

Funzionalità di OVSDB e rapporto con Openflow

Stefano Pievaioli 816592

Sistemi e Servizi di Telecomunicazione, Università Milano-Bicocca

Estratto – In questa relazione tratto le funzionalità di OVSDB e il suo rapporto con OpenFlow. Nella relazione ci sarà prima una breve descrizione dell'ambiente in cui opera il protocollo OVSDB. Più precisamente viene cos'è, e da dove deriva SDN, il suo protocollo standard, OpenFlow e come opera. Ed infine cos'è l'OVS e OVSDB.

I. Introduzione

Negli ultimi anni, c'è stata una crescita esponenziale degli utenti connessi alla rete internet e di conseguenza le infrastrutture della rete sono dovute crescere aumentando così anche la difficoltà nel gestirle. Oltre alla crescita degli utenti è nata la necessità di poter accedere ai contenuti aziendali, e alle proprie applicazioni ovunque ci si trovi, in qualunque momento e con qualsiasi tipo di dispositivo. Tutto questo ha portato a un'inevitabile richiesta di maggiori infrastrutture per poter accedere alle proprie risorse anche con una adeguata sicurezza. Quindi è emersa la necessità di reti più elastiche, in modo da addarsi al traffico di dati in quel momento. Un primo approccio è avvenuto con la diffusione di tecnologie di "cloud computing", che sono un insieme di infrastrutture di elaborazione, Database, applicazioni accessibili da remoto e attraverso qualsiasi rete. In tutti i servizi di cloud i server sono virtuali, ovvero una macchina non contiene un singolo server ma contiene più server sotto forma di macchine virtuali. L'utilizzo di queste virtualizzazioni ha portato anche alla virtualizzazione di tutte le componenti della rete, come per esempio degli switch. Da qui nasce OVS, Open vSwitch, ovvero degli switch virtuali fatti per interconnettere macchine virtuali. E da qui anche il suo protocollo OVSDB. OVS ha diversi vantaggi, tra i quali offrire una buona flessibilità oltre ad avere un controllo centrale, dinamico e facilmente scalabile della rete. Lo svantaggio è che le risorse hardware devono essere molto performanti e per garantire l'affidabilità le infrastrutture a volte devono essere ridondanti. L'OVS viene

gestito attraverso gli SDN e in particolare dal protocollo di OVSDB, Open vSwitch Database, un'interfaccia moderna per la gestione e l'automazione della rete.

II. Software Defined Networking

Software Defined Networking (SDN) è una rappresentazione della rete che permette, tramite un software, di avere una gestione centralizzata e intelligente di tutti i componenti hardware utilizzati nella rete. SDN utilizza dei protocolli aperti come OpenFlow, che consentono l'accesso a dispositivi di rete come switch, router o firewall che altrimenti non sarebbero controllabili a causa del firmware proprietario. Il SDN garantisce una notevole semplificazione nella gestione delle reti e ne consente un'innovazione ed evoluzione. SDN ha come obiettivo principale quello di permettere agli sviluppatori software di fare affidamento alle risorse di rete con la stessa facilità con cui utilizzano risorse di storage e di calcolo. Diversamente dalle reti

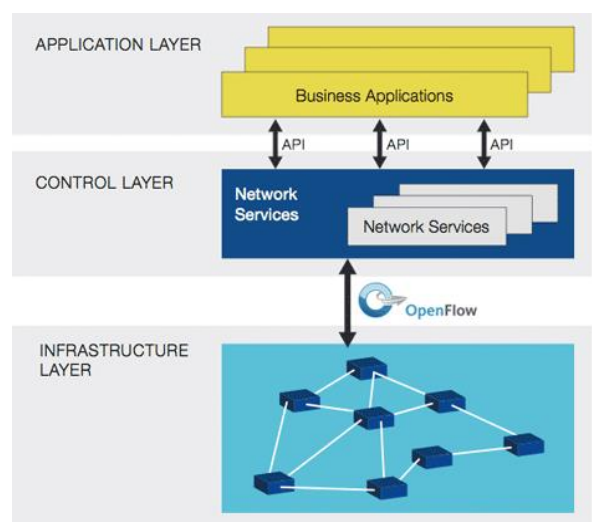


Figura 1, Architettura SDN

tradizionali il control plane e il data plane sono separati per permettere una migliore amministrazione e un controllo più flessibile.

