Masarykova univerzita
Fakulta informatiky

Možnosti využití nástroje kontrolní vrstvy pro konfiguraci sítě v prostředí Kybernetického polygonu

Bakalářská práce

Petr Horáček

Brno, podzim 2017

Masarykova univerzita
Fakulta informatiky

Možnosti využití nástroje kontrolní vrstvy pro konfiguraci sítě v prostředí Kybernetického polygonu

Bakalářská práce

Petr Horáček

Brno, podzim 2017

Na tomto místě se v tištěné práci nachází oficiální podepsané zadání práce a prohlášení autora školního díla.

Prohlášení

Prohlašuji, že tato bakalářská práce je mým původním autorským dílem, které jsem vypracoval samostatně. Všechny zdroje, prameny a literaturu, které jsem při vypracování používal nebo z nich čerpal, v práci řádně cituji s uvedením úplného odkazu na příslušný zdroj.

Petr Horáček

Vedoucí práce: RNDr. Martin Vizváry

Poděkování

TBD

Shrnutí

TBD

Klíčová slova

«keywords»

Obsah

Seznam obrázků

Předběžné zadání: V prostředí Kybernetického polygonu je pro síťování využito zařízení Open vSwitch. Pro využití všech vlastností softwarově definované sítě je potřeba zapojit kontrolní vrstvu. Student se nejprve seznámi s problematikou softwarově definovaných sítí. Následně provede analýzu možností použití některého z dostupných nástrojů kontrolní vrstvy pro nastavení směrování u jednotlivých zařízení datové vrstvy. Ověření funkčnosti proběhne v prostředí Kybernetického polygonu. Všeobecné podmínky pro spolupráci naleznete na stránkach `http://www.muni.cz/ics/services/csirt/thesis`

1 Úvod

Kybernetický polygon (KYPO) je platforma poskytující uzavřené prostředí pro simulaci a analýzu kybernetických hrozeb na informačních systémech. Součástí simulovaného prostředí je zde i počítačová síť sestávající ze switchů, routerů a koncových stanic. Tuto síť je dále možné upravovat pomocí řízení síťového provozu (traffic shaping) a simulovat tak například ztrátovost reálné sítě [3].

Právě konfigurace sítě je však jednou z nejproblematičtějších částí současné verze KYPO. Část dosažení požadovaného stavu je automatizována, vždy je ale třeba provést několik dodatečných kroků manuálního nastavení. Cílem této práce je automatizovat co největší část této konfigurace a zpřístupnit tak KYPO i uživatelům bez dostatečných znalostí administrace sítě.

Jednou z cest pro dosažení tohoto cíle je použit software-defined networking (SDN), tím můžeme přesunout konfiguraci logické topologie do tzv. overlay sítě. Vše tak bude probíhat centralizovaně a nezávisle na poskytnuté fyzické topologii. Nástroje kontrolní vrstvy SDN navíc krom virtualizace routerů a switchů poskytují celou řadu virtuálních síťových funkcí, například ACL či traffic shaping [2].

V následující kapitole se budeme věnovat analýze současného síťového řešení KYPO tak, abychom identifikovali jeho problematické částí. Analýza je také nutná pro správné porozumění současného kódu a možnost jeho rozšíření či úpravy. Na závěr této kapitoly si stanovíme požadavky na nové síťové řešení tak, aby řešilo identifikované problémy a zároveň jej bylo možné snadno integrovat do stávající infrastruktury. Jednotlivé sekce budou doplněny příklady zapojení či konfigurace.

Dále navrhneme architekturu nového řešení vzhledem k daným požadavkům a existující infrastruktuře. Zpracujeme přehled dostupných nástrojů kontrolní vrstvy SDN a pokusíme se vybrat ten nejvhodnější. Na závěr kapitoly upřesníme architekturu nového řešení využívající vybraný nástroj.

V další kapitole popíšeme jednotlivé kroky vedoucí k integraci nového řešení. Popíšeme aplikační rozhraní nových modulů a vytvoříme jejich implementaci.

