
La virtualització de les xarxes

PID_00265727

Víctor Huertas García

Temps mínim de dedicació recomanat: 4 hores



**Víctor Huertas García**

Enginyer de Telecomunicacions per la Universitat Politècnica de Catalunya. Actualment treballa com a enginyer de *networking* i expert en NGN/IMS a Departament d'Equips de Comunicació de la multinacional Indra Sistemas. Ha participat en nombrosos projectes de recerca de l'ESA (Agència Espacial Europea) sobre l'aplicació de la tecnologia IP en xarxes satèl·lit. Recentment ha participat en projectes d'integració d'IMS a les xarxes satèl·lit per aconseguir la convergència amb xarxes terrestres.

L'encàrrec i la creació d'aquest recurs d'aprenentatge UOC han estat coordinats pel professor: Víctor García Font (2019)

Segona edició: setembre 2019

Autoria: Víctor Huertas García

Llicència CC BY-NC-ND d'aquesta edició, FUOC, 2019

Av. Tibidabo, 39-43, 08035 Barcelona

Realització editorial: FUOC



Els textos i imatges publicats en aquesta obra estan subjectes –llevat que s'indiqui el contrari– a una llicència de Reconeixement-NoComercial-SenseObraDerivada (BY-NC-ND) v.3.0 Espanya de Creative Commons. Podeu copiar-los, distribuir-los i transmetre'ls públicament sempre que en citeu l'autor i la font (FUOC. Fundació per a la Universitat Oberta de Catalunya), no en feu un ús comercial i no en feu obra derivada. La llicència completa es pot consultar a <http://creativecommons.org/licenses/by-nc-nd/3.0/es/legalcode.ca>

Índex

Introducció	5
Objectius	6
1. Software defined networks (SDN)	7
1.1. Per què anem cap a l'SDN?	7
1.2. Comprensió del canvi de paradigma que introdueixen les SDN	8
1.2.1. Com està organitzat internament el programari d'un PC?	8
1.2.2. Quina similitud guarda un PC amb el model arquitectural de les SDN?	10
1.2.3. El model de referència de les SDN	11
1.2.4. Mètode que usa el controlador SDN per a controlar els elements de la capa d'infraestructura	14
1.3. El controlador SDN	15
1.4. <i>Southbound</i> API i desacoblament entre la capa de control i la de dades	17
1.5. <i>API Northbound</i> i el desenvolupament d'aplicacions	19
1.6. Interconnexió entre controladors SDN	20
1.7. Exemple d'ús de les SDN	21
2. Network function virtualization (NFV)	24
2.1. Arquitectura de referència de NFV	26
2.1.1. VNF (<i>Virtual Network Function</i>)	27
2.1.2. NFVI (<i>network function virtualization infrastructure</i>)	28
2.1.3. EMS (<i>element management system</i>)	28
2.1.4. OSS/BSS (<i>Operation Support System / Business Support System</i>)	29
2.1.5. NFV MANO (<i>Management & Orchestration</i>)	30
2.2. Punts de referència en NFV	34
2.3. Exemple de <i>service chaining</i> : implementació d'un EPC més un nucli IMS	35
2.4. Plataformes existents que implementen NFV	39
Resum	40
Exercicis d'autoavaluació	43
Solucionari	44

Glossari	46
Bibliografia	48

Introducció

D'acord amb el paradigma introduït per les xarxes 5G, la flexibilització total de les xarxes de telecomunicacions busca l'objectiu d'aconseguir una xarxa multiservei amb capacitat per a complir els requisits de QoS que imposen els casos d'ús proposats per a 5G.

Aquesta flexibilitat passa inexorablement per virtualitzar les xarxes on hi ha un node central que aglutina el control, la intel·ligència i coneixement de l'arquitectura de tota la xarxa. És el que anomenem SDN o *programari defined networks*.

En definitiva, és com tornar un model que ja s'ha viscut anys enrere, en concret en els anys setanta del segle passat quan les xarxes de telecomunicacions de dades acabaven de començar. En aquesta època el món de les computadores estava tot centralitzat en un node al qual es connectaven múltiples terminals amb capacitats computacionals molt limitades. La gent usava el sistema central per a executar les seves tasques de computació quotidianes. A la fi dels vuitanta, el model va evolucionar a poc a poc cap a un sistema distribuït i la intel·ligència es va anar desplaçant des del node central cap als terminals remots.

Però, tal com hem dit, amb l'explosió de la informàtica en núvol estem tornant a un model de centralitat. El núvol farà totes les funcions de computació, treball en xarxa, emmagatzematge d'informació i seguretat sense necessitat de tenir infraestructura local.

En aquest mòdul ens endinsarem en la descripció de les SDN (*programari defined networks*) i l'NFV (*network functions virtualization*).

Objectius

Els continguts d'aquest mòdul han de permetre als estudiants els objectius següents:

- 1.** Comprendre la influència dels serveis 5G i les seves característiques en QoS en l'impuls de la virtualització de xarxes.
- 2.** Comprendre els beneficis obtinguts com a resultat de l'ús de tecnologia SDN (*programari defined network*) i NFV (*network function virtualization*).
- 3.** Assimilar i comprendre el perquè del canvi de paradigma en l'arquitectura funcional dels elements d'una xarxa de telecomunicacions en un entorn SDN.
- 4.** Aprendre què és un controlador SDN i quina funció té en el desacoblament entre la capa de dades i la de control.
- 5.** Conèixer els blocs que componen un ecosistema NFV segons l'especificació de l'ETSI.
- 6.** Comprendre el funcionament intern de l'element més important de l'NFV: NFV MANO.
- 7.** Comprendre com es complementen les tecnologies SDN i NFV.

1. Software defined networks (SDN)

En les seccions següents abordarem la descripció en profunditat del canvi de paradigma que introdueixen les SDN. No obstant això, abans de començar creiem que és important entendre per què aquesta arquitectura de referència ha sofert aquest canvi.

1.1. Per què anem cap a l'SDN?

El món de la provisió de serveis multimèdia tal com han introduït les NGN tendeix a convertir les xarxes de transport en xarxes multiservei, en les quals una sola infraestructura de comunicacions, independentment de la tecnologia, a pot usar per a oferir el màxim nombre de serveis. I per a oferir aquests serveis amb uns requeriments molt heterogenis de QoS es necessita aconseguir una total integració de les xarxes de transport.

Les entitats que especifiquen les xarxes de telefonia 5G ja ens donen pistes de quina és la màxima en l'especificació del nou model de referència: **les xarxes han de ser flexibles** per a suportar qualsevol servei, per a poder fragmentar-se en dominis administratius completament independents i per a poder **adaptar-se ràpidament** a canvis en els requeriments en l'àmbit de model de negoci (amb nous serveis, nous perfils, increments en el nombre d'usuaris, etc.).

En definitiva, les xarxes han de permetre que els administradors i enginyers de xarxes i operadors puguin adaptar la seva xarxa a aquests canvis. Les *software defined networks*, seguint un model centralitzat en què la capa de control i la capa de dades estan separades completament, poden oferir la solució.

Però un model distribuït com el que sempre hi ha hagut fins ara no pot oferir el mateix?

Efectivament, el model clàssic, en el qual s'usen protocols estandarditzats la capa de control dels quals se situa en cadascun dels elements de la xarxa, pot oferir una solució a aquest repte. Però té dues dificultats principals:

- **Limita l'escalabilitat.** En una xarxa distribuïda, incrementar la capacitat de la xarxa implica que la capa de control dels nous elements afegits es coordini amb els ja existents sense estar exempts d'impacte en la resta de sistema.
- **Limita l'operativitat.** Relacionat amb l'anterior, el fet d'haver de configurar i gestionar cada element per separat (cadascun amb la seva pròpia in-

Elements de processament de paquets en SDN

Els elements que processen paquets en xarxes SDN han de suportar la comunicació amb l'element de control, anomenat controlador SDN, mitjançant un protocol específic que ha de ser suportat per aquest element. Aquests elements se solen anomenar *bare-metal* per recalcar el seu poder de decisió gairebé nul en el processament i encaminament de paquets.

terfície de monitoratge i control) fa de l'operativitat de la xarxa quelcom complex i propens a fallades.

Quins beneficis aporta passar a un model de xarxa centralitzat com el que es proposa en les *software defined networks*?

El fet de tenir desagregades la part de control i la part de processament de transport permet que amb un sol programari de control es pugui gestionar remotament una infinitat d'equips que l'única cosa que faran és obeir les configuracions que reben des d'aquest programari. Amb això, el problema de l'**escalabilitat** de la xarxa estaria solucionat i alhora es tindria una **xarxa integrada completament**.

També permet a **aplicacions de tercers controlar la xarxa sencera** sense necessitat que aquesta conegui la topologia física real de la xarxa. De fet, en SDN tot es pot considerar una aplicació.

També permet el que anomenen *service chaining*, en el qual podem reconfigurar en temps real parts de la xarxa per a poder concatenar diversos serveis amb l'objectiu de flexibilitzar-ne l'ús.

Finalment, permet **integrar xarxes amb fil i sense fil** de manera transparent.

Així, doncs, en les seccions següents explicarem amb més detalls què són les SDN començant per la descripció del canvi de paradigma i centrant-nos després en l'arquitectura de referència al costat dels principals actors que la defineixen, dels quals remarcarem el controlador SDN.

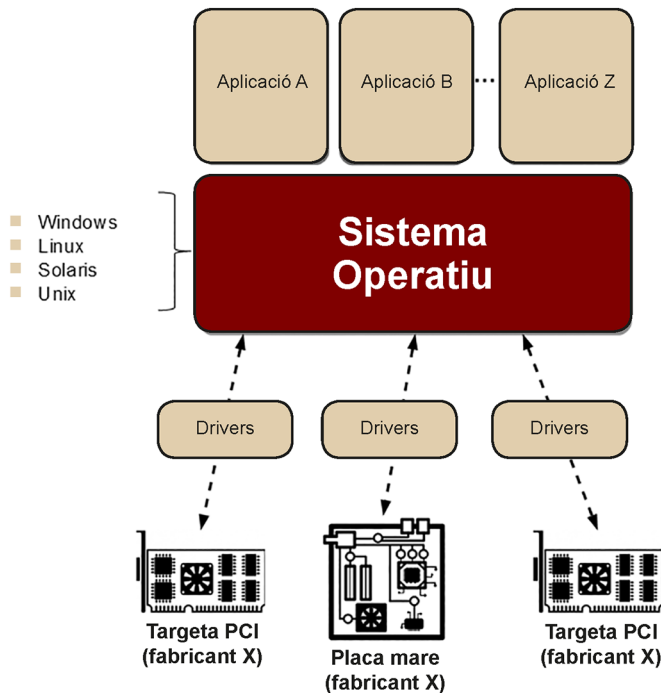
1.2. Comprensió del canvi de paradigma que introdueixen les SDN

Per a comprendre bé el canvi de paradigma que introdueixen les SDN, veurem un símil. Compararem el model d'arquitectura del programari d'un ordinador amb el model de les SDN. Com veurem a continuació, guarden moltes similituds.

1.2.1. Com està organitzat internament el programari d'un PC?

Encara que és àmpliament conegut com s'estructura un ordinador per dins a alt nivell, val la pena donar-li un cop d'ull. La Figura 1 mostra aquesta arquitectura.

Figura 1. Arquitectura del programari d'un ordinador.



En el nivell més baix, tenim els diferents components de maquinari (targetes PCI, placa mare, etc.), els quals poden ser de diferents fabricants.

No obstant això, perquè el sistema operatiu pugui controlar-los és necessari instal·lar els programes controladors, que solament cada fabricant de cada component ens pot facilitar. És a dir, que els programes controladors venen a ser un enllaç entre el maquinari i el sistema operatiu que emmascara l'heterogeneïtat dels fabricants en els diferents dispositius de maquinari. Aquest enllaç l'anomenarem *southbound* tenint en compte que el sistema operatiu fa d'«equador» entre dos hemisferis.

El **sistema operatiu** fa, gràcies als programes controladors, una **abstracció dels recursos de maquinari** pel que fa a les aplicacions que s'hi executen. És conegut que hi ha diferents tipus de sistemes operatius (Windows, Linux, Unix, etc.), i fins i tot entre cada sistema operatiu podem trobar diferents versions o distribucions (especialment en sistemes operatius com Linux, entre els quals podem trobar OpenSuse, Debian, Ubuntu, Red Hat, etc.). La base d'aquestes distribucions és la mateixa (el nucli), però amb petites variacions en les quals s'ofereixen noves prestacions i aplicacions de sèrie que faciliten tasques al desenvolupador d'aplicacions.

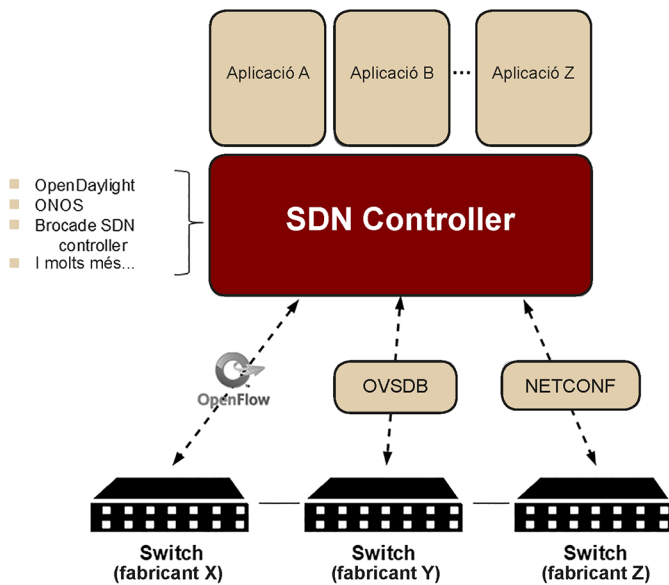
Finalment, mitjançant API (*application programming interface*) especialment adaptades al sistema operatiu, es poden desenvolupar diferents aplicacions sense haver de preocupar-se de les particularitats del maquinari de l'ordinador. Aquestes API fan d'enllaç entre l'aplicació per si mateixa i el sistema opera-

tiu on s'executa. A aquest enllaç l'anomenarem *Northbound*, ja que des del punt de vista de l'emplaçament del sistema operatiu en aquesta arquitectura les aplicacions estan «sobre» ell.