Centralizzando lo stato della rete nel control layer, SDN permette ai gestori di rete di avere flessibilità per la configurazione, la gestione, e l'ottimizzazione delle risorse di rete mediante programmi SDN dinamici e automatizzati, con anche molta sicurezza. La figura 1 mostra l'architettura del SDN. L'architettura SDN si divide in tre parti:

- Application Plane, insieme di tutte le applicazioni utenti che comunicano con il piano sottostante tramite delle API.
- Control Plane, livello centralizzato che gestisce l'intera rete. Il controller ha una vista globale sulla rete e decide le politiche di forwarding dei pacchetti del data plane.
- Data Plane, insieme degli elementi della rete che consentono l'inoltro fisico dei pacchetti (switch). Più precisamente il data plane si preoccupa di fare il forwarding dei pacchetti usando le tabelle di forwarding fornite dal control plane. Questo processo semplifica enormemente la logica del data plane, riducendo la complessità e il costo degli switch.

I servizi API rendono possibile l'implementazione di comuni servizi di rete, inclusi routing, multicast, sicurezza, controllo degli accessi, gestione della banda... tutto per raggiungere il proprio obiettivo.

III. OpenFlow

Il Software Defined Networking è stato sviluppato per facilitare l'innovazione e abilitare un semplice controllo programmatico del percorso dati nella rete, il protocollo che ha avuto più successo è OpenFlow che è diventato lo standard di SDN. OpenFlow, definito dalla Open Networking Foundation (ONF), è utilizzato per l'implementazione di SDN nelle apparecchiature di rete ed è costituito da tre componenti:

- Controller OpenFlow, utilizzato per impostare regole sulla trasmissione dei dati di ogni dispositivo.

- Protocollo OpenFlow, utilizzato dal controller per comunicare in modo sicuro con lo switch OpenFlow.
- Tabelle di flusso per configurare lo switch OpenFlow.

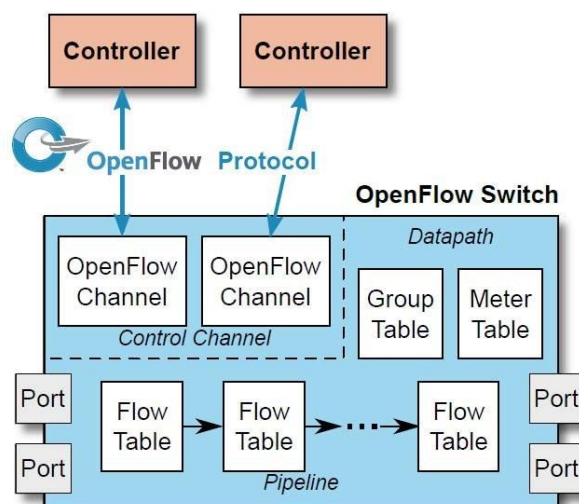


Figura 2, Architettura OpenFlow

OpenFlow utilizza il concetto di flusso per identificare il traffico rete basandosi su predefinite regole di matching che possono essere statiche o dinamiche.

Quando un pacchetto di dati arriva allo switch OpenFlow, esso viene elaborato dallo switch attraverso le tabelle di flusso partendo dalla tabella di flusso zero.

Un controller OpenFlow è in grado di gestire più canali contemporaneamente, provenienti da switch diversi. E può avere un canale indirizzato a uno o più canali per ogni controller. Le connessioni tra controller e switch del canale sono inizializzate da parte dello switch mediante il protocollo crittografico TLS. Alternativamente, attraverso una porta TCP, il controller può instaurare la connessione. Una vera rete compatibile con OpenFlow può essere costituita solo da switch OpenFlow o da una combinazione di switch OpenFlow e switch e router tradizionali. Quest'ultimo tipo di rete è chiamato rete overlay.

Alcune applicazioni OpenFlow richiederanno solo l'implementazione parziale degli switch OpenFlow, mentre altre richiedono una rete composta solo da switch OpenFlow. Un vantaggio di OpenFlow è che le SDN basate su esso possono essere sviluppate su reti sia fisiche che virtuali.

IV. Open vSwitch

In passato, i server si connettevano fisicamente a uno switch basato su hardware situato nel data center. Con le macchine virtuali di VMware si è vista la necessità di cambiare il livello di accesso alla rete, più precisamente la necessità di essere connesso a uno switch virtuale rispetto alla possibilità di connettersi a uno fisico. Questo switch virtuale è un livello software che risiede in un server che ospita macchine virtuali (VM). Le VM, e ora anche i container, come Docker, hanno porte Ethernet logiche o virtuali. Queste porte logiche si collegano a uno switch virtuale. Open vSwitch è uno switch software multilayer con licenza Open-Source Apache 2.0. Il suo obiettivo è quello di implementare una piattaforma di