V poslední kapitole se budeme věnovat testování a použitelnosti vytvořeného řešení.

2 Klasické sítě ve virtuálním prostředí

[TODO jak probiha síťování bez použití SDN, spojení L2, routování L3, problémy s tím spojené]

3 Software defined networking

[TODO o co jde, co ulehčuje, co vyžaduje]

3.1 Přehled dostupných nástrojů kontrolní vrstvy SDN

Aktualizace přehledu z roku 2015.[1] Neudržované a proprietární projekty nejsou uvedeny, seznam byl naopak rozšířen o nově dostupná řešení.

3.1.1 Floodlight

[TODO]

3.1.2 Frenetic

Jde o programovací jazyk určený pro vývoj SDN kontroler. Jeho použití by vyžadovalo implementaci vlastního SDN řešení.

3.1.3 Maestro

Nejde o kompletní řešení, je součásní komerčního Cirries, které využívá Maestro pro kontrolu síťových aplikací.

3.1.4 McNettle

Framework vytvořený pro univerzitní projekt. Nepodařilo se mi najít dokumentaci ani zdrojové kódy.

3.1.5 ONOS

[TODO]

3.1.6 OpenDaylight

[TODO zbytečně velký pro náš učel]

1. `https://is.muni.cz/auth/th/396543/fi_b/bc.pdf`

3.1.7 OVN

[TODO snadno použitelný z příkazové řádky; abstrakce pro L2, L3, L4, QoS; aktivní komunita vývojářů i uživatelů]

3.1.8 ovs-controller

Jde pouze o referenční implementaci, a tedy nevhodné řešení pro reálné nasazení.

3.1.9 Faucet

Poměrně nový kontrolér postavený na Ryo. [TODO]

3.1.10 Trema

Framework C/Ruby pro vytvoření vlastního kontroléru.

3.1.11 Cherry

[TODO Neudržovaný projekt s jedním autorem a bez aktivity]

3.1.12 OpenContrail

Aktivní a poměrně dobře zdokumentovaný projekt. Nabízí kompletní řešení SDN.

3.1.13 Neutron

[TODO má celou řadu závislostí na OpenStack a lze jej jen těžko použít samostatně]

4 Analýza současného řešení

V minulých kapitolách jsme si postali základní prvky sítí ve virtuálním prostředí a popsali několik existujících řešení. V této kapitole se budeme věnovat současnému síťovému řešení KYPO. Porozumění současnému řešení je nutné nejen pro identifikaci jeho problémů, ale také pro následnou implementaci nového řešení, které bude muset nevyhnutelně vycházet ze stávajícího kódu a infrastruktury.

Nejprve si popíšeme logickou síťovou topologii tak, jak je viděna z pohledu koncové stanice. Následně si popíšeme proces nutný pro vytvoření této sítě ve dvou krocích. Prvním krokem je vytvoření virtuálního hardwaru prostřednictvím virtualizační platformy (CMP) využívané v KYPO. V dalším kroku na tomto hardwaru nakonfigurujeme částečně automatizovaným procesem požadovanou síť.

Na základě popsaného procesu vyhodnotíme nedostatky současného řešení a obtíže s ním spojené. Na závěr stanovíme funkční a návrhové požadavky na nové řešení.

