Afstudeeropdracht: Software Defined Networking

Type document: Literatuurstudie

Auteur: Dhr. François Kikken
Datum: 29 mei 2015

Afstudeeropdracht:
Software Defined Networking

Type document: Literatuurstudie

Format Versie:	**1.0**
Status document:	**Definitief**
Naam van dit document:	**Software Defined Networking - Literatuurstudie_V1.0.docx**
Auteur:	**Dhr. François Kikken**
Datum:	**29 mei 2015**

Document versie beheer

Configuratie beheer			
Versie nr.	**Datum**	**Wijzigingen**	**Auteur**
0.1.0	03-03-2015	Opzet template	Dhr. François Kikken
0.2.0	05-03-2015	Hoofdstuk MAASnet	Dhr. François Kikken
0.3.0	25-03-2015	Hoofdstuk Software Defined Networking	Dhr. François Kikken
0.4.0	20-04-2015	Verplaatsing Hoofdstuk MAASnet naar onderzoeksrapport	Dhr. François Kikken
0.5.0	25-03-2015	Hoofdstuk Software Defined Networking	Dhr. François Kikken
0.6.0	01-05-2015	Invulling begrippenlijst	Dhr. François Kikken
0.7.0	13-05-2015	Verwerken opmerkingen review	Dhr. François Kikken
0.8.0	18-05-2015	Eerste concept versie	Dhr. François Kikken
0.9.0	29-05-2015	Verwerken opmerkingen review	Dhr. François Kikken
1.0	29-05-2015	Definitieve versie	Dhr. François Kikken

Tabel 1: Configuratie beheer

Distributie beheer			
Versie nr.	**Datum**	**Wijzigingen**	**Ontvanger**
0.8.0	18-05-2015	Aanlevering ter review	Ing. Daniël Heynen
1.0	10-06-2015	Definitieve inlevering afstudeerportfolio	Bianca Fintelman

Tabel 2: Distributie beheer

Gerelateerde documenten		
Versie nr.	**Naam**	**Functie**
1.0	Software Defined Networking - Project Initiatie Document	Goedkeuring van de start van het project.

Tabel 3: Gerelateerde documenten

Algemene gegevens

Bedrijfsbegeleider (opdrachtgever)	
Eigenschap	**Gegeven**
Naam	Maastricht University - ICTS Netwerken
Adres	Grote Looierstraat 17, 6211 JH Maastricht
Contactpersoon	Ing. Marc Jonkers

Afstudeerder (opdrachtnemer)	
Eigenschap	**Gegeven**
Naam	Dhr. François Kikken
Adres	Dassenkuillaan 8, 6162 JE Geleen
Telefoon	+31 655 689 801

Schoolbegeleider	
Eigenschap	**Gegeven**
Naam	Zuyd Hogeschool - HBO-ICT
Adres	Nieuw Eyckholt 300, 6419 DJ Heerlen
Telefoon	+31 454 006 060
Contactpersoon	Ing. Daniël Heynen

Projectgegevens	
Eigenschap	**Gegeven**
Documentversie	1.0
Documentsoort	Literatuurstudie
Omschrijving	Software Defined Networking – Literatuurstudie

Voorwoord

Het afstudeertraject betreft het onderwerp Software Defined Networking (SDN) en is aangedragen door de afstudeerder. Ik ben met dit onderwerp in aanraking gekomen tijdens het bijwonen van een, door Cisco Systems aangeboden, conferentie. Na het bijwonen van deze conferentie zijn bij mij hoge verwachtingen van deze techniek en de bijbehorende interesses ontstaan. Naast mijn interesses op dit gebied, heb ik voor SDN gekozen om meer kennis m.b.t. dit onderwerp te vergaren.

De keuze voor de bedrijfslocatie is bij mij ontstaan, doordat ik tijdens mijn MBO-stage eerder in aanraking met de universiteit ben gekomen. Echter, heb ik toen der tijd niets met het netwerk of de netwerktechnieken van doen gehad. Dit terwijl de IT-infrastructuur juist de IT-richting is, waar mijn hartstocht ligt. Het lijkt me de ideale situatie om beide aspecten met elkaar te kunnen combineren.

Ten slotte wil ik ICTS (netwerken) en in het bijzonder ing. Marc Jonkers bedanken voor het mede mogelijk te maken en begeleiden van deze opdracht. Tevens wil ik ing. Daniël Heynen bedanken voor het begeleiden van mijn gehele afstudeertraject.

Maastricht, 7 juli 2015

Begrippenlijst

Begrippenlijst	
Begrip	**Betekenis**
Abstractie	Een abstractie draagt zorg voor een logische en gesimplificeerde weergave van het netwerk. Hierdoor wordt de gedetailleerde weergave van de netwerkapparatuur verborgen.
Application Programming Interface (API)	Een application programming interface is een interface die aan de hand van een uniforme programmeertaal kan communiceren met een computerprogramma.
Application Specific Integrated Circuit (ASIC)	Een ASIC is een integrated circuit (computer chip) gemaakt voor één specifiek doel. De functie en taal van deze chip is na het maken niet meer aan te passen.
ATM-netwerk	Een ATM-netwerk is een netwerk dat gebruik maakt van het protocol Asynchronous Transfer Mode (ATM). Aan de hand van dit protocol wordt een virtueel circuit, over het huidige netwerk heen, gecreëerd. Dit protocol draagt zorg voor het bieden van Quality of Service.
authenticatie-, autorisatie- en accounting- (AAA)	Het AAA-proces wordt gebruikt om elektronische systemen te voorzien van een authenticatie en autorisatie mechanisme voor computernetwerken. Naast het voorzien van een toegangssysteem (authenticatie en autorisatie) worden eveneens de, door de elektronische systemen, uitgevoerde handelingen bijgehouden door het accounting deel van het AAA-proces.
Automatisering	Met behulp van automatisering worden menselijke handelingen (veelal repetitieve taken) uitgevoerd door computersystemen. Binnen netwerkinfrastructuren kan gedacht worden aan het configureren van diverse switches die over dezelfde basisconfiguratie beschikken.
Autonome netwerken	Een autonoom systeem is een zelf- lerend en denkend systeem. Binnen netwerkarchitecturen wordt met autonome systemen gedoeld op routingprotocollen. Deze protocollen berekenen, op basis van algoritmes, het beste pad naar de bestemming. In het geval van een storing ontstaat bij het routingprotocol een bewustheid hiervan en wordt automatisch een nieuwe berekening gemaakt.
Bridged network	Een bridged netwerk is een voorloper van een switched netwerk. Een netwerkbridge werkt volgens hetzelfde principe (distribueren van netwerkverkeer op laag 2 van het OSI-model) als een netwerkswitch. Echter, de bridge is minder geavanceerd.
Central Processing Unit (CPU)	Een CPU is een hardwarecomponent van een computer dat zorg draagt voor basisbewerkingen en -controle van uitvoerbare programmacode.
Cisco Express Forwarding (CEF)	CEF is een Cisco Systems propriëtair geavanceerde, op laag 3 van het OSI-model uitgevoerd, switching technologie.
Cloudoplossing	Cloudoplossingen zijn producten of IT-diensten die op aanvraag (tegen vergoeding) via het internet geleverd worden. Zo bestaat bijvoorbeeld de mogelijkheid om applicaties via software as a service (SaaS), besturingsplatformen via platform as a service (PaaS), infrastructuren via infrastructure as a service (IaaS) en opslag via storage as a service (STaaS) te leveren.
Command-Line Interface (CLI)	De CLI is een programmeertaal specifieke interface die voor het aansturen van computerprogramma wordt gebruikt. Het aansturen van dit programma gebeurd op basis van een enkele programmeertaalregel per uitvoer.

Common Open Policy Service (COPS)	Het protocol COPS maakt onderdeel uit van de Internet Protocol Suite en is verantwoordelijk voor het creëren van een simpel client/server model. Aan de hand van dit model kunnen policy's voor QoS worden gedistribueerd en gebruikt.
Consoleconnectie	Een consoleconnectie is een fysieke connectiemethode om een computersysteem te besturen. Het besturen van dergelijk systeem in combinatie met deze connectiemethode gebeurd aan de hand van een Command-Line Interface.
Content-addressable memory (CAM)	Content-addressable memory is een type geheugen dat wordt gebruikt voor opzoekapplicatie die op zeer hoge snelheden functioneren. Dit type geheugen kan alleen statische gegevens aanvoeren naar de opzoekapplicaties.
Control Elements (CE's)	Een CE is een onderdeel van de ForCES netwerk architectuur en draagt zorg voor het aansturen van de Forward Elements.
Datacenter	Een datacenter is een faciliteit waar bedrijfskritische IT-apparaten zijn ondergebracht.
East-West API	Een East-West API is een interface die gebruik maakt van een uniforme programmeertaal om federaties tussen diverse SDN-controllers mogelijk te maken.
Federaties	Een federatie is een verband van samenwerkende apparaten die eveneens hun zelfstandigheid behouden. Binnen SDN-omgevingen wordt hiermee gedoeld op SDN-controllers die met elkaar samenwerken.
Field-programmable gate array (FPGA)	Een FPGA is een variant van een ASIC. Echter, deze variant is wel opnieuw te programmeren en aan te passen. Doordat deze chip wel de mogelijkheid biedt tot aanpassingen, is deze chip trager en duurder dan een ASIC. Daarnaast biedt deze chip de mogelijkheid om logische functies, zoals AND, XOR, et cetera te gebruiken.
Forwarding Elements (FE's)	Een FE is een onderdeel van de ForCES netwerk architectuur en draagt zorg voor het distribueren van datapakketten.
Forwarding Information Base (FIB)	De FIB wordt binnen zowel geswitchte- als gerouteerde netwerken gebruikt en is verantwoordelijk voor het bijhouden van een adres - interface tabel. Deze tabel kan zowel in laag 2 als 3 van OSI-model zijn uitgevoerd.
Framework	Een framework is een geheel aan componenten dat gezamenlijk één functionaliteit beid.
High-level design	Een high-level design verzorgt een weergave van een complete systeem zonder in detail te treden over specifieke detailelementen te benoemen.
HyperText Transport Protocol (HTTP)	Het protocol HTTP is een applicatie gebaseerd protocol voor het distribueren van hypermedia informatiesystemen. Deze informatiesystemen worden gebruikt voor het weergeven van webpagina's.

HyperText Transport Protocol Secure (HTTPS)	Het protocol HTTPS is de beveiligde variant van het protocol HTTP.
Hypervisor	Een hypervisor is een stuk software of hardware dat verantwoordelijk is voor het beheren van virtuele machines en resources.
Internet Protocol (IP)-adres	Een IP-adres is een uniek adres voor elk apparaat wat connectie maakt met het netwerk. Dit adres is uitgevoerd op laag 3 van het OSI-model en wordt statisch of dynamisch toegekend.
Internet Protocol Suite	Het internet protocol suite is een set van communicatieprotocollen die communicatie over het internet mogelijk maken. Deze set van communicatieprotocollen staat eveneens bekend als TCP/IP.
Load balancing	Load balancing is een techniek waar de werklast wordt verdeeld over meerdere gelijkwaardige apparaten.
Local Area Network (LAN)	Een LAN is een computernetwerk dat verantwoordelijk is voor de interconnectie van computersystemen.
Mainframe networking	Een mainframe netwerk is een vorm van computernetwerken waarbij de rekenkracht op een gecentraliseerde server (mainframe) plaatsvind.
Media Access Control (MAC)-adres	Het MAC-adres is een unieke adressering voor het hardware component van een computersysteem dat verbinding maakt met een computernetwerk. Deze adressering is uitgevoerd op laag 2 van het OSI-model.
Mulitlayer switch	Multilayer switch - switch capabel voor het uitvoeren van laag 2 en 3 functionaliteiten.
Multipathing	Multipathing is een netwerktechniek dat meer dan één pad gebruikt om connectie met de eindbestemming te maken.
Multitenancy	Een tenant is een gebruikersgroep die een gedeelde toegang heeft tot een softwarepakket. De rechten- en beveiligingsscheidingen vinden niet op het netwerk, maar binnen het softwarepakket plaats. Bij multitenancy worden meerdere tenants tot één softwarepakket toegelaten.
Network bridge	Een network bridge is een voorlopen van een switch. Het voornaamste verschil tussen een bridge en een switch bestaat uit het forwarden van datapakketten. Een bridge stuurt standaard de datapakketten over alle poorten naar buiten. Een switch doet dit alleen wanneer het bestemmingsadres niet in de CAM staat.
Northbound API's	Northbound API's zijn, binnen een SDN-omgeving, uniforme interfaces en dragen zorg voor de communicatie tussen applicaties en de SDN-controller.

Orkestratie	Orkestratie is een gecentraliseerde manier van aansturen. Binnen computernetwerken wordt de aansturing van de onderliggende netwerkapparatuur bedoeld.
OSI-model	Het OSI-model is een door International Organization for Standardization (ISO) gestandaardiseerd referentiemodel voor datacommunicatie. Dit model bevordert de interoperabiliteit tussen heterogene netwerktopologieën.
Peer-to-Peer	Een peer-to-peer netwerk is een logisch computernetwerk van gelijkwaardige computers. Binnen een dergelijk netwerk leveren de diverse computers verschillende services aan elkaar. Het welbekende torrentnetwerk is hier een voorbeeld van.
Point-to-Point	Een point-to-point netwerk is de minimale vorm van een computernetwerk. Bij dergelijk netwerk worden computersystemen rechtstreeks met elkaar verbonden. Een point-to-point netwerk bestaat veelal uit maximaal twee computersystemen.
Port-based Network Access Control (PNAC)	PNAC is een onderdeel van de IEEE 802.1X standaardisering. Deze standaardisering draagt zorg voor het authentiseren van netwerktoegang van identiteiten.
Programmeertalen	Een programmeertaal is een taal met vocabulaire syntaxis dat wordt gebruikt voor het samenstellen van een computerprogramma.
Quality of Service	Quality of Service is een protocol dat zorg draagt voor het prioriteren van netwerkverkeer.
RADIUS	RADIUS is een protocol dat aan de hand van het AAA-mechanisme netwerkconnectie voor gebruikers beheerd.
Remote terminals	Een remote terminal is een apparaat dat verbinding maakt met een gecentraliseerde server om vervolgens op deze server zijn rekenkracht af te handelen. Deze remote terminals zijn veelal terug te vinden binnen mainframe netwerken.
Routed network	Binnen een gerouteerd netwerk kunnen diverse netwerksegmenten (over laag 3 van het OSI-model) met elkaar communiceren.
Routing Information Base (RIB)	De RIB, eveneens routing tabel genoemd, draagt zorg voor de bewustheid van diverse routes, metrics, costs en netwerkbestemmingen. Met behulp van deze tabel kan een netwerkapparaat het verkeer over de juiste poort distribueren.
Scriptingtalen	Een script is een combinatie van meerdere losstaande commando's voor het (in één keer) uitvoeren van diverse opdrachten. Met de talen van een script wordt gedoeld op de taal die een applicatie ondersteund voor het uitvoeren van losstaande commando's.
Secure Socket Layer (SSL)	SSL is een crypto grafisch protocol dat zorg draagt voor de beveiliging van datacommunicatie op een computernetwerk.

