

## ENERGOOSZCZĘDNE TRASOWANIE RUCHU W SIECI IP Z OSPF

**Streszczenie:** W artykule rozważa się sieć IP ze standardowym protokołem trasowania ruchu po najkrótszych ścieżkach. Przedstawione są heurystyczne strategie wyboru zbioru łączy, które mają pozostać wyłączane tak, aby minimalizować bieżący pobór mocy przez sieć – zależny od łącznego ruchu w sieci i od liczby aktywnych łączy. Działanie algorytmów zostało przetestowane symulacyjnie dla dwu popularnych modeli sieci rzeczywistych.

### 1. WSTĘP

Potrzeba redukcji zużycia energii zarówno w sieciach przewodowych, jak i bezprzewodowych jest niekwestionowana – choć u jej podstaw leżą nieco inne przyczyny. Również mechanizmy oszczędzania energii różnią się w obu przypadkach. Podczas gdy technologie oszczędzania energii w sieciach bezprzewodowych (głównie mobilnych) rozwijały się z uwagi na ograniczone możliwości ich zasilania, sieci przewodowe pozostały z tyłu, bez należytej motywacji do oszczędzania energii.

Paradoksalnie, to właśnie sieci przewodowe dają najwięcej możliwości uzyskania oszczędności energii – choćby przez zmianę samego medium komunikacyjnego, co w przypadku łączności radiowej jest po prostu niewykonalne. Jednak nawet najoszczędniejsze media transmisyjne nie zwalniają routerów z obowiązku przetwarzania przekazywanych pakietów, co może być zadaniem bardzo energochłonnym, zważywszy obecną przepustowość interfejsów sieciowych. Jasne staje się, że po uzyskaniu oszczędności poprzez zmianę mediów komunikacyjnych i ściśle związanego z nimi sprzętu, kolejnych zysków energetycznych należy szukać w umiejętnym zarządzaniu ruchem, czyli takiej konfiguracji urządzeń sieciowych, która przynosi oszczędności przy niezminionej jakości usług (QoS – *quality of service*).

Pałaca potrzeba opracowania i wdrożenia energooszczędnych technologii w każdej warstwie sieci została wyrażona w [1], w odniesieniu do sieci każdego typu: dostępowych, metropolitalnych i szkieletowych. Kolejne publikacje [2,3] dokonują przeglądu aktualnych inicjatyw odnoszących się do tematu i ich osiągnięć. W dalszej części wywodu skupiamy się na energooszczędnych strategiach trasowania ruchu. Różnią się one wymaganiami wobec dostępności danych pomiarowych, jak i wobec możliwości technicznych sprzętu.

Jak zauważono w [2], moc zużywana przez dowolny podzespół urządzenia sieciowego może być przedstawiona jako funkcja jakiejś miary wykorzystania tego urządzenia – np. aktualnego obciążenia ruchem. W jednym skrajnym przypadku moc ta może być w ogóle niezależna od obciążenia (funkcja stała). W innym, bieżąco odległym przypadku moc ta może być ściśle liniowo zależna od aktualnego wykorzystania podzespołu. W praktyce zależność ta jest rosnąca i nieliniowa;

może być ona przybliżana funkcją wklęsłą lub wypukłą, choć zależność wklęsła jest w tym przypadku bardziej powszechna, naturalna. Dodatkowo, z uwagi na dyskretną naturę omawianych podzespołów, zależność ta również jest w praktyce zdyskretyzowana, często przyjmując przebieg schodkowy (o wypukłej lub wklęsłej obwiedni). Na domiar złego, tego rodzaju charakterystyki łączą się w wielowymiarowe funkcje opisujące zależność co bardziej złożonych podzespołów routera od jego aktualnego stanu. Zautomatyzowane zadanie wyszukania konfiguracji sieci (trasowania ruchu i wynikających zeń stanów energetycznych podzespołów i całych urządzeń) jest wówczas NP-trudnym problemem optymalizacji.

W praktyce tego typu skomplikowane zadania rozwiązuje się stosując relaksacje oraz inne przybliżenia i heurystyki. Przykładowo, w [4] przedstawiono sposób rozwiązania zadania energooszczędnej inżynierii ścieżek MPLS i doboru parametrów pracy przełączników poprzez zastosowanie metody podziału i oszacowań. W podejściu tym każdy, opisywany powyżej, schodek zależności traktowany jest jako jeden wirtualny „podzespół”, którego pobór mocy sam w sobie nie zależy od „obciążenia”. Wynikające z takiej dekompozycji zadanie kombinatoryczne jest stopniowo relaksowane do postaci mieszanej, aby, przy pomocy autorskiej heurystyki, zakwalifikować, które „podzespoły” powinny pozostać aktywne, aby sumaryczne zużycie energii przez sieć było jak najmniejsze. Podobnie, [5] wykorzystuje informację o położeniu schodków i w sposób heurystyczny przetrasowuje ruch tak, aby maksymalnie wypełniał „schodki”, jednocześnie unikając skokowego wzrostu poboru mocy.