4.1 Logická topologie

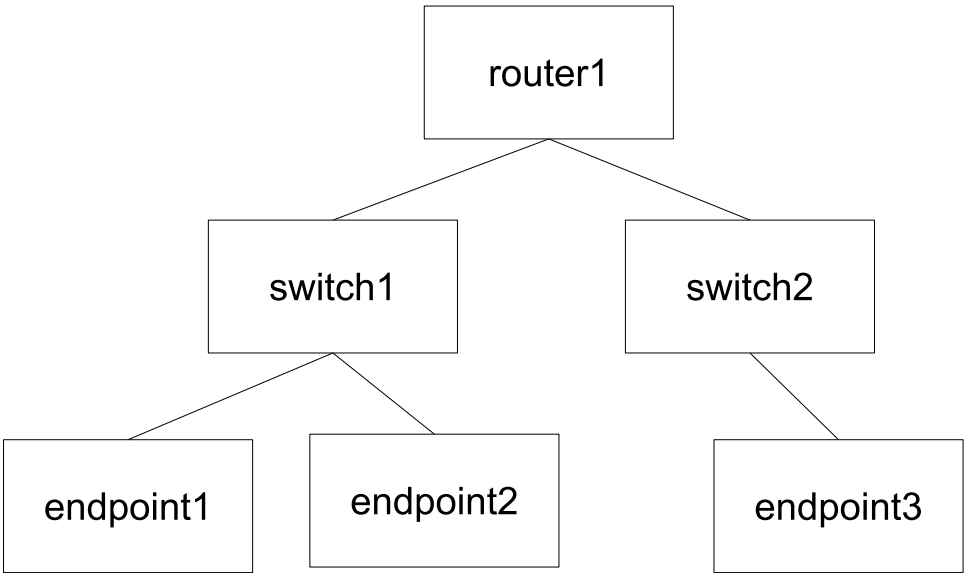
Před sestavením simulovaného prostředí je v grafickém rozhraní KYPO specifikována logická topologie sítě. Tímto způsobem definujeme to, jak bude výsledná síť vypadat z pohledu uživatele koncové stanice. Součástí modelu jsou koncové stanice, switche a routery. Tyto uzly jsou mezi sebou libovolně propojeny. Pro příklad logické topologie viz 4.1.

Tato specifikace je předána backendu KYPO, který vytvoří požadovanou hardwarovou strukturu pomocí CMP a částečně automatizovaně ji nastaví, jak si popíšeme v následujících dvou sekcích.

4.2 Fyzická topologie

Prvním krokem pro dosažení stanovené logické topologie je vytvoření virtuálních strojů a síťových spojení prostřednictvím CMP.

KYPO vytvoří virtuální stroj (VM) pro každou koncovou stanici a switch. Všimněte si, že není vytvořena zvláštní VM pro router, to je



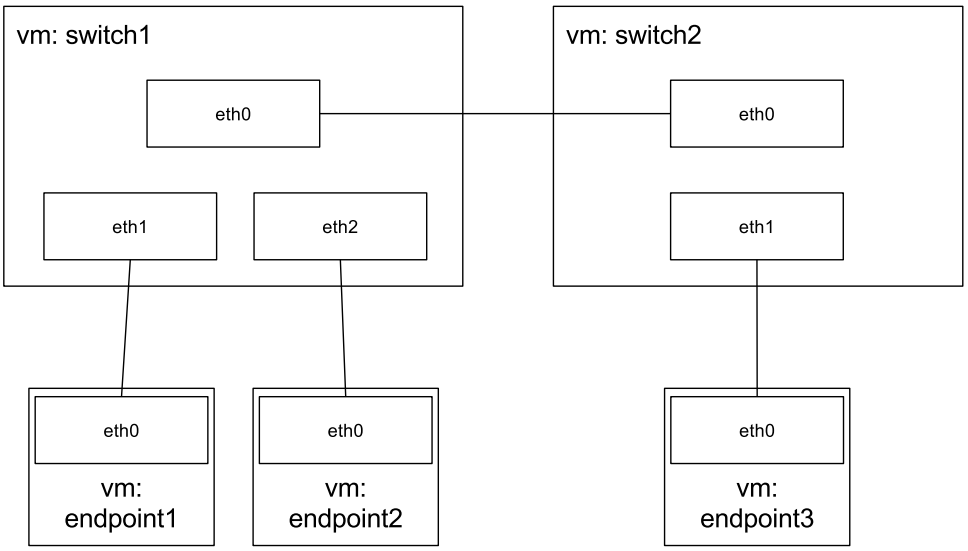
Obrázek 4.1: Příklad logické topologie z pohledu koncového uzlu.

dáno tím, že routování je nastaveno v rámci stanice switche (jde tedy o jakýsi L3 switch).