Simple Network Management Protocol (SNMP)	SNMP is een protocol uit de Internet Protocol Suite en is verantwoordelijk voor het beheren van netwerkapparatuur.
Single point of failure	Single point of failure is een term die wordt gebruikt binnen computerarchitecturen. Met deze term wordt geduid op een, in het geval van een storing, complete uitval van de systeemfunctionaliteit.
Southbound API's	Southbound API's zijn, binnen een SDN-omgeving, uniforme interfaces en dragen zorg voor de communicatie tussen de diverse netwerkapparaten en de SDN-controller.
SSH-connectie	SSH is een crypto grafisch protocol uit de Internet Protocol Suite en is verantwoordelijk voor de beveiliging van een tekst-based shell sessie.
Switching-loops	Een Switching-loop is het oneindig forwarden van een, op laag 2 van het OSI-model uitgevoerd, datapakket. Dergelijk loop ontstaat doordat meerdere paden tussen twee eindebestemmingen gebruikt kunnen worden.
Telnetconnectie	Telnet is een protocol uit de Internet Protocol Suite dat verantwoordelijk is voor het initiëren van een text-based shell sessie. In tegenstelling tot de SSH-connectie, is de Telnetconnectie niet beveiligd.
Tenants	Een tenant is een gebruikersgroep die een gedeelde toegang heeft tot een softwarepakket. De rechten- en beveiligingsscheidingen vinden niet op het netwerk, maar binnen het softwarepakket plaats.
Tenary Content-addressable memory (TCAM)	Een TCAM is een meer geavanceerde versie van een CAM. Waar een CAM alleen waardes van een '1' of een '0' kan bevatten, kan een TCAM een 'don't care bit' toevoegen. Aan de hand van dit bit kunnen waardes worden overgeslagen en ontstaat een bredere output.
Virtual LAN (VLAN)	Een VLAN is een netwerksegmentatietechniek om meerdere kleine netwerken binnen één LAN te maken. Een VLAN is uitgevoerd op laag 2 van het OSI-model.
Virtualisatie	Virtualistatie is een techniek die het mogelijk maakt om hardware onderdelen te verplaatsen naar software. Ter adstructie kan worden gedacht aan het een computersysteem (combinatie van diverse hardware onderdelen). Met behulp van virtualisatie kan een dergelijk systeem, door een softwareapplicatie, worden nagebootst.
Wide Area Networks (WAN's)	Een WAN is een computernetwerk dat over een brede geografische gescheiden locatie. Deze netwerken worden veelal bewerkstelligd met behulp van gehuurde tele- of datacommunicatielijnen.
Wirespeed	Wirespeed is binnen computernetwerken een hypothetische term die verwijst naar de maximale snelheid van de bekabeling. Indien een netwerkapparaat datapakketten op wirespeed afhandelt, worden deze pakketten op de maximale snelheid van de bekabeling verwerkt.

Tabel 4: Begrippenlijst

Inhoudsopgave

Inleiding

Deze literatuurstudie wordt geschreven om een overzicht te creëren van de bevindingen die tijdens de onderzoeksfase worden opgedaan. Het wordt geschreven door de afstudeerder (dhr. François Kikken) en is in eerste instantie gericht aan zowel de bedrijfsbegeleider (ing. Marc Jonkers) als de schoolbegeleider (ing. Daniël Heynen). Daarnaast is dit document geschikt voor iedereen die meer duidelijkheid in dit onderwerp wenst te verkrijgen. De distributie van deze literatuurstudie vindt plaats op de woensdag van de 16e week samen met het gehele afstudeerdossier.

De opdracht omvat de SDN-techniek en is bedoeld om de meerwaarde van deze techniek, binnen het netwerk van Maastricht University (MAASnet), aan te tonen. Een meer gedetailleerde doelstelling is in het Project Initiatie Document (Kikken, 2015) terug te vinden.

Deze literatuurstudie is opgebouwd uit het hoofdstuk Software Defined Networking en draagt zorg voor een verduidelijking in deze techniek. Deze verduidelijking wordt bewerkstelligd door aan de hand van een opbouwende lijn de techniek te beschrijven. Zo wordt eerst de relatie met de huidige traditionele netwerken beschreven en vervolgens de stap gemaakt naar SDN. Verder is binnen dit hoofdstuk de historie-, het ontstaan van SDN, diverse modellen / uitvoeringen / protocollen en zowel de voor- als nadelen van een SDN-omgeving beschreven.

1 Software Defined Networking

De term Software Defined Networking is, op het moment van schrijven, een veel besproken onderwerp bij zowel IT-professionals als de netwerkapparatuur fabrikanten. Deze architecturale netwerkdesignmethode draagt, bij de diverse partijen, zorg voor een groot aantal vraagstukken en onduidelijkheden. Dit hoofdstuk creëert, aan de hand van de onderzoekbevindingen van de auteur, meer duidelijkheid in deze architecturale netwerkdesignmethode en de werking hiervan. Daarnaast komt een stuk uitleg over de werking van traditionele netwerken aan bod. De kennis hiervan is essentieel voor het begrijpen van de werking van SDN.

1.1 Traditionele netwerken

Basiskennis over het functioneren van traditionele netwerkapparatuur is vereist om een goed beeld van SDN en haar achterliggende techniek te creëren. Aan de hand van onderstaande sub paragrafen wordt de benodigde kennis aangetoond.

1.1.1 Historie

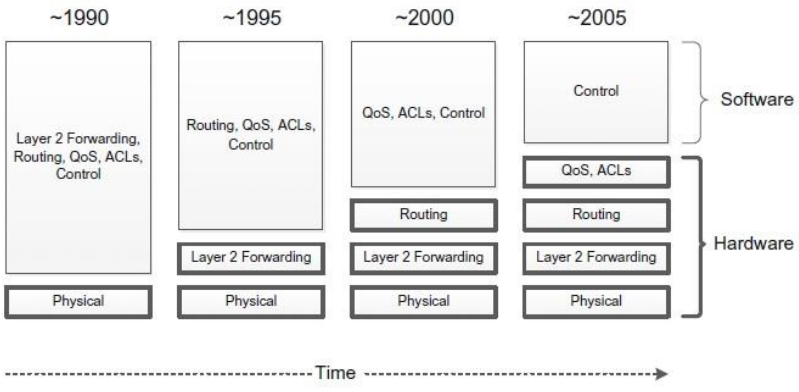
Een van de eerste vormen van netwerken was Mainframe networking. Deze vorm van netwerken bestond uit remote terminals die connectie maakte naar een mainframe en is het beste te vergelijken met SSH-connecties naar een Unix server. Op het mainframe werd een stuk schijfruimte en een proces-ID aan een gebruiker gekoppeld. Deze gebruiker kon vervolgens met dit proces-ID resources aan zichzelf toewijzen en gebruiken. Connectie tussen de diverse remote terminals gebeurde als het ware op software (kernel) niveau. Daarnaast waren softwareapplicaties veelal gepatenteerd aan de fabrikanten van het mainframe.

Een daaropvolgende netwerktechniek was Peer-to-Peer, Point-to-Point Connections. Deze techniek is ontstaan doordat de computers steeds meer gepersonaliseerd werden. Het personaliseren gebeurde door de rekenkracht naar, vanuit het mainframe, de computer zelf te verplaatsen. Bij het verplaatsen van de rekenkracht en processen ging de onderlinge connectie verloren. Om deze onderliggende connectie opnieuw te bewerkstelligen werden protocollen ontworpen voor het delen van data tussen twee peer-machines.

Ten slotte ontstonden de Local Area Networks (LAN)'s zoals we ze nu kennen. Deze LAN's waren in twee vormen terug te vinden, namelijk bridged- (laag 2 van het OSI-model) en routed (laag 3 van het OSI-model) networks. Beide vormen zijn gebaseerd op de TCP/IP-protocolstack.

1.1.2 Ontwikkeling

In de loop der jaren heeft diverse ontwikkeling, betreffende de laag 2 netwerkapparatuur, plaatsgevonden. Figuur 1 geeft een visuele weergave van deze ontwikkelingsperiode. De jaren omstreeks 1990 was de netwerkapparatuur alleen verantwoordelijk voor de 1e laag van het OSI-model. Alle resterende intelligentie werd destijds bewerkstelligd met behulp van een softwarepakket. Voor het routeren van pakketten[1] – dit gebeurd door het bestemmingsadres uit een IP-datapakket te vergelijk met de netwerkadressen van de aangesloten interfaces – werd bijvoorbeeld gebruik gemaakt van een Unix computer.



Figuur 1: Tijdlijn ontwikkeling laag-2-netwerkapparatuur (Göransson & Black, Networking functionality migrating to hardware, 2014)

[1] Het routeren van pakketten gebeurd op laag 3 van het OSI-model.

Het expediëren van laag 2 pakketten werd bewerkstellig met een netwerk bridge. Echter, de intelligentie van deze apparatuur was een softwarematige toevoeging aan het apparaat.
Omstreeks de jaren 1995 werd het, dankzij de ontwikkeling van application-specific integrated circuits (ASIC), mogelijk om de laag 2 pakketafhandeling te verplaatsen naar een hardware functionaliteit. De ASIC's werden geprogrammeerd om op hoge snelheden hash-functie uit te voeren. Deze hash-functies werden gebruikt voor het opzoeken van laag 2 adressen (MAC-adres) in de content-addressable memory (CAM)-tabel. Aan de hand van deze tabel worden Media Access Control (MAC)-adressen gelinkt aan een switchinterface en is de switch niet meer genoodzaakt om de pakketten over alle interfaces te distribueren. Door de ASIC's te combineren met de CAM-tabel ontstond een revolutionaire ontwikkeling op het gebied van laag 2 pakketafhandeling en de daarbij behorende snelheden.

Gedurende de 21e eeuw (2000 t/m het moment van schrijven) is de hardware van de laag 2 apparatuur aangevuld met laag 3 en 4 functionaliteiten. Hierdoor ontstond het concept multilayer switch (MLS). De hardware werd aangevuld met Field-programmable gate array (FPGA)- en Ternary content-addressable memory (TCAM) chips. Door deze aanvulling is het apparaat, naast het op hoge snelheden uitvoeren van hash-functies, in staat om logische eenvoudige wiskundige berekeningen uit te voeren. Deze berekeningen zijn een vereiste bij het routeren, filteren of prioriteren van datapakketten. Daarnaast vinden deze berekeningen eveneens op hoge snelheden (wirespeed) plaats.

Wat de laag 3 netwerkapparatuur (routers) betreft, vindt het routeren van datapakketten nog steeds plaats op softwareniveau. Weliswaar niet op aparte apparatuur of machines, zoals dit bij laag 2 apparatuur het geval was, maar op een software capabele microprocessor(en). Deze processoren zijn daarnaast tevens verantwoordelijk voor het afhandelen van overige laag 3 en 4 pakketten. (Göransson & Black, §1.4 - Traditional Switch Architecture, 2014)

1.1.3 Protocol 'soup'

De complexheid van de hedendaagse autonome netwerken is (mede) ontstaan door de grote hoeveelheden aan protocollen. Zo bestaan protocollen voor het traceren en up-to-date houden van het netwerk, het voorkomen van switching-loops in redundant omgevingen, het segmenteren van LAN's, het prioriteren van netwerkverkeer, et cetera. *"We have created a network management nightmare that is entirely avoidable"* (Göransson & Black, Software Defined Networks: A Comprehensive Approach, 2014). Deze protocollen zijn verantwoordelijk voor het goed en betrouwbaar laten functioneren van de diverse netwerken en zijn ontstaan om de tekortkomingen van TCP/IP op te vullen. Hieraan vasthangend wordt een bepaalde verwerkingscapaciteit van het controle draagvlak (zie paragraaf 1.1.4.2 voor meer informatie over het controle draagvlak) vereist. Zo gaf Gahsinky, I. op de Open Networking Summit aan dat het controle draagvlak in datacenters 30% van haar Central Processing Unit (CPU)-cyclussen gebruikt voor het traceren van de netwerktopologie (Gahsinsky, Oktober 2011). Deze verwerkingscapaciteit is, door de huidige netwerkarchitectuur, op elk netwerkverwerkingsapparaat (hetzij routers en switches) terug te vinden. Dit brengt tevens de nodige kosten met zich mee.

1.1.4 Management-, Control- en data(forwarding) plane

Beide netwerkapparaat categorieën (laag 2 en 3) zijn opgebouwd uit drie draagvlakken (planes). Ieder draagvlak vervult een eigen doelstelling m.b.t. de werking van het apparaat. De verschillende draagvlakken zijn: management-, controle- en data (forwarding) plane. De onderstaande paragrafen (1.1.4.1 t/m 1.1.4.3) beschrijven deze draagvlakken in meer detail. Daarnaast wordt met behulp van Figuur 2 een visueel beeld, gebaseerd op een multi-layer capabele switch, van de diverse draagvlakken en haar functionaliteiten gecreëerd.