Warto zauważyć, że oba przytoczone podejścia są w gruncie rzeczy bardzo wymagające w stosunku do ilości i jakości danych pomiarowych, jak i w stosunku do możliwości technicznych trasowania ruchu. Oba zakładają więc, że dostępna jest aktualna macierz zapotrzebowań na pasmo (pomiędzy każdą parą węzłów) oraz że możliwe jest przesyłanie ruchu po arbitralnie wybranej (niekoniecznie najkrótszej) ścieżce. Natomiast w praktyce estymacja zapotrzebowania (poprzez próbkowanie albo bardziej zaawansowane techniki) jest niezwykle niedokładna albo kosztowna; również transmisja ruchu po ścieżkach innych niż najkrótsze w sensie liczby przeskoków jest wymaganiem odbiegającym od rzeczywistości, w której wciąż używa się protokołu OSPF.

Z tego powodu warto pochylić się nad zdecydowanie mniej zaawansowanymi algorytmicznymi, ale bardziej przystającymi do rzeczywistości propozycjami algorytmów o skromniejszych wymaganiach, z zamiarem wprowadzenia i sprawdzenia własnych pomysłów. W pozostałej części pracy przedstawimy aktualnie proponowane podejścia do energooszczędnego trasowania, które zakładają kierowanie ruchu do przeznaczenia po najkrótszych ścieżkach. Umiejscowimy je wobec dwu

skrajnych możliwości reakcji podzespołów na przekazywany ruch. Następnie, w rozdz. 2 zostaną przedstawione własne propozycje heurystycznych procedur wyłączenia i włączania łączy w sieci, wobec zmieniającego się ruchu. Ich działanie dla standardowych przypadków testowych jest przedstawione w rozdz. 3, a wnioski i perspektywy dalszych prac – w rozdz. 4.

Jedno z typowych podejść do energooszczędnego trasowania ruchu w sieci z działającym trasowaniem OSPF zostało przedstawione w [6]; polega ono na nadaniu wybranym łączom tak dużej wagi, aby OSPF skutecznie je pominął podczas wyznaczania tras w sieci. Następnie łącza te są zupełnie wyłączane, co z reguły skutkuje o rząd wielkości mniejszym zużyciem energii, niż gdyby zostały one jedynie pozbawione wszelkiego ruchu. Oczywiście, problemem pozostaje, które łącza wytypować do zupełnego odstawienia tak, aby nie pogorszyć jakości usług w sieci. Zaproponowano dwie heurystyki: pierwsza wyznacza kliki węzłów, a następnie odstawia jak największą liczbę łączy, posiłkując się informacją o klikach. Druga polega na kolejnym odstawianiu najmniej obciążonych łączy. Testy symulacyjne wykazały zdecydowaną przewagę pierwszego podejścia. Ponadto, do wdrożenia wystarczy zaimplementowanie algorytmu tylko na jednym routerze, rozsyłającym aktualizacje wag łączy.

Innym interesującym podejściem jest algorytm GRiDA [7], postulujący umieszczenie na każdym routerze stosunkowo prostego oprogramowania agentowego. Agenty w sposób autonomiczny decydują o wyłączeniu łączy przynależnych do własnych routerów, wykorzystując uczenie maszynowe do opracowania własnych preferencji odstawiania i aktywacji łączy. Reguły buduje się w oparciu o reakcję sieci na odstawienie łącza: ewentualne przeciążenie skutkuje uznaniem danego łącza jako potencjalnie kluczowego i niewyłączaniem go w przyszłości. Agenty dysponują szczegółową wiedzą o naturze kontrolowanego przez nie sprzętu; w szczególności znają charakterystyki zużycia energii podzespołów. Wiedza ta jest wykorzystywana lokalnie; agenty przekazują sobie jedynie wartości aktualnego obciążenia poszczególnych interfejsów. W odróżnieniu od [6], nie następuje interakcja wbudowanego w router oprogramowania trasującego i algorytmu agentowego.

Kolejne interesujące podejście zaproponowano w [8]: podobnie jak w [6], w jednym z routerów uruchamia się specyficzny, scentralizowany algorytm heurystyczny, podejmujący decyzje na podstawie aktualnego obciążenia wszystkich łączy w całej sieci.

W zależności od charakterystyk energetycznych poszczególnych podzespołów, różnie kształtują się naturalne możliwości oszczędzania energii przez każdą sieć z trasowaniem wykorzystującym liczbę przeskoków jako metrykę. Jeśli rozważymy sieć bez ograniczeń na pasmo, wówczas możemy wyróżnić dwa skrajne scenariusze.

- *Każdy z podzespołów zużywa moc dokładnie proporcjonalnie do swojego obciążenia* (np. port ethernet zużywa energię proporcjonalnie do liczby przesłanych bajtów, bez żadnych kosztów stałych). W takim przypadku niewymagany jest jakikolwiek dodatkowy algorytm energooszczędnego trasowania, gdyż trasowanie wg najkrótszej ścieżki w sposób naturalny minimalizuje liczbę przeskoków (a zatem, i re-

transmisji pakietów). Wynika stąd, że sieci z zaawansowanymi, autonomicznymi mechanizmami oszczędzania energii na poziomie podzespołów (np. proponowane w [9]) praktycznie nie wymagają dodatkowych mechanizmów energooszczędnego trasowania poza godzinami szczytu.