1.2.2. Quina similitud guarda un PC amb el model arquitectural de les SDN?

La Figura 2 permet comprendre de seguida l'equivalència entre tots dos models.

Figura 2. Arquitectura equivalent en SDN.



En lloc de targetes PCI i plaques mare, tenim tot un seguit de **commutadors que estan interconnectats físicament entre ells** amb cablejat de xarxa però sense cap tipus de configuració carregada (a excepció de l'agent SDN, que han de tenir integrat i configurat perquè puguin comunicar-se amb el corresponent controlador SDN). La localització d'aquests commutadors pot ser molt dispar, de manera que poden estar separats completament del servidor on s'executa el controlador SDN.

Així, doncs, tenim despleats diversos elements de maquinari de xarxa que poden ser tots d'un fabricant diferent.

Pel que fa al *southbound*, tenim l'enllaç entre els commutadors i el controlador SDN via connexió de tipus client-servidor, que és implementat mitjançant protocols de M&C, alguns dedicats específicament a implementar les SDN. Aquests protocols tenen funcions assimilables als programes controladors de la xarxa. Entre aquests protocols, podem destacar-ne un: **OpenFlow**, el qual es considera l'estàndard que ha de suportar gairebé qualsevol fabricant. En paral·lel, han aparegut més protocols que també són molt utilitzats, com OVSDB i NETCONF. Però podem trobar-ne molts més.

M&C

Aquestes sigles corresponen a *monitoratge* i *control*, i es refereixen a les funcions de configuració d'un equip i a l'enviament d'esdeveniments en temps real associats al seu funcionament.

El controlador SDN és un element que té una visió holística de la xarxa, de manera que absteu les aplicacions dels recursos de xarxa. **Es considera fins i tot el sistema operatiu de la xarxa.** I hi ha diverses versions de sistema operatiu, algunes de font pública (com OpenDaylight o ONOS), i altres de propietàries (com VMware NSX o Cisco APIC). També podem trobar diverses versions basades en aquests controladors. Per exemple, Brocade, Avaya i Coriant són controladors SDN comercials que estan basats en OpenDaylight.

Pel que fa al *Northbound*, el controlador SDN ofereix una sèrie d'API als desenvolupadors d'aplicacions que en faciliten la implementació. A diferència dels protocols de tipus *southbound*, no hi ha API que es considerin estàndards, i cada aplicació haurà d'estar desenvolupada basant-se en el controlador SDN i les API que ofereixi.

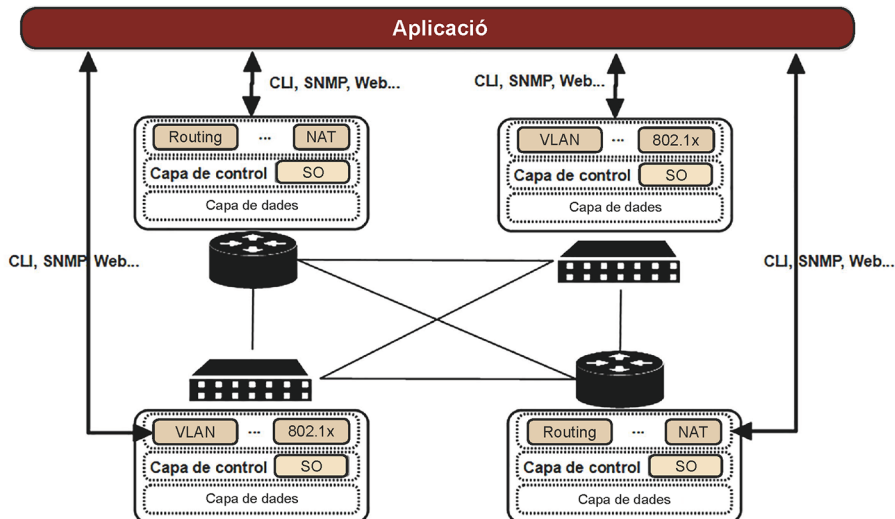
No obstant això, els desenvolupadors d'aplicacions no hauran de bregar amb la complexitat d'haver de configurar tots els elements de xarxa un a un ni amb les particularitats de les interfícies de configuració (CLI particular, MIB SNMP particular) de cada fabricant.

1.2.3. El model de referència de les SDN

Aquest símil ha ajudat a veure cap a on s'encamina l'arquitectura de les xarxes en el futur i el paper que té el controlador SDN, però centrem-nos en l'impacte que aquest nou paradigma tindrà en els elements de la xarxa que processen paquets i s'encarreguen de conferir «intel·ligència» a la xarxa: els encaminadors i els commutadors.

El diagrama següent (Figura 3) descriu a alt nivell l'arquitectura d'una xarxa senzilla amb un model previ a les SDN. El diagrama també descriu l'arquitectura funcional interna en capes d'un encaminador o un commutador en una xarxa que compleix el paradigma d'intel·ligència distribuïda tradicional.

Figura 3. Model arquitectònic previ a les SDN.



Cadascun dels elements que conformen la xarxa tradicional (encaminadors i commutadors) es pot estructurar internament en diverses capes, que descrivim a continuació:

- **Capa d'aplicació.** Implementa una funcionalitat (*feature*) concreta dins del dispositiu. Per exemple, en l'encaminador, la funcionalitat de NAT (*network address translation*) o IP *routing* estàtic o dinàmic, que a vegades implica intercanviar missatges entre encaminadors adjacents (protocols RIP o OSPF), generats per aquesta capa.
- **Capa de control.** Ve a ser el sistema operatiu del dispositiu i és la que pren decisions d'encaminament entre ports i/o prioritització de paquets IP d'acord a les instruccions provinents de les funcionalitats en la capa d'aplicació. Aquesta capa sol ser un programari propietari que està associat totalment al maquinari de l'equip (sol ser un programari propietari del fabricant).
- **Capa de dades.** És simplement la capa on es transfereixen els paquets IP entrants i sortints de l'encaminador o les trames Ethernet entre ports entrants o sortints en un commutador. En aquesta capa s'apliquen les decisions rebudes des de la capa de control. Normalment, aquesta informació de transferència entre ports s'emmagatzema en una taula específica optimitzada per a fer el processament de paquet a paquet al més ràpidament possible.

En l'àmbit de l'arquitectura funcional el més distintiu d'aquest model tradicional és que totes aquestes capes estan integrades en el mateix dispositiu maquinari i les decisions d'encaminament es decideixen localment. No obstant això, tal com es veu en la Figura 3, si es vol implementar una aplica-

ció global per a xarxa s'han de configurar diversos elements de xarxa simultàniament, sia mitjançant un operador de xarxa (enginyer) o un script (procés automatitzat).

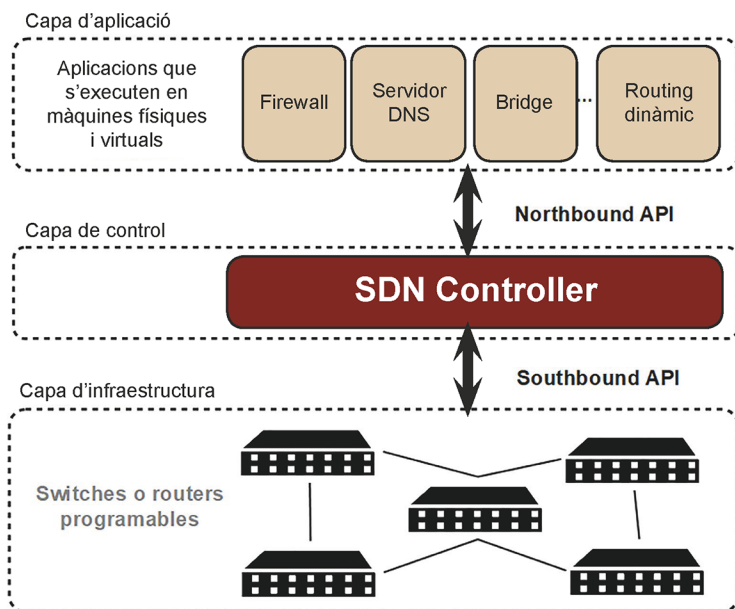
L'operador (o programari extern que implementa l'aplicació) ha de comunicar-se individualment amb cada dispositiu i configurar-los un a un. Aquesta connexió es fa directament amb la capa d'aplicació de cada dispositiu usant les interfícies de configuració que el fabricant hagi implementat (per exemple, pot ser via línia d'ordres mitjançant una connexió Telnet o SSH, o via protocol SNMP modificant el contingut d'una MIB).

Tal com hem indicat en seccions anteriors, la gestió d'aquestes xarxes implica també reconfigurar un gran nombre d'equips, sobretot en cas de xarxes molt grans amb molts dispositius quan cal fer reconfiguracions o personalitzacions en la tipologia per a adaptar-la a una aplicació en concret.

En definitiva, aquest model arquitectònic té una **certa rigidesa i falta de flexibilitat** per a poder adaptar els recursos de la xarxa als requisits de robustesa, amplada de banda i QoS.

El que proposa el nou paradigma de les SDN, que es mostra en la Figura 4, trenca completament amb aquesta arquitectura funcional.

Figura 4. Model arquitectural de les SDN.



Encara que aquest model ja s'ha introduït en seccions anteriors, vegem com és la principal novetat que s'introdueix amb aquest nou paradigma.

MIB

Aquesta sigla correspon en anglès a *management information base*, un tipus de base de dades que conté informació jeràrquica, estructurada en forma d'arbre, de tots els paràmetres gestionables en cada dispositiu gestionat d'una xarxa de comunicacions.

La principal novetat que introdueix aquest nou paradigma és el **desacoblament físic entre la capa de control i la capa de dades o de *forwarding*** que processa i encamina els paquets entre els ports del commutador o encaminador.

Aquest desacoblament és físic. És a dir, que la capa de control i la capa de dades ja no estan en el mateix dispositiu físic. Molts commutadors o encaminadors es despleguen geogràficament, mentre que el controlador SDN pot estar situat en un servidor d'un centre de dades.

D'una banda, tenim un controlador SDN (solament un, encara que pot ser redundat) que controla un gran nombre de commutadors i/o encaminadors habilitats per SDN (és a dir, que implementen l'API de *southbound*). Això ens porta a concloure que **un sol controlador SDN és amo d'un domini sencer al qual pertanyen diferents commutadors i encaminadors en la capa d'infraestructura o de dades.**

1.2.4. Mètode que usa el controlador SDN per a controlar els elements de la capa d'infraestructura

El controlador SDN es comunica directament amb els elements de la capa d'infraestructura (habilitats degudament per SDN) mitjançant els protocols de l'API de *southbound*. D'alguna manera, el controlador SDN programa directament la taula CAM (*content addressable memory*) d'un commutador. De fet, en un context SDN el concepte de taula CAM canvia lleugerament de nom per **taula de fluxos** (*flow table* en anglès), i cada entrada que programa el controlador SDN se l'anomena **entrada de flux** (*flow entry*).

Aquestes taules de fluxos estan formades per camps de filtratge que abracen tant camps de la capçalera Ethernet com de la capçalera IP i TCP/UDP. Si un paquet fa coincidència en una d'aquestes entrades de fluxos, se li **aplica una acció**, la qual també és programada des del controlador SDN.

Aquesta acció pot ser des d'indicar simplement el port de sortida del commutador fins a enviar el paquet al controlador SDN (degudament encapsulat en el protocol *southbound* que s'està usant en aquesta dispositiva). Normalment, quan s'envia un paquet al controlador SDN aquest el retorna al mateix commutador d'on ha partit una vegada ha estat processat degudament per l'aplicació. No obstant això, depenent de l'aplicació el mateix controlador SDN pot ser l'origen o la destinació de trànsit IP des del commutador o cap a ell.

Per exemple, un controlador SDN pot fer una comprovació de la interconnexió de dos o més commutadors generant ell mateix un paquet IP de tipus ICMP i configurant prèviament entrades de fluxos en els commutadors involucrats perquè torni aquest ICMP al mateix controlador SDN per un altre commutador. Així es comprova que encara hi ha continuïtat i que els cables segueixen connectats entre certs commutadors.

Southbound API

Hi ha nombrosos protocols per a controlar remotament dispositius habilitats per a SDN, com OpenFlow, NETCONF i OVSDB.

Taula CAM

Una taula CAM indica a un commutador què ha de fer amb una trama que entra per un port X. Aquestes accions sobre la trama poden ser treure'l per un port Y o lliurar-lo a la capa de control per a fer un altre tipus de processament més complex.

Si el commutador ho suporta, es poden programar **altres accions suplementàries abans de treure el paquet per un port físic**. Aquestes accions es poden fer en el mateix commutador i poden ser des de modificar camps de diverses capçaleres del paquet fins a fer tasques de limitació de flux a una taxa de bit configurada.

Una xarxa SDN que es considera ben implementada **ha d'intentar configurar amb antelació les màximes entrades de fluxos possibles** que afectin el processament de paquets dins de cadascun dels dispositius gestionats, **alhora que es minimitza la reexpedició de paquets al controlador SDN**, ja que pot tenir efectes en la latència.

En les seccions següents veurem més detalladament els elements i interfícies més importants de l'arquitectura de referència de les SDN.

1.3. El controlador SDN

El controlador SDN és l'element que simbolitza millor el desacoblament entre el programari i el maquinari de la xarxa i també, com s'ha dit abans, entre la capa de control i la capa de dades o de *forwarding*.

El controlador SDN és el principal cervell i controlador d'un entorn SDN i ofereix una visió centralitzada de tota la xarxa, de manera que és un punt estratègic de control dins de la xarxa i responsable de comunicar informació als elements següents:

- Commutadors, encaminadors i altres elements després d'aquests. Per a això, el controlador SDN usa API o protocols anomenats de *southbound* (com OpenFlow o NETCONF). Aquesta comunicació es fa amb qualsevol encaminador o commutador independentment del fabricant.
- Aplicacions i fins i tot a la capa d'aplicació i control d'altres encaminadors i commutadors que no estiguin gestionats per SDN. Per a aquesta comunicació, s'usen API o altres protocols anomenats de *Northbound*.

