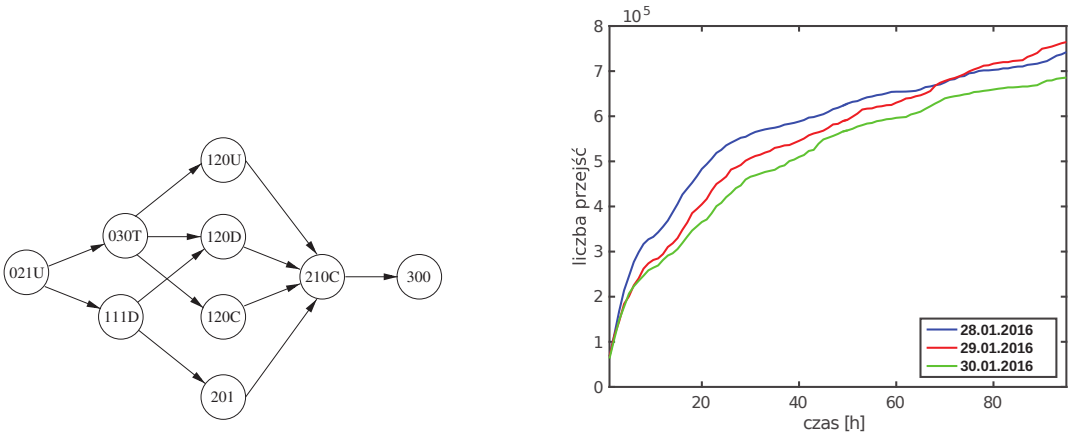
[22]Réseaux IP Européens

[23]https://catalog.caida.org

będą tam udostępnione jedynie połączenia odpowiadające umowom długookresowym. Zaletą natomiast jest możliwość jednoznacznego określenia rodzaju relacji na drodze analizy polityk rozgłaszania i przyjmowania tras (tzw. sekcji *export* i *import*). Bazy projektu CAIDA zawierają wstępnie przetworzone trasy BGP zarejestrowane przez pewną liczbę (rzędu kilkudziesięciu) próbników. W naturalny sposób oddają one stan sieci w chwili rejestracji próbek, a więc widoczne są w nich połączenia nieuwzględnione w bazach RIPE, wynikające np. z kontraktów krótkoterminowych zawieranych w celu równoważenia obciążenia. Połączenia te są ekstrahowane z tras BGP rejestrowanych w ograniczonej liczbie punktów, a oznaczenie typu relacji (klient-dostawca, peering) wynika głównie z analizy przepływu ruchu. Stąd koncepcja połączenia obu rodzajów danych w celu stworzenia spójnej sieci. Dzięki opisaniu połączeń rodzajami relacji możliwe było zbadanie udziału poszczególnych typów łączy w polskiej sieci Internet i wpływu, jaki to ma na niezawodność połączeń. Co ciekawe, w chwili przeprowadzenia badania wykorzystanie peeringu było stosunkowo niskie (około 15%). Było to naturalną konsekwencją stosunkowo niewielkiego udziału dużych dostawców usług w badanej sieci. Tym bardziej ciekawe jest jednak, że wpływ usunięcia połączeń realizowanych jako peering był, dla niezawodności sieci, nieznaczny. Świadczy to o wysoce hierarchicznej strukturze sieci, w której dostawcy usług są jednocześnie klientami innych dostawców a peering jest nawiązywany tylko między równorzędnymi sieciami.

5.2 Bezpieczeństwo sieciowe

Wpływ ataków sieciowych typu DoS na topologię obserwowanych połączeń. W pracy [P3] prezentuję wyniki badań wpływu ataków sieciowych typu odmowa dostępu (DoS – Denial of Service) na topologię obserwowanych w sieci połączeń. Hipoteza o wpływie ataku, czyli sytuacji, gdy ruch generowany jest w sposób nietypowy, wynika z wcześniejszego założenia o bezpośrednim powiązaniu relacji, na jakich nawiązywane są połączenia, z ich zapotrzebowaniem na informacje. W takich warunkach przeprowadzenie ataku powinno polegać na przesłaniu istotnej objętości danych na niewykorzystywanych wcześniej trasach, a więc wygenerowaniu przepływów, które mogą być uznane za nieprzydatne, czy wręcz szkodliwe. W celu sprawdzenia postawionej tezy wykorzystałem próbki ruchu zebrane za pomocą protokołu NetFlow z jednego z ruterów w rzeczywistej sieci szkieletowej. Tak obserwowane połączenia pozwalają na zbudowanie sieci będącej ograniczonym, lecz nadal stosunkowo rozległym, wycinkiem Internetu. W rozważanym przypadku budowałem sieci z zebranych w przeciągu kolejnych godzin danych, co skutkowało liczbą łącz wahającą się pomiędzy 2,5 a 6 tysiącami. Z tego powodu analiza topologii mogła być wykonana tylko metodami statystycznymi stosowanymi m.in. w analizie sieci złożonych, przez porównanie stosunkowo prostych do wyznaczenia wskaźników takich jak liczba triad. Skupiłem się na triadach pełnych stanowiących przypuszczalnie rdzeń sieci, czyli węzły połączone dwustronną wymianą danych oraz triadach typu 021U. Triady takie zawierają dwa łącza jednokierunkowe skierowane do tego samego węzła co w oczywisty sposób może, przez fakt braku transmisji zwrotnej, świadczyć o przesyłaniu niechcianego ruchu. Analiza zmian liczby triad wykazała, że triady pełne są w badanej sieci stosunkowo rzadkie, a ich liczba jest względnie stabilna. Wprost przeciwnie triady typu 021U występują znacznie częściej i można zaobserwować krótkie przedziały czasu, gdy ich liczba gwałtownie wzrasta. Jak wykazała dalsza analiza, w tych przedziałach można również zaobserwować znaczący wzrost ruchu do usług DNS i NTP, co jest o tyle znamienne, że



