HSR
HOCHSCHULE FÜR TECHNIK
RAPPERSWIL
FHO Fachhochschule Ostschweiz

ins institute for networked solutions

Network Virtualization

Studienarbeit

Abteilung Informatik
Hochschule für Technik Rapperswil

Herbstsemester 2015

Autor:	Pascal Meier
Betreuer:	Prof. Beat Stettler
Projektpartner:	INS Institute for Networked Solutions
Experte:	
Gegenleser:	

Aufgabenstellung

Ausschreibung: In der IT ist der Trend zur Virtualisierung unaufhaltbar. Dank der Trennung von Hard- und Software kann eine bessere Auslastung erzielt werden und gleichzeitig können flexiblere Lösungen zur Verfügung gestellt werden. Dieser Trend hat nun auch die Netzwerk-Industrie erreicht, wo vermehrt auch Router, Switches, Firewalls und vieles mehr als "virtuelle Maschinen" erhältlich sind und beliebig im Netzwerk platziert werden können.

In der Netzwerk-Industrie tobt zurzeit ein heisser Kampf zwischen den Big Players (Cisco, Aruba, HP usw.) Neu drängen auch "artfremde" Hersteller wie zum Beispiel VMware mit der Softwarelösung NSX in diesen Markt.

Marketing technisch versprechen alle Produkte die ideale Lösung zu sein. Ziel dieser Arbeit ist es, drei Lösungen zu implementieren, testen und analysieren. Basierend auf den vorab zu definierenden Anforderungen sollen neben einem funktionellen Vergleich auch operative Aspekte wie die Einfachheit (Installation und Betrieb), Skalierung, Performance, Wartbarkeit, Einsatzgebiete und die Technische Umsetzung untersucht werden. Die Resultate werden im Rahmen einer Informationsveranstaltung an Kunden und Interessierte präsentiert.

Voraussetzungen: Die Studenten sollten über gutes Know-How im Netzwerk-Umfeld verfügen und die Neugier mitbringen, auch brandneue Technologieren zu testen. Kreativität zur Erweiterung der Aufgabenstellung ist ebenfalls erwünscht

Ort, Datum:

Name, Unterschrift:

Erklärung

Ich erkläre hiermit,

- dass ich die vorliegende Arbeit selber und ohne fremde Hilfe durchgeführt habe, ausser derjenigen, welche explizit in der Aufgabenstellung erwähnt ist oder mit dem Betreuer schriftlich vereinbart wurde,
- dass ich sämtliche verwendeten Quellen erwähnt und gemäss gängigen wissenschaftlichen Zitierregeln korrekt angegeben habe.
- dass ich keine durch Copyright geschützten Materialien (z.B. Bilder) in dieser Arbeit in unerlaubter Weise genutzt habe.

Ort, Datum:

Name, Unterschrift:

Abstract

Software Defined Networking (SDN) ist im Moment das Thema in der IT Branche. Immer mehr Firmen spielen mit dem Gedanken auf den aktuellen Hype aufzuspringen. Im Zuge von immer grösseren Datenmengen und dem Trend in Richtung Cloud Computing, haben sich auch die Anforderungen an das Netzwerk verändert. Mit dem klassischen SDN Ansatz strebt man die Trennung der Kontroll - und Datenebene an. Dabei erfolgt die Weiterleitung von Daten weiterhin durch festgelegte Regeln in Routern und Switches, doch die Steuerung findet zentralisiert statt.

SDN wird fälschlicherweise oftmals mit OpenFlow gleichgesetzt. Dass das nicht stimmt, wird unter anderem in dieser Studienarbeit aufgezeigt. Speziell möchte der Frage nachgegangen werden, ob die ganzen Marketingversprechen der grossen Hersteller eingehalten werden. Dazu stehen zwei Produkte zur Verfügung. Zum einen Cisco ACI und zum anderen VMware NSX.

Die Ergebnisse sind leicht ernüchternd. Beide Produkte bieten auf ihre Weise eine durchaus interessante Lösung, dennoch braucht es einen enormen Mehraufwand um davon zu profitieren. Eine typische SDN Lösung „von der Stange" gibt es nicht. Je nach Anforderung, muss ein anderes Produkt gewählt werden

VMware NSX punktet mit ihrer leichten Bedienbarkeit und profitiert von zahlreichen Funktionen. Leider muss weiterhin auf ein separates physikalisches Netzwerk gesetzt werden, was einen Zusatzaufwand bedeutet. Die Lösung von Cisco bietet hingegen einen vollständigen SDN Ansatz. Wobei zurzeit noch einige Abstriche in der Bedienbarkeit gemacht werden müssen.

Inhaltsverzeichnis

I. Technischer Bericht

1 Einleitung

Bis vor einigen Jahren, waren Datenspeicher, Server und Netzwerkelemente alles rein physikalische Geräte, die für sich selbst fungierten. Sie standen irgendwo zentralisiert in einem Rechenzentrum. Die Überwachungssysteme waren meist ausschliesslich auf das eigene Produkt beschränkt und hatten keinerlei Synergie. So liefen jene Applikationen wie Web-, Mail- oder Datenbankserver unabhängig ihrer Last und Gebrauch 24 Stunden separat auf dedizierter Hardware.

Der Wandel kam etwa vor 12 Jahren. Eine Firma namens VMware erfand eine interessante Technologie, welches ihr ermöglichte, mehre Betriebssysteme parallel auf einer Maschine laufen zu lassen. Dabei haben sie ein kleines Programm, den Hypervisor, entwickelt, die es ihr erlaubte, eine komplett virtuelle Umgebung zu betreiben, auf der ein Betriebssystem läuft. Die virtuellen Komponenten agierten dabei mit der realen Hardware und konnten mehrfach gemeinsam genutzt werden.

Mit dem Beginn der Servervirtualisierung, war es nicht mehr länger notwendig, ausschliesslich ein Betriebssystem auf einer Maschine zu betreiben. Die neuen Möglichkeiten der Servervirtualisierung führten dazu, dass die Ressourcen der Hardware besser ausgenutzt werden konnten. Dieser Umstand hatte ebenfalls die Hardwarehersteller dazu veranlasst, ihre Produkte den neuen Bedürfnissen anzupassen. Die bis anhin gewohnten Rack-Server in Form eines Pizzakartons hatten nicht mehr die gewünschte Leistung und Flexibilität. Neu wurde auf Bladeserver gesetzt. Die einzelnen Einschubeinheiten die mit Prozessoren, Speicher und Mainboard ausgestattet sind, werden in das Bladegehäuse eingeführt und zentral angebunden. Mit der Backplane erhalten sie alle notwenigen Ressourcen wie Strom und Netzwerkzugang, welche nun zentralisiert angeboten werden. Der Vorteil dieses Systems liegt in der kompakten Bauweise, der höheren Leistungsdichte, der Skalierbarkeit und der vereinfachten Verkabelung. Das alles führt dazu, das enorme Kosten eingespart werden konnten. Es wird weniger Platz, Strom und Kühlung benötigt.

Mit dem ständigen Fortschritt der Hypervisor-Technologie und ihren Verwaltungsmöglichkeiten, stellt es heute keine Probleme mehr da, hunderte von virtualisierten Servern zu betreiben und zu administrieren. Es besteht die zentrale Möglichkeit, die VMs zu pausieren, kopieren und über weit entfernte Strecken zu migrieren.

Ebenfalls hat sich der Speicherbereich weitgehend so weiterentwickelt, dass zentral je nach Bedürfnis Speicher zur Verfügung gestellt werden kann.

Grosse Unternehmen wie Amazon und Google, die für ihre Dienstleistungen Unmengen an Rechenpower, Speicher und Netzwerkkomponenten brauchen, sind dazu bedacht, so effizient und kostensparend zu agieren wie möglich. Der ständige Ausbau der Kapazitäten skalierte aber nicht ergonomisch wie im Storage und Computing Bereich. Die operativen Kosten fielen im Netzwerk nicht wie in den anderen beiden Bereichen. Dieser Umstand legte die Weiche für SDN, Software Defined Networking.

2 SDN - Software Defined Network

In letzter Zeit hört man immer öfters von SDN. Es wird gehypt und ist in aller Munde. Aber um was handelt es sich bei diesem Begriff genau? So eindeutig kann diese Frage leider nicht beantwortet werden. Es gibt diverse Ansätze und Visionen, welche mit SDN verfolgt werden.

2.1 Open SDN

Bei der ersten und ursprünglichen Version handelt es sich um Open SDN, basierend auf der Arbeit an der Standfort Universität, bei der OpenFlow zum Einsatz kam. Weitergeführt wurde diese Idee von der Open Networking Fundation (ONF). ONF ist eine benutzerorientierte Organisation zur Förderung und Umsetzung von SDN durch offene Standards. Solche Standards sind notwendig, um die Netzwerkbranche voran zu treiben.

Bei dieser Architektur wird die Control Plane von der Data Plane entkoppelt und ermöglicht eine programmierbare Netzwerksteuerung. Damit kann die zugrunde liegende Infrastruktur besser von den Anwendungs- und Netzwerkdiensten abstrahiert werden. Das OpenFlow Protokoll ist ein grundlegendes Element für den Aufbau von SDN-Lösungen.

In der Praxis wird noch zwischen verschiedenen Varianten Unterschieden. Distributed Control ist die bis anhin eingesetzte Methode. Dabei entscheidet jedes Gerät für sich selbst. Es findet keine zentrale Verwaltung statt. Anders ist es beim zentralisierten Kontroller. Ist einem Gerät kein Pfad für das Datenpaket bekannt, wird eine Anfrage an den Kontroller gestellt. Jegliche Logik ist ausgelagert. Eine Mischform von beiden stellt die Hybrid Variante dar. Dabei wird versucht, die Vorteile beider Varianten zu vereinen.

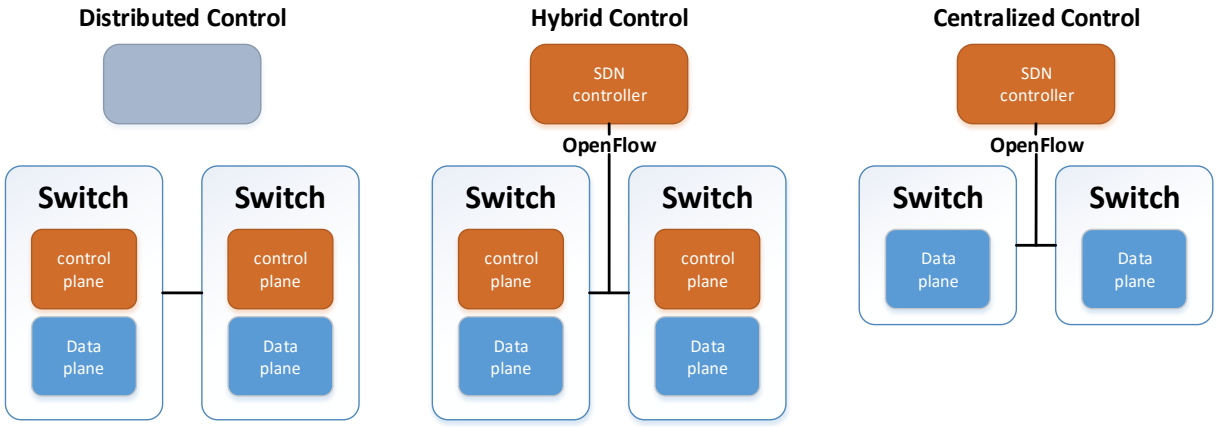


Abbildung 1: Open SDN Architektur

2.2 SDN via Overlays

Ein anderer Ansatz verfolgt beispielsweise VMware NSX mit SDN via Overlays. Dabei wird ein virtuelles Netzwerk über das physikalische Netzwerk gespannt. Durch das Erstellen von Overlay-Netzwerke auf existierenden Switches, wird ein Grossteil der Komplexität vom physischen Netzwerk getrennt, da der Traffic über Point-to-Point Verbindungen mittels Protokollen wie VXLAN und NVGRE transportiert wird. Folglich werden Aufgaben wie das Erstellen, Verschieben, Löschen oder Verändern von Netzwerkelementen vereinfacht und es kann leicht automatisiert geschehen.

2.3 White Box SDN

Ein dritter Ansatz wird mit Brite Box bzw. White Boxes SDN verfolgt. Dabei hat man sich die Frage gestellt, wieso es immer proprietäre Hardware mit proprietärer Software geben muss. Wäre es nicht einfacher, man hätte die Wahl für irgendein OS, welches beispielsweise auf einem Switch läuft? Firmen wie Pica8 und Cumulus Networks verfolgen genau diese Vision.

2.4 SDN via APIs

Eine andere Definition findet man unter SDN via APIs. Bis anhin wurde das Netzwerk via Command Line Interface (CLI) konfiguriert. Dabei war man sehr eingeschränkt und es bot einem nicht viel Spielraum, um etwas automatisiert auszuführen. Mit SNMP besteht zwar noch eine weitere Möglichkeit um das Gerät zu verwalten, jedoch mit zu geringer Leistungsfähigkeit. Mit SDN via APIs soll sich das ändern. Die Netzwerkgeräte besitzen zwar immer noch eine lokale Data- und Control-Plane, aber sie verfügen über eine verbesserte oder umfangreichere API zur Geräteverwaltung und Konfiguration.

Die Schattenseite dieser Lösung ist die Tatsache, dass die grundsätzlichen Möglichkeiten nicht verändert wurden. Es gibt weder eine Netzwerkvirtualisierung wie mit SDN via Overlays noch eine Verbesserung wie mit Open SDN.

3 Anforderungen

Das rasante Wachstum der digitalen Welt, kann mit traditionellen Netzwerktechnologien nicht mehr Schritt halten. Es fehlt an Geschwindigkeit, Flexibilität, Skalierbarkeit und einem akzeptablen Verwaltungsaufwand. Dieser Abschnitt soll auf die neuen Anforderungen in den Data Centern eingehen.

3.1 Automatisierung

Automatisierung sollte es dem Netzwerk erlauben, so flexibel wie nur möglich zu agieren. Je nach Anforderung, sollten Netzwerkelemente oder Services dynamisch instanziiert und bei Nichtgebrauch wieder gestoppt werden. Das ganze sollte schnell, effizient und mit einem minimalen menschlichen Aufwand geschehen. Es sollte auch möglich sein, bestehende Netze zu vergrössern oder zu verkleinern, je nach Bedürfnis.

3.2 Skalierbarkeit

Mit Data Centern und Cloud Umgebungen hat die enorme Anzahl an Endstationen im Netzwerk fast exponentiell zugenommen. Es braucht immer mehr Subnetze und isolierte Bereiche. Dabei kann die Limitation von MAC-Tabellen und die Anzahl an VLANs zu Problemen führen. Die Verwendung von Tunnels und virtuellen Netzwerken soll dabei Abhilfe schaffen.

3.3 Multipathing

Neben der Anforderung an die Skalierbarkeit, ist es ebenfalls notwendig, dass das Netz effizient und zuverlässig ist. Das heisst, das Netzwerk muss eine optimale Ausnutzung garantieren. Es sollten keine Pfade permanent blockiert werden, weil Protokolle wie das Spanning Tree Protokoll redundante Pfade blockiert. Ebenfalls sollte es sicher gegenüber Ausfällen sein. Falls es dennoch zu einzelnen Ausfällen kommt, muss das Netzwerk in der Lage sein, sich entsprechend anzupassen, damit kein Unterbruch auftritt. Falls eine Applikation ungeplant mehr Ressourcen braucht, soll automatisch eine Optimierung stattfinden. Sei es durch Anpassungen von Routen oder durch Priorisierung der Daten.

3.4 Multitenancy

Mit dem Fortschritt der Data Center Technologie und dem Ausbau von Cloud Computing, ist es eine Voraussetzung geworden, dass Dutzende, Hunderte oder sogar Tausende Klienten auf derselben physikalischen Plattform agieren können. Multitenancy bedeutet, dass jeder Klient im Data Center sein eigenes (virtuelles) Netzwerk betreiben und verwalten kann, als basiere es auf physikalischer Basis. Auf diese Weise sollen die Daten voneinander getrennt und geschützt werden.

3.5 Network Virtualization

Die allgemeine Idee der Virtualisierung ist, dass eine übergeordnete Abstraktion auf dem physikalischen Layer erstellt werden kann. Der Ausbau von Server- und Storagevirtualisierung hat ebenfalls das Bedürfnis von Virtualisierung im Netzwerkbereich geweckt. Mit der Virtualisierung, sollte der Netzwerkadministrator neu in der Lage sein, ein Netzwerk jederzeit und überall zu erstellen und anzupassen. Dabei sollte es der übergeordneten Abstraktion egal sein, was auf der unteren Schicht passiert.

4 Use Case

Als sich Software Defined Networking immer mehr ausbreitete, gab es vermehrt Vorschläge, wie die heutigen aufkommenden Probleme in der Netzwerkwelt gelöst werden können. Um die Möglichkeiten von SDN aufzuzeigen, werden nachfolgend einige Use Cases aufgezeigt. Dabei wird Bewusst nicht einen Bezug auf bestimmte Hersteller oder Technologien gemacht. Es soll lediglich eine Gegenüberstellung zwischen traditionellen Lösungen und Lösungen mit SDN Ansätzen aufzeigen.

4.1 Network Management

Unter Network Management versteht man die Verwaltung, Betreibung und Überwachung eines Datennetzwerks. Die ISO hat dafür ein Modell entwickelt, um die wichtigsten Tätigkeiten abzudecken. Mithilfe des FCAPS-Modells, werden folgende Aufgaben beschrieben: Fault Management, Configuration Management, Accounting Management, Performance Management und Security Management. Um all diese Bereiche abzudecken, werden die verschiedensten Arten von Tools eingesetzt. Das ist Notwendig, weil bis anhin eine Gesamtübersicht fehlte. Jedes Netzwerkgerät arbeitet für sich und besitzt keine zentrale Logik. Änderung an einer Konfiguration müssen einzeln via CLI oder mit sehr eingeschränkten Skriptmöglichkeiten ausgeführt werden. Kommt es zu einem Fehler im Netzwerk, gestaltet sich das Troubleshooting als sehr mühsam und schwierig. Die Systeme müssen dabei meist einzeln untersucht werden. Daher wird wenigstens mithilfe von Kollektoren versucht, alle bedeutsamen Informationen zu sammeln und zentral aufzuarbeiten. Dabei kommen Produkte wie Nagios, Cisco Prime, Cacti und WhatsUpGold zum Einsatz, die ihrerseits SNMP, Netflow, Telnet, SSH und diverse andere Protokolle verwenden.

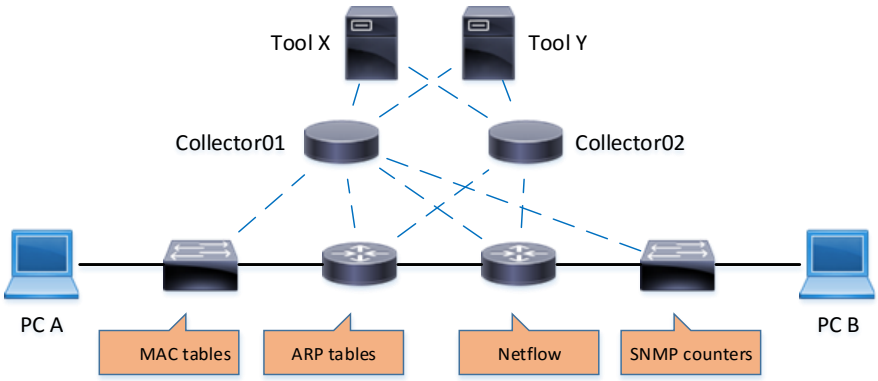


Abbildung 2: Traditionelles Network Management

Einfacher und dynamischer gestaltet sich das Network Management mit SDN. Mithilfe der Southbound API wird die Kommunikation zwischen dem SDN-Kontroller und den Netzwerkgeräten ermöglicht. Informationen und Statistiken lassen sich mittels OpenFlow relativ einfach auslesen, da jeder Flow-Eintrag einen Byte- und Packetzähler besitzt. Damit kann ebenfalls ein Rückschluss über den Pfad eines Datenpakets geschlossen werden. Eine End-zu-End Verbindung lässt sich somit transparent ermitteln.

Möchte eine Anpassung an der Konfiguration des Switches oder Routers gemacht werden, kann ein SDN Protokoll wie NetConf verwendet werden. Eine mühsame Anpassung via CLI entfällt. Die zentrale Sicht über die gesamte Umgebung erleichtert ebenfalls das Troubleshooting. Der Kontroller kann den Datenfluss zentral verifizieren und gegebenenfalls einzelne Regelwerke anpassen.

Um eine effiziente Orchestration und Automatisierung zu erhalten, kann die Northbound API genutzt werden. Die Schnittstelle soll die Ansteuerung aller relevanten Ressourcen erlauben. Im Endeffekt kann alles zentral verwaltet, betrieben und überwacht werden.

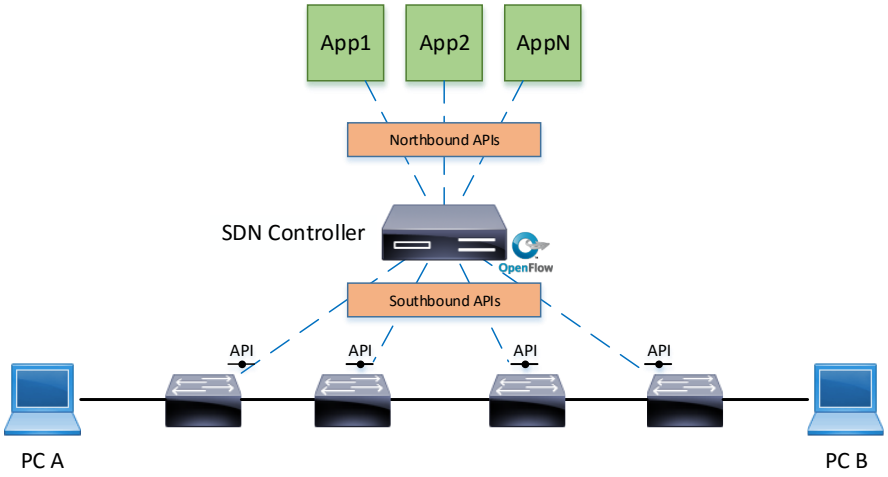


Abbildung 3: Controller-Based Network Management

4.2 Dynamic Routing

Routingprotokolle sammeln Informationen über alle Geräte und Routen und bewahren sie zentral in der Routingtabelle (RIB) auf. Werden mehrere Protokolle verwendet, werden ebenfalls unterschiedliche Tabellen erstellt. Um dabei nicht ineffizient zu werden, wird eine sogenannte Forwardingtabelle (FIB) erstellt. Die eigentliche Entscheidung wie ein Paket weitergeleitet wird, wird dann anhand der FIB vorgenommen.