El controlador SDN es considera com un sistema operatiu de xarxa sobre el qual desenvolupar nombroses aplicacions. De fet, sense una aplicació que s'executi sobre un controlador SDN, aquest no farà per si sol cap tipus de gestió ni configuració de la xarxa. El controlador SDN no té sentit si no hi ha aplicacions que s'executin.

Seguint la mateixa filosofia que un sistema operatiu en un ordinador, un controlador SDN no és un programari monolític, sinó una composició de nombrosos programes, els quals estan estructurats en forma de mòduls i connectors. Cada mòdul pot fer diferents tasques de xarxa, que alhora són utilitzades per les aplicacions a manera de «serveis» que completen capacitats que les aplicacions ofereixen.

En el mercat hi ha nombrosos controladors SDN entre els quals triar. L'elecció d'un o un altre per a implementar un domini SDN és quelcom que els administradors de xarxa han d'estudiar amb deteniment per a garantir estabilitat i suport a llarg termini en futures actualitzacions. Hi ha controladors SDN tant de codi lliure com propietaris. Un altre criteri a tenir en compte en l'elecció és l'ecosistema d'aplicacions desenvolupades per cada controlador SDN en qüestió, i la maduresa de les *Northbound API* que ofereixen.

L'elecció del controlador SDN és condicionada també per la finalitat de la xarxa a gestionar. Es poden trobar controladors SDN amb característiques, mòduls i connectors inclosos que estan més focalitzats per a un cas d'ús o un altre.

Per exemple, no és recomanable utilitzar un controlador SDN per a un centre de dades si ha estat adaptat per desplegar-se en una xarxa d'un campus universitari. De la mateixa manera, no té sentit triar un controlador SDN per a un campus que inclogui un connector específic per a OpenStack.

Hi ha algun controlador SDN adoptat àmpliament?

Des de l'aparició del primer controlador SDN (anomenat NOX en 2009, el qual no era de codi obert), han anat evolucionant, de manera que n'han aparegut de nous i també n'han desaparegut al llarg dels anys (per obsolescència i desús per la comunitat d'enginyers). Per aquests motius, avui dia és difícil considerar un sol controlador SDN com a dominant en el mercat, encara que sí que val la pena esmentar-ne a un en concret: **OpenDaylight**.

OpenDaylight (ODL) és un controlador SDN de codi obert impulsat avui dia per la Linux Foundation, encara que originàriament, quan la primera versió va aparèixer en 2013, era liderat per la col·laboració entre Cisco i IBM.

Si destaquem aquest controlador és perquè, a part de ser-ho per si mateix, s'ha adoptat com a base per a desenvolupar altres controladors SDN comercials als quals s'ha afegit funcionalitats extres propietàries.

Exemples d'aquests controladors SDN comercials basats en ODL són Brocade, Ericsson i Ciena, entre d'altres.

També podem destacar un altre controlador SDN de codi obert que repta la prevalença d'ODL. És l'*Open Networking Operating System* (ONOS).

SDN

Vegeu també el web <https://www.sdxcentral.com> com a font d'informació dels conceptes relacionats amb SDN i de les últimes tendències en aquest camp.

OpenStack

Es tracta un entorn per a implementar un centre de dades basat en informàtica en núvol. Ofereix una plataforma d'orquestració de màquines virtuals i, a l'hora d'interconnectar-les, se sol usar tecnologia SDN. Per tant, ha de tenir un enllaç adaptat amb un controlador SDN (connector).

OpenDaylight

Vegeu també el web <https://www.opendaylight.org/>

Distribucions ODL i Linux

Es pot trobar un paral·lisme entre les diferents distribucions d'ODL i Linux

Aquest controlador SDN està focalitzat en les necessitats dels proveïdors de servei (proveïdors de connectivitat WAN). Encara que al principi aquest controlador SDN no era de codi obert, l'any 2014 es va convertir finalment a aquesta modalitat. Aquest producte rep el suport de nombroses organitzacions, com AT&T, Cisco, Fujitsu, Ericsson, Ciena, Huawei, NTT, SK Telekom, NEC i Intel. Algunes d'aquestes organitzacions també donen suport a OpenDaylight.

1.4. **Southbound API i desacoblament entre la capa de control i la de dades**

Les API *southbound* són les que fan possible realment el desacoblament entre la capa de control i la capa de dades. Hi ha nombroses API en forma de protocols que els controladors SDN utilitzen per a programar els commutadors. Però entre tots aquests protocols n'hi ha un que sens dubte ha de ser destacat: **OpenFlow**.

OpenFlow (OF) es considera l'estàndard quant a protocols *southbound* perquè va ser el primer de tots. Encara que avui hi ha més API *southbound*, aquest protocol és suportat per tots els controladors SDN i per commutadors o encaminadors amb capacitats SDN.

De fet, OF es va crear fins i tot abans que els mateixos controladors SDN. OF va ser creat per investigadors en xarxes de computadors a la Universitat de Stanford l'any 2008 i la seva primera versió (v1.0) va ser publicada al desembre del 2009. Ara el desenvolupament d'OF és liderat per l'**Open Networking Foundation (ONF)**, que és una organització formada per usuaris que concentren el seu esforç a avançar en el desenvolupament d'estàndards oberts i l'adopció de tecnologies SDN.

OF és un protocol que està dissenyat específicament per configurar taules de fluxos en un commutador. És a dir, configura bàsicament entrades de fluxos en diferents taules encadenades (trucades *pipelines* en anglès), i a cada entrada s'associa una acció que cal portar a terme amb el paquet. OF, per exemple, no és un protocol per a configurar adreces IP en les interfícies ni per a configurar polítiques de QoS. És a dir, no és un protocol de gestió. Intentant suplir les mancances d'OF quant a configuració de dispositius, ONF ha especificat un protocol anomenat **OF-Config**.

El següent protocol *southbound*, l'ús del qual està molt estès, és l'**Open vSwitch Data Base (OVSDB)**, que va ser creat per Nicira. Originàriament, aquest protocol era una part de l'Open vSwitch (OVS), que és un commutador virtual de codi obert basat en Linux dissenyat per hipervisors de màquines virtuals.

ONF

Vegeu també el web <https://www.opennetworking.org/>

Nicira

Va ser l'empresa que va crear NOX, el primer controlador SDN. Després aquesta empresa va ser adquirida per VMware.

Ara aquest protocol ja no està emmarcat exclusivament en l'ús de l'OVS, sinó que està implementat en altres switches de diversos fabricants, com Arista i Dell, entre d'altres.

En casos en què es desplega un OVS (tant si està físicament en un equip com si està virtualitzat), OpenFlow es continua usant amb aquest commutador virtual per a configurar les taules de fluxos que defineixen les regles de traspàs i enviament de paquets entre ports. No obstant això, OVSDb és usat per a configurar el mateix OVS. És a dir, es poden crear, esborrar i modificar ports i interfícies, i també ponts. Això és perquè OVSDb es considera més un protocol de gestió d'un commutador que un protocol de control d'un commutador (a diferència de OF, que sí que ho és).

Seguim amb el **Network Configuration Protocol (NETCONF)**, que és definit per l'IETF per a instal·lar, manipular i esborrar la configuració de dispositius de xarxa. Aquest protocol s'executa normalment sobre un canal segur usant encapsulats com SSH o TLS i conforma una sèrie de missatges que contenen ordres (anomenats RPC o *remote procedure call*) per a canviar l'estat del dispositiu. No obstant això, la informació continguda en els missatges de NETCONF no és definida per aquest protocol, sinó que està estructurada segons un llenguatge de modelatge de dades anomenat YANG (Yet Another Next Generation). YANG no conté dades per si mateix, sinó que solament especifica com s'estructuren. La informació continguda en les ordres de NETCONF es codifica usant XML o JSON obeint l'estructura que s'ha definit prèviament usant YANG.

YANG defineix com s'estructuren les dades (estructures en forma d'arbre) i proporciona moltes característiques de modelatge incloent un extens catàleg de tipus de variables, configuració de dades i una varietat de regles de semàntica i sintaxi.

Les definicions de dades estan contingudes en mòduls, i això proporciona un conjunt molt potent de característiques per a poder estendre aquestes definicions i reutilitzar-les.

Aquests models de dades basades en YANG poden ser definits completament, per exemple, pel fabricant del commutador i també poden estar basats en models ja existents que han estat definits per organitzacions d'estandardització (com l'IETF o IEEE, que defineixen models genèrics per a configurar les interfícies, rutes IP i polítiques de QoS, entre d'altres) i també per un grup de treball anomenat OpenConfig, format per proveïdors de servei i operadors de telefonia que intenten suplir els buits que han deixat les organitzacions d'estandardització.

NETCONF vs RESTCONF

Hi ha un protocol anomenat RESTCONF. Es pot dir que és el mateix que NETCONF però està encapsulat en HTTP i usa les ordres que proporciona aquest protocol (GET, PUT, POST, etc.). És comú utilitzar RESTCONF com a implementació d'API Northbound.

YANG i ASN.1

ASN.1 era un llenguatge de modelatge de dades de propòsit general usat per a implementar l'estructura d'una MIB de SNMP. YANG ofereix un llenguatge molt més entenedor pel fet de ser llegible per un humà i permet definir moltes més coses, fins i tot els missatges que s'intercanvien entre dos elements de xarxa.

1.5. *APINorthbound* i el desenvolupament d'aplicacions

Tal com hem comentat en seccions anteriors, l'objectiu principal de les *API Northbound*(NBI) és proporcionar una capa d'abstracció que permeti als desenvolupadors d'aplicacions poder ancorar-se a la xarxa i fer-hi canvis per acomodar les necessitats de l'aplicació sense haver d'entendre exactament quins equips de xarxa cal modificar.

Encara que el principal canvi de paradigma que les SDN ofereixen s'implementa gràcies a les *API southbound*, és molt important recalcar el paper que tenen les *APINorthbound*.

Les *API Northbound* són possiblement les més crítiques, ja que aporten el valor afegit a les SDN via aplicacions innovadores que es desenvolupen.

Actualment els protocols REST (Representational State Transfer) semblen els més utilitzats per a implementar les *API Northbound* i la majoria dels controladors SDN que les implementen. Els missatges d'un protocol implementat com REST contenen informació en format XML o JSON, que pot seguir un model basat en YANG.

Així, doncs, es pot dir que cada *API Northbound* dona suport a un tipus d'aplicació en concret, i aquesta és possiblement la raó per la qual aquestes API són el component menys estandarditzat en el context de les SDN.

L'ONF ha obert un grup de treball específic per a les *API Northbound*. Aquesta organització pretén desenvolupar codi i prototips per avaluar si és viable o no desenvolupar un estàndard en aquest component.

Conseqüentment, en el controlador SDN hi haurà un ampli ventall d'interfícies per a controlar diferents tipus d'aplicacions.

Nivell d'abstracció que ofereixen les *API Northbound*

És relativament comú que els controladors SDN ofereixin per defecte una sèrie d'aplicacions senzilles en forma de connectors que ajuden a desenvolupar noves aplicacions. Aquests connectors es poden usar passant per les NBI respectives, però el ventall de funcionalitats que ofereixen aquestes NBI depèn molt del nivell d'abstracció de la capa d'infraestructura.

De fet, és possible trobar connectors de molt baix nivell, com per exemple un connector relacionat directament amb els missatges del protocol OpenFlow (és a dir, que l'aplicació té l'oportunitat de gestionar directament els missatges d'aquest protocol). Però, en canvi, també podem trobar NBI de molt alt nivell, que absteuen totalment qualsevol informació de topologia i de la tecnologia

subjacent (per exemple, aquests connectors poden oferir una API que conté ordres de configuració de polítiques de QoS i rutes IP, en les quals les polítiques s'apliquen en un equip i les rutes en un altre equip diferent sense que l'usuari en sigui conscient).

Seguint aquesta última premissa, podem afirmar que la comunitat que s'encarrega de desenvolupar noves NBI segueix dos corrents:

- **Interfícies relacionades amb la tecnologia subjacent:** aquestes NBI tenen un nivell d'abstracció baix i publiquen capacitats específiques que el sistema de xarxa subjacent pot proporcionar, i, gràcies a una combinació d'aquestes capacitats, els usuaris poden fer aplicacions de xarxa. Per a poder utilitzar aquestes API, l'usuari ha de tenir coneixements de configuració d'equips (configuració de fluxos IP directa en dispositius) a causa del baix nivell d'abstracció, ja que s'han d'utilitzar aquestes NBI per a indicar com oferir un servei de xarxa.

IETF ha fet molts esforços per a aconseguir NBI estàndards d'aquest tipus. IETF ha generat, per exemple, un protocol anomenat ALT (Application-Layer Traffic Optimization), que proporciona informació de xarxa (proporcionant serveis de xarxa com vistes de l'arquitectura de xarxa simplificades, vistes dels recursos de la xarxa, inclosos costos). També es poden trobar altres serveis en forma de NBI, com VPNService, que permet definir una xarxa virtual aïllada d'ús molt típic en xarxes virtualitzades de centres de dades o sistemes d'orquestració (per exemple, en crear dues màquines virtuals i necessitar que estiguin interconnectades en la mateixa subxarxa).

- **Interfícies agnòstiques en la tecnologia subjacent.** Aquestes NBI tenen un nivell d'abstracció més alt i solen anomenar-se en anglès *intent* NBI. L'usuari d'aquestes API es focalitza en allò que el servei necessita oferir específicament sense haver de preocupar-se de configuracions de més baix nivell, reduint molt la dificultat d'operació de la infraestructura de xarxa. En aquestes API no cal especificar com oferir el servei, sinó que ofereixen simplement el servei mateix. Aquestes NBI són molt més complexes, ja que tenen capacitats d'orquestració integrades d'altres connectos que busquen aquest nivell d'abstracció tan alt.

Hi ha un exemple d'aquest tipus d'API que val la pena remarcar. OpenStack necessita una NBI específica per a poder interconnectar dues o més màquines virtuals formant una xarxa completament virtual. Aquesta NBI es diu Neutron.

OpenStack

L'OpenStack, desenvolupat en col·laboració entre la NASA i RackSpace, és el projecte de programari de codi obert per a gestionar plataformes núvol més actiu avui dia.