- *Każdy z podzespołów pobiera stałą moc, niezależnie od własnego obciążenia.* W tej skrajnej sytuacji kierowanie ruchu po najkrótszych ścieżkach może powodować wyjątkowo nieefektywne wykorzystanie infrastruktury oraz duże straty energii. Rozważmy dla przykładu topologię pełnej siatki połączeń: wówczas OSPF wykorzysta wszystkie możliwe ścieżki, powodując największe możliwe zużycie energii. Natomiast w tym przypadku optymalną topologią byłoby minimalne drzewo rozpinające (po wyłączeniu wszystkich zbędnych łączy). Złożoność algorytmów je wyznaczających [10,11] jest wielomianowa. W takiej sieci algorytmy [6,7] również będą dążyć do wyłączenia tylu łączy, aby nie naruszyć spójności sieci – ale ponieważ czynią to w sposób heurystyczny, mogą doprowadzać do sytuacji patologicznych, np. szeregowego połączenia wszystkich węzłów.

W sytuacji zróżnicowanej efektywności energetycznej urządzeń lub aktywnych ograniczeń na przepustowość, trasowanie ruchu po najkrótszej ścieżce nie będzie skutkowało rozwiązaniem optymalnym dla żadnego z powyższych przypadków. Energooszczędne trasowanie wymaga wówczas dzielenia ruchu na poszczególnych relacjach i kierowania go po arbitralnie dobranych ścieżkach.

## 2. PROPONOWANE HEURYSTYKI

Rozważmy inne, podobne do zaprezentowanych już, możliwości minimalizacji mocy zużywanej przez sieć. Zakładamy, że w sieci działa OSPF jako podstawowy mechanizm trasowania ruchu po najkrótszej ścieżce do miejsca przeznaczenia. Wówczas jedynym środkiem oddziaływania na proces trasowania jest umiejętny wybór łączy, które mają zostać wyłączone. Proponujemy zastosowanie szeregu alternatywnych, heurystycznych strategii dezaktywacji łączy o różnym stopniu skomplikowania i wynikających zeń wymaganiach na dane zewnętrzne. Proponowane algorytmy wykazują podobieństwo do przedstawionych w [7], gdyż przechowują historię wykonanych operacji; jednakże działają one w sposób skoordynowany, analogicznie do samego OSPF. Nie korzystają też z lokalnych profili energetycznych podzespołów. Proponowane algorytmy wykazują również podobieństwo do przedstawionych w [6], ponieważ do działania wykorzystują pełną informację o topologii sieci, natomiast różnią się, ponieważ wykorzystują stan poszczególnych łączy jako wektor zmiennych decyzyjnych (zamiast zmiennych decyzyjnych opisujących przebieg całych ścieżek).

Oznaczmy przez  $N$  liczbę węzłów w sieci, przez  $y_l$  – natężenie ruchu na łączu  $l$ , przez  $w_l$  – wagę dla łącza  $l$  (wykorzystywaną przez OSPF), a przez  $A(w)$  – macierz trasowania dla wektora wag  $w$ . Liczba kolumn  $A$  odpowiada liczbie przepływów ( $N^2$ ), liczba wierszy – liczbie łączy. Jeśli przepływ  $j$  wykorzystuje łącze  $i$ , wówczas  $a_{ij}=1$ ; w przeciwnym przypadku – 0. Przyjmijmy, że

przepływy, posortowane najpierw wg źródła, a następnie przeznaczenia, tworzą wektor  $x$  tak, że przepływ z węzła  $i$  do  $j$  jest elementem  $x_{(i-1)N+j}$ . Wówczas  $y=Ax$ . Wektor  $w$  jest wektorem zmiennych decyzyjnych: ustawienie  $w_i=\infty$  powoduje ominięcie przez przepływy łącza  $i$  (które może być następnie fizycznie wyłączone).

Każdy z proponowanych tutaj algorytmów heurystycznych działa według jednolitego schematu, stale nadzorując ruch w sieci. Po wykryciu istotnej zmiany, podejmowane jest jedno z działań:

- Łączny ruch  $\sum_l y_l$  zmalał i podejmowana jest próba wyłączenia kolejnego łącza.
- Ruch na określonym łączu wzrósł i podejmowana jest próba załączenia kolejnego łącza.

Każdy z proponowanych algorytmów wykorzystuje trzy listy łącza:

$Y_{ON}$  łącza, które muszą być stale włączone, aby utrzymać spójność sieci (*links on*);

$Y_{OFF}$  łącza aktualnie wyłączone w celu zmniejszenia zużycia energii (*links off*);

$Y_{BY}$  łącza, które muszą pozostać włączone, aby uniknąć przeciążeń (*bypass links*).

W przypadku zaobserwowanego wzrostu natężenia ruchu, każdy z algorytmów podejmuje próby aktywacji kolejnych łącza, dopóki sieć nie osiągnie stanu akceptowalnego. Wyboru łącza dokonuje się spośród  $Y_{OFF}$ ; po ustawieniu wagi ruch powraca na to łącze, które następnie trafia na listę  $Y_{BY}$ . Proponuje się następujące strategie wyboru łącza do aktywacji:

RL Załącz łącze ostatnio wyłączone (*Rollback Last*). Postępując w ten sposób, ma się gwarancję powrotu do kolejnych konfiguracji sieci, lecz odzyskana konfiguracja może okazać się nieadekwatna do aktualnego ruchu, zatem nie gwarantuje szybkiej poprawy (choć ostatecznie zostaną załączone wszystkie wcześniej dezaktywowane łącza).