1.1.4.1 Managementplane

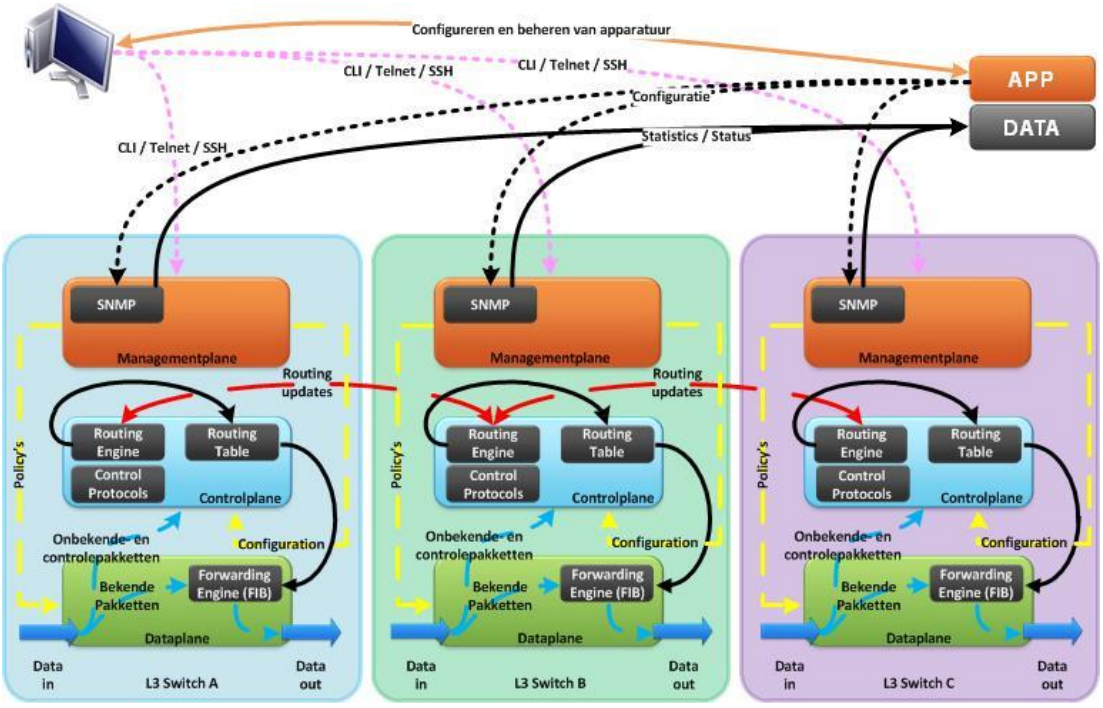
De managementplane draagt zorg voor het configureren en het distribueren van systeem- en netwerkinformatie van het aansluitend netwerkapparaat. Wanneer netwerkengineers een connectie met dergelijk apparaat maken, wordt in principe een connectie naar de managementplane gecreëerd. Deze connectie vindt plaats op basis van een console-, telnet-, of SSH-connectie en wordt veelal gebruikt voor het configureren van het netwerkapparaat. De configuratie gebeurt vervolgens met behulp van een translatie van het high-level design en zorgt voor uiteindelijke regels en policy's die door het apparaat worden begrepen. Deze regels en policy's bieden tevens de basis voor de functionaliteiten van de control- en dataplane. Daarnaast is dit draagvlak verantwoordelijk voor het inzamelen van systeem- en netwerkinformatie. Het verwerven van deze informatie gebeurt én wordt verstuurd op basis van het protocol Simple Network Management Protocol (SNMP). Figuur 2 geeft een duidelijkere weergave van dit draagvlak en haar functionaliteiten en doelstellingen.

1.1.4.2 Controlplane

De controlplane is verantwoordelijk voor het afhandelen van datapakketten die niet door de dataplane kunnen worden afgehandeld. Routering, filtering, priorisering en categorisering zijn hier o.a. een voorbeeld van. Echter, het hoofddoel van dit draagvlak is het up-to-date houden van haar Routing Information Base (RIB). De RIB, eveneens routing table genoemd, is gevuld met routinggegevens afkomstig van de diverse (geconfigureerde) routingprotocollen. De routinggegevens ontstaan, doordat elk routingprotocol (routingengine) zich – door een onderliggende connectie met haar 'buren' – van een deel of de gehele – afhankelijk van het gebruikte protocol – netwerktopologie bewust is. Deze gegevens worden vervolgens naar de Forwarding Information Base (FIB) in de dataplane gestuurd. Figuur 2 geeft een duidelijkere weergave van dit draagvlak en haar functionaliteiten en doelstellingen.

1.1.4.3 Data (forwarding) plane

De data plane, eveneens forwarding plane genoemd, is een draagvlak die zich voornamelijk bezig houdt met het expediëren van datapakketten. Alle datapakketten, waarvan een bestemming zich in de FIB bevindt, wordt direct door diverse ASIC's op dit draagvlak afgehandeld. Dit principe heeft binnen de diverse fabrikanten een andere naamgeving. Binnen de Cisco Systems termen heet dit principe Cisco Express Forwarding (CEF). Daarnaast vindt, doordat geen tussenkomst van andere draagvlakken benodigd is, de pakketafhandeling snelheid op wirespeed plaats.



Figuur 2: Draagvlakken netwerkapparatuur

1.2 Wat is SDN

Software Defined Networking is, op het moment van schrijven, een nieuw architecturaal netwerk design. Binnen dit concept wordt het controle[2]- van het data (forwarding) draagvlak gescheiden en op een centrale locatie binnen het netwerk geplaatst. Daarnaast worden forwarding besluiten gebaseerd op flows en policy's. Deze aanpak draagt zorg voor een gesimplificeerd, gecentraliseerd en programmeerbaar computernetwerk.

SDN heeft bij iedere belanghebbende partij een andere definitie. Doordat Open Network Foundation (ONF) helemaal is toegewijd aan SDN en één van de eerste SDN-communicatie standaarden (t.w. OpenFlow) heeft voorgesteld, wordt de definitie van ONF als leidraad in dit document gebruikt. (Open Networking Foundation , 2015) benoemd de definitie als volgt:

> *Software-Defined Networking (SDN) is an emerging architecture that is dynamic, manageable, cost-effective, and adaptable, making it ideal for the high-bandwidth, dynamic nature of today's applications. This architecture decouples the network control and forwarding functions enabling the network control to become directly programmable and the underlying infrastructure to be abstracted for applications and network services.* (p.1)

Andere SDN-definiëringen zijn hieronder terug te vinden.
Brocade definieert SDN als volgt: *"Software-Defined Networking is new approach to networking that unbundles the traditional device-bound, vertically integrated network stack to give you greater network automation, architectural flexibility, and programmability for policy-driven control and self-service innovation."* (Brocade, 2015)

Alcatel-Lucent definieert SDN als volgt: *"Software Defined Networks is the term used to refer to the many approaches proposed by standards bodies and application virtualization and network vendor communities to solve this problem."* (Alcatel-Lucent, 2015)

Oracle definieert SDN als volgt: *"The Oracle SDN (Software Defined Network) provides virtual network services, eliminating the need for purpose-built network appliances. It also simplifies data center infrastructure by dynamically connecting virtual machines (VMs) and servers to any resource in your data center fabric."* (Oracle, 2015)

SDxCentral definieert SDN als volgt: *"Software-defined networking is a new approach to designing, building, and managing networks that separates the network's control (brains) and forwarding (muscle) planes to better optimize each."* (SDxCentral, 2015)

IETF definieert SDN als volgt: *"SDN refers to the ability of software applications to program individual network devices dynamically and therefore control the behavior of the network as a whole."* (IETF, RFC 7426, 2015)

Juniper definieert SDN als volgt: *"A simple, open and agile software defined networking solution that automates and orchestrates the creation of highly scalable virtual networks."* (Juniper Networks, 2015)

Door de grote hoeveelheid verschillende definities en interpreteringen van SDN kan verwarring ontstaan of een product nu wel of geen SDN-product is. Deze verwarring wordt opgeheven door een zestal eigenschappen op te stellen. Om tot de categorie SDN-producten te behoren, dient een product aan deze eigenschappen te voldoen. De eigenschappen zijn: draagvlak splitsing, simplificeerde apparatuur, gecentraliseerd beheer, openheid en zowel netwerkautomatisering als netwerkvirtualisatie (Göransson & Black, Software Defined Networks: A Comprehensive Approach, 2014).

[2] Binnen het concept SDN wordt het controle- en management draagvlak als één draagvlak gezien.

1.2.1 Draagvlak scheiding, gecentraliseerd beheer en simplificeerde apparatuur

De eerste drie eigenschappen (te weten: Draagvlak scheiding, gecentraliseerd beheer en simplificeerde apparatuur) kunnen als één eigenschap worden gezien. Deze drie eigenschappen sluiten op elkaar aan en zijn logische vervolgstappen van elkaar. Als eerste wordt een scheiding tussen het controle- en data (forwarding) draagvlak toegepast. De doelstelling en functionaliteiten van de verschillende draagvlakken is in paragraaf 1.1.4 al naar voren gekomen. Door deze scheiding toe te passen is het mogelijk om de intelligentie van het netwerkapparaat (t.w. het controledraagvlak) te verplaatsen naar een gecentraliseerde locatie en ontstaat een gecentraliseerde locatie om het netwerk te beheren. Door deze verplaatsing hoeft het netwerkapparaat geen complexe berekeningen meer uit te voeren om een overzicht van het netwerk te creëren en ontstaat een gesimplificeerd netwerkapparaat.

1.2.2 Netwerk- automatisering en virtualisatie

Binnen de softwareprogrammeringswereld heeft Barbara Liskov de 2008 Turning Award gekregen voor het object-georiënteerd programmeringstaalontwerp. Zij zei: *"Modularity based on abstraction is the way things get done"* (Shenker, 2011) over Computer Wetenschappen. Volgens Shenker is abstractie binnen een SDN-netwerk eveneens van groot belang. Hij benoemde de abstracties van een gedistribueerde staat, van forwarding en van configuratie als de drie vereisten van een SDN-netwerk. Aan de hand van deze abstracties wordt een globaal netwerkoverzicht gecreëerd, waarbij geen kennis van de onderliggende netwerkapparatuur nog de diverse configuraties benodigd is. De abstractie ontstaan door het toepassen van een SDN-controller met de daarbij behorende Application Programming Interface (API)'s (zie paragraaf 1.5. voor meer informatie over de diverse API's). Deze API's dragen zorg voor één of meerdere gestandaardiseerde communicatieprotocollen en programmeertalen waardoor automatisering en virtualisatie mogelijk wordt.

1.2.3 Openheid

De laatste eigenschap van een SDN-netwerk is de openheid van de API's en broncodes van de diverse controllers. Door deze openheid krijgen onderzoekers en engineers de mogelijkheid om te innoveren naar nieuwe methodieken op het gebied van netwerk- en beheerstechnieken. Deze API's, broncodes en innovaties dienen echter wel goed gedocumenteerd te zijn. (Göransson & Black, Software Defined Networks: A Comprehensive Approach, 2014)

1.3 Historie

De aanloop naar het concept SDN is ontstaan aan de hand van een aantal voorafgaande projecten en standaardiseringsvoorstellen. Deze projecten en standaardiseringsvoorstellen worden in de onderstaande paragrafen 1.3.1 t/m 1.3.7 meer in detail beschreven.

1.3.1 Open Signaling

Een van de eerste denkwijzen richting SDN was het voorstel van The Open Signaling (OPENSIG) working group. Deze organisatie begon in 1995 met diverse conferenties toegewijd aan het ultieme doel: *"The ultimate goal of introducing open programmable technology into the network is to make the network as programmable as the PC thereby enabling new network services to be launched with the ultimate ease of clicking and downloading an application."* (Campbell, Katzela, Miki, & Vicente, 1998). Echter, om deze doelstelling te behalen was een scheiding van de communicatie hardware (data plane) van de controle software (controle plane) noodzakelijk. Deze scheiding was destijds, door de verticale integratie van fabrikanten, vrijwel onmogelijk te realiseren en ontstond een meer bereikbare (sub)doelstelling, namelijk: om ATM-, Internet- en mobiele netwerken meer open, uit te breiden en programmeerbaar te maken. Aan de hand van deze denkwijze creëerde een The Internet Engineering Task Force (IETF) werkgroep het protocol General Switch Management Protocol (GSMP). Dit protocol heeft het doel om ATM-switches te beheren (IETF, RFC 1987, 1996).

1.3.2 Network Access Control

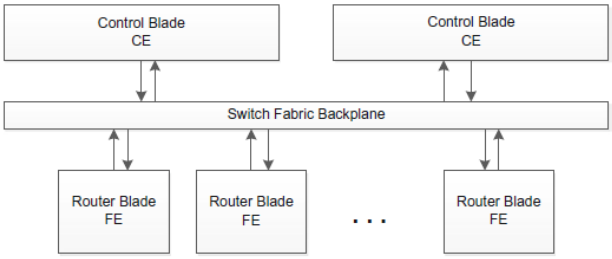
Port-based Network Access Control (PNAC) is een protocol uit de IEEE 802.1X standaard. Dit protocol is o.a. verantwoordelijk voor het verlenen van netwerktoegang (IEEE Std 802.1X-2010, 2009). De toegangscontrole gebeurd met behulp van een RADIUS- of een Common Open Policy Service (COPS)-server en aan de hand van, door de netwerkbeheerder(s) opgestelde, toegangspolicy's. Doordat deze servers geautomatiseerd aanpassingen (afhankelijk van de ingestelde policy) aan het netwerk kunnen maken, worden deze als voorganger van SDN gezien. Echter, de oorspronkelijke bedoeling van dit protocol is het mogelijk maken van het authenticatie-, autorisatie- en accounting- (AAA) proces.

1.3.3 Orkestratie

Een van de eerste pogingen om orkestratie[3] geautomatiseerd toe te passen, werd bewerkstelligd met behulp van orkestratieprogramma's die gebruik maakte van API's. Deze API's bestaan veelal uit een Command-Line Interface (CLI) of SNMP-verkeer. Daarnaast beschikt het programma over diverse plug-ins om communicatie met verschillende fabrikant specifieke API's mogelijk te maken. Doordat het uitvoeren van complexe talen niet tot nauwelijks geprogrammeerd konden worden, bleven zulke applicaties gelimiteerd tot het uitvoeren van simpele taken.

1.3.4 Forwarding and Control Elements Seperation (ForCES)

Omstreeks het jaar 2003 werd door een IETF-werkgroep een standaardiseringsvoorstel gedaan voor het protocol Forwarding and Control Element Separation (ForCES). Het doel van dit protocol is om communicatie tussen Control Elements (CE's) en Forwarding Elements (FE's) mogelijk te maken (IETF, RFC 5810, 2010) en is van toepassing op een ForCES-framework (IETF, RFC 3746, 2004). Dit architecturaal framework is bedoeld om simpele hardware gebaseerde fowarding entiteiten, op de fundering van het netwerk, te creëren. Deze entiteiten worden vervolgens gebruikt voor het afhandelen van het simpel netwerkverkeer, zoals cell- en tag-switching. Daarnaast worden de entiteiten aangevuld met software-beheerde elementen voor het verwerken van het meer complexe netwerkverkeer, zoals het routeren van netwerken. Figuur 3 geeft een duidelijker beeld van dit architecturaal framework. Dit framework bleef echter bij een standaardiseringsvoorstel (Göransson & Black, Software Defined Networks: A Comprehensive Approach, 2014).



Figuur 3: ForCES design (Göransson & Black, ForCES Design, 2014)

1.3.5 A Clean Slate: 4D Approach to Network Control and Management

Het voorstel van 4D, genaamd A Clean Slate: 4D Approach to Network Control and Management, kan worden gezien als een SDN-voorganger met een compleet nieuw architecturaal design. *"We advocate a complete refactoring of the functionality and propose three key principles—network-level objectives, network-wide views, and direct control—that we believe should underlie a new architecture."* (Greenberg, et al., 2005). Dit architecturaal design bestaat uit een viertal lagen, namelijk decision-, dissemination-, discovery- en een data laag. De achterliggende gedachte voor dit geheel nieuwe design was ontstaan, doordat het volgende door (Greenberg, et al., 2005) werd beweerd:

> *Today's data networks are surprisingly fragile and difficult to manage. We argue that the root of these problems lies in the complexity of the control and management planes—the software and protocols coordinating network elements—and particularly the way the decision logic and the distributed-systems issues are inexorably intertwined.* (p. 1)

Deze bewering zorgde voor een gedachte gang om de huidige autonome systemen links te laten liggen en richting een gecentraliseerd beheer- en managementsysteem te werken. De gedachte voor een gecentraliseerde design diende zorg te dragen voor een gesimplificeerde netwerkomgeving en werd bewerkstelligd door geen pakket afhandelingsbesluiten op de netwerk elementen uit te voeren. Echter, het voorstel van 4D – eveneens zoals dat van ForCES – bleef absent van daadwerkelijke realisatie (Greenberg, et al., 2005).