Následně jsou tyto VM propojeny „virtuálními kabely,"' tedy dvoubodovými sítěmi sdílenými VM na obou koncích kabelu. Tyto sítě mohou být implementovány například pomocí VXLAN. Pro ilustraci zapojení jednotlivých VM, VXLAN spojení, rozhraní a Open vSwitch viz 4.2.

Důležitým prvkem této konfigurace je to, že pro síťové spojení koncových stanic není přímo využita síťová infrastruktura CMP. Namísto toho je nad síťovým řešením CMP postaveno další skrze virtuální switche spravované prostřednictvím KYPO. Díky tomu není sestavení KYPO závislé na použitém CMP, jediný požadavek na použitou platformu je možnost vytvoření virtuálních sítí. Tento „overlay nad overlay"' rozhodně není vhodný pro produkční nasazení virtualizace, pro sandboxování je ale naopak ideální, poskytuje plnou kontrolu nad implementací sítě a zmíněnou nezávislost na CMP.

Tato topologie je dostatečná pro dvoubodové spojení jednotlivých stanic, samotný routing a switching větších L2 sítí však nezajišťuje,



Obrázek 4.2: Příklad zapojení VM a síťových rozhraní.

pouze poskytuje virtuální hardwarové prostředí nad kterým můžeme následně provést konfiguraci daných komponent.

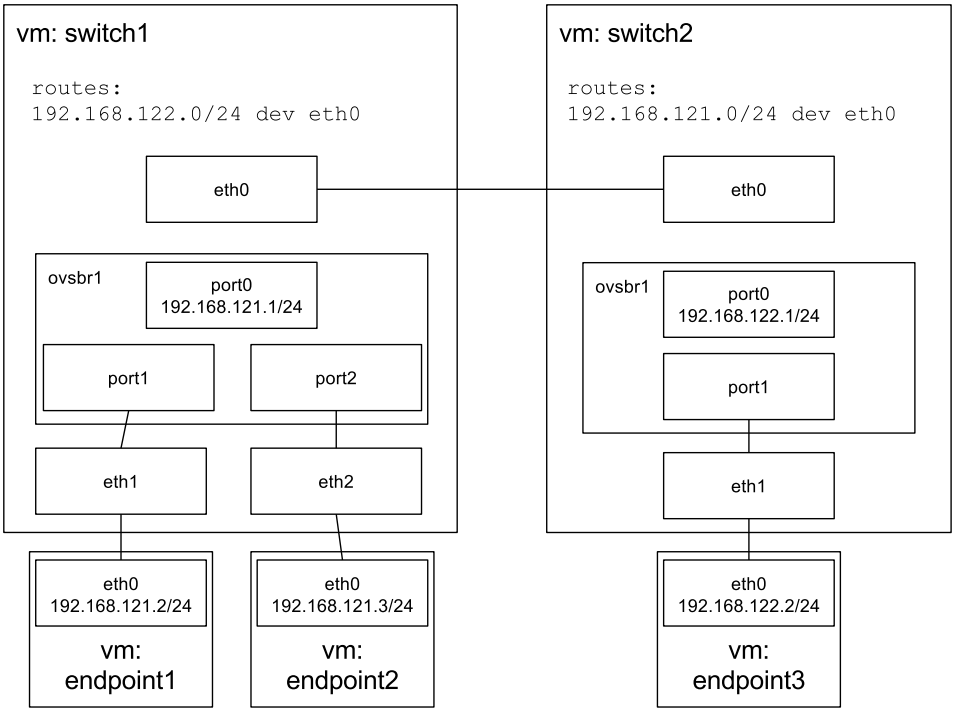
4.3 Konfigurace

Finálním krokem pro dosažení specifikované topologie je konfigurace uzlů v připraveném virtuálním clusteru. Tento krok je pouze částečně automatizován.