RT Spośród  $Y_{OFF}$  do załączenia wybierane jest łącze łączące 2 aktualnie najbardziej obciążone węzły (w sensie sumarycznego ruchu na podłączonych doń pozostałych łączach – stąd akronim *Relieve most Traversed endpoints*). W odróżnieniu od RL, strategia ta nie zależy od historii podjętych działań, a jedynie od bieżącego stanu algorytmu, w pełni definiowanego przez  $Y_{ON}$ ,  $Y_{OF}$  i  $Y_{BY}$ . Propozycja wynika z przeświadczenia, że ponowne połączenie mocno obciążonych węzłów gruntownie poprawi sytuację w sieci – skracając większość ścieżek i w szczególności odciążając aktualnie przeciążone łącza.

RB Spośród  $Y_{OFF}$  do załączenia wybierane jest łącze, którego aktywacja spowoduje największy spadek liczby przepływów korzystających z aktualnie najbardziej zatłoczonego łącza (ozn.  $l$ ). Strategia ta nie bierze pod uwagę faktycznych wartości tych przepływów, a jedynie ich liczbę (stąd akronim *Relieve Blind to traffic matrix*). Badane są hipotetyczne tablice trasowania, wynikające z załączenia poszczególnych łącza zawartych w  $Y_{BY}$ , w celu znalezienia łącza  $m^*$  minimalizującego  $\sum_k a_{l,k}$ . Łącze  $m^*$  może być postrzegane jako najskuteczniejsze obejście dla  $l$  ze względu na liczbę przekierowanych przepływów.

RM Spośród  $Y_{OFF}$  do załączenia wybierane jest łącze, którego aktywacja spowoduje największy spadek ruchu na łączu  $l$ . Strategia ta wymaga i posługuje się aktualną macierzą ruchu  $x$  (stąd akronim *Relieve by traffic Matrix*). Analogicznie do RB,  $m^* = \arg \min \sum_k a_{l,k} x_k$ . Łącze  $m^*$  może być postrzegane jako najskuteczniejsze obejście dla  $l$  ze względu na sumę przekierowanego ruchu.

W przypadku zaobserwowanego spadku natężenia ruchu, każdy z algorytmów podejmuje próby odstawienia kolejnych łącza, dopóki sieć nie osiągnie stanu akceptowalnego. Wyboru łącza dokonuje się spośród  $Y_{BY}$  (albo spośród ogółu łącza, w fazie inicjalizacji algorytmu); po ustawieniu wagi ruch rugowany jest z łącza, wówczas może ono zostać fizycznie wyłączone. Proponuje się następujące strategie wyboru łącza do dezaktywacji:

LLL Wyłącz łącze aktualnie najbardziej obciążone (*Least Loaded Link*).

LTL Wyłącz łącze wykorzystane przez najmniejszą liczbę przepływów (*Least Traversed Link*). Strategia ta jest zainspirowana analizą sieci społecznych: eliminowane jest powiązanie o najmniejszym *pośrednictwie* (*betweenness*), czyli takie, którego brak zostanie najmniej odczuty przez inne węzły. Informacja o aktualnej liczbie przepływów wykorzystujących poszczególne łącza może zostać uzyskana bezpośrednio z OSPF.

SIT Wyłącz łącze wykorzystywane przez najmniejsze przepływy (*Smallest Individual Traversing flows*). Strategia nie wykorzystuje faktycznej macierzy ruchu; zakłada, że wszystkie przepływy w łączu są jednakowe i równe ilorazowi liczby przepływów i aktualnego ruchu na łączu. U podstaw strategii leży nadzieja, że przekierowane przepływy rozprósza się równomiernie po wielu różnych łączach. Strategia ta rozmyślnie nie wykorzystuje macierzy ruchu (wykorzystuje ją LDM, omówiona dalej).

FFT Wyłącz łącze z najmniejszą liczbą przepływów, które kończą się lub zaczynają wykorzystując to łącze (*Fewest Flows Terminating on either node*). Strategia wynika z założenia, że pozostałe przepływy zaczynające się lub kończące w węzłach połączonych przez wytypowane łącze, w większości wykorzystują inne łącza alternatywne; zatem skutki rozłączenia wytypowanego łącza będą znikome.

LDB Wyłącz łącze, które spowoduje przekierowanie najmniejszej możliwej liczby przepływów aktualnie je wykorzystujących. Strategia ta nie bierze pod uwagę faktycznych wartości tych przepływów, a jedynie ich liczbę (stąd akronim *Least Diversion Blind to traffic matrix*). Koszt przekierowania definiowany jest jako przyrost długości wszystkich ścieżek w sieci (równej sumie elementów macierzy  $A$ ). Analogicznie do RB i RM, podjęcie decyzji wymaga przeanalizowania potencjalnych zmian macierzy trasowania dla każdego z łącza branego pod uwagę.

LDM Jak dla LDB, ale z uwzględnieniem faktycznej macierzy ruchu (*Least Diversion with traffic Matrix*). Dezaktywowane jest łącze  $m^* = \arg \min \|Ax\|_1$ , czyli wiążące się z najmniejszym możliwym wzrostem ruchu w sieci.