Heutzutage ist die Intelligenz über die verschiedenen Routen dezentralisiert geregelt. Sprich, jeder Router berechnet für sich selbst den schnellsten bzw. besten Weg. Dabei erlauben die gängigen Protokolle wie ISIS, OSPF und EIGRP schnelle Konvergenzzeiten. Falls ein Link oder Router ausfallen würde, sind die Protokolle selbst in der Lage, das Problem zu lösen. Es wird automatisch nach einem Alternativweg gesucht und die Forwardingtabelle wird dementsprechend angepasst.

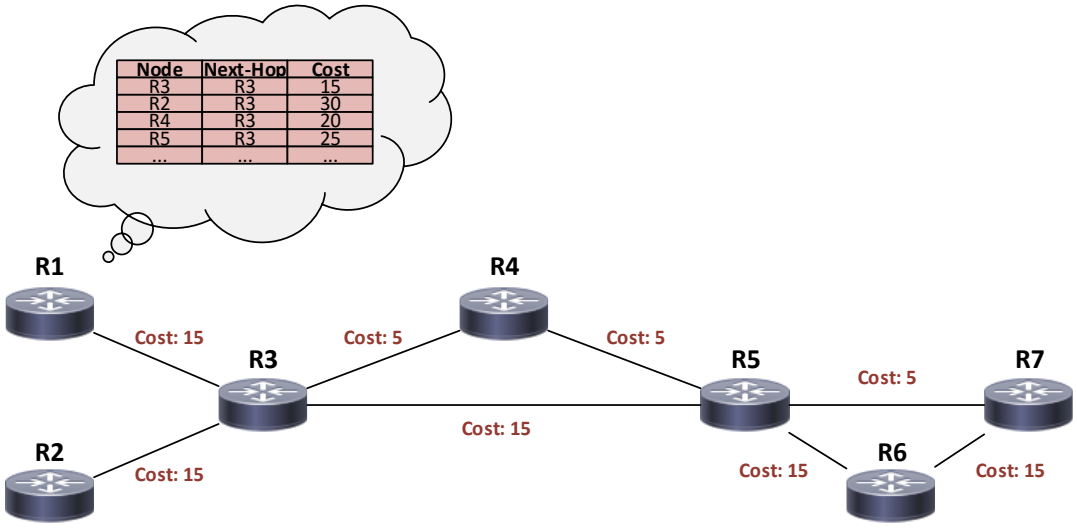


Abbildung 4: Dynamic Routing without OpenFlow

Leider weisen die heutigen Routingprotokolle einige Schwachstellen auf, die es mit der nächsten Generation zu beseitigen gilt. Dabei kommt OpenFlow ins Spiel. Mithilfe von OpenFlow soll die

komplexe Arbeit der einzelnen Control Planes zentral auf dem SDN-Kontroller ausgeführt werden. Damit kann bereits die Auslastung der CPU von den einzelnen Routern massiv gesenkt werden. Der Kontroller verfügt ausserdem über eine Echtzeitdarstellung des Netzwerks bezüglich Topologie, Störungen, Einstellungen, Performance und Kapazitäten. Diese Daten können zusammengefasst und einer modernen Netzwerkapplikation via API zur Verfügung gestellt werden. Die Applikation hat das Potenzial, den besten Pfad je nach Datenfluss zu bestimmen. Es können je nach Auslastung dynamisch Alternativpfade gesetzt und einzelne Daten priorisiert werden.

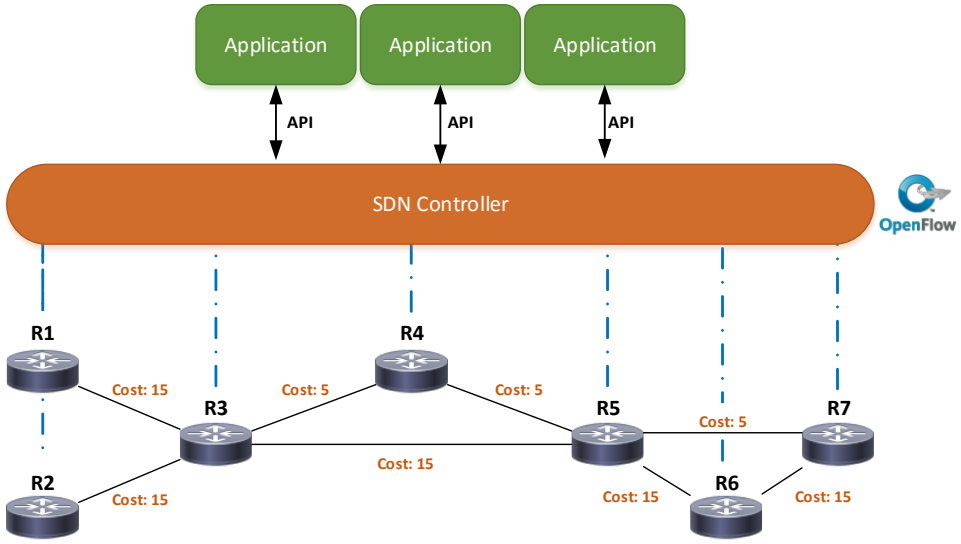


Abbildung 5: Dynamic Routing with OpenFlow

4.3 Network Access Control

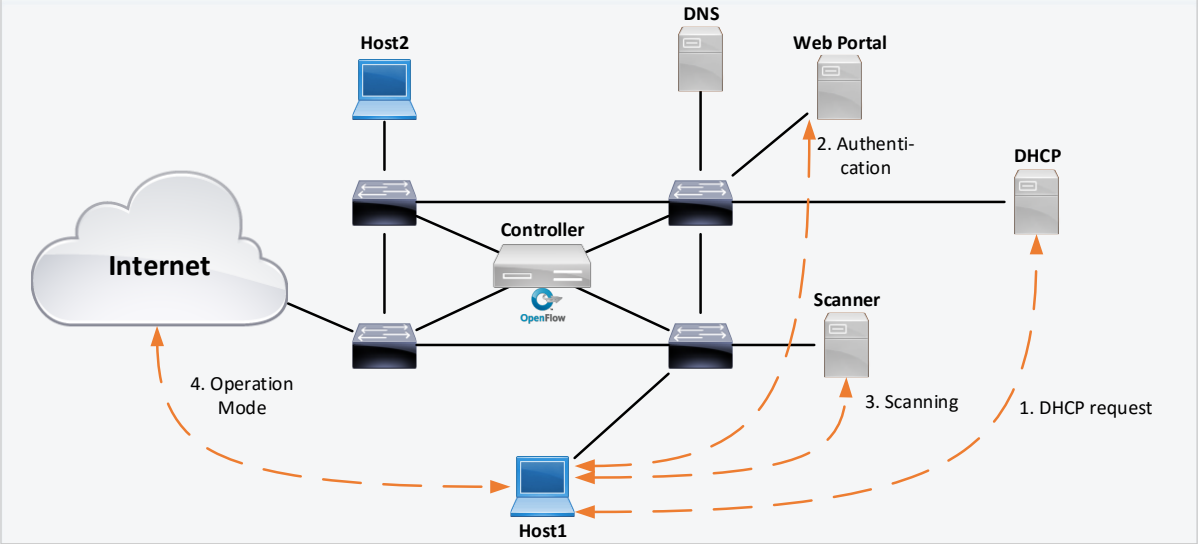
Das Ziel von Network Access Control ist die Abwehr von Viren, Würmern und unautorisierten Zugriffen. Endgeräte werden während der Authentisierung auf ihre Berechtigungen geprüft. Basierend auf den Zugriffsrechten kann anschliessend der Zugriff auf das Netzwerk gewährt werden, einschliesslich speziellen Restriktionen und Quality of Service (QoS).

Aktuelle Lösungen setzen meist auf 802.1X, MAB oder Captive Portals für die Authentifizierung der Endgeräte. Dabei kann es zu Problemen bei der Implementierung kommen. Falls der Supplicant keine 802.1X Implementierung besitzt, muss auf eine unsichere Methode wie MAB zugegriffen werden, wobei ausschliesslich anhand der MAC Adresse den Zugriff erteilt wird. Weiter erfordert die BYOD Philosophie, dass immer mehr Rücksicht auf die unterstützen Betriebssysteme wie Windows, OSX, Linux und Android genommen wird. In Endeffekt führt das oftmals zu Kompatibilitätsproblemen. Eine andere grosse Schwachstelle in der heutigen Architektur ist die Starrheit der Zugriffsteuerung. Die Regeln sind statisch, unflexibel und zu grobkörnig. Diese Probleme versucht SDN zu beseitigen. Dadurch dass jeder einzelne Datenfluss gesteuert werden kann, ermöglicht SDN ganz neue Ansätze in der Logikprogrammierung von Applikation.

Anhand von Resonance soll eine mögliche Umsetzung erläutert werden. Wobei Resonance eine Network Access Control Applikation ist, welche auf OpenFlow basiert.

Die nachfolgende Abbildung zeigt einen einfachen Aufbau mit zwei Endgeräten, vier OpenFlow Switches, vier Servern und einem SDN-Kontroller. Sobald ein neuer Benutzer das erste Mal an das Netzwerk angeschlossen wird, sendet dieser ein DHCP-Discover. Der Switch empfängt die Frames und sendet eine Anfrage an den Kontroller. Entsprechend den Richtlinien wird ein Flow-Eintrag auf

dem Switch hinterlegt und der DHCP-Service wird erlaubt. Der Benutzer wird zu einer Datenbank hinzugefügt und sein Status auf „Registration“ gesetzt. Jeglicher Datenverkehr auf TCP 80, 443 oder 8080 wird dem Web Portal Server weitergeleitet, ansonsten werden die Daten verworfen. Wird der Benutzer vom Web Portal erfolgreich authentifiziert, ändert sich sein Status auf „Authenticated“ und die Flow-Table wird aktualisiert. Der Benutzer kann jetzt vom Scanner auf fehlende Updates und Viren bzw. Würmer überprüft werden. Sofern keine Sicherheitsverletzungen auftreten, wird der Status auf „Operational“ gesetzt und seine Berechtigungen auf den OpenFlow Switches ausgerollt. Dem Benutzer ist nun möglich, gemäss seinen Rechten sich frei im Netz zu bewegen.



Registration State

Match	Action
Ethernet Type ARP (0x806)	FLOOD
UDP src_port=68 dst_port=67 (DHCP ports)	FLOOD
UDP src_port=67 dst_port=68 (DHCP ports)	FLOOD
TCP Dst port 80/443/8080	REDIRECT (to web portal: 192.168.1.3)
*	DROP

Operation State

Match	Action
*	FORWARD/DROP (according to policy lookup)

Authenticated State

Match	Action
Ethernet Type ARP (0x806)	FLOOD
Dst_IP=192.168.1.4 (Scanner's IP)	FORWARD
Src_IP=192.168.1.4 (Scanner's IP)	FORWARD
TCP dst_IP=update sites	FORWARD
TCP src_IP=update sites	FORWARD
UDP dst_port=53 (DNS port)	FORWARD
UDP src_port=53 (DNS port)	FORWARD
*	DROP

Quarantined State

Match	Action
TCP dst_port=80/443/8080	REDIRECT (to quarantine web page)
*	DROP

Abbildung 6: NAC with Resonance

4.4 QoS

Das primäre Ziel von Quality of Service (QoS) ist, die notwendige Übertragungsqualität und Serviceverfügbarkeit sicherzustellen. Dabei wird die Qualität von folgenden Faktoren bestimmt: Latency, Jitter und Packet Loss. Grundsätzlich ermöglicht QoS einen besseren Service für bestimmte Datenströme. Dies wird entweder durch das Setzen von Prioritäten oder durch das Begrenzen der Datenströme erreicht. Heutzutage ist es so, dass in einem ersten Schritt die Datenpakete klassifiziert werden müssen. Das könnte mittels Access Listen geschehen. Anschliessend wird meist eine Markierung im Header vorgenommen. Auf Layer 2 wird dabei mithilfe von 802.1Q bzw. 802.1p die Priorität gesetzt. MPLS könnte das EXP Feld manipulieren und auf Layer 3 stände das ToS Feld zur Verfügung. Durch Congestion Management Techniken kann danach der Traffic beeinflusst werden.

Die derzeitige Umsetzung hat dabei einige Schwachstellen. Damit QoS einen End-zu-End Service anbieten kann, muss jeder einzelne Knoten dafür konfiguriert werden. Ansonsten kann die QoS Policy nicht durchgängig durchgesetzt werden. Ein weiteres Problem besteht darin, dass ein Datenpaket stets die Route gemäss Routingprotokoll verfolgt. Ein policy-based Routing wird nur selten angewendet. Dafür müsste es möglich sein, alle wichtigen Parameter wie Latency und Jitter in Echtzeit zu analysieren und anschliessend dementsprechend zu handeln, beispielsweise mit Wegoptimierungen.

Genau diese Ziele verfolgt SDN. Im Falle von schlechten Reaktionszeiten oder zu hohen Jitter-Wert, kann ein Alternativpfad gesucht werden. Anschliessend informiert der SDN-Kontroller die betroffenen Router. Konkret könnte ein mögliches Szenario wie folgt aussehen: Der SDN-Kontroller stellt bei der Analyse der Routerdaten fest, dass es einen überdurchschnittlichen Anstieg an Datenaufkommen gab. Damit haben sich die Reaktionszeiten und die Paketverluste auf der ursprüngliche Routingstrecke R1-R2 bzw. R2-R3 verändert. Es kann nun nicht mehr länger garantiert werden, dass ein stabiler VoIP-Service zu Stande kommt. Als Sofortmassnahme wird dem Voice-Traffic ein Alternativpfad über eine langsamere aber stabilere Leitung angeboten. Somit kann dynamisch auf Veränderungen im Netzwerk reagiert werden.

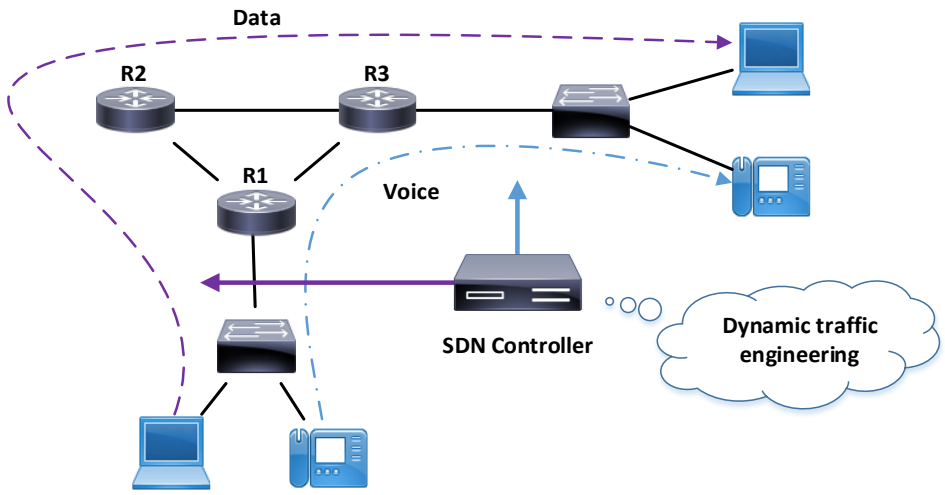


Abbildung 7: QoS with SDN

5 Anwendungsgebiete

5.1 Enterprise Campus

In den letzten paar Jahren hat sich der Druck auf Firmen und öffentliche Institute zunehmend erhöht. Die Benutzer fordern nach immer mehr Freiheit und Zugriff. Daten müssen von überall und zu jeder Zeit erreichbar sein. Dabei vermehren sich mobile Geräte wie Smartphones und Tablets rasant. Problematisch dabei ist, dass oftmals sensible Daten auf den Geräten gespeichert werden, welche nicht mal den Firmen gehören. Das Campus Netzwerk muss also nicht nur sicher, skalierbar und verwaltbar sein, sondern muss sich auch der ständig wachsenden Vielfalt an Benutzern anpassen.

Bis anhin war das Campus Netzwerk typischerweise auf 3 Layer aufgeteilt, dem Core-, Distribution- und Access-Layer. Dabei fungierte der Access Layer auf Layer 2 und der Distribution bzw. Core auf Layer 3. Diese traditionelle Architektur beinhaltete einige operationale Einschränkungen. Redundante Pfade konnten nicht immer genutzt werden und Protokolle wie Spanning Tree mussten Ports blockieren. Weiter kamen Wireless Umgebungen dazu, welche ebenfalls in das bestehende Netzwerk eingebunden werden mussten und einen Mehraufwand bedeuteten.

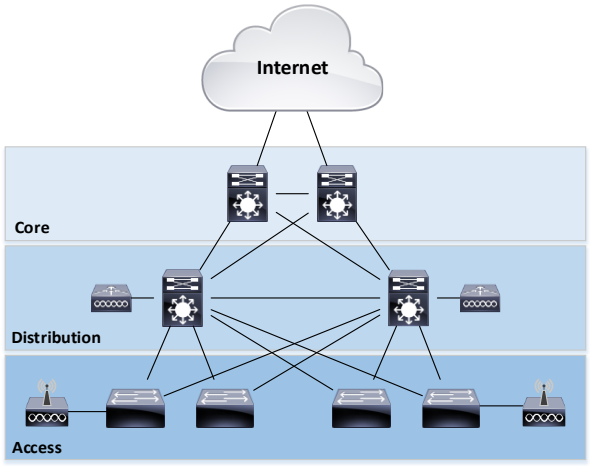


Abbildung 8: Campus Design

Um den Herausforderungen gerecht zu werden, wurden bis anhin verschiedene Methoden eingesetzt. Es können Wireless-Controller für die zentrale Verwaltung von Access-Points eingesetzt werden, Layer 3 Segmente können mittels VLANs auf den Switches isoliert werden, Access Listen werden geschrieben und VRFs werden im Routing Bereich eingesetzt um den Traffic zu separieren. All diese Lösungen sind an und für sich nicht schlecht, aber darunter leiden oftmals die Kosten, die Skalierbarkeit und die Zuverlässigkeit.

Ein OpenFlow fähiges SDN bietet einen viel einfacheren Ansatz den Datenverkehr zu isolieren und den Verwaltungsaufwand drastisch zu minimieren. Durch die Trennung der Data Plane von der Control Plane, sind beispielweise Szenarien wie nachfolgend aufgeführt schnell umsetzbar:

- Neue Services in Betrieb nehmen ohne das bestehende Netz zu beeinflussen
- Datenverkehr isolieren, egal ob auf Layer 2 oder Layer 3
- Verfügbarkeit des Netzes erhöhen
- Bessere Ressourcenauslastung

Mithilfe von SDN Richtlinien können Regeln einfacher durchgesetzt werden. Für Zugriffe oder Restriktionen von Verbindungen können Kriterien wie die Art des Zugangs oder die Zugehörigkeit des Benutzers ausgewählt werden.

Die nachfolgende Tabelle soll eine bessere Übersicht zwischen traditionellen Lösungen und Lösungen mit SDN aufzeigen. Dabei werden einige Szenarien, die es in einem Campus Netzwerk zu bewältigen gibt, beschrieben.

Use Case	Situation	traditionelle Lösung	SDN Lösung
Traffic Isolieren (Network Virtualization)	Gesetzliche Vorschriften und Firmenrichtlinien erfordern, dass bestimmter Datenverkehr im Netzwerk getrennt wird.	Um verschiedene virtuelle Netzwerke zu erstellen, werden Technologien wie VRF, VLAN und MPLS verwendet.	Entsprechend den logischen Netzen, können unterschiedliche Datenflüsse mithilfe von programmierbaren Regeln getrennt werden.
Network Access Control (NAC)	Statische Sicherheitsrichtlinien berücksichtigen nicht Echtzeitanforderungen. Die steigende Anzahl an Bedrohungen könnte im Edge Bereich zum Problem führen.	ACLs, 802.1X, MAC Authentifizierung und zusätzliche Tools zur Richtlinienumsetzung.	Mithilfe von detaillierteren Regeln, die sich je nach Benutzerkontext dynamisch verändern können, soll die Sicherheit erhöht werden. Die Richtlinien sind zudem von der physischen Hardware gelöst.
Mobilität und BYOD (Bring your own device)	Nahtlose Netzanbindung über Kabel und Wireless anbieten. Richtlinien je nach Kontext implementieren.	Limitierte Integration von WLAN Kontrollern in bestehende Switch-Umgebungen. QoS, 802.1X, Access Control und VLANs werden normalerweise eingesetzt.	OpenFlow-Fähige Switches und Access Points einsetzen. Damit können dieselben Richtlinien wie im restlichen Netz angewendet werden. Datenverkehr kann besser kontrolliert werden.
Traffic optimieren (QoS)	Applikationen und Services benötigen auf Abruf Netzwerkressourcen.	Statische QoS Richtlinien, Netzwerk gemäss prognostizierter Verkehr aufbauen.	Der Applikation erlauben, direkt via SDN Controller Einfluss auf den Datenverkehr zu nehmen.
Network Management	Geräte mit unterschiedlichen Interfaces und Konfigurationen sind schwierig zu handhaben.	CLI, Scripts, SNMP und andere MGMT-Tools verwenden. Limitierte Transparenz.	SDN-Kontroller sammelt Statistiken über alle Geräte. Ermöglicht transparente Netzwerkpfade.

Tabelle 1: Campus Use Cases

5.2 WAN

Heutzutage wird es immer wichtiger, alle Ressourcen effizient zu nutzen. Dabei wird auch kein halt bei der Optimierung des Wide Area Networks (WAN) gemacht. Der erste Schritt in die richtige Richtung wurde bereits getan. Alte Technologien wie ISDN, Frame Relay, ATM und DSL Lösungen, gehören bei uns der Vergangenheit an. Unternehmen und Institute sind heute meist über Ethernet, vorwiegend MPLS, an geographisch verteilte Netze erschlossen, die entweder direkt die Backbones der ISPs verwenden oder ansonsten das Internet. Da jedoch die Kosten für WAN Verbindungen immer noch deutlich höher liegen als im LAN Bereich, gilt es weiter zu optimieren. Die Frage ist nun, wie SDN uns dabei unterstützen kann.

Im Routing Bereich ist es immer noch so, dass die Konvergenzzeit zu hoch liegt. Fallen einzelne Verbindungen aus, sind die Auswirkungen nicht immer vorsehbar. Es werden unter Umständen nicht die optimalen Routen verwendet, weil schlichtweg die Gesamtübersicht fehlt. Ganz anders sollte es mit einem SDN Ansatz aussehen. Dadurch, dass ein SDN-Kontroller über alle Informationen und eine höhere Performance verfügt, kann schneller auf Ereignisse reagiert werden.