Rysunek 20: Możliwe ścieżki domykania się triady 021U (rysunek lewy) i dynamika przejścia triady 021U w kolejną triadę w grafie 111D dla trzech dni, w których obserwowano anomalie (wykres prawy).

były to w badanym czasie typowe wektory ataku. Wskazane anomalie są o tyle łatwe do wykrycia, że usługi DNS, a w szczególności NTP przy prawidłowym działaniu generują niewielkie objętości ruchu. W typowej sieci społecznej triady niepełne z czasem stopniowo domykają się. W badanej sieci dla triad 021U proces taki występuje rzadko i powoli co pokazuje rysunek 20. Analizując trzy przypadki wystąpienia anomalii, obserwowałem czasy rzędu nawet kilkunastu godzin, co przewyższa zarówno czas trwania ataku jak i typowego, uprawnionego połączenia sieciowego. Obserwacja ta może potwierdzać nietypową naturę triad. Prawdopodobnie domknięcie nie ma bezpośredniego związku z ruchem obserwowanym w chwili ataku, być może jest wręcz wynikiem działań mających na celu mitygację skutków zdarzenia.

Systemy detekcji i analizy ataków sieciowych. W ramach prowadzonych w NASK prac brałem udział w projektach ARAKIS-Enterpise, ARAKIS-GOV i FLDX. System ARAKIS-Enterpise służy do wzmocnienia bezpieczeństwa sieciowego używającej go instytucji przez zbieranie i analizę danych z podsystemu pułapek sieciowych umieszczonych wewnątrz i na zewnątrz chronionej infrastruktury. Pułapka sieciowa (ang. *honeypot*) emuluje określony zestaw usług, stwarzając iluzję systemu produkcyjnego. Połączenia nawiązywane z pułapką są analizowane przez system i służą do generowania alarmów. Alarmy poddawane są dalszej obróbce, w tym wzbogacaniu o dodatkowe informacje, np. geolokalizację, łączeniu z zapisanymi próbkami ruchu, zarejestrowanymi sesjami[24] i plikami. Dalszym etapem obróbki zebranych danych jest agregacja, czyli łączenie podobnych alarmów występujących w określonym przedziale czasu, a także generacja sygnatur, które mogą być wykorzystane przez systemy wykrywania zagrożeń[25]. Dodatkową funkcjonalnością systemu jest możliwość integracji danych pochodzących z systemów zewnętrznych: zapór ogniowych (ang. *firewall*) oraz oprogramowania antywirusowego. Również te dane są źródłem alarmów i mogą być korelowane z pozostałymi alarmami. Służy temu wyspecjalizowany język zapytań AQL. System ARAKIS-Enterprise jest przeznaczony dla instytucji

[24]W przypadku usług interpretujących polecenia jak serwery baz danych czy powłoka systemu operacyjnego.

[25]Głównie systemy IDS – Intrusion Detection Systems.

zarządzających infrastrukturą krytyczną, stąd w zestawie pułapek sieciowych znajduje się również opracowana w NASK wersja pułapki CONPOT[26] emulująca protokoły obsługiwane przez wybrane sterowniki PLC i towarzyszące im systemy SCADA. System ARAKIS-GOV stanowi rozwinięcie systemu ARAKIS-Enterprise przeznaczone do wykorzystania w instytucjach państwowych w ramach Krajowego Systemu Cyberbezpieczeństwa i jest realizowany w ramach umowy niejawnej. System FLDX służy do wykrywania i mitygacji rozproszonych ataków typu odmowa dostępu (ang. *DDoS – Distributed Denial of Service*) przeprowadzanych na infrastrukturę sieciową operatora. Zaimplementowane w systemie detektory porównują statystyki ruchu sieciowego odbierane z nadzorowanych urządzeń sieciowych z adaptowanym na bieżąco modelem dynamicznym. Umożliwia to szybkie wykrycie anomalii i wdrożenie odpowiednich filtrów na urządzeniach brzegowych chronionej sieci. Co ważne, tak zidentyfikowane przepływy nie są odrzucane, lecz tłumione w stopniu zależnym od obciążenia sieci zgodnie z algorytmem sprawiedliwego podziału łącza. Rozwiązanie takie wyklucza możliwość zablokowania uprawnionego ruchu, zapewniając jednocześnie dostateczne oddziaływanie na ruch złośliwy.

Wspomniane projekty wymagały przeprowadzenia szeregu prac o charakterze badawczym, rozwojowym i wdrożeniowym, w których uczestniczyłem. Do moich osiągnięć zaliczają się:

- opracowanie architektury klastra obliczeniowego dla systemu ARAKIS,
- opracowanie architektury sieciowej i zabezpieczeń dla systemu ARAKIS,
- opracowanie architektury podsystemu analizy złośliwego oprogramowania (ang. *sandbox*) dla systemu ARAKIS,
- opracowanie architektury mitygacji dla nowej wersji systemu FLDX,
- opracowanie nowych analiz dla systemu ARAKIS.

Dwa ostatnie osiągnięcia wiązały się z budową instalacji laboratoryjnej i przeprowadzeniem szeregu eksperymentów. Laboratoryjna instalacja systemu FLDX pozwoliła opracować sposób, w jaki mogą być na urządzeniach sieciowych implementowane filtry tłumiące niechciane przepływy. Jest to zadanie bardzo ambitne, gdyż w okresach zmasowanych ataków może być konieczne tłumienie tysięcy przepływów co stawia bardzo wysokie wymagania tak urządzeniom sieciowym jak również sterującemu je oprogramowaniu. Jak pokazują wyniki badań, typowe urządzenia, których wykorzystanie jest brane pod uwagę[27] mają ograniczoną zdolność filtracji i kształtowania ruchu, co więcej potrafią stracić stabilność w sytuacji przeciążenia. Z tego powodu istotne jest określenie limitów i opracowanie metod agregacji filtrów.