Zaproponowane powyżej strategie mają różne wymagania odnośnie danych potrzebnych do ich działania, począwszy od najskromniejszych (LLL+RL), bazujących wyłącznie na obciążeniu łączy i przeanalizowanych przez innych autorów, poprzez wykorzystujące macierz trasowania (LTL/SIT/LDB+RB), kończąc na najbardziej wymagających, posługujących się aktualną macierzą ruchu (LDM+RM). Ten ostatni przypadek nie był przeanalizowany w [6]; natomiast został uznany za nieprzystający do rzeczywistości w [7]. Możliwe są również inne kombinacje strategii włączania i wyłączania łączy, co skutkuje różnorodnością algorytmów. Oczywiście nie wszystkie kombinacje wydają się sensowne (np. nieposługiwanie się informacją o macierzy ruchu podczas włączania łączy, podczas kiedy wykorzystuje się ją do wyłączania), ale nie należy w ich ocenie polegać wyłącznie na intuicji.

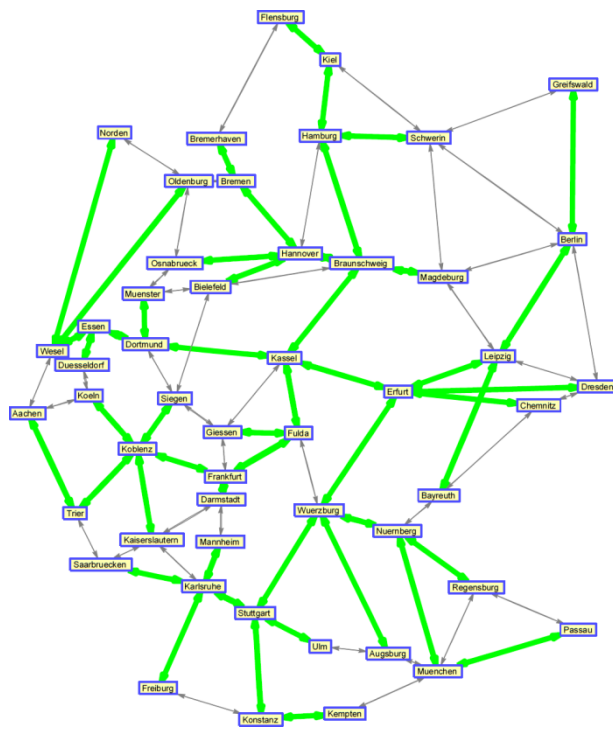
W zależności od rodzaju wykorzystywanych danych, implementacja algorytmów również będzie różnita; niemniej jednak ich ogólny schemat działania jest jednakowy: sterują one trasami ruchu wyznaczanymi poprzez OSPF pośrednio, poprzez dobór (pomiędzy wartościami 1 a  $\infty$ ) wag algorytmu. Analogicznie do działania OSPF, agenty na każdym z routerów działają równolegle, ale w sposób deterministyczny i używając takich samych danych (rozgłaszanych, w miarę potrzeby, między sobą). Dlatego proponowany system będzie działał stabilnie.

Jak już zauważono, algorytmy mają różne wymagania odnośnie danych wejściowych. W przypadku LLL+RL potrzebujemy wyłącznie informacji o obciążeniach wszystkich łączy. Podobnie jak inni autorzy, proponujemy, aby agenty zbierały te informacje lokalnie i wymieniały się nimi wykorzystując np. technologię rozszerzeń *opaque LSA* do protokołu OSPF. Uniknie się w ten sposób hazardów, gdyż informacje OSPF są rozgłaszane jednocześnie do wszystkich zainteresowanych routerów. Inna, konkretna wskazówka implementacyjna, odnosząca się do strategii sprawdzających szereg hipotetycznych rozwiązań (i tablic trasowania), polega na wykorzystaniu możliwości routerów i uruchomieniu na nich dodatkowych procesów OSPF, sprawdzających jedynie hipotezy zamiast faktycznie sterować siecią (np. w przypadku implementacji RB czy LDM). Kolejna uwaga implementacyjna odnosi się do praktycznego wykorzystania macierzy ruchu: jej mierzenie jest niezwykle kosztowne, a estymacja – niestety, problematyczna (ruch sieciowy gwałci większość założeń czynionych przez rozmaite algorytmy estymacji macierzy ruchu – por. [12,13]). Niemniej jednak strategie wykorzystujące macierz ruchu zostały uwzględnione w prezentowanych badaniach.

### 3. WYNIKI

Algorytmy implementujące rozmaite kombinacje strategii aktywacji i dezaktywacji łączy zostały sprawdzone symulacyjnie dla dwu zagadnień optymalizacji sieci szkieletowych, dostępnych w uznanym przez środowisko naukowe repozytorium problemów telekomunikacyjnych [14]. Wybrane zagadnienia definiują topologię sieci, przepustowość połączeń oraz wartości zapotrzebowania na pasmo. Jako że wybrane dwa zagadnienia nie doczekały się jeszcze rozwiązań (tj. dopuszczalnych ścieżek trasowania), podczas testów postanowiono zliberalizować pierwotne ograniczenia zasobowe.

Żadne z wybranych zagadnień nie definiuje natomiast charakterystyk energetycznych elementów sieci. Na własny użytek zakładamy, że zużycie mocy przez pojedyncze łącze jest liniową funkcją ruchu, jaki przez nie przechodzi, plus pewna wartość stała dla wszystkich aktualnie aktywnych łączy. Zakładamy również, że łącza są jednorodne pod względem charakterystyk energetycznych – zdając sobie sprawę, że stanowi to być może zbyt duże uproszczenie. Z drugiej jednak strony, ułatwia ono interpretację wyników. Odmienne niż większość autorów, postanowiliśmy nie zakładać konkretnej relacji pomiędzy mocami i konsekwentnie analizować problem jako dwukryterialny: minimalizacji sumarycznego przepływu (a zarazem kosztów realizacji przepływów) przy jednoczesnej maksymalizacji liczby odstawionych łączy (a zarazem oszczędności w działaniu sieci).

