

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 osoby chcące zaimplementować w swojej sieci mechanizm SDN jest kontrola parametrów wśród . 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. W pracy [1] autorzy proponują kompleksowe rozwiązanie tego problemu.

2 Podział zasobów

Pierwszą częścią algorytmu zaproponowanego przez Lin i in. 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.

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:

$$\sum_{n \in N} \max_{\{X_{f,i,j}^n; \forall (i,j) \in E_n^f, f \in F^n\}} \min_{f \in F^n} \lambda_f^n \quad (1)$$

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

$$\sum_{n \in N} \sum_{f \in F^n} X_{f,i,j}^n \leq c_{ij} \quad (2)$$

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:

$$\sum_{j:(i,j) \in E_n^f} X_{f,i,j}^n - \sum_{j:(j,i) \in E_n^f} X_{f,i,j}^n = \lambda_f^n \mathbb{1}_{\{i=s_f^n\}} \quad (3)$$

$$\forall i \neq d_f^n, f \in F^n, n \in N$$

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

$$X_{f,i,j}^n = 0 \quad \forall (i,j) \notin E_n^f, f \in F^n, n \in N \quad (4)$$

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:

$$\prod_{(i,j) \in E_n^f} p_{f,i,j}^n < p_{n,f}^{max} \quad (5)$$

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 łącz, które nie są jego:

$$\forall f \in F^n X_{f,i,j} = 0 \quad \text{jeśli} \quad x_{i,j}^f = 0 \quad (6)$$

4 Główny skrypt

//TODO: Eryk

5 testy

6 Rozszerzenie

Po zaimplementowaniu podstawowego modelu postanowiliśmy podejść do tematu rozszerzenia podstawowego 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 decydowania o wykonalności problemu.

6.1 Zapewnienie minimalnych wartości przepływu

Istotnym aspektem nie poruszonym przez autorów pracy jest zapewnienie minimalnej wartości przepływu. Choć autorzy odnieśli się do problemu "zagłódzenia" ruchu poprzez maksymalizację minimalnego przepływu, czyli w praktyce zrównoważeniu podziału zasobów pomiędzy przepływy. Natomiast w przypadku jeżeli różnym przepływom chcemy zapewnić różne minimalne wartości potrzebne jest rozszerzenie problemu o nowe dane (minimalną liczbę danych per przepływ) oraz dodatkowe ograniczenie:

$$\forall t \in T \forall f \in F_t \quad \lambda_{tf} \geq f.minimal \quad (7)$$

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:

$$\sum_{t \in T} \max_{f \in F_t} \sum_{a \in A_f} a.cost \quad (8)$$

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.