Analizy dla nowej wersji systemu ARAKIS były przedmiotem projektu wewnętrznego NASK, w którym brał udział m.in. zespół pod moim kierownictwem. Wynikiem prac są propozycje metod klastryzacji złośliwych plików, analizy sieci połączeń do honeypotów oraz grupowania sesji opisane w pracy [P9]. Ostatnia z metod wykorzystuje wstępną klasyfikację poleceń systemu Linux do stworzenia ograniczonego słownika. W słowniku tym

[26] https://conpot.org/.

[27] Należy brać pod uwagę przełączniki sieciowe i rutery typowo instalowane w węzłach sieci. Urządzenia te dysponują znaczną przepustowością. Możliwości śledzenia przepływów są jednak ograniczone do pojedynczych tysięcy adresów w przypadku przełączników sieciowych i o około dziesięć razy więcej dla ruterów.

jednoliterowe symbole odpowiadają grupom poleceń związanych z istotnymi dla analizy sesji kategoriami poleceń, takimi jak np. operacje na plikach czy uruchamianie procesów. Takie przekształcenie zapisu sesji pozwala pominąć drobne różnice, zachowując jednocześnie znaczenie. Przekodowane sesje są następnie grupowane z wykorzystaniem drzew sufiksowych. Metoda ta pozwala nie tylko na hierarchiczne grupowanie podobnych sesji, lecz również na szybkie wyszukiwanie wybranych podciągów.

Oprócz wspomnianych prac w NASK brałem udział w projekcie NCBiR Cybermine (Centrum monitorowania instalacji przemysłowych w podziemnych zakładach górniczych i wykrywania cyberzagrożeń), którego przedmiotem było opracowanie i wdrożenie w jednej z kopalń Jastrzębskiej Spółki Węglowej systemu monitorowania sieci informatycznych i przemysłowych. System składa się z sond zbierających dane z sieci naziemnych, koncentratora analizującego ruch w sieciach podziemnych oraz centralnego systemu integrującego i prezentującego dane. W ramach tego projektu uczestniczyłem w zadaniach związanych z gromadzeniem i analizą danych sieciowych (byłem kierownikiem zadania) oraz wykorzystaniem metod sztucznej inteligencji do analizy danych, w tym złośliwego oprogramowania, co zostało podsumowane w artykule [U12].

5.3 Sieci IP z dynamiczną alokacją połaczeń

Jednym z proponowanych rozwiązań problemu zatorów sieciowych (ang. congestion) i związanej z nimi niedostatecznej jakości przesyłu danych w sieci jest wprowadzenie dynamicznie zawieranych kontraktów na transmisję danych w kanale o określonej jakości. Podejście takie stara się, oprócz rozwiązania problemu technicznego niedostatecznej jakości, wprowadzić również nowy rodzaj rozliczeń za korzystanie z usług sieciowych prowadzący do poprawy ich rentowności. W powszechnie przyjętym modelu usługi świadczone są bez gwarancji jakości na danej relacji. Zamiast tego dostawca zobowiązuje się do spełnienia określonych w umowie parametrów dotyczących zazwyczaj np. przepustowości i dostępności łącza abonenckiego. Oznacza to brak gwarancji dla konkretnych usług, np. przesyłania obrazu na wybranej relacji i w konsekwencji konieczność zakupu znacznie przewymiarowanego łącza w nadziei, iż wystarczy to do zapewnienia akceptowalnej jakości. Rozwiązanie takie skutkuje nie tylko problemami z osiągnięciem niezbędnej jakości, ale również dość niskim wykorzystaniem łącz i niewielkimi, w stosunku do wymaganych inwestycji wynikających z przewymiarowania sieci, dochodami dostawców. Zawieranie kontraktów dynamicznych, tak detalicznych, jak i hurtowych, może poprawić tę sytuację, dzięki, z jednej strony zapewnieniu jakości transmisjom, które tego wymagają, z drugiej zaś wprowadzeniu rozliczeń, w których opłaty są uzależnione od wykorzystywanych zasobów.

Architektura systemu dynamicznej kontraktacji usług sieciowych. W pracy [P6] wraz z współpracownikiem, zaproponowaliśmy model deregulacji świadczenia usług, w którym dzięki występowaniu wielu podmiotów zajmujących się pośredniczeniem w zestawieniu połączenia możliwe jest zapewnienie odpowiedniej skalowalności. Model ten uwzględnia hierarchię dostawców usług i umożliwia podmiotom zajmującym w niej wyższe pozycje wykorzystanie kontraktów hurtowych. Dzięki temu stan poszczególnych połączeń nawiązywanych przez użytkowników końcowych jest zapamiętywany w sposób rozproszony, w miarę możliwości u ich bezpośrednich usługodawców. Od strony technicznej, zgodnie z dostępnymi w chwili tworzenia koncepcji technologiami, proponowałem wykorzystanie elementów architektury usług zróżnicowanych (DiffServ – ang. *differentiated services*) [11] na poziomie końcowego dostawcy, wspomaganych przez wykorzystanie inżynierii ruchu