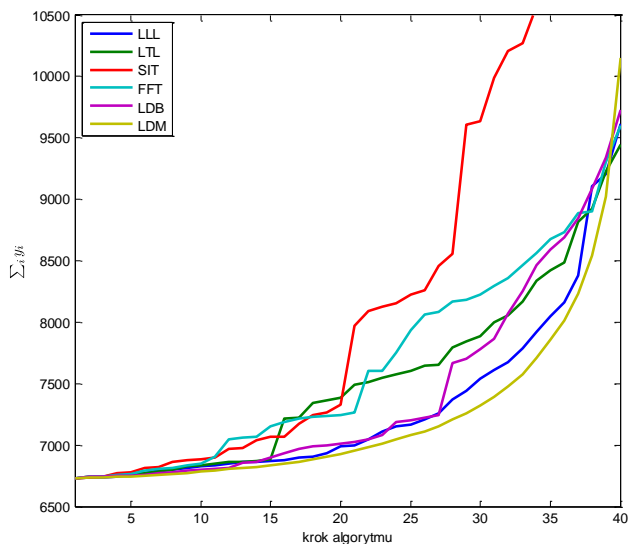


Rys. 1. Sieć testowa z przykładowym drzewem rozpoczynającym (w kolorze zielonym).

Wstępne testy obejmowały porównanie wyłącznie strategii wyłączania łączy. Do testów wybrano problem *germany50-D-B-L-N-C-A-N-N* (50 węzłów, z ograniczeniami zasobowymi, które w tej fazie zostały pominięte). Dzięki pominięciu ograniczeń, algorytm

zawsze zredukuje graf aktywnych łączy do postaci drzewa rozpinającego, tak jak pokazano na rys. 1. Jest do docelowa postać grafu, po wyłączeniu 39 łączy. Sumaryczne, końcowe obciążenie łączy jest w tym przypadku najniższe dla stosunkowo niewymagającej strategii – LTL.

Na rys. 2 przedstawiono sumaryczny ruch w sieci w kolejnych krokach strategii wyłączania łączy. Można go utożsamiać z całkowitym poborem energii, związanym z przekazywaniem ruchu przez łączy. Natomiast kolejne kroki algorytmów wiążą się z kolejno wyłączanymi łączy; można je utożsamiać z oszczędnościami w związku z fizycznym odłączaniem kolejnych łączy. Dopiero uwzględnienie faktycznych relacji pomiędzy zależnym od ruchu kosztem transmisji a stałym kosztem włączenia łączy pozwoliłoby na znalezienie optymalnego energetycznie stanu sieci (na rys. 2 należałoby nałożyć poziomice funkcji kosztu). Niezależnie od tych relacji, strategia LDM ewidentnie dominuje wszystkie pozostałe, co wydaje się być naturalne, gdyż wykorzystuje ona najbardziej szczegółowe i dokładne informacje o ruchu. Niemniej jednak w końcowej fazie (gdy struktura grafu zbliża się do drzewiastej), LDM powoduje znaczący, w stosunku do innych strategii, wzrost ruchu łącznego.



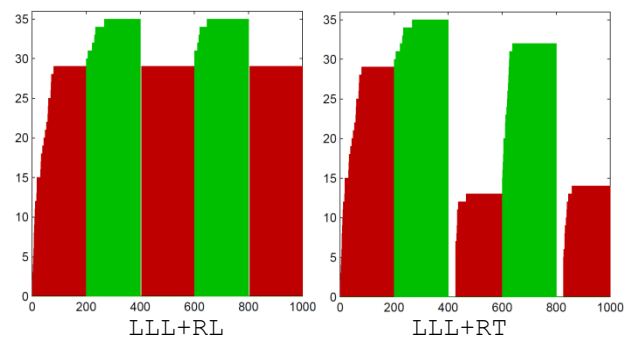
Rys. 2. Całkowity ruch w sieci przy kolejno wyłączanych łączy

Całkiem dobrą alternatywą dla LDM wydaje się strategia LDB (operująca liczbą przekierowanych przepływów, zamiast ich sumy) oraz LLL – najprostsza z możliwych. Również LTL daje początkowo dobre rezultaty. Powyższe wyniki, uzyskane dla konkretnego scenariusza zapotrzebowania, zweryfikowano dla szeregu losowych macierzy ruchu. W tab. 1 podano, o ile gorzej jest rozwiązanie uzyskane przez wybrane strategię w 20. kroku algorytmu, względem aktualnie najlepiej działającej strategii. Według takiego kryterium, LDB i LDM okazują się najlepszymi algorytmami (różnice są pomijalnie małe), z LLL tuż za nimi. Pozostałe strategię wypadają nieznacznie gorzej, lecz wciąż są tylko o znikomy ułamek gorzej.