Anhand eines Beispiels soll konkret aufgezeigt werden, wie sich die einzelnen Lösungen bei einem Verbindungsunterbruch unterscheiden. Dabei möchte ich erwähnen, dass sich wirkliche Performancesteigerungen erst in grösseren Netzen bezahlt machen. Nichtsdestotrotz sollte das Szenario den Unterschied veranschaulichen. Dabei wird zuerst auf die Situation ohne SDN Ansatz eingegangen und anschliessend mit einem SDN Ansatz.

Angenommen der Link zwischen R5 und R7 fällt aus. Das hätte zur Folge, dass das Netz unter Umständen neu berechnet werden muss. Im Falle von EIGRP wäre das zwar nicht nötig, da es eine Feasible Successors Route gäbe, aber EIGRP ist proprietär und kann nicht überall eingesetzt werden. Daher wird in diesem Beispiel von einer OSPF Area ausgegangen, die mit Standard Timern ausgestattet ist. Nachdem der Link als Down erkannt wurde, wird ein LSA Update an den DR Router versendet, welche anschliessend alle anderen Router mit dem Update ausstattet. Nachdem der SPF Timer abgelaufen ist, fängt jeder Router an alle Routen neu zu berechnen, was sehr Zeit, CPU und Memory intensiv ist.

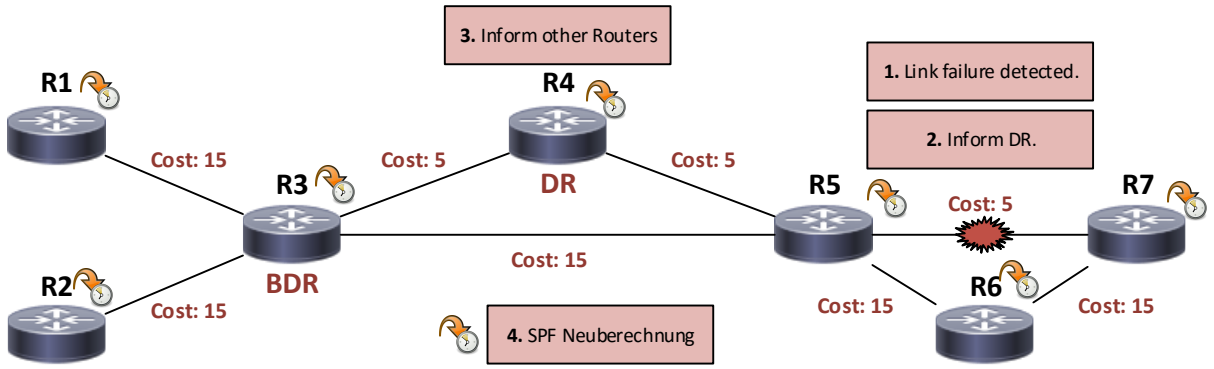


Abbildung 9: WAN without OpenFlow

Im Vergleich sehen wir ein SDN System mit einem zentralisierten Kontroller, welcher über alle Routen-, Kosten- und Bandbreitenabgaben verfügt. Die Neuberechnung einer optimalen Route kann einmalig und effizient durchgeführt werden. Anschliessend bekommen alle SDN-Fähigen Endgeräte ein Update.

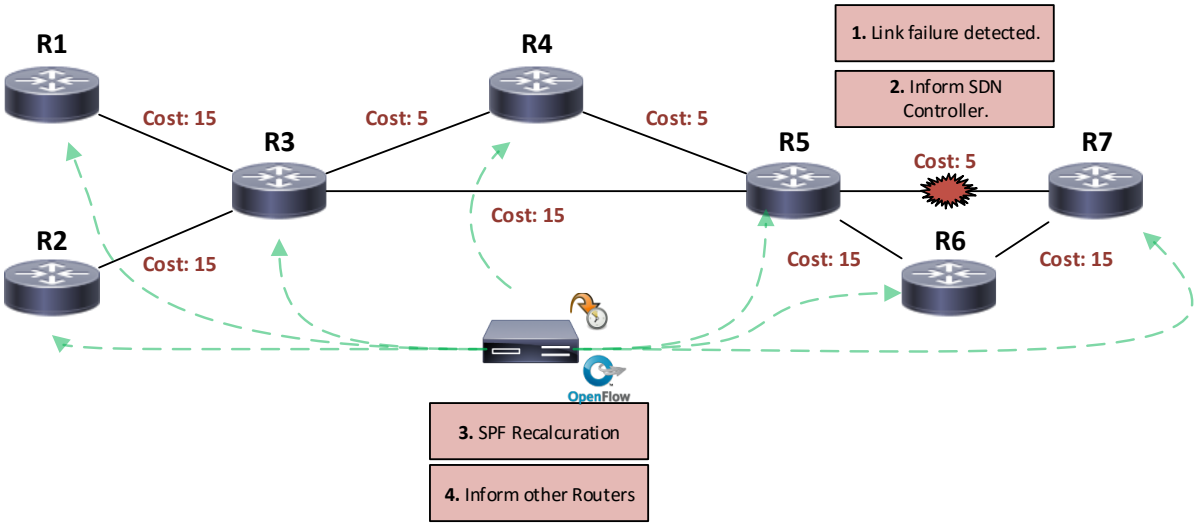


Abbildung 10: WAN with OpenFlow

5.3 Private Cloud

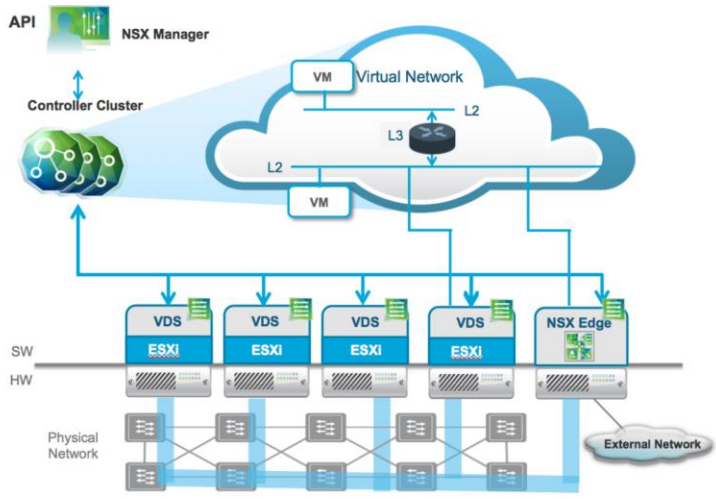
Das Bedürfnis nach einer IT-Gesamtlösung in den KMUs hat sich stark erhöht. Mit dem Ziel die komplette IT-Umgebung oder einzelne IT-Services davon auszulagern. Dabei bietet eine Private Cloud die ideale Lösung. Im Gegensatz zu einer Public Cloud, hat der Kunde die Infrastruktur für sich alleine und ist dabei von den anderen abgeschottet.

Eine Private Cloud benötigt eine Netzwerkarchitektur, die flexibel auf unterschiedlichste Anforderungen reagiert. Es braucht fein abgestufte Sicherheitsrichtlinien, flexible Netzwerkpfade, eine netzwerkweite Automatisierung und die Unterstützung für unterschiedliche Produkthersteller. Eine weitere wichtige Voraussetzung ist die Orchestrierung. Darunter wird die zentrale Steuerung der einzelnen Services und Komponenten verstanden. Es soll dem Anwender auf Knopfdruck ermöglichen, seine IT-Umgebung seinen Wünschen anzupassen.

Herkömmliche Data Center Infrastrukturen stossen dabei oft an ihre Grenzen. Aktuelle Netzwerklösungen sind starr, komplex und herstellerabhängig. Die Verfügbarkeit von neuen Clouds gestaltet sich durch die langsame Bereitstellung des Netzwerks als mühsam. Zusätzlich schränkt die physische Topologie die Platzierung und die Mobilität des Workloads stark ein. Viele Schritte müssen noch manuell getätigt werden, was die Effizienz vermindert und die Kosten in die Höhe treibt.

Um alle Vorteile einer Virtualisierung auszunutzen, muss in einem letzten Schritt ebenfalls das Netzwerk virtualisiert werden. Wie bei der Servervirtualisierung kann mit einem SDN via Overlays Ansatz, die komplette Netzwerkinfrastruktur virtualisiert werden. Logische Netzwerkelemente und Services wie Switches, Router, Firewalls, Load-Balancer, VPNs, QoS, Überwachung und Sicherheit können bereitgestellt werden. Virtuelle Netzwerke werden via API erstellt, angepasst und überwacht. Das zugrundeliegende physikalische Netzwerk wird zweitrangig und wird lediglich für den Transport der Datenpakete gebraucht. Einzelne Services können nun unabhängig an einzelne virtuelle Maschinen verteilt werden. Es ermöglicht ebenfalls die freie Platzierung der VMs im Rechenzentrum.

Das nachfolgende Bild soll eine mögliche Implementierung von SDN via Overlays verdeutlichen. Die fertige Lösung basiert dabei auf VMware NSX. Als Alternative könnte ebenso gut ein Ansatz mit OpenStack verfolgt werden.



[1]*Abbildung 11: VMware NSX Architecture Overview*

[1] VMware® NSX for vSphere (NSX-V) Network Virtualization Design Guide

6 Mitspieler auf dem SDN Markt

Mit dem Erscheinen von Software Defined Networking, haben sich die Möglichkeiten geändert, wie zukünftige Netzwerke aussehen könnten. Diese Veränderung könnte einen bedeutsamen Einfluss auf die Hardwarehersteller haben. Einige Netzwerkanbieter waren dabei ziemlich schnell, sich der SDN Bewegung anzuschliessen. Noch bevor das erste offizielle Dokument zum OpenFlow Standard erschien, haben sie den Entwicklern passende Hardware zu Testzwecken zur Verfügung gestellt. Andere haben finanzielle Mittel zur Förderung von Standards und Forschungseinrichtungen bereitgestellt. Dabei haben einige begonnen Open SDN, SDN via Overlays oder SDN-via-APIs zu unterstützen.

Praktisch jeder Anbieter von Netzwerkhardware hat heute SDN als einen Teil seiner Geschichte. Um dabei den Überblick nicht zu verlieren, ist es notwendig, einige Hersteller genauer anzuschauen. Dabei ist es nicht die Absicht, jeden Hersteller unter die Lupe zu nehmen, sondern eher bestimmte Hersteller auszuwählen, die aufgrund ihrer Grösse oder ihrem Engagement im SDN Bereich hervorstechen.

Um einen Vergleich zwischen den eher komplexen Lösungen aufzuzeigen, soll in der nachstehenden Tabelle die unterstützten Use Cases aufgezeigt werden.

	Cisco ACI	**VMware NSX**
Network Management	Der Zustand des Netzwerks wird mittels Health-Score angezeigt. Die Verwaltung der Komponenten wird zentral gesteuert.	Das Underlay Netzwerk kann mit NSX nicht überwacht oder verwaltet werden. Für das Overlay Netzwerk stehen dazu Mittel bereit.
Dynamic Routing	Das Routing innerhalb der Fabric wird automatisch von der ACI vorgenommen.	Unterstützung der gängigen Routingprotokolle. Keine erweiterte Logik.
Network Virtualization	Es wird eine Segmentierung auf Basis von Endpoint Groups angeboten.	Es wird eine Micro Segmentierung angeboten.
QoS	Wird auf L2 und L3 angeboten.	Wird auf L2 und L3 angeboten.

Tabelle 2: Feature Overview, ACI vs. NSX

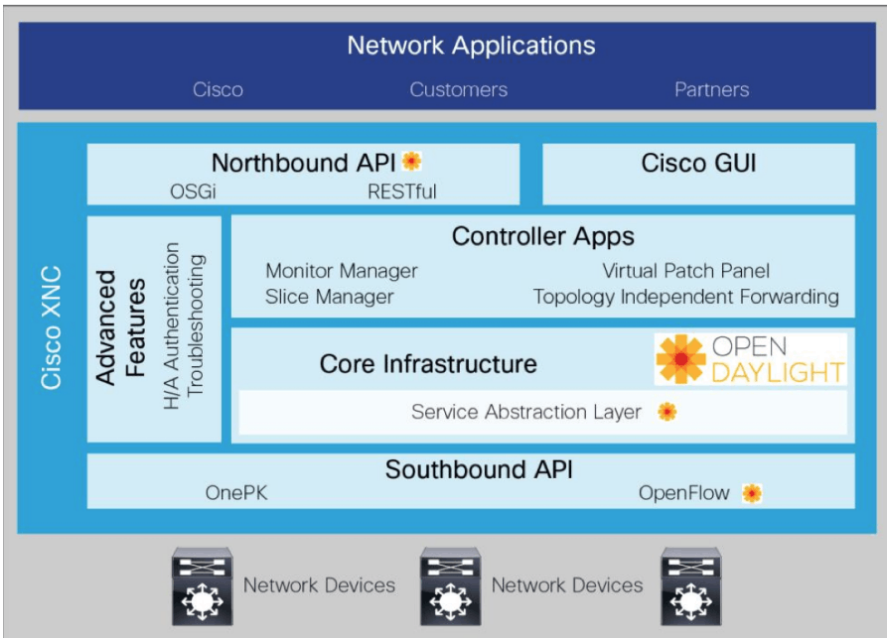
6.1 Cisco

Cisco war nicht an den frühen Arbeiten von OpenFlow involviert. Erst 2012 begann Cisco OpenFlow in seinen Geräten zu unterstützen. Im November 2013 kündete das Unternehmen mit Application Centric Infrastructure (ACI), seine erste Reaktion auf SDN an. Darüber hinaus realisierte der Konzern die Notwendigkeit, für mehr Know-How, um damit der steigenden Bedrohung durch SDN gerecht zu werden. Dafür wurde die Firma Insieme Networks aufgekauft.

Obwohl Cisco nur ein mässiger Unterstützer der Open Networking Foundation (ONF) ist, ist Cisco eine treibende Kraft hinter dem OpenDaylight Projekt, dem branchenweit führenden Open Source SDN-Kontroller.

Nebst den Open Source Bemühungen, arbeitet Cisco ebenfalls an proprietären Lösungen. Dabei stehen, ausser ACI, zwei Entwicklungen im Fokus, „Extensible Network Controller" (XNC) und „Open Network Environment Platform Kit" (onePK).

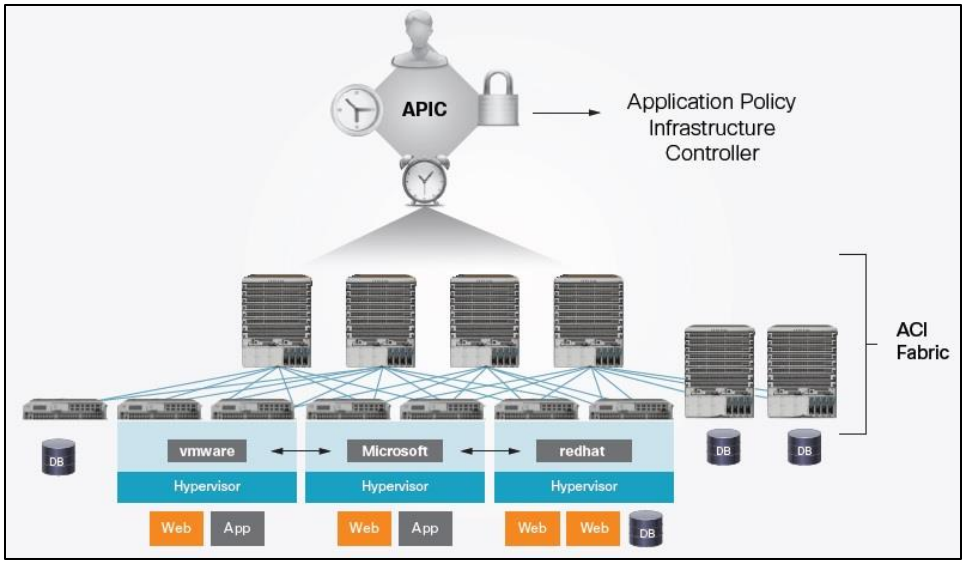
XNC ist die erste kommerzielle Version des OpenDaylight Kontrollers. Die programmierbare Netzwerksteuerung unterstützt einem dabei, das Netzwerkverhalten zu automatisieren. Auf diese Weise kann schneller auf wandelnde Anwendungsanforderungen reagiert werden. XNC wurde so entwickelt, dass mehrere Protokolle für die Gerätekommunikation unterstützt werden. Zum Release wird Openflow und onePK unterstützt. Bei onePK handelt es sich um eine SDK, die in verschiedenen Programmiersprachen bereitgestellt wird. Mit dessen Hilfe kann direkt zwischen Applikationen und Netzwerkkomponenten kommuniziert werden.



[2]Abbildung 12: XNC Controller

[2] http://www.cisco.com/c/dam/en/us/products/collateral/cloud-systems-management/extensible-network-controller-xnc/data_sheet_c78-729453.doc/_jcr_content/renditions/data_sheet_c78-729453_0.jpg

Für eine komplett anwendungszentrierte Data Center Infrastruktur bietet Cisco ACI an. Jede beliebige Anwendung sollte damit On-Demand bereitgestellt werden können. Eine komplette ACI Infrastruktur besteht aus folgenden drei Komponenten: dem Application Policy Infrastructure Controller (APIC), Cisco Switches der 9000er Serie und dem Application Virtual Switch (AVS), basierend auf einem Hypervisor.



[3]*Abbildung 13: Cisco Application Centric Infrastructure Overview*

[3] http://gblogs.cisco.com/fr-datacenter/wp-content/uploads/sites/14/2013/11/ACI-infra.jpg

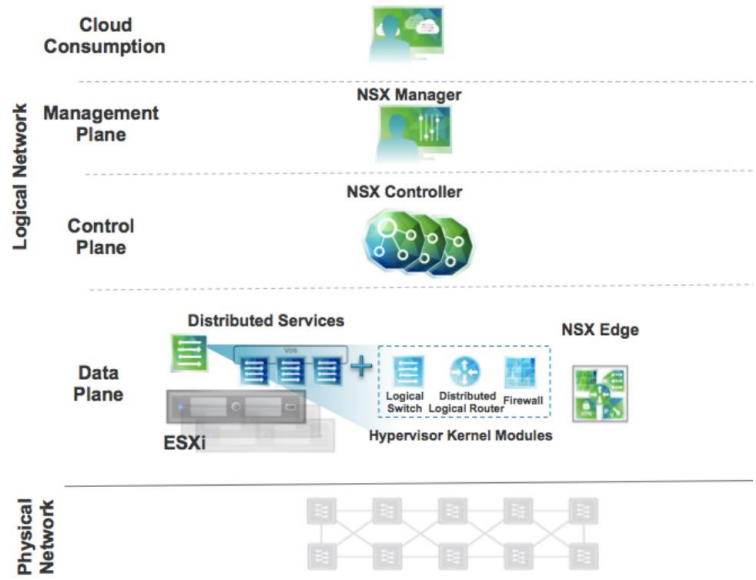
6.2 VMware

Der Schritt in Richtung Netzwerkvirtualisierung hat nicht nur die Hardwarehersteller hellhörig gemacht, sondern vermehrt auch Softwareanbieter. Nebst Microsoft und Big Switch, ist VMware einer der bedeutendsten Konkurrenten im SDN Bereich geworden. Mit dem Kauf von Nicira, konnte das nötige Wissen im Netzwerksegment, speziell im SDN- und Netzwerkvirtualisierungsbereich, aufgebaut werden. Mit diesem Schritt könnte die Firma ähnlich wie mit der Servervirtualisierung, eine dominante und wichtige Rolle auf dem Markt einnehmen.

VMware NSX ist eine softwarebasierte Netzwerk- bzw. Security-Virtualisierungsplattform. Damit lassen sich die OSI Layer 2 – 7 vollständig in Software abbilden. Das ermöglicht, dass komplexe Multi-Tier Netzwerktopologien erstellt und programmgesteuert innert Sekunden bereitgestellt werden können.

Die zwei Hauptkomponenten der VMware NSX Lösung ist einerseits der NSX Manger und anderseits der NSX Controller. Der Manager bietet eine zentralisierte Management-Ebene für das Data Center und ist fest in die vSphere Plattform integriert. Zusammen mit der Bereitstellung einer Benutzeroberfläche für die Administration und einer Management API, installiert der NSX Manger diverse VIBs (vSphere Installation Bundle) für die Host Erstellung. Das beinhaltet VXLAN, Distributed Routing, Distributed-Firewall und einen speziellen Benutzer. Der Vorteil der Nutzung einer VMware Lösung ist der einfachere Zugang zum Kernel. Damit kann die Firewall- und Routingfunktion bereits im Kernel realisiert werden.

Der NSX-Controller ist der Steuerpunkt für alle logischen Switches im Netzwerk und verwaltet Informationen aller virtuellen Maschinen, Hosts, logische Switches und VXLANs.



[4]*Abbildung 14: NSX Components*

[4] VMware® NSX for vSphere (NSX-V) Network Virtualization Design Guide

7 Evaluationskatalog

Mit dem Trend in Richtung SDN kommt auch vermehrt die Frage auf, wie man bei der Evaluierung einer passenden SDN-Lösung vorgehen soll. Jedes Unternehmen hat persönliche Bedürfnisse und eine eigene Ausgangslage. Umso wichtiger ist es, sich in einem ersten Schritt einige Gedanken zu machen. Folgende Punkte können bei der Lösungsfindung helfen.

7.1 Möglichkeiten

Bevor irgendeine Lösung genauer betrachtet wird, sollte zuerst bestimmt werden, was überhaupt gewünscht wird. Welche Aufgaben möchte man mit einem SDN Ansatz verbessern. Es gibt ein breites Spektrum was mit SDN abgedeckt wird. Möchte man Netzwerkressourcen besser ausnutzen, das Netzwerk einfacher skalieren lassen, die Komplexität verringern oder CAPEX-Kosten minimieren. Je nach Fokus muss eine Reihenfolge bestimmt werden. Das hilft herauszufinden, welcher Anbieter die persönlichen Bedürfnisse am besten unterstützt.

7.2 Realität erkennen

Um kompetent die Hersteller oder Produkte zu evaluieren, braucht es ein fundiertes Verständnis von SDN. Damit kann zwischen Hype und Realität getrennt werden. Der Markt verändert sich ständig und entwickelt sich rasant weiter. Dabei werden viele Versprechungen gemacht. Daher ist es wichtig zu erkennen, welche Use Cases konkret abgedeckt werden.