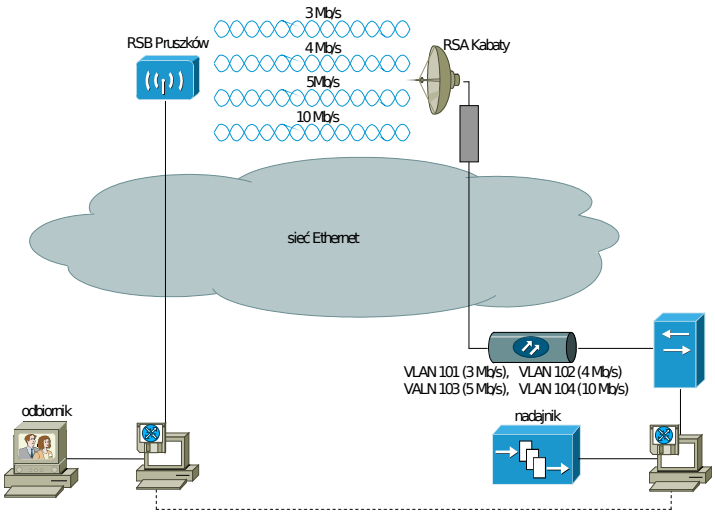
MPLS-TE[28] do realizacji połączeń hurtowych. Podejście takie pozwala z jednej strony ściśle nadzorować spełnienie warunków kontraktu na brzegu sieci, z drugiej zaś agregować połączenia w rdzeniu. Jest to istotne ze względów technologicznych, gdyż ułatwia skalowanie, jak również ekonomicznych dzięki wystąpieniu multipleksacji statystycznej. Pozwala ona na pełniejsze wykorzystania kanału transmisyjnego a przez to osiągnięcie zysku pokrywającego inwestycje.

Generator połączeń do testowania systemu IP-QoS. W ramach prac projektu PBZ-MNiSW-02-II/2007 – PBZ-ZR (01.01.2008 - 30.03.2011); „Usługi i sieci teleinformatyczne następnej generacji – aspekty techniczne, aplikacyjne i rynkowe", którego byłem wykonawcą opracowany został system IP-QoS umożliwiający dynamiczne zestawianie połączeń sieciowych z gwarancjami jakości, zgodnie z architekturą tzw. sieci następnej generacji (ang. *NGN – Next Generation Network*). Dla weryfikacji wydajności tego systemu konieczna była budowa generatora połączeń, czyli symulatora działań użytkowników sieci. Opracowany i wykonany przeze mnie symulator został przedstawiony w pracy [P7]. Odwzorowanie zachowania użytkownika sprowadzało się do generowania odstępów czasu między połączeniami oraz czasu trwania połączenia, zaś interakcja z systemem polegała na wykorzystaniu zdefiniowanych w projekcie interfejsów. Ze względu na różnorodne scenariusze uwzględnianie w projekcie konieczne było dostarczenie zróżnicowanych generatorów wskazanych parametrów, w szczególności zaś uwzględnienie występującego w strumieniu żądań samopodobieństwa. Wynika ono zarówno z zmiennego zapotrzebowania użytkowników na pobierane z sieci informacje, jak również znacznej zmienności ich objętości, co jest widoczne szczególnie w przypadku danych multimedialnych. Przeprowadzone eksperymenty obejmowały interakcję z kompletnym systemem zawierającym elementy programowe oraz fizyczną sieć transmisyjną zbudowaną z wykorzystaniem ruterów Cisco serii 7200 w rdzeniu i 1800 na styku abonenckim zgodnie z ówczesnym standardem. Badania potwierdziły możliwość wykorzystania zaproponowanego systemu w praktyce, wykazując w szczególności, stosunkowo dobrą odporność na zbitki zgłoszeń.

Stanowisko laboratoryjne i badania transmisji danych multimedialnych z wykorzystaniem sieci w technologii VectaStar. W czasie realizacji projektu WKP 1/1.4.1/1/2006/125/125/682/2007 „Platforma budowy usług multimedialnych nowej generacji dla sieci komputerowych i mobilnych" istotnym zagadnieniem, z którym borykali się dostawcy usług, był tzw. „problem ostatniej mili", czyli budowa odcinka sieci łączącego użytkownika końcowego z najbliższym punktem koncentracji[29]. O ile sieci szkieletowe dysponowały zazwyczaj dostateczną przepustowością to, szczególnie w obszarach o mniejszej gęstości zabudowy, sieci abonenckie były niedostępne lub oferowały niskie prędkości, typowe dla sieci telefonicznej. Problem ten próbowano rozwiązać z pomocą sieci radiowych. W pracy [P8] przedstawiłem konstrukcję stanowiska laboratoryjnego do badania transmisji wideo z wykorzystaniem sieci radiowej wykonanej w technologii VectaStar firmy Cambridge Broadband. Schemat stanowiska testowego przedstawia rysunek 21. Sieć ta składała się z stacji bazowych i abonenckich wyposażonych w interfejsy Ethernet. We-

[28]Z punktu widzenia omawianego rozwiązania najwazniejszym rozszerzeniem w stosunku do bazowego protokołu MPLS jest możliwość wykorzystania protokołu RSVP-TE do rezerwacji ścieżki z uwzględnieniem zasobów dostępnych w węzłach pośrednich. Protokół RSVP-Te jest opisany w RFC3209, https://datatracker.ietf.org/doc/html/rfc3209

[29]Problem ten istnieje oczywiście również dzisiaj, jednakże jego rozwiązanie jest prostsze dzięki upowszechnieniu względnie tanich i zapewniających wysoką przepustowość technologii, takich jak np. pasywne sieci optyczne.



Rysunek 21: Schemat stanowiska testowego.

wnętrznie posługiwała się protokołem ATM, co predysponowało ją do transmisji z ścisłymi gwarancjami jakości. Zaproponowane rozwiązanie, dzięki wykorzystaniu urządzeń ulokowanych w węzłach sieci NASK w Warszawie oraz w Pruszkowie, pozwalało na użycie dostępnej tam infrastruktury światłowodowej NASK do sprowadzenia ruchu z obu węzłów do laboratorium w siedzibie NASK. Ułatwiło to przeprowadzenie pomiarów dzięki możliwości połączenia maszyn pełniących rolę nadajnika i odbiornika dodatkowym łączem Ethernet zapewniającym niezawodną synchronizację czasu protokołem PTP,[30]. Mozliwy był więc precyzyjny pomiar opóźnienia i strat pakietów. Wykonane pomiary potwierdziły możliwość zapewnienie dobrej jakości transmisji dla ruchu wideo o standardowej rozdzielczości oraz w jakości HD-ready. Jednakże ostatni przypadek wymagał zaangażowania większości (10 Mb/s z dostępnych 16 Mb/s) zasobów transmisyjnych sektora. Oznaczało to, że, przy odległości rzędu 30 km, pojedyncza stacja bazowa może transmitować maksymalnie cztery strumienie HD-ready, przy czym każdy z nich musi należeć do innego sektora, co jest naturalnym argumentem za wykorzystaniem wspomnianych wcześniej dynamicznych kontraktów przesyłowych.

