
# Virtualització de xarxes

TREBALL FINAL DE GRAU

Ricard Mañalich Dionisio
Assignatura: TFG ARSO
Consultor: Manel Mendoza Flores
Curs acadèmic 2015-16

*Dedicatòria*

*A Gemma, el meu suport.*

*"Obstacles are things a person sees when he takes his eyes off his goal."*

*"If there is a way to do it better, find it."*

*Thomas A. Edison*

*Agraïments*

En primer lloc, agrair a la meva companya de viatge Gemma, per recolzar-me i ajudar-me sempre. Gràcies per tenir tanta paciència amb mi als meus moments d'estrès, per aguantar-me, donar-me ànims quan més els necessitava i per estar sempre al meu costat. Juntament amb ella, la meva petita Xènia, per distreure'm dels meus pensaments i portar un somriure al meu rostre. Us estimo.

En segon lloc, vull agrair als meus pares per l'esforç que han posat en donar-me la millor educació i recolzar-me sempre en tot el que he fet. Gràcies per ajudar-me en tot moment i per tot l'afecte que m'heu demostrat sempre.

També, vull agrair a tots els meus professors de la universitat per haver-me ensenyat tant i per demostrar-me l'apassionant que pot arribar a ser la informàtica. Especialment a la meva tutora Neus Heras, per ser una gran tutora i ajudar-me a aconseguir unes condicions favorables per així, poder afrontar aquesta carrera.

Finalment, volia agrair també al meu professor de TFG, en Manel Mendoza, per donar-me la possibilitat de realitzar aquest projecte i ensenyar-me tant en tan poc temps. Ha estat un plaure poder treballar amb tu.

A tots, gràcies.

# Resum

L'actual administració i configuració de les xarxes a través de diferents dispositius individuals (Routers, switches, etc.) mitjançant interfícies, fent servir en la majoria dels casos modes d'operació que varien segons el fabricant, ha generat una major complexitat operativa alhora que ha alentit el procés d'innovació, augmentant considerablement els costos.

L'arquitectura Software Defined Networking (SDN) està canviant la forma en què es dissenyen i es gestionen les xarxes. SDN compta amb dues característiques principals ben definides, en primer lloc separa el pla de control (encarregat de regular el tràfic), de la part de dades. En segon lloc, gestiona el procés de control del tràfic unificant-ho mitjançant un únic Software el qual gestiona múltiples elements. SDN actua sobre els diferents dispositius mitjançant una sèrie d'aplicacions (Application Programming Interface o API).

OpenFlow per exemple, és una de les API's encarregades en aquest cas de gestionar el tràfic a través d'interruptors simulant switchos, mitjançant taules de regles. Cadascuna d'aquestes regles realitza diferents accions sobre el tràfic generat (forwarding, drooping, flooding, etc.). Depenent de les regles instal·lades, OpenFlow pot comportar-se com router, switch, firewall, traductor d'adreces de xarxa o de qualsevol dispositiu intermedi.

En els darrers anys, SDN i NFV han guanyat protagonisme en les empreses. Molts fabricants de switchos han donat suport a la API OpenFlow, entre els quals es troben proveïdors tan importants com HP o NEC. Grans empreses TIC (proveïdors Cloud, serveis financers, proveïdors d'equips, etc.) s'han unit a consorcis de l'arquitectura SDN com Open Networking Foundation.

# ÍNDEX PÀGINES PRELIMINARS



# ÍNDEX DE CONTINGUTS

# ÍNDEX DE FIGURES

# ÍNDEX DE TAULES

# CAPÍTOL 1

# 1. INTRODUCCIÓ

La necessitat per part de les empreses d'atendre i respondre a les demandes del mercat en un temps cada vegada més reduït, requereix que les xarxes hagin de proporcionar aquesta agilitat i flexibilitat sense comprometre el rendiment, la seguretat, l'escalabilitat i l'estabilitat. Les arquitectures SDN (Software Defined Networking) i NFV (Network Functions Virtualization) proveeixen infraestructures de xarxa més ràpides, eficients, dinàmiques i escalables.

## 1.1. Justificació del TFC

Aquesta tecnologia en constant creixement, poc coneguda entre els usuaris professionals i empreses, però, amb moltes expectatives en quan a la seva present i futura utilitat, fa d'ella un tema imprescindible pel que fa al continuo desenvolupament de les xarxes. Existeixen diversitat d'opinions relatives a la seva funcionalitat, pel que aquest projecte intentarà dotar del suficient coneixement del tema, el qual permeti compartir o bé discrepar amb aquesta tecnologia.

La virtualització de xarxes proporcionarà a les empreses una major facilitat en quan a usabilitat i les farà més programables i compatibles, aconseguint d'aquesta forma una gestió més senzilla i automatitzada.

## 1.2. Objectius del TFC

El principal objectiu és poder recopilar informació relativa a aquesta tecnologia de manera que qualsevol persona que no hagi tractat mai el tema pugui extreure el suficient coneixement per poder entendre-ho. Es pretén a més, especificar en profunditat cadascuna de les dues arquitectures per desmitificar conceptes erronis i saber diferenciar les principals característiques que aporten.

## 1.2. Estructura de la memòria

La memòria està estructurada per capítols de la següent forma:

- Capítol 1: Justificació i estructura de la memòria.
- Capítol 2: es fa un resum de la l'evolució que ha patit la virtualització de les xarxes, des de els seus orígens fins a l'any 2015.
- Capítol 3: es fa referència als dos tipus d'arquitectures que formen l'actual virtualització de xarxes.
- Capítol 4: es parla dels protocols Openflow i Openstack, principals protocols de la anomenada virtualització, fent una descripció especial i concisa del funcionament del primer.
- Capítol 5: tracta els principals estàndards disponibles avui en dia, tant a nivell de software com de hardware.
- Capítol 6: exemples de casos d'ús de les dues arquitectures.
- Capítol 7: creació d'una petita xarxa SDN mitjançant l'emulador de xarxes Mininet.
- Capítol 8: conclusions finals del treball

# CAPÍTOL 2

## 2. INTRODUCCIÓ A L'EVOLUCIÓ DE LES XARXES

Tot i que el boom de la virtualització de xarxes s'ha magnificat als darrers anys, el seu origen i la seva evolució es remunten als últims 20. Dividint aquests darrers 20 anys en tres etapes, cada etapa aporta la seva pròpia contribució al desenvolupament final de l'arquitectura de la xarxes virtualizadas. En la primera etapa (1995-2000) es troben les xarxes actives, introduint funcions programables a la xarxa, permetent d'aquesta forma augmentar la innovació. La segona etapa (2001-2007), es centra en el desenvolupament d'interfícies obertes entre els plànols de control i dades. Finalment, la tercera etapa (2007-2010) API's i OpenFlow, fa possible el primer desenvolupament mitjançant interfícies obertes de la separació dels plànols d'una forma pràctica i escalable.

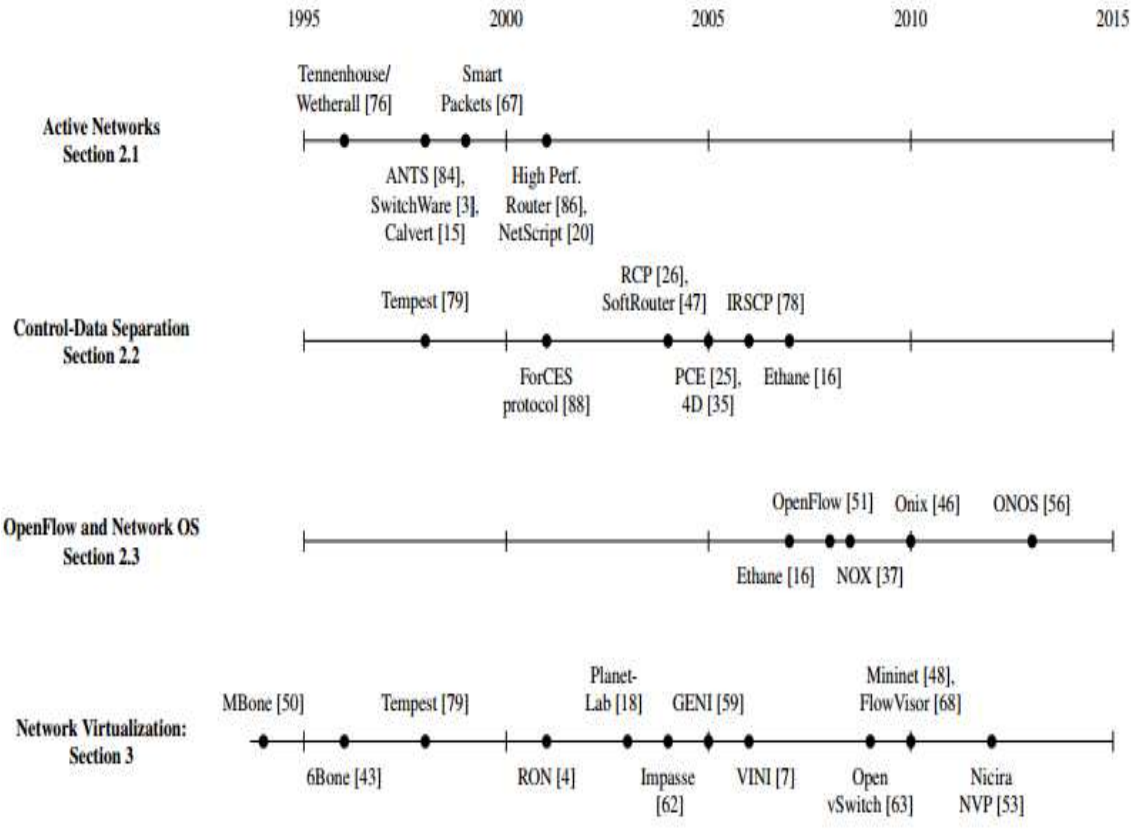



Figura 1. Cronologia als avanços cap a la virtualització de xarxes.

https://www.cs.princeton.edu/courses/archive/fall13/cos597E/papers/sdnhistory.pdf

## 2.1. Xarxes actives.

La gran explosió d'Internet, l'aparició de les primeres aplicacions de transferència d'arxius així com el correu electrònic, van generar de seguida expectació entre els científics desitjosos de treballar i implementar noves idees per millorar els serveis de xarxa. Van crear nous protocols de xarxa en entorns petits simulant el comportament de xarxes més grans. Aquestes millores, es van consolidar i estandarditzar a la Internet Engineering Task Force (IETF), tot i que el procés de normalització va ser relativament lent i frustrat en molts moments.

Alguns investigadors, van decidir seguir un estudi alternatiu del control de la xarxa basat en la facilitat amb la qual es configura un sol PC. Les xarxes convencionals no eren programables, en canvi, les xarxes actives permetien accedir al control de la xarxa mitjançant API's, deixant disponibles recursos com els processos, emmagatzematge i cua de paquets en nodes individuals, els quals permetien assignar funcions personalitzades i aplicar-les als paquets que passaven per aquest node.

Les xarxes actives van aportar alternatives radicals als serveis prestats per la pila tradicional d'Internet a través d'IP o per el mode de transferència asíncron. En certa manera, la xarxes actives van suposar el punt de partida cap a una nova forma de veure l'arquitectura de xarxes, seguida per programes com el GENI (Global Environment for Network Innovations) i el NSF FIND (Future Internet Design) als Estats Units, i EU FIRE (Future Internet Research and Experimentation Initiative) a la Unió Europea.

Es creen dos mecanismes de codi de control en els nodes, aquells que incorporen la informació de control dins dels paquets de dades, 'in-band' o model d'encapsulació, i els que utilitzen missatges expressos de control, 'out-of-band' o model router/switch programable. Tots dos mecanismes han deixat un llegat durador, però el model d'encapsulació s'associa estretament a les xarxes actives. L'encapsulació incorporava noves funcionalitats a la capa de dades incrustant codi en els paquets o trames, informació que més tard era tractada pels routers.

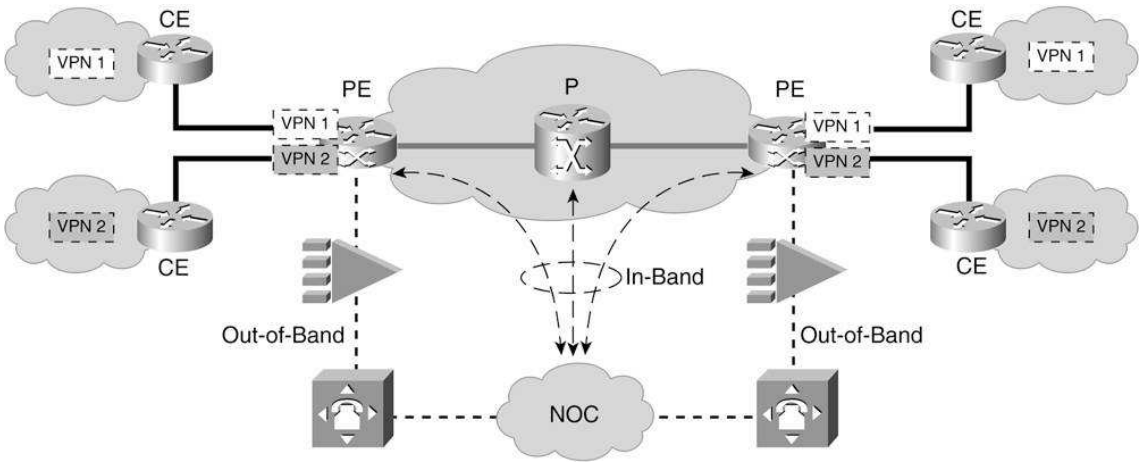



Figura 2 In band versus out of band

http://etutorials.org/Networking/MPLS+VPN+security/Part+I+MPLS+VPN+and+Security+Fundamentals/Chapter+1.+MPLS+VPN+Security+An+Overview/Fundamentals+of+MPLS+VPNs/

Les xarxes actives reduïen el cost en computació i facilitaven l'augment de processament a la xarxa. Es van crear programes avançats com Java i es va crear la tecnologia que va derivar cap a les màquines virtuals. Agències com la U.S. Defense Advanced Research Projects Agency (DARPA) van crear un projecte particular de xarxes actives des de mitjans dels 90 fins a principi del 2000, encara que no tot el treball de recerca ho van dur a terme ells, es van crear projectes paral·lels amb la fi, això sí, d'avançar tots en la mateixa direcció.

La motivació que va existir al seu moment per a la creació de les xarxes actives és comparable a l'existent avui dia per a crear SDN. La frustració per part dels proveïdors de serveis de xarxes pel que fa als terminis per desenvolupar nous productes, l'interès de tercers en el valor afegit que suposa, o bé la inquietud dels investigadors per experimentar amb una nova plataforma són alguns exemples d'aquesta motivació.

Les xarxes actives, van oferir tres contribucions remarcables relacionades amb SDN:

- Funcions programables a la xarxa que reduïen les barreres cap a la innovació, tot i que SDN es va centrar més en la programació en el pla de control, mentre que les xarxes actives ho van fer en el pla de dades, al igual que actualment realitza l'arquitectura NFV.

- Virtualització de xarxes, que va cobrir la necessitat de recolzar l'experimentació amb múltiples models de programació. Els components clau d'aquesta plataforma són:

    - Un sistema operatiu de nodes compartit (NodeOS), el qual s'encarrega de gestionar els recursos compartits.
    - Un conjunt d'entorns d'execució (EE's), cadascun dels quals defineix una màquina virtual per a les operacions de paquets.
    - Un conjunt d'aplicacions actives (AA's), els quals treballen dins d'un entorn d'execució donat per proporcionar el servei punt a punt.

- Visió d'una arquitectura unificada en diferents aparells de xarxa (middleboxes). Encara que mai es va arribar a realitzar completament dins del programa de les xarxes actives, sense que es va establir la necessitat de la seva existència.

A la següent figura es pot observar el funcionament d'un router tradicional:

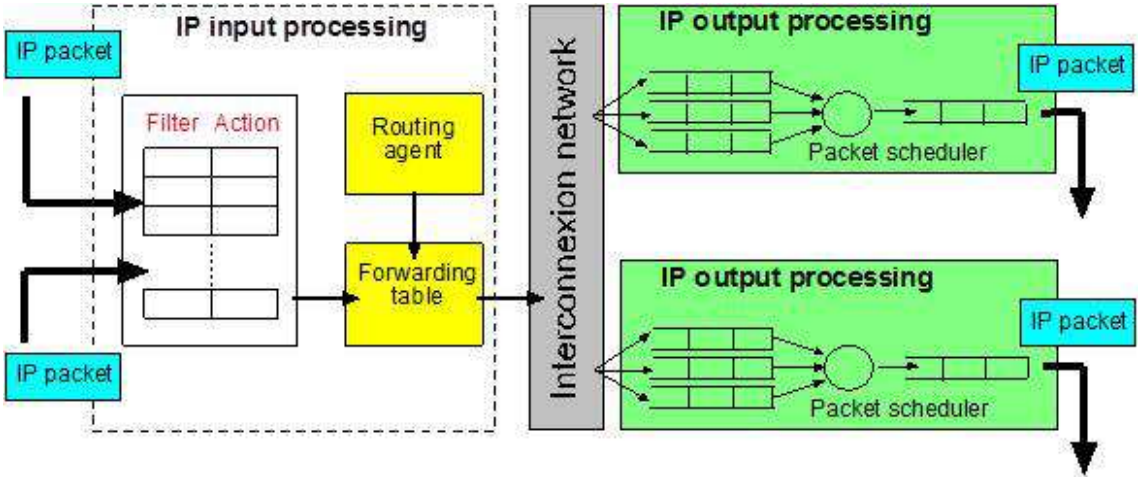



Figura 3 Router tradicional

https://www.google.es/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=0ahUKEwj0886-ueDJAhUDQBQKHdZEAuAQFggiMAA&url=http%3A%2F%2Fweb.univ-pau.fr%2F~cpham%2FPaper%2FTalkMIM2.ppt&usg=AFQjCNEKubI_KokCHNqWzRlBme9YydNXeQ&sig2=tjkFtNu-8nloiS6zk8uCGg

Separa la injecció de programes del processament de paquets, a les trames s'afegeix informació dels usuaris que permet dur a terme el control i la gestió del tràfic.

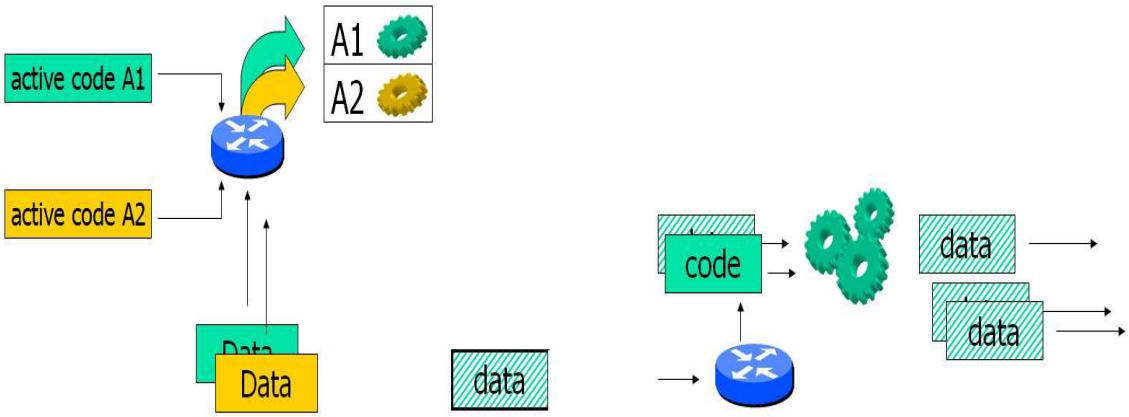



Figura 4 Informació de control a les trames

A la següent figura es pot veure el procés actualitzat resultant d'un router de xarxa activa on 'AL' és la capa activa.

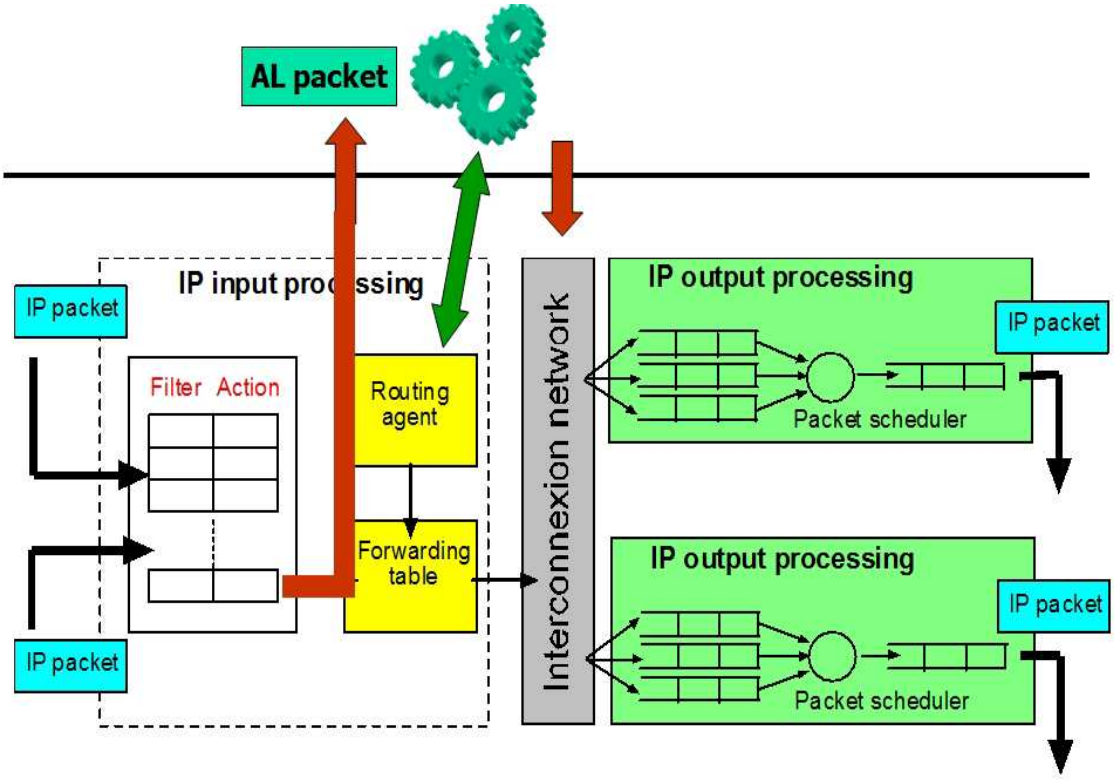



Figura 5 Router xarxa activa

## 2.2. Separació dels plànols de control i de dades

A la primera dècada del segle XXI, l'augment en el volum del tràfic i un major èmfasi en la fiabilitat de la xarxa, la previsibilitat i el rendiment, va portar als operadors de xarxa a la recerca de millors enfocaments per a certes funcions de gestió, com el control de rutes utilitzades per lliurar el tràfic (una pràctica comunament coneguda com l'enginyeria del tràfic).

Els mitjans per a la realització de l'enginyeria del tràfic, fent servir protocols d'encaminament convencionals, eren primitius. Els operadors, frustrats, van sol·licitar ajuda a uns investigadors que treballaven amb ells regularment. Aquests investigadors van descobrir que els routers i switchos convencionals tenien una estreta integració entre els plànols de control i de dades. Aquest acoblament realitzava diverses tasques de gestió a la xarxa, com la depuració de problemes de configuració i la de predir o controlar el comportament de l'encaminament, una tasca realment complicada. Per fer front a aquests desafiaments, es va optar per centrar els esforços a separar els plànols de dades i control.

A mesura que creixia Internet en la dècada dels 90, les velocitats d'enllaç en xarxes troncals van augmentar considerablement, els principals proveïdors d'equips van implementar la lògica de la reenvio de paquets directament en el maquinari, separat del pla de control. A més, els proveïdors de serveis d'Internet (ISP) estaven lluitant per gestionar la creixent grandària, l'abast de les seves xarxes i les demandes d'una major fiabilitat i nous serveis (com les xarxes privades virtuals). Paral·lelament a aquestes dues tendències, els continus i ràpids avanços en les plataformes informàtiques de productes bàsics, va significar que els servidors sovint tenien molts més recursos de memòria i de processament que el processador del pla de control d'un router implementat només un o dos anys abans.

Aquestes tendències catalitzen dues innovacions:

- Una interfície oberta entre els Plànols de control i dades, al igual que Forwarding and Control Element Separation (ForCES) interfície estandarditzada pel IETF, o bé la Netlink de Linux.

- Control centralitzat de la xarxa, com a Routing Control Platform (RCP) i arquitectures SoftRouter, o bé Path Computation Element (PCE), protocol de la IETF.

Aquestes innovacions van ser impulsades per les demandes de la indústria tecnològica per a la gestió de l'encaminament dins d'una xarxa ISP. En comparació amb les xarxes actives, aquestes tendències es van centrar en els problemes que presentava la gestió de la xarxa, especialment en els administradors, en la innovació, a la programació del pla control i en la visió de tota la xarxa.

Aplicacions de gestió de xarxa com la Intelligent Route Service Control Point (IRSCP), incloïen les millors rutes per al tràfic generat, minimitzant d'aquesta forma les interrupcions generades a l'encaminament, donant als usuaris més control sobre el mateix. Les primeres proves es van realitzar sobre un únic proveïdor ISP, a mesura que es va anar avançant cap a proves més obertes. Els avanços a la tecnologia de servidor, significava que un sol servidor de productes bàsics podria emmagatzemar tot l'estat de l'encaminament i calcular totes les seves decisions per a una gran xarxa ISP.

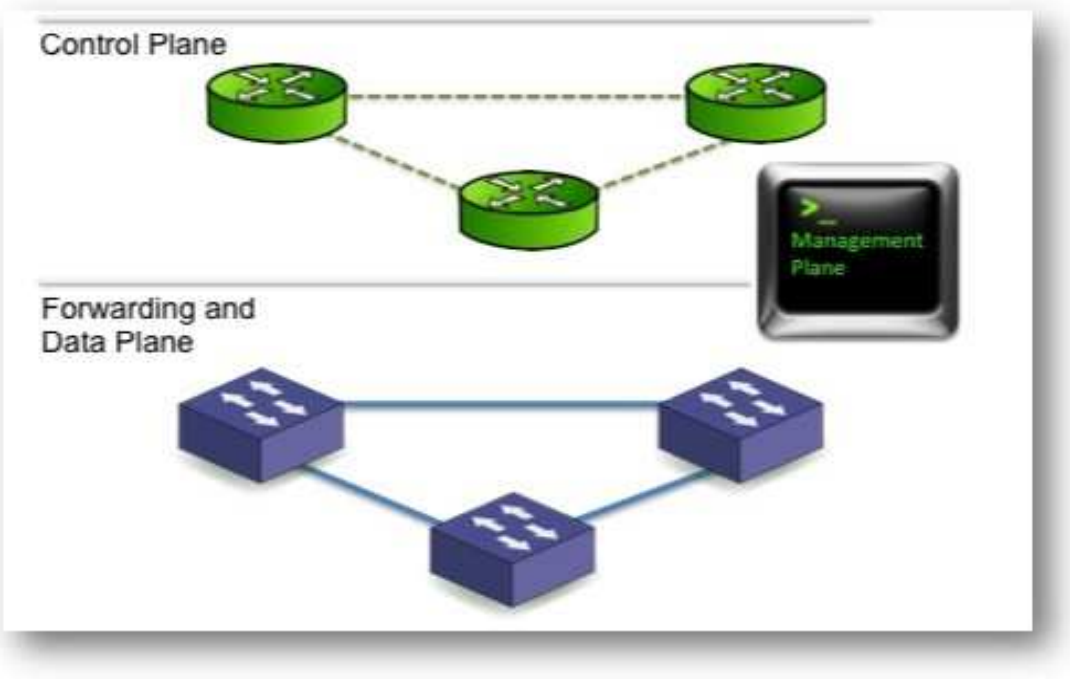



Figura 6 Separació dels plànols de control i dades

http://networkstatic.net/the-control-plane-data-plane-and-forwarding-plane-in-networks/

Els esforços realitzats per a la separació dels Plànols de dades i control han servit per recollir diversos conceptes que s'han implementat més tard a SDN:

- El control centralitzat fa servir una interfície oberta cap al pla de dades. Des de la IETF es va proposar una interfície de codi obert, per a d'aquesta forma permetre la seva constant innovació. El SoftRouter utilitza una API de ForCES per permetre a un controlador separat, instal·lar una taula de reenvio en el pla de dades, donant lloc a la completa eliminació de control dels routers. Lamentablement, ForCES no va ser incorporat pels principals fabricants de routers, esperant que els nous API's apareguessin, RCP va utilitzar un protocol estàndard existent de pla de control (Border Gateway Protocol) per instal·lar les taules d'adreces en els routers.

- Els controladors centralitzats havien de ser replicats per evitar fallades, però aquesta acció podia ocasionar estats inconsistents en les rèpliques. Totes les rèpliques calculaven les mateixes rutes, per tant, generava redundància d'informació. Per aconseguir una millor escalabilitat, cada instància de controlador podria ser responsable d'una part de la topologia de xarxa, intercanviant informació d'encaminament entre elles per garantir decisions coherents. SDN aplicaria anys més tard aquest sistema de controladors distribuïts, requerint solucions més sofisticades per al seu control.