[3] Orkestratie, binnen de IT, is bedoel voor het (geautomatiseerd) gedistribueerd aansturen van diverse netwerkapparaten.

1.3.6 Ethane

Waar de voorstellen van ForCES en 4D een standaardiseringsvoorstel bleven, ging het Ethane-project nog een stap verder. Ethane is een netwerk architecturaal design, waarbij een gecentraliseerde controller het netwerk beheert. Dit gebeurd aan de hand van vooraf ingestelde policy's. Het project Ethane is voorgesteld en gedemonstreerd in de jaren omstreeks 2007 - 2009 en is ontstaan uit de vraag *"How could we change the enterprise network architecture to make it more manageable?"* (Casado, et al., Rethinking Enterprise Network Control, 2009). Daarnaast diende dit concept, eveneens als de andere, te zorgen voor een gesimplificeerd netwerk.

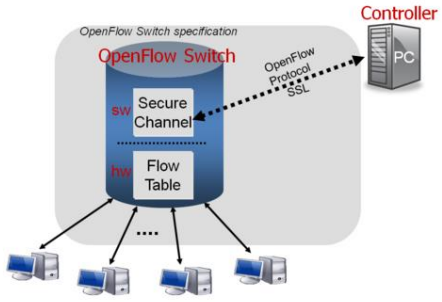
Ethane bestaat uit drie fundamentele principes die, aldus de diverse participanten, belangrijk zijn voor een netwerkmanagementoplossing. Het eerste principe luidt als volgt: *"The network should be governed by policies declared over high-level names."* (Casado, et al., Rethinking Enterprise Network Control, 2009) en is bedoelt om een makkelijker en meer beheersbaar netwerk te creëren. Het netwerk is op basis van entiteiten die toegang vragen – zoals gebruikers, computers en connectiepunten – , in plaats van low-level- en veelal dynamische adressen, beter en makkelijker te managen. Het tweede principe luidt: *"Network routing should be policy-aware"* (Casado, et al., Rethinking Enterprise Network Control, 2009) en is verantwoordelijk voor het routeren tussen diverse entiteiten. Dit principe is niet veel afwijkend van pakketfiltering binnen de huidige traditionele netwerken. Ten slotte luidt het derde principe als volgt: *"The network should enforce a strong binding between a packet and its origin."* (Casado, et al., Rethinking Enterprise Network Control, 2009). Dit principe is verantwoordelijk voor een nauw verband tussen de oorsprong en de bestemming van een datapakket. Daarnaast wordt het versterkt door de combinatie met het eerste principe. De high-level namen maken het namelijk mogelijk om datapakketten beter en makkelijker te identificeren.

Naast de drie fundamentele principes heeft Ethane nog een andere, van traditionele netwerken, afwijkende eigenschap. Traditionele netwerken hebben een open mentaliteit en wil zeggen dat alle connectiemogelijkheden, binnen een niet geconfigureerd netwerk, worden toegestaan. Een Ethane netwerk daarin tegen heeft een gesloten mentaliteit en staat geen verkeer toe in een niet geconfigureerde staat. Hierdoor is alleen dataverkeer mogelijk, wanneer dit tevens in de policy is opgenomen. (Casado, et al., Ethane: Taking Control of the Enterprise, 2007)

1.3.7 OpenFlow

Omstreeks het jaar 2008 is het voorstel voor OpenFlow ontstaan en is het antwoord op de onderzoeksvraag: *"As researchers, how can we run experiments in our campus networks?"* (McKeown, et al., 2008). Het concept achter OpenFlow (netwerkarchitectuur simplificeren door het netwerk programmeerbaar te maken) is gelijk aan eerder genoemde oplossingen, zoals 4D en Ethane. Echter, voor Ethane is speciale netwerkapparatuur vereist en is daardoor niet capabel om op korte termijn binnen een grote hoeveelheid instellingen toe te passen. Hierdoor is het idee ontstaan om een set van functies te ontwerpen die de flowtabellen van de diverse vendoren (veelal terug te vinden in het TCAM-geheugen) kunnen exploiteren. Dit idee leidde vervolgens tot het protocol OpenFlow.

OpenFlow verzorgt een open standaard protocol dat de mogelijkheid geeft om flowtabellen binnen de diverse, eventueel van verschillende fabrikanten, netwerkapparatuur (t.w. routers en switches) te programmeren. De flowtabellen worden, door de netwerkadministrator (statisch) of door de applicaties aangesloten op de northbound API's (dynamisch en meer gedetailleerd besproken in paragraaf 1.5), op een gecentraliseerde controller geprogrammeerd en vervolgens via het protocol OpenFlow naar de diverse netwerkapparatuur verstuurd. Figuur 4 geeft hier een weergave van. Hierdoor kan een netwerkadministrator participeren in de datapakkettenflow en is het netwerk programmeerbaar. Het protocol OpenFlow (t.w. de communicatie tussen de controller en de diverse netwerkapparatuur) vindt plaats op basis van het protocol Transmission Control Protocol (TCP) uit de Internet Protocol Suite. Daarnaast is het raadzaam om OpenFlow aan te vullen met het encryptieprotocol Secure Socket Layer (SSL).



Figuur 4: OpenFlow (McKeown, et al., 2008)

De dataflowtabellen binnen OpenFlow zijn, in tegenstelling tot traditionele netwerken, de breedste zin van het woord toe te passen en worden alleen gelimiteerd door de implementatiemanier. Zo kan een flow gebaseerd zijn op laag 1 t/m 4 van het OSI-model (binnen traditionele netwerken is dit laag 2 óf 3). Hierdoor wordt het mogelijk om datapakketten op basis van bijvoorbeeld inkomende poort, MAC-adres, IP-adres, Virtual LAN (VLAN)-ID, et cetera te identificeren (match-field). Figuur 5 geeft een voorbeeld van een OpenFlow datapakketheader. Na het identificeren van deze datapakketten wordt, afhankelijk van de in de controller gemaakte instellingen, een actie uitgevoerd. De onderstaande opsomming geeft een overzicht van de verschillende mogelijke acties[4].

In	VLAN	Ethernet			IP			TCP	
Port	ID	SA	DA	Type	SA	DA	Proto	Src	Dst

Figuur 5: OpenFlow header (McKeown, et al., 2008)

- **Output** (port_no)*
- **Group** (group id)*
- **Drop***
- **Set-Queue** (queue id)
- **Meter** (meter id)
- **Push-Tag/Pop-Tag** (ethertype)
- **Set-Field** (field type value)
- **Copy-Field** (src field type dst field type)
- **Change-TTL** (ttl) (Open Networking Foundation, 2014)

OpenFlow kan op twee verschillende manier worden toegepast binnen een netwerk, namelijk met behulp van dedicated- en enabled OpenFlow netwerkapparatuur. Dedicated OpenFlow apparatuur is een 'dom' netwerkapparaat dat gelimiteerd is tot alleen het expediëren van datapakketten. Dit apparaat is dus geheel afhankelijk van een OpenFlow ondersteunende controller. Enabled OpenFlow netwerkapparatuur daarin tegen, is netwerkapparatuur van diverse fabrikanten waar de OpenFlow functie op geactiveerd wordt. Deze apparatuur is gelijk aan de traditionele netwerkapparatuur en wordt door een firmware update aangevuld met de OpenFlow functie.

Omstreeks het jaar 2011 is de organisatie ONF opgericht. ONF is een non-profit en gebruikers-gedreven organisatie toegewijd aan de versnelling om open Software Defined Netwerken te erkennen en te adopteren. *"The Open Networking Foundation (ONF) is a non-profit, user-driven organization dedicated to accelerating the adoption of open Software-Defined Networking (SDN)."* (Open Networking Foundation, 2013) Deze organisatie heeft tevens, in ditzelfde jaar, ervoor gezorgd dat het protocol OpenFlow zijn bekendheid kreeg bij de commerciële sector. Op het moment van schrijven bevindt het protocol zich op versie 1.3. (McKeown, et al., 2008)

1.4 Ontstaan

In de voorgaande paragrafen is de historie en ontwikkeling van de SDN-voorlopers besproken. Binnen deze paragrafen is grotendeels duidelijk geworden waardoor het concept is ontstaan. Echter, de waaromvraag "Waardoor de vraag voor een SDN-omgeving is ontstaan" is hiermee niet concreet beantwoord. In deze paragrafen is voornamelijk sprake over gecentraliseerd en simplificeert netwerk en de mogelijkheid tot innovatie. Dit zijn echter niet de enige onderwerpen waarom de vraag naar een andere netwerkarchitectuur is ontstaan. De explosieve groei van data, datacenters- en cloudoplossingen zijn hier mede een oorzaak van. Deze oplossingen vragen voor een geautomatiseerd-, gevirtualiseerd- en schaalbaar netwerk. Daarnaast dienen deze functionaliteiten te bieden voor multipathing en multitenancy.

1.4.1 Automatisering

Met een geautomatiseerd netwerk wordt bedoeld dat het netwerk, naar aanleiding van de vraag, in staat is om virtuele netwerken aan te maken, te manipuleren en te verwijderen. Zo is bijvoorbeeld een laag 2 tunnel, indien deze gebruik maakt van de VMware VMotion techniek, bij het verplaatsen van een VM gewenst. Met een geautomatiseerd netwerk vervalt de noodzaak voor menselijke interactie of het ontwerpen van een apart fail-over netwerk. De vraag voor een geautomatiseerd netwerk ligt hierdoor in nauw verband met netwerk virtualisatie.

[4] Acties met een asteriskmarkering (*) worden minimaal als verplichte actie geacht.

1.4.2 Netwerk virtualisatie

De groei van server- en opslagvirtualisatie heeft zorg gedragen voor de enorme vraag naar schaalbaarheid, automatisering, multitenancy en multipathing. Daarmee is tevens het idee ontstaan om het netwerk te gaan virtualiseren. De achterliggende techniek wordt op dezelfde manier bewerkstelligd als dat dit bij server- en opslagvirtualisatie het geval is. Een high-level abstractie wordt bovenop de fysieke netwerkapparatuur gecreëerd en draagt zorg voor een 'nieuw' virtueel netwerk. Echter, door deze techniek toe te passen ontstaat een complexer en lastiger te beheersen netwerk.

1.4.3 Schaalbaarheid

Door de explosieve groei van virtuele omgevingen – Gartner voorspelt dat in het jaar 2016 80% van de servers is gevirtualiseerd (Dawson & Wolf, 2011) – dient een netwerk hierop mee te kunnen schalen. Een virtuele omgeving heeft immers tevens een netwerkconnectie nodig. Daarnaast speelt het gemak van het creëren van een virtuele omgeving tevens een rol. Deze is binnen enkele seconden tot minuten gemaakt. Dit is bij een traditionele netwerkomgeving niet het geval. Kortom de huidige netwerken kunnen deze schaalbaarheid niet aan.

1.4.4 Multipathing

Multipathing is een techniek waar datapakketten over meerdere verschillende paden naar de bestemming reist. Deze techniek is echter niet nieuw, maar wel lastig op globale schaal toe te passen binnen de huidige traditionele netwerken. Binnen deze netwerken werkt de techniek niet over meerdere Wide Area Networks (WAN's) en dienen de diverse paden over een gelijke lengte te beschikken. Daarnaast werden laag 2 frames veelal aangepast om deze techniek te bewerkstelligen. Door de statische aard van deze techniek wordt niet voldaan aan de dynamische omgevingen van een datacenter en cloudoplossing. (Bredel, 2013)

1.4.5 Multitenancy

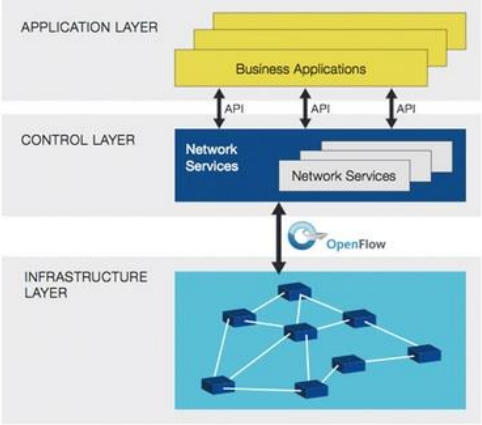
Multitenancy is een techniek afkomstig uit de software architectuur. Binnen deze techniek wordt één enkele softwareapplicatie (functionerend op één of meerdere servers) gedeeld door meerdere tenants[5]. Daarbij impliceert multitenancy dat iedere tentant over een eigen VLAN dient te beschikken. Deze eis, gecombineerd met de explosieve groei van datacenters en cloudoplossingen, verlangt een betere schaalbaarheid binnen de netwerk scheidingstechnieken. De huidige traditionele netwerken faciliteren een maximum van 4096 VLAN's. (Göransson & Black, Software Defined Networks: A Comprehensive Approach, 2014)

[5] Een tenant is een gebruikersgroep die een gedeelde toegang heeft tot een softwarepakket. De rechten- en beveiligingsscheidingen vinden niet op het netwerk, maar binnen het softwarepakket plaats.

1.5 SDN-framework

Hoewel meerdere variaties op het SDN-concept bestaan, is de basis van het framework veelal hetzelfde. Het SDN-framework bestaat uit een 4-tal basiselementen, namelijk een SDN-controller, Southbound API's, Northbound API's en een East-West API. Figuur 6 geeft een weergave van een dergelijk SDN-framework.

De SDN-controller, eveneens het brein van het netwerk genoemd, is een logisch gecentraliseerde netwerkentiteit die de verantwoording heeft over (1) het vertalen van SDN-applicatie eisen naar de onderliggende infrastructuur, (2) SDN-applicaties voorzien van een abstract netwerkoverzicht, (3) SDN-applicaties voorzien van eventmeldingen en statistieken van het netwerk, (4) netwerkadministrators voorzien van de mogelijkheid om te kunnen participeren in de flow van datapakketten en (5) het programmeren van forwarding entries. Daarnaast zijn de diverse API's tevens terug te vinden op de SDN-controller.

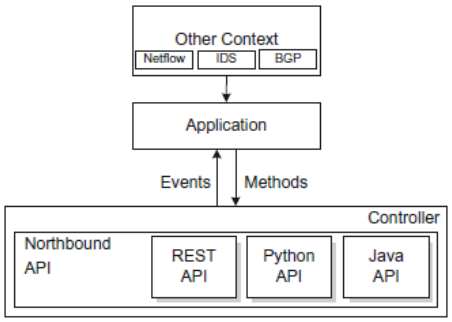


Figuur 6: SDN-Framework (Open Networking Foundation , 2015)

De Southbound API's zijn verantwoordelijk voor de communicatie tussen de SDN-controller en de diverse netwerkapparatuur. Binnen figuur 6 wordt gebruik gemaakt van het protocol OpenFlow om deze communicatie te bewerkstelligen. Echter, om deze communicatie te bewerkstelligen kunnen meerdere protocollen worden gebruikt. Paragraaf 1.8 geeft meer duidelijkheid in de diverse protocollen. Daarnaast kunnen de huidige CLI's en het protocol SNMP eveneens voorzien worden van de Southbound API naamgeving.

