****

**AKADEMIA GÓRNICZO-HUTNICZA IM. STANISŁAWA STASZICA W KRAKOWIE**

**WYDZIAŁ ELEKTROTECHNIKI, AUTOMATYKI,**

**INFORMATYKI I INŻYNIERII BIOMEDYCZNEJ**KATEDRA INFORMATYKI STOSOWANEJ

Praca dyplomowa magisterska

Protokoły OpenFlow i OVSDB w sieciach programowalnych SDN.  
  
OpenFlow and OVSDB in programmable Software Defined Networks.

Autor: Marek Ryznar

Kierunek studiów: Informatyka

Opiekun pracy: dr Michał Turek

Uprzedzony o …..

Strona na serdecznie dziękuje itd

**Spis Treści:**

1. **Wprowadzenie**
   1. **Cele pracy**
   2. **Zawartość pracy**
2. **Wstęp teoretyczny**
   1. **SDN**
   2. **OVS**
   3. **OpenFlow**
   4. **OVSDB**
   5. **MEF**
3. **Rozwinięcie**
   1. **Elementy MEF jako kryterium porówniania (do zmienienia)**
   2. **Różnice między protokołami wg dokumentacji**
   3. **Tu dalej sie zobaczy**
4. **Porównanie wyników**
5. **Wnioski**
6. **Podsumowanie**
7. **Wprowadzenie**

Współcześnie pojęcie informatyka jest tak dużym zagadnieniem, że trudno znaleść jedną osobę, która mogła by się nazwać specjalistą od wszystkich jej tematów. W związku z tym dzieli się ona na dziedziny, skupiające się wokół konkretnych problemów. Dziedzina w której będziemy się obracali w tej pracy to sieci komputerowe, czyli jedna z kwestii, która musiała być załatwiona w procesie digitalizacji, który nas doprowadził do ery komputerów w jakiej dzisiaj żyjemy.

Jednym z najważniejszych standardów zdefiniowanych w specjalizacji sieci komputerowych jest OSI/ISO RM (*ISO Open Systems Interconnection Reference Model*) czyli siedmiowarstwowy model odniesienia dla większości protokołów komunikacyjnych. Warstwy druga (*łącza danych*) i trzecia (*sieciowa*) są to miejsca w których *znajdują się* przełączniki i routery. Zestawienie i utrzymanie sieci przy użyciu wspomnianych urządzeń nigdy nie było prostym zadaniem.

Z pomocą w wykoniu tego celu wkroczyła koncepcja sieci programowalnych czyli SDN (*Software Defined Network*). Odbiorca niniejszej pracy zostanie zapoznany z ich zastowaniem oraz z głównymi protokołami komunikacyjnymi używanymi w ich działaniu.

* 1. **Cele pracy**

Celem pracy jest porówanie dwóch głównych protokołów komunikacyjnych, czyli OVSDB (*Open vSwitch Database Management Protocol*) i OpenFlow działających w programowalnych sieciach komputerowych. Kryterium porównania są elementy składowe usług zgodnych ze standardem Carrier Ethernet 2.0 definiowanych przez organizację MEF (*Metro Ethernet Forum).*

Na samym początku protokoły zostaną porównane na podstawie ich specyfikacji. Nestępnie poostawiona zostanie wirtualna sieć komputerowa typu SDN pozwalacjąca na zestawienie możliwości protokołów w praktyce. Wyniki symulacji zostaną przedstawione za pomocą dedykowanej do tego aplikacji.