Hi havia molt escepticisme entre els operadors i investigadors, el fet de separar els Plànols de control i dades representava per a molts d'ells una mala idea ja que si un controlador fallava, no veien la forma de com podia continuar operant la xarxa. En el control centralitzat cada router tenia coneixement local de les rutes, en canvi al sistema de xarxes tradicionals amb el temps, cada router tenia una visió general de la xarxa. Posteriorment, es va poder observar com fins i tot les solucions d'encaminament distribuït tradicionals violaven aquestes normes, siguin els casos dels protocols OSPF o BGP.

Els principals proveïdors d'equips observaven escassos beneficis en adoptar les API's estandarditzades, amb el temps i malgrat els grans esforços, l'adopció generalitzada

continua sent difícil. Amb la finalitat d'ampliar la visió relativa a la separació dels Plans, es va optar per quatre capes principals:

- Pla de dades: per als paquets de processament, basat en regles configurables.
- Pla de descobriment: recollida de la topologia i mesuraments de tràfic.
- Pla de difusió: per a la instal·lació de regles de processament de paquets.
- Pla de decisió: controladors centralitzats que transformen els objectius de nivell de xarxa en un estat d'enviament de paquets.

Diversos grups van procedir a dissenyar i construir sistemes que apliquen aquest enfocament d'alt nivell per a noves àrees d'aplicació, en particular el projecte Ethane, que redueix els switchos, fluint taules de controladors basades en polítiques de seguretat d'alt nivell. Aquest projecte, va asseure les bases per a la creació de OpenFlow, convertint-se en la seva base d'API's original.

## 2.3. Aparició de Openflow i el seu Network OSes

A mitjans de la primera dècada del 2000, investigadors i organismes van centrar el seu interès en l'escalabilitat de la xarxa, un grup d'investigadors de Stanford va crear el programa Clean Slate, el qual es va centrar en les xarxes de campus. A finals de la primera dècada del 2000, el grup OpenFlow en Stanford va implementar bancs de proves a través de molts campus i va demostrar les capacitats del protocol, tant en una sola xarxa del campus com en més d'una xarxa troncal d'àrea extensa que abastava diversos campus. Abans de l'aparició de OpenFlow, SDN es debatia entre la visió de xarxes totalment programables i el seu desplegament al món real. OpenFlow va aconseguir un equilibri entre aquests dos objectius en habilitar més funcions que els controladors anteriors i aplicar-les als switchos. Plataformes de controladors com NOX, van fer ús de la API de OpenFlow, permetent la creació de moltes aplicacions noves de control.

L'èxit de OpenFlow radica principalment en la seva acceptació a la indústria, proveïdors d'equips, dissenyadors de chipsets, operadors i investigadors de xarxes. La

decisió d'obrir el chipset proporciona l'impuls necessari per a una indústria que ja estava demanant a crits un major control sobre els dispositius de xarxa, permetent a altres empreses fabricar switchos sense incórrer en el cost substancial de dissenyar i fabricar el seu propi maquinari de pla de dades. OpenFlow va començar a afermar-se en altres àmbits, com les xarxes de centres de dades, on hi havia una necessitat clara per gestionar el tràfic de xarxa a grans escales. Aviat es va poder observar una forta reducció en els costos, pel que diversos fabricants van apostar per ell.

## 2.4. Virtualització de la xarxa

La virtualització de la Xarxa representa l'abstracció d'una xarxa que està desacoblada de l'equip físic subjacent. Network virtualization permet a múltiples xarxes virtuals que s'executin a través d'una infraestructura compartida, i cadascuna d'elles pot tenir una topologia molt més simple que la xarxa física subjacent. Per exemple, una xarxa d'àrea local virtual (VLAN) proporciona la il·lusió d'una sola LAN però abasta múltiples subxarxes físiques, i a més, diverses VLAN's es poden executar en el mateix conjunt de switches i routers. Tot i que la virtualització de la xarxa és conceptualment independent de SDN (tal i com es pot apreciar a la següent figura), la relació entre aquestes dues tecnologies s'ha tornat molt més propera en els darrers anys.

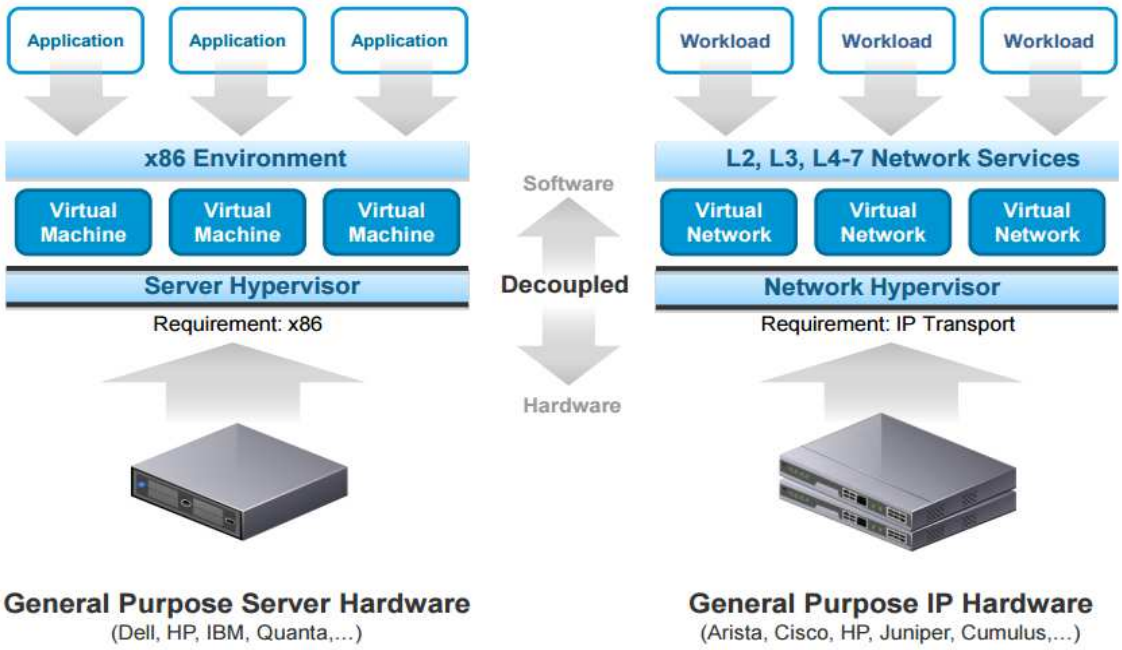



Figura 7 Virtualització != SDN

http://www.networkworld.com/article/2224526/cisco-subnet/breaking-down-vmware-s-talk-at-the-open-networking-summit.html

Tant la virtualització de xarxes com SDN comparteixen similituds, les xarxes programables compten amb mecanismes per compartir la infraestructura i el suport a les topologies de xarxa lògiques que difereixen de la xarxa física, els quals, són principis en la virtualització de la xarxa.

## 2.5. La virtualització de la xarxa abans de SDN.

Durant molts anys, els equips de xarxa han suportat la creació de xarxes virtuals, en forma de les VLAN o VPN.

No obstant això, només els administradors de xarxa poden crear aquestes xarxes virtuals, a més, aquestes es limiten a l'execució dels protocols de xarxa existents. Com a tal, el desplegament de les noves tecnologies de forma incremental va resultar molt difícil. Al seu lloc, els investigadors i els professionals van recórrer a xarxes superposades en execució, on un petit conjunt de nodes actualitzats utilitzaven túnels per formar la seva pròpia topologia en la part superior d'una xarxa heretada. En una xarxa superposada, els nodes actualitzats executen el seu propi protocol del pla de control i de tràfic de dades, directes entre si, encapsulant paquets i enviant-los a través de la xarxa heretada on són desencapsulats a l'altre extrem.

Aquestes primeres xarxes superposades consistien en nodes dedicats que executaven protocols especials per aportar millores a la infraestructura de xarxa. El concepte de xarxes superposades aviat es va expandir per incloure a qualsevol ordinador que instal·li i executi una aplicació especial, esperonat per l'èxit de les primeres aplicacions peer-to-peer per compartir arxius. A més de la recerca significativa en els protocols peer-to-peer, la comunitat de recerca de xarxes va reviure la recerca sobre l'ús de xarxes superposades com una forma de millorar la infraestructura de la xarxa, tals com el treball en Resilient Overlay Networks, on una petita col·lecció de hosts que es comuniquen formen un superposició que reacciona ràpidament a les fallades de la xarxa i problemes de rendiment.

# CAPÍTOL 3

# 3. ARQUITECTURES

Actualment es poden distingir dos arquitectures de xarxes virtualitzades, NFV i SDN, desplegades per a incrementar les eficiències i simplificar el desplegament de nous serveis.

## 3.1. SDN

Les xarxes tradicionals han evolucionat al llarg de la seva història des que es van crear, responent a una arquitectura jeràrquica client/servidor. En l'actualitat, aquesta evolució no segueix un camí paral·lel amb el progrés d'altres tecnologies, no resulten prou àgils ni es poden reprogramar. L'arquitectura client/servidor tendeix a utilitzar-se cada vegada menys, la comunicació entre servidors (bases de dades, aplicacions, business inteligence, service to service, etc.) acaparen la majoria del tràfic generat a les xarxes.

L'aparició del "Cloud computing" requereix d'una major flexibilitat a la xarxa, la informació viatja sobre sistemes distribuïts replicant les dades en màquines virtuals que al seu torn poden canviar d'allotjament físic. A més, la tecnologia Big Data, conjuntament amb els serveis business inteligence, requereixen d'una alta escalabilitat a causa de la immensa quantitat de dades que es mouen entre infinitat de processos llançats per múltiples servidors.

Tot això sumat a la forta tendència de globalització tecnològica en la qual l'augment de dispositius permet connectar-se des de qualsevol part del món, fa de les xarxes tradicionals un coll d'ampolla. A més, cada xarxa dins d'Internet té la seva pròpia arquitectura així com els seus criteris de commutació, això provoca que cada xarxa tingui la seva pròpia definició d'una forma tancada i no fa que sigui escalable.

Això fa que l'arquitectura de xarxa es modeli cap a un nou format que li aporti més escalabilitat, agilitat, automatització, control i flexibilitat. En resposta a tot aquest tipus de manques la indústria ha creat l'arquitectura Software Defined Networking (SDN).

### 3.1.2. Definició de SDN

En certa manera, SDN reprèn idees de les xarxes de telefonia, que utilitzen una clara separació dels plànols de control i de dades per simplificar la gestió de xarxes i el desplegament de nous serveis.

A diferència de les xarxes tradicionals, el control de les mateixes passa a estar en mans d'un controlador de software. Aquesta arquitectura permet separar el pla de control del pla de dades aconseguint d'aquesta forma xarxes més flexibles, programables, àgils i automatitzades. En separar tots dos plànols es prescindeix de la gestió que realitza el maquinari (routers, switches) així com tot el que comporta el seu ús (costos, personal, etc.).

En una xarxa definida per software, a diferència de la xarxa tradicional on cada paquet o trama és tractat en un dispositiu (commutador o switch) de la mateixa manera independentment del tipus de paquet que sigui, es pot fer d'una forma centralitzada sense la necessitat de recórrer a cadascun dels commutadors existents en el camí. Això resulta especialment interessant en una xarxa amb una gran càrrega de dades orientades d'est a oest, el que la fa més flexible, àgil i eficient. La programació ha passat de ser de node a node, a una manera de programació centralitzada mitjançant software.

Segons la ONF, l'arquitectura SDN és:

- Flexible: Al estar els plànols de control i dades separats, es permet realitzar canvis en les polítiques de funcionament sense afectar als paquets o trames.
- Àgil: L'abstracció del control d'envio permet als administradors ajustar dinàmicament tot el flux del tràfic de la xarxa, satisfent les necessitats canviants en cada moment.
- Control unificat de l'estructura de xarxa: La intel·ligència de la xarxa està centralitzada en controladors SDN basats en software que mantenen una visió global de la xarxa, que apareix per a les aplicacions i les polítiques de les màquines com switch lògics.
- Configurada mitjançant programació: SDN permet als administradors de xarxa configurar, administrar, assegurar, i optimitzar els recursos de xarxa ràpidament mitjançant programes SDN dinàmics, automatitzats, que poden escriure ells mateixos perquè els programes no depenen d'un software propietari.

- Basada en estàndards oberts i neutrals: Quan s'implementa a través d'estàndards oberts, SDN simplifica l'operació i el disseny de la xarxa perquè les instruccions són proporcionades pels controladors SDN en lloc de per múltiples protocols i dispositius específics del proveïdor.
- Solució d'errors funcional: gràcies al control centralitzat els errors són més fàcils de detectar i per tant solucionar.
- Completament programable: el control de la xarxa es pot programar directament ja que està desacoblada de les funcions de reexpedició. SDN permet als administradors de xarxa configurar, administrar, protegir i optimitzar els recursos de la xarxa molt ràpidament a través de programes dinàmics i automatitzats, que poden escriure ells mateixos ja o que els programes no depenen del software propietari de cap dispositiu.
- Simplicitat amb Opensource: en implementar-se a través d'estàndards oberts, SDN simplifica el disseny de la xarxa i la seva operativa, ja que les instruccions són proporcionades per controladors SDN en lloc de per múltiples dispositius, proveïdors específics i protocols.

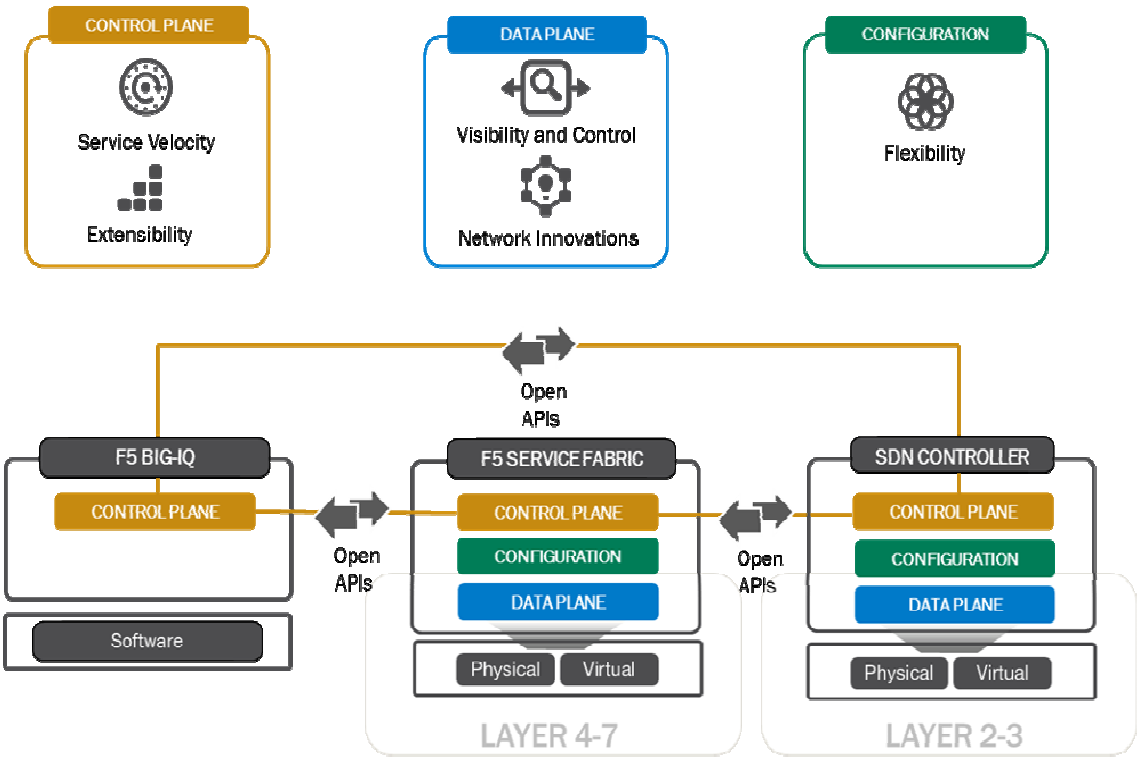



Figura 8 Capes de influencia de SDN / F5

http://sdn.sys-con.com/node/2918106

### 3.1.3. Avantatges de SDN

El benefici clau de SDN radica en la creació d'una xarxa amb switchos i routers virtuals. Per aquest motiu, els nous serveis o els canvis que es facin en ells es poden introduir de forma ràpida i amb uns costos menors. Un altre avantatge és que l'operador de xarxa no ha d'aprofundir fins al nivell de dispositiu per a realitzar aquets canvis, lo que simplifica la seva gestió. A més, el control centralitzat i l'automatització generalitzada milloren l'eficiència i seguretat de la xarxa.

La velocitat i agilitat en la provisió de serveis és un altre punt a favor, resulta relativament fàcil configurar xarxes SDN mitjançant la creació de màquines virtuals (VM).

En quan a la flexibilitat de la xarxa i gestió integral, SDN permet eludir les limitacions que planteja el Protocol simple d'administració de xarxes (SNMP) i experimentar amb noves configuracions. Amb la flexibilitat addicional proporcionada per l'entorn, les organitzacions poden crear els seus propis serveis de xarxa fent servir eines de desenvolupament estàndard. Hi ha una millor resposta en seguretat, els entorns VM porten problemes de seguretat afegits a la xarxa. SDN, d'altra banda pot proporcionar seguretat en el nivell més baix per a aplicacions, terminals i dispositius Bring Your Own Device (BYOD).

A més, la seva usabilitat gràcies a la programació de la xarxa, permet que les empreses puguin desenvolupar noves aplicacions que poden millorar encara més la manejabilitat, la integració o la comunicació. L'organització no depèn de l'ajuda externa per abordar aquests aspectes. La centralització i automatització del control de polítiques, permet entre d'altres la inspecció en profunditat de paquets (DPI), una tecnologia de filtrat de paquets de xarxa que controla l'incompliment de protocol, virus, spam i altres inconsistències d'un paquet. Permet funcions d'automatització de xarxa avançada, compliment i seguretat de la política a realitzar. En proporcionar dades detallades sobre el tràfic de xarxa al controlador SDN, la xarxa es pot considerar com un únic recurs en la seva totalitat, en lloc de com un grup de diversos dispositius com ara switchos, etc. La combinació de DPI i SDN pot proporcionar un control centralitzat de la política i l'automatització de les xarxes.

Per finalitzar, la càrrega de treball virtual que proporciona SDN permet als administradors establir múltiples xarxes en lloc d'haver de configurar directament cada dispositiu de xarxa. Programes de control d'alt nivell poden destinar ample de banda per al trànsit de dades directes que condueixen a un millor rendiment. La superposició de les xarxes virtuals sobre les físiques pot millorar la capacitat d'adaptació i flexibilitat de càrrega de treball, la qual cosa porta a una xarxa àgil, escalable i receptiva.

### 3.1.4. Arquitectura SDN.

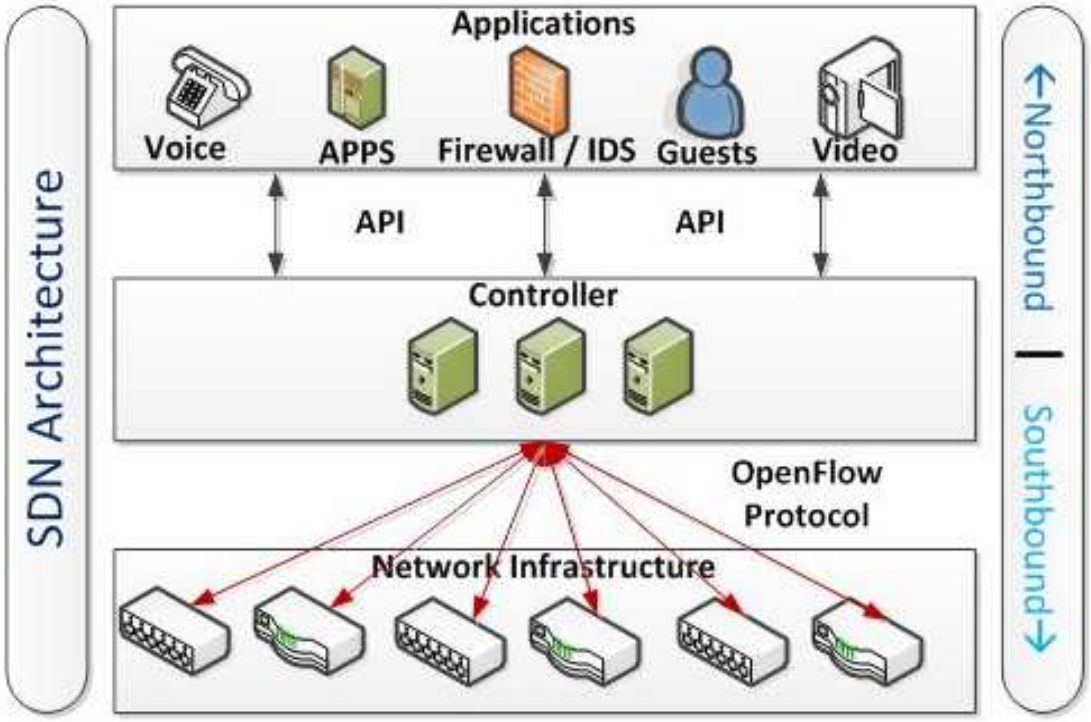



Figura 9 Arquitectura SDN

A la figura anterior es poden observar les següents capes:

- Capa d'infraestructura, on es situen els dispositius (switches, routers).
- Capa de control, estableix el software encarregat de gestionar els diferents serveis de la xarxa. Aquesta capa es comunica amb l'anterior mitjançant la southbound API, incloent protocols de comunicació com OpenFlow, el protocol

extensible de missatgeria i comunicació de presència (XMPP) i el protocol de configuració de xarxa.

- Capa d'aplicacions, ubicació de les aplicacions relatives als usuaris finals. Es comunica amb la capa de control a través de la Northbound API, actualment no existeix cap basada en estàndards.

### 3.1.5. Funcionament de SDN

La primera característica fonamental de SDN és la separació del pla de dades del de control. El procés de reexpedició de paquets inclou característiques tals com l'adreça MAC, adreça IP i l'identificador de VLAN. La capa de transmissió a l'arribada dels paquets reenvia, descarta o replica aquestes trames. En el reenvio tradicional, el dispositiu determina el port de sortida correcte mitjançant una cerca a la seva taula d'adreces. Durant aquest procés, un paquet pot ser descartat ja sigui per desbordament del buffer o bé per haver superat el nombre d'intents de reenvio. En casos especials, paquets que requereixen d'un procés de manteniment del pla de control són dirigits cap al mateix. Un altre cas és el dels paquets multicast, que han de ser replicades còpies dels mateixos per diferents ports.

Els protocols, lògica i algorismes que s'utilitzen per programar la transmissió dels paquets es troben en el pla de control. Molts d'aquests protocols i algorismes requereixen un coneixement complet de la xarxa, el pla de control en aquest cas determina com s'han de programar les taules d'adreces al pla de dades. A les xarxes tradicionals cada dispositiu disposa del seu propi pla de control sent la seva tasca principal la d'executar l'encaminament o commutació de protocols de manera que les taules d'adreces en tots els dispositius romanguin sincronitzades.

Encara que aquests plànols tradicionalment han estat considerats lògicament separats en realitat comparteixen ús en els diferents dispositius d'Internet. En SDN el pla de control s'extreu dels dispositius feia un control centralitzat.

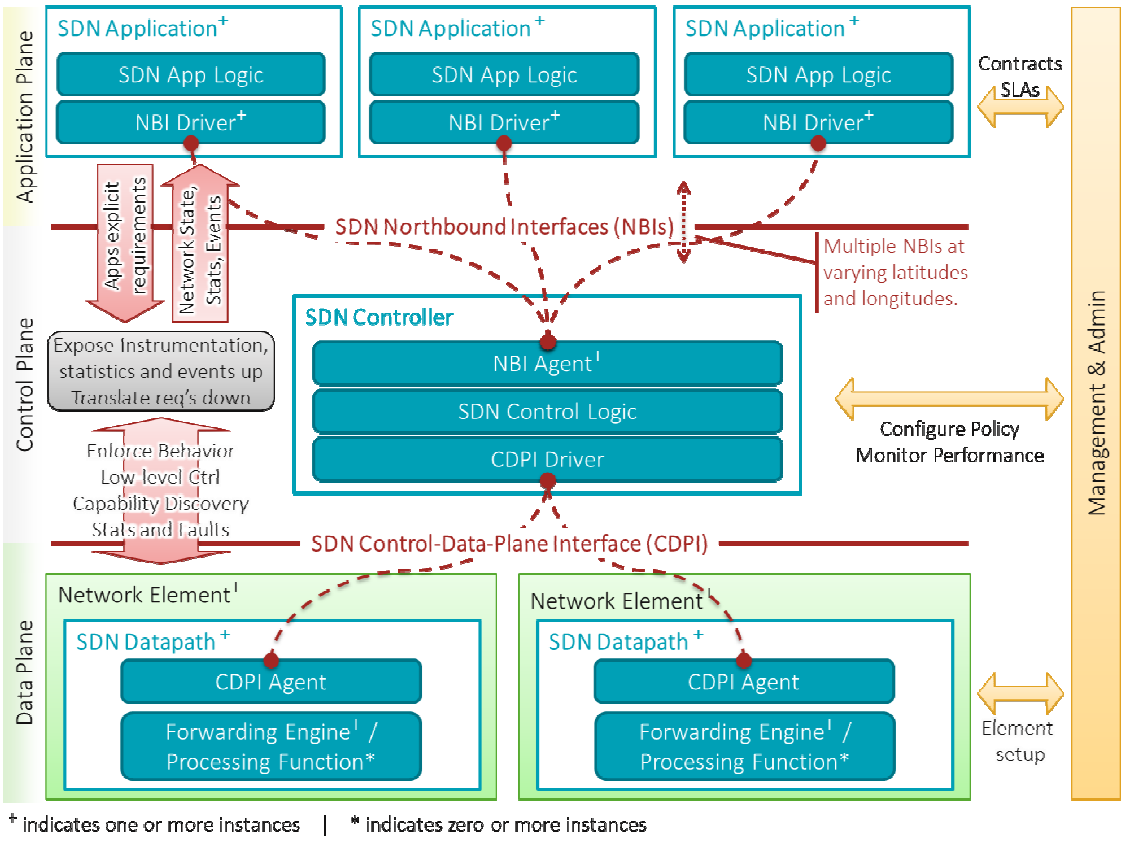



Figura 10 Funcionament de la Arquitectura de SDN

### 3.1.6. Mites de SDN

Un mite relatiu a SDN és que el primer paquet de cada flux de tràfic ha d'anar al controlador per a la seva manipulació. De fet, alguns sistemes com Ethane treballen d'aquesta manera, ja que van ser dissenyats per recolzar les polítiques de granularitat fina en xarxes petites. De fet, SDN en general i OpenFlow en particular, no imposen cap hipòtesi sobre la granularitat de les regles o si el controlador s'encarrega de tot el tràfic de dades. Algunes aplicacions SDN només responen a canvis en la topologia i les estadístiques de tràfic de granularitat gruixuda i regles d'actualització, amb poca freqüència en resposta a vincular fallades o congestió de la xarxa. Altres aplicacions poden enviar el primer paquet d'alguns més grans agregat de tràfic al controlador, però no un paquet des de cada connexió TCP o UDP.

Un segon mite que envolta SDN és que el controlador ha d'estar centralitzat físicament. De fet, Onix i Onos demostren que els controladors poden i han de ser distribuïts. Desplegaments d'àrea àmplia de SDN, com ara Google, tenen molts controladors repartits per tota la xarxa.

Finalment, un error molt comú és que SDN i OpenFlow són equivalents, de fet, OpenFlow és més que una (molt popular) exemplificació dels principis de SDN. Diferents APIs podrien ser utilitzats per controlar el comportament de reenvio de tota la xarxa, el treball previ que es va centrar a l'encaminament (usant BGP com a API) podria ser considerada com una exemplificació de SDN, i les arquitectures de diferents proveïdors (Cisco ONE i Junos SDK) representen altres instàncies de SDN que difereixen de OpenFlow.

### 3.1.7. Controladors SDN

El principal component d'una xarxa definida per software, tal i com es presenta, és el controlador de xarxes basat en software, també anomenat sistema operatiu de xarxa. El controlador és el que defineix la naturalesa d'aquest paradigma. És el component responsable de concentrar la comunicació amb tots els elements de xarxa programables, proporcionant una visió unificada de la mateixa. Els controladors utilitzats per l'arquitectura SDN es divideixen en dos grups, els comercials i els Opensource, els primers a aparèixer.