Tab. 1. Porównanie strategii wyłączania w ujęciu statystycznym

Strategia	LLL	LTL	SIT	FFT	LDB	LDM
Pogorszenie [%]	0,9	1,2	4,9	2,3	0	0
Odchylenie stand.	0,3	0,1	1,1	0,1	0	0

Przedstawione strategię wyłączania stanowią jedynie połowę każdego algorytmu trasowania; drugą jego połową są strategię aktywacji łączy. Do konstrukcji konkretnych algorytmów zakwalifikowano LLL, LDB i LDM (jako punkt odniesienia), oraz ich odpowiedniki RL, RB i RM (w sensie złożoności decyzyjnej i zapotrzebowania na dane). Porównano więc algorytmy: LLL+RL, LDB+RB oraz LDM+RM w scenariuszu, w którym ruch waha się, a ograniczenia zasobowe są istotne. W porównaniu również uwzględniono kombinację LLL+RT, aby sprawdzić, jaki jest wpływ zastosowania ciągle dość prostej strategii aktywacji, ale już niezależnej od sekwencji wykonanych wcześniej ruchów, w odróżnieniu od RL. Ponieważ pierwotny [14] problem routingu jest jeszcze nierozwiązany, zdecydowano się przyjąć pojemność każdego z łączy równą 220, tj. nieco powyżej obserwowanej największej zajętości pojedynczego łączy w sytuacji zwykłego trasowania ruchu w pełnej sieci – wynoszącej 216. Daje nam to gwarancję dopuszczalności rozwiązania początkowego. Takie ustawienia wstępne reprezentują sytuację w sieci w godzinach szczytu; poza szczytem ruch w naszym scenariuszu maleje skokowo o 12 procent.



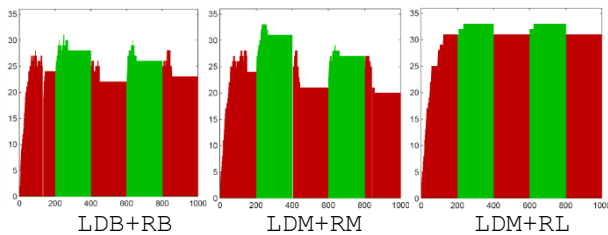
Rys. 3. Liczba wyłączonych łączy w kolejnych krokach symulacji algorytmów LLL+RL oraz LLL+RT

Dokonano symulacji działania algorytmów w scenariuszu z trzema okresami wysokiego obciążenia, rozdzielonych okresami niskiego obciążenia. Liczbę wyłączonych łączy w kolejnych krokach symulacji dla algorytmów LLL+RL oraz LLL+RT przedstawiono na rys. 3 (kolor czerwony odpowiada wysokiemu obciążeniu). Algorytm LLL+RL działa przewidywalnie, efektywnie i stabilnie; natomiast taka sama strategię wyłączania wraz ze strategię załączania RT, nieuwzględniając ściśle historii wcześniejszych operacji, skutkuje skrajnie nieefektywnym działaniem. Wynika to stąd, że RT, próbując w fazie niskiego zapotrzebowania wyłączać kolejne łączy, sporadycznie rozłącza graf i natychmiast zaznacza takie łączy jako niemożliwe do rozłączenia w przyszłości ( $Y_{ON}$ ). W kolejnych fazach sytuacja zmienia się – np. w godzinach szczytu aktywowane są przez LLL dodatkowe łączy, aby nie dopuścić do przeciążenia;



wówczas łącza wcześniej umieszczone w  $Y_{ON}$  powinny zostać zweryfikowane (a nie są).

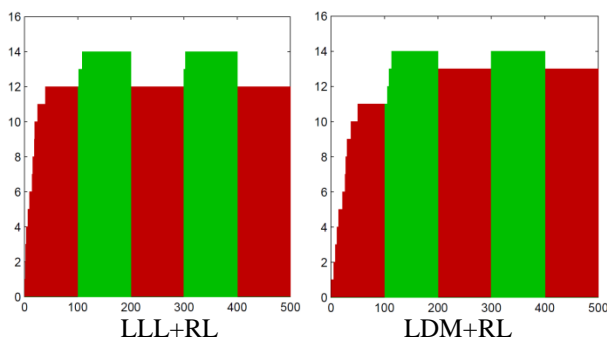
Również bardziej zaawansowane strategie rozczarowują swoją niestabilnością i nieskutecznością (por. rys. 4). LDB+RB zachowuje się zdecydowanie gorzej, niż we wstępnym rankingu: przy założonym ruchu wyłącza zdecydowanie mało łączy, a co gorsza, osiąga rozwiązanie w sposób chwiejny, niestabilny. Najwyraźniej założenie, że wszystkie przepływy mają jednakową wartość (co jest nieprawdą, a jednocześnie fundamentem działania LDB+RB) jest dla algorytmu zgubne.



Rys. 4. Liczba dezaktywowanych łączy w kolejnych krokach symulacji pozostałych badanych algorytmów

Symulacja najbardziej zaawansowanego z wybranych algorytmów, (rys. 4, pośrodku) ukazuje ponownie mankamenty zaproponowanych strategii załączania łączy: wszystkie z wyjątkiem RL traktują  $Y_{OFF}$  jako zbiór, a nie listę. Wskutek tego wszystkie aktualnie nieaktywne łącza są traktowane jako kandydaci do włączenia na równych prawach. W szczególności zdarza się, że włączane jest łącze, które zostało niegdyś wyłączone jako jedno z pierwszych, determinując w ten sposób całą dalszą sekwencję i obecny stan sieci. Ponowne załączenie tego łącza, wyrwane z tak zdefiniowanego kontekstu, powoduje destabilizację działania algorytmu. Co więcej, skutki niekonsekwencji nawarstwiają się w miarę postępów symulacji, wyraźnie objawiając się już w kolejnym okresie zwiększonego zapotrzebowania na ruch (kroki 400–600). Zastąpienie strategii RM przez prostą RL w tym przypadku znacząco poprawia stabilność algorytmu (por. rys. 4, z prawej).