6 Osiągnięcia dydaktyczne, organizacyjne oraz popularyzatorskie

6.1 Dydaktyka i opieka naukowa

Pracując w Instytucie Automatyki i Informatyki Stosowanej Politechniki Warszawskiej prowadziłem i nadal prowadzę wykłady oraz zajęcia laboratoryjne, a także zajmowałem się opieką naukową nad pracami inżynierskimi i magisterskimi. Działalność ta stanowi ważne uzupełnienie własnych badań poprzez możliwość współpracy z studentami reprezentującymi świeże i często bardzo kreatywne podejście do realizowanych prac. Również przygotowanie wykładów i przede wszystkim laboratoriów pozwala na bezpośrednią weryfikację poznanych i opracowanych autorskich rozwiązań, na lepsze zrozumienie bada-

[30]Protokół PTP – *Precision Time Protocol* pozwala na synchronizację czasu z dokładnością rzędu μs.

nych zagadnień dzięki bezpośredniej dyskusji. Dodatkowo praca w laboratorium umożliwia z jednej strony kontakt z technologią z drugiej zaś pozwala przekazywać studentom doświadczenia zdobyte podczas prac wdrożeniowych.

Prowadzone wykłady i laboratoria

WSO – Wirtualne Środowiska Obliczeniowe – wykład dla studentów II stopnia, który opracowałem i uruchomiłem w 2022 r.

TASS – Techniki Analizy Sieci Społecznych – wykład dla studentów II stopnia opracowany z współpracownikiem, jestem kierownikiem przedmiotu, od 2015 r.

SST – Sieci i Sterowanie Systemami – wykład dla studentów II stopnia opracowany z zespołem współpracowników, prowadzony w latach 2014 - 2022.

PROZ – Programowanie obiektowe – wykład dla studentów I stopnia, z współpracownikiem, w latach 2008 - 2020.

SSK – Sterowanie Sieciami Komputerowymi – wykład dla studentów II stopnia, w którego opracowaniu i realizacji brałem udział, lata 2005-2007.

Sieci Komputerowe – wykład i laboratoria przygotowane i prowadzone razem z współpracownikami z NASK dla Wydziału Inżynierii Produkcji Szkoły Głównej Gospodarstwa Wiejskiego w roku akademickim 2005-2006. Byłem kierownikiem przedmiotu i autorem projektu laboratorium.

SKM – Sieci Komputerowe – laboratorium dla studentów I stopnia prowadzone w latach 2003 - 2022. Byłem jednym z twórców laboratorium.

CN – Computer Networks – laboratorium dla studentów I stopnia Wydziału Matematyki i Nauk Informacyjnych prowadzone w latach 2003 - 2016.

ECONE – Computer Networks – laboratorium dla studentów studiów w języku angielskim prowadzone w latach 2003 - 2016.

Laboratoria z trzech ostatnich przedmiotów dotyczą podobnej tematyki, prowadziłem je w wybranych semestrach w ramach większego zespołu, w wymiarze wynikającym z liczebności grup studenckich.

Oprócz tego, w wcześniejszych latach prowadziłem także laboratoria Administrowania Systemem Unix (ASU), a także sporadycznie zastępowałem prowadzącego wykłady z ASU, SKM i CN.

Za główne moje osiągnięcia w zakresie dydaktyki uważam opracowanie i uruchomienie dwóch nowych wykładów. W 2015 r., z współpracownikiem, uruchomiłem przedmiot Techniki Analizy Sieci Społecznych, którego jestem kierownikiem. W jego ramach omawiamy zagadnienia związane z sieciami społecznymi i złożonymi w odniesieniu do dostępnej technologii, co znajduje odzwierciedlenie w projektach realizowanych przez studentów. Przedmiot ten od początku cieszy się dużą popularnością, a sądząc po opiniach zainteresowanych wynika to w dużej mierze właśnie z możliwości realizacji zadań związanych z otaczającą ich rzeczywistością. Doświadczenia zdobyte podczas prowadzenia wykładów miały istotny wpływ na postać monografii [P1] – „Sieci społeczne i technologiczne. Jak zrozumieć, jak wykorzystać" napisanej wspólnie z dr hab. inż. Mariuszem Kamolą i wydanej przez PWN.

Drugi przedmiot: Wirtualne Środowiska Obliczeniowe, który został uruchomiony w 2022 r. opracowałem samodzielnie. Przedmiot ten jest wynikiem moich doświadczeń zebranych w trakcie realizacji projektów związanych z budową dużych systemów informatycznych i przygotowaniem dla nich infrastruktury obliczeniowej i sieciowej. W ramach tego wykładu staram się przybliżyć, często dziś pomijane, zagadnienia związane z budową fizycznej infrastruktury obliczeniowej i wspierającej ją sieci. Zwracam uwagę na zrozumienie roli, jaką bezpieczna i niezawodna infrastruktura pełni w realizacji projektów informatycznych i dlaczego zastąpienie jej dzierżawą zasobów obliczeniowych nie zawsze jest dobrym rozwiązaniem – tak ze względu bezpieczeństwa danych jak i zachowania swobody wyboru technologii. Ten przedmiot również cieszy się dużym zainteresowaniem studentów.