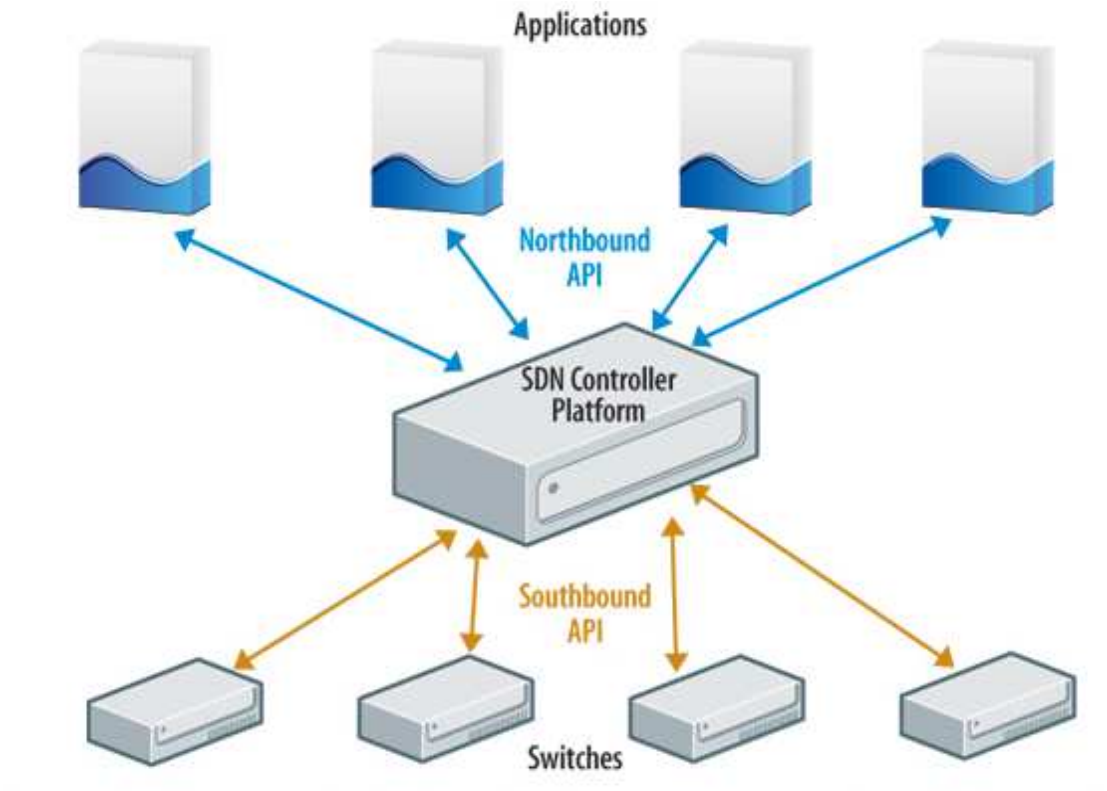



Figura 11 Esquema gràfic d'un controlador SDN

http://www.techweb.com/networking/240008034/sdn-your-next-network.html

Dins l'extensa gamma dels comercials podem remarcar els més importants:

- Application Policy Infrastructure Controller (APIC): és l'encarregat d'automatitzar i administrar la Infraestructura centrada en aplicacions (ACI) actuant com a únic punt de control. Cisco APIC ofereix accés centralitzat a tota la informació d'estructura, aplica polítiques i funcions de xarxa de forma automatitzada i optimitza el cicle de vida de les aplicacions en quan a escala i rendiment. Utilitza una API i estàndards oberts per resultar més accessible de cara a proveïdors de serveis. Disposa d'una sòlida implementació de seguretat.

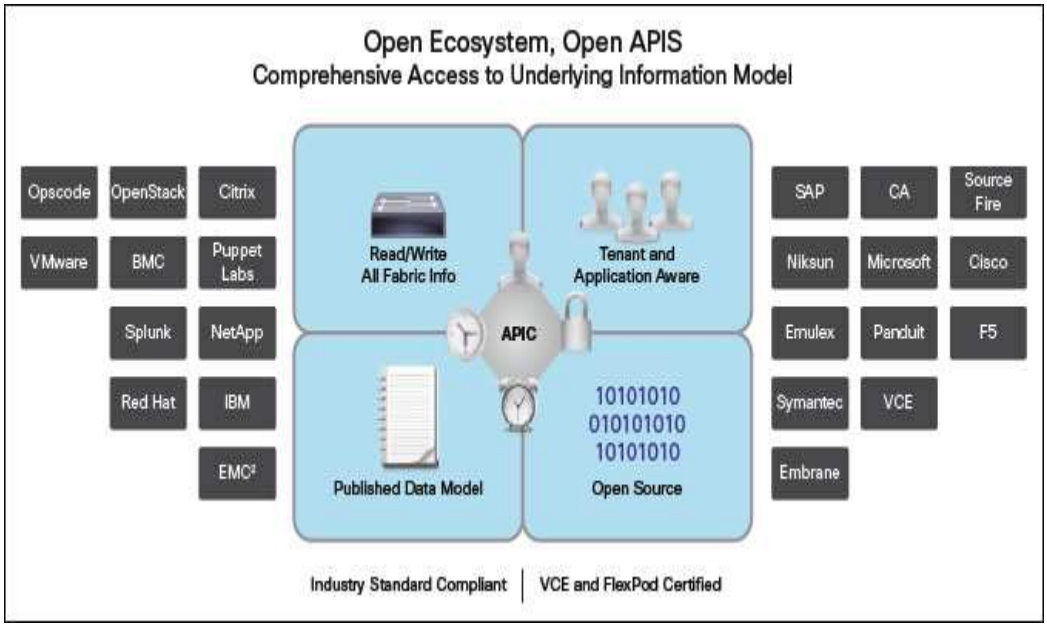



Figura 12 Serveis APIC CISCO
http://www.bradreese.com/blog/11-21-2013.htm

- NEC ProgrammableFlow PF6800 Controller: basat en el protocol Openflow, automatitza i simplifica l'administració i gestió de la xarxa mitjançant una interfície programable. Incorpora la tecnologia de Nec per crear xarxes aïllades amb diversos inquilins (VTN) oferint polítiques de xarxa i seguretat de forma independent a cadascuna d'elles.

- HP Virtual Application Networks (VAN): mitjançant el protocol Openflow, fa servir la intel·ligència proporcionada pels servidors, l'emmagatzematge i maquinari de xarxa per automatitzar l'aprovisionament i les tasques de configuració, al mateix temps que s'optimitza el rendiment. Aquest controlador agilitza el desplegament d'aplicacions mitjançant un conjunt de plantilles de connexió, inclouen paràmetres i polítiques predeterminades per a màquines

virtuals. Ofereix entre altres, automatització de les polítiques de xarxa, mètodes de seguretat basats en autenticació i autoritzacions encriptades, interfície programable, etc.

- NSX de VMware: crea i controla xarxes virtuals, manté la informació de les màquines virtuals, hosts, switches lògics i VXLANs. Reprodueix serveis de xarxa des de la capa dos fins a la capa set com ara Switching, Routing, Balanceig de càrrega, Firewall, QoS, etc.

- Nuage Networks Serveis Virtualizados (VSC): proporciona el control de la xarxa oferint una visualització total d'ella i de la topologia dels seus serveis per inquilí. Utilitza el protocol Openflow per gestionar les polítiques de xarxa mitjançant el control de màquines virtuals, aquest en operatives o siguin de nova creació.

Dins la gamma dels controladors Opensource, ressalten els següents:

- Beacon: es tracta d'un controlador estable, podent suportar llargs períodes de temps sense cap interrupció, ràpid i multiplataforma, ja que està desenvolupat en Java.

- Floodlight: com l'anterior, està desenvolupat en Java i s'executa sobre una màquina virtual JVM. Pot treballar tant amb switches físics com a virtuals, permet fer-ho amb dispositius sense Openflow pel que pot operar amb xarxes tradicionals.

- NOX: va ser el primer controlador utilitzat per Openflow, proporciona una plataforma que permet als administradors disposar de la gestió i control de la xarxa de forma centralitzada. Disposa a més, d'entrada i sortida ràpida (síncrona i asíncrona) controlant el tràfic en tot moment. Ofereix mètodes i baix nivell per poder interactuar amb la xarxa en el seu format més bàsic. Actualment ofereix suport exclusivament per a C++ mitjançant una API, tot i que existeix una versió més antiga que suporta Python. Aquesta versió segueix utilitzant-se ja que compta amb un rang d'aplicacions més ampli.

- ODL: OpenDaylight Project pretén convertir-se en una plataforma SDN comuna i oberta, mitjançant un framework que permeti accelerar els serveis de xarxa en plataformes virtuals.


Figura 13 Col·laboradors projecte OpenDaylight

- POX: s'ha desenvolupat amb la finalitat de reemplaçar a NOX, sobretot en recerca i docència, quan el rendiment no és un requisit fonamental. Executa el seu control de la xarxa utilitzant Python mitjançant una interfície denominada 'Pythonic'.

- Trema: és un framework per programar un controlador mitjançant els llenguatges C i Rubi, creant un fitxer de configuració que una vegada executat es comportarà com a tal. NEC ho fa servir pel seu ProgrammableFlow.

- Ryu: al igual que Trema, Ryu és un framework. Ofereix components de software mitjançant l'accés per API, fent-ho fàcil d'usar.

- Opencontrail: l'empresa Juniper va llançar al mercat alhora el software comercial Juniper Network Contrail i el de codi obert Opencontrail. Els principals beneficis d'aquest controlador radiquen en els proveïdors de serveis i centres de dades ja que exigeixen major automatització, flexibilitat i programabilitat alhora que una notable reducció en els costos de la seva infraestructura.

| | Beacon | Floodlight | NOX | POX | Trema | Ryu | ODL |
|---|---|---|---|---|---|---|---|
| Soporte OpenFlow | OF v1.0 | OF v1.0 | OF v1.0 | OF v1.0 | OF v1.3 | OF v1.0, v1.2, v1.3 y extensiones Nicira | OF v1.0 |
| Virtualización | Mininet y Open vSwitch | Mininet y Open vSwitch | Mininet y Open vSwitch | Mininet y Open vSwitch | Construcción de una herramienta virtual de simulación | Mininet y Open vSwitch | Mininet y Open vSwitch |
| Lenguaje de desarrollo | Java | Java | C++ | Python | Rudy/C | Python | Java |
| Provee REST API | No | Si | No | No | Si (Básica) | Si (Básica) | Si |
| Interfaz Gráfica | Web | Web | Python+, QT4 | Python+, QT4, Web | No | Web | Web |
| Soporte de plataformas | Linux, Mac OS, Windows y Android para moviles | Linux, Mac OS, Windows | Linux | Linux, Mac OS, Windows | Linux | Linux | Linux, Mac OS, Windows |
| Soporte de OpenStack | No | Si | No | No | Si | Si | Si |
| Multiprocesos | Si | Si | Si | No | Si | No | Si |
| Código Abierto | Si | Si | Si | Si | Si | Si | Si |
| Tiempo en el mercado | 4 años | 2 años | 6 años | 1 años | 2 años | 1 años | 5 meses |
| Documentación | Buena | Buena | Media | Pobre | Media | Media | Media |

Figura 14 Taula comparativa 2014 de característiques controladors Opensource

http://revistatelematica.cujae.edu.cu/index.php/tele/article/viewFile/164/153

### 3.1.10. A la recerca dels programes de control.

Malgrat l'entusiasme inicial que envolta SDN, val la pena reconèixer que SDN no és més que una eina que permet la innovació en quan al control de la xarxa. SDN no dicta com aquest control ha de ser dissenyat ni resol cap problema en particular. Els investigadors i els operadors de xarxes tenen ara una plataforma a la seva disposició per ajudar a resoldre problemes de llarga durada en la gestió de les seves xarxes i en el desplegament de nous serveis. En última instància, l'èxit i l'adopció de SDN depèn de si es pot utilitzar per resoldre problemes urgents en xarxes que eren difícils o impossibles de resoldre amb els protocols anteriors. SDN ja ha demostrat ser útil per resoldre problemes relacionats amb la xarxa de virtualització.

## 3.2. NFV

La virtualització de funcions de xarxes o Network functions virtualization (NFV) ofereix una nova forma de dissenyar, implementar i administrar serveis de xarxa. NFV dissocia funcions de la xarxa, tals com la traducció d'adreces de xarxa (NAT), firewalls, serveis de noms de domini (DNS) entre d'altres, dels dispositius de maquinari dedicat, perquè puguin executar-se mitjançant aplicacions de software allotjades en màquines virtuals (VM).

Està dissenyat per consolidar i lliurar els components de xarxa necessaris per donar suport a una infraestructura totalment virtualitzada. Utilitza tecnologies de virtualització de TI estàndard que s'executen en switches i maquinari d'emmagatzematge. Aquests dispositius virtuals apareixen i es comporten igual que els seus homòlegs físics a les xarxes a les quals serveixen, sense la necessitat de dispositius individuals per omplir les seves diverses funcions especialitzades. És aplicable a qualsevol procés del pla de dades o funció del pla de control, tant en infraestructures de xarxes cablejats com a les sense fils. Si té èxit, NFV disminuirà la quantitat de maquinari propietari que es necessita per engegar i operar els serveis de xarxa.

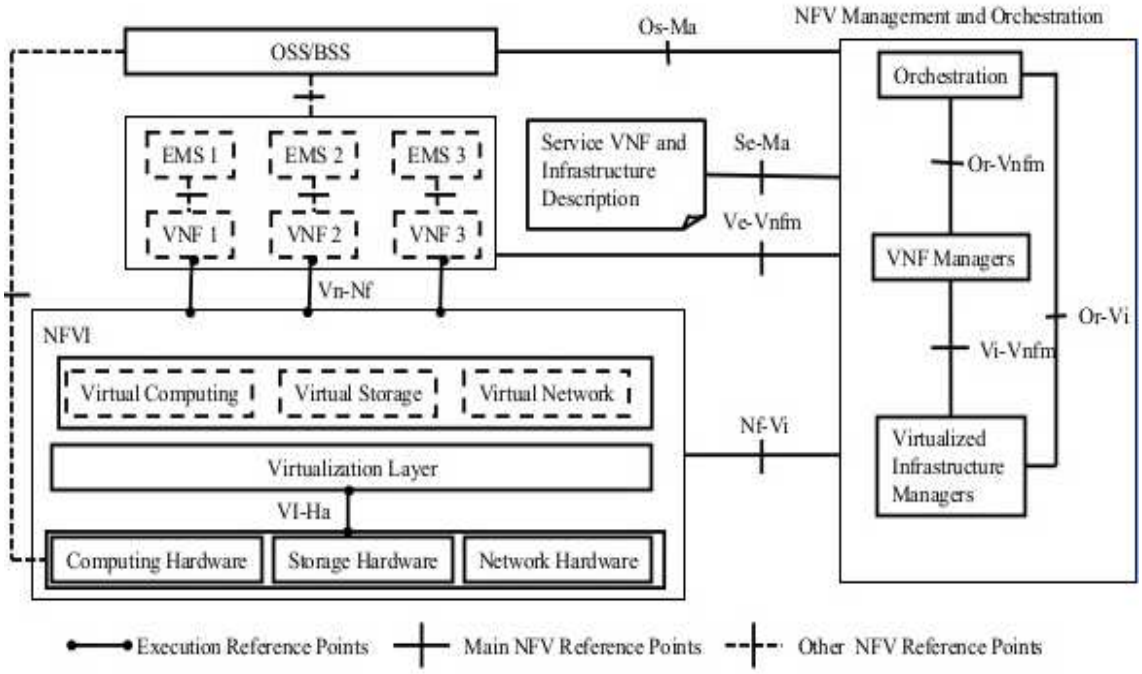



Figura 15 Arquitectura NFV

http://www.slideshare.net/rjain51/m-17nfv

NFV es va convertir en realitat quan els proveïdors de serveis van tractar d'accelerar el desplegament de nous serveis de xarxa amb la finalitat de fomentar els plànols d'ingressos i creixement. En adonar-se que els dispositius basats en maquinari limitaven la seva capacitat per aconseguir aquests objectius, van observar les tecnologies de virtualització de TI estàndard i van descobrir que NFV va ajudar a accelerar la innovació de serveis i aprovisionament. Això va conduir a la creació del Institut Europeu de Normes de Telecomunicacions (ETSI), que estableix els requisits bàsics i l'arquitectura de NFV. Aquest grup recentment va crear un projecte de codi obert, OPNFV.

A l'actualitat, l'enfocament principal de NFV al mercat actual es centra en els següents punts:

- Virtual Switching, o ports físics que enllacen amb els ports virtuals en servidors virtuals, on els routers virtuals empren IPsec virtualitzats i gateways SSL o VPN.
- Virtualized Network Appliances, quan les funcions de xarxa que poden emprar dispositius dedicats, en el seu lloc poden emprar dispositius virtualizats per a una extensa gamma de funcions especialitzades.

- Virtualized Network Services, que proporcionen supervisió i gestió dels serveis de xarxa basats en software, incloent l'anàlisi de tràfic, ‘monitoreo’ de xarxa, balanceig i qualitat de càrrega o classes de usabilitat de servei.

- Virtualized Applications, que ofereixen els marcs optimitzats de xarxa i les API’s per a aplicacions al núvol, amb la finalitat de recolzar a una població d'usuaris cada vegada més mòbil i basada en BYOD.

Tots aquests objectius estan ajudant als clients a reduir els seus costos en quan a capital i operacions es refereix, accelerant el temps de comercialització d'elements de servei i oferint solucions flexibles i àgils.

Tres dels principals beneficis de NFV són:

- Reduir costos: NFV permet reduir el CAPEX i OPEX a través de la reducció dels següents punts:

- Reducció dels costos d'equipament.
- Reducció del consum energètic.
- Reducció en la logística.
- Reducció en les operacions gràcies a la simplificació i automatització de les mateixes.
- Reducció de temps i recursos dedicats a la planificació i dimensionament de la xarxa.
- Reutilització de servidors.

- Simplificació del desplegament de nous serveis: NFV permet accelerar el desplegament de nous serveis o la modificació dels ja existents mitjançant la compartició de recursos, proporcionant una major flexibilitat a l'hora d'escalar. Els nous serveis es poden introduir de forma més controlada i amb menys riscos i cost.

- Afavoreix la innovació: NFV permet accelerar la velocitat d'innovació i la diferenciació de serveis, doncs afavoreix un ecosistema obert, ja que l'absència de maquinari propietari redueix les barreres d'entrada a nous proveïdors de software. A més és més fàcil donar entrada a més proveïdors, ja que les economies d'escala requerides per cobrir inversions en funcionalitats basades en maquinari, no són mai més aplicables al desenvolupament de funcions basat en software.

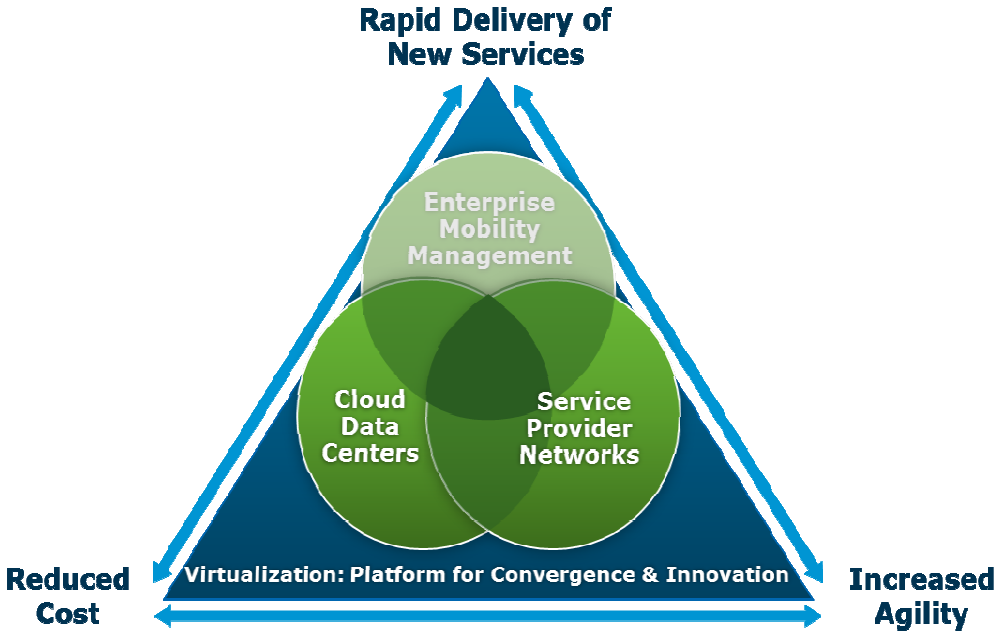



Figura 16 Gràfica d'avantatges de NFV

### 3.2.1. Comparativa amb SDN

Xarxes definides per software (SDN) i Network functions virtualization (NFV) són tots dos, enfocaments complementaris.

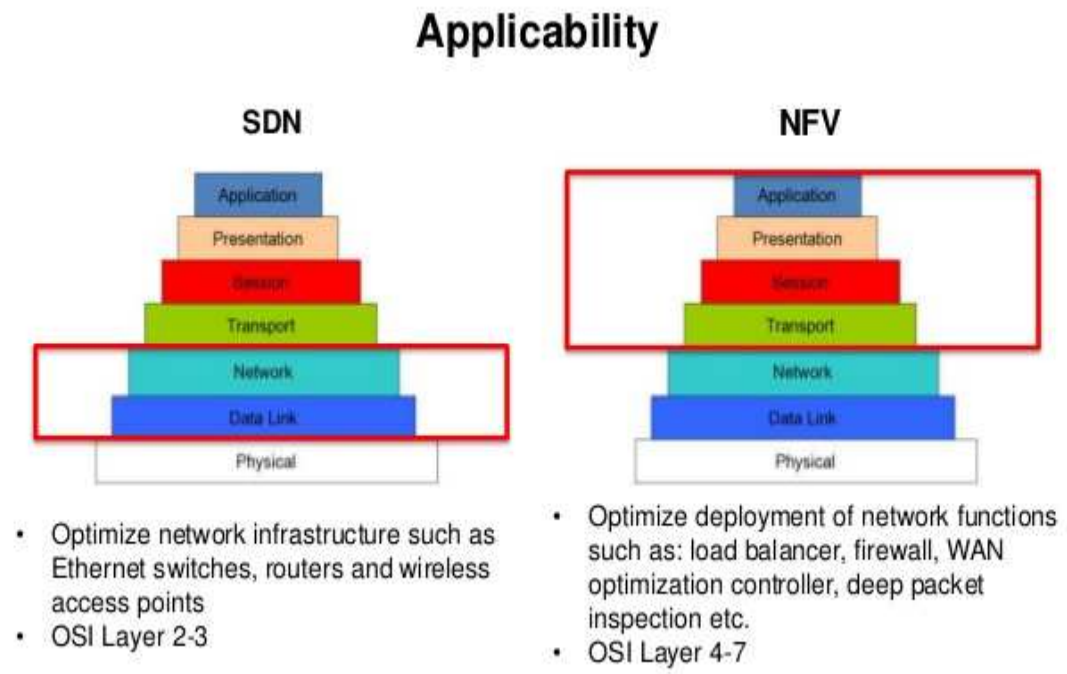



Figura 17 Comparativa Model OSI de SDN i NFV

Cadascun d'ells ofereix una nova forma de dissenyar implementar i administrar la xarxa i els seus serveis:

- SDN: separa el pla de control de la xarxa del pla de dades i proporciona una visió centralitzada de la xarxa distribuïda per a una organització i automatització de serveis de xarxa més eficient.

- NFV: es centra en l'optimització dels propis serveis de xarxa. NFV desacobla les funcions de xarxa, com DNS, emmagatzematge en caché, etc., dels dispositius de maquinari de propietat, per la qual cosa es pot executar al software per accelerar la innovació de serveis i aprovisionament, en particular en els entorns de proveïdors de serveis.

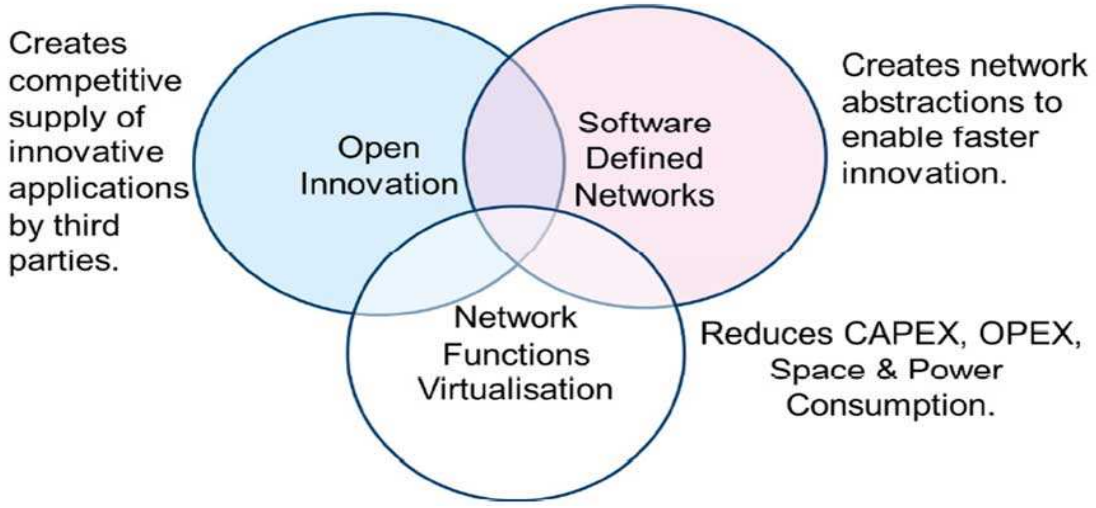



Figura 18 Relació de Network Functions Virtualisation amb SDN

http://www.slideshare.net/buildacloud/l4l7-services-for-sdn-and-nvf-by-youcef-laribi

Com es mostra a la Figura anterior, NFV és altament complementària de SDN, però no depenen l'una de l'altre. NFV es pot implementar sense que es requereixi SDN, encara que les dues arquitectures es poden combinar i obtenir d'aquesta forma un major valor potencial. Però l'enfocament basant-se en la separació dels plànols de control i de reexpedició de dades segons el proposat per SDN pot millorar el rendiment, simplificar la compatibilitat amb les implementacions existents, i facilitar els procediments d'operació i manteniment. NFV a més, proporciona la infraestructura sobre la qual el software SDN es pot executar. A la següent figura es pot observar com un router tradicional és gestionat.

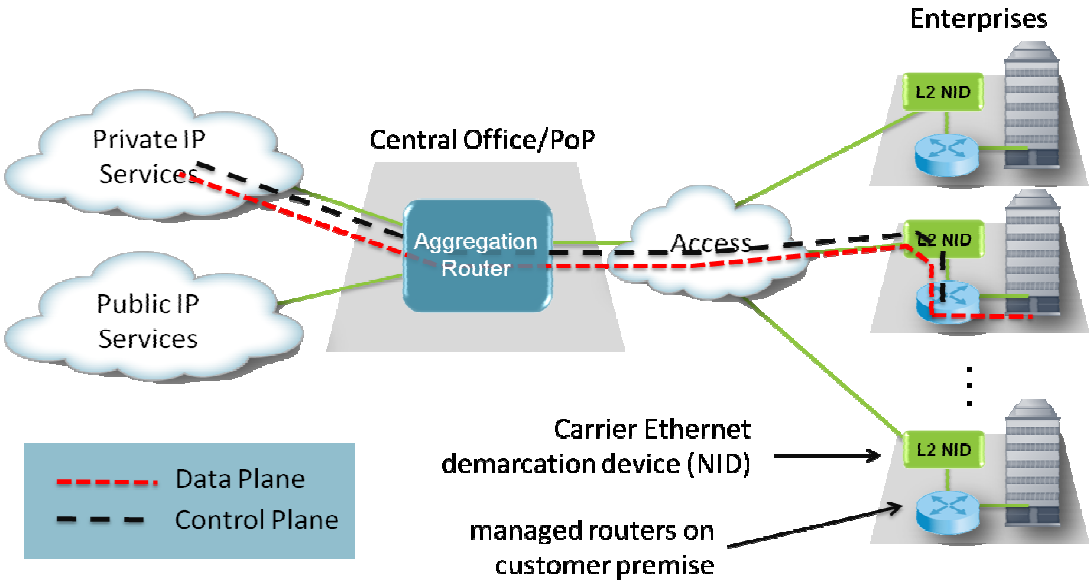



Figura 19 Gestió d'un router tradicional

A la següent figura, NFV s'aplicarà a aquesta situació mitjançant la virtualització de la funció del router. El resultat és un dispositiu d'interfície de xarxa (NID) que proporciona un punt de demarcació i calcula el rendiment.

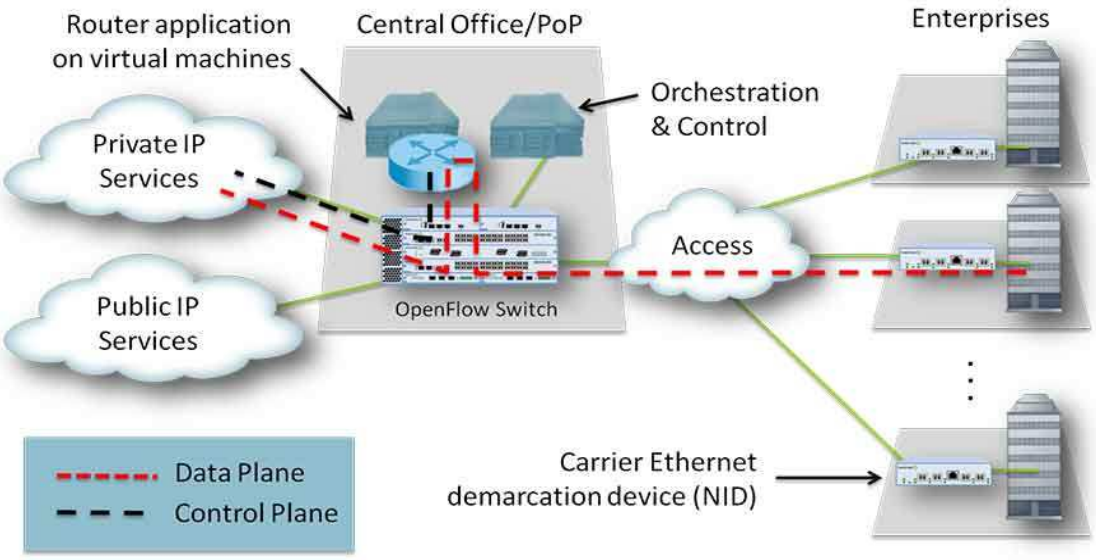



Figura 20 Router fent servir NFV

Finalment, SDN s'introdueix per separar els plànols de control i dades. Ara, els paquets de dades es transmeten per un pla de dades optimitzat, mentre que la funció d'encaminament (pla de control) s'està executant en una màquina virtual de forma centralitzada.