De East-West API is verantwoordelijk voor communicatie tussen groepen of federaties van SDN-controllers. Synchronisatie tussen de SDN-controllers wordt aan de hand van deze API bewerkstelligd en wordt het single point of failure principe opgeheven. Daarnaast ontstaat door deze API de mogelijkheid om meerdere en diverse SDN-controllers, binnen hetzelfde netwerk, te implementeren.

De Northbound API's zijn verantwoordelijk voor de communicatie tussen de diverse SDN-applicaties en de SDN-controller. SDN-applicaties kunnen bestaan uit netwerkservices (t.w. switching- en routingprotocollen, firewall-, loadbalancing oplossingen, et cetera) en applicaties (t.w. applicaties voor automatisering, orkestratie, monitoring, hypervisor, et cetera) die het netwerkverkeer manipuleren of als informatiebron gebruiken. De communicatie tussen de controller en SDN-applicaties bestaat uit events (netwerkinformatie) en methodes (flowmodificaties).
Met behulp van figuur 7 wordt hier een verduidelijking in gecreëerd. Daarnaast kunnen de API's in een low- of high-level worden uitgevoerd. Bij een low-level API is de SDN-applicatie bewust van de diverse onderliggende individuele netwerkapparatuur. Echter, de verschillen tussen de diverse apparaten zijn afgeschermd. Bij een high-level API wordt het bewustzijn van de diverse onderliggende individuele netwerkapparaten extra afgeschermd. Hierdoor ontstaat alleen een globaal beeld van het netwerk. In tegenstelling tot de Southbound API's zijn, op het moment van schrijven, nog geen standaarden voor de Northbound API's ontwikkelt. Hierdoor worden de meest gangbare scripting- en programmeertalen door deze API ondersteund. ONF heeft tegen het einde van 2013 een werkgroep opgericht voor het standaardiseren van een Northbound API protocol. Echter, tot op heden is daar nog geen standaardiseringsverzoek uit ontstaan. (SDxCentral, 2015) (Göransson & Black, Software Defined Networks: A Comprehensive Approach, 2014)



Figuur 7: Northbound API (Göransson & Black, Software Defined Networks: A Comprehensive Approach, 2014)

Naast de meest gangbare scripting- en programmeertalen kan Representational state transfer (REST) eveneens als Northbound API worden gebruikt. REST is een architecturale methode om communicatie van- en naar een grote diversiteit aan softwarefunctionaliteiten te bewerkstellen. REST is opgebouwd uit een zestal architecturale randvoorwaarden en levert – wanneer aan deze randvoorwaarden is voldaan – meer schaalbaarheid, onafhankelijkheid en comptabiliteit naar de API's. De onderstaande opsomming geeft een overzicht van de zes verschillende randvoorwaarden en wat hiermee wordt bedoeld.

1. **Client-Server**[6] - Door de userinterface- van de data storage zorgen te scheiden, worden de userinterfaces platformonafhankelijk en ontstaat meer schaalbaarheid. Binnen SDN betekent dit dat zowel verschillende applicaties als programmeertalen van dezelfde functionaliteiten, binnen de REST API, kunnen genieten.
2. **Stateless** - Een stateless client-server connectie draagt ervoor zorg dat, wanneer een client een request aan de server doet, het verzoek alle informatie bevat om de opdracht te begrijpen. Door gebruik te maken van een stateless connectie ontstaat een efficiënter gebruik van de API en SDN-controller.
3. **Caching** - De prestatie en schaalbaarheid van de API wordt vergroot door de veel gebruikte informatie in een lokaal kopie op de client op te slaan. Door aan de server kant te definiëren of een lokaal kopie is toegestaan, wordt een extra beveiliging toegevoegd.
4. **Layered System** - Door een gelaagd systeem toe te passen is de client in staat te communiceren met alle, op die laag aangelegen, servers en niet verder. Voor SDN betekent dit dat de applicatie alleen met de controller en niet met de onderliggende architectuur kan communiceren.
5. **Uniform Interface** - Het toepassen van een uniforme interface draagt zorg dat alle informatie, ongeacht afkomst of gebruikte taal, op dezelfde manier wordt weergegeven. Wat de REST API (binnen een SDN omgeving) betreft, vindt deze weergave veelal plaats op basis van de programmeertaal XML of JSON.
6. **Code-on-Demand** - Via code-on-demand ontstaat de mogelijkheid om, bij een request, een functionaliteit volledig op de server uit te voeren. Deze voorwaarde is optioneel en wordt gebruikt als de API's niet naar behoren (en wel juist geconfigureerd zijn) functioneren.

De meest gebruikte protocollen binnen deze REST API is het protocol HyperText Transport Protocol (HTTP) en de beveiligde variant HyperText Transport Protocol Secure (HTTPS). Deze protocollen maken gebruik van gestandaardiseerde methodes om informatie weer te geven, te manipuleren en te verwijderen. De methodes zijn zowel 'GET', 'HEAD', 'POST', 'PUT', 'DELETE', 'TRACE' als 'CONNECT' en staan beschreven in IETF RFC 2616 (IETF, RFC 2616, 1999). Met behulp van deze uniforme methodes kunnen applicaties eenvoudig zogenaamde CRUD-acties, tegen de SDN-controller aan, uitvoeren. CRUD is een acroniem voor 'Create', 'Read', 'Update' en 'Delete' (James, Create, read, update and delete, 1983) (Fielding R. , 2000).

[6] De servers (binnen de opsomming) staan in verband met de SDN-applicaties en de clients in verband met de SDN-controller.

1.6 SDN-Modellen

Binnen een SDN-omgeving kan, op het moment van schrijven, het controledraagvlakmechanisme op twee manieren worden uitgevoerd. Deze manieren betreffen zowel een imperatief- als een declaratief model en beschrijven hoe de onderliggende netwerkapparatuur datapakket afhandelingen uitvoert.

Binnen het imperatieve model functioneert de SDN-controller als het centrale zenuwcenter van een SDN-omgeving. Zoals in paragraaf 1.5 beschreven staat, ontvangt de controller – vanuit de Northbound API's – applicatieverzoeken en vertaald deze naar flows. Deze flows worden vervolgens via de Southbound API's naar het data (forwarding) draagvlak van de onderliggende netwerkapparatuur verzonden. Daarnaast benaderd de onderliggende netwerkapparatuur, wanneer datapakket afhandelingsopties onbekend zijn, de SDN-controller voor verdere afhandelingsinstructies. Door deze werkwijze te hanteren, blijft alle intelligentie binnen de SDN-controller(s) (SDxCentral, 2015).

Het flowververkeer – binnen het imperatieve SDN-model – kan gereguleerd worden aan de hand van drie verschillende methodieken, namelijk proactief-statisch, proactief-dynamisch en reactief. Bij een proactief-statische methode is de netwerkbeheerder geheel verantwoordelijk voor het netwerkverkeer. Om dit verkeer te bewerkstellen, dient de netwerkbeheerder alle mogelijke datapakketflows handmatig in de SDN-controller te programmeren. Gebruik van wildcards is mogelijk, maar verlaagt de granulariteit van de forwardingtabel. Deze methodiek is vergelijkbaar met het gebruik van statische routes binnen een traditioneel netwerk. Door de volledig statisch aard is deze methodiek de meest veilige, maar tevens de meest tijdrovende oplossing. De proactief-dynamische methode is een dynamische oplossing. Bij deze methode is de SDN-controller bewust van de individuele aangesloten eindpuntapparaten. De bewustheid ontstaat, hetzelfde als bij traditionele netwerken, doordat het aangesloten eindpunt apparaat – bij de eerste connectie met het netwerk – een ARP-request verzendt. De SDN-controller gebruikt deze gegevens om vervolgens de netwerktopologiedatabase te vullen en de kortste paden bij deze locatie te berekenen. De flowtabellen worden na de berekening, bij beide proactieve methodes, naar het geheugen van de onderliggende netwerkapparaten gestuurd. Hierdoor hoeft niet elk datapakket naar de controller te worden gestuurd en ontstaat een efficiënter gebruik van zowel de SDN-controller als de onderliggende netwerkarchitectuur. De reactieve methode, daarin tegen, dient wel alle datapakketen via de SDN-controller naar de eindbestemming te versturen. Hierdoor worden geen flowtabelgegevens in het cachegeheugen van de onderliggende netwerkapparatuur opgeslagen. Naast het feit dat – bij deze methode – zowel de SDN-controller als de onderliggende netwerk apparatuur minder efficiënt wordt benut, is het netwerk tevens onbruikbaar bij een controller failure (Klebanov, 2013).

Het declaratieve model draagt zorg voor een SDN-omgeving met meer gedistribueerde intelligentie. Dit model wordt eveneens het 'wat / hoe' model genoemd. Binnen dit model vertaalt de SDN-controller applicatieverzoeken naar policy's (wat). Deze worden onderverdeeld in deel policy's en vervolgens naar de controledraagvlakken van de netwerkapparatuur verstuurd. Door deze werkwijze te hanteren, verneemt de onderliggende netwerkapparatuur wat wel en niet is toegestaan. De methodes om deze regels te hanteren, daarin tegen, worden door de netwerkapparatuur zelf bepaald (hoe). (SDxCentral, 2015)

1.7 SDN-Uitvoeringen

Naast de onderscheiding tussen de twee SDN-modellen kan een SDN-omgeving in een viertal uitvoeringen worden gecategoriseerd. Deze categorieën zijn zowel Classic SDN, Hybrid SDN, Virtuele Overlays als Policy gericht en geven weer op welke manier een SDN-omgeving wordt ingevoerd. Figuur 9 geeft een visueel beeld in de diverse categorieën. Binnen deze afbeelding worden de Classic- en Hybrid SDN samengevoegd tot één categorie.

1.7.1 Classic SDN

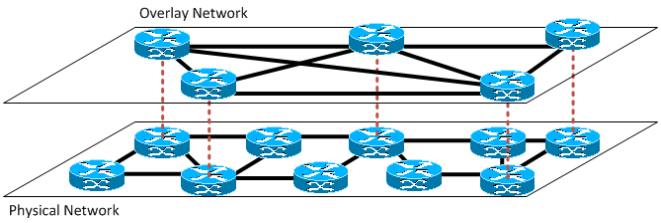
Classic SDN is een implementatiemanier waarbij een duidelijke scheiding tussen het controle- en het data (forwarding) draagvlak is te herkennen. De SDN-controller is volledig verantwoordelijk voor het beheren van de onderliggende netwerkapparatuur en de daarbij behorende forwarding tabellen. Door de gecentraliseerde geaardheid en de volledige controllerverantwoording valt deze uitvoering onder het imperatieve SDN-model. Deze SDN-uitvoering is de uitvoering die ONF oorspronkelijk voor SDN heeft bedoeld.

1.7.2 Hybird SDN

De Hybrid SDN implementatiemanier maakt gebruik van een gecombineerde netwerkomgeving architectuur. Een deel van de netwerkomgeving wordt bewerkstelligd aan de hand van de Classic SDN-uitvoering en is afhankelijk van een gecentraliseerde SDN-controller. Het andere deel van de netwerkomgeving is opgebouwd uit de huidige traditionele netwerk architectuur en is volledig gedecentraliseerd. Door deze combinatie te maken kan een SDN-omgeving gefaseerd worden ingevoerd.

1.7.3 Virtuele Overlays

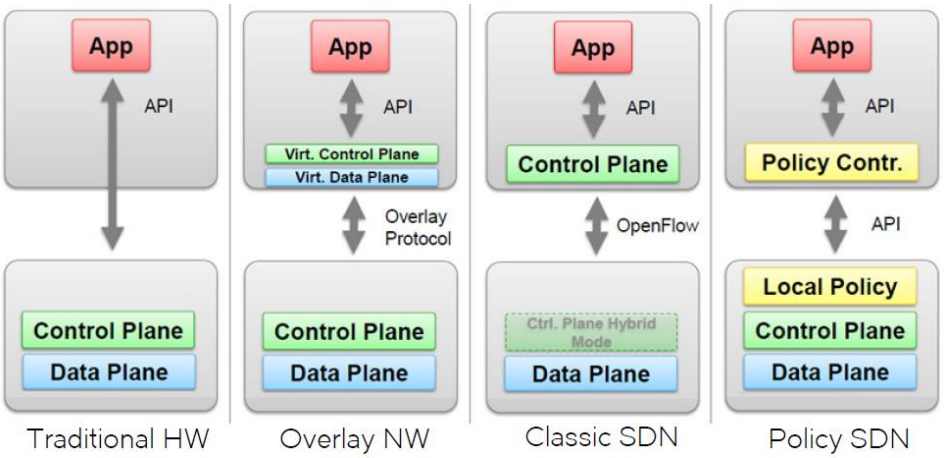
Een virtuele overlay is een virtuele laag bovenop de fysieke netwerkapparatuur. Deze virtuele laag wordt gezien als een apart gescheiden netwerk. Echter, de virtuele netwerk laag is wel afhankelijk van de onderliggende fysieke netwerkapparatuur. Figuur 8 geeft hier een visuele verduidelijking in. Hoewel deze virtuele overlays niet aan alle SDN-definiëringseisen voldoen, wordt deze uitvoering toch meegenomen als SDN-uitvoering. Deze uitvoering kan binnen bepaalde netwerkdelen (zoals datacenter omgevingen) een waardevolle toevoeging zijn.



Figuur 8: Virtuele Overlays (Elsen, 2013)

1.7.4 Policy gericht

De laatste categorie van SDN-uitvoeringen is een policy gerichte uitvoering. Deze uitvoering is ontstaan, doordat – aldus de SDN-visie van Cisco System: *"Many solutions suffer from scalability and performance limitations that result from the use of a centralized, rather than a distributed, control plane."* (Cisco Systems, 2015) – een imperatief SDN-model niet schaalbaar is voor de huidige datacenternetwerken. Deze uitvoering wordt bewerkstelligd met een SDN-controller die gebruik maakt van het declaratieve SDN-model en draagt zorg voor een gecentraliseerde netwerkarchitectuur met gedeeltelijk gesimplificeerde apparatuur.



Figuur 9: SDN-uitvoeringen (Laarhoven, SDN-uitvoeringen, 2015)

1.8 SDN-Southbound protocollen

In paragraaf 1.3.7 is het protocol OpenFlow in grote lijnen besproken. Echter, dit protocol is niet het enige protocol dat gebruikt kan worden voor het aansturen van de netwerkapparatuur binnen een SDN-omgeving. Deze paragraaf geeft een verduidelijking in de diverse Southbound (management) protocollen.