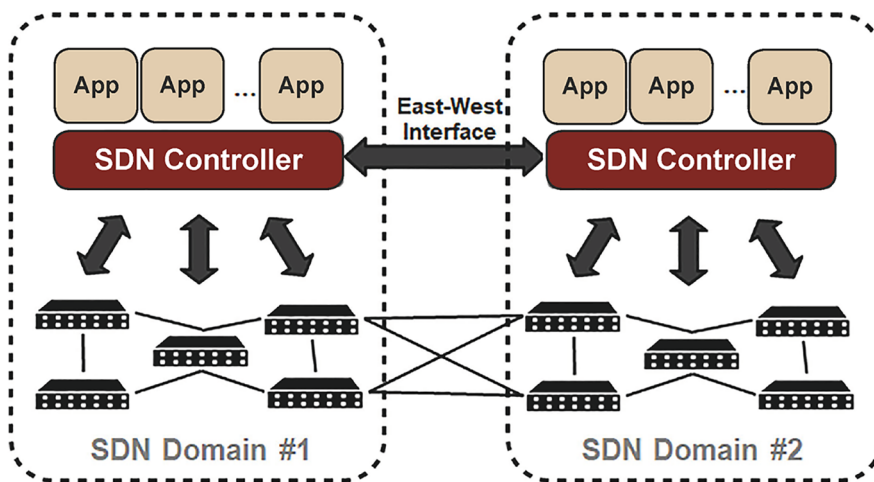
1.6. Interconnexió entre controladors SDN

Tal com hem comentat en seccions anteriors, des del punt de vista d'un controlador SDN hi ha una interfície *Northbound* per a desenvolupar aplicacions i una interfície *Southbound* per a controlar la capa d'infraestructura. També hem dit que un controlador SDN gestiona un domini sencer que engloba un nombre determinat d'equips de xarxa en la capa d'infraestructura.

Ara bé, per raons òbvies d'escalabilitat (xarxa multidomini), de capacitat de gestió, de separació de xarxes de dos operadors o simplement per raons de privadesa, és possible que s'utilitzin més d'un controlador SDN per a controlar una xarxa.

Aquests controladors SDN han de poder comunicar-se entre ells per a coordinar-se en el control dels respectius dominis, i aquesta comunicació és possible mitjançant una interfície horitzontal, anomenada en anglès *east-west interface* o SDNI (vegeu la Figura 5). Aquesta interfície és gestionada pels mateixos controladors.

Figura 5. Interconnexió entre dos controladors SDN.



Els controladors SDN poden intercanviar-se, amb aquesta interfície SDNI, informació de condicions i topologia de la seva xarxa, esdeveniments, fallades dels mateixos controladors, requeriments de QoS per a certes aplicacions, etc.

1.7. Exemple d'ús de les SDN

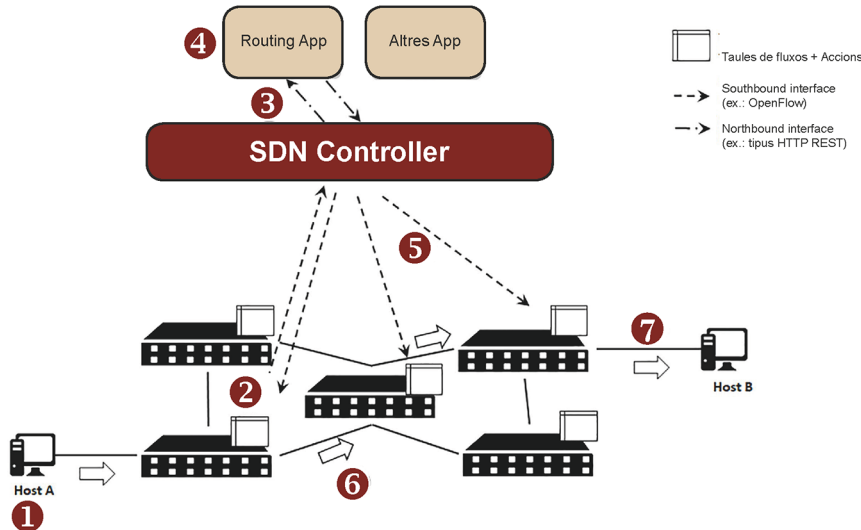
En aquesta secció mostrarem un exemple o cas d'ús molt senzill de com interacciona un controlador SDN amb la capa d'infraestructura i d'aplicació.

En la Figura 6 tenim una aplicació que fa funcions d'encaminament senzill (*routing app*) i que interacciona amb el controlador SDN passant per una API NB (API *Northbound*) basada en HTTP REST. Amb aquesta API el controlador rep notificacions provinents de la capa d'infraestructura, però processades i orquestrades degudament gràcies al controlador SDN. Aquesta aplicació també té coneixement de la topologia de xarxa, ja que el controlador SDN l'ha posat a disposició de l'aplicació.

El controlador SDN interacciona amb la capa d'infraestructura (commutadors) mitjançant una API *southbound* (que podria ser OpenFlow, per exemple) i configura les taules de fluxos amb filtres associant accions concretes per al cas de *match* o *miss* (és a dir, quan no fa coincidència).

El cas d'ús que proposem és el d'un amfitrió A que vol enviar un missatge instantani (paquet IP de petita grandària) a altre amfitrió B, el qual està en un lloc remot respecte a l'amfitrió A.

Figura 6. Exemple d'interacció entre la capa d'infraestructura i el controlador SDN.



Vegem pas a pas i a grans trets el que passa:

- 1) L'amfitrió A envia el paquet IP (això no significa que A estigui connectat directament al primer commutador; volem indicar més aviat per on entra aquest paquet en el domini que gestiona el controlador SDN).
- 2) El primer commutador aplica els filtres de la taula de flux que té configurada en aquest moment i detecta que el paquet rebut no fa coincidència en cap entrada de flux (*flow entry*) i, per tant, no sap per quin port treure el paquet. Així, aplica l'acció configurada per defecte, que en aquest cas és enviar el paquet IP encapsulat al canal de control de l'SBI (per exemple, OpenFlow permet encapsular paquets IP per fer-los arribar al controlador SDN) i també sol·licitar que es prengui una decisió d'encaminament.
- 3) El controlador SDN fa funcions d'orquestració i sap que aquest paquet ha de ser enviat via NBI cap a l'aplicació d'encaminament de manera específica (i no a altres aplicacions que també poden estar en execució).
- 4) L'aplicació rep el paquet i, segons la seva informació, elabora la configuració d'encaminament fins a la destinació final (amfitrió B). Respon amb el paquet IP i amb les ordres de configuració de rutes (segons que especifiqui l'API de l'NBI).
- 5) El controlador SDN configura la taula de fluxos del primer commutador enviant el paquet IP de tornada perquè surti pel port que toqui, i també configura les taules de fluxos amb *flow entry* més l'acció corresponent (en

aquest cas l'acció és indicar el port de sortida) de tots els commutadors afectats que hi ha pel camí que s'ha decidit.

- 6) Una vegada configurats tots els commutadors, el paquet segueix el seu camí sense cap intervenció més del controlador SDN pel que fa a aquest flux. És a dir, si cal encaminar paquets iguals a l'inicial, els commutadors usaran les taules de fluxos ja configurades prèviament (aquestes entrades en les taules de fluxos duraran fins que es dispari un temporitzador d'inactivitat, moment en el qual les *flow entry* s'eliminaran i ho notificaran al controlador SDN).
- 7) El paquet arriba a la seva destinació.

2. Network function virtualization (NFV)

NFV és la sigla anglesa de *network function virtualization* (**virtualització de funcions de xarxa**). Abans d'explicar aquest concepte és útil especificar què entenem per funcions de xarxa.

Les funcions de xarxa s'entenen com a aplicacions o serveis que compleixen una funció molt concreta i donen intel·ligència i valor afegit a l'ús d'una xarxa.

Parlem, per exemple, d'un tallafoc, un servidor DNS, la funcionalitat d'encaminament d'un encaminador (decisió de rutes IP) o també balanceig de càrrega.

De fet, es pot dir que aquestes funcions de xarxa s'emmarquen en les aplicacions ja esmentades que s'executen en comunicació amb un controlador SDN usant NBI.

No obstant això, entre aquestes funcions de xarxa es preveuen arquitectures més complexes, com per exemple un nucli IMS (amb tots els servidors intermediaris SIP i elements de processament de senyalització SIP i emmagatzematge de perfils que requereix), la part de control i processament de paquets d'una estació base de telefonia mòbil o un EPC sencer.

Fa uns anys era normal trobar en el mercat aquestes funcions implementades en plataformes maquinari completament propietàries, el manteniment i configuració de les quals eren costosos i, a més, aquestes plataformes es mostraven com a solucions molt poc flexibles davant increments de demanda.

En l'*SDN and OpenFlow World Congress* de l'any 2012, celebrat a Darmstadt (Alemanya), es va proposar una solució: les NFV. Aquest mateix any, després d'aquest congrés, l'ETSI va crear un grup de treball específic per estudiar-ne l'estandardització: l'ETSI NFV Industry Specification Group (ISG). Aquest grup es va obrir a qualsevol empresa membre i no membre de l'ETSI, i des de llavors s'ha avançat molt en aquesta especificació.

NFV

Durant el *SDN and OpenFlow World Congress* de 2012 es va publicar un *white paper* (accessible a https://portal.etsi.org/nfv/nfv_white_paper.pdf) en el qual es va definir per primera vegada l'ecosistema de NFV. En la web <http://www.etsi.org/technologies-clusters/technologies/nfv> es poden seguir els avenços del grup de treball de l'ETSI encarregat de l'especificació de NFV.

La **virtualització de les funcions de xarxa** és una iniciativa que busca implementar serveis i aplicacions de xarxa en màquines virtuals que s'executin en un maquinari barat (anomenat **maquinari de propòsit general** o *commodity* maquinari en anglès), fàcilment intercanviable i àmpliament disponible en el mercat.

Cadascuna d'aquestes aplicacions o funcions de xarxa monolítiques virtualitzades s'anomenen **VNF** (*virtual network function*). Diverses VNF (màquines virtuals) poden ser executades en el mateix maquinari monitoritzades i controlades per un hipervisor. En definitiva, es pot dir que és l'aplicació de la informàtica en núvol per a prestar serveis de telecomunicacions.

Quins beneficis comporta la introducció de NFV?

A continuació, descrivim alguns dels avantatges principals d'introduir el model de referència de NFV:

- **Reducció de costos d'implementació (CAPEX) i operació (OPEX).** El fet d'executar programari en un entorn de màquines virtuals fa possible utilitzar maquinari de propòsit general, que és molt més barat que el maquinari propietari. A més, en un entorn així, el manteniment de l'entorn per a proveir serveis correctament requereix menys personal i agilita els procediments d'operació.
- **Rapidesa d'adaptació a variacions de demanda.** La virtualització permet iniciar gairebé instantàniament l'execució de noves instàncies d'un servei adaptant-se als increments o decrements (esporàdics o permanents) del nombre d'usuaris. A més, permet incorporar i integrar molt fàcilment noves unitats de maquinari, augmentant potencialment la capacitat de processament davant futurs increments de demanda.
- **Reducció del temps d'implementació d'un servei (*time-to-market*).** Si un proveïdor de servei vol implementar un servei nou, sia fruit de funcions de xarxa aïllades (és a dir, VNF que implementen tot un servei) o fruit d'un encadenament de serveis (*service chaining* en anglès), el fet de poder executar programari de manera virtualitzada permet compondre i engegar un nou servei gairebé a l'instant.
- **Aïllament total entre diferents implementacions del mateix servei.** És el que s'anomena en anglès *multi-tenancy*. Es poden oferir serveis totalment adaptats i dimensionats a un grup d'usuaris en concret. A més, es fa possible que l'execució de diferents instàncies de funcions de xarxa no s'afecti mútuament gràcies a mecanismes d'aïllament de programari entre domi-

nis administratius (els recursos de computació, emmagatzematge i xarxa estan compartimentats totalment).

- **Reducció del consum d'energia elèctrica.** El desacoblament que la virtualització habilita entre la plataforma maquinari i el programari de funció de xarxa permet utilitzar òptimament els recursos de computació (s'executen diverses màquines virtuals en un sol maquinari esprement-ne al màxim les capacitats sense malgastar recursos de CPU i memòria sobrants).

Per exemple, si durant la nit la demanda d'un servei cau considerablement, es pot concentrar l'execució de programari en menys servidors i apagar la resta de les plataformes o posar-les en mode d'estalvi d'energia.

2.1. Arquitectura de referència de NFV

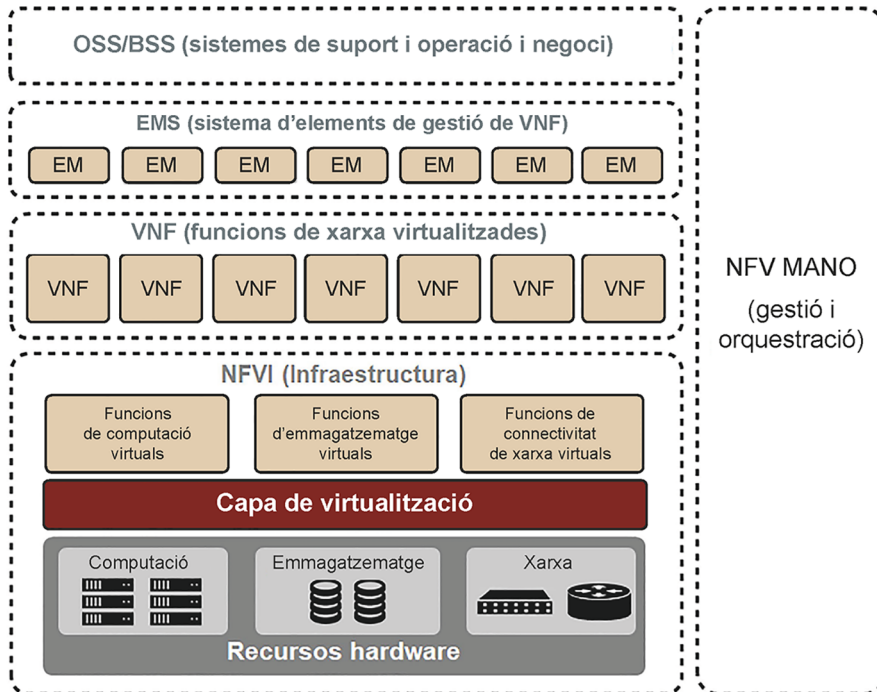
L'ETSI, mitjançant el grup de treball específic per a NFV, ha definit una arquitectura de referència per a entendre bé quines funcionalitats formen aquest ecosistema i com es relacionen o comuniquen entre elles.