* 1. **Zawartość pracy**

TODO

1. **Wstęp teoretyczny**
   1. **SDN**

Urządzenia sieciowe są od wielu lat instalowane i zarządzane w wielu firmach dostarczających usługi sieciowe. Mostki, przełączniki i rutery od zawsze były używane w wielu środowiskach wykonując funkcję filtrowania i przekazywania pakietów przez. Pomimo wielu dobrze działających tradycyjnych technologii, wielkość i skomplikowanie współczesnych zestawień sieciowych buduje potrzebę innowacji w tej dziedzinie. Głównymi powodami są wiecznie rosnące koszty posiadania i zarządzania sprzętem sieciowym oraz wzrastająca potrzeba na nowoczesne centra danych. Obecnie wiele firm sieciowych powoli odchodzi od tradycyjnych metod w kierunku bardziej wolnodostępnego i otwartego na innowacje paradygmatu SDN. [1]

Aby zrozumieć fenomen sieci programowalnych trzeba najpierw zdefiniować tradycyjną architekturę przełącznika. Przedstawić można ją w świetny sposób za pomocą trzech warstw:

1. **Data Plane** – składa się z portów odpowiedzialnych za pobieranie pakietów oraz   
   z tablicy skojarzeń (*forwarding table*). Warstwa ta jest odpowiedzialna za modyfikację nagłówka ramki oraz buforowanie i przekazywanie pakietów,
2. **Control Plane** – głównym zadaniem tej warstwy jest utrzymywanie aktualnych informacji w tablicy skojarzeń, tak aby data plane mogło kierować ruchem niezależnie bez ingerecji innych warstw. Tutaj właśnie procesowane są wszystkie protokoły odpowiedzialne za zmiane rekordów w tablicy skojarzeń, czyli odpowiedzialne za zarządzanie topologią sieci (np. RIP,OSPF, BGP itd.),
3. **Management Plane** – jest to ostatnia (najwyższa) warstwa modelu, odpowiedzialna zarządzanie urządzeniem bezpośrednio przez administratora   
   (np. Poprzez protokół SNMP).

Taki model warstwowy został przedstawiony na obrazku poniżej.  


Rysunek 1: Role data, control i management planes [1].

Warstwa control plane jest nieustannie *bombardowana* przez protokoły do wyznaczania ścieżek pakietów. Faktem jest, że w wielkich sieciach center danych, około 30% czasu ruterów jest wykorzystywane do śledzenia topologii sieciowej [1]. Mimo zaawansowanych protokołów trasowania, możliwe jest usprawnienie obecnych sieci, tym rozwiązaniem jest użycie sieci programowalnych. Polega to na tym, że warstwa control plane nie jest zarządzana przez protokoły trasowania, ale przez centralnie zlokalizowany *inteligentny* kontroler posiadający informacje o topologii sieciowej. Na rysunku poniżej przedstawiona jest wersja sieci zarządzana przez kontroler SDN.



Rysunek 2: Zcentralizowany kontroler zarządzający warstwą control plane [1].

Prostota tego rozwiązania wynika z tego, że topologia sieciowa, jest stabilna i pod ścisłą kontrolą administratora sieciowego w jednym zcentralizowanym miejscu. Dodatkowym atutaem przemawiającym za tym rozwiązaniem jest szybka diagnoza potencjalnych błędów w dużej sieci komputerowej.

* 1. **OVS**

Kolejnym ważnym pojęciem, które będzie się często przewijało w niniejszej pracy jest OVS (*Open Virtual Switch*) czyli wirtualny przełącznik. Jest to oprogramowanie przedstawiające wielowarstwowy switch, udostępniany na podstawie darmowej licencji (*Open Source Apache 2 License*). Jest doskonale dostosowany aby pełnić funkcję wirtualnego przełącznika w środowisku maszyn wirtualnych. Jest stworzony do automatyzacji sieci przy czym wspiera obecne standardy i protokoły (np. NetFlow, Sflow, IPFIX, RSPAN, CLI, LACP, 802.1ag). OVS wspiera wiele bazujących na linuxie technologii takich jak Xen/XenServer, KVM czy VirtualBox [2].