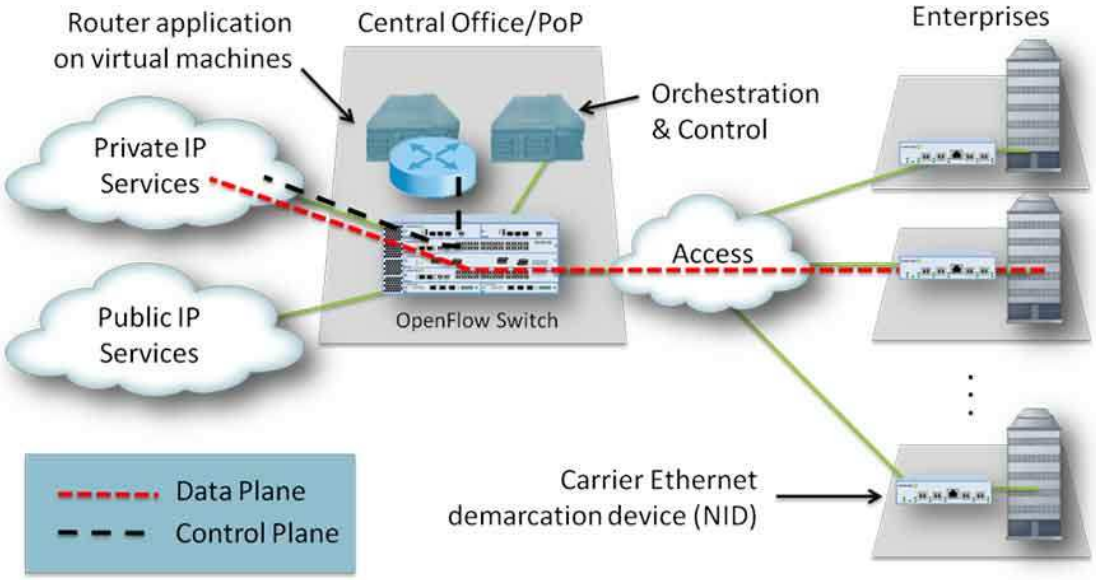



Figura 21 Router fent servir NFV i SDN

La solució anterior mostra la combinació de SDN i NFV, proporcionant els següents beneficis:

- Un dispositiu car i dedicat es substitueix pel maquinari genèric i un software avançat.
- El pla de control de software es mou des d'una plataforma dedicada a una ubicació optimitzada (servidor en un centre de dades o POP).
- El control del pla de dades ha estat extret i estandarditzat, la qual cosa permet l'evolució de la xarxa i l'aplicació sense necessitat de millores dels dispositius de xarxa.

| Categoria | SDN | NFV |
|---|---|---|
| **Justificació** | Separació de control y dades, centralització del control y capacitat de programació de la xarxa. | La reubicació de les funcions de xarxa de dispositius dedicats a servidors genèrics. |
| **Ubicació de destí** | Campus, data center, Cloud. | Proveïdors de serveis de xarxa. |
| **Dispositius de destí** | Commodity servers i switchos | Commodity servers i switchos |
| **Principals aplicacions** | Xarxes i Cloud. | Routers, firewalls, gateways, CDN, acceleradores WAN, SLA. |
| **Nous protocols** | Openflow | - |
| **Formalització** | Open Networking Forum (ONF) | ETSI NFV Working Group |

Taula 1 comparativa SDN / NFV

# CAPÍTOL 4

# 4. PROTOCOLS

## 4.1. Openflow

És important remarcar que OpenFlow no és SDN. OpenFlow és un protocol que mitjançant una API oberta proporciona una interfície estàndard per a la programació dels switchos en el pla de dades, oferint a un servidor de software determinar el camí de reenvio de paquets que hauria de seguir en una xarxa de switchos i/o routers. D'aquesta forma, la xarxa pot ser programada independentment dels commutadors individuals i de l'equip del centre de dades.

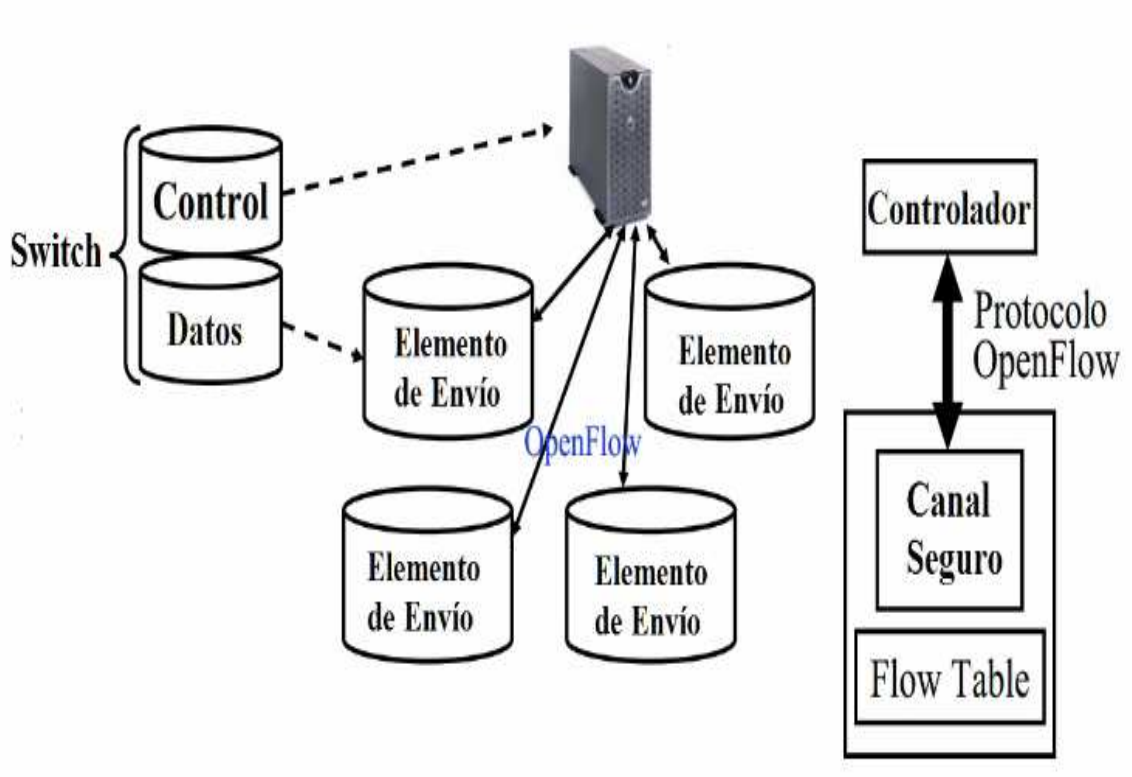



Figura 22 Separació de plànols amb Openflow

http://docplayer.es/865687-Indice-1-openflow-y-sus-herramientas-2-software-defined-networking-sdn-3-network-function-virtualization-nfv.html

El principal motiu pel qual es va crear OpenFlow és que els proveïdors de dispositius venen commutadors o routers amb una limitació en quan a programació, la qual cosa

comporta nombroses dificultats en quan a la gestió del tràfic i l'enginyeria, així com fluxos de tràfic inconsistents entre el maquinari de diferents proveïdors. OpenFlow ofereix aquesta coherència en prendre remotament el control del maquinari, implementant-ho amb el software.

OpenFlow va oferir diverses contribucions addicionals:

- Estandarditzant dispositius i funcions de xarxa: A les xarxes tradicionals, el control de rutes es centrava principalment en la coincidència de tràfic mitjançant el prefix IP de destinació. Per contra, les normes de OpenFlow podien definir el comportament de reenvio de fluxos de tràfic basat en un conjunt de 13 capçaleres de paquets diferents, unificant molts dispositius que es diferenciaven només en alguns termes dels camps de la capçalera. Un router coincideix amb el prefix IP de destinació i envia un enllaç, mentre que un switch coincideix amb l'origen i destinació de l'adreça MAC. Traductors d'adreces de xarxa i tallafocs coincideixen en les cinc tuples (adreces IP d'origen i destinació, nombres de port i protocol de transport). OpenFlow també va estandarditzar les tècniques d'instal·lació de regles. Tot i així, no ofereix suport de pla de dades en quan a la inspecció a fons de paquets o a la reconnexió.

- La visió d'un sistema operatiu de xarxa: OpenFlow va portar cap a la noció d'un sistema operatiu de xarxa. Un sistema operatiu de xarxa és un software que abstreu la instal·lació d'estat en els commutadors de xarxa, de la lògica i les aplicacions que controlen el comportament de la mateixa. De manera més general, l'aparició d'un sistema operatiu de xarxa ofereix una descomposició conceptual de funcionament de la xarxa en tres capes:
    - Pla de dades amb una interfície oberta.
    - Una capa de gestió d'estat, que s'encarrega de mantenir una visió coherent de l'estat de la xarxa.
    - Lògica de control, que realitza diverses operacions en funció de la seva visió de l'estat de la xarxa.

- Tècniques de gestió d'estats distribuïts: L'execució de diversos controladors és crucial per a l'escalabilitat, fiabilitat i rendiment, no obstant això, aquestes rèpliques han de treballar juntes per actuar com un sol controlador, lògicament centralitzat. Per suportar aplicacions de controlador arbitràries, el controlador

Onix va introduir la idea d'una xarxa d'informació de base, una representació de la topologia de xarxa i una altra de l'estat de control compartit per totes les rèpliques del controlador. Onix també va incorporar treballs anteriors en els sistemes distribuïts per satisfer els requisits de durabilitat i consistència d'estat. Més recentment, el sistema ONOS ofereix un controlador de codi obert amb una funcionalitat similar, l'ús de software de codi obert existent per mantenir la coherència en tot l'estat distribuït i proporcionar una base de dades de topologia de xarxa al controlador d'aplicacions.

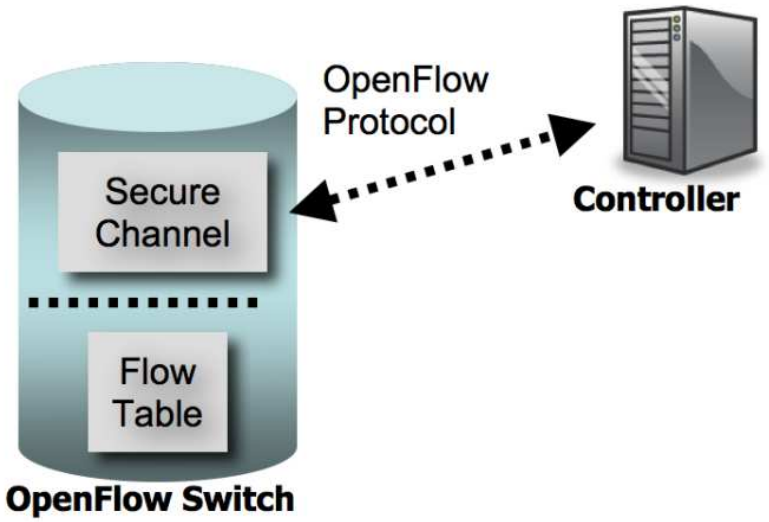



Figura 23 Protocol Openflow

https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-spec-v1.0.0.pdf

### 4.1.1. Switch Openflow

Els components d'un switch OpenFlow consisteixen en una taula de flux, que realitza operacions de cerca de paquets i enviament, un canal segur i un controlador extern, tal com es pot apreciar a la figura anterior. El switch Openflow es comunica amb el controlador mitjançant el canal segur (connexió segura) usant un protocol Openflow. La taula de fluxos conté un conjunt d'entrades de flux. Cada entrada de flux inclou valors de capçalera (per comparar-la amb la que contenen els paquets), un comptador d'activitat i una acció per aplicar als paquets que concordin amb aquesta capçalera.

Tots els paquets processats pel switch es comparen amb la taula de fluxos. Si es troba una entrada coincident, l'acció associada a aquesta entrada s'executa en el paquet (per exemple, l'acció podria ser la d'enviar un paquet a un port especificat). Si no es troba cap coincidència, el paquet s'envia al controlador a través del canal segur. El controlador és el responsable de determinar com actuar amb els paquets sense entrades de flux vàlides, a més, gestiona la taula de fluxos del switch mitjançant l'addició, edició o eliminació d'entrades de flux.

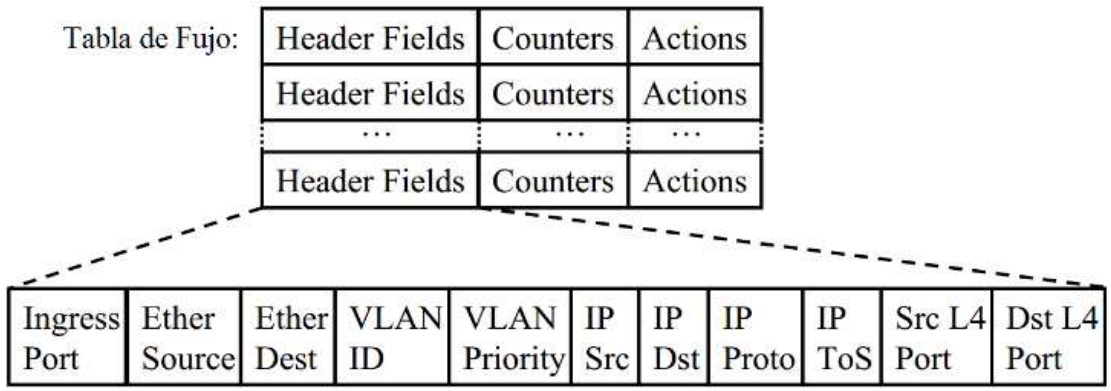
Les entrades de flux poden enviar paquets a un o més ports OpenFlow, els quals poden ser o no ports físics (ports agregats o tràfic VLAN). Els Ports OpenFlow es poden trobar en dos estats, habilitats o inhabilitats. La configuració addicional dels ports pot realitzar-se exclusivament a través del protocol de configuració de OpenFlow.

### 4.1.2. Taula de fluxos

La taula de fluxos consta dels següents components:

- Camp de capçalera.
- Comptador.
- Acció.

El camp de capçalera conté les dades a comparar amb la capçalera del paquet entrant, es pot observar a la següent figura.




Taula 2 Capçalera fluxos

http://docplayer.es/865687-Indice-1-openflow-y-sus-herramientas-2-software-defined-networking-sdn-3-network-function-virtualization-nfv.html

Els dissenyadors de switchos poden implementar el disseny de les seves taules lliurement sempre que respectin les especificacions de Openflow.

### 4.1.3. Comptadors

Existeixen comptadors per taula, per flux, per port i per cua. Es creen en OpenFlow mitjançant software i actualitzats mitjançant el maquinari. En la següent figura es pot observar la taula de comptadors:

| Counter | Bits |
|---|---|
| Per Table | |
| Active Entries | 32 |
| Packet Lookups | 64 |
| Packet Matches | 64 |
| Per Flow | |
| Received Packets | 64 |
| Received Bytes | 64 |
| Duration (seconds) | 32 |
| Duration (nanoseconds) | 32 |
| Per Port | |
| Received Packets | 64 |
| Transmitted Packets | 64 |
| Received Bytes | 64 |
| Transmitted Bytes | 64 |
| Receive Drops | 64 |
| Transmit Drops | 64 |
| Receive Errors | 64 |
| Transmit Errors | 64 |
| Receive Frame Alignment Errors | 64 |
| Receive Overrun Errors | 64 |
| Receive CRC Errors | 64 |
| Collisions | 64 |
| Per Queue | |
| Transmit Packets | 64 |
| Transmit Bytes | 64 |
| Transmit Overrun Errors | 64 |

Taula 3 Comptadors fluxos

https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-spec-v1.0.0.pdf

### 4.1.4. Accions

Cada entrada de flux pot associar-se des de cap, a diverses accions, les quals dicten al switch que fer amb el paquet. En el cas de no existir cap acció en aquest flux el paquet es descarta. Les accions han de ser executades dins de l'ordre ja especificat però el flux de paquets de sortida no comporta aquest ordre, un paquet pot ser replicat per un mateix port de sortida a diverses VLAN's diferents sense poder predeterminar l'ordre de sortida. Un switch pot rebutjar, en un moment determinat, una entrada de flux si no pot processar la llista d'accions en l'ordre especificat, en aquest cas retornarà un error.

Els switchos poden ser exclusivament Openflow o compatibles amb ell, per tant, les accions marcades com a necessàries són d'obligatòria execució, no passa el mateix amb les accions opcionals en les quals el switchos han de comunicar al controlador quines d'elles poden suportar. El controlador pactarà amb el switch si pot realitzar enviaments simultanis a múltiples ports físics.

A continuació s'enumeren uns exemples d'accions necessàries i opcionals.

Accions necessàries:

- ALL: Envia el paquet a totes les interfícies, no incloent la interfície d'entrada.
- CONTROLLER: Encapsula i envia el paquet al controlador.
- LOCAL: Envia el paquet a la pila de xarxa local de switchos.
- TABLE: Realitza accions de la taula de fluxos. Només per als missatges de paquets sortints.
- IN PORT: Envia el paquet pel port d'entrada.
- DROP: Una entrada de flux sense cap acció especificada indica aquests paquets han de ser rebutjats.

Accions opcionals:

- NORMAL: Processa el paquet comprovant el camp VLAN prèviament, fent servir la ruta d'enviament tradicional.
- FLOOD: Envia el paquet per tots els ports excepte el d'entrada.
- ENQUEUE: Envia un paquet a través d'una cua connectada a un port.

### 4.1.5. Coincidència (matching)

Després de la recepció d'un paquet, un switch OpenFlow realitza alguna de les funcions descrites abans. Els camps de capçalera utilitzats per a la cerca a la taula dependran del tipus de paquet, tal i com es descriu a continuació:

- Regles que especifiquen un port d'entrada es comparen amb el port físic que rep el paquet.
- Les capçaleres Ethernet s'utilitzen per a tots els paquets.
- Si el paquet és d'una VLAN (tipus Ethernet igual a 0x8100), els camps de ID i PCP s'utilitzaran en les operacions de cerca.
- Per a paquets ARP (tipus Ethernet igual a 0x0806), els camps de cerca també poden incloure els camps d'origen i de destinació.
- Per als paquets IP (tipus Ethernet igual a 0x0800), els camps de cerca inclouen també els de la capçalera IP.
- Per a paquets IP que són TCP o UDP, la cerca inclou els ports de transport.
- Per als paquets IP que són ICMP, la cerca inclou els camps Codi i Tipus.

Un paquet coincideix amb una entrada de la taula de fluxos si els seus valors en els camps de capçalera coincideixen amb els definits en ella. Si un camp té un valor de 'ANY', coincidirà amb qualsevol valor possible.

Els paquets es comparen amb les entrades de flux basant-se en prioritats. Una entrada que especifica una coincidència exacta (és a dir, no té comodins) sempre és la més alta prioritat. Totes les entrades de comodí tenen una prioritat associada amb ells. Entrades de major prioritat han de coincidir abans que les de menor prioritat. Si diverses entrades tenen la mateixa prioritat, el switch és lliure de triar qualsevol comanda. Els nombres més alts tenen major prioritat. Per a cada paquet que coincideix amb una entrada de flux, els comptadors associats per a aquesta entrada s'actualitzen. Si no hi ha cap entrada coincident per a un paquet, s'envia al controlador a través del canal segur.

### 4.1.6. Canal segur

El canal segur és la interfície que connecta cada switch OpenFlow a un controlador. A través d'aquesta interfície, el controlador configura i administra el switch, rep esdeveniments des d'ell, i li envia paquets.

Entre la ruta de dades (datapath) i el canal segur, la interfície és específica de l'aplicació, no obstant això tots els missatges del canal segur han de tenir un format d'acord amb el protocol OpenFlow.

### 4.1.7. Generalitats del protocol OpenFlow

El protocol OpenFlow es compatible amb tres tipus de missatges cadascun amb múltiples subtipus:

- **Controlador a switch**. els missatges s'inicien des del controlador i s'utilitzen per a l'administració o inspecció de l'estat del switch. El llistat de missatges és el següent:

    - Característiques: durant una sessió d'establiment TLS, el controlador envia un missatge amb una petició pel switch, el qual ha de respondre informant-li de les accions suportades.
    - Configuració: El controlador pot establir els paràmetres de configuració i realitzar query's al switch. El switch per la seva banda només respon a query's del controlador.
    - Modify-State: El seu propòsit principal és afegir, eliminar o modificar els fluxos en les taules de flux a més d'establir les propietats de port del switch.
    - Read-State: són utilitzats pel controlador per recollir estadístiques dels switches (taules de fluxos, ports i entrades individuals).
    - Send-Packet: Aquests són utilitzats pel controlador per enviar paquets a un port específic del switch.
    - Barrier: funció utilitzada pel controlador per al control de notificacions.

- **Asíncron**. en aquest cas els missatges són iniciats pel switch, s'utilitzen per realitzar canvis en el seu estat i per actualitzar el controlador dels

esdeveniments dels fluxos. A continuació es descriuen els quatre principals tipus de missatges asíncrons:

- Packet-in: Per als paquets que no tenen coincidència en la seva capçalera o els que l'acció de la regla indica “send to controller”, es procedeix de dues formes, en el cas que el switch tingui memòria suficient en el buffer s'envia una porció dels paquets al controlador. En el cas que el switch no disposi d'aquesta memòria no queda més remei que enviar-li els paquets complets al controlador.

- Flow-Removeu: Quan s'agrega una entrada de flux al switch mitjançant una modificació de flux, existeix un temps d'espera que eliminarà aquesta entrada si es superi per inactivitat. El switch en aquest cas, envia un missatge al controlador informant-li de l'eliminació d'aquesta entrada.
- Port d'estat: el switch envia missatges al controlador informant-li dels canvis en la configuració de l'estat dels ports.
- Error: El switch pot notificar al controlador qualsevol problema mitjançant missatges d'error.

- **Simètric**. els missatges es poden iniciar ja sigui pel switch o bé pel controlador i enviar-los sense sol·licitud. Els missatges són els següents:

  - Hello: aquests missatges s'intercanvien entre el switch i el controlador en l'inici de sessió.
  - Echo: són missatges enviats tant des del controlador com des del switch i serveixen per indicar la latència, ample de banda o simplement per saber si la connexió encara és vàlida.
  - Vendor: missatges reservats per a futures funcions addicionals. Es tracta d'un àrea d'assaig destinada a futures revisions OpenFlow.

### 4.1.8. Configuració de la connexió

El switch ha de facilitar un accés mitjançant adreça IP per a d'aquesta forma a través d'un port definit per l'usuari, poder accedir a la configuració del mateix. Abans de realitzar cap canvi en la configuració, el switch ha d'assegurar-se que el tràfic entrant

pel canal segur pertany al controlador. Per a això, la primera vegada que s'estableix una connexió OpenFlow, cada costat de la connexió ha d'enviar un missatge "OFPT_HELLO" amb el camp 'versió' establert en la versió més alta suportada. Després de la recepció d'aquest missatge, el destinatari pot calcular la versió del protocol OpenFlow per ser utilitzat durant el transcurs de la comunicació. Si el destinatari no pot suportar el nombre de versió respondrà amb un missatge d'error "OFPT_ERROR" i donarà per finalitzada la connexió.

### 4.1.9. Interrupció de connexió

En el cas que un switch perdi el contacte amb el controlador, ha d'intentar posar-se en contacte amb un o més controladors de còpia de seguretat. Si algun intent falla, el switch ha d'entrar en "mode d'emergència" i restablir immediatament la connexió TCP actual. En aquest mode, el procés de 'matching' ve dictat per les regles de la taula que tenen el bit d'emergència marcat, la resta s'eliminen en adquirir aquest estat. Al connectar a un controlador de nou, les entrades de fluxos d'emergència romanen, decidint en aquest cas el propi switch si esborrar-les o no. Sempre que un switch s'engega per primer cop, entra en mode emergència.

### 4.1.10. Encriptació

El switch es comunica amb el controlador a través d'una connexió TLS mitjançant el port 6633, autenticant-se mútuament mitjançant l'intercanvi de certificats signats per una clau privada. Cada switch ha de ser configurable per l'usuari amb un certificat per a l'autenticació del controlador (certificat de controlador) i un altre per autenticar al controlador (certificat de switch).

### 4.1.11. Spanning Tree

Els switches OpenFlow poden suportar opcionalment el protocol STP 802.1D, per a això ha d'establir el bit OFPC_STP en el camp 'capacitats' del seu missatge OFPT_FEATURES_REPLY. Un switch que implementa STP ha de posar-ho a

disposició de tots els seus ports físics, però no en els seus ports. Els switchos que no suporten 802.1D Spanning Tree han de permetre que el controlador especifiqui l'estat del port per 'packet flooding' a través de missatges port-mod.

### 4.1.12. Missatges de modificació de la taula de fluxos

Els missatges de comunicació per variar les taules de fluxos, explicat als punts anteriors, poden ser dels següents tipus:

OFPFC_ADD- Indica que s'afegirà el flux actual a la taula de fluxos existent.

OFPFC_MODIFY- Indica que s'afegirà o es modificarà un flux.

OFPFC_MODIFY_STRICT- Indica que s'afegirà o es modificarà un flux amb coincidència total.

OFPFC_DELETE- Indica que s'eliminarà un flux.

OFPFC_DELETE_STRICT -Indica que s'eliminarà un flux amb coincidència total.

Accions a realitzar:

- Per a les sol·licituds de ADD amb el flag OFPFF_CHECK_OVERLAP marcat, el switch ha de comprovar que no existeix ja a la taula, en cas contrari, el switch ha de rebutjar l'agregació i respondre amb un missatge ofp_error_msg amb el tipus OFPET_FLOW_MOD_FAILED i el codi OFPFMFC_OVERLAP.
- Si una entrada de flux (sense contenir el flag OFPFF_CHECK_OVERLAP marcat) amb camps de capçalera i prioritat idèntics ja resideix a la taula, llavors aquesta entrada, incloent els seus comptadors, ha de ser eliminada i la nova entrada afegida.
- Si un switch no pot trobar cap taula en la qual afegir la nova entrada de flux, ha d'enviar un missatge OFP_ERROR_MSG amb el tipus OFPET_FLOW_MOD_FAILED i el codi OFPFMFC_ALL_TABLES_FULL.
- Si es produeix una entrada que fa referència a un port que no existeix a un switch, aquest ha de retornar un ofp_error_msg amb el tipus OFPET_BAD_ACTION i el codi OFPBAC_BAD_OUT_PORT. Si més endavant aquest port s'activés i arribessin paquets a través d'ell, el switch podria deixar

caure aquests paquets i immediatament retornaria un error OFPBAC_BAD_OUT_PORT.

- Per a les sol·licituds de modificació (MODIFY), si una entrada de flux amb camps de capçalera idèntics no existeix a la taula, s'actua com un ADD, i la nova entrada de flux s'ha d'inserir amb comptadors a zero, en cas contrari, el camp ‘acció’ es canvia a l'entrada existent i els seus comptadors i camps de temps d'inactivitat es deixen sense canvis.
- Per a sol·licituds d'eliminació (DELETE), si no hi ha entrada de flux coincident, no es produeix cap modificació en la taula de flux ni s'envia missatge algun. Si les entrades de flux coincideixen, cada entrada de sol·licitud d'eliminació amb el flag OFPFF_SEND_FLOW_REM marcat ha de generar un missatge de flux eliminat, en cas contrari no informaran.
- Si les modificacions o eliminacions de fluxos apareixen amb el comando STRICT, només es veuran afectades les entrades d'aquesta taula, en cas contrari afectarà a les entrades de totes les taules.
- Cada entrada de flux té els comptadors de temps ‘idle_timeout’ i ‘hard_timeout’ associats. Si cap paquet ha coincidit amb la regla en els segons que marca el camp ‘idle _timeout’, o han passat els segons que marca l'altre camp ‘hard_timeout’ des de la seva inserció, el switch elimina l'entrada i envia un missatge de flux eliminat.

## 4.2. Openstack

Openstack és un software lliure i de codi obert usat per a la construcció de clouds públiques i privades. Principalment actua a la capa de Infrastructure as a Service (IaaS), la capa més complexa de totes dins del Cloud computing, on s'estableixen especialment l'emmagatzematge i la capacitat de còmput (màquines virtuals) oferts tots dos com a serveis en el núvol i amb l'obligació d'estar virtualitzats. Desenvolupada amb la idea de ser senzilla d'implementar, massivament escalable i amb moltes prestacions, aquesta tecnologia consisteix en un grup de projectes interrelacionats que controlen el processament, emmagatzematge i recursos de la xarxa mitjançant l'administració per part dels usuaris, a través d'una API o bé d'interfícies en mode de panell de control, basats a la web.

