

# Wirtualizacja zasobów oraz alokacja przepływów w sieciach sterowanych programowo przy jednoczesnym zapewnieniu wymagań QoS

Chodacki, Maksymilian `maksymilian.chodacki@gmail.com`

Grzanka, Antoni `antoni.grzanka@gmail.com`

Leniart, Eryk `eryk.leniart@gmail.com`

Niedziałkowski, Adam `adam.niedzialkowski@gmail.com`

17 Stycznia 2017

## 1 Wstęp

Sieci sterowane programowo (SDN, z ang. Software Defined Networks) to koncept który coraz bardziej zyskuje na wadze w dzisiejszym świecie architektury systemów i sieci. Koncepcja oddzielenia warstwy kontrolującej sieć od mechanizmów związanych z transmisją danych rozwiązuje wiele problemów przed którymi stają codziennie architekci sieci. Jednym z wyzwań przed którym stają operatorzy chcący zaimplementować w swojej sieci mechanizm SDN jest sprawiedliwy podział zasobów pomiędzy swoich klientów (dzierżawców) oraz zapewnienie takiej alokacji przepływów, która zapewni płynne działanie sieci. Jest to krytycznie ważna kwestia ponieważ w rzeczywistych zastosowaniach, każdy z klientów (dzierżawców) sieci ma swoje wymagania odnośnie świadczonych przez operatora usług, których nie spełnienie może wiązać się z poważnymi konsekwencjami finansowymi. Wśród najważniejszych z takich parametrów można wymienić takie jak jitter, opóźnienie przesyłu, packet loss czy [cos tam jeszcze].

W pracy [1] autorzy proponują kompleksowe rozwiązanie powyższego problemu, obejmujące dwustopniowy model optymalizacyjny (wirtualizacja zasobów poprzez podział sieci oraz maksymalizacja minimalnego przepływu).

W poniższej pracy autorzy skupili się na optymalizacji sposobu alokacji przepływów względem innych parametrów niż te zaproponowane w [1], co może bardziej realistycznie odpowiadać zapotrzebowaniom klientów. W tym celu przeanalizowano algorytm zaproponowany w [1], dokonano jego implementacji w języku ILOG a także rozszerzono go, proponując dwa alternatywne modele ostatniej fazy działania algorytmu.

## 2 Podział zasobów

Pierwszą częścią algorytmu zaproponowanego w [1] jest podział sieci pomiędzy dzierżawców (klientów) tak, aby możliwie ich od siebie odizolować. Dopiero na tak przygotowanej, podzielonej sieci dokonuje się alokacji przepływów oraz sprawdzenia wymagań dotyczących parametrów Quality of Service.

Celem optymalizacji w pierwszym modelu jest maksymalna izolacja infrastruktury wykorzystywanej przez różnych dzierżawców, co można osiągnąć np. przez minimalizację łącz współdzielonych przez różnych klientów:

$$\min \sum_{t_1 \in T} \sum_{t_2 \in T} \sum_{a \in A} x_{at_1} x_{at_2} \quad (1)$$

W celu zapewnienia łączności każdego z węzłów z dowolnym innym dodano ograniczenia powodujące zbudowanie drzewa rozpinającego na grafie:

$$\forall t \in T \sum_{a \in A} x_{at} \geq N - 1 \quad (2)$$

$$\forall t \in T \forall Z \in V: Z \subset V, Z \neq \emptyset \sum_{a \in S_n} x_{at} \geq 1 \quad (3)$$

gdzie  $Z$  to zbiór podgrafów  $V$  a  $S_n$  to zbiór wszystkich łącz mających swój początek w  $Z$  a koniec w  $V \setminus Z$ . Powyższe ujęcie problemu zbudowania drzewa rozpinającego znane jest jako *cutset formulation* i zapewnia, że w każdym zbudowanym drzewie wszystkie węzły będą miały łączność z dowolnym innym.

Z uwagi na to, że implementacja wymagała wykorzystania dwóch łącz dla każdego połączenia, należało dodać również ograniczenie zabraniające jakiemukolwiek dzierżawcy korzystanie z łącza tylko w jedną stronę:

$$\forall t \in T \forall Z \in V: Z \subset V, Z \neq \emptyset \forall a \in S_n x_{at} x_{a't} = 0 \quad (4)$$

gdzie  $x_{a't}$  oznacza łącze komplementarne do  $x_{at}$  łączące te same węzły w drugą stronę, tj  $\forall a \in A: a = (i, j) \exists a' \in A: a' = (j, i)$ .