7.3 Top-Level Analyse

Nach den ersten Analysen könnte man Anbieter in Bezug auf folgende Punkte genauer untersuchen:

7.3.1 Architektur

- Welche Komponenten werden bei der Lösung zur Verfügung gestellt.
- Welche Komponenten müssen bei einem Drittanbieter bezogen werden.
- Wie kann die SDN-Lösung, nebst im Data Center, eingesetzt werden.
- Welche Protokolle werden (vom SDN-Kontroller) unterstützt/genutzt.
- Welche Aufgaben werden vom Kontroller übernommen.
- Wie wird die Hochverfügbarkeit des Systems sichergestellt.
- Welche Möglichkeiten werden von der API zur Verfügung gestellt.
- Welche Programmiersprachen werden von der API unterstützt.
- Wie lässt sich die Architektur skalieren.

7.3.2 Integration

- Wie lässt sich die SDN-Lösung in das bestehende Netzwerk integrieren.
- Wie werden Daten zwischen der SDN Lösung und dem traditionellen Netzwerk ausgetauscht.
- Kann das SDN Management in das bestehende Management integriert werden.

7.3.3 Management

- Welche Fähigkeiten müssen die Administratoren besitzen.
- Kann der Verwaltungsaufwand minimiert werden gegenüber heutigen Herausforderungen.
- Können bestehende Managementlösungen abgelöst/integriert werden.

7.3.4 Security

- Welche Funktionen werden angeboten um die Sicherheit zu erhöhen.
- Erhöht die Lösung die Gesamtsicherheit der IT Infrastruktur.

8 Detaillierte Anforderungen

Bevor die Auflistung der detaillierten Anforderungen erstellt werden konnte, musste definiert werden, welche Aspekte mit SDN abgedeckt werden sollten. Für meine Analyse wurde davon ausgegangen, dass eine SDN Umgebung für einen Private Cloud Anbieter erstellt werden sollte. Dabei war es besonders wichtig, dass die Lösung mandantenfähig ist. Die genaueren Anforderungen sind dem nachfolgenden Katalog zu entnehmen.

8.1 Automatisierung

- Das Implementieren von Multi-Tier Services benötigt keine manuelle Konfiguration am physikalischen Netzwetzwerk.
- Das Implementieren von Multi-Tier Services kann innert wenigen Minuten, max. 10 Minuten, anstelle von Tagen erfolgen.
- Es lassen sich Templates für widerkehrende Mandantensetups erstellen.
- Änderungswünsche an einem bestehenden Multi-Tier Service, zieht keine manuelle Konfiguration am physischen Netzwerk nach sich.
- Es lassen sie ganze Multi-Tier Services auf andere Cluster verschieben, ohne dass manuelle Anpassungen am System vorgenommen werden müssen.

8.2 Skalierbarkeit

- Der Kontroller muss mindestens 100‘000 MAC Adressen verwalten können.
- Der Kontroller muss mindestens 5000 Mandanten handhaben können.
- Der Kontroller muss mindestens 50‘000 Mikro Segmente logisch voneinander trennen können.
- Der Ausbau des Netzwerks durch zusätzliche Links zwischen Leaf und Spine soll jederzeit im laufenden Betrieb möglich sein.
- Der Ausbau des Netzwerks durch einen zusätzlichen Leaf Switch soll jederzeit im laufenden Betrieb möglich sein.
- Der Ausbau des Netzwerks durch einen zusätzlichen Spine Switch soll jederzeit im laufenden Betrieb möglich sein.
- Der Datenverkehr kann für die gewünschten Services Clusterübergreifen priorisiert werden.
- Ausgelastete Services können automatisch neue Instanzen initialisieren.
- Services sind elastisch und lassen eine dynamische Erhöhung von Ressourcen wie CPU und RAM zu.
- Es wird mindestens ein North-South Datendurchsatz von 10 Gbit/s gewährleistet.
- Es wird mindestens ein East-West Datendurchsatz von 40 Gbit/s gewährleistet.

8.3 Mobility

- Das Verschieben einer VM Instanz mit 4GB RAM und zentralisiertem Datenspeicher darf nicht länger als 120 Sekunden dauern.
- Das gleichzeitige Verschieben von vier VM Instanzen mit jeweils 4GB RAM und zentralisiertem Datenspeicher darf nicht länger als 10 Minuten dauern.
- Die Migrationsdauer einer VM Instanz innerhalb eines Racks oder Rack übergreifend darf sich nicht um mehr als 10% unterscheiden.

- Durch Funktionen wie vMotion oder Distributed Resource Scheduler (DRS), wie man es von VMware kennt, wird keine manuelle Anpassung im Gesamtsystem gebraucht. Das bedeutet, dass jegliche Parameter und Policies automatisch mitverschoben werden.
- Mithilfe einer Ethernet-Fabric oder einem Overlay Netzwerk, kann die Mobility einer VM-Instanz gewährleistet werden.

8.4 Multipathing

- Es muss eine gewisse Entropie (für die Berechnung des Hash Wertes) gewähreistet werden, damit VXLAN Traffic nicht ausschliesslich dieselben Netzwerkpfade verwenden.
- Es werden keine Pfade durch das Spanning Tree Protokoll blockiert.
- Auch wenn Instanzen zwischen verschieden Cluster/Data Center verschoben werden, wird weiterhin der optimale Pfad verwendet.
- Equal-Cost Load Sharing sollte sowohl im Underlay - sowie im Overlay-Netzwerk unterstützt werden.

8.5 Multitenancy

- Die Anzahl der Mandanten soll mittels Tunneling Techniken über 5000 betragen können.
- Mandanten können ihre private IP-Adresse durch NATing in eine öffentliche IP-Adresse umwandeln.
- Unterschiedliche Mandanten können die gleiche IP-Adresse und VLAN-ID verwenden.
- Einzelne VM Instanzen können, ähnlich wie mit private VLAN, voneinander getrennt werden.
- Getrennte Mikro Segmente dürfen keinen Broadcast Traffic von anderen Mikro Segmente n empfangen.
- Der Kontroller muss mindestens 50‘000 Mikro Segmente logisch voneinander trennen können.
- Mandanten können gemäss PCI-DSS Bestimmungen logisch voneinander getrennt werden.
- Die Control Plane muss nicht explizit von anderen Mandanten getrennt sein.

8.6 Network Services

- Traditionelle Protokolle wie IGP und BGP werden unterstützt.
- Es wird IPv6 für Endgeräte unterstützt.
- Ein starker Anstieg an Broadcast- oder Multicast-Nachrichten sollte erkannt und unterdrückt werden.
- Die Kommunikation zwischen Kontroller und Netzwerkgeräten muss verschlüsselt sein.
- Fibre Channel over Ethernet (FCoE) wird unterstützt.
- iSCSI wird unterstützt.
- Für Firewall-Services braucht es keine Zusatzprodukte.
- Es lassen sich problemlos virtualisierte Firewalls von Drittherstellern einbinden.
- Für Load Balancer-Services braucht es keine Zusatzprodukte.
- Der Health-Check eines Load Balancers muss mindestens folgende Arten unterstützen: ICMP, HTTP, HTTPS, TCP, UDP.
- Es lassen sich problemlos virtualisierte Load Balancer von Drittherstellern einbinden.
- Layer 3 Funktionen lassen sich virtualisieren.

8.7 Management

- Ein Ausfall eines einzelnen Knotens darf keinen Einfluss auf den laufenden Betrieb haben.
- Die wichtigsten Komponenten können hochverfügbar realisiert werden.
- Der SDN Kontroller muss in der Lage sein bei Performanceproblemen eine Benachrichtigung abzusetzen.
- Es wird ein Reporting über Performancedaten angeboten. Diese müssen mindestens eine Zeitperiode von 3 Monaten umfassen.
- Änderungen im Netzwerk werden rapportiert.
- Die Orchestration ist zentral mit einer einzigen Applikation möglich.
- Der Data Flow von Endgeräten kann nachverfolgt werden.
- Es kann eine Statistik über die Auslastung pro Kunde erstellt werden.
- Es kann eine Statistik über die Auslastung pro Service erstellt werden.
- Die Komplette SDN Konfiguration sollte ein Backup/Restore zulassen.
- Nicht mehr gebrauchte Firewallregeln sollten automatisch deaktiviert/gelöscht werden.
- Es wird eine API für die Automatisierung zur Verfügung gestellt.
- Mithilfe der API kann ein Multi-Tier Service vollständig aufgesetzt werden.
- Mithilfe der API können vorgefertigte Profile/Templates benutzt werden.
- Die API deckt alle gängigen Funktionen der Orchestrationssoftware ab.
- Es gibt eine gute Troubleshootingunterstützung.
- Komplette Umgebungen können visualisiert dargestellt werden.
- Fehlgeschlagene Updates an den SDN Komponenten dürfen zu keinem längeren Totalausfall führen (max. 30 Minuten). Es muss ein Rollback/Recovery Mechanismus angeboten werden.
- Die Administration der SDN Umgebung soll für IT-Leute mit einer 3 tägigen Schulung gut machbar sein.

9 VMware NSX

9.1 Infrastruktur

Bevor VMware NSX implementiert werden kann, müssen einige Voraussetzungen erfüllt sein. Das beinhaltet die komplette Bereitstellung der Infrastruktur samt passenden Vorkonfigurationen. Nach Best-Practice sollten die einzelnen Rollen in unterschiedliche vSphere Cluster aufgeteilt werden:

- **Compute Cluster**: ESXi-Cluster für die virtuellen Maschinen (3-Tier Applikation)
- **Management Cluster**: Darauf sollten alle benötigten Managementkomponenten wie NSX Manager, NSX Controller und vCenter platziert werden.
- **Edge-Cluster:** Darauf werden die NSX Services wie „NSX-Edge" oder „Logical Router" betrieben.

Das mir in meinem LAB nicht unbegrenzte Ressourcen zur Verfügung standen, wurde mein Setup wie folgt gewählt.

- 1x Computer Cluster, bestehend aus ESXi01 und ESXi02 (3-Tier Applikation).
- 1x vCenter Appliance (ESXi03) für die Verwaltung aller ESXi-Hosts.
- 1x Management & Edge Cluster, bestehend aus ESXi03 und ESXi04.
- 1x Storage Cluster für den zentralen Datenspeicher aller VMs.
- 1x physikalischer Layer 3 Switch
- 1x physikalischer Router

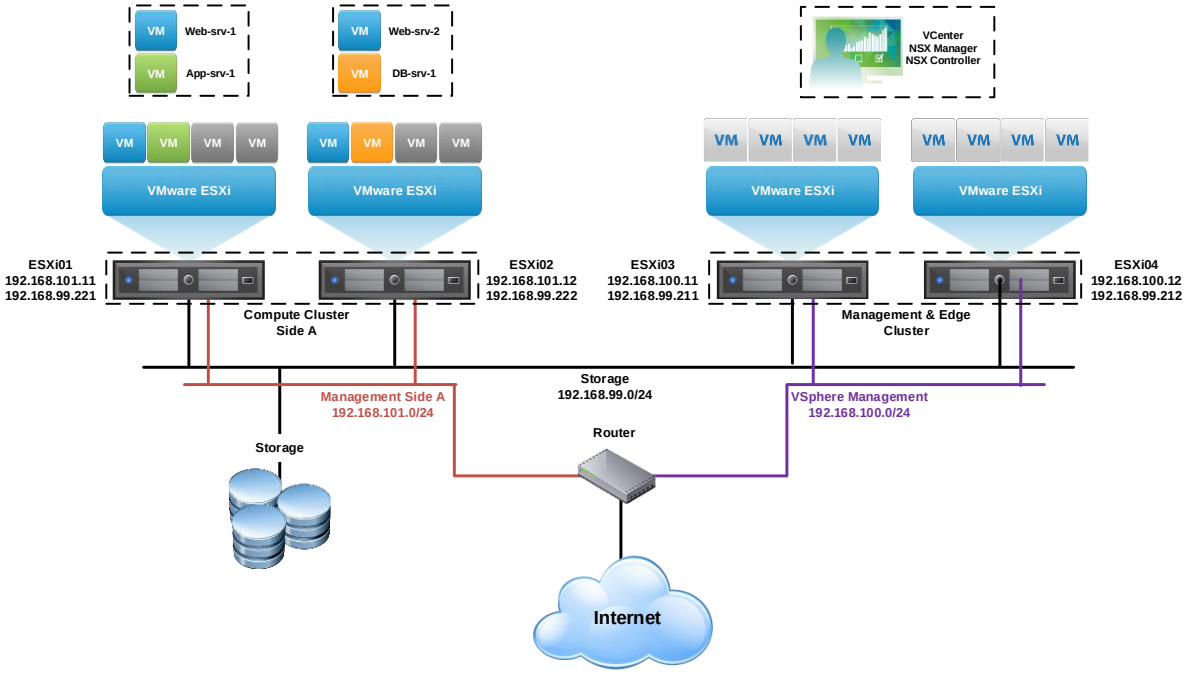
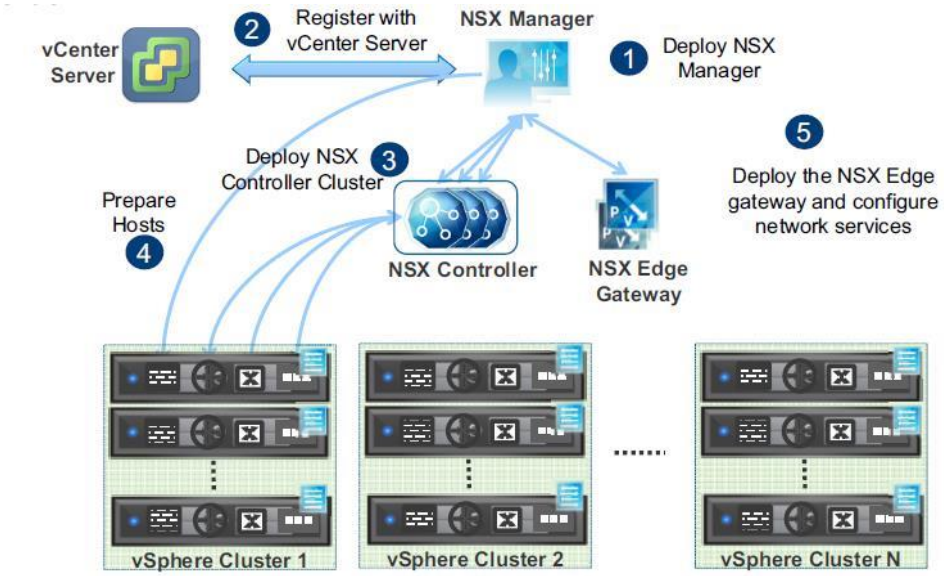


Abbildung 15: Infrastructure VMware NSX

9.2 Kurzübersicht NSX Installation

Die Installation von NSX beinhaltet die Bereitstellung von mehreren virtuellen Systemen, einigen Vorbereitungen an den ESXi-Host sowie Konfigurationsanpassungen, um die Kommunikation zwischen allen physischen und virtuellen Geräten zu ermöglichen.



[5]*Abbildung 16: NSX Installation Steps Sequence*

Der Prozess startet mit der Bereitstellung einer OVF-Vorlage für den NSX Manager. Nachfolgend müssen NSX Manager und vCenter miteinander verknüpft werden. Dabei besteht eine 1:1 Beziehung, sprich jede vCenter Instanz kann genau eine NSX Manger Instanz beinhalten. Der Registrierungsvorgang ermöglicht die Bereitstellung eines NSX Controller Clusters. Die NSX Controller werden ebenso wie der NSX Manger als virtuelle Appliance ausgeführt. Daraufhin erfolgt die Vorbereitung der ESX-Hosts für NSX. Es werden mehrere VIBs auf den einzelnen Hosts installiert. Diese VIBs ermöglichen VXLAN-Funktionalität, verteiltes Routing und Distributed Firewall. Nachdem Transport Zonen und VXLAN mithilfe von Distributed Switch (VDS) eingerichtet wurden, kann der Aufbau der NSX-Overlay-Topologie beginnen.

[5] http://www.vmwarearena.com/wp-content/uploads/2015/01/NSX-Installation-Order-of-Tasks.jpg

9.3 Lab Design

Als Testszenario soll eine Multi-Tier Applikation erstellt werden, bestehend aus zwei Web-, einem Applikation- und einem Datenbankserver. Für die Lastverteilung der Demowebseite soll ein Load Balancer im Round-Robin Modus eingesetzt werden. Die einzelnen Bereiche werden durch passende Massnahmen voneinander getrennt, sodass Zugriffe von aussen lediglich auf die Webserver gelangen. Weiter sollen nur direkt benachbarte Segmente auf bestimmten Ports miteinander kommunizieren können. Die Netze werden durch einen Distribution Logical Router miteinander verbunden und gelangen durch den NSX Edge Node in die Aussenwelt.

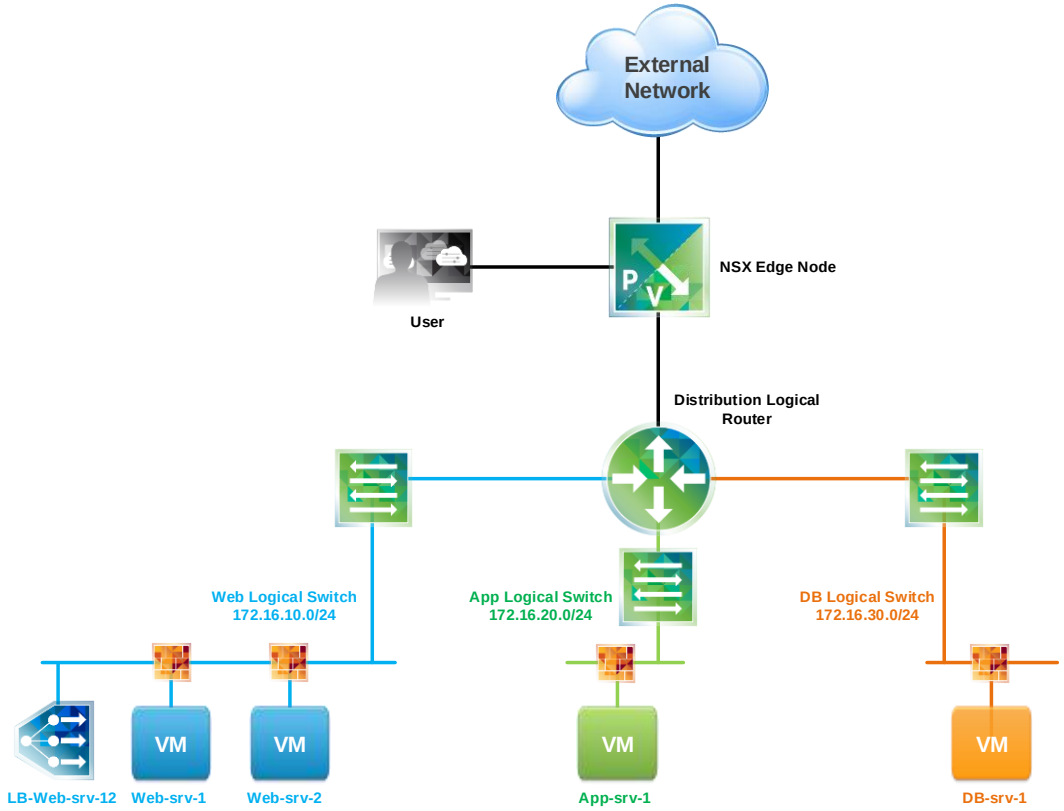


Abbildung 17: Multi-Tier Application

9.4 Lab Konfiguration

Für die Konfiguration der gesamten NSX Umgebung wurde auf die offizielle VMware NSX 6.2 Dokumentation zurückgegriffen. Daher wird auf eine detaillierte Installationsanleitung verzichtet. Vollständigkeitshalber werden die einzelnen Schritte aufgelistet, jedoch werden nur spezifische Einstellungen und Abweichungen von der Norm genauer erläutert.

9.4.1 ESXi-Hosts Vorbereitung

Bevor mit der NSX Installation begonnen wird, muss die Basiskonfiguration im vCenter für das Netzwerk vorgenommen werden. Alle ESXi-Host verfügen über zwei physikalische Netzwerkschnittstellen die wie folgt konfiguriert sind:

ESXi Host	NIC	VMkernel Adapter	VMkernel Name	Switch	VLAN ID	IP Address
ESXi01	vmnic0	vmk0	Management Network	vSwitch0	101	192.168.101.11
ESXi01	vmnic0	vmk1	Storage	vSwitch0	99	192.168.99.221
ESXi01	vmnic1	vmk2	Compute_VDS - Mgmt	Compute_VDS	101	-
ESXi02	vmnic0	vmk0	Management Network	vSwitch0	101	192.168.101.12
ESXi02	vmnic0	vmk1	Storage	vSwitch0	99	192.168.99.222
ESXi02	vmnic1	vmk2	Compute_VDS - Mgmt	Compute_VDS	101	-
ESXi03	vmnic0	vmk0	Management Network	vSwitch0	100	192.168.100.11
ESXi03	vmnic0	vmk1	Storage	vSwitch0	99	192.168.99.211
ESXi03	vmnic1	vmk2	Mgmt_VDS - Mgmt	Mgmt_VDS	100	-
ESXi04	vmnic0	vmk0	Management Network	vSwitch0	100	192.168.100.12
ESXi04	vmnic0	vmk1	Storage	vSwitch0	99	192.168.99.212
ESXi04	vmnic1	vmk2	Mgmt_VDS - Mgmt	Mgmt_VDS	100	-

Tabelle 3: NIC Settings ESXi-Hosts

Es existieren zwei Distributed Switches (VDS), welche Host übergreifend zur Verfügung stehen. Diese werden später für VXLAN gebraucht.

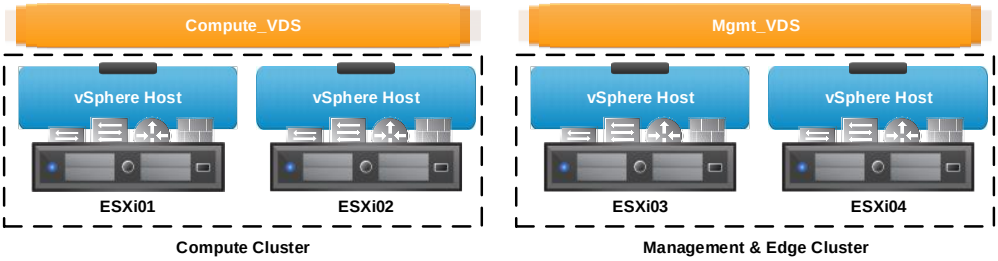


Abbildung 18: Distributed Switches

9.4.2 Installieren der virtuellen NSX Manager-Appliance

Standardinstallation gemäss Anleitung. Die Appliance wir auf dem Host ESXi03 an den „Mgmt_VDS“ Switch angebunden und verwendet „Mgmt_VDS – Mgmt“ als Distributed Port Group. Das Management wird unter der IP-Adresse 192.168.100.20 erreichbar sein.