Els criteris clau en els quals es va basar aquest grup de treball per a elaborar aquesta arquitectura de referència són:

- **Desacoblament:** separació completa entre el programari i el maquinari.
- **Flexibilitat:** desplegament de funcions de xarxa automatitzades i escalables.
- **Operativitat dinàmica:** control dels paràmetres operacionals de les funcions de xarxa mitjançant un control i monitoratge granular de l'estat de la xarxa.

La Figura 7 mostra aquesta arquitectura.

Figura 7. Model arquitectural de NFV.



A continuació descriurem cadascun dels elements principals d'aquesta arquitectura de referència.

2.1.1. VNF (*Virtual Network Function*)

La VNF es considera el bloc bàsic de l'ecosistema NFV. És un element de xarxa que s'executa en un entorn virtualitzat (màquina virtual). Aquest programari pot ser des d'un encaminador (el qual es podria *router* VNF) fins a una estació base de telefonia mòbil sencera.

Encara que una VNF es concep com la implementació d'una funcionalitat monolítica en forma de programari virtualitzat, diverses subfuncions d'una funcionalitat de xarxa poden ser també implementades com a VNF independents.

Per exemple, un nucli IMS virtualitzat pot estar compost per diverses VNF, cadascuna de les quals implementa un element de nucli: P-CSCF VNF, I-CSCF VNF, S-CSCF VNF i HSS VNF.

Cal tenir en compte que cada VNF és com una màquina virtual que es crea o s'instal·la, s'executa i, finalment, es destrueix quan ja no es necessita. En tot aquest cicle de vida de la VNF el sistema de gestió i orquestració (NFV MANO) s'encarrega de gestionar-la i controlar-la.

Tal com veurem en posteriors seccions, NFV MANO s'encarregarà d'orquestrar l'ús de recursos que els diversos VNF consumeixen amb l'objectiu de fer *service chaining* (o *encadenament de serveis*). És a dir, la provisió d'un servei pot ser el resultat de la combinació i/o concatenació de diverses VNF, cadascuna de les quals implementa una funció de xarxa en concret.

Qui implementa les VNF?

Diferents proveïdors poden implementar i comercialitzar VNF perquè després un proveïdor de servei les utilitzi i/o combini per oferir un servei als usuaris.

Hi ha proveïdors de VNF que implementen VNF que per si soles i, sense necessitat de fer encadenament de serveis, ja ofereixen una funció de xarxa completa i suficient per a oferir un servei.

S'anomena *VNFset* el grup de VNF desplegades per un proveïdor de servei que no té cap interrelació entre elles (no hi ha una seqüència d'encadenament fixa entre elles). En *VNF set* diverses VNF poden treballar de manera col·laborativa per oferir un servei, però la seva implementació és completament independent. No obstant això, quan en un servei es requereix processar dades en una seqüència específica aquest servei s'anomena encadenament de serveis.

2.1.2. NFVI (*network function virtualization infrastructure*)

L'NFVI és l'entorn en el qual s'executa una VNF. Inclou recursos físics, recursos virtuals i una capa de virtualització entre tots dos. D'aquí ve el seu nom: infraestructura. A continuació descrivim cadascun d'aquests recursos:

- **Recursos físics.** Per recursos físics entenem tot allò que és maquinari i que comporta capacitats brutes de processament de paquets, que després la capa de virtualització s'encarregarà de repartir entre les diferents VNF. S'identifiquen tres tipus de recursos físics: recursos **de computació** (servidors de propòsit general), **d'emmagatzematge** (servidors d'emmagatzematge) i **de xarxa** (commutadors i encaminadors físics), juntament amb tot el cablejat que els interconnecta.
- **Recursos virtuals.** Els recursos físics són abstracts en forma de recursos virtuals, que són els que les VNF usen realment. De la mateixa manera, s'identifiquen tres tipus de recursos virtuals: **de computació**, **d'emmagatzematge** i **de xarxa**.
- **Capa de virtualització.** És la responsable de desacoblar el programari del maquinari fent possible que el programari pugui progressar independentment del maquinari, i viceversa. Hi ha dos mètodes de virtualització: basat en **hipervisors** (*hypervisor* en anglès) o basat en **contenidors** (*containers* en anglès). En tots dos casos, els recursos físics virtualitzats són els computacionals i els d'emmagatzematge, mentre que en el cas dels recursos de xarxa aquest element és el controlador SDN que ja coneixem.

Un exemple molt conegut i consolidat de virtualització basat en contenidors és el proposat per OpenStack, anomenat Docker.

2.1.3. EMS (*element management system*)

Abans ens hem referit a les VNF com el programari que implementa una funcionalitat de xarxa en concret en la provisió d'un servei i hem dit que aquest programari s'executa en un entorn virtualitzat.

Recursos físics computacionals

Els recursos computacionals (CPU i memòria) poden venir en forma d'apilament de servidors seguint tècniques de *cluster computing*.

Recursos físics d'emmagatzematge

Els recursos d'emmagatzematge poden venir en forma de dispositius NAS (*network attached storage*) o dispositius connectats a una SAN (*storage area network*).

Diferència entre l'hipervisor i el contenidor

Un hipervisor executa sobre un maquinari que és virtualitzat màquines virtuals que tenen el seu propi sistema operatiu. Un contenidor és molt més lleuger (ocupa molt menys que una màquina virtual) i s'executa sempre sobre un sistema operatiu per a amfitrió.

Però aquestes VNF han d'estar configurades i gestionades perquè compleixin la seva funció. Aquest és el paper del sistema EMS. Aquest sistema alberga el programari (també creat en temps real quan una VNF es crea), que s'encarrega de gestionar funcionalment (cobrint FCAPS) cadascuna de les VNF *security* que s'executen en l'ecosistema NFV. La interfície de gestió pot ser estàndard o propietària de la VNF en particular (i, per tant, proporcionada pel mateix fabricant de la VNF).

FCAPS

És la sigla de *fault* (avaria), *configuration* (configuració), *accounting* (cost), *performance* (rendiment) i *security* (seguretat).

Pot haver-hi un EMS per a VNF o també un EMS que gestiona nombroses VNF. El mateix EMS també pot ser una VNF. No obstant això, el model de referència de NFV preveu a molt llarg termini que l'EMS desaparegui i passi el testimoni de tota la gestió de les VNF a mans d'un element orquestrador de VNF dins de l'NFV MANO: el VNFM o VNF *manager*, que descriurem més tard.

En un pas intermedi a aquesta migració final, els EMS poden ser usats directament pel subsistema OSS per a configurar algunes VNF o fins i tot per a gestionar equips físics (anomenats PNF o *physical network functions*). L'ETSI preveu també que, de manera provisional, una versió reduïda de l'EMS pugui assistir el VNFM per a gestionar funcionalment una VNF en concret (sobretot si es tracta d'una interfície de gestió propietari entre EMS i VNF).

2.1.4. OSS/BSS (Operation Support System / Business Support System)

OSS/BSS inclou una col·lecció de sistemes i aplicacions que un proveïdor de serveis utilitza per a operar el seu negoci.

D'una banda, l'OSS (Sistema de Suport d'Operacions) és l'element que s'encarrega de gestionar tot el sistema en l'àmbit de xarxa. Utilitza un operador de telecomunicacions per a gestionar tota la infraestructura de xarxa que administra. Per exemple, aquest element s'encarrega dels processos de subministrament d'equips i la seva configuració, i també de registrar avaries i incidències.

D'altra banda, BSS (*Business Support System*) o Sistema de Suport de Negoci s'encarrega de la gestió que té a veure amb els clients. És a dir, és el sistema que s'encarrega de registrar les peticions de servei provinents dels clients i de facturar els serveis que han consumit.

En l'arquitectura NFV, el subsistema OSS/BSS pot estar integrat amb NFV MANO (que es veurà amb més detalls més endavant) mitjançant interfícies estandarditzades. Aquesta integració amb NFV no està exempta de reptes que afecten tots dos subsistemes i especialment el de l'OSS. En aquest subsistema sempre s'assumia que la infraestructura de xarxa era més aviat estàtica i la seva

modificació o ampliació havia de ser planificada degudament. En definitiva, l'OSS no hi havia estat pensat per a adaptar-se ràpidament a canvis en la tipologia i a la inclusió de nous serveis.

Per a aprofitar la flexibilitat que ofereix NFV, el subsistema OSS necessita ser automatitzat i basat en un **concepte de catàlegs** (veurem què significa en la secció següent). A més, l'OSS també necessita tenir la capacitat d'actualitzar en temps real el seu coneixement dels recursos disponibles. Conseqüentment, també necessita tenir implementats i automatitzats mecanismes molt ràpids per a aplicar noves configuracions en els equips tenint en compte aquesta capacitat d'actualització en temps real.

2.1.5. NFV MANO (Management & Orchestration)

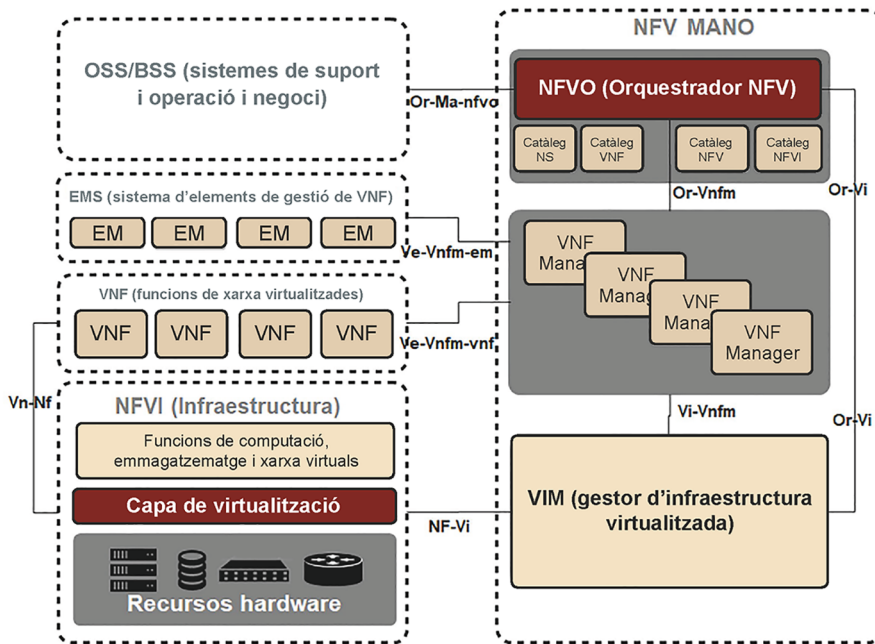
MANO és la sigla anglesa de Management & Orchestration. Aquesta part del model de referència introduït per l'ETSI sobre les NFV és molt important, ja que es considera el cor i cervell de tota l'arquitectura.

Tal com es veu en la figura següent, l'arquitectura de NFV MANO se subdivideix en tres subblocs, que descriurem amb detalls posteriorment:

- VIM (*virtualized infrastructure manager* o gestor d'infraestructura virtualitzada).
- VNFM (*virtual network function manager* o gestor de VNF).
- NFVO (NFV Orchestrator o Orquestador de NFV).

Cadascun d'aquests subblocs interacciona en l'àmbit de gestió amb el seu homòleg de la part de l'arquitectura que hem descrit abans.

Figura 8. Model arquitectural de NFV MANO.



VIM (*virtualized infrastructure manager*)

Aquest element és responsable de gestionar tots els recursos (computació, emmagatzematge i xarxa) dins el domini administratiu d'infraestructura d'un operador, anomenat també **domini NFVI**.

Dit d'una altra manera, el VIM s'encarrega de crear, mantenir i destruir màquines virtuals pel que fa als recursos físics en aquest domini NFVI. Cadascuna d'aquestes màquines virtuals tindrà associats una sèrie de recursos de maquinari mitjançant la capa de virtualització. Així, el VIM manté un inventari de totes les màquines virtuals associades a cada recurs físic.

El VIM també s'encarrega de mantenir un inventari de tots els dispositius de *maquinari*, a més de monitorar i controlar el rendiment i les fallades d'aquests recursos de maquinari, programari i funcions de recursos virtualitzades.

Aquest subbloc implementa la seva pròpia API NB per exposar així els recursos disponibles, tant físics com virtuals, a altres elements de gestió dins de l'arquitectura NFV i especialment a l'NFV Orchestrator.

VNFM (*virtual network function managers*)

Els VNFM gestionen totes les VNF que s'executen al llarg de tot el seu cicle de vida. És a dir, creen, mantenen i destrueixen cadascuna de les instàncies de VNF. Aquestes VNF necessiten ser instal·lades en màquines virtuals, les quals són creades i gestionades pel VIM, com ja hem vist.