## 3 Model alokacji przepływu

Naszym kolejnym krokiem było zaimplementowanie modelu alokacji przepływów, w którym funkcją celu jest maksymalizacja minimalnego przepływu, czyli zrównoważenie ruchu generowanego przez przepływy dzierżawców na wszystkie dostępne łącza:

$$\sum_{n \in N} \max_{\{X_{f,i,j}^n; \forall (i,j) \in E_n^f, f \in F^n\}} \min_{f \in F^n} \lambda_f^n \quad (5)$$

W modelu zostało zaimplementowanych pięć ograniczeń. Pierwsze z nich zapewnia, że suma wartości przepływów na danym łączu nie przekroczy jego pojemności:

$$\sum_{n \in N} \sum_{f \in F^n} X_{f,i,j}^n \leq c_{ij} \quad (6)$$

Kolene ograniczenie nazywane konserwacją przepływów zapewnia, że dla danego węzła nie należącego do węzła docelowego dla dowolnego przepływu suma wartości wychodzących przepływów jest równa sumie wartości przychodzących przepływów plus wartość danego przepływu jeśli jest to węzeł źródłowy dla tego przepływu. W innym wypadku suma ta równa się zero:

$$\begin{aligned} \sum_{j; (i,j) \in E_n^f} X_{f,i,j}^n - \sum_{j; (j,i) \in E_n^f} X_{f,i,j}^n &= \lambda_f^n \mathbb{1}_{\{i=s_f^n\}} \\ \forall i \neq d_f^n, f \in F^n, n \in N \end{aligned} \quad (7)$$

Następne ograniczenie zapewnia, że przepływ nie może zostać rozdzielony na dwa lub więcej łączy:

$$X_{f,i,j}^n = 0 \quad \forall (i,j) \notin E_n^f, f \in F^n, n \in N \quad (8)$$

Czwarte ograniczenie związane jest z zapewnieniem jakości obsługi. Zapewnia ono, że iloczyn straty pakietów na łączach przypisanych do danego przepływu nie przekroczy ustalonej maksymalnej wartości:

$$\prod_{(i,j) \in E_n^f} p_{f,i,j}^n < p_{n,f}^{max} \quad (9)$$

Ostatnie ograniczenie nie jest opisane przez autorów, zostało dodane przez nas na potrzeby naszej struktury danych wejściowych a dokładnie podzielonego grafu pomiędzy dzierżawców. Ograniczenie to zapewnia, że dla danego dzierżawcy, każdy jego przepływ nie będzie wykorzystywał łączy, które nie są jego:

$$\forall f \in F^n X_{f,i,j} = 0 \quad \text{jeśli} \quad x_{i,j}^f = 0 \quad (10)$$

## 4 Główny skrypt

//TODO: Eryk

## 5 testy

## 6 Rozszerzenie

Po zaimplementowaniu podstawowego modelu postanowiliśmy podejść do tematu rozszerzenia problemu na dwa sposoby: dodaniu nowych ograniczeń oraz zastosowaniu go w innej formie. Zaproponowanym przez nas ograniczeniem jest zapewnianie minimalnej wartości przepływu, a nowym zastosowaniem jest wprowadzenie kosztu jako głównej metryki optymalizacji sieci.

## 6.1 Zapewnienie minimalnych wartości przepływu

Istotnym aspektem nie poruszonym przez autorów pracy jest zapewnienie minimalnej wartości przepływu. Autorzy odnieśli się do problemu zapewnienia wszystkim przepływom dodatniej wartości (w praktyce wszystkie przepływy będą miały równą wartość). Pozwala to na sprawiedliwy podział zasobów, natomiast w przypadku jeżeli różnym przepływom chcemy zapewnić różne minimalne wartości potrzebne jest rozszerzenie problemu o nowe dane (minimalną liczbę danych dla każdego przepływu) oraz dodatkowe ograniczenie:

$$\forall_{t \in T} \forall_{f \in F_t} \quad \lambda_{tf} \geq f.minimal \quad (11)$$

## 6.2 Koszt przesyłu danych

W przypadku nowego zastosowania postanowiliśmy postawić na praktyczne podejście; rozszerzyliśmy model o koszt przesyłanych danych i zmieniliśmy funkcję celu tak, by zrównoważyć koszt przepływów:

$$\sum_{t \in T} \max_{f \in F_t} \sum_{a \in A_f} a.cost \quad (12)$$

## 6.3 Wyniki rozszerzonych modeli

Poniżej przedstawiamy porównanie wyników modeli rozszerzonych i podstawowego wyliczonych w środowisku CPLEX. Tabela zawiera wyniki dla trzech zestawów danych: sieci małej (4 hosty), średniej (6 hostów) i dużej (17 hostów).

# 7 Podsumowanie

## Literatura

- [1] Lin, Shih-Chun and Wang, Pu and Luo, Min, *Jointly Optimized QoS-aware Virtualization and Routing in Software Defined Networks*, Comput. Netw. Vol. 96, Elsevier North-Holland, Inc. February 2016.