9.4.3 Registrieren von vCenter Server mit NSX Manager

Standardkonfiguration gemäss Anleitung.

9.4.4 Bereitstellen von NSX Controller

Ein Kontroller Cluster, bestehend aus 3 aktiven virtuellen Appliances, ist einer der Kernkomponenten von NSX. Es ist die Control Plane und beinhaltet die MAC-, ARP- und VTEP-Tabelle. Die drei Knoten sollten am besten auf drei verschiedene Hosts instanziiert werden. Da mir lediglich zwei Hosts zur Verfügung stehen, werden jeweils zwei Instanzen auf ESXi03 und eine auf ESXi04 ausgeführt. Gleich wie der NSX Manager, werden die Kontroller in die „Mgmt _VDS – Mgmt" Port Group eingebunden und erhalten für das Management einen IP-Pool im Bereich 192.168.100.30 – 192.168.100.33.

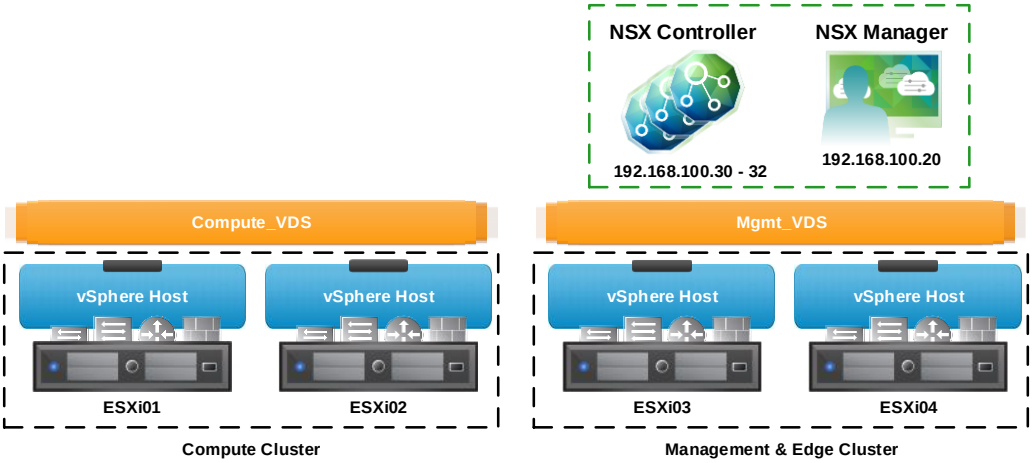


Abbildung 19: NSX Manager & Controller

9.4.5 Vorbereiten von Host-Clustern für NSX

Standardkonfiguration gemäss Anleitung. Dabei werden die benötigten VIBs auf die ESXi Hosts installiert. Das wird jeweils für den Compute- und Management/Edge-Cluster getan.

9.4.6 Konfigurieren von VXLAN-Transportparametern

VXLAN wird pro Cluster konfiguriert, wobei jedem Cluster, der an NSX teilnimmt, einem vSphere Distributed Switch (VDS) zuordnet wird. Es wird jeweils ein VXLAN Netz pro Cluster angelegt. Der Compute Cluster erhält einen IP Pool im Bereich 192.168.168.250.100 – 192.168.250.110, welches sich im VLAN 250 befindet. Für den Management Cluster ist der IP Pool 192.168.150.100 – 192.168.150.110 im VLAN 150 vorgesehen.

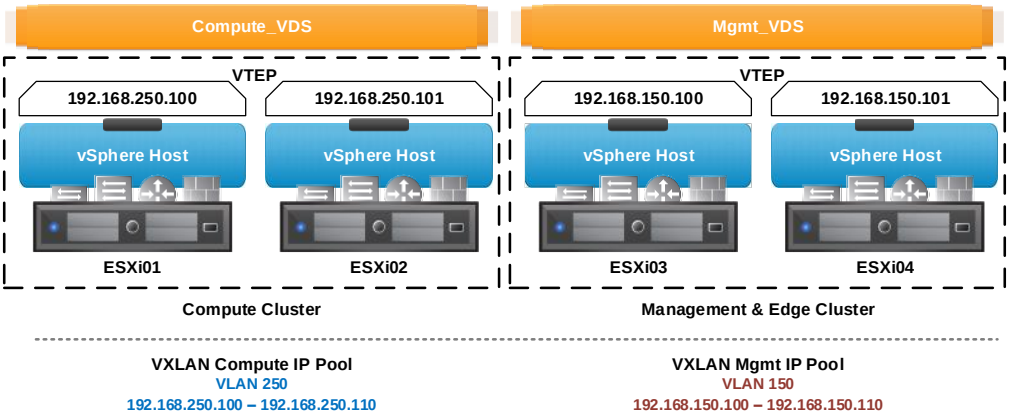


Abbildung 20: VXLAN Transport

9.4.7 Zuweisen des Segment-ID-Pools und des Multicast-Adressbereichs

Standardkonfiguration gemäss Anleitung. Segment ID Pool liegt im Bereich 5000 – 5999.

9.4.8 Hinzufügen einer Transportzone

Die Transportzone kann einen oder mehrere vSphere Cluster umfassen, dabei steuert sie, welche Hosts ein logischer Switch erreichen kann. Für meine Testumgebung wird eine globale Transportzone über beide Cluster erstellt. Dabei wird der Unicast Mode verwendet.

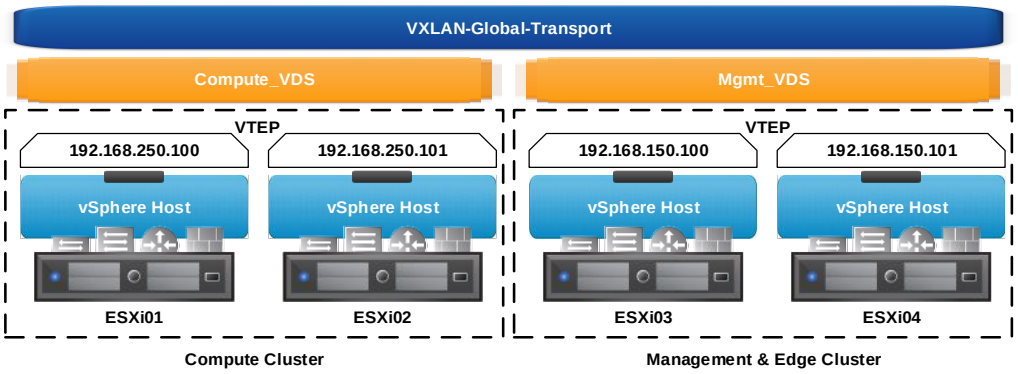


Abbildung 21: Transport Zones

9.4.9 Hinzufügen eines logischen Switches

Ein logischer Switch erstellt eine Broadcast Domäne bzw. Segment, in der virtuelle Maschinen logisch miteinander verbunden sind. Die VMs können dann über VXLAN miteinander kommunizieren. Jeder logische Switch hat eine Segment-ID, was nichts anderes als ein VXLAN Network Identifier (VNI) ist. Damit lassen sich bis zu 16 Millionen Segment-IDs erstellen. Für mein Lab werden vorerst 3 logische Switches erstellt um die VMs der 3-Tier Applikationsarchitektur einzubinden.

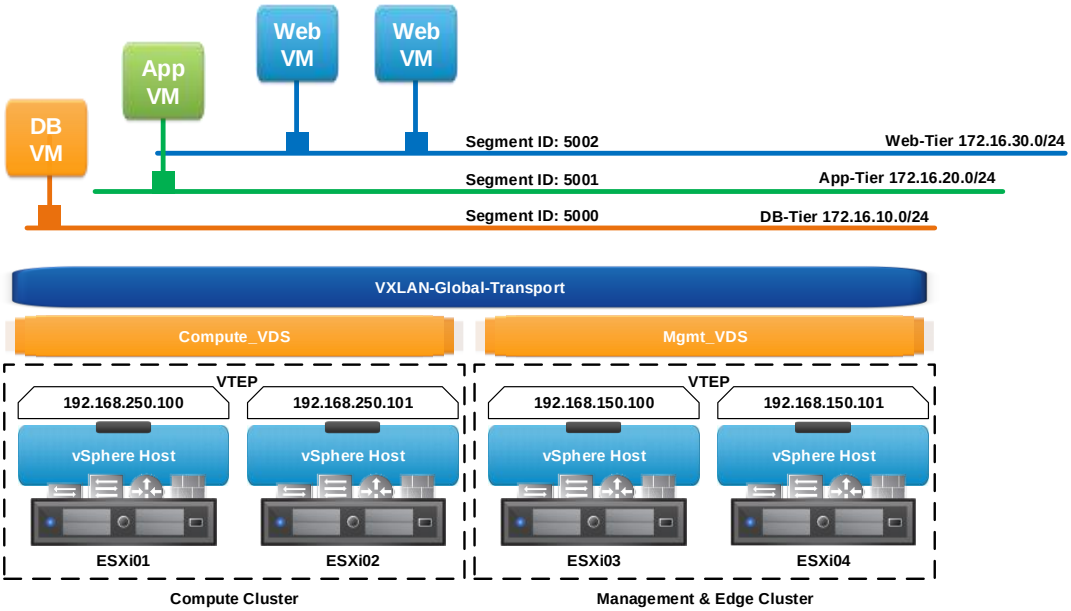


Abbildung 22: Logical Switches

9.4.10 Hinzufügen eines Distributed Logical Routers

Auf der virtuellen Appliance wurden 3 Interfaces definiert um alle zuvor erstellten logischen Switches anzubinden. Die erste verfügbare Adresse dient dabei als Gateway.

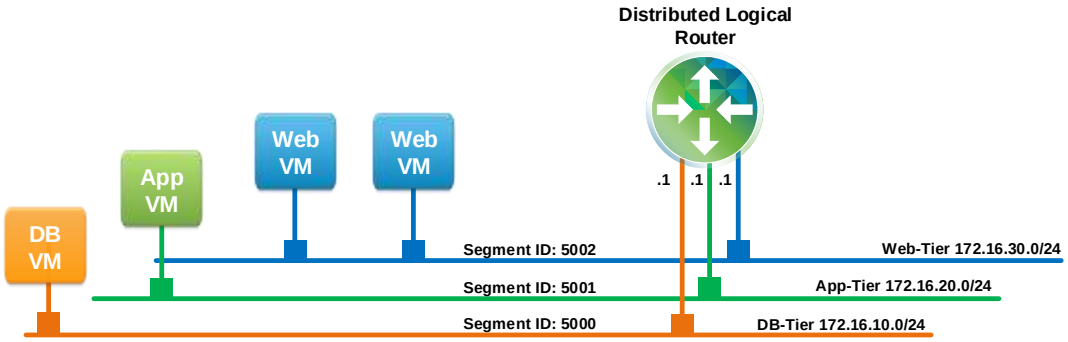


Abbildung 23: Distributed Logical Router

9.4.11 Hinzufügen eines Edge Services Gateway

Um eine Verbindung ins Internet herzustellen, braucht es einen Edge Service Gateway. Dieser verbindet das interne Netz mit der Aussenwelt. Die virtuelle Appliance wird auf dem ESXi-04 instanziiert und wird über die „Mgmt_VDS – Mgmt“ Port Group auf der IP Adresse 192.168.100.52 verwaltet. Für die interne Schnittstelle wird ein Port auf dem Transport Switch verwendet, welcher ebenfalls mit dem Distributed Logical Router verbunden ist. Für den Uplink wird eine neue Port Group auf dem Mgmt_VDS Switch angelegt, um ein weiteres Subinterface auf dem ESX Hosts zu erstellen, welches mir eine Verbindung nach aussen ermöglicht.

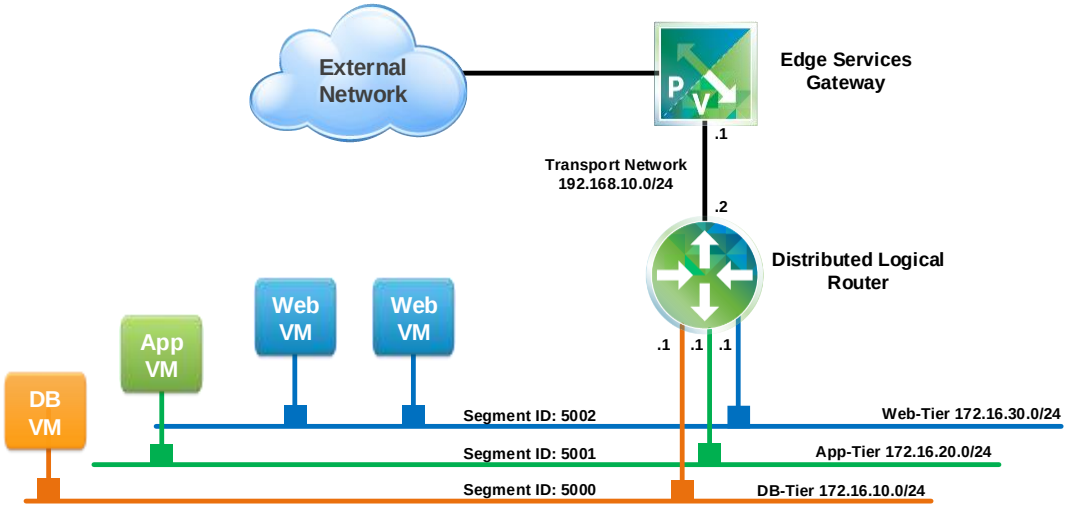


Abbildung 24: Edge Services Gateway

9.5 Technische Umsetzung

Im Hinblick auf die später folgenden Testszenarien ist es notwendig, einzelne Funktionen von VMware NSX genauer zu erläutern. Damit lassen sich mögliche Testresultate erklären.

9.5.1 Multi-Destination Traffic

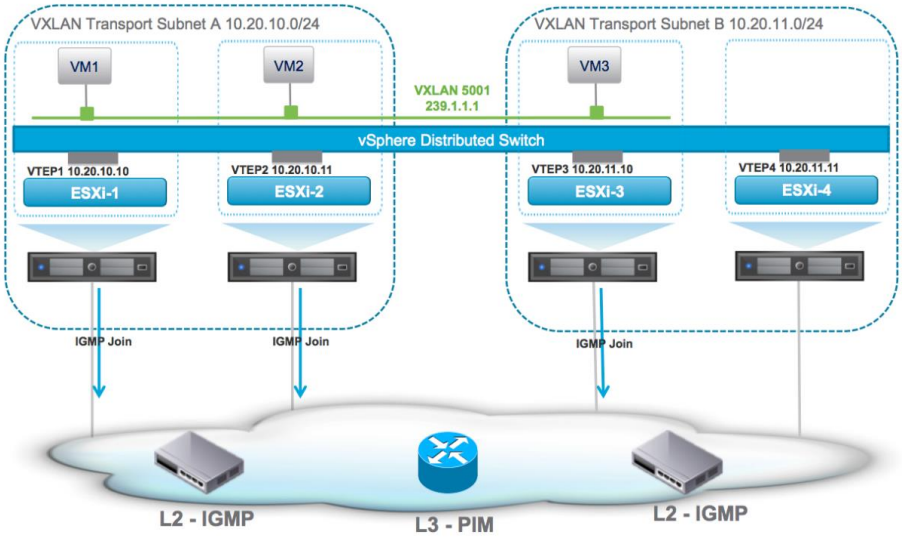
Wenn zwei VMs auf verschiedenen ESXi Hosts über VXLAN miteinander kommunizieren müssen, werden Datenpakte zwischen den VTEP IP-Adressen, welche den zwei Hypervisor gehören, ausgetauscht. Manchmal kann es jedoch vorkommen, dass eine VM Daten an mehrere Empfänger desselben logischen Netzwerks senden muss. Das kann bei folgenden drei Typen vorkommen, Broadcast, Unknown Unicast und Multicast, kurz BUM genannt.

Egal welches der drei Szenarien eintrifft, die Daten von dem ESXi Host müssen auf verschiedene Hosts repliziert werden. Dabei unterstützt NSX drei verschiedene Replikationsmodi – Multicast, Unicast und Hybrid, wobei Unicast die bevorzugte Variante ist.

9.5.1.1 Multicast-Mode

Falls Multicast als Replikationsmodus gewählt wird, muss sichergestellt werden, dass die darunterliegende Infrastruktur, sprich das physikalische Netzwerk, multicastfähig ist. Bei dieser Variante ist der NSX Kontroller nicht involviert, daher gibt es auch keine Abstraktion zwischen dem logischen und physikalischem Netzwerk.

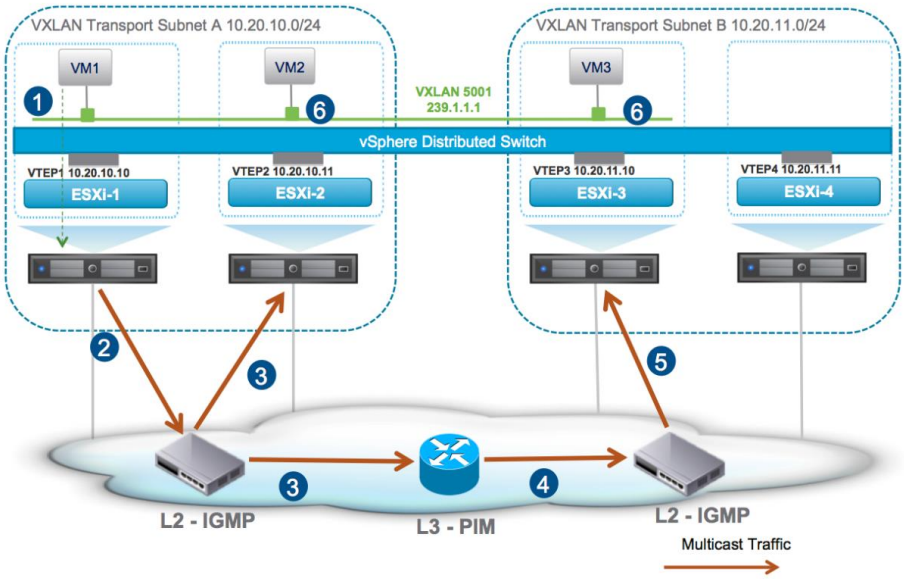
Im nachfolgenden Beispiel wird das VXLAN Segment 5001 der Multicast-Gruppe 239.1.1.1 zugewiesen. Sobald eine VM dem logischen Switch zugewiesen wird, generiert der ESXi Host eine IGMP Anfrage um der Multicastgruppe beizutreten. Wie es auf der Abbildung ersichtlich ist, ist ESXi-4 davon nicht betroffen. Dieser besitzt zum momentanen Zeitpunkt keine VM im besagten Netzwerk.



[6]*Abbildung 25: IGMP Joins*

Die genaue Abfolge von Ereignissen, welche bei einem BUM Frame auftreten, soll anhand der nächsten Abbildung erläutert werden.

[6] VMware® NSX for vSphere (NSX-V) Network Virtualization Design Guide



[7]*Abbildung 26: Multicast Mode*

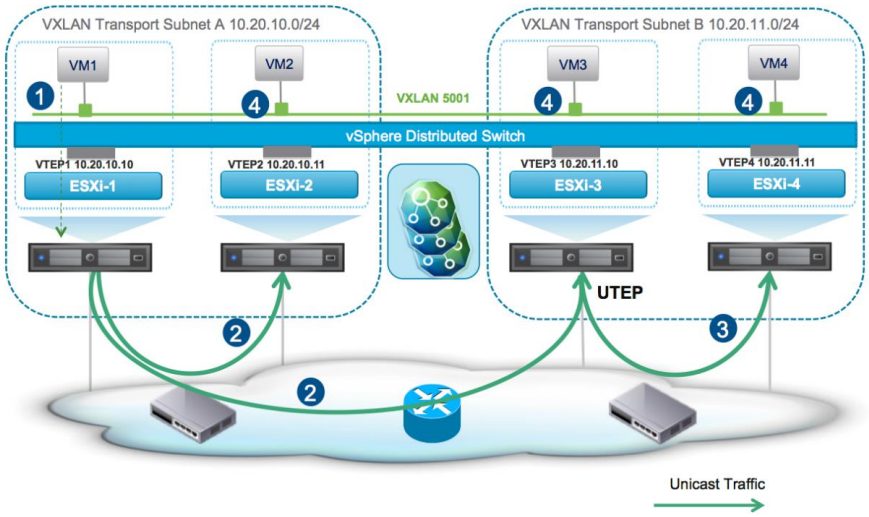
1. VM1 generiert ein BUM Frame.
2. Das VXLAN verschaltete original Frame wird von ESXi-1 an die Multicast-Gruppe 239.1.1.1 versendet.
3. Der Layer 2 Switch im Transportnetzwerk repliziert das Datenpaket und leidet es anschliessend an den ESXi-2 Host und dem Router weiter.
4. Vom Router gelangt das Paket an den relevanten Switch im Transportnetzwerk B.
5. Der Layer 2 Switch repliziert nötigenfalls das Paket nochmals und sendet es den betreffenden ESXi Hosts.
6. ESXi-3 entpackt das verschachtelte Frame wieder und sendet das originale Datenpaket an die VM3 weiter.

9.5.1.2 Unicast Mode

Eine komplett andere Methode wird mit dem Unicast Mode realisiert. Hierbei gibt es eine Entkoppelung vom physikalischen und logischen Netzwerk. Die Hosts in einer NSX Domain werden in separate Gruppen (VTEP Gruppen), basierend auf den VTEP Subnetze, aufgeteilt. Einem einzelnen ESXi Host, pro VTEP Subnetz, wird die Rolle des Unicast Tunnel End Point (UTEP) zugewiesen. Der UTEP ist verantwortlich, den BUM Datenverkehr zu kopieren und innerhalb seines Zuständigkeitsbereichs an alle relevanten ESXi Host zu verteilen. Um keine unnötigen Datenpakete zu generieren, werden nur ESXi Hosts angesprochen, welche ebenfalls VMs im relevanten Netzwerk haben.