1.8.1 Management Protocollen

Zoals eerder besproken draagt het protocol OpenFlow zorg voor het uitwisselen van flowinformatie tussen de controller en de netwerkapparatuur. Het modificeren van configuratiedata op een OpenFlow (enabled) switch is echter niet mogelijk met dit protocol. Om dit te bewerkstelligen worden protocollen als OF-Config, OVSDB, NETCONF of YANG gebruikt.

1.8.1.1 OF-Config

Het protocol OpenFlow Management and Configuration Protocol (OF-Config) is een aanvulling op het protocol OpenFlow en wordt doorontwikkeld door ONF. Dit protocol maakt omder andere het toekennen van poorten, IP-adressen en OpenFlow controllers mogelijk. Het protocol is opgebouwd uit OpenFlow capable switches, OpenFlow logical switches en OpenFlow configuration points. Een OpenFlow capable switch is een fysiek of virtueel uitgevoerde switch. De logische switch, daarin tegen, bestaat uit een aantal poorten van de OpenFlow capable switch. Daarnaast is een OF-config geconfigureerde switch afhankelijk van het protocol NETCONF (Open Networking Foundation, 2014).

1.8.1.2 OVSDB

Het protocol Open vSwitch Database Management Protocol (OVSDB) wordt gebruikt bij een virtualisatie infrastructuur met Open vSwitch. Het protocol draagt, eveneens, zorg voor het modificeren van configuratiedata en fungeert naast het protocol OpenFlow (Margaret, 2015).

1.8.1.3 NETCONF

Het protocol Network Confirguration Protocol (NETCONF) is een netwerkmanagement protocol ontworpen in 2006 en gestandaardiseerd door IETF. Het standaardiseringsvoorstel staat beschreven in RFC 4741 (IETF, RFC 4741, 2006) en is omstreeks het jaar 2011 gewijzigd naar RFC 6241 (IETF, RFC 6241, 2011). Het protocol verzorgt – op basis van de programmeertaal Extensible Markup Language (XML) – een mechanisme om netwerkapparatuur configuraties te installeren, manipuleren en verwijderen. Door de XML programmeertaal te hanteren, ontstaat een volledige en duidelijke API die door diverse applicaties wordt gebruikt. Netconf maakt gebruik van de commando's get, edit, copy, delete, lock, unlock, close en kill.

1.8.1.4 YANG

Het protocol Yet Another Next Generation (YANG) is een datamodelleringstaal voor het protocol NETCONF. Met behulp van YANG wordt configuratie- en statusdata van NETCONF gestructureerd en gecategoriseerd. Het protocol is ontwikkeld in 2010 als aanvulling op de eerste versie van NETCONF. De RFC 6020 (IETF, RFC 6020, 2010) is gebruikt voor de standaardisering van dit protocol.

1.8.2 Cisco OnePK API

In tegenstelling tot OpenFlow, OF-Config, OVSDB, NETCONF, YANG en OpFlex is Cisco OnePK API geen protocol, maar een compilatie van diverse Cisco Systems propriëtair netwerkapparaat configuratietalen (zoals IOS, IOSd/XE, XR en NX-OS). Deze API is afkomstig van de Cisco Open Network Environment (ONE) architecturale oplossing en functioneert alleen op OnePK ondersteunde hardware. Cisco ONE was één van de eerste SDN-oplossing van Cisco Systems en is later opgevolgd door Cisco Extensible Network Controller (XNC). Cisco XNC is een SDN-oplossing gebaseerd op de OpenDayLight broncode en aangevuld met Cisco propriëtaire netwerktechnieken. Deze oplossing is uiteindelijk gewijzigd naar de Cisco Open SDN Controller. (Wolfgang, 2015)

1.8.3 Cisco OpFlex

Het protocol OpFlex is het antwoord van Cisco Systems op het protocol OpenFlow. Het protocol is bedoeld om de tekortkomingen van een imperatieve SDN-uitvoering aan te vullen. *"Many solutions suffer from scalability and performance limitations that result from the use of a centralized, rather than a distributed, control plane."* (Cisco Systems, 2015). OpFlex wordt gebruikt bij een declaratieve SDN-omgeving en draagt zorg voor zowel het distribueren van policy's als configuraties van diverse netwerkapparatuur. Daarnaast zijn zowel de policy's als de configuratie, die via OpFlex worden gedistribueerd, opgebouwd uit de programmeertaal XML of JavaScript Object Notation (JSON) (Lippis Enterprises., 2012). Op het moment van schrijven is OpFlex een informatieve RFC van IETF (IETF, iRFC draft-smith-opflex-01, 2014).

1.9 SDN-Controllers (Longlist)

Een longlist is bedoeld om een eerste (grove) selectie te maken uit de mogelijk toepasbare oplossingen. Deze lijst is ontstaan door zowel (SDxCentral, 2015) als (Network Computing Editors, 2015) te raadplegen en hieruit een selectie te maken op basis van volledig Enterprise geschikte oplossingen. Daarnaast is deze lijst – door de grote hoeveelheid start-ups en de onvolwassenheid van SDN – opgebouwd uit alleen de bekende traditionele netwerkapparatuur fabrikanten. Ten slotte is declaratieve SDN- (ter vergelijking op het imperatieve model) en een volledig open source (als eis van de opdrachtgever) oplossing meegenomen naar deze lijst. De onderstaande opsomming geeft een overzicht van de mogelijk toepasbare oplossingen:

- Cisco Application Policy Infrastructure Controller;
- Cisco Application Policy Infrastructure Controller - Enterprise Module;
- Cisco Open SDN Controller (voormalig Cisco Extensible Network Controller);
- HP Virtual Applcation Networks (VAN) Controller;
- Brocade Vyatta Controller;
- Extreme Networks OneController;
- NEC ProgrammableFlow Controller;
- Floodlight.

In Bijlage I: Longlist SDN-Controllers is een gedetailleerde tabel over de SDN-controller eigenschappen, SDN-definiëring en typering terug te vinden.

1.10 SDN-Voordelen

De overstap naar een SDN-omgeving levert een aantal voordelen voor zowel de IT-afdeling, de business als de klanten van de organisatie op. De paragrafen 1.10.1 t/m 1.10.10 beschrijven enkele voordelen van een SDN-omgeving.

1.10.1 CapEx

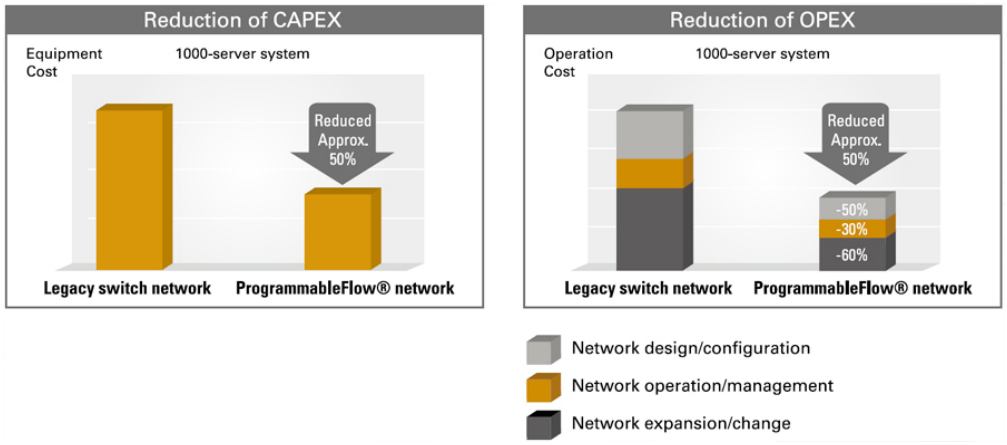
De kosten die met Capital expenditures (CapEx) bedoeld worden, hebben betrekking op kosten voor apparatuur die IT-services dienen te ondersteunen. Deze kosten functioneren als investering en zijn veelal niet terug te verdienen. Binnen de huidige traditionele netwerken dient de netwerkarchitect, bij het ontwerpen of wijzigen van een netwerkdesign, een schatting te maken van de te verwachtte gebruikte netwerkcapaciteit en -apparatuur. Deze schatting omvat meestal een periode van vijf tot zes jaar. Door op deze manier te werk te gaan, wordt de netwerkcapaciteit niet optimaal benut. De gecentraliseerde aard van een SDN-omgeving draagt zorg voor een globaal en up-to-date overzicht van de duurzaamheid van het netwerk. Hierdoor is een netwerkarchitect niet meer genoodzaakt om de voorafgaande capaciteitsschatting te maken en kan het netwerk aan de hand van de vraag worden geschaald. Dit resulteert, op lange termijn, in een reducering in de CapEx. Daarnaast kan netwerkapparatuur (zoals firewalls, intrusion detection/protection systems, load balancers, et cetera) met behulp van Network Function Virtualisation (NFV) virtueel worden uitgevoerd (SDxCentral, 2015).

1.10.2 OpEx

De kosten die met Operating Expenditures (OpEx) bedoeld worden, hebben betrekking op de terugkerende kosten voor het beheren en onderhouden van de netwerkapparatuur. Deze kosten kunnen bestaan uit bijvoorbeeld servicecontracten, werknemerskosten, et cetera. Door de programmeerbaarheid, gecentraliseerde- en geautomatiseerde aard van een SDN-omgeving, worden deze kosten op langere termijn gereduceerd. In de huidige traditionele netwerken is het vereist om elk netwerkapparaat individueel te configureren. Hierdoor zijn fouten in de configuratie snel gemaakt en over het hoofd gezien. Binnen een SDN-omgeving, daarin tegen, kunnen repetitieve taken (zoals het configureren van diverse access- en distributie laag switches) geautomatiseerd worden uitgevoerd. De configuratie wordt eenmaal op de gecentraliseerde SDN-controller geprogrammeerd en vervolgens via het southbound configuratie protocol naar de desbetreffende hardware gedistribueerd. Daarnaast kunnen, indien gemachtigd en ondersteund, SDN-applicaties geautomatiseerd (tijdelijke) aanpassingen aan het netwerk maken en is menselijke interactie niet noodzakelijk. Ter adstructie kan het voorbeeld van het verplaatsen van virtuele machines aan de hand van VMware VMotion worden gebruikt. Tijdens het, met deze techniek, verplaatsen van virtuele machines is een laag 2 tunnel tussen beide locaties gewenst.

Door de integratie van VMware en de SDN-controller kan VMware de SDN-controller (via de Northbound API) verzoeken om de laag 2 tunnel aan te maken en te verwijderen na het voltooien van de verplaatsing (SDxCentral, 2015).

Figuur 10 geeft een visuele verduidelijking in de verlaging van de CapEx en OpEx kosten. Deze vergelijking is gemaakt en aangetoond door NEC en haar, open OpenFlow gebaseerde, ProgrammableFlow SDN-controller.



Figuur 10: Reduction CapEx/OpEx (NEC, 2015)

1.10.3 Innovatie

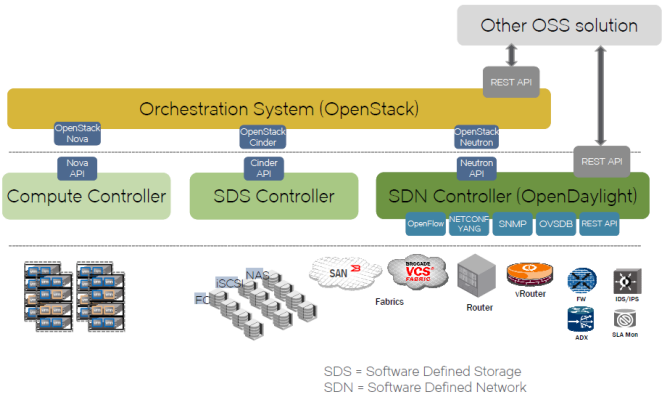
Door gebruik te maken van SDN heeft de IT-afdeling meer tijd voor innovatie beschikbaar. Binnen de huidige traditionele netwerken spendeert een gemiddelde IT-afdeling 60% van de tijd aan onderhoudswerkzaamheden en 40% aan het verbeteren en innoveren van de netwerkarchitectuur. Deze 60% zijn te relateren aan de OpEx kosten en daardoor, voor de business, niet winstgevend. Een SDN-omgeving draagt, door de programmeerbaarheid, geautomatiseerde- en gecentraliseerde aard, zorg voor een kanteling in deze percentages. Door deze kanteling kan de IT-afdeling meer tijd spenderen voor zowel het verbeteren als innoveren en kunnen de kosten meer naar de business worden uitgelijnd (Samson, 2014).

1.10.4 Time-to-market

IT-projecten worden sneller gerealiseerd door invoering van een SDN-omgeving. IT-project hebben, binnen de huidige traditionele netwerken, veelal een levensduur van een aantal maanden. De lange levensduur van deze projecten is te wijden aan de complexiteit van het netwerk en zowel de slechte coöperatie als communicatie tussen de diverse IT-afdelingen. Binnen een SDN-omgeving dienen – door de integratie van bijvoorbeeld Hypervisors, zowel cloud- als storage voorzieningen en netwerk bewuste applicaties – de diverse IT-afdelingen meer met elkaar samen te werken. Daarnaast draagt de programeerbaarheid, netwerk abstractie en zowel de geautomatiseerde- als de gecentraliseerde aard zorg voor een gesimplificeerd netwerk. De combinatie van een betere coöperatie als een gesimplificeerd netwerk reduceert de levensduur van IT-projecten naar slechts dagen tot weken (Lenten, 2015).

1.10.5 Ontschotting IT-afdelingen

Een SDN-omgeving draagt zorg voor betere coöperatie en communicatie tussen de diverse IT-afdelingen. Zoals in paragraaf 1.10.4 eveneens al blijkt, draagt de invoering van een SDN-omgeving zorg voor een naadloze integratie van 'compute', 'storage' en 'networking'. Deze integratie lijdt uiteindelijk tot één SDx-framework en dient te worden aangevuld met de kennis van software en programmering. Hierdoor is een goed samenwerkingsverband tussen de afdelingen noodzakelijk. Iedere IT-afdeling is verantwoordelijk voor één en hetzelfde framework. Figuur 11 geeft een visuele weergave van dit framework. Door deze samenvoeging van afdelingen ontstaat een uniforme IT-afdeling gericht op het leveren van goed afgestemde IT-services (Laarhoven, Gastcollege Brocade, 2015).



Figuur 11: SDx-Framework (Laarhoven, SDx-Framework, 2015)

1.10.6 Hogere netwerkbetrouwbaarheid

De betrouwbaarheid wordt beter gewaarborgd door de invoering van een SDN-omgeving. Binnen traditionele netwerken worden incidenten veelal pas opgemerkt als de performance van het netwerk niet naar behoren is. Daarnaast is deze methode van incidentafhandeling reactief en vaak tijdrovend. De gecentraliseerde-, geautomatiseerde en abstracte aard van een SDN-omgeving draagt zorg een gedetailleerd (in real-time uitgevoerd) monitoringsniveau. Daarnaast ontstaat de mogelijkheid om via applicaties het netwerk geautomatiseerd en proactief incidenten te laten afhandelen. Hierdoor merkt de eindgebruiker niet eens dat een incident plaatsgevonden heeft (Lenten, 2015).