OVS jest przeznaczony do wieloserwerowch środowisk. Stan urządzeń wirtualnych   
w jednostce sieciowej (np. w maszynie wirtualnej) powinien być prosty zreprodukowania w innym miejscu. Wirtualne przełączniki to umożliwiają, poprzez możliwość migrowania tablic skojarzeń, listy kontroli dostępu (ACLs), polityki QoS itd [2].

Struktura danych wirtualnego przełącznika OVS ukazana jest na rysunku poniżej.



Rysunek 3: Struktura danych OVS'a [3].

Na powyższym rysunku możemy zauważyć, że wirtualny przełącznik posiada trzy podstawowe elementy:

1. Port – interfejs odpowiedzialny za połączenie OVSa z innymi urządzeniami. Jego konfiguracja jest ustawiana przez protokół OVSDB. W czasie działania wirtualnego przełącznika można w dowolnym czasie dodawać nowe porty i subinterfejsy,
2. Controller – każdy wirtualny switch może być zarządzany przez jeden lub więcej kontrolerów SDN,
3. Flow\_Table – tablica regół OpenFlow, czyli odpowiednik tablicy skojarzeń w tradycyjnym przełączniku (z tym, że posiada większe możliwości).

Najnowsza wersja oprogramowania OVS to 2.7.0 i ta wersja będzie wykorzystywana w zestawieniu sieci wirtualnej (TODO: przy tworzeniu upewnić się czy na pewno ta wersja czy nie starsza).

* 1. **OpenFlow**

Wraz z powstaniem sieci programowalnych, na scenę wkroczył protokół komunikacyjny OpenFlow. Protokół ten ma za zadanie komunikować warstwę danych *(data plane)* w przełączniku z warstwą *control plane* w kontrolerze. Aby kontroler mógł się porozumieć z switch’em za pomocą opisywanego protokołu, ten musi posiadać do niego wsparcie. Na teraźniejszym rynku sieci komputerowych istnieje coraz więcej przełączników wspierających protokół OpenFlow. Najbardziej znane są przełączniki wirtualne takie jak opisywany wcześniej OVS lub mniej znany *Indigo* stworzony przez firmę *BigSwitch* [1]*.* Niektóre firmy produkujące sprzęt sieciowy również wyposarzyły swoje przełączniki o wsparcie dla protokołu OpenFlow (tzw. switch’e hybrydowe). Warto tutaj wspomieć, że kontrolery SDN komunikują się z urządzeniami zarządzanymi nie tylko za pomocą protokołu OpenFlow, używają również innych protokołów by być w stanie porozumiewać się z jak największą liczbą urządzeń (SNPM, Netconf itd.).

OpenFlow powstał w celu ujednolicenia sposobu komunikacji kontrolera z przełącznikami. Na wysokim poziomie abstrakcji można powiedzieć, że standard OpenFlow definiuje następujące komponenty potrzebne do komunikacji kontroler - przełącznik:

1. Kontroler,
2. Interfejs OpenFlow w przełączniku,
3. Kanał do bezpiecznej komunikacji,
4. Tabele przepływów (*flow table*) zawierającą instrukcje przepływów[4].

Powyżej wymienione komponenty wizualizuje rysunek poniżej:



Rysunek 4: Podstawowe komponenty OpenFlow [4].

Więcej właśniwości OpenFlow zostanie omówione w dalszych rozdziałach.

* 1. **OVSDB**

Jak już to wcześniej było wspomiane, kontroler SDN może komunikować się z przełącznikami za pomocą wielu protokołów komunikacyjnych, jednym z nich jest OVSDB (*Open vSwitch Database Management Protocol*) stworzony przez firmę Nicra (później przejętą przez VMware). Został on stworzony wraz z stworzeniem wirtualnych przełączników OVS jako protokół ich zarządzania.