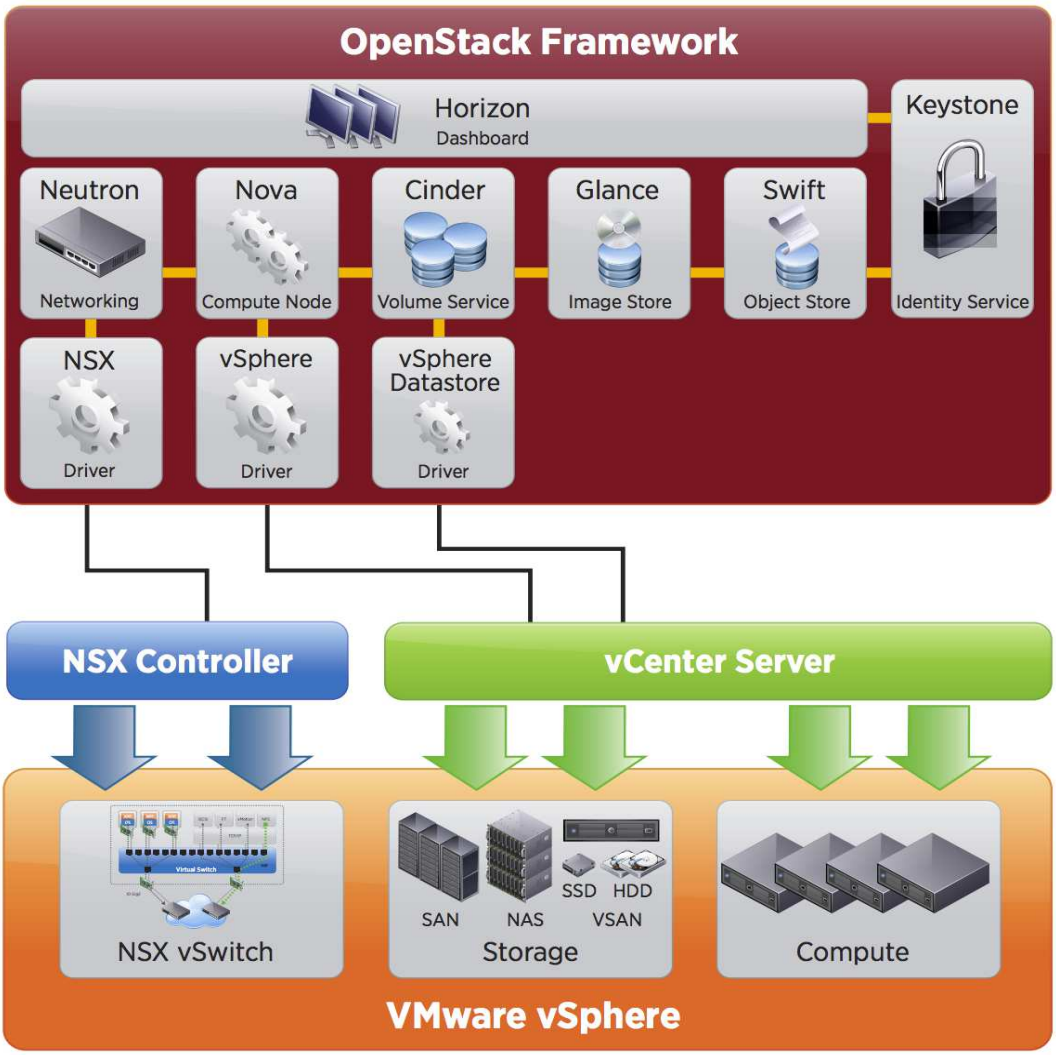



Figura 24 Diagrama Openstack

http://www.ntpro.nl/blog/archives/2555-Getting-Started-with-OpenStack-and-VMware-vSphere.html

OpenStack es va iniciar al 2010 com un projecte conjunt de Rackspace Hosting i la NASA. Des del 2015 està gestionat per la Fundació OpenStack, una entitat corporativa sense finalitat de lucre, establerta al setembre de 2012 per promoure el software OpenStack i la seva comunitat. Actualment més de 500 empreses s'han unit al projecte. Openstack.org i sota llicencia Apatxe, realitza lliuraments de noves versions en cicles que ronden els sis mesos.

L'arquitectura modular de OpenStack està formada per diferents components d'entre els que es vol destacar els sis més importants:

### 4.2.1. OpenStack Compute (Nova)

És el principal component de OpenStack i s'encarrega de gestionar les instàncies (màquines virtuals) d'usuaris i grups d'usuaris, així com la xarxa virtual per a cadascuna de les que formen part d'un projecte (tenant).

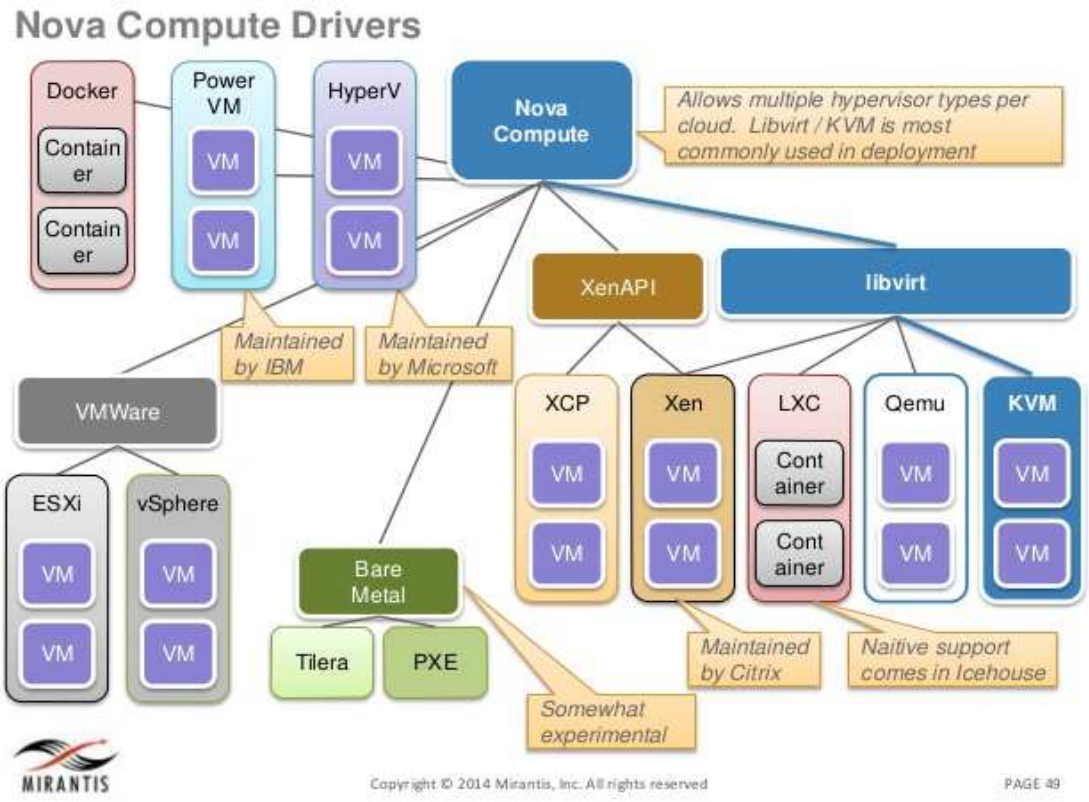



Figura 25 Drivers i hipervisors suportats per Nova

http://www.slideshare.net/mirantis/openstack-architecture-43160012

Les seves principals característiques són les següents:

- Suporta diversos models de hypervisor (ESXi, vSphere, LXC, Docker, HyperV, KVM, XCP i Xen) , tal com es mostra a la següent figura.
- Els usuaris es comuniquen fent servir una API a través del component nova-api, mitjançant el qual s'apliquen les polítiques i inicien serveis com ara el llançament d'instàncies.
- El llenguatge de programació és el Python.
- Es compon de múltiples processos de servidor.

- Les bases de dades suportades són PostgreSQL, MySQL i sqlite3.

Interactua amb Glance per a la recuperació d'imatges, amb Keystone per lo relatiu a l'autenticació i amb Horizon per a la interfície administrativa i d'usuaris.

### 4.2.2. OpenStack Object Storage (Swift)

S'encarrega de l'emmagatzematge d'objectes mitjançant un sistema tolerant a fallades, escalable i redundant, tot i que no permet muntar directoris tal i com ho faria un sistema de fitxers basat en NFS. Entre els seus components s'inclou un servidor proxy que accepta sol·licituds (pujada i descàrrega de fitxers u objectes) mitjançant API's o HTTP, un servidor de comptes d'usuari que s'encarrega de gestionar les mateixes, servidors d'objectes (fitxers o contenidors) distribuïts en els diferents modes d'emmagatzematge i finalment un conjunt de processos entre els quals es troben el servei de replicació, auditors, actualitzadors i reapers.

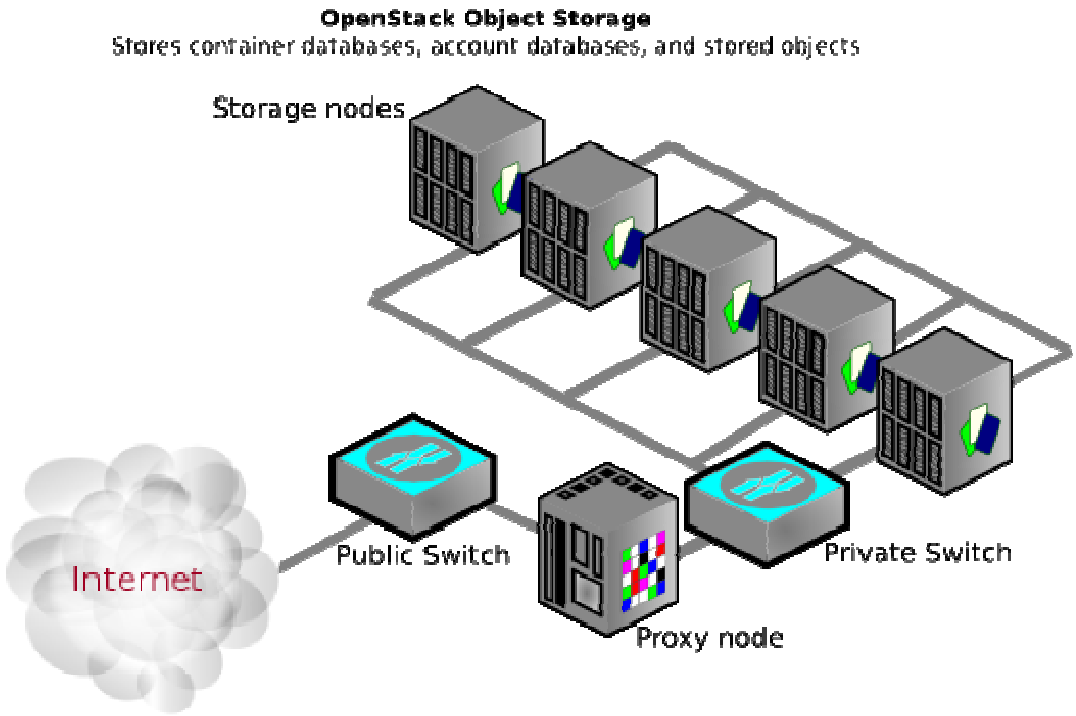



Figura 26 Arquitectura genèrica Openstack Object Storage

http://kernal.blog.51cto.com/8136890/1540500

### 4.2.3. OpenStack Identity Service (Keystone)

La funció d'aquest servei és la de gestionar l'autenticació entre la resta d'elements (components o usuaris), utilitzant per a això un sistema basat en tokens. A la vegada, pot realitzar el manteniment d'un catàleg i un repositori de polítiques d'identitat.

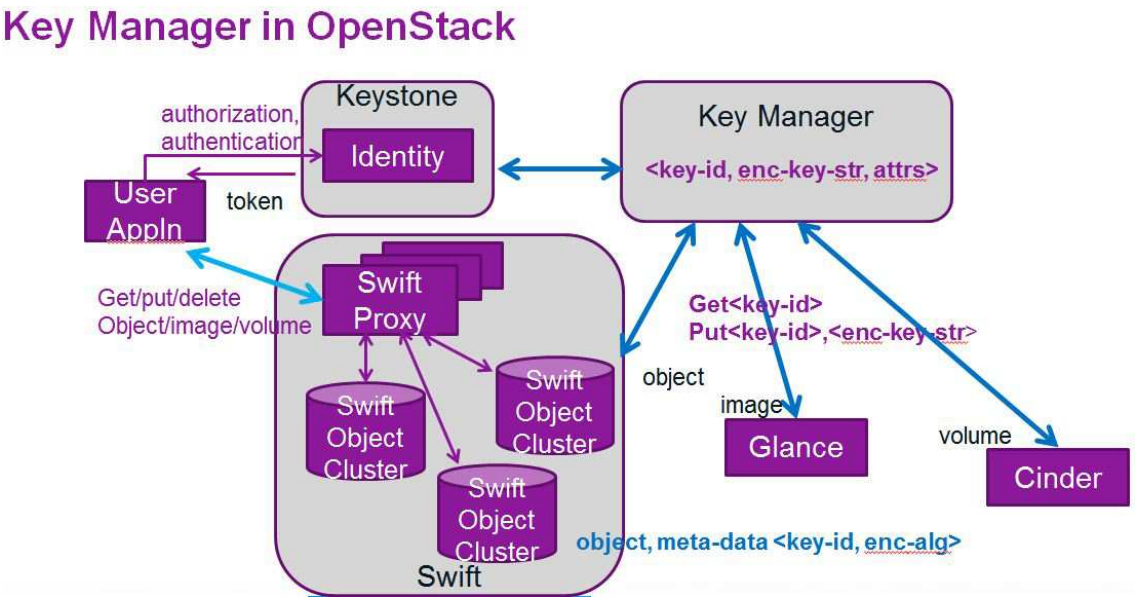



Figura 27 Gestor de claus de Openstack

https://wiki.openstack.org/wiki/KeyManager

### 4.2.4. OpenStack Image Service (Glance)

És l'encarregat de gestionar les imatges de les màquines virtuals, podent emmagatzemar-les directament o bé mitjançant mecanismes com ara Object Storage o bé Amazon's Simple Storage Solution. Per dur a terme aquesta tasca es fa valer de quatre components:

- glance-api que gestiona les sol·licituds provinents de la seva API.
- glance-registry que registra metadades de cadascuna de les imatges.
- una base de dades per allotjar aquesta informació.
- un magatzem per allotjar les imatges.

El seu paper és centralitzat ja que les sol·licituds provenen tant d'usuaris finals com d'altres components.

### 4.2.5. OpenStack Networking (Neutron)

Neutron, ha reemplaçat a Quàntum com a component de networking, sent el component d'administració de xarxa un dels més complexos. A través de plugins i agents és pot accedir al seu API amb la finalitat d'habilitar o deshabilitar ports, crear xarxes i subxarxes controlant el seu adreçament IP, exercir un control addicional sobre les polítiques de seguretat, tot això vinculat als dispositius que és vagin a utilitzar ja siguin físics o virtuals.

La API de Neutron inclou suport per a la capa 2 de xarxes i la gestió d'adreces IP (IPAM), així com una extensió per a la capa 3, permetent l'encaminament entre xarxes de Capa 2 i portes d'enllaç a xarxes externes. Neutron inclou una creixent llista de plugins que permeten la interoperabilitat amb diferents tecnologies de xarxes comercials i de codi obert, incloent routers, switches, commutadors virtuals i controladors definits per software de xarxes (SDN). Per a xarxes petites o senzilles els plugins poden utilitzar VLAN's de Linux i IP tables, però per a infraestructures més grans es necessiten tecnologies avançades com L2-in-L3 tunneling o OpenFlow.

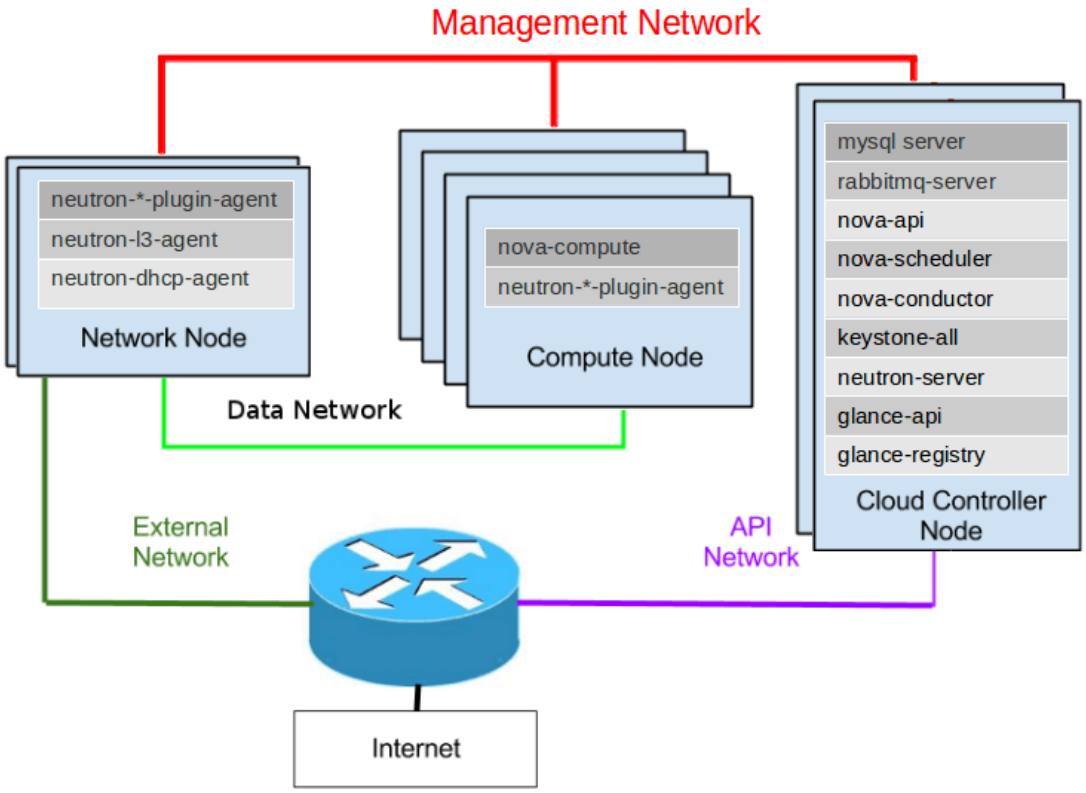



Figura 28 Comunicació entre l'administració de xarxa i els nodes

http://cloudn1n3.blogspot.com.es/2014/11/openstack-series-part-8-neutron.html

El component de Networking s'encarrega de proporcionar recursos de xarxa a les instàncies que s'executen als mòduls de Nova, definint adrecis IP que s'assignaran a les instàncies i grups de switch distribuïts, on es connectaran les màquines virtuals. Pot interactuar amb els switches que incorporen el protocol de comunicació OpenFlow.

Els tres principals agents de Neutron són:

- Neutron L3 agent
- DHCP agent
- Neutron plugin agent.

### 4.2.6. OpenStack Dashboard (Horizon)

Desenvolupat en Django (framework de Python), és una aplicació web que permet comunicar-se a tots els components, inclòs els usuaris, a través de les diferents API's. L'execució de Horizon es realitza mitjançant el mòdul d'Apatxe mod_wsgi, separada en dos fragments de codi Python reutilitzables, el primer d'ells interactua amb diverses de les API's mentre que el segon s'encarrega de la usabilitat i integració del site web.

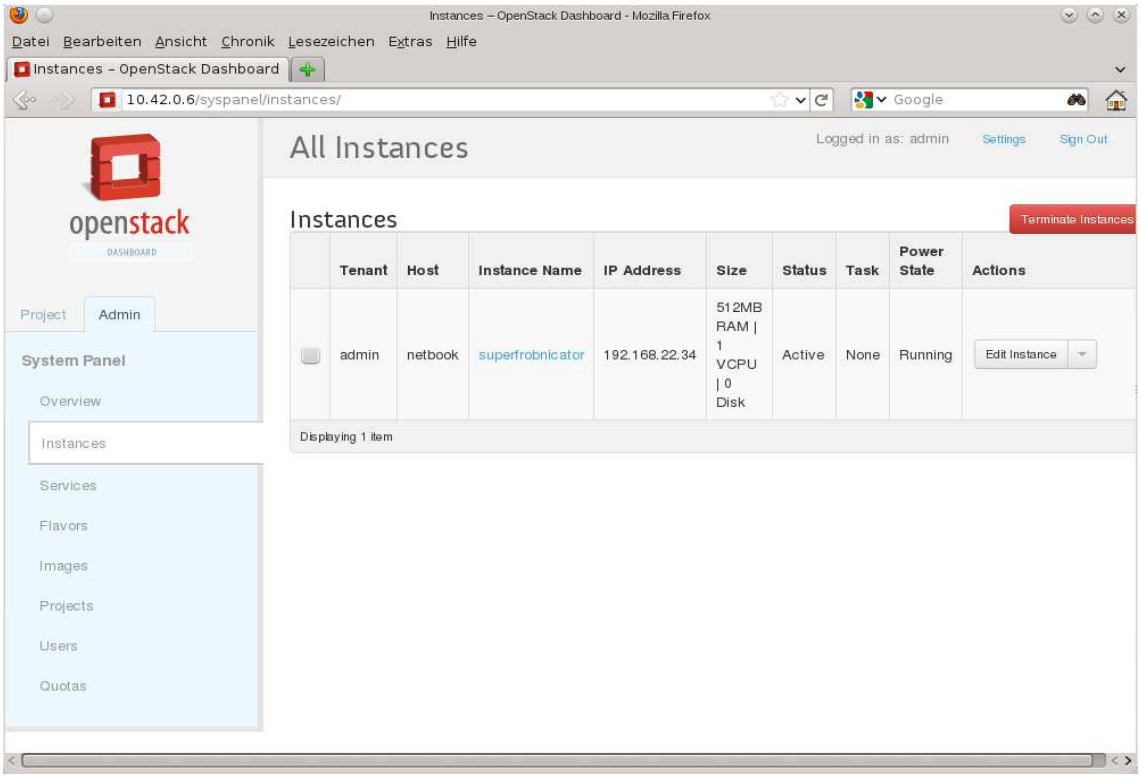



Figura 29 Mostra de Openstack Dashboard

### 4.2.7. Arquitectura conceptual

Per poder aconseguir lliurar un sistema operatiu que permeti el desplegament de clouds altament escalable, cadascun dels serveis ha de treballar conjuntament amb la resta d'ells mitjançant l'oferta i consum de API's. Les mateixes API's serveixen tant per a una comunicació entre serveis com per a la gestió per part dels usuaris. Les relacions entre els serveis es veuen representades a la següent figura:

La següent figura mostra una relació bastant simplificada entre serveis, des del punt de vista de l'operador del cloud, sense representar l'ús per part dels serveis que ho consumeixen.

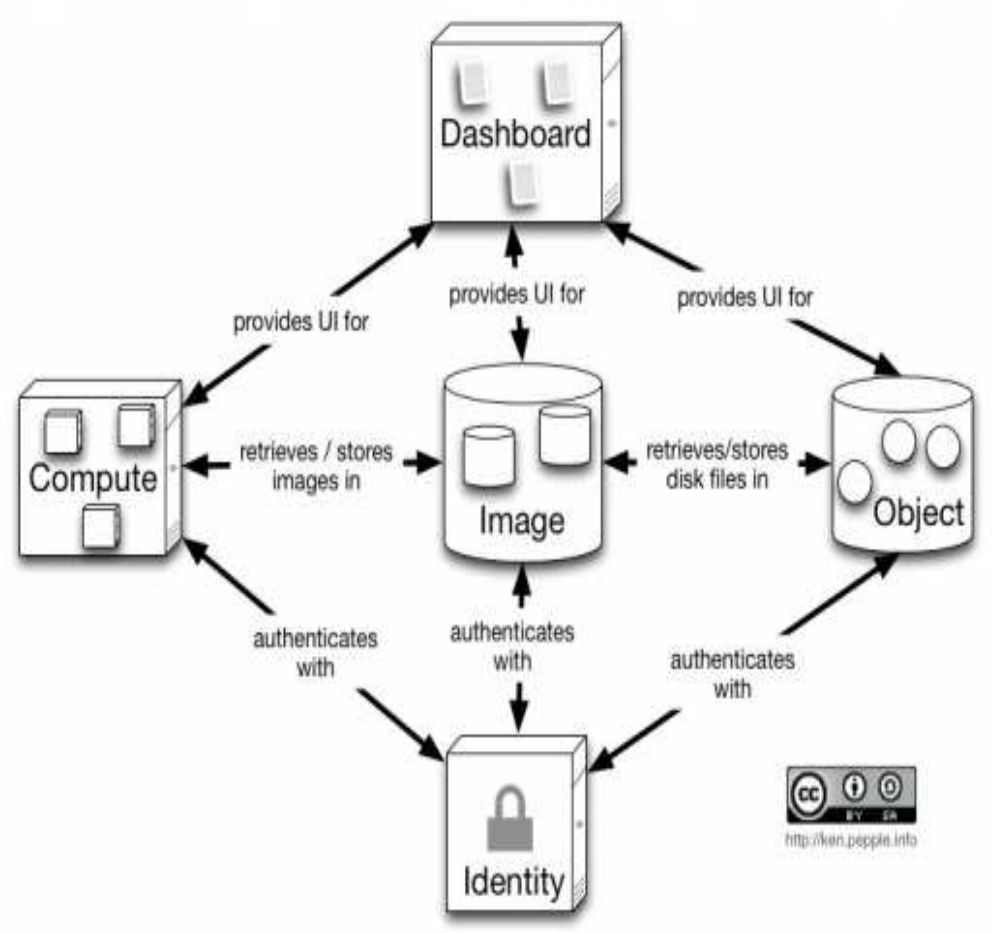



Figura 30 Relació entre serveis a Openstack

### 4.2.8. Arquitectura lògica

La arquitectura lògica tracta de relacionar la totalitat dels components i usuaris cloud. A la següent figura es pot observar un exemple aproximat d'aquesta relació.

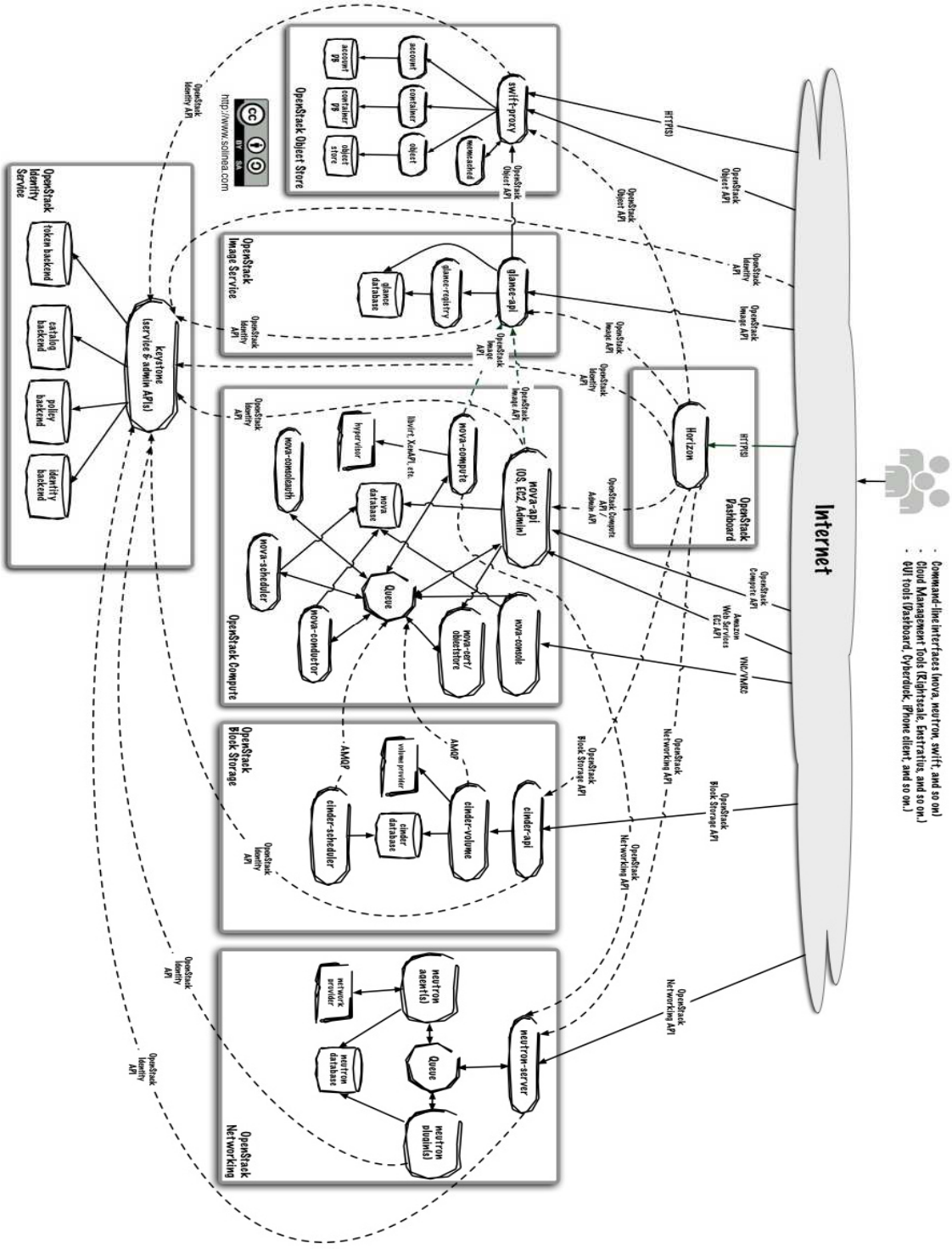



Figura 31 Arquitectura lògica Openstack

http://docs.openstack.org/icehouse/training-guides/content/operator-getting-started.html

# CAPÍTOL 5

# 5. IMPLEMENTACIONS EXISTENTS DELS ESTÀNDARDS O APROXIMACIONS

Existeixen dos models d'estàndards d'implementació, els basats en software i els basats en hardware.