1.10.7 Gesimplificeerd beheer

Een SDN-omgeving verlaagt de beheerlast aanzienlijk. Zoals in paragraaf 1.10.2 eveneens duidelijk wordt, dient de netwerkapparatuur – binnen traditionele netwerken – handmatig en individueel geconfigureerd en beheert te worden. Deze manier van werken is tijdrovend en foutgevoelig. Doordat de SDN-controller een gecentraliseerde aard heeft, hoeft de onderliggende netwerkapparatuur maar één keer – op de controller – geconfigureerd te worden. Deze SDN-controller zorgt vervolgens voor het distribueren van de configuraties naar de juiste netwerkapparatuur. Het wijzigen van configuraties wordt eveneens op deze manier bewerkstelligd. Hierdoor ontstaat het zogenaamde 'zero touch deployment'. Daarnaast zorgt de apparatuur abstractie voor een duidelijker en gesimplificeerd beeld, voor zowel de IT-beheerders als de SDN-applicaties, van de onderliggende netwerkapparatuur. Ter adstructie wordt het invoeren van het protocol Quality of Service (Qos) gebruikt. Om dit protocol toe te passen binnen een netwerk, dienen – bij een traditioneel netwerk – alle ondersteunde netwerkapparaten worden voorzien met dezelfde QoS-policy. Binnen een SDN-omgeving wordt dit protocol toegepast aan de hand van een SDN-applicatie. Binnen deze applicatie wordt aangegeven welk verkeer prioriteit vereist en de SDN-controller verwerkt dit vervolgens naar de onderliggende netwerkapparatuur (Lenten, 2015).

1.10.8 Automatisering

In een SDN-omgeving zijn repetitieve taken verleden tijd. De gedecentraliseerde aard van traditionele netwerken zorgt ervoor dat elk netwerkapparaat apart geconfigureerd dient te worden. Binnen een SDN-omgeving worden deze repetitieve taken bewerkstelligd door de gecentraliseerde SDN-controller. Daarnaast kunnen configuraties tijdelijk of permanent door SDN-applicaties, geheel geautomatiseerd, worden gewijzigd. Ter adstructie wordt een malware-detectie applicatie gebruikt. Een malware-detectie applicatie is een SDN-applicatie die het netwerkverkeer monitort op virussen en malware. Indien zulk malafide netwerkverkeer wordt ontdekt, kan de applicatie bijvoorbeeld ervoor zorgen dat al het netwerkverkeer – van het malware veroorzakende apparaat – naar een blackhole netwerk wordt geleid (SDxCentral, 2015).

1.10.9 Doelgroep segmentatie

Een SDN-omgeving maakt doelgroep segmentatie overzichtelijker en beter beheersbaar. Doelgroep segmentatie wordt in de huidige traditionele netwerken bewerkstelligd met behulp van VLAN's. Echter, deze VLAN's zijn veelal gecompliceerd en zonder uniforme naamgeving uitgevoerd. Daarnaast zijn VLAN's van origine niet bedoeld om netwerk in segmenten te delen. De oorspronkelijke bedoeling van VLAN's is het reduceren van broadcastdomeinen en spanning-tree instances. De complexheid binnen deze VLAN's ontstaat, doordat deze bestaan uit een virtueel layer 2 broadcast domein bestaan en veelal alleen de access laag op layer 2 is uitgevoerd. Een distributie laag switch wordt hierdoor als VLAN Trunking Protocol (VTP)[7] domeingrens gezien. Binnen de diverse SDN-uitvoeringen wordt doelgroep segmentatie op een tweetal manier bewerkstelligd, namelijk aan de hand van Virtual Extensible Local Area Network (VXLAN), Virtual Tenant Network (VTN) en ACI End-point Group (EPG) (Lenten, 2015).

1.10.9.1 VXLAN

Binnen een virtueel overlay SDN-uitvoering wordt veelal gebruik gemaakt van VXLAN om netwerksegmentatie toe te passen. VXLAN is te benoemen als de opvolger van het traditionele VLAN. Met behulp van VXLAN wordt het mogelijk om de functionaliteiten van een VLAN over een layer 3 netwerk te spannen. Daarnaast biedt, in tegenstelling tot de 4096 mogelijkheden binnen een VLAN, VXLAN ruimte voor ongeveer 16 miljoen laag 2 broadcast domeinen. VXLAN staat beschreven in de RFC 7348 (IETF, RFC 7348, 2014) en wordt door de auteurs beschreven als: *"VXLAN could also be called a tunneling scheme to overlay Layer 2 networks on top of Layer 3 networks."* (Mahalingam, et al., 2014). Met behulp van de combinatie tussen een SDN-omgeving en het gebruik van VXLAN draagt zorg voor een gesimplificeerde doelgroep segmentatie met een uniforme naamgeving (Mahalingam, et al., 2014).

1.10.9.2 Virtual Tenant Network

Binnen de Classic- en Hybrid SDN-uitvoeringen wordt veelal gebruik gemaakt van een VTN om netwerksegmentatie toe te passen. VTN wordt bewerkstelligd aan de hand van een SDN-applicatie en maakt gebruik van de netwerkapparatuur abstractie van de SDN-controller. Deze abstractie draagt zorg voor de mogelijkheid om meerdere logische netwerken (virtuele tenants), binnen hetzelfde fysieke netwerk, te maken. Deze logische netwerken zijn in eerste instantie geheel van elkaar gescheiden en worden weergegeven als een normaal laag 2-3 netwerk. Indien gewenst kan met behulp van een vBridge of een vRouter tussen de diverse VTN's worden gecommuniceerd. Daarnaast maakt een VTN, door de netwerkapparatuur abstractie, alleen gebruik van de benodigde netwerkapparatuur om een end-to-end communicatie op te zetten. Met behulp van de combinatie tussen een SDN-omgeving en het gebruik van de VTN-applicatie draagt zorg voor een gesimplificeerde doelgroep segmentatie met een uniforme naamgeving (Kudo, 2014) (Linux Foundation Collaborative Projects, 2015).

1.10.9.3 ACI End-point Group

Binnen een policygerichte SDN-uitvoering, wordt netwerksegmentatie bewerkstelligd met behulp van End-Point Group (EPG)'s. Een EPG is een verzameling van endpoints (groep gebruikers of eindstations) en vertegenwoordigen een divers assortiment van functionaliteiten. Zo kan een EPG dienen als VLAN, VXLAN, subnet, applicatieonderdeel groep, ontwikkelfasering en als een netwerkzone. Door deze diversiteit aan functionaliteiten kan segmentatie, binnen deze SDN-uitvoering, op vele manieren worden toegepast. Doordat, binnen traditionele netwerken, VLAN's eveneens als netwerksegmentatie worden gebruikt, kunnen deze veelal één-op-één worden overgenomen naar EPG's. Echter, door de gecentraliseerde- en op applicatie gebaseerde aard van deze SDN-uitvoering ontstaat een gesimplificeerde doelgroep met een uniforme naamgeving (Cisco Systems, 2015).

[7] Het protocol VTP draagt zorg voor het transporteren van de juiste VLAN-naamgeving binnen het desbetreffende VTP-domein.

1.10.10 Testen in Live-omgeving

Een SDN-omgeving maakt het mogelijk om nieuwe ontwikkelingen in een productieomgeving te testen. Het testen van nieuwe ontwikkelingen dient, binnen traditionele netwerken, altijd op gesepareerde netwerkapparatuur te geschieden. Echter, deze manier van testen brengt diverse complicaties met zich mee. Zo is het bijvoorbeeld lastig rekening te houden met alle mogelijke situaties die zich voordoen in een productieomgeving. Met behulp van een SDN-omgeving daarin tegen kan eenvoudig aan de hand van protocollen zoals VXLAN, VTN of EPG een 'virtuel overlay' worden gecreëerd. Deze 'virtuele overlays' worden aangemaakt over de huidige productieomgeving heen en kunnen daardoor met deze omgeving rekening houden. Daarnaast is een dergelijk test overlay, in tegenstelling tot een testomgeving in huidige traditionele netwerken, binnen enkele seconden tot minuten gemaakt en kan het gefaseerd worden ingevoerd (Bijlsma, 2014) (Laarhoven, Gastcollege Brocade, 2015).

1.11 SDN-nadelen

De overstap naar een SDN-omgeving levert – op het moment van schrijven – een aantal nadelen voor zowel de IT-afdeling, de business als de klanten van de organisatie op. Een aantal van deze nadelen worden in paragraaf 1.11.1 t/m 1.11.5 beschreven.

1.11.1 Single-point-of-failure

De gecentraliseerde aard van een SDN-omgeving draagt zorg voor een single-point-of-failure. Binnen een niet in federatie fungerende SDN-controlleromgeving wordt gebruik gemaakt van één enkele SDN-controller. Hierdoor is deze controller het enige aansturingspunt van het gehele netwerk. Indien bij deze controller een storing ontstaat of door een Distributed Denial of Service (DDoS) aangevallen wordt, kan het gehele netwerk gecompromitteerd worden (Hogg, 2014).

1.11.2 Application Programmable Interface

Een aantal van de Southbound-, Northbound- en East/Westbound API's zitten – op het moment van schrijven – nog in de eerste levensfase. Hierdoor zijn zwakheden binnen deze interfaces niet geëxploiteerd en daardoor nog niet bekend. Daarnaast kunnen door de grote hoeveelheid aan de diverse API's meer te exploiteren zwakheden ontstaan.

1.11.3 Onbekend gebied

Door de jonge geaardheid van het SDN-framework ontstaat veel verwarring en onduidelijkheden bij IT-professionals. Software-Defined Networking is, op het moment van schrijven, een hot-topic bij zowel de IT-professionals als de fabrikanten van netwerkapparatuur. Echter, de grote hoeveelheid verschillende SDN-definities en SDN-methodes nog het ontbreken van een volwaardig functionerende SDN-omgeving levert de nodige onduidelijkheden op. Daarnaast werkt een SDN-omgeving op een geheel andere manier dan traditionele netwerken en is het vergaren van kennis – door de grote hoeveelheid aan zowel API's als SDN-methodes – een tijdrovende opgave.

1.11.4 Hoge investeringskosten

Afhankelijk van de te implementeren SDN-methode kunnen de investeringskosten hoger uitvallen dan bij andere SDN-methodes. Ter adstructie wordt het de policy gerichte SDN-methode gebruikt. Deze methode wordt, op het moment van schrijven, maar door één fabrikant aangeboden en is – binnen het Datacenter netwerkgedeelte – geheel afhankelijk van bepaalde apparatuur. Daarnaast dient deze methode, in tegenstelling tot andere methodes, voorzien te zijn van een zogenaamde Spine / Leaf bekabelingsontwerp. Een Spine / Leaf bekabelingsontwerp wil zeggen dat iedere Leaf (access laag apparatuur) met elke Spine (core laag apparatuur) verbonden dient te zijn (Bijlsma, 2014).

1.11.5 Kwetsbaarheden onderliggend besturingssysteem

SDN-controller kunnen zowel virtueel als een fysiek apparaat worden uitgevoerd. Echter, beide uitvoeringen zijn veelal gebaseerd op een Unix besturingssysteem. Hierdoor is niet alleen de controller, maar eveneens het onderliggende besturingssysteem kwetsbaar voor het exploiteren van systeemfouten. Daarnaast ontstaat bij de virtueel uitgevoerde variant nog een extra kwetsbaarheid, namelijk de hypervisorsoftware. Indien deze software of het onderliggende besturingssysteem wordt geëxploiteerd, is de stap naar het uitvoeren van een Denial of Service (DoS)-aanval op de controller nihil (Hogg, 2014).

1.11.6 Ontschotting IT-afdelingen

De ontschotting van de verschillende IT-afdelingen kan zowel een voor- als nadelig effect op de business hebben. Het voordelig effect van deze ontschotting is in paragraaf 1.10.5 al naar voren gekomen. Het nadelig effect ontstaat doordat deze ontschotting een hele impact op de business en IT-afdelingen heeft. Zo kan bijvoorbeeld een reorganisatie in de business of de eis tot bij- / omscholing van het IT-personeel ontstaan door deze ontschotting (Laarhoven, Gastcollege Brocade, 2015).

Tabellenlijst

Figurenlijst

Bibliografie

Alcatel-Lucent. (2015, 05 01). *Software Defined Networks: Optimizing Your Environment*. Retrieved from Alcatel-Lucent Enterprise: http://enterprise.alcatel-lucent.com/?innovation=NewThinking&page=SoftwareDefinedNetworks

Bijlsma, T. (2014). Application Centric Infrastructure. *Application Centric Infrastructure.* Amsterdam: Cisco Systems.

Bjorklund, M. (2010). *IETF, RFC 6020.* Internet Engineering Task Force. Retrieved from http://tools.ietf.org/html/rfc6020

Bredel, M. (2013, 06 18). Openflow-based Multipath Switching in the Wild. Pasadena, California, USA. Retrieved from http://indico.cern.ch/event/236955/contribution/13/material/slides/0.pdf

Brocade. (2015, 05 01). *Software-Defined Networking (SDN)* . Retrieved from Brocade: http://www.brocade.com/solutions-technology/technology/software-defined-networking/index.page

Campbell, A., Katzela, I., Miki, K., & Vicente, J. (1998). *OPEN SIGNALING FOR ATM, INTERNET ANDMOBILE NETWORKS (OPENSIG'98).* Ontario: University of Toronto. Retrieved from https://www.academia.edu/9312316/Open_Signaling_for_ATM_INTERNET_and_Mobile_Networks_OPENSIG98

Casado, M., Freedman, M., Pettit, J., Luo, J., Gude, N., McKeown, N., & Shenker, S. (2009, August). Rethinking Enterprise Network Control. *IEEE/ACM TRANSACTIONS ON NETWORKING, 17*(4), 1270-1283. doi:10.1109/TNET.2009.2026415

Casado, M., Freedman, M., Pettit, J., Luo, J., Shenker, S., & McKeown, N. (2007). Ethane: Taking Control of the Enterprise. *ACM SIGCOMM 2007.* Kyoto Japan: Berkeley, University of California | International Computer Science Institute | Stanford University.