Powyższe wnioski odniesiono do sieci o innej strukturze, złożonej z 22 węzłów i 36 łączy (model sieci GÉANT, [14], por. rys. 5). Zmodyfikowany algorytm wykorzystujący pełną informację o zapotrzebowaniu na ruch działa konsekwentnie nieco lepiej od najprostszych strategii, LLL+RL.



Rys. 5. Model sieci GÉANT, symulacja działania algorytmów LLL+RL oraz LDM+RL

#### 4. PODSUMOWANIE

Okazuje się, że implementacja energooszczędnego trasowania w sieci z OSPF poprzez zastosowanie heurystyk wykorzystujących szczegółowe dane o aktualnym ruchu nie daje zdecydowanie lepszych rezultatów niż heurystyki najprostsze. Najprostsza heurystyka, „wyłącz łącze najmniej obciążone/włącz łącze ostatnio wyłączone” działa stabilnie i ma znikome wymagania dotyczące danych. Strategia taka zakłada jednak wzrosty i spadki ruchu w sieci według takiego samego, symetrycznego scenariusza. W rzeczywistości macierz ruchu w sieci ewoluuje w ciągu doby, zatem pożądane jest poszukiwanie strategii nieuwzględniającej historii już podjętych decyzji.

Wiele proponowanych obecnie algorytmów energooszczędnej inżynierii ruchu wykorzystuje deklarowane przez producentów sprzętu profile energetyczne podzespołów, których połączenie skutkuje zadaniami optymalizacji o dużym stopniu złożoności. Modele te nie wykorzystują jednak informacji o faktycznym zużyciu energii przez sprzęt, które powinno stanowić podstawowe kryterium zatrzymania bądź przekonfigurowania algorytmu. W odniesieniu do zbadanych tutaj heurystyk można zaproponować, aby w procesie wyłączania łączy zaobserwowany faktyczny wzrost zużycia energii potraktować na równi z przekroczeniem ograniczeń zasobowych (przeciążeniem łączy), zaprzestając tym samym dalszych działań polegających na wyłączaniu bądź załączaniu kolejnych łączy.

#### SPIS LITERATURY

- [1] R. Bolla, R. Bruschi, A. Cianfrani, M. Listanti, „Enabling backbone networks to sleep”, *Network*, IEEE, 25(2), s. 26–31, 2011.
- [2] A. P. Bianzino, C. Chaudet, D. Rossi, J.-L. Rougier, „A survey of green networking research”, *Communications Surveys & Tutorials*, IEEE, 14(1), s. 3–20, 2012.
- [3] Wang, Jiayuan et al., „Energy-aware routing optimization in dynamic GMPLS controlled optical networks”, *14th International Conference on Transparent Optical Networks (ICTON)*, 2012.
- [4] E. Niewiadomska-Szynkiewicz, A. Sikora, P. Arabas, J. Kolodziej, „Control system for reducing energy consumption in backbone computer network”, *Concurrency and Computation: Practice and Experience*, 2012. <http://dx.doi.org/10.1002/cpe.2964>
- [5] N. Vasić, D. Kostić, „Energy-aware traffic engineering”, *Proceedings of the 1st International Conference on Energy-Efficient Computing and Networking*, ACM, 2010, s. 169–178.
- [6] A. Cianfrani, V. Eramo, M. Listanti, and M. Polverini, „An OSPF enhancement for energy saving in IP networks”, *Computer Communications Workshops (INFOCOM)*, IEEE, 2011, s. 325–330.
- [7] A. P. Bianzino, L. Chiaraviglio, M. Mellia, and J.-L. Rougier, „GRiDA: Green distributed algorithm for energy-efficient IP backbone networks”, *Computer Networks*, 2012.

- [8] Ho, Kin-Hon, Chi-Chung Cheung. “Green distributed routing protocol for sleep coordination in wired core networks”, *6th International Conference on Networked Computing (INC)*, 2010.
- [9] M. Karpowicz, “On the design of energy-efficient service rate control mechanisms: CPU frequency control for Linux”, *Proceedings of TIWDC 2013*.
- [10] V. Jarník, “O jistém problému minimálním”, *Práce Moravské Přírodovědecké Společnosti*, 6, s. 57–63, 1930.
- [11] J. B. Kruskal, “On the shortest spanning subtree of a graph and the traveling salesman problem”, *Proceedings of the American Mathematical Society*, 7(1), s. 48–50, 1956.
- [12] A. Lakhina, K. Papagiannaki, M. Crovella, C. Diot, E. D. Kolaczyk, N. Taft, “Structural analysis of network traffic flows”, *Proceedings of the joint international conference on Measurement and modeling of computer systems (SIGMETRICS '04)*, 2004, s. 61–72,  
<http://doi.acm.org/10.1145/1005686.1005697>
- [13] I. Juva, “Sensitivity of traffic matrix estimation techniques to their underlying assumptions”, *IEEE International Conference on Communications*, s. 562–568, 2007.
- [14] Sndlib, <http://sndlib.zib.de>