## 5.1. Basats en software

Es passa a detallar dos dels més importants.

### 5.1.1. Vmware NSX

Com ja s'ha comentat a l'apartat de controladors SDN, Vmware NSX aporta una major agilitat, escalabilitat, millor seguretat, reducció de costos i un més alt rendiment en entorns com Data Centers o proveïdors Cloud, alliberant els nexes amb la xarxa física e implementant el control i la intel·ligència a la capa de software. És l'equivalent a un hipervisor de servidors però en aquest cas en Networking, amb un rang operacional que va des de la capa 2 a la capa 7 del model OSI.

Vmware va ser pioner en la virtualització de servidors x86 a través de solucions com ESX i posteriorment ESXi. La virtualització de servidors permetia la independència del maquinari amb tot el que comporta, com la reducció de costos. Amb l'aparició del Programari Defined Datacenter (SDDC) la virtualització es va estendre a la resta de recursos dels Datacenter, el que va donar origen a SDN. Vmware va veure una oportunitat en això llançant el seu aplicatiu NSX, nomenat com el més visionari per Gartner, en el DataCenter Networking.

A diferència de la seva anterior aplicació per a xarxes virtuals VSphere, els switches són virtuals i ofereixen una configuració i una administració centralitzada mitjançant VCenter.

Amb aquesta tecnologia es poden realitzar múltiples tasques sense la necessitat d'exercir cap canvi en la infraestructura física, entre elles es poden trobar:

- Creació de xarxes mitjançant switchos lògics a través de una VXLAN Network Identifier (VNI).
- Routers lògics para la comunicació entre maquines virtuals allotjades tant en un mateix host como en distints hosts (fora de la plataforma NSX).
- Servicios de xarxa (sense la necessitat de modificar la infraestructura de xarxa física) com Firewalls, NAT, VPN, etc.

**Funcionament**

NSX utilitza els dispositius que formen la infraestructura de xarxa ja existent, per la qual cosa no és necessari invertir en nous equips. A aquesta infraestructura es connecten els servidors ESXi, els quals implementen switches virtuals.

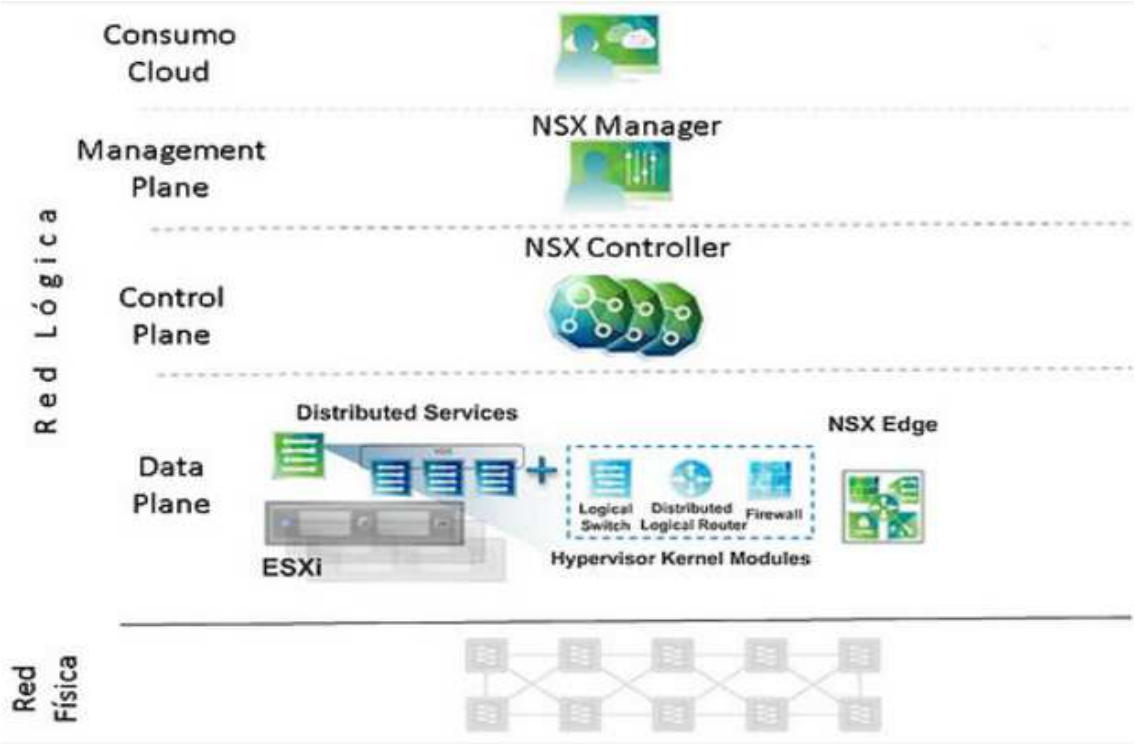



Figura 32 Diagrama NSX

La virtualització de la xarxa es realitza mitjançant una xarxa lògica, proveint serveis de xarxa i administrant la connexió de màquines virtuals tant a xarxes lògiques com a físiques. Aquesta xarxa lògica consta de les següents capes:

- Pla d'infraestructura (Data Plane) on es situen els servidors i commutadors virtuals.
- Pla de control (Control Plane) destinada als controladors.
- Pla de gestió (Management Plane).
- Consum del cloud.

**Pla de infraestructura**

Consistent en els següents components:

- VMware Distributed Switch: determina els serveis de la capa 2 i sobre ell es recolzen la resta de serveis NSX. Permet crear un servei de capa 2 que pugui abastar múltiples hosts, independentment que es trobin connectats a una xarxa de transport de capa 2 o de capa 3.

- Hypervisor Kernel Modulis: s'instal·len en cada switch distribuït en un procés conegut com ‘Host Preparation’. En aquest procés s'instal·len diversos vSphere Installation Bundles (VIB), els tres principals són els següents:

    - Distributed Logical Router (DLR): és l'encarregat de gestionar el procés de Routing dins de l'entorn NSX d'Est a Oest entre les VLAN’s o les VXLAN’s. En el supòsit d'una comunicació entre dues màquines virtuals allotjades en un mateix host però en diferents xarxes de capa 2, no seria necessari que el tràfic utilitzés un dispositiu de capa 3 ja que el procés de Routing es faria a nivell de NSX, tal com es mostra a la següent figura.

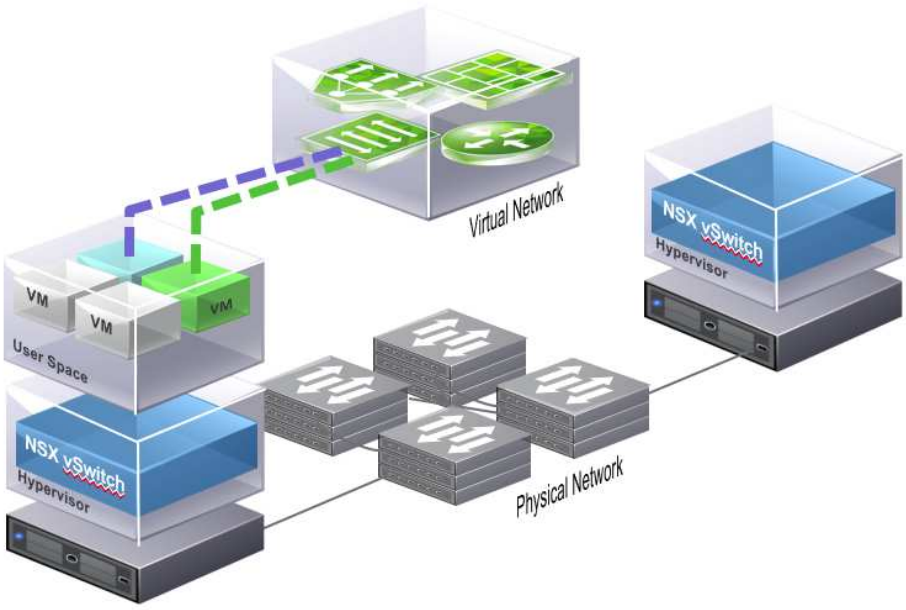



Figura 33 Procés Distributed Logical Router NSX 2 VM's 1 Host

El mateix passaria en el supòsit d'una comunicació entre dues màquines virtuals allotjades en diferents hosts i en diferents xarxes de capa 2, tot i que el tràfic de dades sí que en aquest cas utilitzaria la xarxa física, no es dependrà de cap dispositiu de capa 3 físic per realitzar el Routing. Aquest procés es durà a terme mitjançant el Distributed Logical Router tal com es pot observar a la següent figura.

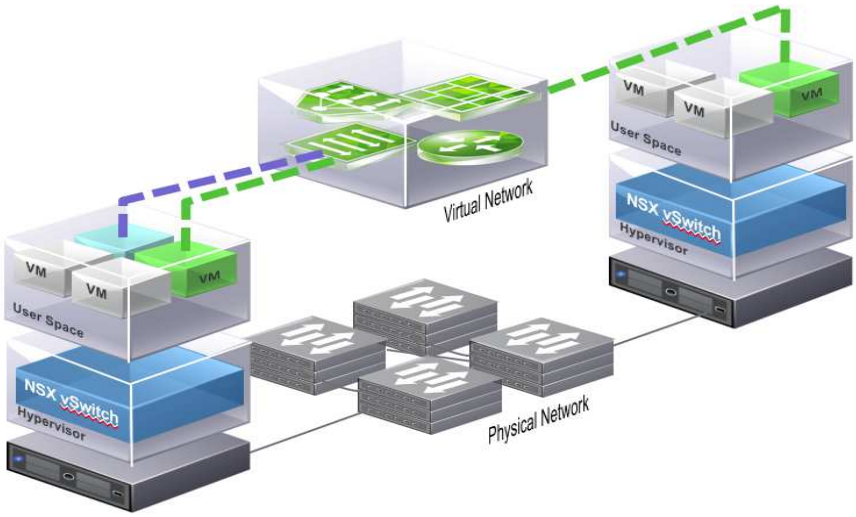



Figura 34 Procés Distributed Logical Router NSX 2 VM's 2 Hosts

- Distributed Firewall: controla el tràfic entre les VM's (Aquest-Oest), solució basada en un agent que executa cada servidor ESXi de l'entorn NSX.

- VXLAN: protocol de superposició de xarxes que implementa segments LAN similars a les VLAN's. Utilitza una encapsulació diferent que li fa possible augmentar el nombre de xarxes possibles de transportar afegint flexibilitat a la implementació de xarxes de capa 2.

- NSX Virtual Switch: Una vegada assignades els VIB's al switch distribuït en el procés de preparació del host, aquest passa a definir-se com switch virtual.

- NSX Edge Services Gateway: NSX Edge és una màquina virtual que ofereix diversos serveis des de la capa 3 fins a la capa 7, d'entre els quals, es troben els següents:

  - Routing: S'encarrega d'encaminar el tràfic de Nord a Sud i d'Est a Oest (tot i que el DLR és millor en aquest darrer cas). Pot ser estàtic o dinàmic (BGP/OSPF/ ISIS).
  - Firewall: Permet crear regles per controlar el tràfic de Nord a Sud entre els segments lògics a partir d'IP's, objectes de VCenter, ports o grups.
  - Network Address Translation (NAT): S'encarrega de gestionar la traducció o l'emmascarament d'adreces IP.
  - DHCP: Realitza les funcions de servidor d'adreces.
  - Load Balancing: Suporta Inline i One-Armed com a mètodes de balanceig.
  - Virtual Private Network (VPN): suporta Site-to-Site, SSL VPN (Remote Access) i L2VPN (VPNs de capa 2).
  - DNS Relay: Permet tenir un cache temporal perquè futures consultes puguin ser contestades.

**Pla de control**

En aquest pla es poden trobar els següents components:

- NSX Logical Router Control Virtual Machine: és l'encarregat de controlar el DLR del pla de dades, entaulant la comunicació entre Routers i formant les taules d'encaminament. Una vegada generades aquestes taules són distribuïdes a cada ESXi, d'aquesta forma es procedeix a controlar i encaminar el tràfic.
- NSX Controller: és una màquina virtual desplegada pel NSX Manager i pot ser construïda en un clúster de nodes. És capaç d'eliminar la necessitat de configurar IGMP i/o PIM de la xarxa física, depenent de la forma de replicació. S'encarrega d'administrar els mòduls Logical Switch i Logical Router dels ESXi, també emmagatzema les taules de VTEPs, de MACs, ARP i d'encaminament. Pot dur a terme l'eliminació dels missatges de broadcast fent-se valer de les seves taules d'adreces. Compte a més amb els següents rols:

    - Distribució de les VXLAN(Logical Switch).
    - Informació de xarxa d'encaminament lògic (Logical Router).

  Tots dos rols són distribuïts a través del Clúster (NSX Controller pot ser desplegat mitjançant diverses instàncies formant un Clúster).
- User World Agent (UWA): s'instal·la en cada host ESXi mitjançant el component del pla de gestió NSX Manager a través de l'agent de bus de missatges. Intervé en les comunicacions entre els següents components:
    - Entre NSX Controller i els mòduls Kernel VXLAN i Kernel Distributed Firewall mitjançant el servei Ntcpad.
    - Entre NSX Manager i el món de Kernel Distributed Firewall mitjançant el servei Vsfwd.
    - Entre NSX Controller i els host ESXi.
    - Entre NSX Manager i els host ESXi.
    - La comunicació entre ell mateix i NSX Controller, la qual es duu a terme a través del port TCP 443 mitjançant connexió segura SSL.

  S'encarrega a més de recuperar informació del NSX Manager a través de l'agent de bus de missatges.

**Pla de gestió**

En aquest pla es poden trobar els següents components:

- NSX Manager: és el component motor de NSX el qual permet configurar tota la solució de NSX for vSphere, per exemple, preparar VXLAN, lliurar controladors, serveis de tercers, etc. NSX manager és un appliance virtual que s'importa a través d'un OVA. Per esmentar algunes de les funcions que NSX manager té:

    - Proporciona l'ambient gràfic "UI" i la NSX API.
    - Instal·la el software i els virtual appliances necessaris per NSX.

- vCenter Server: es connecta amb NSX Manager en una relació 1:1 i és la principal eina per configurar i gestionar el centre de dades virtual. Conté un servei que pot ser usat per a connexió remota, JMX RMI. Qualsevol modificació que es vulgui realitzar a nivell de xarxa ha de ser feta directament en aquest component.
- Message Bus Agent (RabbitMQ): és l'agent del pla de gestió i es comunica amb el UWA per a la seva instal·lació i posterior comunicació entre NSX Manager i els ESXi.

**Avantatges**

- El temps d'aprovisionament de xarxa passa a ser de dies a segons.
- Augment d'eficàcia operacional mitjançant l'automatització.
- Desplaçament i ubicació de càrregues de treball de forma independent a la topologia física.
- Realitzar implementacions a qualsevol hipervisor, a qualsevol plataforma d'administració del núvol.
- Integrar solucions de xarxa i seguretat de tercers mitjançant API's estàndards.

**Protocols**

- VXLAN
- Stateless Transport Tunneling (STT)
- Generic Routing Encapsulation (GRE)

**Requisits NSX**

- Connectivitat IP entre hosts.
- MTU 1600 bytes (la xarxa de transport a l'efecte de VXLAN ha de ser capaç d'operar amb paquets IPS d'aquesta grandària).

### 5.1.2. OpenContrail

És el controlador SDN de codi obert del programari comercial Juniper Networks Contrail, diferenciant-se exclusivament en el servei i suport que ofereix Juniper per la compra del seu paquet de pagament. Les dues versions es basen en un programari de xarxa virtual que corre en servidors x86 permetent la interacció i interconnexió de xarxes físiques i virtuals, aconseguint d'aquesta manera agilitat (serveis més ràpids), escalabilitat i augment de control, fent-les més fàcil d'administrar i organitzar.

Depenent de les necessitats del negoci, Opencontrail ofereix l'aprovisionament de serveis mitjançant la connexió dels mateixos entre diferents dispositius. Redueix a la vegada el temps de sortida al mercat de nous productes i serveis gràcies a la versatilitat i automatització en la creació de xarxes virtuals, les quals interconnecten sistemes Cloud públiques i privades.

**Funcions**

Serveis de porta d'enllaç: connecta perfectament amb routers i switchos Juniper tant amb les càrregues de treball heretades com amb els serveis físics no virtualitzats. Opera també amb la majoria d'equips de routers compatibles amb L3VPN o I-VPN.

VPN flexible i resistent: Ofereix L3VPN, I-VPN, IPSec punt a punt i VPN.

Balanceig de càrrega: Integrat directament en el pla de reenvio per al balanceig de càrrega del tràfic en diferents nivells d'aplicacions o serveis de xarxa.

Commutació i encaminament: El pla de reenvio proporciona encaminament i commutació en un entorn virtualitzat per a múltiples usuaris totalment desvinculat de l'estructura de commutació física.

Seguretat: Aplicació de polítiques i grups de seguretat integrats directament en el pla de reenvio, amb serveis de tallafocs, reconeixement d'aplicacions i prevenció d'amenaces distribuïdes.

Alta disponibilitat: OpenContrail està configurat com actiu-actiu i cada vRouter obté la mateixa taula d'encaminament i llistes ACL.

Serveis d'anàlisis: Visualització avançada i diagnòstic de xarxes físiques i virtualizadas. Proporciona anàlisi d'infraestructura en temps real i històrica que pot ser processada mitjançant API's.

Serveis API: per a la configuració, operació i anàlisi per obtenir una perfecta integració amb sistemes d'organització Cloud, com CloudStack i OpenStack, o sistemes OSS/BSS de proveïdors de serveis. Compatibilitat amb API VPC per a una implementació perfecta de les aplicacions en un entorn híbrid que inclou clouds privades i públiques.

**Compatibilitat**

Opencontrail és compatible amb les plataformes de CloudStack i OpenStack, a més té un acord de contribució amb el projecte OpenDaylight. En general és compatible amb una àmplia gamma de hipervisors, sistemes d'organització i xarxes físiques.

## 5.2. Basats en hardware

Es passa a detallar dos dels més importants.

### 5.2.1. Cisco

Amb la finalitat de seguir promovent la venda d'equips en unes arquitectures clarament definides cap al programari i amb una gran tendència alcista com són SDN i NFV, Cisco ha llançat al mercat una nova línia estratègica nomenada com Insieme Networks. Aquesta resposta, orientada principalment als data Center, tracta de mitigar l'efecte negatiu que provoca en la companyia les principals funcions que desenvolupa la virtualització de xarxes, en separar el control del maquinari del propi equip.

Cisco ja va donar suport de plataforma expandida per OpenFlow amb les sèries Nexus 3000, Nexus 7000, ASR 9000 i Catalyst 6500. Amb Insieme ha desenvolupat la línia Nexus 9000 para data Center i switches Cloud, que al revés del seu competidor principal en virtualització Vmware, pretén que la xarxa sigui conscient de les aplicacions i no al revés com realitzen la majoria dels proveïdors.

Application Centric Infrastructure (ACI) és el producte que Cisco ha dissenyat per promoure la importància de les aplicacions a la xarxa. ACI es defineix com una solució SDN enfocada en les aplicacions de TI. Els switches de la sèrie Nexus 9000 formen part d'aquesta infraestructura.

Figura 35 Nexus 9000

http://www.cisco.com/c/en/us/products/switches/nexus-9000-series-switches/index.html

Nexus 9000 incorpora el sistema operatiu Cisco NX-OS, proporcionant-li unes característiques que ho fan més resistent, altament programable (a través de API's), segur, flexible i escalable, tot això en un entorn molt més fàcil d'usar. Cisco NX-OS és un sistema operatiu modular que admet actualitzacions de programari en calent, el reinici de processos stateful, "parches" de aplicacions tant en fred com en calent i scripts online. Existeix una sola imatge binària en tots els switchos de la sèrie, simplificant d'aquesta forma l'administració d'imatges.

La sèrie Cisco Nexus 9000 suporta els següents protocols:

- SNMPv1, v2 y v3
- Syslog
- RMON

- Protocol de configuració de xarxa (NETCONF)
- CLI y CLI scripting
- XMPP

Amb aquesta sèrie s'aconsegueix una total mobilitat del host, desacoblant-ho de les polítiques de les aplicacions des de la infraestructura IP. La xarxa s'adapta als requisits de les aplicacions mitjançant serveis aplicats que van des de la capa 4 fins a la 7, aconseguint amb tot això una major flexibilitat de forma simplificada. El rendiment es veu considerablement incrementat gràcies a una taxa de transferència de 40Gb arribant fins i tot als 100Gb. Gran escalabilitat amb 60Tbps de capacitat de switching (30Tbps assegurats sense bloqueig), fins a 1152 ports Ethernet o 288 de 40Gbps, compatibilitat amb gateway VXLAN i un rendiment multicast superior al de qualsevol altre switch modular.

Inclou suport per OpenStack (Neutron). Permet als clients crear fàcilment les seves xarxes IaaS oferint simplicitat i les operacions dels serveis en el núvol.

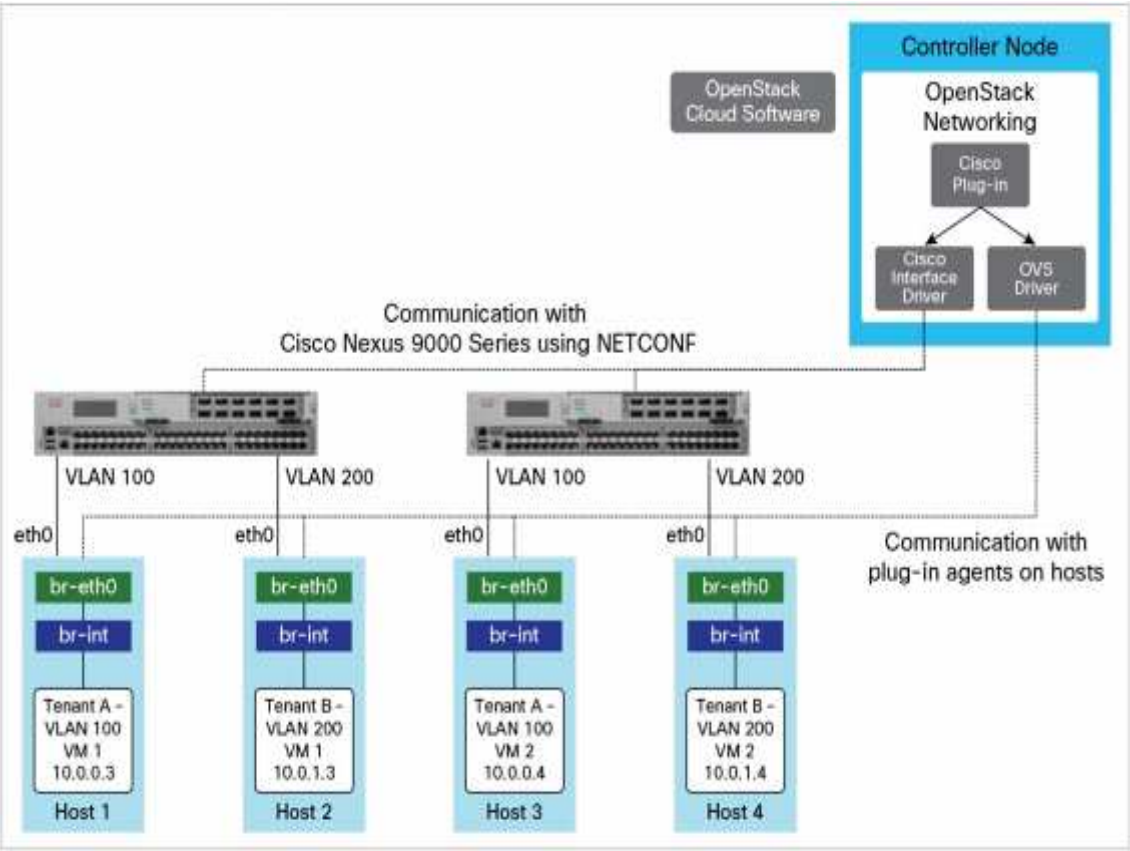



Figura 36 Plug-in Openstack Nexus 9000

Openstack està limitat en certs casos, en el supòsit que diversos clusters s'estiguin executant al mateix temps en diferents hosts d'una o diverses VLAN's, el plug-in podrà configurar la xarxa virtual o la xarxa física, però no ambdues. El plug-in de Cisco resol aquest problema en permetre l'ús de diversos plug-ins de forma simultània. Accepta sol·licituds de la API de OpenStack i configura els switches Nexus amb els paràmetres obtinguts.

Dóna suport total per OpenFlow i facilitarà la integració en el projecte de codi obert OpenDayLight.

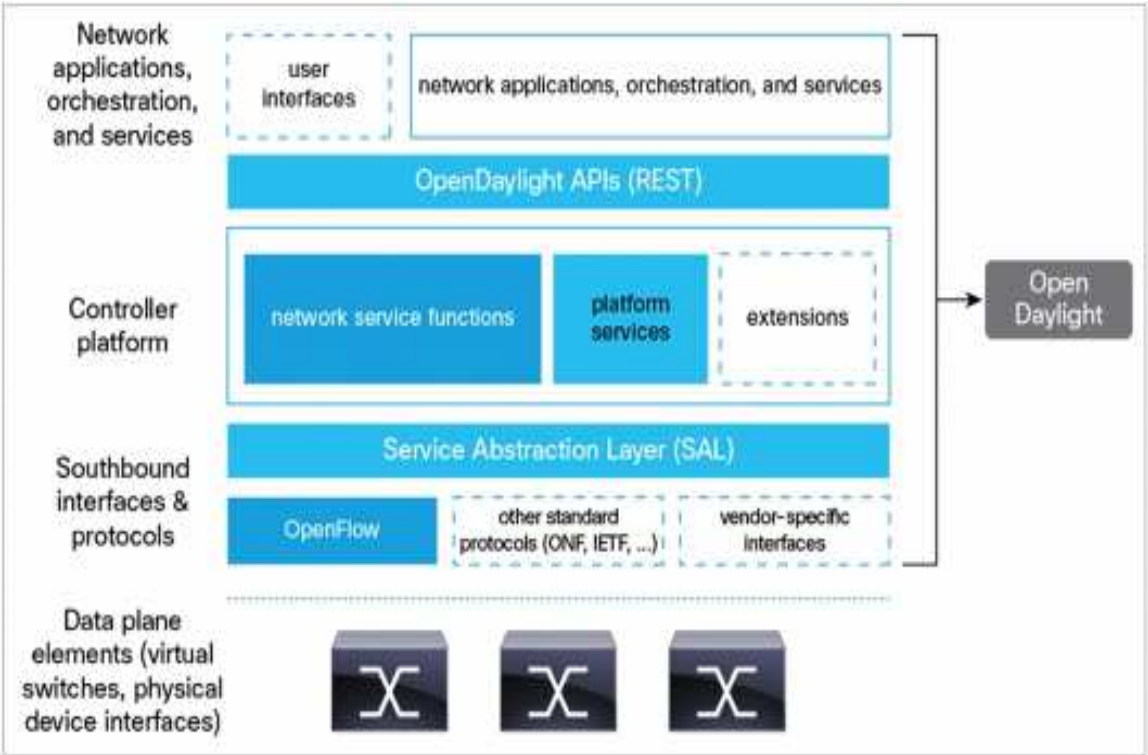



Figura 37 Suport a Opendaylight de Nexus 9000

http://www.cisco.com/c/en/us/products/collateral/cloud-systems-management/aci-fabric-controller/white-paper-c11-729385.html

La família Nexus 9000 consta dels següents equips:

- Cisco Nexus 9300: plataforma compatible amb les funcions de routing, ponts VXLAN i NVGRE en hardware. Switch de 1, 10 i 40 Gigabit Ethernet sense bloqueig amb latència d'1 a 5 microsegons. Connexió mitjançant fibra òptica i cablejat de coure, transceptor bidireccional QSFP de 40 Gigabit Ethernet

connectable. Alta eficiència energètica, el consum energètic típic en una configuració completament carregada és inferior a 3,5 W per port de 10 Gigabit Ethernet i el consum energètic típic per port de 40 Gigabit Ethernet és inferior a 14 W.

- Cisco Nexus 9396PX: Pertany a la sèrie 9300 i està format per dues unitats muntades en rack amb capacitat de retransmissió de capa 2 i 3 de 960Gbps. 48 ports de 10Gbps i 12 ports Enhanced Quad SFP (QSFP+) de 40 Gigabit Ethernet.