Istnieje ogólne przekonanie, że kontroler może porozumiewać się z wirtualnym przełącznikiem tylko za pomocą protokołu OpenFlow, ale jeżeli chodzi o konfigurację, tu wkracza OVSDB. Za jego pomocą można dodawać, modyfikować i usuwać wirtualne przełączniki OVS, ich porty i interfejsy. Możliwe jest zarządzanie urządzeniem OVS, wyłącznie przy użyciu protokoło OVSDB [5].

Na poniżeszym rysunku widać interfejsy wirtualnego przełącznika OVS, gdzie *ovsdb-server* to serwer bazy danych protokołu OVSDB, *ovs-vswitchd* to deamon vswitch’ai *openvswitch.ko* czyli moduł kernela wykonujący szybkie przekierowywanie ścieżek.



Rysunek 5: Interfejsy przełącznika OVS [6]

Wraz z pojawieniem się na rynku OVSDB, niektóre firmy postanowiły wprowadzić wsparcie dla tego protokołu, te firmy to np. *Cumulus, Arista i Dell* [5]*.* Więcej o OVSDB zostanie powiedziane w późniejszych rozdziałach.

* 1. **MEF**

Aby dopełnić wstęp teoretyczny do niniejszej pracy, trzeba powiedzieć czym jest MEF. Jest to skrót od *Metro Ethernet Forum* i jest to międzynarodowa organizacja zrzeszająca ponad 220 przedsięciorstw zajmujących dostarczaniem usług sieciowych, produkcją sprzętu sieciowego itd. Zajmuje się ona przyśpieszeniem zastosowania sieci komputerowych i usług typu *Carrier-class Ethernet* [7].

Organizacja MEF definiuje w jaki sposób dostawcy usług sieciowych powinni tworzyć usługi. Definiuje trzy podstawowe typy usług:

1. E-Line – usługa definiująca bezpośrednie połaczeniem między dwoma punktami końcowymi UNI (*User Network Interface*),
2. E-LAN – usługa umożliwiająca na więcej niż dwa interfejsy użytkownika (UNI), gdzie każdy z nich może przekazywać ramki eternetowe do każdego innego interfejsu UNI,
3. E-Tree – usługa mająca wiele interfejsów, posiada strukturę drzewiastą, przy czym zapobiega przekazywaniu ramek eternetowych w kierunku od interfejsów użytkownika o roli Liść *(Leaf)* do tych o roli Korzeń (*Root)*.

MEF do wszystkich wymienionych wyżej usług definiuje atrybuty, takie jak jakość usługi (*Quality of Sevice*), ramki usług na podstawie ramek Ethernet itd. Właśnie tego typu atrybuty będą wybrane jako kryterium porównania protokołów OpenFlow i OVSDB w niniejszej pracy i zostaną one dokładniej opisane w następnym rozdziale.

1. **Rozwinięcie**
   1. **Kryteria porównania**
2. **cos**

**Bibliografia**

# Paul Goransson, Chuck Black. Software Defined Networks: A Comprehensive Approach. Elsevier, 2014.

# Open vSwitch. Getting Started. <http://docs.openvswitch.org/en/latest/intro>, 2016. [dostęp: 2017-05-20].

# Rajdeep Dua. Open vSwitch deep dive. <https://www.slideshare.net/rajdeep/openvswitch-deep-dive>, 2013. [dostęp: 2017-05-20].

# D. Marschke, J. Doyle, P. Moyer. SDN: Anatomy of OpenFlow Volume I. Lulu, 2015.

# SdxCentral. What is Open vSwitch Database or OVSDB? <https://www.sdxcentral.com/cloud/open-source/definitions/what-is-ovsdb/>, [dostęp: 2017-05-21]

# B. Davie, The Open vSwitch Database Management Protocol, <https://tools.ietf.org/html/draft-pfaff-ovsdb-proto-00>, 2013, [dostęp: 2017-05-21]

# J. Kieffer, R. Satitoro, MEF-CECP Study Guide for Carrier Ethernet Professionals, Fujitsu Network Communications Inc., 2015.