Cisco Systems. (2008, 05 21). *Campus Network for High Availability Design Guide*. Retrieved from Campus Network for High Availability Design Guide: http://www.cisco.com/c/en/us/td/docs/solutions/Enterprise/Campus/HA_campus_DG/hacampusdg.html

Cisco Systems. (2012, 04 11). Building the Mobile Internet. Amsterdam, Noord Holland, Nederland. Opgehaald van http://www.slideshare.net/wierenga/building-the-mobile-internet

Cisco Systems. (2015, 04 15). *Cisco Application Centric Infrastructure*. Retrieved from OpFlex: An Open Policy Protocol: http://www.cisco.com/c/en/us/solutions/collateral/data-center-virtualization/application-centric-infrastructure/white-paper-c11-731302.html

Cisco Systems. (2015, 04 30). *Cisco Application Centric Infrastructure (ACI) - Endpoint Groups (EPG) Usage and Design*. Retrieved from Cisco Application Centric Infrastructure: http://www.cisco.com/c/en/us/solutions/collateral/data-center-virtualization/application-centric-infrastructure/white-paper-c11-731630.html

Dawson, P., & Wolf, C. (2011, 07 22). *Gartner.* Retrieved from Virtualization Key Initiative Overview: https://www.gartner.com/doc/1745020/virtualization-key-initiative-overview

Doria, A., Hadi Salim, J., Haas, R., Khosravi, H., Wang, W., Dong, L., . . . Halpern, J. (2010). *IETF, RFC 5810.* Internet Engineering Task Force. Retrieved from https://tools.ietf.org/html/rfc5810

Elsen, C. (2013, 09 04). *Physical networks for VMware NSX*. Retrieved from EdgeCloud: https://www.edge-cloud.net/2013/09/physical-networks-for-vmware-nsx/

Enns, R. (2006). *IETF, RFC 4741.* Internet Engineering Task Force. Retrieved from http://tools.ietf.org/html/rfc4741

Enns, R., Bjorklund, M., Schoenwaelder, J., & Bierman, A. (2011). *IETF, RFC 6241.* Internet Engineering Task Force. Retrieved from http://tools.ietf.org/html/rfc6241

Fielding, R. (2000). *Architectural Styles and the Design of Network-based Software Architectures.* Irvine, CA, USA: University of California.

Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., & Berners-Lee, T. (1999). *IETF, RFC 2616.* Internet Engineering Task Force.

Gahsinsky, I. (Oktober 2011). Warehouse Scale Datacenters:The case for a new approach to networking. *Open Networing Summit* (p. 5). Palo Alto, CA, USA: Stanford University: Open Network Foundation. Retrieved from http://opennetsummit.org/archives/apr12/site/talks/gahinsky-tue.pdf

Göransson, P., & Black, C. (2014). In S. Elliot, *Software Defined Networks: A Comprehensive Approach* (p. 15). Waltham: Morgan Kaufmann (an imprint of Elsevier).

Göransson, P., & Black, C. (2014). §1.4 - Traditional Switch Architecture. In S. Elliot, *Software Defined Networks: A Comprehensive Approach* (pp. 7-11). Waltham: Morgan Kaufmann (an imprint of Elsevier).

Göransson, P., & Black, C. (2014). ForCES Design. In S. Elliot, *Software Defined Networks: A Comprehensive Approach* (p. 46). Waltham, MA: Morgan Kaufmann (an imprint of Elsevier).

Göransson, P., & Black, C. (2014). Networking functionality migrating to hardware. In S. Elliot, *Software Dfined Networks: A Comprohensive Approach* (p. 22). Waltham: Morgan Kaufmann (an imprint of Elsevier).

Greenberg, A., Hjalmtysson, G., Maltz, D., Myers, A., Rexford, J., Xie, G., . . . Zhang, H. (2005). *A Clean Slate 4D Approach to Network Control and Management.* Pittsburgh, PA, USA: Carnegie Mellon University: School of Computer Science. Retrieved from http://www.cs.cmu.edu/~4D/papers/greenberg-ccr05.pdf

Haleplidis, E., Pentikousis, K., Denazis, S., Hadi Salim, J., Meyer, D., & Koufopavlou, O. (2015). *IETF, RFC 7426.* Internet Engineering Task Force.

Hogg, S. (2014, 10 28). *SDN Security Attack Vectors and SDN Hardening*. Retrieved from http://www.networkworld.com/article/2840273/sdn/sdn-security-attack-vectors-and-sdn-hardening.html

IEEE. (2009). *IEEE Std 802.1X-2010.* Institude of Electrical and Electronics Engineers. Retrieved from http://www.ieee802.org/1/pages/802.1x-2010.html

James, M. (1983). Create, read, update and delete. In M. James, *Managing the database environment* (p. 381). Englewood Cliffs, NJ, USA: Prentice-Hall.

Juniper Networks. (2015, 05 01). *SDN*. Retrieved from Juniper Networks: http://www.juniper.net/us/en/dm/sdn/

Kikken, F. (2015). *Software Defined Networking - Project Initiatie Document.* Maastricht: Maastricht University.

Klebanov, D. (2013, 09 16). *An alternate method of SDN network programmability*. Retrieved from NetworkWorld: http://www.networkworld.com/article/2169967/tech-primers/an-alternate-method-of-sdn-network-programmability.html

Kooij, J. v. (2013). *SLA_MAASnet_0.5.pdf.* Maastricht: Maastricht University.

Kudo, M. (2014). Virtual Tenant Network. *OpenDaylight Network Virtualization and its Future Direction* (pp. 17-19). NEC Corporation.

Laarhoven, W. v. (2015). Gastcollege Brocade. *Gastcollege Brocade.* Heerlen: Brocade.

Laarhoven, W. v. (2015). SDN-uitvoeringen. *SDN From a Brocade Point of View* (p. 85). Heerlen: Brocade.

Laarhoven, W. v. (2015). SDx-Framework. *SDN From a Brocade Point of View* (p. 87). Heerlen: Brocade.

Lenten, E. (2015). *Zeven business voordelen van Software Defined Networking.* Capelle aan den IJssel: Axians.

Linux Foundation Collaborative Projects. (2015, 02 26). *OpenDaylight Wiki*. Retrieved from OpenDaylight Virtual Tenant Network (VTN):Overview: https://wiki.opendaylight.org/view/OpenDaylight_Virtual_Tenant_Network_(VTN):Overview

Lippis Enterprises. (2012). *Lippis Report 220: How Open Is Cisco's ACI?* Retrieved from the Lippis Report: http://lippisreport.com/2014/04/lippis-report-220-how-open-is-ciscos-aci/

Maastricht University. (2013, 03 04). *MAASnet*. Opgehaald van MAASnet: https://myum.unimaas.nl/irj/portal?NavigationTarget=ROLES://portal_content/DEEPLINK/nl.ifa.pages.DeeplinkPage&deepLink=https%3A%2F%2Fintrassl.myum.unimaas.nl%2Fweb%2FICTS2%2FICTSProductenEnDienstencatalogus%2FMAASnet.htm

Maastricht University. (2015, 03 18). *Network Overview*. Retrieved from Network Overview: http://- IP-adres -/

Mahalingam, M., Dutt, D., Duda, K., Agarwal, P., Kreeger, L., Sridhar, T., . . . Wright, C. (2014). *IETF, RFC 7348.* Internet Engineering Task Force. Retrieved from http://tools.ietf.org/html/rfc7348#

Margaret, R. (2015, 04 15). *SearchSDN*. Retrieved from OVSDB (Open vSwitch Database Management Protocol): http://searchsdn.techtarget.com/definition/OVSDB-Open-vSwitch-Database-Management-Protocol

McKeown, N., Parulkar, G., Shenker, S., Anderson, T., Peterson, L., Turner, J., . . . Reford, J. (2008, April). OpenFlow: Enabling Innovation in Campus Networks. *ACM SIGCOMM Computer Communication Review, 38*(2), 69-74.

NEC. (2015, 04 28). *Proven Customer Benefits*. Retrieved from Innovation: OpenFlow/SDN: http://www.nec.com/en/global/rd/research/cl/sdn/innovation/page03.html

Network Computing Editors. (2015, 03 03). *SDN Controller Handbook*. Retrieved from Network Computing: Connecting the infrastructure community: http://www.networkcomputing.com/networking/sdn-controller-handbook/d/d-id/1319300?page_number=1

Newman, P., Edwards, W., Hinden, R., Hoffman, E., Ching Liaw, F., Lyon, T., & Minshall, G. (1996). *IETF, RFC 1987.* Internet Engineering Task Force. Retrieved from http://www.networksorcery.com/enp/rfc/rfc1987.txt

Open Networking Foundation . (2015, April 08). *Software-Defined Networking (SDN) Definition*. Retrieved from What is SDN?: https://www.opennetworking.org/sdn-resources/sdn-definition

Open Networking Foundation. (2013). *What is ONF?* Palo Alto, California, USA: Open Networking Foundation. Retrieved from https://www.opennetworking.org/images/stories/downloads/about/onf-what-why.pdf

Open Networking Foundation. (2014). *OpenFlow Management and Configuration Protocol v1.2.* Palo Alto, CA, USA: Open Networking Foundation.

Open Networking Foundation. (2014). *OpenFlow Switch Specification.* The Open Networking Foundation. Retrieved from https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-switch-v1.5.0.noipr.pdf

Oracle. (2015, 05 01). *Oracle SDN*. Retrieved from Oracle: http://www.oracle.com/us/products/networking/virtual-networking/sdn/overview/index.html

Samson, M. (2014). Business Allignment. *The future of networking* (p. 2). Amsterdam: Cisco Systems.

SDxCentral. (2015, 04 24). *Comprehensive List of SDN Controller Vendors & SDN Controllers*. Retrieved from SDxCentral: https://www.sdxcentral.com/resources/sdn/sdn-controllers/sdn-controllers-comprehensive-list/

SDxCentral. (2015, 05 28). *SDN Automation and Programmability*. Retrieved from SDxCentral: https://www.sdxcentral.com/resources/devops/programmability-network-automation-sdn-networks/

SDxCentral. (2015, 04 15). *SdxCentral*. Retrieved from What is Cisco OpenFlow?: https://www.sdxcentral.com/resources/cisco/cisco-openflow/

SDxCentral. (2015, April 09). *Software-Defined Networking*. Retrieved from What are SDN Northbound APIs?: https://www.sdxcentral.com/resources/sdn/north-bound-interfaces-api/

SDxCentral. (2015, 04 28). *What's Software-Defined Networking (SDN)?* Retrieved from SDxCentral: https://www.sdxcentral.com/resources/sdn/what-the-definition-of-software-defined-networking-sdn/

Shenker, S. (2011). The Future of Networking, and the Past of Protocols. *Open Networking Summit* (p. 11). Stanford, CA, USA: ONS. Retrieved from http://opennetsummit.org/archives/oct11/shenker-tue.pdf

Smith, M., Adams, R., Dvorkin, M., Laribi, Y., Pandey, V., Garg, P., & Weidenbacher, N. (2014). *IETF, iRFC draft-smith-opflex-01.* Internet Engineering Task Force. Retrieved from http://tools.ietf.org/id/draft-smith-opflex-01.txt

SURFnet. (2015, 02 23). *SURFdrive*. Retrieved from SURFdrive: https://www.surf.nl/diensten-en-producten/surfdrive/surfdrive.html

Wikipedia. (2015, 03 14). Create, read, update and delete. In M. James, *Managing the database environment* (p. 381). Englewood Cliffs, NJ, USA: Prentice-Hall. Retrieved from Wikipedia: http://en.wikipedia.org/wiki/Create,_read,_update_and_delete

Wikipedia. (2015, 01 31). *Unified communications*. Retrieved from Unified communications: http://en.wikipedia.org/wiki/Unified_communications

Wolfgang, R. (2015). The APIC Story: APIC-EM - SDN in the Enterprise. *Cisco Live 2015 Milaan: TECSDN-3600 - Advanced - APIC Enterprise Module - SDN in the Enterprise* (pp. 1 - 129). Milaan: Cisco Live.

Yang, L., Dantu, R., Anderson, T., & Gopal, R. (2004). *IETF, RFC 3746.* Internet Engineering Task Force. Retrieved from https://tools.ietf.org/html/rfc5810

Bijlage I: Longlist SDN-Controllers

Naam	Cisco Application Policy Infrastructure Controller (APIC)	Cisco APIC - Enterprise Module (EM)	Cisco Open SDN Controller (XNC)	Brocade Vyatta Controller	HP Virtual Applcation Networks (VAN)	Extreme Networks OneController	NEC Programmable Flow Controller	Floodlight
SDN-controller eigenschappen								
Fabrikant	Cisco Systems	Cisco Systems	Cisco Systems	Brocade	Hewlett-Packard	Extreme Networks	NEC	Project Floodlight community
Type controller	Appliance	Software / Appliance	Software	Software	Software	Fysiek- / Virtueel Appliance	Appliance	Software
Source code afkomstig van	n.v.t. (eigen bron)	n.v.t. (eigen bron)	OpenDayLight	OpenDayLight	n.v.t. (eigen bron)	OpenDayLight	OpenDayLight	Eigen source
South protocol	OpFlex, L4-7 Scripting API	CLI	OpenFlow v1.3, OnePK API	OpenFlow v1.3, NETCONF, YANG, OVSDB	OpenFlow v1.3	OpenFlow v1.3, NETCONF, OVSDB	OpenFlow v1.3	OpenFlow
North protocol	REST	REST	OSGi, REST	REST	REST	REST	REST	REST
Status	Operationeel	Aangekondigd	Operationeel	Operationeel	Operationeel	Operationeel	Operationeel	Operationeel
SDN-definitie normering								
Draagvlak scheiding	Nee	Nee	Ja	Ja	Ja	Ja	Ja	Ja
Gesimplificeerde apparatuur	Gedeeltelijk	Gedeeltelijk	Ja	Ja	Ja	Ja	Ja	Ja
Gecentraliseerd management	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
Mogelijkheid tot automatisering	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
Netwerk abstractie (virtualisatie)	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
Openheid	Sourcecode: Ja Hardware: Nee	Sourcecode: Nee Hardware: Nee	Sourcecode: Ja Hardware: Ja	Sourcecode: Ja Hardware: Ja	Sourcecode: Nee Hardware: Ja	Sourcecode: Ja Hardware: Ja	Sourcecode: Ja Hardware: Ja	Sourcecode: Ja Hardware: Ja
SDN-typering								
SDN-uitvoering	Declaratief	Declaratief	Imperatief	Imperatief	Imperatief	Imperatief	Imperatief	Imperatief
SDN-type	Policygericht	Policygericht	Classic-, Hybrid SDN	Classic-, Hybrid SDN	Classic-, Hybrid SDN	Classic-, Hybrid SDN	Classic-, Hybrid SDN	Classic SDN
Omgeving	Datacenter	Enterprise	Enterprise	Enterprise, Datacenter	Enterprise	Enterprise, Datacenter	Enterprise, Datacenter	Enterprise, Datacenter
Overig								
Customerservice	Support via fabrikant	Support via fabrikant	Support via fabrikant	Support via fabrikant	Support via fabrikant	Support via fabrikant	Support via fabrikant	Support via Open Source Community
Opmerkingen	-	-	-	-	≥ 50 OF-apparaten	Virtueel uitgevoerd via servicecontract	≥ 100 OF-apparaten	-

Tabel 5: SDN-Controller vergelijking