Der Ablauf wird mithilfe der nachfolgenden Grafik erläutert:

[7] VMware® NSX for vSphere (NSX-V) Network Virtualization Design Guide



[8]*Abbildung 27: Unicast Mode*

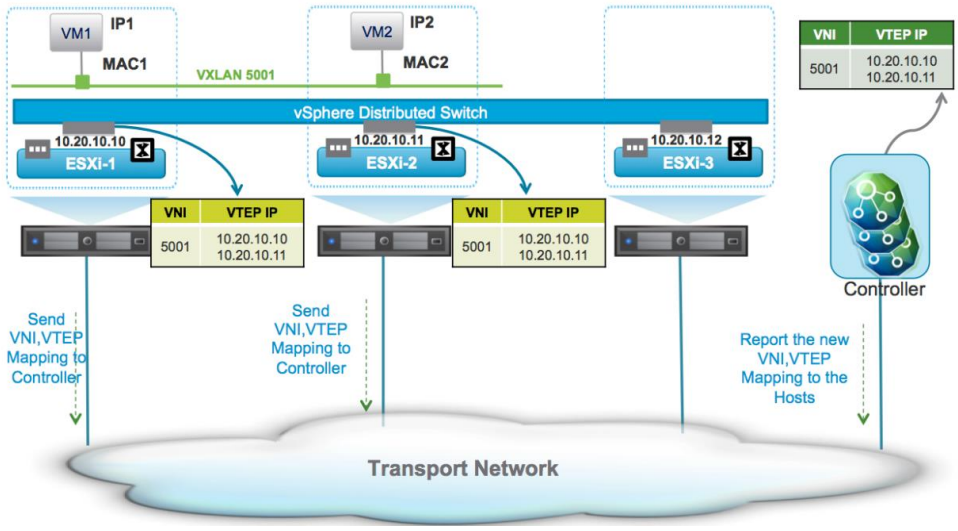
In diesem Beispiel gibt es 4 VMs die sich im selben logischen Segment (VXLAN 5001) befinden.

1. VM1 versendet ein BUM Paket an alle VMs desselben logischen Netzwerks.
2. ESXi-1 prüft seine lokale VTEP Tabelle, welche seine Einträge vom NSX Kontroller bezieht, um zu bestimmten, wohin das Datenpaket per Unicast versendet werden soll. Anschliessend ist der Host in der Lage, das Paket ausschliesslich für den Host ESXi-2 und für den UTEP Host zu replizieren. Mithilfe eines spezifischen Bits im VXLAN Header kann der Host ESXi-3 feststellen, dass das Paket von einem entfernten VTEP Segment kommt und es möglichenfalls an weitere lokale Host kopiert werden muss.
3. ESXi-3 überprüft in seiner lokalen VTEP Tabelle, ob es weitere Hosts im Segment gibt, die eine Kopie des Datenpakets brauchen. Daraus resultiert, dass ebenfalls das Paket an ESXi-4 versendet wird.

[8] VMware® NSX for vSphere (NSX-V) Network Virtualization Design Guide

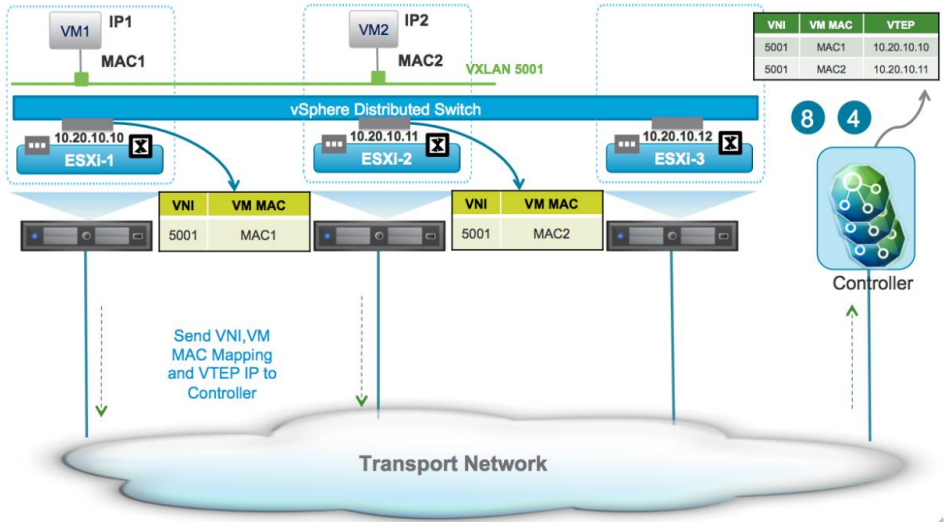
9.5.2 Informationsaustausch der Control Plane

Dieser Abschnitt befasst sich mit der Unicast Kommunikation auf Layer 2. Der Fokus liegt dabei auf der Control Plane Kommunikation. Es soll aufzeigen, wie die VTEP, MAC und ARP Informationen zwischen den ESXi Hosts und den NSX Kontrollern ausgetauscht werden.



[9]*Abbildung 28: VNI-VTEP Table*

Sobald sich die erste VM an das VXLAN Segment verbindet, generiert der ESXi Host eine Control Plane Nachricht, welche an den NSX Kontroller versendet wird. Diese Nachricht beinhaltet eine VNI zu VTEP Abbildung. Daraufhin überarbeitet der Kontroller seine lokale Tabelle und versendet seine neuen Daten allen relevanten ESXi Hosts. Dabei werden Hosts ausgelassen, die keine VMs im betreffenden Segment besitzen.

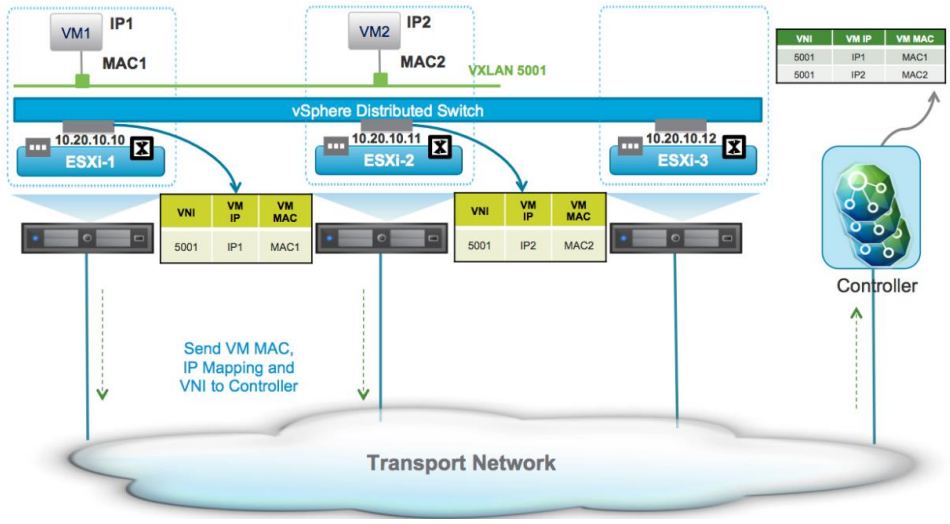


[10]*Abbildung 29: VNI-MAC Table*

Eine weitere Information die dem NSX Kontroller zugespielt wird, ist die VNI zu VM-MAC Zugehörigkeit. Anders als im ersten Teil, wird diese Information anschliessend nicht allen ESXi Hypervisor mitgeteilt. Somit besitzt der Host nur einen Überblick über die lokalen MAC Adressen.

[9] VMware® NSX for vSphere (NSX-V) Network Virtualization Design Guide

[10] VMware® NSX for vSphere (NSX-V) Network Virtualization Design Guide



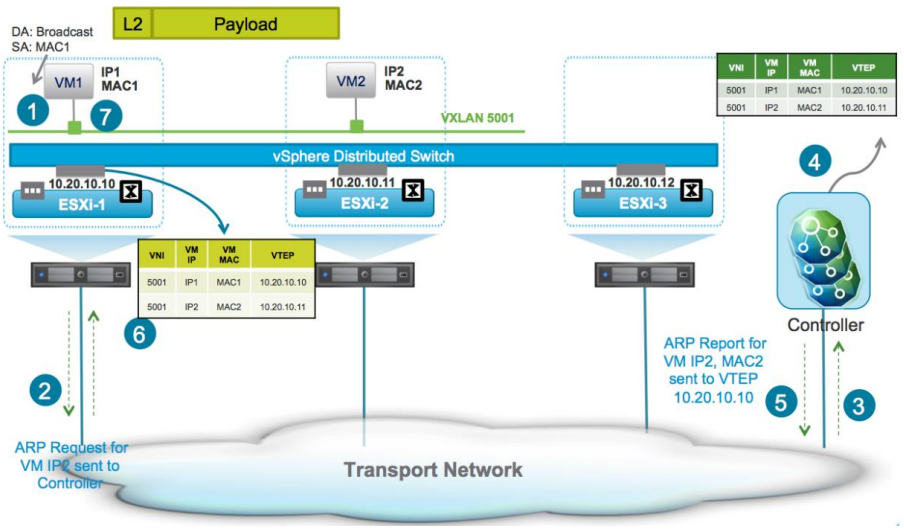
[11]*Abbildung 30: VNI-IP Table*

Zum Schluss benötigt der NSX Kontroller ebenfalls die IP Adresse von den VMs. Auch diese Information wird den ESXi Hosts anschliessend nicht propagiert. Der Kontroller benötigt dieses Wissen, um die ARP Auslastung zu minimieren.

9.5.3 Unicast Traffic

Werden die Informationen aus den zuvor beschriebenen Tabellen genommen, kann der NSX Kontroller „ARP Suppression" durchführen. Damit wird eine Überflutung von ARP Traffic unterbunden. ARP Anfragen stellen den Grossteil des Broadcast Traffic dar. Kann dieser vermieden werden, steigert das die Stabilität und Skalierbarkeit des gesamten Netzwerks.

Die nachfolgende Abbildung zeigt wie eine ARP Auflösung gehandhabt wird.



[12]*Abbildung 31: ARP Resolution*

1. VM1 generiert ein ARP Request, um die MAC Adresse seines Ziels (VM2) zu bestimmen.

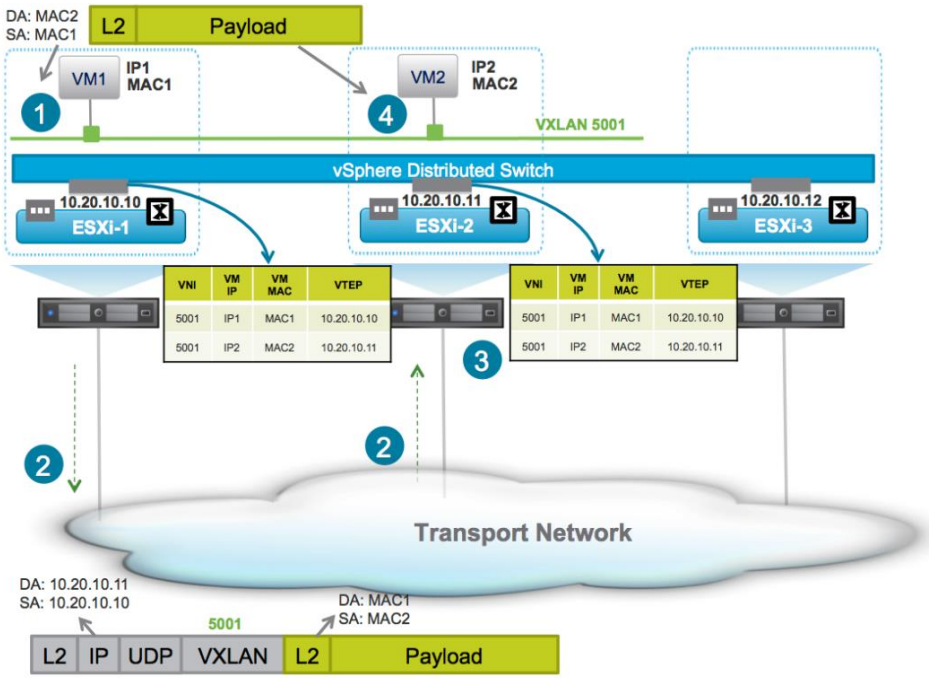
[11] VMware® NSX for vSphere (NSX-V) Network Virtualization Design Guide

[12] VMware® NSX for vSphere (NSX-V) Network Virtualization Design Guide

2. Da ESXi-1 keine lokale Information über diese Abfrage hat, muss der NSX Kontroller kontaktiert werden.
3. Der zuständige Kontroller erhält die Anfrage vom ESXi Hosts.
4. Der Kontroller durchsucht seine lokale ARP Tabelle um die passende Antwort zu liefern.
5. Die benötigten Informationen werden dem ESXi Host gesendet. Dies beinhaltet die VTEP Adresse des ESXi Hosts, bei der die VM2 beherbergt ist.
6. ESXi-1 erhält die Control Plane Nachricht und aktualisiert seine lokale Tabelle .
7. Im letzten Schritt kann der VM1 mitgeteilt werden, welche MAC Adresse die VM2 besitzt.

Sollte der NSX Kontroller über keine Angabe betreffend der Anfrage verfügen, wird wie im Abschnitt „Multi-Destination Traffic“ beschrieben, ein BUM Frame abgesetzt und je nach Replikationsmodus abgehandelt.

Sobald VM1 seinen ARP Cache aktualisiert hat, ist die VM in der Lage die Daten an VM2 zu senden. Dabei ergibt sich folgender Ablauf.



[13]Abbildung 32: L2 VM to VM Communication

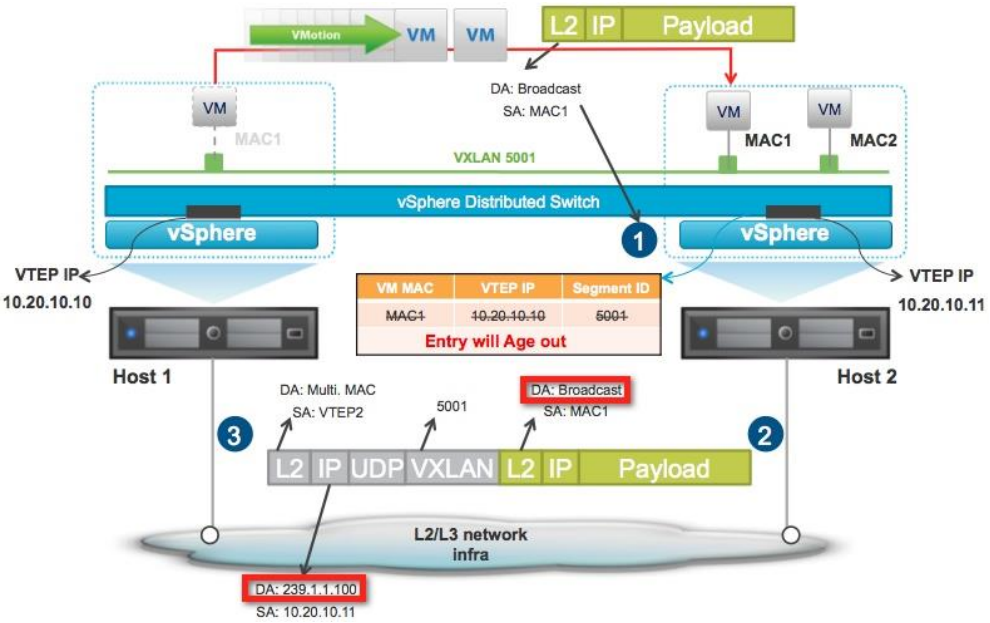
1. VM1 sendet ein Datenpaket an VM2.
2. ESXi-1 verfügt über alle Informationen und verschachtelt das Paket in ein VXLAN Paket. Anschliessend wird das Paket an die VTEP Adresse von ESXi-2 versendet.
3. Beim entpacken des Datenpakets kann ESXi-2 alle wichtigen Informationen auslesen, um anschliessend seine lokale Tabelle anzupassen.
4. Das Datenpaket wird VM2 zugestellt.

Jetzt da beide ESXi-Hosts über alle relevanten Informationen verfügen, kann der Traffic in beide Richtungen fliessen.

[13] VMware® NSX for vSphere (NSX-V) Network Virtualization Design Guide

9.5.4 Einfluss von vMotion auf die Forwarding Table

Dieser Abschnitt soll aufzeigen, was geschieht, wenn eine VM von einem Host zu einem anderen Host verschoben wird. Interessant ist dabei, wie sich die „VTEP forwarding table" der einzelnen Hosts verändert. Unten abgebildet sind Host 1 und Host 2, welche bereits die VMs im logischen Netzwerk (VXLAN 5001) kennen.



[14]Abbildung 33: vMotion VTEP changes

1. Sobald die VM-MAC1 vom Host 1 zu Host 2 durch vMotion verschoben wird, wird vom Zielhost ein Reverse ARP initialisiert. Dabei wird im Beispiel davon ausgegangen, dass der Broadcast Replikationsmodus verwendet wird.
2. Das Ethernet Broadcast Paket wird vom VTEP auf Host 2 eingepackt und an die Multicast Adresse 239.1.1.100 versendet.
3. Alle ESXi Host die in derselben Multicast Gruppe sind, erhalten das Paket. Mithilfe der Informationen aus dem äusseren und inneren Header des Pakets, werden alle nötigen Informationen für eine Anpassung der „forwarding table" geliefert.

[14] http://blogs.vmware.com/vsphere/files/2013/07/Update-vMotion-2.jpg

9.5.5 QoS Tagging

Es werden vorwiegend zwei Arten von QoS Parameter auf der physikalischen Infrastruktur unterstützt. Eines davon wird auf Layer 2 abgehandelt und wird oftmals als Class of Service (CoS) bezeichnet und das andere auf Layer 3 und nennt sich „DSCP marking“. NSX ist dabei in der Lage, der ursprünglichen DSCP Markierung der VM zu vertrauen und das Paket dementsprechend weiterzuleiten. Zusätzlich besteht die Möglichkeit, den QoS Parameter auf dem logischen Switch explizit zu setzen. Dafür stehen drei verschiedene Filter Optionen zur Verfügung. Bei der ersten Variante kann je nach Systemdatentyp unterschieden werden. Zur Auswahl steht unteranderem vMotion, Management oder iSCSI Traffic. Die zweite Variante ist der MAC Filter. Damit kann einerseits nach einer Quell- bzw. Zieladresse gefiltert werden oder nach einem bestimmten VLAN Tag. Bei der letzten Variante kann eine Filteroption auf die IP-Adresse oder auf den Layer 4 Port gesetzt werden.

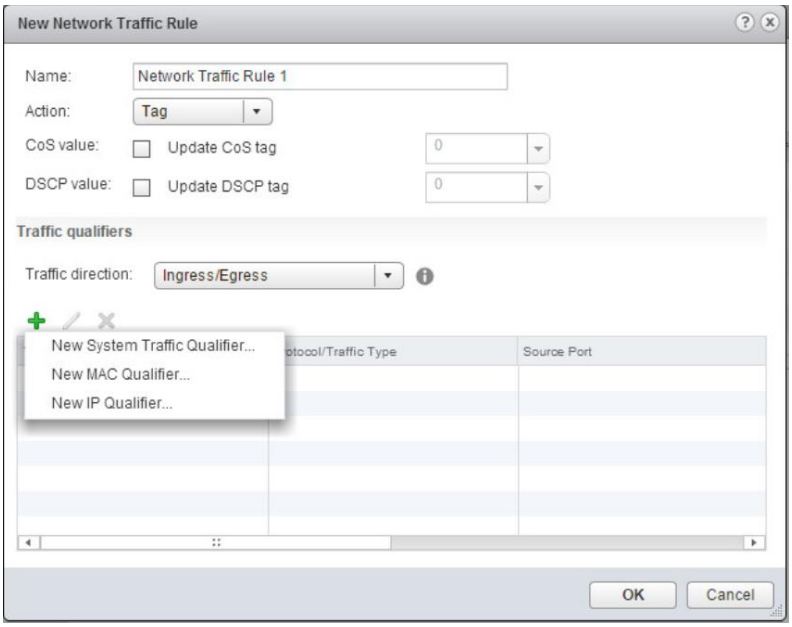


Abbildung 34: Logical Switch- Traffic Tagging

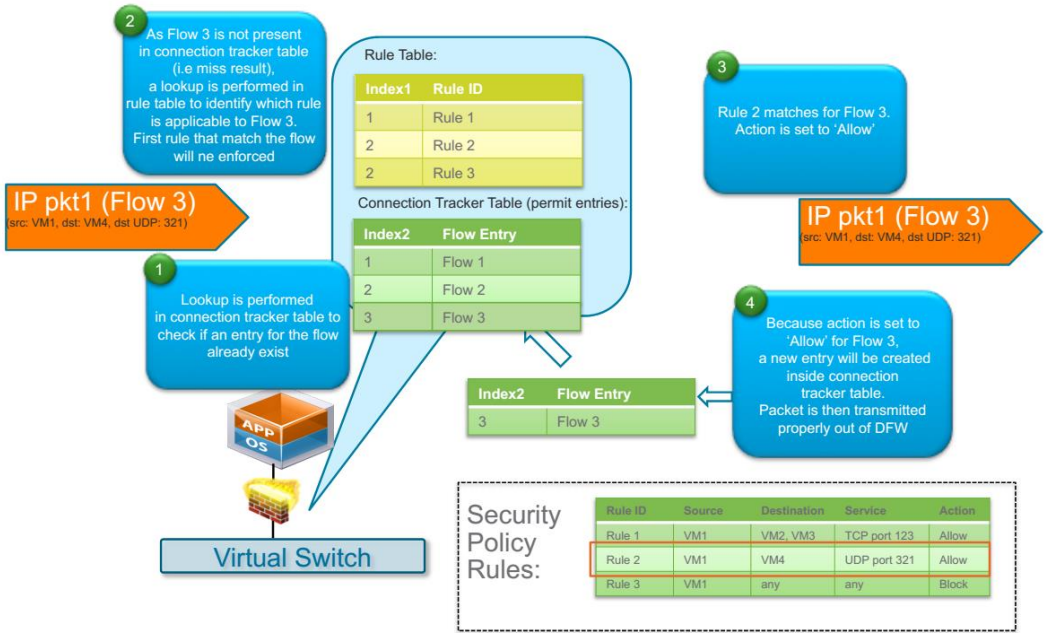
9.5.6 Funktionsweise der NSX Distributed Firewall

Die NSX Distributed Firewall (DFW) bietet einen L2-L4 Stateful Firewallservice und wird direkt mit der Installation der VIBs auf den einzelnen ESXi Hosts aktiviert. Primär wurde der Service entwickelt, um „workload-to-workload“ Datenverkehr zu sichern, sprich Ost-West Verkehr. Eine DFW Instanz wird für jede einzelne virtuelle Netzwerkkarte pro VM und ESXi-Host erstellt. Dabei kann eine Firewallregel auf zwei unterschiedliche Wege erstellt werden. L2 Regeln, welche auf dem Layer 2 des OSI Models basieren oder L3/L4 Regeln welche die Schicht 3 und 4 des OSI Models verwenden.