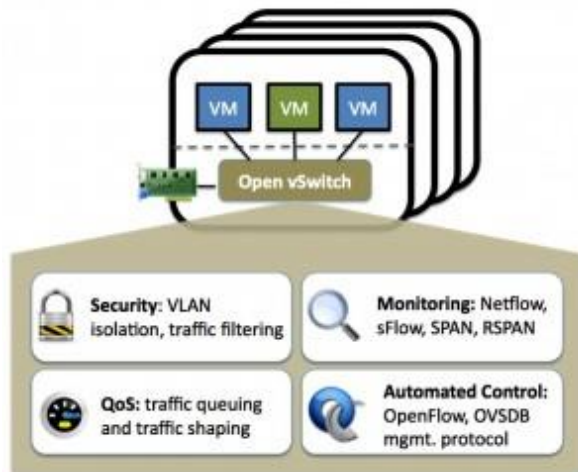


Figura 3, Architettura OVS

switch di rete multistrato virtuale, che supporti le interfacce di gestione standard, e aprire al controllo e alla programmabilità delle funzioni di inoltro. Open vSwitch è adatto a funzionare come switch virtuale negli ambienti con macchine virtuali. OVS non espone solo interfacce per il controllo standard e di visibilità al layer di networking virtuale, ma è stato progettato per essere applicato a server fisici. Open vSwitch inoltre supporta diverse tecnologie di virtualizzazione, tra cui Xen/XenServer, KVM e VirtualBox.

La versione corrente di Open vSwitch supporta le seguenti funzionalità:

- Modello standard VLAN 802.1Q con trunk e porte di accesso;
- Collegamento NIC con o senza LACP, che consente di raggruppare più porte fisiche per avere un unico canale logico.

- NetFlow, sFlow e mirroring.
- Configurazione QoS.
- OpenFlow 1.0 più numerose estensioni.

V. Funzionalità di OVSDb

OVSDb, Open vSwitch Database è una struttura composta da uno o più database, che sono accessibili dalla rete. Il protocollo OVSDb è progettato per la sincronizzazione dello stato della rete. Il protocollo consente a ciascun client di monitorare totalmente i contenuti di un intero database in modo efficace. Infatti, il server comunica immediatamente al client le ultime modifiche avvenute.

OVSDb supporta tre modelli di servizio per i database:

- standalone
- backup attivo
- cluster

I modelli di servizio offrono diversi compromessi tra coerenza, disponibilità e tolleranza delle partizioni. Differiscono anche nel numero di server richiesti e in termini di prestazioni.

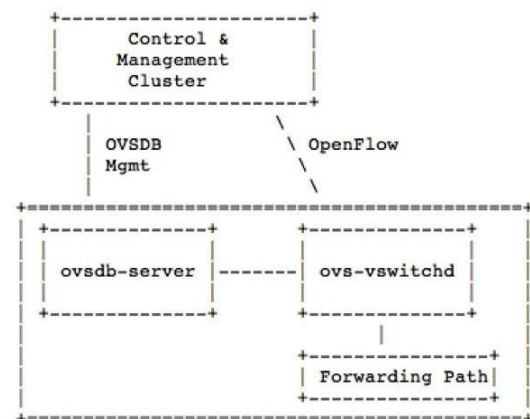


Figura 4, Architettura OVSDb

OVSDb standalone viene spesso utilizzato in contesti in cui i contenuti del database non sono particolarmente preziosi. Ad esempio, in molti sistemi, il database per la configurazione di ovs-vswitchd viene essenzialmente ricostruito da zero al momento dell'avvio. Quando OVSDb viene utilizzato per dati preziosi, è consigliabile prendere in considerazione una strategia di backup.

Gli schemi in OVSDb, scritti in JSON con protocollo RFC 7047, servono per descrivere

le tabelle di uno dei database, il tipo di dato per ogni colonna e possono includere dati, unicità e vincoli di integrità referenziale.

Per accedere al OVSDb ci sono diversi metodi, classificabili in attivi e passivi, un metodo attivo effettua una connessione attraverso una stringa di connessione. Invece un metodo di connessione passivo ascolta le connessioni da host remoti.

I metodi che vengono più utilizzati in un OVSDb sono:

- **List Databases**, comando che recupera un array i cui elementi sono i nomi dei database a cui è possibile fare accesso tramite OVSDb
- **Get Schema**, recupera uno schema nel DB.
- **Transact**, permette di eseguire su un certo database una serie di operazioni nell'ordine da noi specificato.
- **Cancel**, istruisce il database server a completare o cancellare la richiesta "transact".
- **Monitor**, abilita un client a replicare delle tables o dei sottoinsiemi di tables contenute in un database OVSDb
- **Update Notification**, inviato da un server ad un client per notificare una modifica.