- Cisco Nexus 93128TX: Pertany a la sèrie 9300 i està format per tres unitats muntades en rack amb capacitat de retransmissió de capa 2 i 3 de 1280Gbps. 96 ports de 10Gbps (podent-se executar a 100Mb per poder ser compatible amb versions anteriors) i 8 ports Enhanced Quad SFP (QSFP+) de 40 Gigabit Ethernet.

- Cisco Nexus 9500: Mateixes característiques que la sèrie 9300 però amb una plataforma totalment redundant que consta de dos supervisors, dos controladors de sistema, tres safates de ventilador i fins a quatre fonts d'alimentació. Xassís de switchos dissenyat sense pla mitjà

- Cisco Nexus 9508: Pertany a la sèrie 9500 i està format per 13 unitats modulars en un rack (13RU) de 1, 10 i 40Gb Ethernet. Disposa de vuit targetes de línia i sis connexions per a mòduls externs. Switching de fins a 30Tbps i 288 ports 40Gb Ethernet o 1152 ports de 10Gb Ethernet.

### 5.2.2. Juniper

Al igual que Cisco, Juniper Networks ha desenvolupant un nou switch de nucli programable per a xarxes definides per programari (SDN) als data Centers. Anomenat EX9200, el nou switch està basat en el model anterior, la sèrie MX. Simplifica el desplegament d'aplicacions al núvol, la virtualització de servidors i les eines de col·laboració enriquides en entorns d'agregació, centres de dades i campus. Ofereix accés segur i senzill per a la distribució d'aplicacions imprescindibles i simplifica les

operacions per alinear la xarxa segons els constants canvis en els requisits empresarials.

L'EX9200 està basat en el ASIC programable Juniper One i suporta un slot de 240G i targetes d'interfície de 40 de 1Gigabit Ethernet, 32 de 10Gb, 4 de 40Gb i 2 de 100Gb. Les característiques de programació, al costat del ASIC Juniper One, inclouen un conjunt d'eines d'automatització Netconf i XML, i interfícies Puppet, Python i OpenFlow.

| Dispositiu | Versió Openflow | Versió Junos OS | Paquet Openflow |
|---|---|---|---|
| EX4550 Ethernet Switches | v1.0 | 13.2X51-D20 | jsdn-powerpc-release |
| EX9200 Line of Ethernet Switches | v1.0<br>v1.3.1 | 13.3R1<br>14.2R1 | jsdn-i386-release<br>jsdn-x386-release |
| MX80 3D Universal Edge Routers | v1.0<br>v1.3.1 | 13.3R1<br>14.2R1 | jsdn-powerpc-release<br>jsdn-powerpc-release |
| MX240, MX480, and MX960 3D Universal Edge Routers | v1.0<br>v1.3.1 | 13.3R1<br>14.2R1 | jsdn-i386-release<br>jsdn-x386-release |
| MX2010 and MX2020 3D Universal Edge Routers | v1.0<br>v1.3.1 | 15.1R2<br>15.1R2 | jsdn-i386-release<br>jsdn-x386-release |
| vMX 3D Universal Edge Routers | v1.3.1 | 14.2R4 | jsdn-i386-release |

Taula 4 Dispositius que suporten Openflow

El nou switch suportarà a més, plug-ins per a sistemes d'orquestració de VMware i OpenStack, un milió d'adreces MAC, 256.000 rutes, 32.000 VLAN i 256.000 ACL, així com switching de Nivell 2/3, MPLS, VPLS, L3VPN, punt a multipunt, i convergència a 50 milisegons utilitzant MPLS Fast Re-Route. Així mateix pot ser configurat com un Virtual Xassís de quatre switchos físics.

**Sistema operatiu Junos**

Mitjançant el sistema operatiu Junos de Juniper Networks (Junos OS) es poden configurar dispositius com els switchos compatibles amb OpenFlow. Per a això s'utilitza un procés que inclou el sistema operatiu Junos, openflowd (OFD). En els dispositius amb sistema operatiu Junos, es pot aïllar el tràfic OpenFlow configurant un o més switches virtuals que actuen com a dominis lògics separats. El switch virtual i el controlador es comuniquen mitjançant l'intercanvi de missatges de protocol OpenFlow, que el controlador utilitza per afegir, eliminar i modificar els fluxos en el switch.

Figura 38 Serie EX9200 (EX9204-EX9208-EX9214)

http://www.juniper.net/us/en/products-services/switching/ex-series/ex9200/

La seriï EX9200 permet la integració amb Vmware i Openstack. És compatible a més amb una gran gamma de protocols per a la comunicació en data Center i campus (IP, MPLS, IS-IS, VPLS, NVGRE, VXLAN, CAPWAP, GRE, etc.).

Per poder configurar i habilitar OpenFlow es realitzen els següents passos:

- Configurar les interfaces OpenFlow com a interfaces de capa 2.
  [edit interfaces]

  user@switch# set ge-1/0/0 unit 0 family ethernet-switching

  user@switch# set ge-1/1/0 unit 0 family ethernet-switching

  user@switch# set xe-0/0/0 unit 0 family ethernet-switching

  Configure the virtual switch routing instance.

  [edit routing-instances]

  user@switch# set OF-ri instance-type virtual-switch

  user@switch# set OF-ri interface ge-1/0/0.0

  user@switch# set OF-ri interface ge-1/1/0.0

  user@switch# set OF-ri interface xe-0/0/0.0

  user@switch# set OF-ri vlans of-bridge vlan-id none

- Configurar l'adreça IP del controlador OpenFlow y el protocol de connexió.

  [edit protocols openflow switch OFswitch1]

  user@switch# set controller address 198.51.100.174

  user@switch# set controller protocol tcp port 6633

- Configurar les interfases lògiques que intervenen a OpenFlow sota la instancia del switch virtual.

  [edit protocols openflow switch OFswitch1]

  user@switch# set interfaces ge-1/0/0.0

  user@switch# set interfaces ge-1/1/0.0

  user@switch# set interfaces xe-0/0/0.0

- Configurar la acció predeterminada per als paquets que no tenen una entrada de flux coincident.

  [edit protocols openflow switch OFswitch1]

  user@switch# set default-action packet-in

  Configure OpenFlow traceoptions.

  [edit protocols openflow]

  user@switch# set traceoptions flag all

  Commit the configuration.

  [edit]

  user@switch# commit

**Característiques serie 9200**

| | **EX9204** | **EX9208** | **EX9214** |
|---|---|---|---|
| Slots | 4 | 8 | 14 |
| Fabric | fins a 3,2 Tbps | fins a 9,6 Tbps | fins a 13,2 Tbps |
| Switch Fabrics | 2/1 | 2/1 | 3/2 |
| Routing Engines | 2/1 | 2/1 | 2/1 |
| 1Gb Ethernet ports | 80/120 | 240/280 | 440/480 |
| 10Gb Ethernet ports | 64/96 | 160/160 | 320/320 |
| 40Gb Ethernet ports | 8/12 | 24/28 | 44/48 |
| 100Gb Ethernet ports | 4/6 | 10/10 | 20/20 |

Taula 5 característiques EX9200 configuració redundant/no redundant

# CAPÍTOL 6

# 6. CASOS D'US

## 6.1. Cas d'ús SDN

La implementació de SDN en un centre de dades (datacenter) d'una empresa és un clar exemple de cas d'ús. A l'entorn dels data center, l'operativa de la capa d'aplicacions no té el coneixement de la xarxa subjacent, resultant més complex l'aprovisionament de la xarxa així com la gestió dels serveis.

SDN millora l'organització, gestió i control de la xarxa , la qual cosa ajuda als enginyers i administradors de xarxa a respondre ràpidament als canvis en els requisits del negoci. A través de SDN, els administradors dels centres de dades poden controlar el flux de tràfic des d'un lloc centralitzat, la qual cosa elimina la necessitat de registres i actualitzacions manuals dels switchos individuals. El flux, les prioritats i, fins i tot, la seguretat del tràfic de dades es poden definir centralment i distribuir a tots els switchos de forma ràpida i eficient.

En un Datacenter, diversos casos d'ús poden ser agrupats en una categoria denominada panell de connexió virtual, entre els quals es troben els següents:

- Simplificar les comunicacions de les VM: Els grans Datacenter, com els utilitzats per Amazon i Google, serien extremadament difícil de manejar sense una estratègia definida per software. Amb SDN, a les màquines virtuals se'ls permet comunicar-se entre si sense preocupar-se de la xarxa subjacent. Això simplifica el desplegament de VM's alhora que redueix les despeses operatives.
- Connexió entre els Datacenter empresarials: Extrapolant el model de comunicacions de les màquines virtuals, a través de múltiples centres de dades. SDN maneja dinàmicament el flux de dades sobre un gran ample de banda i baixa latència. El resultat és el de serveis de xarxa més resistents que funcionen sobre la xarxa abstracta, i un accés més ràpid amb un baix control de costos.

- Integració de les xarxes heretades: Per a les xarxes heretades o centres de dades multi-tenant, SDN simplifica la gestió dels sistemes heretats alhora que millora el rendiment. El tràfic de dades es pot dirigir als balancejadors de càrrega i firewalls instal·lats per millorar el rendiment de la xarxa, sense incórrer en despeses de personal per canviar les configuracions de maquinari de forma manual.
- Xarxes sense fils: SDN es pot utilitzar per incorporar fonts de dades sense fils a l'empresa. SDN pot abastar tant els senyals Wifi com les connexions per cable, sobretot perquè la nova generació de controladors de LAN sense fils ofereixen major seguretat i accés, aplicant els controls de seguretat i accés en cadascun dels tipus.

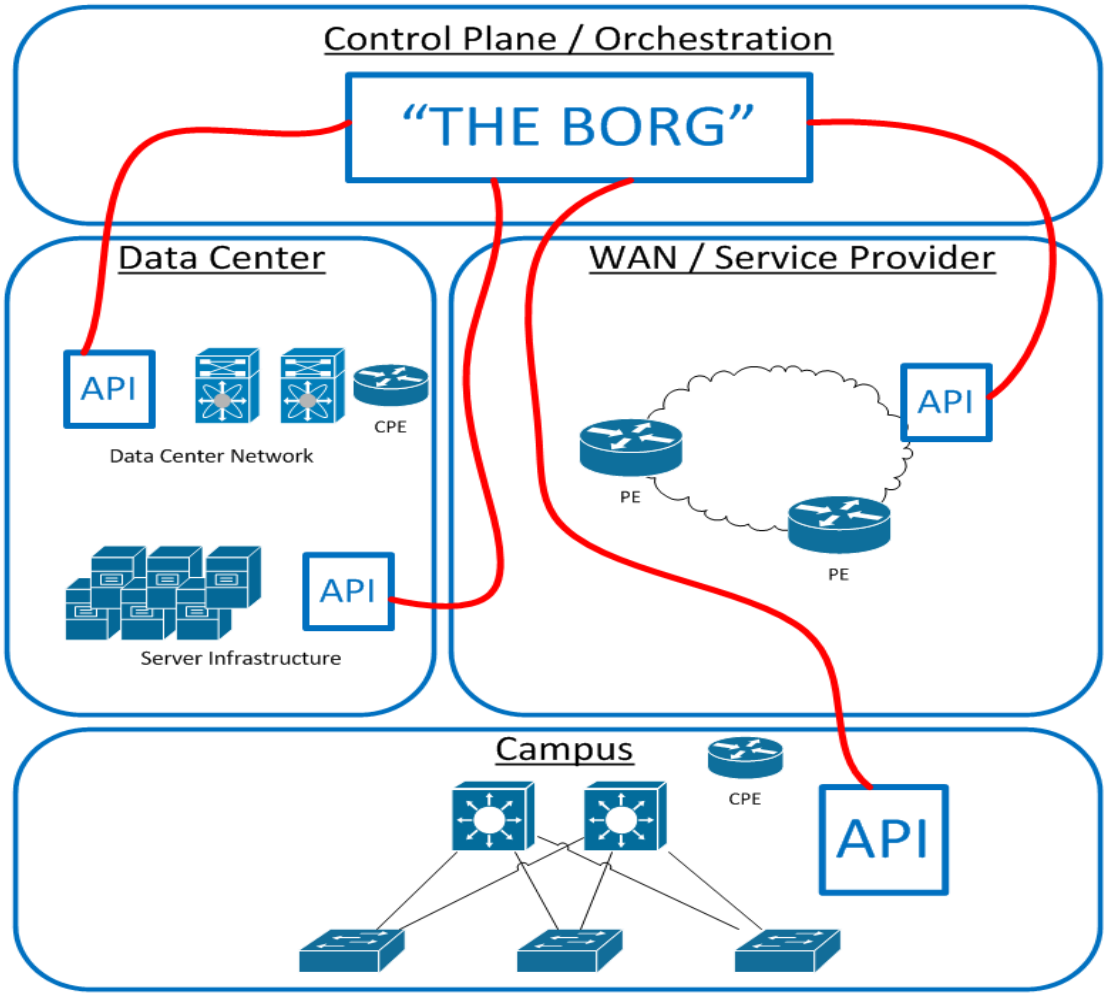



Figura 39 Aplicacions SDN

http://keepingitclassless.net/2013/05/sdn-use-case-end-to-end-qos/

L'escalabilitat en els data center és extremadament difícil, SDN simplifica el problema en permetre que les màquines virtuals (VM) es comuniquin entre si sense deixar de ser conscient de la xarxa subjacent. Això augmenta significativament la facilitat amb la qual les màquines virtuals poden ser desplegades i reorientades als centres de dades, i redueix els costos mitjançant la millora de la utilització d'actius i la reducció de despeses d'operacions.

## 6.2. Casos d'ús NFV

L'automatització de les funcions de xarxa i els serveis és el punt fort que NFV ofereix al mercat.

**Cas 1**

En aquest cas d'ús NFV, es descriu l'encadenament de serveis mitjançant dispositius virtuals de manera flexible. Un usuari, una aplicació, o un flux de contingut han de passar per diversos dispositius virtuals abans de ser lliurats, això es coneix com a encadenament de serveis.

L'enfocament basat en maquinari actual fa d'aquest procés un tema molt complex, requereix molt temps, més costos, difícil d'escalar i administrar. Els dispositius han de ser instal·lats i cablejats físicament, s'han d'assignar dominis físics tals com VLAN's, subxarxes, etc., que normalment limiten la seva connectivitat. La configuració ha de ser manual i meticulosa per implementar la cadena de serveis.

Se suposa un flux procedent d'un punt extrem A, el qual passa a través de diversos dispositius virtuals (NFV en aquest cas), una xarxa de monitoratge, un balanceig de càrrega i un firewall abans d'arribar al punt de destinació B. En un entorn NFV, aquest encadenament de serveis pot ser creat, actualitzat, escalat, i eliminat d'una forma molt més ràpida i eficient. Per agregar una nova funció o servei per exemple, es pot realitzar mitjançant la creació de noves instàncies i tot el procés s'actualitza.

**Cas 2**

Un altre cas d'ús NFV promogut pel ETSI és NFVIaaS, requerit per a la prestació de serveis en el núvol. En aquest cas d'ús, un proveïdor de serveis pot oferir serveis que utilitzen la infraestructura NFV d'un altre proveïdor de serveis. Aquest enfocament pot ampliar enormement l'abast d'un servei en llocs on no manté cap actiu de xarxa físic.

En aquest exemple, un proveïdor de serveis ofereix un servei de balanceig de càrrega virtualitzada. Alguns dels clients de suport d'aquest proveïdor necessiten serveis de balanceig de càrrega en llocs en els quals l'empresa no manté NFV, però on un altre proveïdor de serveis si els té.

NFVIaaS ofereix un mitjà per tal que aquest segon proveïdor pugui cedir la infraestructura NFV (còmput, xarxa, hipervisors, etc.) per donar servei al primer proveïdor, que dóna l'últim accés a la infraestructura que d'una altra manera seria molt costós d'obtenir. Aquesta capacitat està disponible a demanda, i es pot ampliar segons sigui necessari.

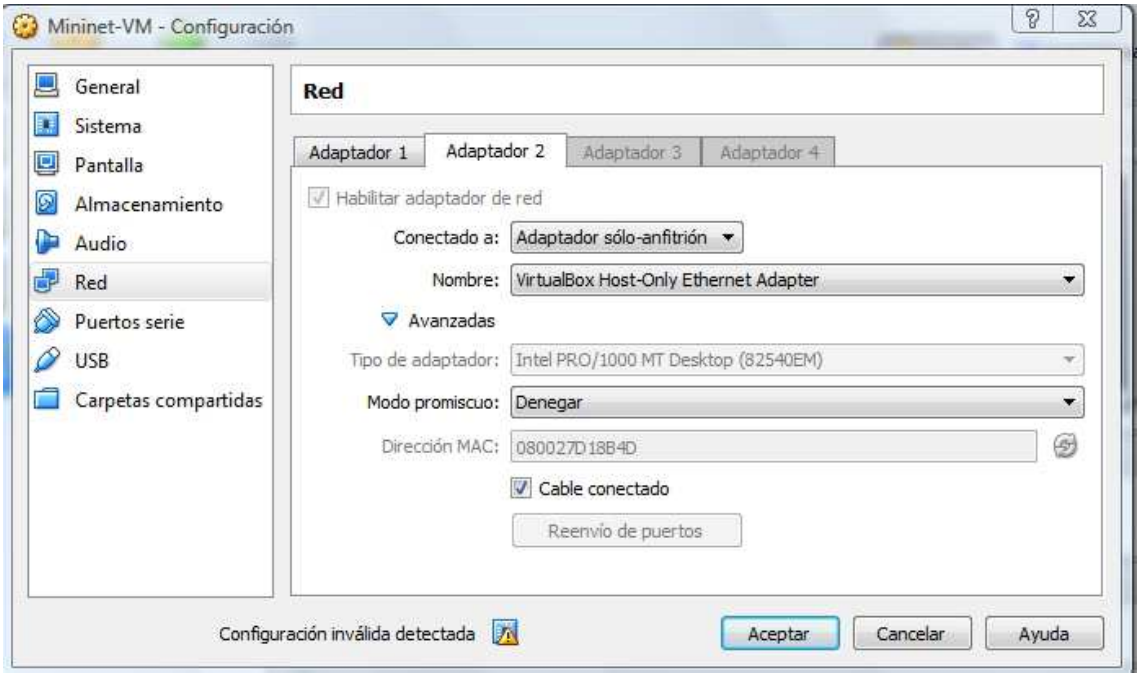
La proposta de valor per a aquest cas d'ús NFV és significatiu. En eliminar el cost i la complexitat de la implementació de nous serveis fixos de maquinari o el seu arrendament, el proveïdor pot desplegar i escalar els serveis virtualitzats ràpidament, i ampliar l'abast a través d'altres proveïdors de serveis.

No tots els proveïdors de serveis compten amb la infraestructura suficient per desenvolupar els seus recursos arreu del món. Gràcies a aquesta tecnologia, tant els proveïdors que no disposen del servei en una ubicació com els quals ho ofereixen en el seu lloc, es veuen beneficiats.

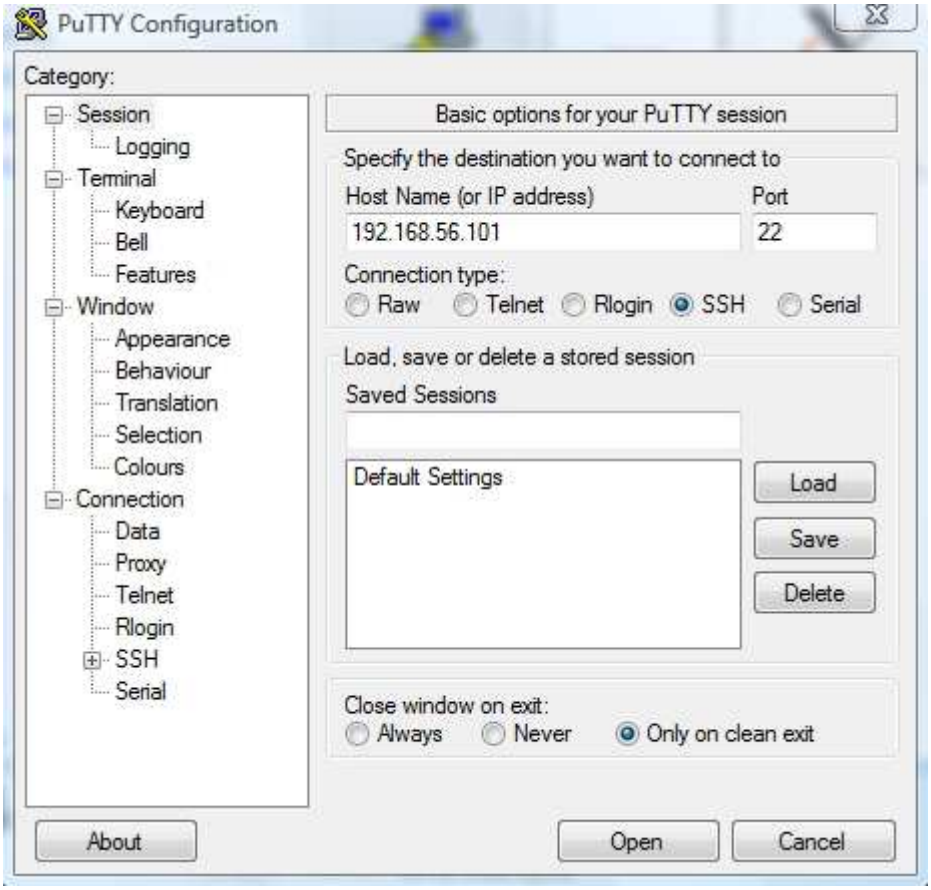
# CAPÍTOL 7

## 7. MININET

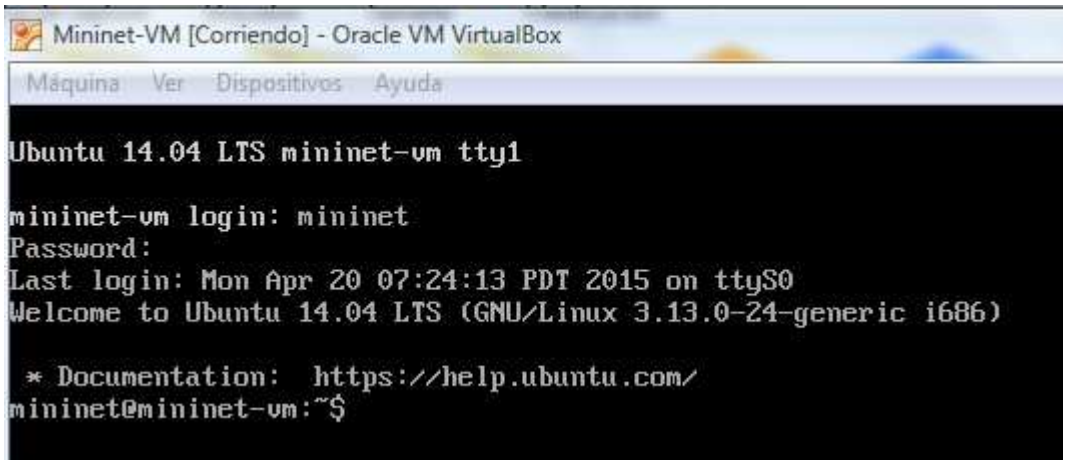
Per tal de pogué experimentar amb una xarxa SDN, es farà servir l'emulador Mininet, capaç d'executar hosts, switchos i controladors des d'un entorn Linux. Per fer-ho, hi ha disponible una màquina virtual 'Mininet' la qual una vegada descarregada e importada a l'entorn de Virtual Box, es procedeix a afegir un segon adaptador del tipus 'Host-Only adapter' tal i com es mostra a la següent imatge.




El següent pas és el de instal·lar i configurar dos aplicacions que seran necessàries per establir les comunicacions amb terminals mitjançant connexions SSH, Putty i Xming. Una vegada instal·lades, s'executa la VM Mininet, l'aplicació Xming (resta resident) i Putty.exe, el qual es configura amb l'adreça de la connexió 'Host-only adapter' que per defecte és la 192.168.56.101 i el port 22.




Els accessos per la màquina virtual, tant el login com per la contrasenya, és ‘mininet’.




Executant un ‘ifconfig –a’ es pot observar com s’han creat dos adaptadors de xarxa, eth0 que equival a ‘Host-only adapter’ i que té assignada la IP 192.168.56.101 que ja s’havia establert al programa de connexió ‘Putty’, i eth1 que correspon a l’adaptador de xarxa NAT, on es comprova que no té assignada cap adreça IP.

```
mininet@mininet-vm:~$ ifconfig -a
eth0      Link encap:Ethernet  HWaddr 08:00:27:d1:8b:4d
          inet addr:192.168.56.101  Bcast:192.168.56.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:251 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:26708 (26.7 KB)  TX bytes:684 (684.0 B)

eth1      Link encap:Ethernet  HWaddr 08:00:27:1c:5e:1c
          BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:272 errors:0 dropped:0 overruns:0 frame:0
          TX packets:272 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:21696 (21.6 KB)  TX bytes:21696 (21.6 KB)

mininet@mininet-vm:~$
```

A la següent imatge s'executa l'ordre 'sudo dhclient eth1' per assignar una IP a la interfície eth1. Es pot comprovar que l'adreça assignada és la 10.2.15.

```
mininet@mininet-vm:~$ sudo dhclient eth1
mininet@mininet-vm:~$ ifconfig -a
eth0      Link encap:Ethernet  HWaddr 08:00:27:d1:8b:4d
          inet addr:192.168.56.101  Bcast:192.168.56.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:616 errors:0 dropped:0 overruns:0 frame:0
          TX packets:4 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:61236 (61.2 KB)  TX bytes:1086 (1.0 KB)

eth1      Link encap:Ethernet  HWaddr 08:00:27:1c:5e:1c
          inet addr:10.0.2.15  Bcast:10.0.2.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:14 errors:0 dropped:0 overruns:0 frame:0
          TX packets:13 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2466 (2.4 KB)  TX bytes:1592 (1.5 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:322 errors:0 dropped:0 overruns:0 frame:0
          TX packets:322 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:26260 (26.2 KB)  TX bytes:26260 (26.2 KB)

mininet@mininet-vm:~$
```

A continuació s'executa una topologia de xarxa amb els següents paràmetres:

--topo: representa la topologia de la xarxa que en aquest cas és 'single' amb 3 hosts.

--mac: activa un sistema de adreces MAC senzill.

--switch: selecciona els diferents tipus de switchos, en aquest cas és ‘ovsk’, switch per defecte preinstal·lat a la VM.

--controller: selecciona el tipus de controlador, a l’exemple es fa servir un controlador remot.

```
mininet@mininet-vm:~$ sudo mn --topo=single,3 --mac --switch=ovsk --controller=r
emote
*** Creating network
*** Adding controller
Unable to contact the remote controller at 127.0.0.1:6633
*** Adding hosts:
h1 h2 h3
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1) (h3, s1)
*** Configuring hosts
h1 h2 h3
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Starting CLI:
mininet>
```

Amb el comando ‘nodes’ es llisten els dispositius virtuals que formen la topologia creada, a l’exemple es poden veure un controlador c0, un switch s1 i tres hosts h1, h2 i h3.

```
mininet> nodes
available nodes are:
c0 h1 h2 h3 s1
```

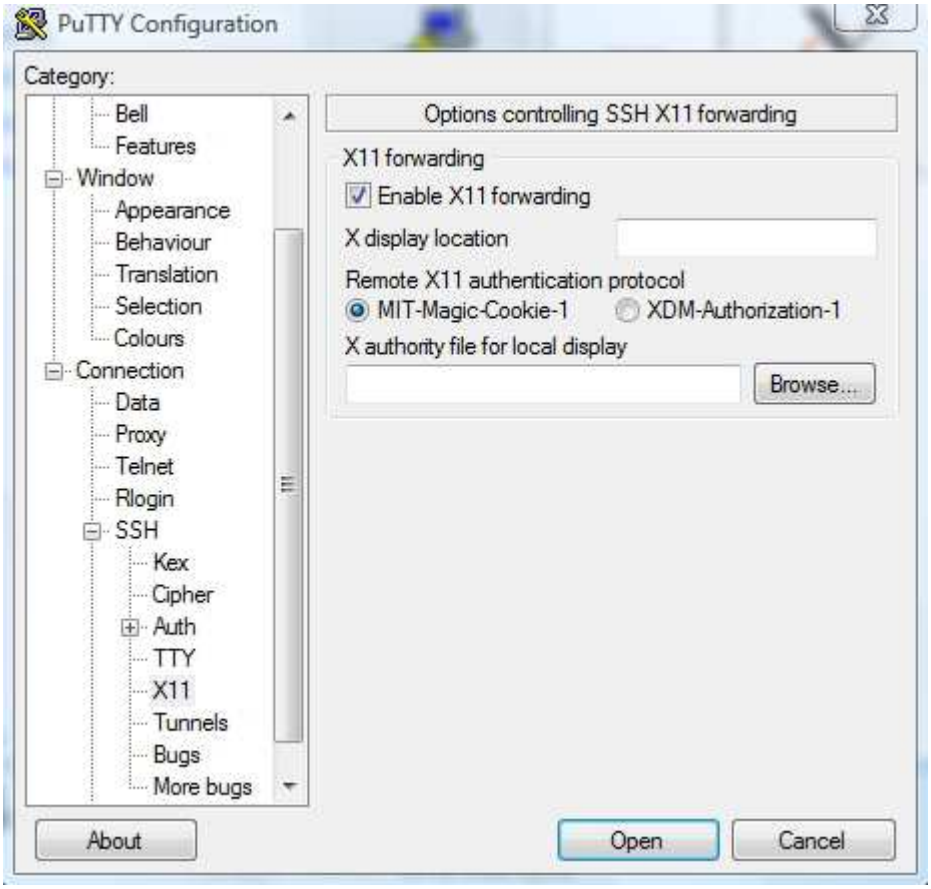
Amb el comando ‘net’ es mostren tots els links.

```
mininet> net
h1 h1-eth0:s1-eth1
h2 h2-eth0:s1-eth2
h3 h3-eth0:s1-eth3
s1 lo:  s1-eth1:h1-eth0 s1-eth2:h2-eth0 s1-eth3:h3-eth0
c0
```

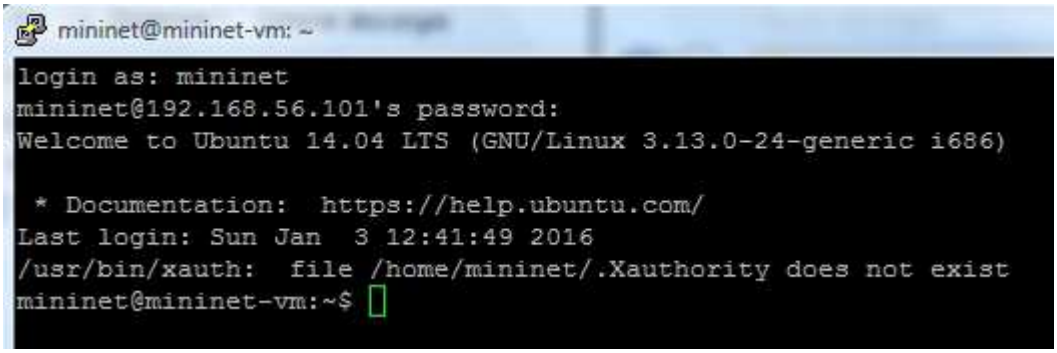
Amb el comando ‘dump’ es mostren les adreces de cada host i els identificadors dels processos que fan servir, entre d’altres.

```
mininet> dump
<Host h1: h1-eth0:10.0.0.1 pid=1568>
<Host h2: h2-eth0:10.0.0.2 pid=1572>
<Host h3: h3-eth0:10.0.0.3 pid=1574>
<OVSSwitch s1: lo:127.0.0.1,s1-eth1:None,s1-eth2:None,s1-eth3:None pid=1579>
<RemoteController c0: 127.0.0.1:6633 pid=1562>
```

Una vegada iniciat Mininet es pot continuar fent les probes mitjançant un terminal SSH i seguint amb l'aplicació 'putty.exe'. S'activa la casella 'Enable X11 forwarding' i es prem 'open' obrint en aquest cas el terminal assignat.