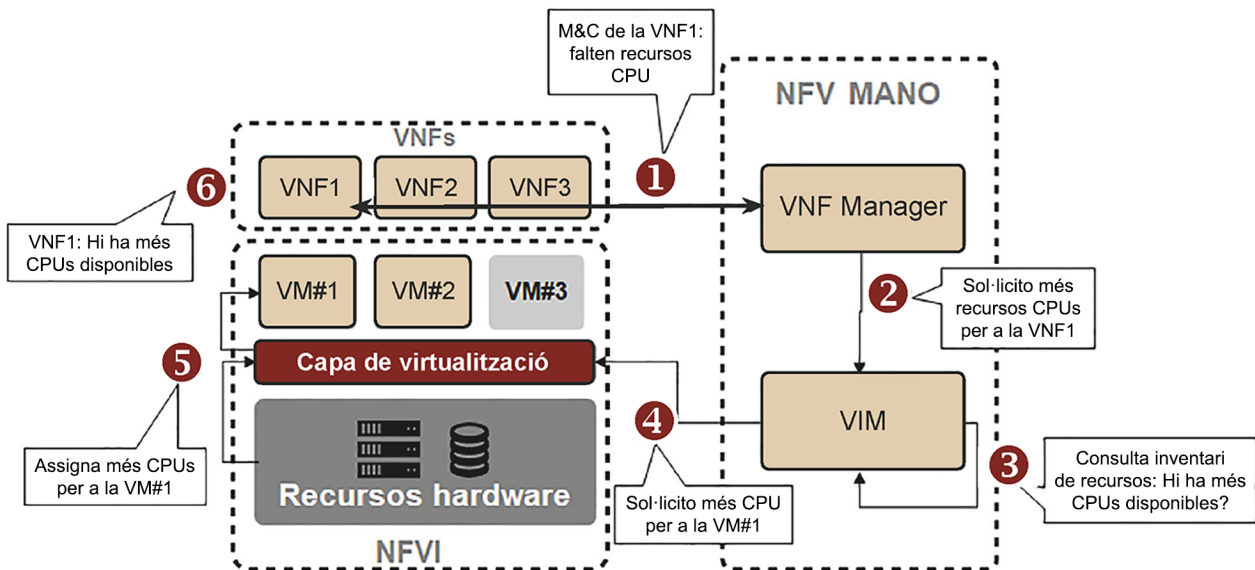
Domini NFVI

En una sola arquitectura NFV pot haver-hi més d'un domini NFVI, cadascun gestionat pel seu corresponent VIM.

Relacionat amb la gestió, una VNFM s'encarrega de cobrir les necessitats de gestió i monitoratge en els àmbits de configuració, costos, rendiment i seguretat de cada VNF. Fins i tot s'encarreguen de sol·licitar al VIM l'increment o alliberament de recursos (computacionals o d'emmagatzematge) segons que necessiti la VNF que gestiona. Se'n pot veure un exemple en la Figura 9.

Cada VNF pot disposar del seu propi VNFM, o un sol VNFM pot gestionar nombroses VNF.

Figura 9. Exemple d'interacció entre elements de NFV per a sol·licitar més recursos CPU.



NFVO (NFV Orchestrator)

Aquest element és la pedra angular de NFV, ja que aglutina tota la intel·ligència de l'arquitectura de referència i la seva funció és crear serveis d'extrem a extrem utilitzant tots els recursos disponibles (NFVI) i les funcions de xarxa virtualitzades (VNF).

Es pot dir que l'NFVO fa dos tipus d'orquestració:

- **Orquestració de recursos.** Gestiona tots els recursos que publica el VIM passant per la seva API *Northbound* dedicada, i en té una visió global. L'NFVO pot gestionar diversos VIM de diferents dominis NFVI. Així, s'encarrega de coordinar, autoritzar, alliberar i assignar aquests recursos.
- **Orquestració de serveis.** Aquesta orquestració s'encarrega de fer la concatenació de funcions de xarxa (és a dir, concatenació de VNF) per oferir un servei d'extrem a extrem. Aquestes VNF seran gestionades per les seves VNFM respectives. Aquestes VNFM són instanciades per aquest orquestrador i iniciaran al seu torn les VNF. L'NFVO té coneixement de la topologia de xarxa per a cada instància de servei d'extrem a extrem (és a

dir, de com les VNF estan interconnectades per a un servei en concret). Aquesta topologia s'anomena VNF-Forwarding-Graph (VNF-FG) o també *service chaining*.

Repositoris

Hi ha quatre tipus de repositoris:

- **Catàleg de VNF.** Aquest catàleg conté tots els descriptors de VNF (VNFD o *VNF descriptor*) que són usables. Un descriptor de VNF és com una plantilla que descriu una VNF en termes de desplegament i requeriments d'operativitat. De manera genèrica, es poden identificar tres paràmetres que identifiquen un VNFD:
 - **VDU** (*virtual deployment unit*). S'hi especifica el nom de la imatge on se situa la VNF i els requisits de CPU, memòria i emmagatzematge recomanats per a executar aquesta VNF en una VM.
 - **CP** (*connection points*). S'hi especifica el nombre d'interfícies de xarxa (virtuals) que la VM ha d'habilitar per a la VNF. Cada CP ha d'anar associada a una VDU i a un enllaç virtual (*virtual link*).
 - **VL** (*virtual link*). Opcionalment, es pot indicar l'identificador del node virtual al qual connectar cada CP que inclogui la VNF. La interconnexió a aquest VL pot ser substituïda per l'aplicació d'un descriptor de servei de xarxa per a interconnectar diverses VNF.

Qui usa aquest descriptor és principalment el VNFM durant el procés de la instanciació de la VNF i la gestió del seu cicle de vida. També és usat per l'NFVO per a gestionar i orquestrar els serveis de xarxa i els recursos virtualitzats en l'NFVI (via VIM, com ja sabem).

- **Catàleg de serveis de xarxa.** Conté un altre tipus de plantilla més relacionada amb els serveis de xarxa que es necessiten per a interconnectar VNF. Els descriptors (NSD o *network service descriptor*) d'aquest catàleg descriuen la connectivitat entre VNF mitjançant enllaços virtuals. D'aquesta manera, poden interconnectar d'un sol cop una sèrie de VNF per a prestar un servei en concret.
- **Instàncies NFV.** Són una llista actualitzada i detallada de totes les instàncies creades i actives, tant de VNF com de serveis de xarxa.
- **Recursos NFVI.** Són un repositori sobre els recursos NFVI utilitzats amb el propòsit d'executar serveis VNF.

2.2. Punts de referència en NFV

Tal com es pot veure en la Figura 8, hi ha una sèrie d'interfícies o punts de referència que comuniquen els blocs de NFV-MANO amb els elements restants de l'arquitectura de referència de NFV.

En les explicacions de les seccions anteriors ja hem deixat entreveure la funcionalitat de cadascuna d'aquestes interfícies. A continuació presentem un resum de les funcionalitats de cadascuna:

- **Os-Ma-nfvo.** Defineix la comunicació entre OSS/BSS i l'NFVO. S'usa per a descriure serveis i indicar els grups de VNF necessaris per a aquests serveis. També s'usa per a gestionar el cicle de vida, tant del mateix servei com de les VNF que participen en el servei (quan crear-les i quan finalitzar-les), i definir qui està autoritzat a accedir a la instància del servei de xarxa. Finalment, amb aquesta interfície s'envien esdeveniments sobre el rendiment i l'ús d'aquestes instàncies de serveis cap a l'OSS/BSS.
- **Ve-Vnfm-vnf.** Defineix la comunicació entre el VNFM i el VNF i és usada pel VNFM per a gestionar el cicle de vida de la VNF i per a intercanviar informació del seu estat (esdeveniments) i de la seva configuració. Per exemple, el VNFM pot instanciar, actualitzar, escalar i eliminar les màquines virtuals (VM) on s'executen les VNF.
- **Ve-Vnfm-em.** Va ser definida junt amb la interfície Ve-Vnfm-vnf, però ara se n'ha separat per especificar la comunicació entre el VNFM i els blocs funcionals de l'EM (*element management*). Aquesta interfície suporta la gestió del cicle de vida de la VNF, però solament s'usa si l'EM és integrable amb el concepte de virtualització que proposa NFV. Amb l'EM, el VNFM també pot instanciar, actualitzar, escalar i eliminar les màquines virtuals (VM) on s'executen les VNF.
- **Nf-Vi.** Defineix l'intercanvi d'informació entre el VIM i els blocs de l'NFVI. El VIM la usa per a assignar, gestionar i controlar els recursos de NFVI. El VIM pot iniciar, actualitzar, migrar i eliminar màquines virtuals (VM). També pot crear, configurar i llevar connexions entre màquines virtuals. Finalment, el VIM rep esdeveniments, registres d'usos i informació de configuració des del NFVI.
- **Or-Vnfm.** Defineix la comunicació entre l'NFVO i el VNFM i s'usa per a instanciar VNF i altres tipus d'informació relacionada amb el cicle de vida d'aquestes (retransmet esdeveniments provinents de les VNF).
- **Or-Vi.** És usada per l'NFVO per a tenir una via de comunicació amb el VIM i influir en la gestió dels recursos d'infraestructura (reserva, actualització i alliberament de recursos per a màquines virtuals, o instal·lació i

desinstal·lació de VNF). Finalment, s'usa per a reexpedir esdeveniments i informació de configuració des de l'NFVI fins a l'NFVO.

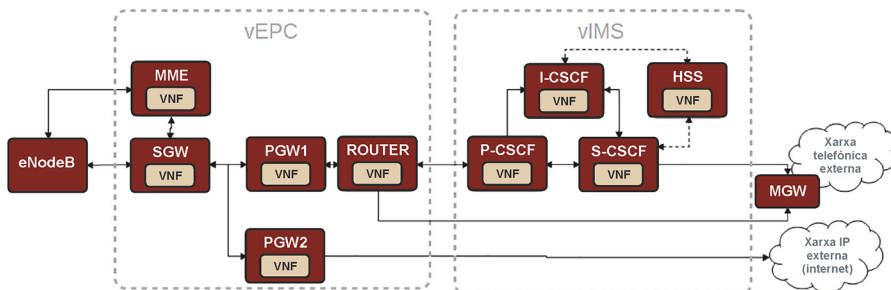
- **Vi-Vnfm.** Defineix les directrius per a intercanviar informació entre el VIM i el VNFM, com per exemple la sol·licitud i actualització de recursos de l'NFVI per a les màquines virtuals que executen una VNF. També s'usa per a rebre esdeveniments i mesures de recursos NFVI usats per una VNF.
- **Vn-Nf.** És l'única interfície que no té una funcionalitat relacionada amb la gestió. Aquest punt de referència comunica necessitats de recursos i portabilitat de la VNF al bloc d'infraestructura.

Cal tenir clar que l'ETSI no ha definit cap recomanació de protocol per a implementar aquests interfícies. Solament n'ha definit les funcionalitats.

2.3. Exemple de *service chaining*: implementació d'un EPC més un nucli IMS

Una vegada vista l'arquitectura de referència de NFV, és hora d'exposar un exemple que enllaci amb el contingut del mòdul anterior. És per això que descriurem a alt nivell com s'implementaria un vEPC (*evolved packet core* virtualitzat) més un vIMS (nucli IMS virtualitzat) en un entorn virtualitzat com l'exposat en aquest model de referència.

Figura 10. Arquitectura a implementar en NFV: EPC virtualitzat i nucli IMS virtualitzat.



En la Figura 10 veiem l'arquitectura del servei que es vol implementar.

- **EPC virtualitzat.** És compost per un MME que autentica els terminals d'una eNB física que és fora de l'àmbit administratiu de l'entorn NFV. Aquesta eNB també té una SGW associada. Finalment, aquest vEPC conté dues PGW: una per al trànsit IMS i una altra per a la connexió a internet.
- **Nucli IMS virtualitzat.** És compost per un P-CSCF, un I-CSCF, un S-CSCF i un HSS. Per a passar les trucades de veu a xarxes heretades, es disposa d'una MGW que té dos punts de connexió físics per separat: un per a la senyalització IMS i un altre per al trànsit de veu RTP que arriba directament de la PGW2.

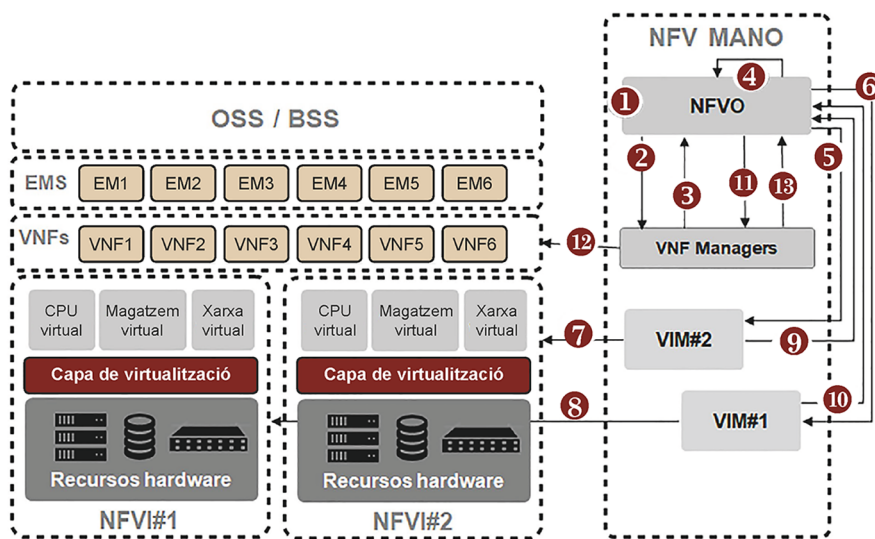
Per a fer l'exemple més complet, el vEPC i el vIMS s'implementen en dos dominis NFVI diferents, el segon dels quals té connectivitat amb les xarxes externes: de telefonia via MGW i a internet.

El proveïdor de VNF ha implementat una VNF per a cada element del vEPC. Cadascun dels elements funcionals que conformen el vEPC s'implementa com una VNF. Així, doncs, en el catàleg de VNF (en l'NFVO) tenim cadascuna de les VNF que conformen un EPC. Això mateix s'aplica als elements del nucli IMS.

Cada VNF disposa d'un descriptor en el qual s'indiquen requeriments aproximats de CPU, memòria, emmagatzematge i xarxa. En aquest descriptor apareixen també el nombre de punts de connexió (*connection points* en anglès) de la VNF, és a dir, quants interfícies de xarxa externes es necessiten com a mínim per a interconnectar-se amb altres VNF.

En el cas del catàleg de servei de xarxa, es disposa de plantilles per a establir els enllaços virtuals (o *virtual links* en anglès) on es detalla la interconnexió en l'àmbit de xarxa entre VNF per a conformar un servei.

Figura 11. Diagrama pas a pas de desplegament de VNF per a vEPC i vIMS.