Nieco innym wyzwaniem było przygotowanie przedmiotu Sieci Komputerowe dla Wydziału Inżynierii Produkcji Szkoły Głównej Gospodarstwa Wiejskiego. Było to pierwsze wdrożenie przedmiotu związanego z budową laboratorium sieciowego na Wydziale Inżynierii Produkcji i wymagało zbudowania od podstaw infrastruktury pozwalającej na prowadzenie zajęć przy bardzo ograniczonym budżecie. Zadanie to udało się zrealizować dzięki wykorzystaniu taniego sprzętu klasy konsumenckiej i oryginalnemu użyciu otwartego oprogramowania. Wykłady były kontynuowane w następnych semestrach przez przeszkolony zespół z Wydziału Inżynierii Produkcji.

Istotnym elementem pracy dydaktycznej jest również prowadzenie prac dyplomowych na studiach I i II stopnia. Byłem promotorem 15 prac inżynierskich i 19 prac magisterskich.

6.2 Działalność organizacyjna i udział w dyskursie naukowym

6.2.1 Działania organizacyjne

- Sekretarz seminarium w Zespole Złożonych Systemów.
- Kierownik Zespołu Inżynierii Systemów Sieciowych NASK – w latach 2018-2022.
- Kierownik Laboratorium Zespołu Inżynierii Systemów Informatycznych NASK – w latach 2016-2018.
- Kierownik Zadania w projekcie NCBiR „Centrum monitorowania instalacji przemysłowych w podziemnych zakładach górniczych i wykrywania cyberzagrożeń" nr: POIR.01.01.01-00-0180/22 – w roku 2023.

W ramach prac związanych z prowadzoną dydaktyką zajmuję się od 2003 r. wraz z Kierownikiem Laboratorium zarządzaniem i utrzymaniem laboratorium, w którym są prowadzone zajęcia z przedmiotów SKM, CN i ECONE. W trakcie tego czasu laboratorium było kilkukrotnie modernizowane, co wiązało się z koniecznością stopniowej wymiany sprzętu i opracowania nowych lub poprawionych instrukcji laboratoryjnych, a także przeszkolenia nowych członków zespołu prowadzącego zajęcia.

6.2.2 Udział w dyskursie naukowym

- Członek komitetu naukowego konferencji: High Performance Modeling and Simulation (HiPMoS) w latach 2015-2017, 22nd International Teletraffic Congress (ITC) Specialist Seminar on Energy Efficient and Green Networking (ITC SSEEGN 2013),

Technologies and Materials for Renewable Energy, Environment and Sustainability (TMREES14).

- Kilkadziesiąt recenzji dla czasopism, m.in. IEEE Journal on Selected Areas in Communications, IEEE Transactions on Services Computing, MDPI Sensors, oraz konferencji naukowych, lista recenzji w załączniku 4.
- Kilkadziesiąt recenzji prac dyplomowych na Wydziale Elektroniki Technik Informacyjnych PW.

6.2.3 Działalność popularyzatorska i otrzymane nagrody

Działalność popularyzatorska

- Udział w prezentacjach analiz rynku domen dla klientów NASK w latach 2012-2016.
- Artykuł „M. Karpowicz, P. Arabas, Projekt ECONET: energooszczędne technologie sieciowe" w Biuletynie NASK, nr 1, 2014.
- Udział w prezentacji osiągnięć Zespołu Złożonych Systemów WEiTI PW w zakresie sterowania sieciami optymalizacji taryf na forum gospodarczym Telekomunikacja-Internet-Media-Elektronika 2011.
- Udział w prezentacji prototypu systemu IPQoS na konferfencji KSTiT 2010.
- Udział w prezentacji osiągnięć Pracowni Sterowania Siecią NASK na konferencji SECURE 2007.

Otrzymane nagrody

- Nagroda Lider Bezpieczeństwa Państwa 2022 za system Arakis Enterprise.
- Nagroda zespołowa I stopnia Rektora PW za osiągnięca naukowe w roku 2022.
- Nagroda zespołowa III stopnia Rektora PW za osiągnięcia dydaktyczne w r. akad. 2014/2015.
- Nagroda „*Best Paper Award*" za artykuł [U4].
- Nagroda zespołowa I stopnia Rektora PW za osiągnięca naukowe w latach 2012-2013.
- Nagroda indywidualna II stopnia Rektora PW, za osiągnięcia naukowe w roku 2004.
- Nagroda zespołowa III stopnia Rektora PW za osiągnięcia naukowe – prace w projekcie 5 Programu Ramowego UE Quality of Service and Pricing Differentiation for IP Services (QOSIPS), 2003.

Wykaz publikacji uzupełniających osiągnięcie naukowe

[**U1**] Niewiadomska-Szynkiewicz, E., **Arabas, P.**, *Resource Management System for HPC Computing*, w: Advances in Intelligent Systems and Computing, vol. 743, Springer, ss. 52-61, 2018, DOI:10.1007/978-3-319-77179-3_5

[**U2**] Niewiadomska-Szynkiewicz E., **Arabas P.**, *Energooszczędne centrum przetwarzania danych*, Przegląd Telekomunikacyjny – Wiadomości Telekomunikacyjne , SIGMA NOT, nr 8/9, 2018, ss. 609-614, DOI:10.15199/59.2018.8-9.13

[**U3**] Kamola M., **Arabas P.**, Jaskóła P., Niewiadomska-Szynkiewicz E., Malinowski K., Karpowicz M., Sikora A., Mincer M., Marks M., *ECONET – energooszczędne sieci IP*, Przegląd Telekomunikacyjny-Wiadomości Telekomunikacyjne, nr 8-9, ss. 964-970, 2013.

[**U4**] Niewiadomska-Szynkiewicz, E., Sikora, A., **Arabas, P.**, Malinowski K., *Energy-saving management in computer networks*, Australian Journal of Electrical and Electronics Engineering, vol. 12, nr 3, ss. 242-252, 2015, DOI:10.1080/1448837X.2015.1093002

[**U5**] Niewiadomska-Szynkiewicz E., Sikora A., **Arabas P.**, Kamola M., Malinowski K., Jaskóła P., Marks M., *Network-Wide Power Management in Computer Networks*, Proceedings of SSEEGN 2013 22nd ITC Specialist Seminar on Energy Efficient and Green Networking, vol. 1, ss. 25-30, 2013, IEEE.