El nou terminal està vinculat a Mininet mitjançant l'adreça 192.168.56.101. S'identifica el login i contrasenya i s'accedeix.



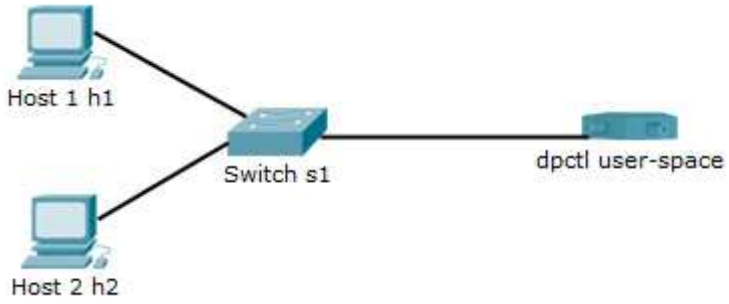
Per tal de crear una nova topologia de xarxa s'executa el comando 'sudo mn –c' on el paràmetre introduït fa un esborrat de les anteriors configuracions.

```
mininet@mininet-vm:~$ sudo mn -c
*** Removing excess controllers/ofprotocols/ofdatapaths/pings/noxes
killall controller ofprotocol ofdatapath ping nox_core lt-nox_core ovs-openflowd
 ovs-controller udpbwtest mnexec ivs 2> /dev/null
killall -9 controller ofprotocol ofdatapath ping nox_core lt-nox_core ovs-openfl
owd ovs-controller udpbwtest mnexec ivs 2> /dev/null
pkill -9 -f "sudo mnexec"
*** Removing junk from /tmp
rm -f /tmp/vconn* /tmp/vlogs* /tmp/*.out /tmp/*.log
*** Removing old X11 tunnels
*** Removing excess kernel datapaths
ps ax | egrep -o 'dp[0-9]+' | sed 's/dp/nl:/'
***  Removing OVS datapaths
ovs-vsctl --timeout=1 list-br
ovs-vsctl --timeout=1 list-br
*** Removing all links of the pattern foo-ethX
ip link show | egrep -o '([-_.[:alnum:]]+-eth[[:digit:]]+)'
ip link show
*** Killing stale mininet node processes
pkill -9 -f mininet:
*** Shutting down stale tunnels
pkill -9 -f Tunnel=Ethernet
pkill -9 -f .ssh/mn
rm -f ~/.ssh/mn/*
*** Cleanup complete.
mininet@mininet-vm:~$
```

A la següent imatge es pot observar la creació d'una nova topologia en format d'arbre. On 'depth=1' indica el nivell de profunditat amb un switch s1 i 'fanout=2' dos hosts que pengen d'ell, h1 i h2. Tant el switch com el controlador són en aquest cas per defecte.

```
mininet@mininet-vm:~$ sudo mn --switch ovs --controller ref --topo tree,depth=1,
fanout=2
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(s1, h1) (s1, h2)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Starting CLI:
mininet>
```

A la següent imatge es pot veure la topologia formada anteriorment.



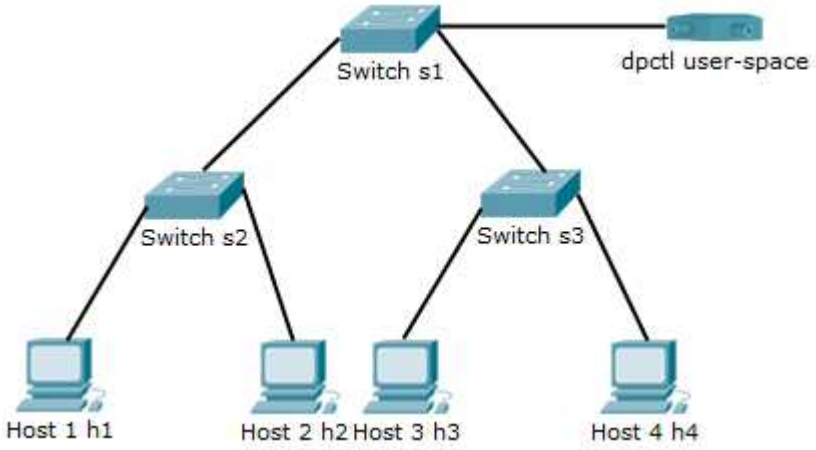

Es pot comprovar la comunicació fent un ping entre els hosts.

```
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2
h2 -> h1
*** Results: 0% dropped (2/2 received)
mininet>
```

La següent topologia està formada per un arbre de profunditat 2 amb 2 hosts cada switch del segon nivell. El controlador és remot, d'aquesta forma la xarxa resultant actua com una xarxa SDN.

```
mininet@mininet-vm:~$ sudo mn --switch ovs --controller remote --topo tree,depth
=2,fanout=2
*** Creating network
*** Adding controller
Unable to contact the remote controller at 127.0.0.1:6633
*** Adding hosts:
h1 h2 h3 h4
*** Adding switches:
s1 s2 s3
*** Adding links:
(s1, s2) (s1, s3) (s2, h1) (s2, h2) (s3, h3) (s3, h4)
*** Configuring hosts
h1 h2 h3 h4
*** Starting controller
c0
*** Starting 3 switches
s1 s2 s3 ...
*** Starting CLI:
```

La topologia es mostra gràficament a la següent imatge, es poden observar al primer nivell un switch s1 que es connecta amb els switchos s2 i s3. Cadascun d'ells (formant el segon nivell de profunditat) tenen assignats dos hosts, h1 i h2 a s2, h3 i h4 a s3. Cal recordar que el nivell de profunditat el defineix el comando 'depth=X' i el nombre de hosts a cada switch el defineix el comando 'fanout=X'.




Al fer un ping, els hosts no es poden veure ja que el controlador remot no està definit.

```
mininet> pingall
*** Ping: testing ping reachability
h1 -> X X X
h2 -> X X X
h3 -> X X X
h4 -> X X X
*** Results: 100% dropped (0/12 received)
```

Per crear un controlador es pot fer des d'un altre terminal, executant el comando 'controller ptcp:'

```
login as: mininet
mininet@192.168.56.101's password:
Welcome to Ubuntu 14.04 LTS (GNU/Linux 3.13.0-24-generic i686)

 * Documentation:  https://help.ubuntu.com/
Last login: Sun Jan  3 04:49:03 2016 from 192.168.56.1
mininet@mininet-vm:~$ controller ptcp:
```

Al tornar a fer el ping, els hosts ja es poden comunicar amb unes regles de replicació assignades automàticament.

```
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 h3 h4
h2 -> h1 h3 h4
h3 -> h1 h2 h4
h4 -> h1 h2 h3
*** Results: 0% dropped (12/12 received)
mininet>
```

A continuació veiem l'estat dels switchos amb la funció 'dpctl show'. Switch 1 s1:

```
mininet> dpctl show
*** s1 ----------------------------------------------------------------------
OFPT_FEATURES_REPLY (xid=0x2): dpid:0000000000000001
n_tables:254, n_buffers:256
capabilities: FLOW_STATS TABLE_STATS PORT_STATS QUEUE_STATS ARP_MATCH_IP
actions: OUTPUT SET_VLAN_VID SET_VLAN_PCP STRIP_VLAN SET_DL_SRC SET_DL_DST SET_N
W_SRC SET_NW_DST SET_NW_TOS SET_TP_SRC SET_TP_DST ENQUEUE
 1(s1-eth1): addr:e2:5b:e8:8f:0a:49
     config:     0
     state:      0
     current:    10GB-FD COPPER
     speed: 10000 Mbps now, 0 Mbps max
 2(s1-eth2): addr:72:fd:fc:5b:2a:18
     config:     0
     state:      0
     current:    10GB-FD COPPER
     speed: 10000 Mbps now, 0 Mbps max
 LOCAL(s1): addr:7e:03:14:41:d3:41
     config:     0
     state:      0
     speed: 0 Mbps now, 0 Mbps max
OFPT_GET_CONFIG_REPLY (xid=0x4): frags=normal miss_send_len=0
```

Switch 2 s2:

```
*** s2 ----------------------------------------------------------------------
OFPT_FEATURES_REPLY (xid=0x2): dpid:0000000000000002
n_tables:254, n_buffers:256
capabilities: FLOW_STATS TABLE_STATS PORT_STATS QUEUE_STATS ARP_MATCH_IP
actions: OUTPUT SET_VLAN_VID SET_VLAN_PCP STRIP_VLAN SET_DL_SRC SET_DL_DST SET_N
W_SRC SET_NW_DST SET_NW_TOS SET_TP_SRC SET_TP_DST ENQUEUE
 1(s2-eth1): addr:0a:d6:95:82:a7:92
     config:     0
     state:      0
     current:    10GB-FD COPPER
     speed: 10000 Mbps now, 0 Mbps max
 2(s2-eth2): addr:3e:c0:b1:26:72:13
     config:     0
     state:      0
     current:    10GB-FD COPPER
     speed: 10000 Mbps now, 0 Mbps max
 3(s2-eth3): addr:76:b1:61:76:b4:3b
     config:     0
     state:      0
     current:    10GB-FD COPPER
     speed: 10000 Mbps now, 0 Mbps max
 LOCAL(s2): addr:ae:fb:3c:62:fe:47
     config:     0
     state:      0
     speed: 0 Mbps now, 0 Mbps max
OFPT_GET_CONFIG_REPLY (xid=0x4): frags=normal miss_send_len=0
```

Switch 3 s3

```
*** s3 ----------------------------------------------------------------------
OFPT_FEATURES_REPLY (xid=0x2): dpid:0000000000000003
n_tables:254, n_buffers:256
capabilities: FLOW_STATS TABLE_STATS PORT_STATS QUEUE_STATS ARP_MATCH_IP
actions: OUTPUT SET_VLAN_VID SET_VLAN_PCP STRIP_VLAN SET_DL_SRC SET_DL_DST SET_N
W_SRC SET_NW_DST SET_NW_TOS SET_TP_SRC SET_TP_DST ENQUEUE
 1(s3-eth1): addr:66:53:39:3b:b5:ee
     config:     0
     state:      0
     current:    10GB-FD COPPER
     speed: 10000 Mbps now, 0 Mbps max
 2(s3-eth2): addr:76:d5:15:82:b2:bd
     config:     0
     state:      0
     current:    10GB-FD COPPER
     speed: 10000 Mbps now, 0 Mbps max
 3(s3-eth3): addr:02:ee:0b:f1:d3:24
     config:     0
     state:      0
     current:    10GB-FD COPPER
     speed: 10000 Mbps now, 0 Mbps max
 LOCAL(s3): addr:e2:80:d6:f6:d7:42
     config:     0
     state:      0
     speed: 0 Mbps now, 0 Mbps max
OFPT_GET_CONFIG_REPLY (xid=0x4): frags=normal miss_send_len=0
mininet>
```

Amb el següent comando ‘dpctl dump-flow’ es visualitzen les taules de fluxes dels switchos, on es pot observar la regla de replicació per defecte assignada amb anterioritat.

```
mininet> dpctl dump-flows
*** s1 ----------------------------------------------------------------------
NXST_FLOW reply (xid=0x4):
*** s2 ----------------------------------------------------------------------
NXST_FLOW reply (xid=0x4):
*** s3 ----------------------------------------------------------------------
NXST_FLOW reply (xid=0x4):
mininet>
```

Amb el mateix comando ‘dpctl’ es poden afegir i configurar noves regles, així com esborrar-les.

# CAPÍTOL 8

## 8. CONCLUSIONS

Com hem pogut observar, la constant evolució a les xarxes de comunicacions ha arribat a tals límits, que requereix d'una reestructuració de les arquitectures de les quals es composa, així com la creació de noves. Existeix una evident necessitat de dotar-les d'una major flexibilitat i agilitat per tal d'atendre noves demandes i pogué generar majors beneficis. Hem vist com la virtualització de xarxes proporciona els requisits exigits per aportar aquesta evolució, fent de les mateixes unes xarxes més escalables, àgils i flexibles, aconseguint a la vegada fortes reduccions econòmiques gràcies a l'abaratiment de les pròpies infraestructures.

En aprofundir en les dues arquitectures relacionades amb la virtualització de xarxes, hem vist que SDN separa el control de la xarxa de la commutació dels paquets de dades i ho situa i dóna accés de forma centralitzada. NFV en canvi, dissocia funcions de la xarxa reorientant-les cap a VM's. Una aporta millores al tràfic i l'altre als serveis, però queda patent que ambdues arquitectures es complementen i beneficien mútuament, sent les dues, tecnologies clau en el present i futur de les xarxes de comunicació.

Per acabar, tot i que no estava previst en aquest projecte de recerca, he simulat una xarxa SDN per comprovar el seu funcionament i pogué visualitzar-ho de forma pràctica. Aquest experiment no és si no, l'inici de futures proves amb l'eina Mininet, la qual ofereix possibilitats molt interessants en quan a la investigació d'aquestes tecnologies.

La idea d'una xarxa programable, flexible i àgil no és nova en el sector de les comunicacions, el que sí té un caràcter nou és el treball conjunt que està fent tota la indústria, ja que fins al moment totes les iniciatives en aquest apartat havien tingut un caràcter propietari. SDN i NFV continuen en constant desenvolupament, les grans empreses les estan incorporant en més o menys mida i les que no ho fan encara, estan a l'expectativa.

Tot i tractar-se d'una tecnologia actualment en auge, la curta antiguitat de la mateixa fa que no existeixi gaire informació que pugui demostrar els beneficis a llarg termini que suposa la seva implementació. Tot i això, les projeccions de les empreses indiquen una imminent migració cap a aquestes dues arquitectures, però l'anàlisi de la conveniència de implantació, ha de realitzar-se cas per cas (i cas d'ús per cas d'ús).

# Bibliografia

Larsen DeCarlo, Amy (Novembre 2013). *Mulling enterprise SDN deployment? Not so fast.* [en línia]. http://searchnetworking.techtarget.com/tip/Mulling-enterprise-SDN-deployment-Not-so-fast [data de consulta: 25/10/2015].

Transparency market research (Juny 2015). *Global Software Defined Networking (SDN) Market to Grow.* [en línia]. http://searchnetworking.techtarget.com/tip/Mulling-enterprise-SDN-deployment-Not-so-fast [data de consulta: 26/10/2015].

Feamster, N., Rexford, J., and E. Zegura, "The Road to SDN: An Intellectual History of Programmable Networks", ACM Queue, Volume 11, Issue 12, 2013.

Diego Kreutz, Member, IEEE (Octubre 2014). *Software-Defined Networking: A Comprehensive Survey.* [en línia]. http://arxiv.org/pdf/1406.0440.pdf [data de consulta: 28/10/2015].

Townsend, Keith (Abril 2014). *Una introducción a SDN puede ayudar a los administradores de servidor a superar los silos* [en línia]. http://searchdatacenter.techtarget.com/es/consejo/Una-introduccion-a-SDN-puede-ayudar-a-los-administradores-de-servidor-a-superar-los-silos [data de consulta: 28/10/2015].

Rouse, Margaret (Maig 2015). *¿Qué es OpenStack?: Definición* [en línia]. http://searchdatacenter.techtarget.com/es/definicion/Que-es-OpenStack [data de consulta: 28/10/2015].

Rouse, Margaret (Octubre 2014). *Virtualización de las funciones de red (NFV): Definición* [en línia]. http://searchdatacenter.techtarget.com/es/definicion/Virtualizacion-de-las-funciones-de-red-NFV [data de consulta: 28/10/2015].

Rouse, Margaret (Octubre 2014). *Redes definidas por software (SDN): Definición* [en línia]. http://searchdatacenter.techtarget.com/es/definicion/Redes-definidas-por-software-SDN [data de consulta: 28/10/2015].

Rouse, Margaret (Novembre 2012). *OpenFlow: Definición* [en línia]. http://searchdatacenter.techtarget.com/es/definicion/OpenFlow [data de consulta: 15/11/2015].

Narcisi, Gina (Juliol 2013). *Virtualización de las funciones de red, alternativa a las redes escalables* [en línia]. http://searchdatacenter.techtarget.com/es/consejo/Virtualizacion-de-las-funciones-de-red-alternativa-a-las-redes-escalables [data de consulta: 15/11/2015].

M. Fos, Ana (Maig 2014). *SDN y NFV protagonizan la transformación de las redes* [en línia]. http://www.telecomunicacionesparagerentes.com/sdn-y-nfv-protagonizan-la-transformacion-de-las-redes/ [data de consulta: 15/11/2015].

Feamster, Nick. *The Road to SDN: An Intellectual History of Programmable Networks* [en línia]. https://www.cs.princeton.edu/courses/archive/fall13/cos597E/papers/sdnhistory.pdf [data de consulta: 25/10/2015].

El blog de inquietud técnica (Maig 2014). *Redes definidas por software* [en línia]. http://principiatechnologica.com/sdn/ [data de consulta: 25/10/2015].

Rodriguez, Sergio (Setembre 2012). *Mecanismos de control de las comunicaciones en la internet del futuro a traves de openflow* [en línia]. http://repositorio.unican.es/xmlui/bitstream/handle/10902/1165/Sergio%20Rodriguez%20Santamaria.pdf?sequence=1 [data de consulta: 25/10/2015].

Wikipedia. *Redes definidas por software* [en línia]. https://es.wikipedia.org/wiki/Redes_definidas_por_software [data de consulta: 25/10/2015].

Wikipedia. *Software-defined_networking* [en línia]. https://en.wikipedia.org/wiki/Software-defined_networking [data de consulta: 25/10/2015].

Wikipedia. *Talk:Software-defined_networking* [en línia]. https://en.wikipedia.org/wiki/Talk:Software-defined_networking [data de consulta: 25/10/2015].

Wikipedia. *OpenFlow* [en línia]. https://es.wikipedia.org/wiki/OpenFlow [data de consulta: 25/10/2015].

Navarro, Francisco (Agost 2015). *Openflow, SDN i NFV* [en línia]. http://docplayer.es/865687-Indice-1-openflow-y-sus-herramientas-2-software-defined-networking-sdn-3-network-function-virtualization-nfv.html [data de consulta: 25/10/2015].

Haleplidis, E., Hadi Salim, J., Denazis, S., and O. Koufopavlou, "Towards a Network Abstraction Model for SDN", Journal of Network and Systems Management: Special Issue on Management of Software Defined Networks, pp. 1-19, 2014.

https://www.citrix.com/content/dam/citrix/en_us/documents/oth/sdn-101-an-introduction-to-software-defined-networking-es.pdf

Cisco. *Cisco Application Policy Infrastructure Controller (APIC)* [en línia]. http://www.cisco.com/c/en/us/products/cloud-systems-management/application-policy-infrastructure-controller-apic/index.html [data de consulta: 25/10/2015].

McNickle, Michelle (Octubre 2014). *Cinco controladores SDN comerciales que hay que conocer* [en línia]. http://searchdatacenter.techtarget.com/es/cronica/Cinco-controladores-SDN-comerciales-que-hay-que-conocer [data de consulta: 25/10/2015].

Araque, Andres (Desembre 2014). *Introducción a VMware NSX. Conceptos iniciales* [en http://www.colombiavirtualizada.com/2014/12/09/introduccion-vmware-nsx-conceptos-iniciales/ [data de consulta: 25/10/2015].

Vmware. NSX. [en línia]. https://www.vmware.com/es/products/nsx [data de consulta: 25/10/2015].

Sdx central. *What is NFV – Network Functions Virtualization?.* [en línia]. https://www.sdxcentral.com/resources/nfv/whats-network-functions-virtualization-nfv/ [data de consulta: 25/10/2015].

Sdx central. *NFV and SDN: What's the Difference?.* [en línia]. https://www.sdxcentral.com/articles/contributed/nfv-and-sdn-whats-the-difference/2013/03/ [data de consulta: 25/10/2015].

Tittel, Ed (Gener 2014). *Understanding How SDN and NFV Can Work Together* [en línia]. http://www.cio.com/article/2379216/business-analytics/understanding-how-sdn-and-nfv-can-work-together.html [data de consulta: 25/10/2015].

Noble, Steve (Abril 2015). *Network Function Virtualization or NFV Explained* [en línia]. http://wikibon.org/wiki/v/Network_Function_Virtualization_or_NFV_Explained [data de consulta: 25/10/2015].

Back, Justyna (Agost 2014). *SDN and NFV: Friends or Enemies* [en línia]. http://www.slideshare.net/pontschek/sdn-and-nfvfriendsorenemies [data de consulta: 25/10/2015].

Openflow. [en línia]. http://archive.openflow.org/ [data de consulta: 25/10/2015].

Heller, Brandon (Abril 2014). *openflow-spec* [en línia]. https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-spec-v1.0.0.pdf [data de consulta: 25/10/2015].

Techpedia. [en línia]. https://www.techopedia.com/definition/28935/openflow [data de consulta: 25/10/2015].

Evans, Steve (Març 2013). *The history of OpenFlow* [en línia]. http://www.computerweekly.com/feature/The-history-of-OpenFlow [data de consulta: 25/10/2015].

Openstack. [en línia]. https://www.openstack.org/ [data de consulta: 25/10/2015].

Wikipedia. *OpenStack* [en línia]. https://en.wikipedia.org/wiki/OpenStack [data de consulta: 25/10/2015].

https://openwebinars.net/que-es-eso-de-openstack-por-que-deberia-conocerlo/ [data de consulta: 25/10/2015].

Martínez de la Cruz, Victoria (Febrer 2013). *En pocas palabras: ¿Cómo funciona OpenStack?* [en línia]. http://vmartinezdelacruz.com/en-pocas-palabras-como-funciona-openstack/ [data de consulta: 25/10/2015].

Pepple, Ken (Novembre 2011). *The Tenrec Rises: Deploying OpenStack, Havana Edition* [en línia]. http://ken.pepple.info/ [data de consulta: 25/10/2015].

Macvittie, Lori (Gener 2014). *What About SDN?* [en línia]. https://wiki.openstack.org/wiki/Neutron [data de consulta: 25/10/2015].

Sdx central. *What is OpenStack Neutron?.* [en línia]. https://www.sdxcentral.com/resources/open-source/what-is-openstack-quantum-neutron/ [data de consulta: 25/10/2015].

Centec networks. *Switch Boost OpenStack Network Virtualization.* [en línia]. http://www.centecnetworks.com/en/SolutionList.asp?ID=79 [data de consulta: 25/10/2015].

Macvittie, Lori (Gener 2014). *What About SDN?* [en línia]. http://sdn.sys-con.com/node/2918106 [data de consulta: 25/10/2015].

Romero Sanchez, Daniel (Abril 2015). *OpenStack desde cero - Neutron* [en línia]. http://www.dbigcloud.com/cloud-computing/176-openstack-desde-cero-neutron-parte-1.html [data de consulta: 25/10/2015].

DOC JMA (Desembre 2012). *Aprendiendo Openstack* [en línia]. http://aprendiendoopenstack.blogspot.com.es/ [data de consulta: 25/10/2015].

Petriuk, Polina (2014). *OpenStack Architecture* [en línia]. http://www.slideshare.net/mirantis/openstack-architecture-43160012 [data de consulta: 25/10/2015].

http://blogs.salleurl.edu/networking-and-internet-technologies/openstack-descripcion-de-la-solucion-cloud-opensource/ [data de consulta: 25/10/2015].

Vmware. *Nsx.* [en línia]. http://www.vmware.com/latam/products/nsx [data de consulta: 25/10/2015].

Rouse, Margaret (Abril 2013). *VMware NSX definition* [en línia]. http://searchvmware.techtarget.com/definition/VMware-NSX [data de consulta: 25/10/2015].

searchsdn.techtarget. *How VMware NSX network virtualization could change networking -- or not.* [en línia]. http://searchsdn.techtarget.com/essentialguide/How-VMware-NSX-network-virtualization-could-change-networking-or-not [data de consulta: 25/10/2015].

Sen Jimenez, Carlos (Novembre 2013) *Llevando la Virtualización de Redes a entornos VMware con NSX* [en línia]. https://www.delegate.com/content/vmware/emea/2014/images/Llevando%20La%20Virt

ualizacion%20de%20Redes%20a%20Entornos%20VMware%20con%20NSX_csen.pdf [data de consulta: 25/10/2015].

Cerda, Patricio (Novembre 2013) *NSX: Introducción a NSX y al Software Defined Network.* [en línia]. http://patriciocerda.com/2015/11/nsx-introduccion-a-nsx-y-al-software-defined-network-2.html [data de consulta: 25/10/2015].

Araque, Andres (Gener 2015) *VMware NSX – Introducción a la VXLAN.* [en línia]. http://www.colombiavirtualizada.com/2015/01/15/vmware-nsx-introduccion-las-vxlan/ [data de consulta: 25/10/2015].

Malanco, Agustin (Decembre 2013) *¿Que es NSX?.* [en línia]. http://hispavirt.com/2013/12/08/que-es-nsx-parte-1/ [data de consulta: 25/10/2015].

Cisco. *Network Programmability and Automation with Cisco Nexus 9000 Series Switches.* [en línia]. http://www.cisco.com/c/en/us/products/collateral/cloud-systems-management/aci-fabric-controller/white-paper-c11-729385.html [data de consulta: 25/10/2015].

Cisco. *Nexus 9000 Programmable Network Environment.* [en línia]. http://www.cisco.com/c/dam/en/us/products/collateral/switches/nexus-9000-series-switches/n9k-programming-research-note.pdf [data de consulta: 25/10/2015].

Cisco. *Switches Nexus de Cisco de la serie 9000.* [en línia]. http://www.cisco.com/web/ES/products/switches/nexus_9000_series_switches.html [data de consulta: 25/10/2015].

Morales, David (Decembre 2013) *Tutorial MININET* [en línia]. https://www.academia.edu/8826530/TUTORIAL_MININET

Juniper. *Example: Enabling OpenFlow on EX9200 Switches.* [en línia]. http://www.juniper.net/documentation/en_US/junos13.3/topics/example/junos-sdn-openflow-support-configuring-ex9200.html [data de consulta: 25/10/2015].

Juniper. *OpenFlow Support on Juniper Networks Devices.* [en línia]. https://www.juniper.net/documentation/en_US/release-independent/junos/topics/reference/general/junos-sdn-openflow-supported-platforms.html [data de consulta: 25/10/2015].

Duffy, Jim (Agost 2015). *Juniper introduce arquitectura llamada Unite.* [en línia]. http://cioperu.pe/articulo/19553/juniper-introduce-arquitectura-llamada-unite/ [data de consulta: 25/10/2015].

Juniper. *EX Series ethernet switches* [en línia]. http://pdf.directindustry.es/pdf-en/juniper-networks/ex-series-ethernet-switches/61504-182784.html#open [data de consulta: 25/10/2015].