A continuació descrivim pas a pas com es despleguen les VNF per a conformar tant el vEPC com el vIMS:

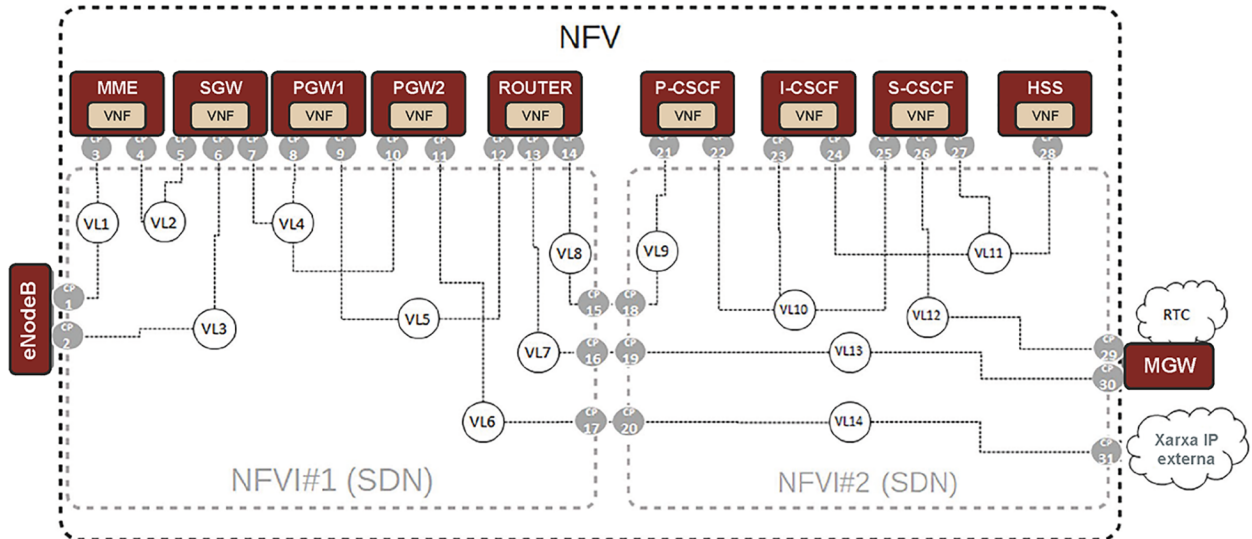
- 1) El NFVO té una visió completa dels tipus de VNF que conformen tant el vEPC com el vIMS.
- 2) El NFVO accedeix al catàleg de VNF, selecciona i instancia les VNF de MME, SGW i dues PGW per al vEPC, una VNF de encaminador d'interconnexió per a la PGW1 i una VNF HSS, P-CSCF, I-CSCF i S-CSCF. Una vegada seleccionades, es comuniquen als VNFM corresponents (un per VNF).

- 3) El conjunt de VNFM determina el nombre de màquines virtuals (VM) necessàries per a executar totes les VNF, i els recursos mínims necessaris de cadascuna d'aquestes VM. Aquesta informació es retransmet a l'NFVO.
- 4) Com que l'NFVO manté informació detallada i actualitzada sobre els recursos de maquinari (en comunicació amb tots dos VIM via NBI Or-Vi), valida si hi ha suficients recursos disponibles per a totes les màquines virtuals (VM) a executar. L'NFVO necessita instanciar una sol·licitud per a disposar d'aquestes màquines virtuals creades.
- 5) L'NFVO envia al VIM#2 la sol·licitud per a crear les màquines virtuals instanciades per al vIMS i reservar els recursos necessaris per a aquestes VM.
- 6) L'NFVO envia al VIM#1 la sol·licitud per a crear les màquines virtuals instanciades per al vEPC i reservar els recursos necessaris per a aquestes VIM.
- 7) El VIM#2 sol·licita a la capa de virtualització del NFVI#2 la creació d'aquestes VIM per al vIMS (inclou també la formació dels *virtual links* entre les VNF usant el controlador SDN present en la capa de virtualització).
- 8) El VIM#1 sol·licita a la capa de virtualització del NFVI#1 la creació d'aquestes VIM pe al vEPC (inclou també la formació dels *virtual links* entre les VNF usant el controlador SDN present en la capa de virtualització).
- 9) El VIM#2 reporta al NFVO la creació reeixida de les VIM del vIMS.
- 10) El VIM#1 reporta al NFVO la creació reeixida de les VIM del vEPC.
- 11) El NFVO notifica al conjunt de VNFM que les màquines virtuals necessàries ja estan disponibles per a executar-hi les VNF.
- 12) Cada VNFM configura cadascuna de les VNF del vEPC i del vIMS amb els paràmetres necessaris per a oferir el servei.
- 13) Una vegada configurades amb èxit totes les VNF, el VNFM comunica al NFVO que les VNF estan preparades, configurades i disponibles per a usar-se.

Finalment, en la Figura 12 veiem el diagrama final de desplegament del vEPC i del vIMS. Podem observar que cada VNF té una sèrie de punts de connexió (CP o *connection points*), que ja hem esmentat abans, però el més sorprenent són els enllaços virtuals (*virtual links*) que interconnecten totes les VNF i que s'implementen usant la tecnologia flexible de SDN, que hem explicat anteriorment. L'encaminador VNF també es podria implementar via SDN.

Cal destacar que en l'àmbit de l'NFV s'identifiquen CP específics que estan mapats a ports físics com a ports d'entrada del trànsit a la zona administrativa del proveïdor de servei que gestiona l'ecosistema NFV.

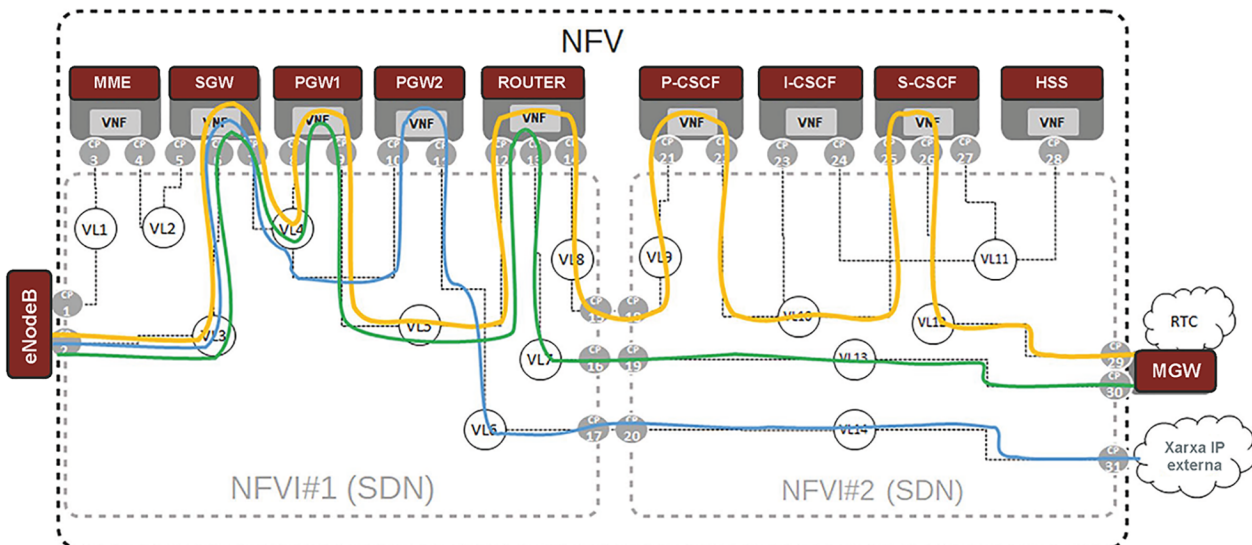
Figura 12. Diagrama de la implementació del vEPC i el vIMS amb els seus *virtual links*.



En la Figura 12 podem veure els *virtual links* que es configuren per connectar totes les VNF entre elles. Cada *virtual link* configura un *VNF forwarding graph* (VNFFG). En aquest exemple tenim fins a 14 VNFFG.

A partir d'aquests VNFFG podem descriure diversos *network forwarding paths*. Per exemple, el trànsit SIP d'una trucada IMS a RTC seguiria el camí següent, indicat amb color groc en la Figura 13. Per al trànsit de veu RTP, se seguiria l'NFP de color verd. Finalment, per al trànsit d'internet seria la línia blava.

Figura 13. *Network forwarding paths* per a diferents trànsits.



2.4. Plataformes existents que implementen NFV

Hi ha proveïdors de servei que ja han optat per fer desplegaments usant NFV. És interessant veure quines opcions d'implementació hi ha disponibles en el mercat a l'hora de prendre decisions d'inversió en la futura provisió de serveis. Les principals plataformes de NFV són OpenStack i vCloud NFV de VMware.

OpenStack

És un programari gratuït de codi obert que ha estat desenvolupat amb la col·laboració entre la NASA i RackSpace. Actualment és el projecte de codi obert per a gestionar plataformes núvol més actiu. OpenStack és un conjunt d'eines de programari de codi obert per a implementar i gestionar arquitectures d'informàtica en núvol. Habilita als proveïdors de servei proveir i orquestrar clústers de recursos de centres de dades. Una empresa que vulgui implementar la seva pròpia plataforma NFV pot intentar internament muntar-la amb els seus propis recursos, encara que hi ha nombroses empreses que comercialitzen la seva pròpia implementació d'OpenStack, entre les quals podem trobar Red Hat, Dell, Hewlett Packard Enterprise, Ericsson, VMware i Mirantis.

VMware vCloud NFV

VMware és el principal proveïdor de programari de virtualització per a grans empreses, inclosos departaments TIC interns dels proveïdors de serveis més importants. vCloud NFV de VMware és una plataforma d'infraestructura modular i integrada que és optimitzada per a desplegar NFV en proveïdors de servei. A més, té una plataforma OpenStack integrada que habilita els proveïdors de servei per a desplegar ràpidament aplicacions NFV sobre plataformes OpenStack.

Qui ha implementat OpenStack fins avui?

Entre els proveïdors de serveis que han implementat OpenStack per a la infraestructura NFV, hi ha: AT&T, Verizon, XO Communications, Comcast, Time Warner Cable, NTT, SK Telecom, China Mobile Ltd., China Unicom, Deutsche Telekom, Swisscom, British Telecom Group i Telefónica.

Qui ha implementat vCloud fins avui?

VMware disposa de més de vuitanta desplegaments de NFV amb més de quaranta-cinc proveïdors de servei.

Resum

La nova generació de serveis que estan introduint les xarxes 5G té una sèrie de requisits de QoS cada vegada més heterogenis, però, a més, aquelles empreses que prestin aquests serveis hauran de ser capaces d'adaptar-se de manera dinàmica i ràpida als canvis de demanda que es puguin produir sense repercutir en costos de desplegament i operativitat elevats.

En l'arquitectura tradicional de les xarxes que s'han implementat fins ara la intel·ligència i coneixement de la tipologia s'ha distribuït al llarg de tots els elements físics que conformen la xarxa. Això comporta una certa rigidesa i costos d'operativitat (hores dedicades per enginyers) elevats a l'hora d'adaptar la xarxa a canvis ràpids de topologia i a canvis en el dimensionament de la xarxa per a adaptar-se a pics de demanda d'un servei.

Les *software defined networks* introdueixen un nou paradigma en què es proposa un model centralitzat totalment en el qual la intel·ligència i coneixement de tota la topologia de la xarxa es concentra en un sol punt (controlador SDN), mentre que es controlen remotament tots els dispositius de xarxa que conformen la capa d'infraestructura i que processen els paquets (commutadors o enrutadors).

Aquest desacoblament entre la capa de dades i la capa de control és possible gràcies al controlador SDN, que actua com un autèntic sistema operatiu de xarxa, i les API *southbound* (SB API) que interconnecten el controlador SDN i els commutadors que conformen la capa d'infraestructura. Entre els controladors SDN destaquen plataformes de codi obert com Opendaylight i Open Networking Operating System (ONOS). En l'àmbit d'API SB, destaquen sobretot Openflow, NETCONF i Open Virtual Switch Database (OVSDB).

Sobre aquests controladors SDN es poden executar aplicacions que implementin funcionalitats tan bàsiques com un enrutador o un tallafoc, o elements més complexos com priorititzadors de paquets. Sigui quina sigui l'aplicació, haurà d'interactuar amb el controlador SDN mitjançant les *Northbound* API que ofereixi.

No obstant això, perquè un proveïdor de serveis pugui disposar d'una infraestructura flexible i adaptable a canvis de models de negoci o a increments o decrements sobtats de demanda, necessita executar les aplicacions (que són les que donen valor afegit i intel·ligència al servei) en un entorn completament virtualitzat on es pugui agilitar al màxim l'assignació a aquestes aplicacions de recursos de maquinari en els àmbits de computació, memòria, emmagatzematge i xarxa.

L'ETSI ha dedicat esforços per a especificar l'ecosistema que defineix la *network functions virtualization* (NFV), que en definitiva és aplicar la tecnologia d'informàtica en núvol per a prestar serveis de telecomunicacions.

Les tecnologies SDN i NFV es complementen a l'hora de proporcionar serveis, ja que la primera virtualitza els recursos de xarxa i la segona ofereix una plataforma de gestió i orquestració per a automatitzar la creació i interconnexió de màquines virtuals per a executar els servidors o funcions en forma de virtual network functions (VNF). La interconnexió de VNF per a proporcionar un servei s'anomena *service chaining*. Aquesta plataforma fa possible, per exemple, virtualitzar tot un EPC (vEPC) o un nucli IMS (vIMS).