VI. Rapporto tra Open vSwitch e OpenFlow

Open vSwitch e OpenFlow sono entrambi utilizzati per applicazioni SDN. OpenFlow è uno dei primi standard SDN. Open vSwitch, invece, è una implementazione open source di uno switch virtuale che lavora su più livelli. Per quanto riguarda la loro relazione, Open vSwitch è una delle implementazioni più popolari di OpenFlow. Oltre a OpenFlow, Open vSwitch supporta anche altri protocolli di gestione degli switch come OVSDb (Open vSwitch Database Management Protocol).

Quando si vuole utilizzare una rete SDN, o più precisamente si vuole distribuire un controller SDN con degli switch virtuali (OVS), per programmare le voci di flusso si deve utilizzare OpenFlow, l'OVS stesso deve essere configurato con il protocollo OVSDb. Se invece lo switch OVS viene distribuito in un ambiente autonomo, quindi senza SDN, non

c'è motivo per cui OVSDb non possa essere utilizzato da solo per configurare OVS (ambiente non OpenFlow). Sebbene ciò sia possibile, esistono davvero pochissime piattaforme di gestione della rete standalone che supportano OVS o, in particolare, OVSDb nativo.

Nelle maggior parti SDN per comunicare con il resto del mondo viene utilizzato OpenFlow mentre per comunicare con il server OVSDb viene utilizzato l'OVSDb protocol.

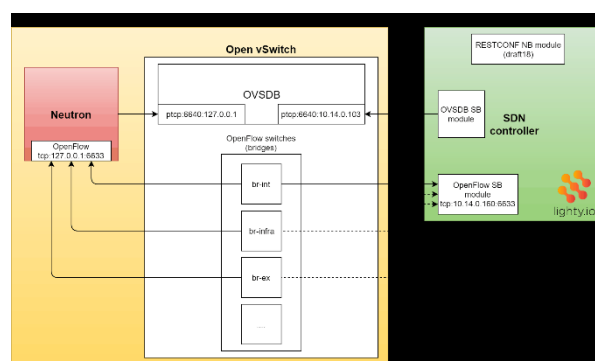


Figura 5, OVS e OF

Nella figura 5 possiamo vedere un esempio di una applicazione SDN con un OVS gestito dal protocollo OVSDb.

VII. Conclusione

L'obiettivo di questa relazione è quello analizzare le funzionalità di OVSDb e come esso si relazioni con l'implementazione di OpenFlow. Come abbiamo visto, OVSDb e OpenFlow possono tranquillamente coesistere all'interno della stessa rete in quanto agiscono su parti di comunicazione diverse. OVSDb risulta molto efficiente però uno degli svantaggi è la difficoltà nel comporre i messaggi JSON-RPC in quanto risulta più facile usare i comandi nativi di Open vSwitch che invece già implementano OVSDb. Inoltre, poiché OVSDb si basa sull'architettura OVS, ogni dispositivo deve essenzialmente avere le stesse caratteristiche o interfacce interne di uno switch virtuale e ciò comporta una limitazione sulla classe di dispositivi su cui è possibile lavorare.

VIII. Bibliografia

- [1] IL PROTOCOLLO OVSDb PER LA GESTIONE DI SWITCH ETHERNET

VIRTUALI, di Luca Falco,

<https://amslaurea.unibo.it/17887/1/II%20protocollo%20OVSDb%20per%20la%20gestione%20di%20switch%20Ethernet%20virtuali.pdf>

- [2] IMPLEMENTAZIONE OPEN VSWITCH DELLE FUNZIONALITÀ DI FORWARDING DEL PROTOCOLLO LISP IN UNO SCENARIO SOFTWARE DEFINED NETWORK, di David Lo Bascio, <https://www.extraordy.com/wp-content/uploads/2015/10/Tesi-di-Laurea-Magistrale-David-Lo-Bascio.pdf>
- [3] OPENFLOW E SOFTWARE-DEFINED NETWORKING: L'EVOLUZIONE DELLA RETE PROGRAMMABILE, di Alessandro Forgione https://amslaurea.unibo.it/7919/1/forgione_alessandro_tesi.pdf
- [4] What Is Open vSwitch Database or OVSDb? <https://www.sdxcentral.com/cloud/open-source/definitions/what-is-ovsdb/>
- [5] SDN: cosa si cela dietro il Software Defined Networking <https://www.ionos.it/digitalguide/server/know-how/software-defined-networking/>
- [6] The basics of SDN and the OpenFlow Network Architecture <https://noviflow.com/the-basics-of-sdn-and-the-openflow-network-architecture/>
- [7] ovsdb, <https://docs.openvswitch.org/en/latest/ref/ovsdb.7/#:~:text=OVSDb%2C%20the%20Open%20vSwitch%20Database,a%20network%2Daccessible%20database%20system.&text=The%20leading%20use%20of%20OVSDb,schema%20documented%20in%20ovs%2Dvswitchd.>