KYPO vzdáleně vytvoří na VM switchů instance virtuálního switch Open vSwitch. Do tohoto switche jsou připojeny rozhraní virtuálních karet dvoubodových sítí vedoucích k dalším zařízením (switchům, či koncovým stanicím). Tímto je dosaženo vytvoření jednotlivých L2 sítí.

Pro nastavení routování je nutný manuální zásah správce do VM switchů. Routovací pravidla se specifikují pomocí standardních linuxových routovacích tabulek, kde síťová rozhraní vedoucí k dalším „routerům“ jsou nastaveny jako výstupní zařízení pro danou síť. Problém nastává když se v celém virtuálním clusteru nachází velké množství sítí a každá vyžaduje nastavení routovacích pravidel na všech zúčastněných switchích. Takový zásah je pro správce-člověka náročný

a náchylný na chyby. Pro příklad kompletně nakonfigurované sítě viz 4.3



Obrázek 4.3: Příklad nakonfigurované sítě.

Aby bylo možné simulovat reálné podmínky je nutné provést zásahy do síťového toku. Jednotlivé stanice se nacházejí v jediném clusteru, někdy i na jediném serveru, proto může být odezva mezi nimi minimální. Pro napodobení reálných sítí pracujících nad větší rozlohou je nutné do toku přidat násilné zpoždění. Pro simulaci bezdrátových sítí je navíc nutné napodobit jejich vysokou ztrátovost. Také tyto úpravy nelze v současné době provést automaticky a je nutný manuální zásah na jednotlivých strojích.

4.4 Nedostatky řešení

Z předchozí kapitoly jsou jasně vidět největší problémy současného řešení. V prvé řadě jde o nutnost manuální konfigurace routovacích

tabulek, což může být problémem především u sítí většího rozsahu. Podobným problémem je nutnost manuálního nastavení traffic shaping přímo na uzlech.

Tyto problémy vystavují uživatele nutnosti manuálního nastavení konfigurace sítě a jejímu porozumění. To může být překážka zabraňující efektivnímu užití KYPO.

Další problémem, ač v tuto chvíli méně závažným, jsou komplikace při změně topologie sítě za chodu. To by v současné době znamenalo nutnost vytvoření nových VXLAN sítí a VM pro switche, dynamickou úpravu portů instancí Open vSwitch a opětovnou manuální editaci routovacích tabulek.

4.5 Funkční a návrhové nároky na nové řešení

Jako vhodné řešení problémů popsaných v minulé sekci se jeví použití SDN. To nám umožní centralizovaně modelovat běžící síť, a to i dynamicky. Pomocí OpenFlow pravidel můžeme nastavit pokročilé funkce sítě, které nemusí být poskytované aplikací kontrolní vrstvy, například load-balancing.

Již integrované instance Open vSwitch použité na switchích lze využít skrze OpenFlow protokol pro datovou část SDN, stačí jej doplnit vhodnou aplikací kontrolní vrstvy [1].

[TODO problém s dvojitostí KYPO, nemůžeme dost dobře použít klasické SDN protože nemáme overlay, je nutné zařídit routování underlay, nebo všechny network hosty sloučit do jednoho. kompromisem mezi použitím plného SDN a současným řešení KYPO by bylo využití OF pro routování, zbytek by zůstal]

Bibliografie

[1] *Features*. 2016. URL: http://openvswitch.org/features/ (cit. 02. 05. 2017).

[2] Matej LEITNER. *Přehled dostupných nástrojů pro kontrolní vrstvu softwarově definovaných sítí [online]*. Bakalářská práce. 2015 [cit. 2017-05-02]. URL: http://is.muni.cz/th/396543/fi_b/.

[3] Pavel Čeleda et al. „KYPO – A Platform for Cyber Defence Exercises“. eng. In: *STO-MP-MSG-133: M&S Support to Operational Tasks Including War Gaming, Logistics, Cyber Defence*. Munich (Germany): NATO Science a Technology Organization, 2015, nestránkováno. ISBN: 978-92-837-2020-1. URL: https://is.muni.cz/repo/1319597/kypo-paper-msg-133.pdf.