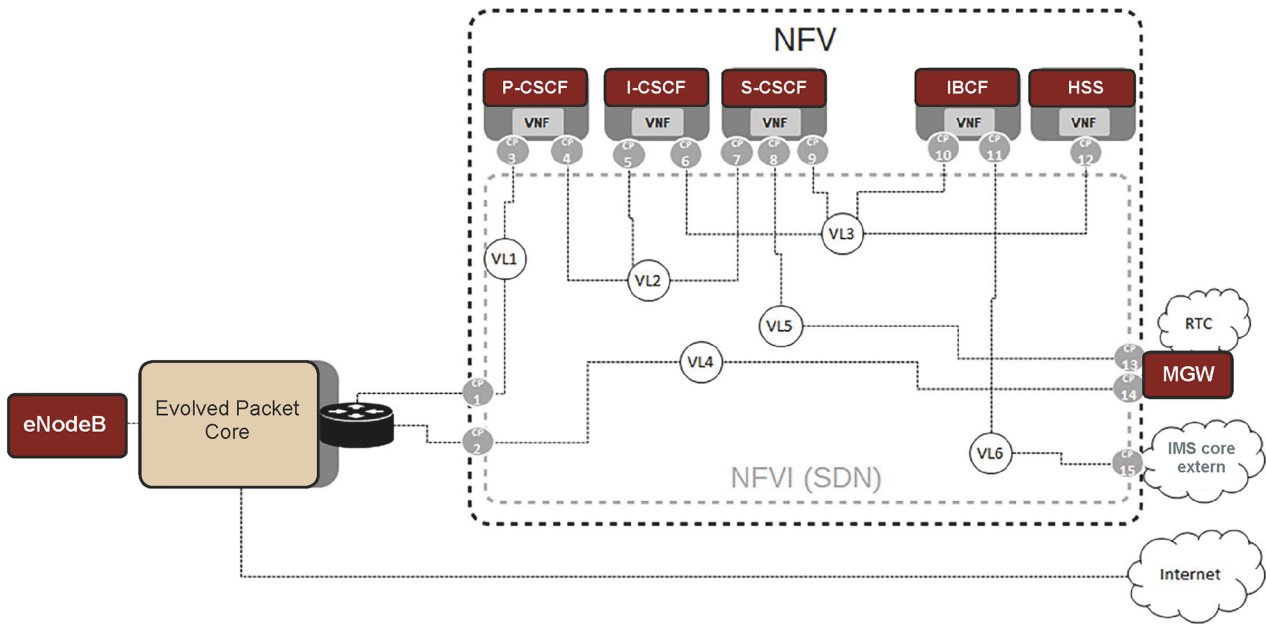
Hi ha dues plataformes que implementen tot l'ecosistema que especifica NFV: OpenStack (plataforma de codi obert de referència) i vCloud de VMware.

Solucionari

1.

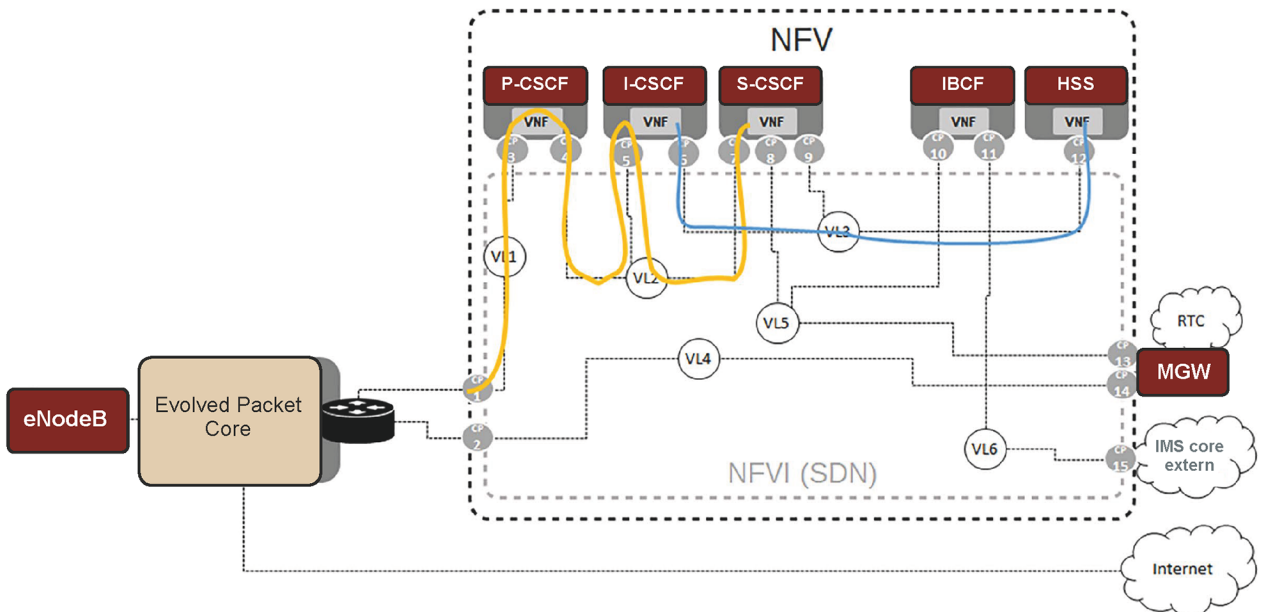
a) El problema que s'observa és la interconnexió entre el CP10 i el VL3, el qual serveix per a interconnectar els elements del nucli IMS que usen Diameter amb l'HSS. És com si haguéssim aïllat l'IBCF de la resta del nucli IMS. La solució proposada és interconnectar el CP10 amb el VL5, que és interconnectar k'S-CSCF amb la xarxa externa (via MGW), tal com mostra la Figura 15.

Figura 15. Solució de l'exercici 1a.



b) La Figura 16 mostra els dos NFP. El groc és l'NFP de SIP i el blau el de Diameter per a un registre IMS

Figura 16. Solució de l'exercici 1b.



c) Neutron.

d) El VL2.

e) OVS (*open vSwitch*), ja que es pot implementar exclusivament via programari.

Glossari

5G Abreviació utilitzada per a referir-se a la cinquena generació de tecnologia de telefonia mòbil. És la successora de la tecnologia 4G.

API *Application programming interface*. Conjunt de funcionalitats de programació estandaritzades per a desenvolupar una funcionalitat comuna en altres aplicacions heterogènies.

ASN.1 Abstract Syntax Notation One. Norma per a representar dades independentment de la màquina que s'estigui usant i les seves formes de representació internes. És un protocol de nivell de presentació en el model OSI. El protocol SNMP usa l'ASN.1 per a representar els seus objectes gestionables.

CAM *content-addressable memory*. Tipus de memòria de computador emprada en determinades aplicacions que requereixen velocitats de cerca molt elevades.

Diameter Protocol evolució del RADIUS per a desenvolupar aplicacions d'AAA.

EMS Element Management System. Sistema que proporciona la gestió de xarxa de les funcions de xarxa virtualitzades (VNF) i els elements de xarxa físics.

ETSI European Telecommunications Standards Institute. Organització d'estandardització de la indústria de les telecomunicacions (fabricants d'equips i operadors de xarxes) europea, amb projecció mundial. <http://www.etsi.org>

FCAPS Model i entorn de treball de xarxa de gestió de telecomunicacions d'ISO per a gestionar xarxes. És un acrònim de *fault, configuration, accounting, performance, security* ('falla, configuració, comptabilitat, acompliment, seguretat'), que són les categories en les quals el model ISO defineix les tasques de gestió de xarxes.

HSS Home Subscriber Server. Base de dades que emmagatzema la informació de subscripció d'un usuari juntament amb informació d'autenticació i autorització en l'àmbit de servei. És el model de referència de 3GPP.

I-CSCF *Interrogating call session control function*. Component del nucli IMS que fa d'element d'encaminador de la senyalització SIP cap a l'S-CSCF correcte dins del seu mateix domini. És un element definit per 3GPP.

IMS IP Multimedia Subsystem. Estàndard definit per 3GPP per a proveir serveis multimèdia en telefonia mòbil, basat en els protocols definits per IETF, com SIP, RTP o Diameter.

MIB *Management information base*. Tipus de base de dades que conté informació jeràrquica, estructurada en forma d'arbre, de tots els paràmetres gestionables en cada dispositiu gestionat d'una xarxa de comunicacions.

NBI *Northbound interface*. Interfície que una entitat usa per a publicar en una altra entitat els serveis programàtics que ofereix.

NETCONF *Network Configuration Protocol*. Protocol de gestió de xarxa desenvolupat i estandaritzat per l'IETF que proporciona mecanismes per a instal·lar, manipular i eliminar la configuració dels dispositius de xarxa.

NFV *Network function virtualization*. Enfocament de xarxa en evolució que permet substituir dispositius de maquinari dedicats i costosos, com encaminadors, commutadors, tallafocs i equilibradors de càrrega, amb dispositius de xarxa basats en programari que s'executen com a màquines virtuals en servidors estàndard de la indústria.

NFV MANO Network Functions Virtualization Management and Orchestration. Marc arquitectònic per a administrar i orquestrar funcions de xarxa virtualitzades (VNF) i altres components de programari.

NFVI *Network function virtualization infrastructure*. Component clau de l'arquitectura NFV que descriu els components de maquinari i programari en els quals es creen les xarxes virtuals. Defineix estàndards per a recursos de computació, emmagatzematge i xarxes que es poden usar per a construir funcions de xarxa virtualitzades.

NFVO *Network function virtualization orchestrator*. Component clau del marc arquitectònic NFV-MANO que fa orquestració de recursos i orquestració de serveis de xarxa, entre altres funcions. És un component central d'una solució basada en NFV. Uneix diferents funcions per crear un servei d'extrem a extrem coordinat.

OF Openflow. Un dels primers estàndards de xarxes definides per programari (SDN). Originàriament va definir el protocol de comunicació en entorns SDN que permet que el controlador SDN interactui directament amb la capa de dades dels dispositius de xarxa, com commutadors i encaminadors, tant físics com virtuals (basats en hipervisor), per adaptar-se millor als requisits comercials canviants.

OpenStack Programari gratuït de codi obert desenvolupat per a implementar NFV. Actualment, és el projecte de codi obert per a gestionar plataformes de núvol més actiu.

OSS/BSS Operational Support System / Business Support System. OSS és un conjunt de programes que ajuden un proveïdor de serveis de comunicacions a monitorar, controlar, analitzar i administrar un telèfon o una xarxa de computadores. BSS són els components que un proveïdor de serveis de telecomunicacions (o companyia de telecomunicacions) utilitza per a executar les seves operacions comercials dirigides als clients.

OVSDB Open vSwitch Database. Protocol de configuració d'OpenFlow dissenyat per a administrar implementacions d'Open vSwitch.

OVS Open vSwitch. Programari de codi obert dissenyat per a ser utilitzat com un commutador virtual en entorns de servidors virtualitzats. És l'encarregat de reexpedir el trànsit entre diferents màquines virtuals (VM) en el mateix amfritrió físic i també de reexpedir el trànsit entre les màquines virtuals i la xarxa física.

P-CSCF *Proxy call session control function*. Component del nucli IMS que fa d'element fronterer amb l'equip d'usuari en l'àmbit de senyalització SIP (IMS). És un element definit per 3GPP.

RTC Xarxa Telefònica Commutada.

RTP Real Time Protocol. Protocol basat en UDP per a transmetre fluxos multimèdia (àudio, vídeo) en temps real.

S-CSCF *Serving call session control function*. Component del nucli IMS que fa de registrador de l'usuari en l'àmbit de capa de control de servei i d'encaminador de la senyalització cap a altres elements que finalitzin la trucada dins del mateix domini o d'un altre diferent. És un element definit per 3GPP.

SBI *Southbound interface*. Interfície usada per una entitat per a programar un tercer dispositiu de manera remota. En el context de SDN, és la interfície que usa el controlador SDN per a controlar la capa d'infraestructura (p. ex., Openflow i NETCONF).

SDN *Software defined network*. Xarxa que utilitza programari en lloc de dispositius especialitzats per a aprovisionar i gestionar els serveis de les aplicacions i de la xarxa, la qual cosa permet la mobilitat i el desplegament d'aplicacions programables, escalables i a demanda. Supervisa de manera intel·ligent i adapta automàticament la xarxa per proveir un millor servei en les condicions de cada moment.

SIP Session Initiation Protocol. Protocol definit per l'IETF per a establir i negociar sessions de serveis multimèdia.

SNMP Simple Network Management Protocol. Protocol de la capa d'aplicació que facilita l'intercanvi d'informació d'administració entre dispositius de xarxa. Els dispositius que normalment suporten SNMP inclouen encaminadors, commutadors, servidors, estacions de treball, impressores, bastidors de mòdem i molts més.

VIM Virtualized Infrastructure Manager. Element de l'entorn arquitectural NFV MANO responsable de controlar i administrar els recursos informàtics, d'emmagatzematge i de xarxa de la infraestructura NFV (NFVI), generalment dins del domini d'infraestructura d'un operador.

VM *Virtual machine*. Sistema operatiu (SO) o entorn d'aplicació instal·lat en un programari que imita el maquinari dedicat.

VNF *Virtual network function*. Tasca virtualitzada duta a terme anteriorment per maquinari propietari i exclusiu. Mou les funcions de xarxa individuals dels dispositius de maquinari dedicats al programari que s'executa en maquinari de propòsit general.

VNFM *Virtual network function manager*. Gestor de VNF durant tot el seu cicle de vida fonamental per a escalar, modificar, agregar nous recursos i comunicar els estats de les VNF a altres blocs funcionals en l'arquitectura NFV-MANO.

YANG Yet Another Next Generation. Llenguatge de modelatge de dades per a definir dades enviades mitjançant el protocol de configuració de xarxa NETCONF.

Bibliografia

Chayapathi, R.; Hassan, S. F.; Shah, P. (novembre de 2016). *Network Functions Virtualization (NFV) with a Touch of SDN*. Addison-Wesley Professional.

Kahn, F. (març de 2015). *A Cheat Sheet for Understanding «NFV Architecture»*. <https://www.telocloudbridge.com/blog/a-cheat-sheet-for-understanding-nfv-architecture/>

Kahn, F. (abril de 2015). *A Beginner's Guide to NFV Management & Orchestration (MANO)*. <https://www.telocloudbridge.com/blog/a-beginners-guide-to-nfv-management-orchestration-mano/>

Rouse, M. (novembre de 2017). *Virtual network functions (VNF)*. <https://searchsdn.techtarget.com/definition/virtual-network-functions>

Rouse, M. (desembre de 2017). *REST (REpresentational State Transfer)*. <https://searchmicroservices.techtarget.com/definition/rest-representational-state-transfer>

Toghraee, R. (maig de 2017). *Learning OpenDaylight*. Packt Publishing.