Grundsätzlich besitzt jede DFW Instanz zwei separate Tabellen die wie folgt definiert sind:

- **Rule Table**: Wird benutzt um alle Firewallregeln zu speichern.
- **Connection Tracker Table**: Speichert die offenen Flow Einträge, die per Policy erlaubt wurden.

Die Überprüfung einer Firewallregel wird wie folgt abgehandelt:



[15]*Abbildung 35: DFW Policy Rule Lookup*

Ein Datenpaket pkt1 möchte von der VM1 zur VM4 gelangen und dabei den UDP Port 321 verwenden:

1. Es gibt eine Überprüfung in der „connection tracker table". Dabei wird nach einer offenen Verbindung gesucht.
2. Da es keinen Eintrag für den Flow 3 gibt, wird die „rule table" nach einem passenden Eintrag durchsucht.
3. Es gibt einen Treffer in der „rule table". Der gewünschte Datenfluss ist erlaubt.
4. Damit der Datenfluss bis zu einem Timeout nicht immer aufs Neue überprüft werden muss, wird ein entsprechender Eintrag in der „connection tracker table" hinterlegt. Das beschleunigt den Auflauf um ein vielfaches.

Ein wichtiger Aspekt bei der Verwendung von DFW Regeln ist die Tatsache, dass vMotion vollständig unterstützt wird. Die beiden Tabellen werden automatisch mitverschoben. Somit wird der erlaubte Datenfluss aufrechterhalten.

9.5.7 Arbeiten mit der NSX API

Die VMware NSX RESTful API ermöglicht es mit dem NSX-Manager zu interagieren und dabei Daten auszulesen oder zu manipulieren. Jegliche Einstellungen die über das GUI vorgenommen werden können, werden ebenfalls vollständig von der API abgedeckt. Die API ist sogar noch mächtiger und bietet zusätzliche Möglichkeiten, die von der grafischen Oberfläche nicht unterstützt werden. Die Kommunikation wird über HTTPS abgewickelt und erfordert eine Authentifizierung. Die Anfragen können dabei wahlweise im JSON oder XML Format abgesetzt werden.

[15] VMware® NSX for vSphere (NSX-V) Network Virtualization Design Guide

9.6 Monitoring Tools

9.6.1 Flow Monitoring

Mithilfe des Flow Monitoring Tools, können detaillierte Informationen zum Datenverkehr eingeholt werden. Die Ausgabe zeigt auf, welche Maschinen miteinander kommunizieren und welche Anwendung dabei verwendet wird. Die Sitzungsdetails umfassen die Quellen, Ziele, Anwendungen sowie die verwendeten Ports. Anhand der Sitzungsdetails können ebenfalls direkt Firewallregeln definiert werden.

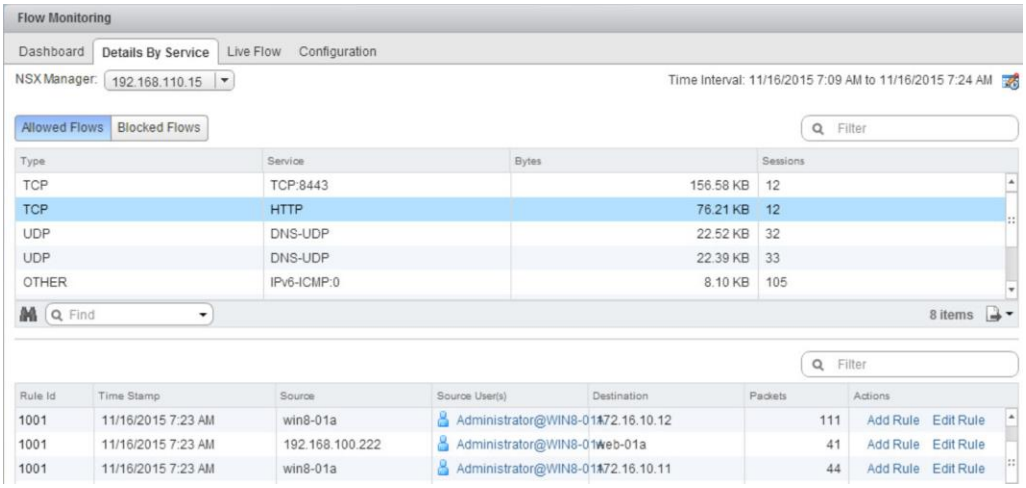


Abbildung 36: Flow Monitoring

Das Tool bietet einerseits ein Ereignisprotokoll und anderseits eine Echtzeit-Überwachung.

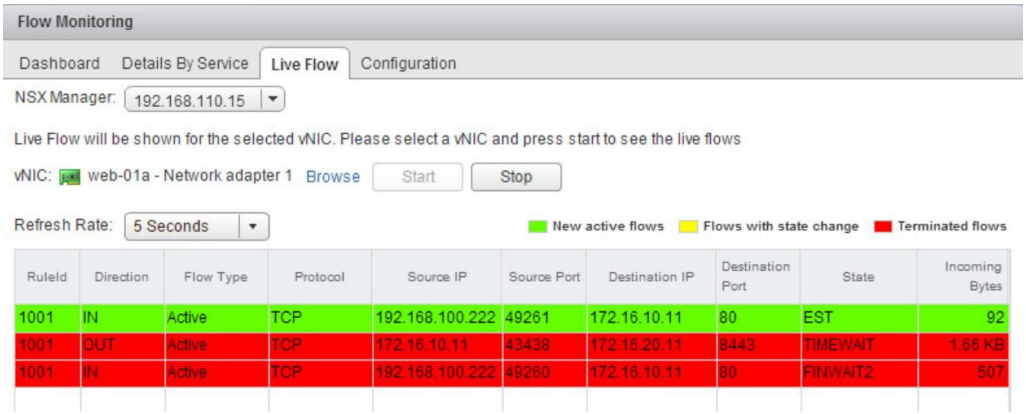


Abbildung 37: Live-Flow-Monitoring

9.6.2 Traceflow

Traceflow ist ein Werkzeug, das zur Fehlerbehebung eingesetzt werden kann. Es ist fähig, spezielle Pakete zur erstellen, um anschliessend den Weg dieses Pakets durch das physische und das logische Netzwerk zu verfolgen. Die Beobachtung erlaubt es, einzelne Knoten zu identifizieren und mögliche Firewallprobleme zu erkennen.

Traceflow unterstützt die folgenden Arten des Datenverkehrs:

- Schicht 2-Unicast
- Schicht 3-Unicast
- Schicht 2-Broadcast
- Schicht 2-Multicast

Es können Pakete mit benutzerdefinierten Kopfzeilen und Paketgrössen erstellt werden. Als Quelle des Traceflows muss immer eine virtuelle Netzwerkkarte (vNIC) einer virtuellen Maschine angegeben werden. Der Zielendpunkt kann aber ein beliebiges Gerät im NSX Overlay oder Underlay sein.

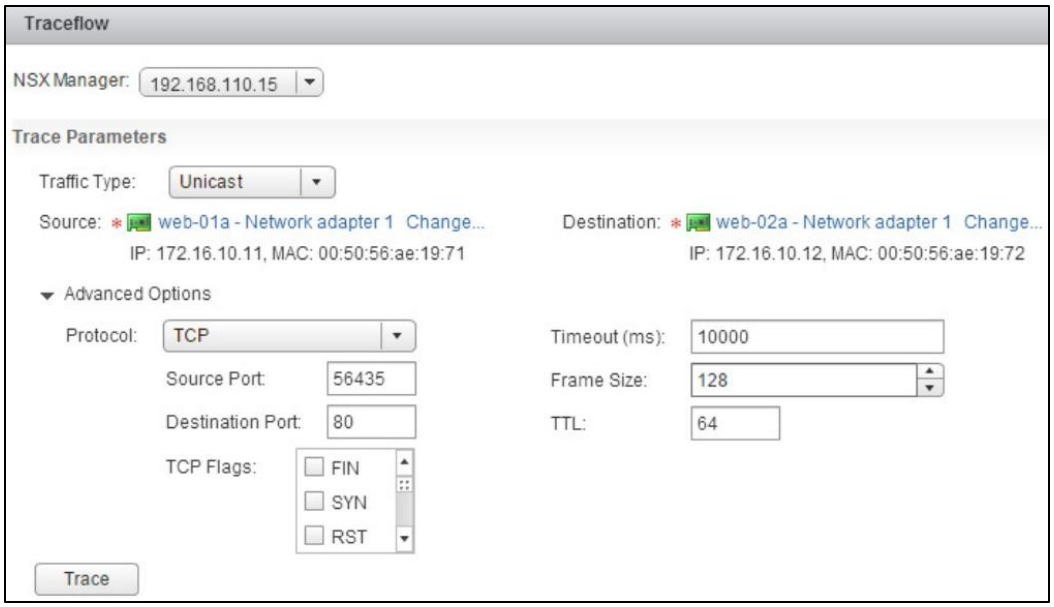


Abbildung 38: Traceflow Parameters

Zur Veranschaulichung wird ein Layer 2 Traceflow gezeigt, die zwei VMs umfasst, welche auf unterschiedlichen ESXi Host ausgeführt werden. Die zwei VMs sind mit einem einzelnen logischen Switch verbunden. Es zeigt die einzelnen Knoten auf und welche Firewallregeln involviert sind.

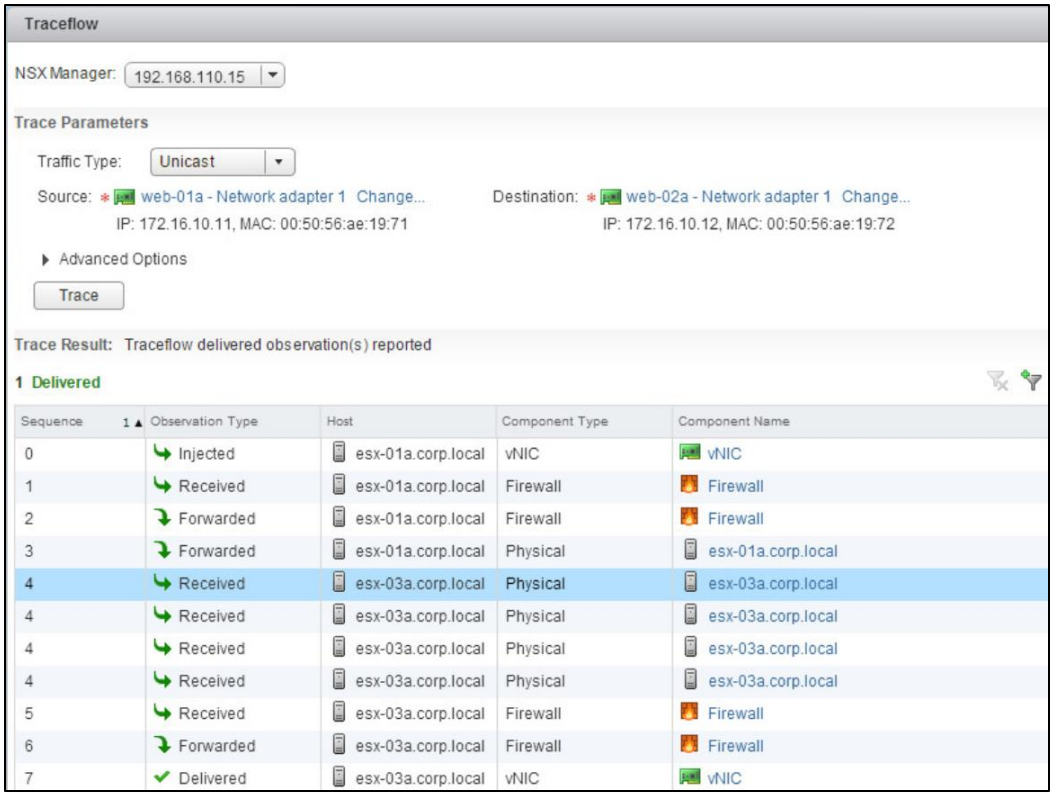
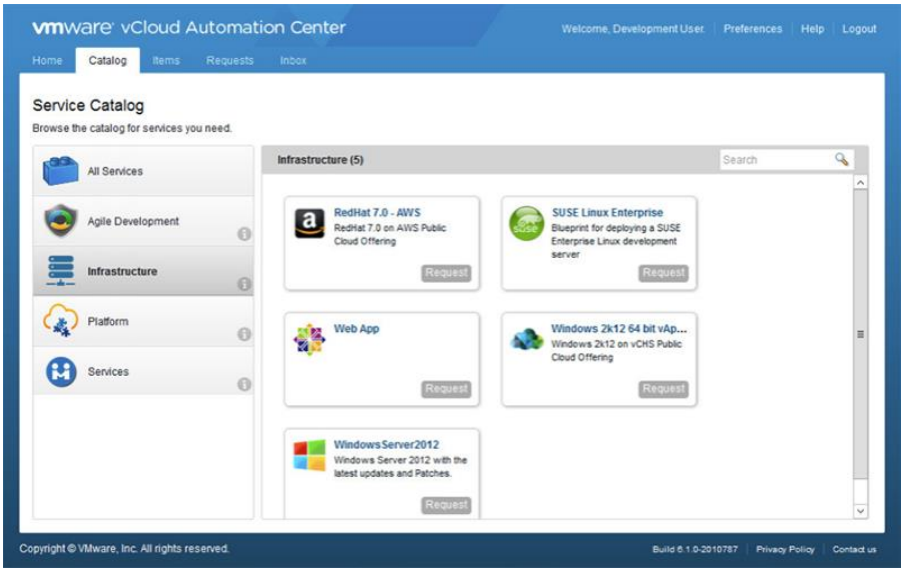


Abbildung 39: Traceflow Example

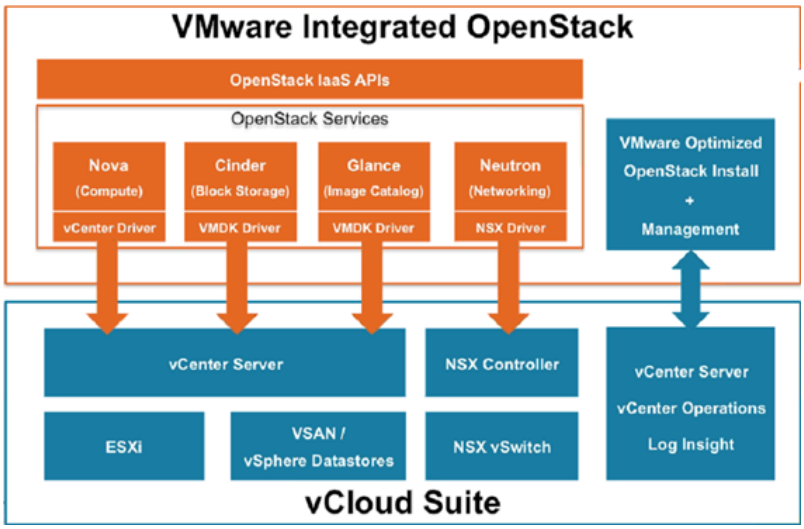
9.7 Cloud Automatisierung

Der Zeitaufwand für die Bereitstellung von Services konnte mit NSX und dem vCenter bereits drastisch reduziert werden. Dennoch wurde bis anhin jeder einzelne Schritt manuell durchgeführt. Um dies zu umgehen, kann auf die RESTful API zurückgegriffen werden. Dazu ist es nicht notwendig, eigene Applikationen zu schreiben. VMware selbst bietet mit ihrer VMware vCloud Suite alle nötigen Komponenten, die es zur Automatisierung von Prozessen braucht. Mithilfe von Blueprint-Modellen können ganze Servicekataloge samt komplexen Netzwerkkonstellationen erstellt werden. Anschliessend kann dem Kunden ein Self-Service Portal zur Verfügung gestellt werden.



[16]*Abbildung 40: vCloud Automation Center*

Eine mögliche alternative wird ebenfalls mit VMware Integrated OpenStack (VIO) angeboten. VIO ist eine OpenStack-Distribution, um auf Basis einer vSphere-Infrastruktur eine Private Cloud aufzubauen. Mithilfe von einzelnen Modulen kann ebenfalls auf die NSX Kontroller zugegriffen werden, was eine Modellierung der Netzwerkkomponenten ermöglicht.



[17]*Abbildung 41: VMware Integrated OpenStack*

[16] http://www.vmware.com/files/images/thumbnails/service-thumbnail.png

[17] http://www.vclouds.nl/wp-content/uploads/2014/08/SNAGHTML4c74ccb.png

9.8 Testszenarien

Um später einen Vergleich zwischen verschiedenen Herstellern bzw. Produkte n durchführen zu können, werden umfassende Tests vorgenommen. Diese basieren auf den zuvor defi nierten detaillierten Anforderungen aus Kapitel 8.

9.8.1 Testfall 1

Test # 1	Das Implementieren von Multi-Tier Services kann innert wenigen Minuten anstelle von Tagen erfolgen.
Datum	10.11.2015 – 08:45 Uhr
Testumgebung	VMware Hands-on Lab
Ausgangslage	Für einen Neukunden soll eine 3-Tier Applikation bereitgestellt werden. Dabei werden bereits vorkonfigurierte Server verwendet. Folgende Komponenten müssen neu konfiguriert werden: • 3x Logical Switches • 2x Web-Server (Portzuweisung) • 1x App-Server (Portzuweisung) • 1x DB-Server (Portzuweisung) • 1x Logical Distributed Router • 1x Load Balancer (one-arm Mode)
Wie wird getestet	Sämtliche Einstellungen werden über das vCenter vorgenommen. Der Ablauf erfolgt sequenziell. Die genauen Parameter sind dem VMware Hans-on Lab „HOL-SDC-1603-HOL“ zu entnehmen.
Erwartetes Ergebnis	Die Implementierung dauert nicht länger als 10 Minuten.
Resultat	Die Implementierung hat 22 Minuten und 35 Sekunden gedauert.
Erläuterung	Werden alle Schritte über das vCenter abgewickelt, dauert die Implementierung viel länger als erwartet. Alle Einstellungen müssen nach wie vor manuell ausgeführt werden. Um diesem Problem entgegen zu wirken, sollte unbedingt mit einem zusätzlichen Produkt wie vCloud oder OpenStack gearbeitet werden. Damit können komplexe Prozesse automatisiert abgewickelt werden. Es würde ebenfalls die Möglichkeit bieten, einen „Self-Service“ Katalog dem Endkunden anzubieten.

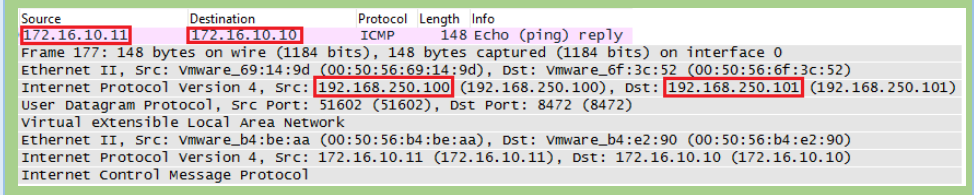
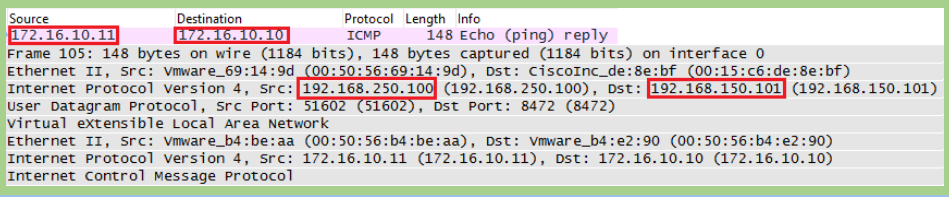
9.8.2 Testfall 2

Test # 2	Das Verschieben einer VM Instanz mit 4GB RAM und zentralisiertem Datenspeicher, darf nicht länger als 300 Sekunden dauern.
Datum	10.11.2015 – 09:45 Uhr
Testumgebung	VMware Hands-on Lab
Ausgangslage	Es existieren zwei unterschiedliche Compute Cluster die sich in einem dedizierten vMotion Netzwerk befinden. Die Datenrate beträgt max. 10Gbit/s.
Wie wird getestet	Eine aktive VM Instanz wird via vCenter auf einen anderen Cluster migriert. Es werden die Standardeinstellungen des Assistenten verwendet. Lediglich die Switchangabe für die VM muss neu gesetzt werden.
Mögliche Auswirkungen	Es gibt einen Verbindungsunterbruch bei der VM. ->Test #3 Der Verkehrsfluss ist nicht mehr optimal. ->Test #4 VM spezifische Policies werden nicht migriert. ->Test #5
Erwartetes Ergebnis	Die Migrationszeit beträgt weniger als 300 Sekunden.
Resultat	Die Migrationszeit beträgt 31 Sekunden.
Erläuterung	vMotion benutzt für gewöhnlich einen eigenen VMkernel Adapter, der sich in einem dedizierten Subnetz befindet. Dabei wird der Datenverkehr nicht über das Overlay Netzwerk abgewickelt. Die Geschwindigkeit und der Traffic Flow hängt massgeblich von der darunterliegenden Architektur ab. Mit einer 10Gbit/s Netzwerkkarte wird das gleichzeitige Verschieben von max. 8 Systemen erlaubt.

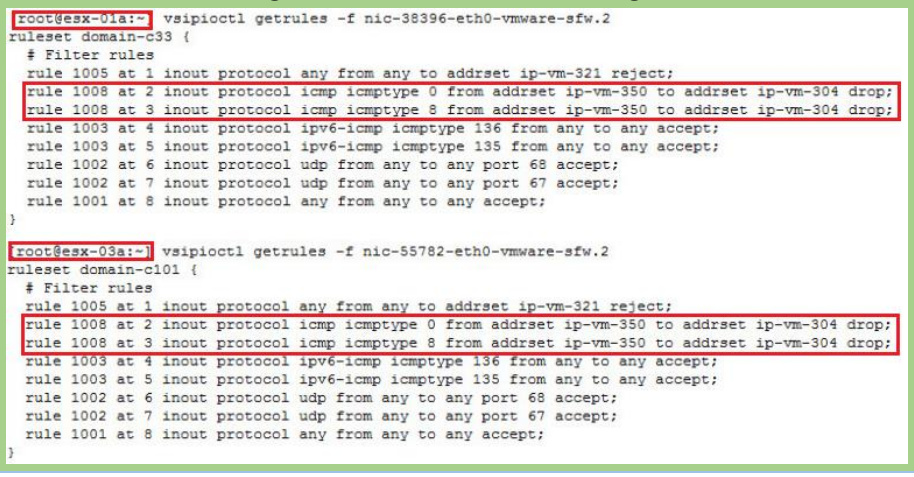
9.8.3 Testfall 3

Test # 3	Das Verschieben einer VM Instanz führt nicht zu einem Unterbruch des Services.
Datum	15.11.2015 – 15:15 Uhr
Testumgebung	HSR-Lab
Ausgangslage	Der Mandant besitzt einen Webserver, der einen Service nach Aussen anbietet. Aus Performance- oder Wartungsgründen kann es vorkommen, dass die VM Instanz von einem Compute Cluster auf einen anderen Cluster verschoben wird. Der Unterbruch des Services darf für einen aussenstehenden Benutzer nicht spürbar sein.
Wie wird getestet	Eine aktive VM Instanz wird via vCenter auf einen anderen Cluster migriert. Während der gesamten Migration wird sowohl die interne sowie die öffentliche IP-Adresse gepingt.
Erwartetes Ergebnis	Es gibt einen minimalen Unterbruch. Dabei gehen 2-3 ICMP Pakete verloren.
Resultat	Es geht kein einziges ICMP Paket verloren, weder von der öffentlichen noch von der internen IP-Adresse.
Erläuterung	Alle nötigen Instanzen werden umgehend informiert, Es ist kein Unterbuch erkennbar. Lediglich eine leichte Erhöhung der Reaktionszeit ist festzustellen.