[**U6**] **Arabas, P.**, Malinowski, K., Sikora, A., *On Formulation of a Network Energy Saving Optimization Problem*, w: 2012 Fourth International Conference on Communications and Electronics (ICCE), 2012, ss. 227-232, DOI:10.1109/CCE.2012.6315903.

[**U7**] **Arabas P.**, Jaskóła, P., *Model energetyczny rutera programowego – pomiary i identyfikacja*, Przegląd Telekomunikacyjny, Wiadomości Telekomunikacyjne; vol. 8-9, ss. 1014-1020, 2014.

[**U8**] Jaskóła P., **Arabas P.**, Karbowski A., *Combined Calculation of Optimal Routing and Bandwidth Allocation in Energy Aware Networks*, Proceedings of the 2014 26th International Teletraffic Congress (ITC), 2014, IEEE, ss. 1-6.

[**U9**] Kamola M., Niewiadomska-Szynkiewicz, E., **Arabas, P.**, Sikora A., *Energy-saving algorithms for the control of backbone networks: A survey*, Journal of Telecommunications and Information Technology vol. 2, ss. 13-20, 2016.

[**U10**] Kamola M., **Arabas P.**, Jaskóła P., Wiśniewski T., Niewiadomska-Szynkiewicz E., Malinowski K.,Karpowicz M., Sikora A., Marks M., Mincer M., Daniluk K.,*Econet – energooszczędne techniki dla przewodowych sieci komputerowych*, Przegląd Telekomunikacyjny – Wiadomości Telekomunikacyjne, nr 8-9, ss. 650-655, 2012.

[**U11**] **Arabas P.**, Karpowicz M., *Server Power Consumption: Measurements and Modeling with MSRs*, w: Challenges in Automation, Robotics and Measurement Techniques, vol. 440, ss. 233-244, 2016, DOI:10.1007/978-3-319-29357-8_21.

[U12] Kamola M., **Arabas P.**, *Wykorzystywanie uczenia ze wzmocnieniem do zadań ryzykownych i w sytuacjach niedostatku danych pomiarowych*, w: Bezpieczeństwo systemów cyberfizycznych i możliwości zastosowania sztucznej inteligencji, ss. 40-47, Główny Instytut Górnictwa – PIB, Katowice 2024.

Literatura

[1] IEEE 802.3az-2010 - IEEE standard for information technology. https://standards.ieee.org/ standard/802_3az-2010.html.

[2] Chipcon CC2420 Datasheet: http://focus.ti.com/lit/ds/symlink/cc2420.pdf, Texas Instruments, 2007.

[3] ETSI ES 203 237 v1.1.1 (2014-03) standard. www.etsi.org, 2014.

[4] M. Al-Fares, A. Loukissas, and A. Vahdat . A scalable, commodity data center network architecture. In *Proc. SIGCOMM 2008 Conference on Data Communications, Seattle, WA*, ss. 63–74, 2008.

[5] P. Arabas. Hierarchiczna struktura w systemie obrony przeciwrakietowej; mechanizmy decyzyjne i badania symulacyjne. praca doktorska, Politechnika Warszawska, 2004.

[6] V. Armant, M. De Cauwer, K. Brown, and B. O'Sullivan. Semi-online task assignment policies for workload consolidation in cloud computing systems. *Future Generation Computer Systems*, 82:89 – 103, 2018.

[7] A. Barabási and R. Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, Oct 1999.

[8] P. Baran. *On Distributed Communications: I. Introduction to Distributed Communications Networks*. RAND Corporation, Santa Monica, CA, 1964.

[9] A. Beloglazov, J. Abawajy, and R. Buyya. Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing. *Future Generation Computer Systems*, 28(5):755–768, 2012. Special Section: Energy efficiency in large-scale distributed systems.

[10] M. Benito, E. Vallejo, and R. Beivide. On the use of commodity ethernet technology in exascale hpc systems. In *Proc. IEEE 22nd International Conference on High Performance Computing (HiPC)*, ss. 254–263, 2015.

[11] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss. RFC2475: An architecture for Differentiated Services. https://datatracker.ietf.org/doc/html/rfc2475, 1998.

[12] R. Bolla, R. Bruschi, F. Davoli, L. Di Gregorio, P. Donadio, L. Fialho, M. Collier, A. Lombardo, D. Reforgiato Recupero, and T. Szemethy. The green abstraction layer: A standard power-management interface for next-generation network devices. *IEEE Internet Computing*, 17(2):82–86, March 2013.

[13] R. Bolla, R. Bruschi, F. Davoli, P. Lago, A. Bakay, R. Grosso, M. Kamola, M. Karpowicz, L. Koch, D. Levi, G. Parladori, and D. Suino. Large-scale validation and benchmarking of a network of power-conservative systems using etsi's green abstraction layer. *Transactions on Emerging Telecommunications Technologies*, 27(3):451–468, 2016.

[14] R. Bolla, R. Bruschi, and P. Lago. Energy adaptation in multi-core software routers. *Computer Networks*, 65:111–128, 2014.

[15] C. Borna. *Combating customer churn*, volume 34, ss. 83–85. 2000.

[16] J. Chabarek, J. Sommers, P. Barford, C. Estan, D. Tsiang, and S. Wright. Power awareness in network design and routing. In *IEEE INFOCOM 2008 - The 27th Conference on Computer Communications*, ss. 457–465, 2008.

[17] T. D., H. Jagode, H. You, and J. Dongarra. Collecting performance data with PAPI-C. In *Proc. 3rd Parallel Tools Workshop*, ss. 157–173, 2010.

[18] B. Dab, I. Fajjari, D. Belabed, and N. Aitsaadi. *Architectures of Data Center Networks: Overview*, ss. 1–27. 2021.

[19] P. Erdős and A. Rényi. On random graphs. I. *Publicationes Mathematicae*, 6(3-4):290–297, 1959.

[20] W. Findeisen, F. Bailey, M. Brdyś, K. Malinowski, P. Tatjewski, and A. Wozniak. *Control and coordination in hierarchical systems*. International series on applied systems analysis. Willey, 1980.

[21] W. Fisher, M. Suchara, and J. Rexford. Greening backbone networks: Reducing energy consumption by shutting off cables in bundled links. ss. 29–34, 08 2010.

[22] C. Gu, Z. Li, H. Huang, and X. Jia. Energy efficient scheduling of servers with multi-sleep modes for cloud data center. *IEEE Transactions on Cloud Computing*, 8(3):833–846, 2020.

[23] D. Hackenberg, T. Ilsche, J. Schuchart, R. Schöne, W. E. Nagel, M. Simon, and Y. Georgiou. Hdeem: High definition energy efficiency monitoring. In *2014 Energy Efficient Supercomputing Workshop*, ss. 1–10, 2014.

[24] D. Hackenberg, T. Ilsche, R. Schöne, D. Molka, M. Schmidt, and W. E. Nagel. Power measurement techniques on standard compute nodes: A quantitative comparison. In *2013 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*, ss. 194–204, 2013.

[25] K. Hee-Su and Y. Choong-Han. Determinants of subscriber churn and customer loyalty in the korean mobile telephony market. *Telecommunications Policy*, 28(9):751–765, 2004.

[26] Hewlett-Packard, Intel, Microsoft, Phoenix Technologies, and Toshiba. *Advanced Configuration and Power Interface Specification, Revision 5.0*. 2011.

[27] Intel. 64 and IA-32 Architectures Software Developer's Manual. http://www.intel.com/content/ dam/www/public/us/en/documents /manuals/64-ia-32-architectures-software-developer-manual-325462.pdf, 2015.

[28] Intel, Hewlett-Packard, NEC, and Dell. Intelligent Platform Management Interface Specification, Second Generation. https://www.intel.com/content/ www/us/en/servers/ipmi/ipmi-intelligent-platform-mgt-interface-spec-2nd-gen-v2-0-spec-update.html, 2015.

[29] F. Juarez, J. Ejarque, and R. Badia. Dynamic energy-aware scheduling for parallel task-based application in cloud computing. *Future Generation Computer Systems*, 78:257 – 271, 2018.

[30] J. Kim, J. Balfour, and W. Dally. Flattened butterfly topology for on-chip networks. In *40th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO 2007)*, ss. 172–182, 2007.

[31] W. Li, Q. Fan, W. Cui, F. Dang, X. Zhang, and C. Dai. Dynamic virtual machine consolidation algorithm based on balancing energy consumption and quality of service. *IEEE Access*, 10:80958–80975, 2022.

[32] K. Nichols, V. Jacobson, and L. Zhang. RFC2638: A two-bit differentiated services architecture for the internet. https://datatracker.ietf.org/doc/html/rfc2638.

[33] E. Niewiadomska-Szynkiewicz. Symulacja komputerowa w analizie i projektowaniu złożonych systemów sterowania. Prace Naukowe Politechniki Warszawskiej. Elektronika, z. 150, 2005.

[34] V. Pallipadi, S. Li, and A. Belay. cpuidle: Do nothing, efficiently. In *Proc. Linux Symposium*, volume 2, ss. 119–125, 2007.

[35] V. Pallipadi and A. Starikovskiy. The ondemand governor. In *Proc. Linux Symposium*, volume 2, ss. 215–230, 2006.

[36] J.-M. Pierson. *Large-scale Distributed Systems and Energy Efficiency: A Holistic View*. John Wiley & Sons, 2015.

[37] G. Politis, P. Sampatakos, and I. Venieris. Design of a multi-layer bandwidth broker architecture. *Lecture Notes in Computer Science*, 12 2000.

[38] A. Qureshi, R. Weber, H. Balakrishnan, J. Guttag, and M. B. Cutting the electric bill for internet-scale systems. In *SIGCOMM Comput. Commun.*, volume 39, ss. 123–134, 2009.

[39] E. Rotem, A. Naveh, A. Ananthakrishnan, E. Weissmann, and D. Rajwan. Power-management architecture of the intel microarchitecture code-named sandy bridge. *IEEE Micro*, 32(2):20–27, 2012.

[40] H. Tamura and T. Yoshikawa. *Large-scale systems control and decision making*. M. Dekker, 1990.

[41] M. Technologies. Power Saving Features in Mellanox Products. http://www.mellanox.com/related-docs/whitepapers/WP_ECONET.pdf, 2013.

[42] L. Valiant and G. Brebner. Universal schemes for parallel communication. In *Proc. of the Thirteenth Annual ACM Symposium on Theory of Computing (STOC'81)*, ss. 263—-277, 1981.

[43] A. Verma, P. Ahuja, and A. Neogi. pmapper: Power and migration cost aware application placement in virtualized systems. In *Proc. ACM/IFIP/USENIX 9th International Middleware Conference*, Middleware '08, ss. 243–264. Springer-Verlag, 2008.

[44] Y. Wei, Z. Zhang, D. Afanasiev, P. Thubert, and T. Przygienda. RIFT applicability (draft-ietf-rift-applicability-11). https://datatracker.ietf.org/doc/draft-ietf-rift-applicability/, 2023.

[45] G. M. Weiss. *Data mining in telecommunications*. Springer, 2005.

[46] Z.-L. Zhang, Z. Duan, and Y. Hou. On scalable design of bandwidth brokers. *IEICE Transactions on Communications*, E84-B, 08 2001.

Warszawa 30.09.2024