9.8.4 Testfall 4

Test # 4	Der Traffic Flow bleibt auch nachdem verschieben einer VM optimal.
Datum	17.11.2015 – 11:15 Uhr
Testumgebung	HSR-Lab
Ausgangslage	VMs werden aus Wartungs- oder Performancegründen regelmässig auf unterschiedliche ESXi Hosts verschoben. Dabei muss weiterhin der bestmögliche Pfad für den Verkehrsfluss gewähreistet sein. Die Testumgebung besteht aus VM1 (172.16.10.10) auf ESXi-4 (192.168.250.101) und VM2 (172.16.10.11) auf ESXi-3 (192.168.250.100).
Wie wird getestet	Eine aktive VM Instanz wird via vCenter auf einen anderen Cluster migriert. Dabei wird der Traffic Flow vor und nach der Migration analysiert. Der Test wird mittels ICMP Ping durchgeführt. Der Ziel Host von VM1 wird ESXi-2 (192.168.150.101) sein.
Erwartetes Ergebnis	Es wird weiterhin der bestmögliche Pfad angeboten.
Resultat	Nachdem verschieben der VM-Instanz wird wie erwartet der direkte und somit optimale Pfad verwendet. **Vorher**:  **Nachher**: 
Erläuterung	Mit vMotion werden auch gleich sämtliche VTEP Tabellen angepasst. Das garantiert, dass immer der optimale Pfad verwendet wird.

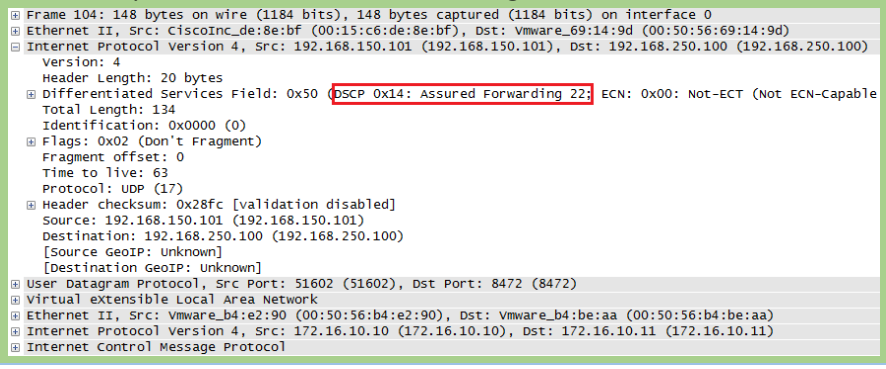
9.8.5 Testfall 5

Test # 5	Distributed Firewall Settings werden mit vMotion problemlos mitmigriert.
Datum	10.11.2015 – 13:15 Uhr
Testumgebung	VMware Hands-on Lab
Ausgangslage	Es existieren 2 VMs (Web01 und Web02), die sich im selben Subnetz befinden. Die Distributed Firewall wurde so konfiguriert, dass ein ICMP Ping von Web01 zu Web02 nicht erlaubt ist.
Wie wird getestet	Die Distributed Firewall wird gemäss Ausgangslage vorkonfiguriert. Anschliessend wird die VM Web01 vom einem ESXi Cluster auf einen anderen Cluster verschoben. Dabei sollte auf dem ESXi Host ersichtlich sein, dass die Firewalleinstellungen mitmigriert wurden.
Erwartetes Ergebnis	Die Firewalleinstellungen werden mit der VM mitmigriert.
Resultat	Die Firewalleinstellungen wurden mit der VM mitmigriert. 
Erläuterung	vMotion wird vollständig von der Distributed Firewall unterstützt. Alle benötigten Einstellungen folgen der VM bei einer Migration.

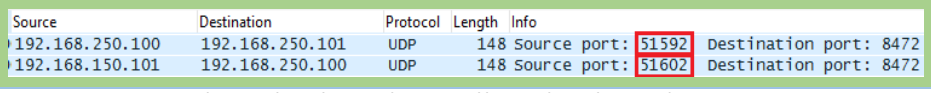
```
root@esx-01a:~ vsipioctl getrules -f nic-38396-eth0-vmware-sfw.2
ruleset domain-c33 {
  # Filter rules
  rule 1005 at 1 inout protocol any from any to addrset ip-vm-321 reject;
  rule 1008 at 2 inout protocol icmp icmptype 0 from addrset ip-vm-350 to addrset ip-vm-304 drop;
  rule 1008 at 3 inout protocol icmp icmptype 8 from addrset ip-vm-350 to addrset ip-vm-304 drop;
  rule 1003 at 4 inout protocol ipv6-icmp icmptype 136 from any to any accept;
  rule 1003 at 5 inout protocol ipv6-icmp icmptype 135 from any to any accept;
  rule 1002 at 6 inout protocol udp from any to any port 68 accept;
  rule 1002 at 7 inout protocol udp from any to any port 67 accept;
  rule 1001 at 8 inout protocol any from any to any accept;
}

root@esx-03a:~] vsipioctl getrules -f nic-55782-eth0-vmware-sfw.2
ruleset domain-c101 {
  # Filter rules
  rule 1005 at 1 inout protocol any from any to addrset ip-vm-321 reject;
  rule 1008 at 2 inout protocol icmp icmptype 0 from addrset ip-vm-350 to addrset ip-vm-304 drop;
  rule 1008 at 3 inout protocol icmp icmptype 8 from addrset ip-vm-350 to addrset ip-vm-304 drop;
  rule 1003 at 4 inout protocol ipv6-icmp icmptype 136 from any to any accept;
  rule 1003 at 5 inout protocol ipv6-icmp icmptype 135 from any to any accept;
  rule 1002 at 6 inout protocol udp from any to any port 68 accept;
  rule 1002 at 7 inout protocol udp from any to any port 67 accept;
  rule 1001 at 8 inout protocol any from any to any accept;
}
```

9.8.6 Testfall 6

Test # 6	Der Datenverkehr kann für die gewünschten Services Clusterübergreifend priorisiert werden.
Datum	17.11.2015 – 09:53 Uhr
Testumgebung	HSR-Lab
Ausgangslage	Es existiert eine VM (172.16.10.10) auf dem Management Cluster und eine VM (172.16.10.11) auf dem Compute Cluster. Auf dem logischen Switch wird definiert, dass jeglicher Datenverkehr von der Quell IP-Adresse zur Ziel IP-Adresse auf Layer 3 getaggt wird. DSCP-Wert soll auf 22 gesetzt werden.
Wie wird getestet	Der Datenverkehr von der VM1 zur VM2 wird auf dem physikalischen Netzwerk mittels Wireshark analysiert.
Erwartetes Ergebnis	Im Wireshark ist ersichtlich, dass eine DSCP Markierung vorliegt.
Resultat	Das Datenpaket hat eine DSCP Markierung. 
Erläuterung	NSX bietet die Möglichkeit eine L2 oder L3 Markierung für ein- oder ausgehende Pakete zu setzen.

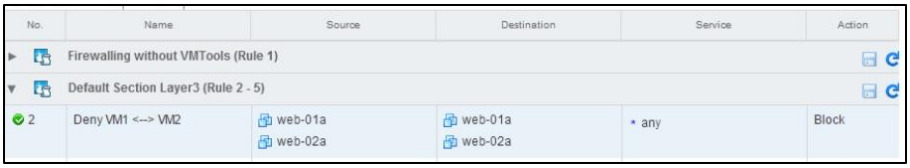
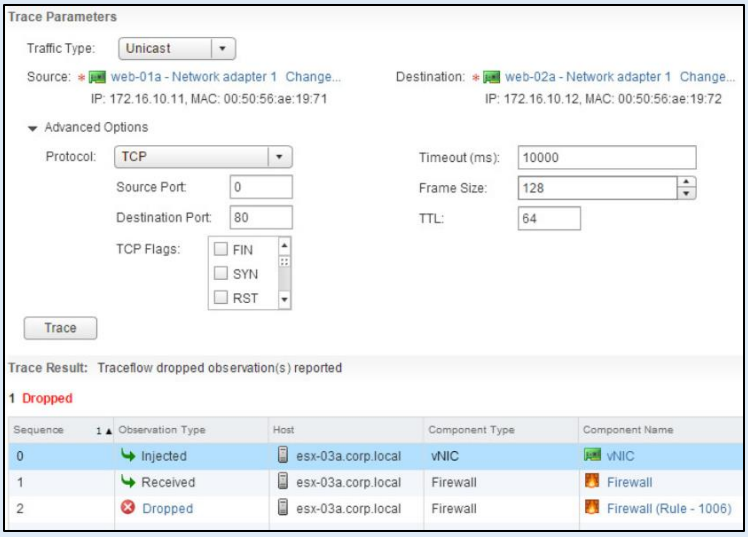
9.8.7 Testfall 7

Test # 7	VXLAN Traffic zwischen zwei unterschiedlichen ESXi Hosts nehmen nicht immer denselben Pfad.
Datum	17.11.2015 – 10.33 Uhr
Testumgebung	HSR-Lab
Ausgangslage	ESXi-1 und ESXi-2 besitzen mehrere VMs die miteinander kommunizieren. Trotz immer gleicher VTEP Adressen, sollen unterschiedliche Kommunikationspartner unterschiedliche Hash-Werte generieren können.
Wie wird getestet	Die Datenpakete werden mithilfe von Wireshark auf dem physikalischen Netzwerk analysiert.
Erwartetes Ergebnis	Der Source UDP Port sollte sich je nach Kommunikationspartner unterscheiden.
Resultat	Der UDP Port unterscheidet sich. 
Erläuterung	Der Source Port berücksichtigt die Quell- und Ziel IP-Adresse.

9.8.8 Testfall 8

Test # 8	Es spielt keine Rolle wo eine VM auf den Compute Cluster instanziiert wird. Es kann durchgängig dasselbe logische Layer 2 Netzwerk angeboten werden.
Datum	15.11.2015 – 16.23 Uhr
Testumgebung	VMware Hands-on Lab
Ausgangslage	ESXi-1 besitzt eine VM im VXLAN Segment 5001. Eine neue VM soll auf ESXi-3 in dasselbe Segment aufgenommen werden.
Wie wird getestet	Es wird ein ICMP Ping von VM1 zu VM2 und umgekehrt abgesetzt. Dabei sind keine Default Gateways konfiguriert.
Erwartetes Ergebnis	Die Kommunikation zwischen VM1 und VM2 ist problemlos möglich.
Resultat	Ein gegenseitiger ICMP Ping zwischen VM1 und VM2 ist problemlos möglich.
Erläuterung	Mithilfe von VXLAN kann ein logisches Layer 2 Netz über ein Layer 3 Netz realisiert werden.

9.8.9 Testfall 9

Test # 9	VMs im selben Subnetz lassen sich segmentieren.
Datum	15.11.2015 – 16:29 Uhr
Testumgebung	VMware Hands-on Lab
Ausgangslage	Web-01a und Web-02a befinden sich beide im selben Subnetz, 172.16.10.0/24. Die beiden Maschinen dürfen nicht miteinander kommunizieren dürfen. Dafür wurde eine DFW (Distributed Firewall) Regel erstellt. 
Wie wird getestet	Web-01a versucht auf den Webservice von Web-02a zuzugreifen. 
Erwartetes Ergebnis	VMs lassen sich segmentieren.
Resultat	Die VMs lassen keine gegenseitige Kommunikation zu.
Erläuterung	Mithilfe der Distributed Firewall lassen sich ebenfalls Verbindungen im selben Subnetz unterbinden.

9.8.10 Testfall 10

Test # 10	Segmentierte VMs im selben Subnetz erhalten keine gegenseitigen Broadcast Nachrichten.
Datum	15.11.2015 – 18:06 Uhr
Testumgebung	VMware Hands-on Lab
Ausgangslage	Web-01a und Web-02a sind im selben Subnetz auf unterschiedlichen ESX-i Hosts. Für die VMs wurde eine L2 Regel eingerichtet, die es den beiden VMs nicht gestattet, eine Kommunikation aufzubauen. Anschliessend generiert Web-01a eine Broadcast Nachricht mittels einer ARP Anfrage. 
Wie wird getestet	Nachdem sichergestellt wurde, dass auf Web-01a kein ARP-Eintrag für Web-02a existiert, wird ein ICMP Ping auf Web-02a abgesetzt. Mithilfe des Flow Monitors von NSX soll nach erlaubten oder blockierten Daten Flows gesucht werden.
Erwartetes Ergebnis	Es wird kein ARP-Request von Web-01a an Web-02a gesendet.
Resultat	Der ARP-Request wurde nicht blockiert.
Erläuterung	Die Stateful Firewalling Funktionen von NSX basieren auf L2-L4 Regeln. Das erlaubt mühelos das Segmentieren von Elementen. Leider lag beim konstruieren des Testszenario ein kleiner Denkfehler vor. Eine Broadcast-Nachricht wird selbstverständlich an die Zieladresse FF:FF:FF:FF:FF:FF versendet. Davon wäre die oben abgebildete L2 Firewallregel nicht betroffen. Und das blockieren der Broadcast-Adresse wäre unsinnig. Es würde jegliche Kommunikation verunmöglichen.

9.8.11 Testfall 11

Test # 11	Der Datenflow kann nachverfolgt werden.
Datum	16.11.2015 – 16:33 Uhr
Testumgebung	VMware Hands-on Lab
Ausgangslage	Der Client win8-01a (ESXi-3) greift auf den Webservice auf web-01a (ESXi-1) zu.
Wie wird getestet	Es wird nach einem Log im vCenter gesucht, der den Traffic Flow angeben kann.
Erwartetes Ergebnis	Es gibt ein passendes Log.
Resultat	VMware NSX stellt ein Tool namens „Flow Monitoring“ bereit. Damit lassen sich Traffic-Analysen erstellen.
Erläuterung	VMware NSX bietet verschiedene Tools um den Daten Flow genauer zu analysieren. Eine ausführlichere Beschreibung ist aus dem Abschnitt 9.6 „Monitoring Tools“ zu entnehmen.

9.8.12 Testfall 12

Test # 12	Ein Ausfall eines redundanten NSX Controller hat keinen Einfluss auf den laufenden Betrieb.
Datum	15.11.2015 – 22:25 Uhr
Testumgebung	VMware Hands-on Lab
Ausgangslage	ESXi-1 bzw. ESXi-3 besitzen je eine laufende VM im selben Subnetz.

Wie wird getestet	Es wird überprüft welcher NSX Kontroller die Master-Rolle übernimmt. Daraufhin wird dieser Kontroller auf „shutdown" gesetzt. Anschliessend soll Web-01a (ESXi-1) ein ICMP Paket an Web-02a (ESXi-3) senden. Dafür muss der ESXi Host bestimmte Informationen vom NSX Kontroller anfordern.
	`nsx-controller # show control-cluster roles`
	`Listen-IP Master? Last-Changed Count`
	`api_provider Not configured Yes 11/15 06:17:39 19`
	`persistence_server N/A No 11/15 06:17:42 15`
	`switch_manager 127.0.0.1 Yes 11/15 06:17:40 19`
	`logical_manager N/A Yes 11/15 06:17:40 19`
	`directory_server N/A Yes 11/15 06:17:40 19`
	`nsx-controller #`

Erwartetes Ergebnis	Die Rolle des Masters wird von einem anderen Kontroller übernommen. Die ESXi Anfrage bezüglich VTEP-Angabe kann problemlos aufgelöst werden.
Resultat	Die erforderliche VTEP Angabe damit ein ARP-Broadcast per Unicast aufgelöst wird, kann problemlos geliefert werden.
Erläuterung	Werden keine „keep-alive" Pakete mehr vom Master NSX Kontroller empfangen, übernimmt ein neuer Kontroller seine Rolle. Die erforderlichen VTEP Tabellen müssen aber neu aufgebaut werden.

10 Cisco ACI

10.1 Infrastruktur

Bevor die von mir gewählte Infrastruktur aufgezeigt wird, möchte ich nochmals kurz auf die einzelnen Bestandteile einer ACI Umgebung eingehen.

ACI basiert primär auf zwei Komponenten:

- **Switches der Cisco Nexus 9000 Serie**, die wahlweise im NX-OS Mode oder im ACI Mode betrieben werden. Dabei wird mit der ersten Variante eine traditionelle Architektur angestrebt und mit der zweiten Variante eine Richtlinien fokussierte Architektur.
- dem **Application Policy Infrastructure Controller (APIC)**, welche den Automatisierungs- und Managementpunkt für die Fabric-Elemente bildet.

Die ACI Switching-Architektur basiert auf einer Spine-/Leaf-Topologie. Dabei wird jeder Leaf-Knoten mit jedem Spine-Knoten erschlossen, wobei es nicht erlaubt ist, eine direkte Verbindung mit demselben Knotentyp herzustellen. Alle Server und externe Anbindungen werden via Leaf-Knoten realisiert, das heisst, die Spine-Switches erhalten nur Leaf-Switches als direkte Nachbarn.

Für meine Cisco ACI Testumgebung wurden folgende Elemente verwendet.

- 2x N9K-C9396PX
- 1x N9K-C9336PQ
- 1x APIC Controller

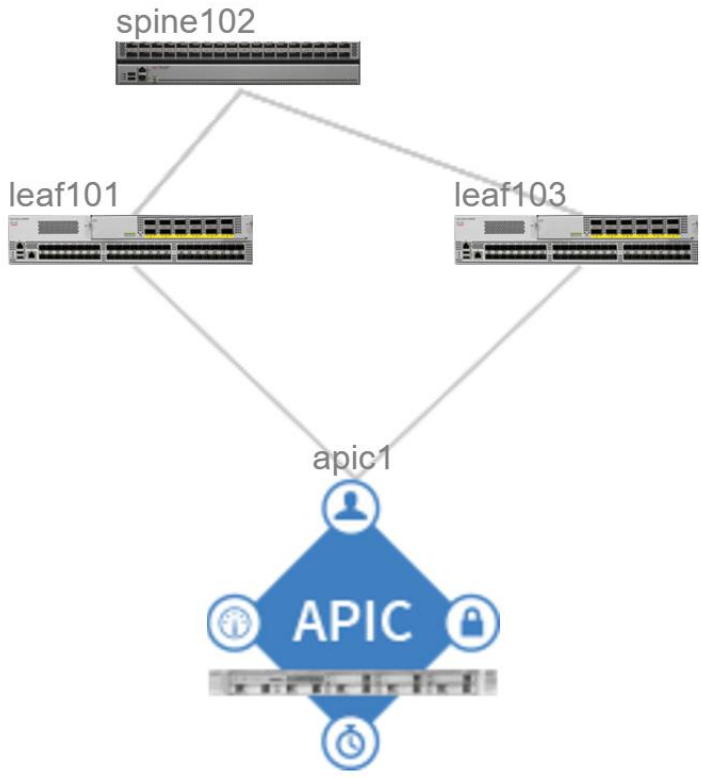


Abbildung 42: ACI Fabric

10.2 Installation ACI Fabric

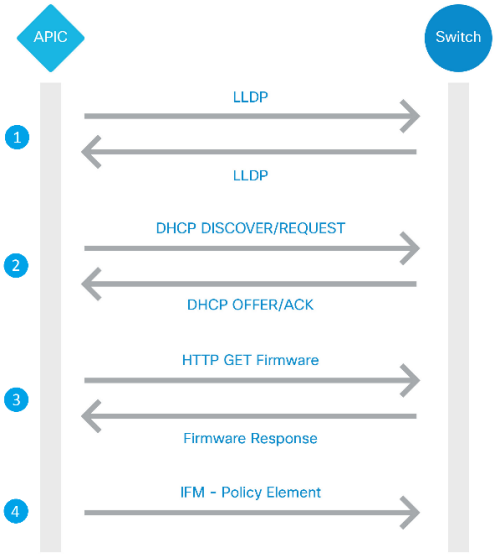
Die ACI Fabric lässt sich theoretisch in wenigen Schritten aufbauen. Nebst den passenden Images, muss lediglich die Netzwerkverkabelung im Vorfeld vorgenommen werden. Die restlichen Schritte werden anschliessend direkt vom APIC Kontroller bei der Initialisierung abgewickelt. Der Benutzer wir dabei durch einen Dialog mit den wichtigsten Parametern geführt.

```
Cluster configuration...
Press any key to continue → press a key
Enter the fabric name[ACI Fabric1] → hit enter to accept the default value
Enter the number of controller in the fabric (1-9) [3] → press 1
Enter the controller ID (1-1[1] → hit enter to accept the default value
Enter the controller name [apic] → hit enter to accept the default value
Enter address pool for TEP addresses [10.0.0.0/16] → hit enter
Enter the VLAN ID for infra network (1-4096) [4096] → enter 3967
Enter address pool for BD multicast address (GIPO) [225.0.0.0/15]: → enter
Out-of-band management configuration...
Enter the IP address [192.168.10.1/24] → enter 10.226.154.21/23
Enter the IP address of the default gateway [None]: enter 10.226.155.254
Enter the interface speed/duplex mode [auto] → hit enter
Add user configuration...
Enable strong passwords? [Y] → N
Enter the password for admin: ********
Reenter the password for admin: ********
The above configuration will be applied...
Would you like to edit the configuration? (y/n) [n]→ hit enter to finish
setup.
```

Abbildung 43: Controller Configuration

Nach der Grundkonfiguration beginnt die automatische Suche nach den einzelnen Knoten. Dabei ist der Prozess in drei Phasen gegliedert. Zu Beginn werden die direkten Leaf-Switches aufgespürt, anschliessend kommen die einzelnen Spine-Switches zum Vorschein und zuletzt werden die restlichen Leaf-Switches gesucht.

Der genaue „discovery" Vorgang kann wie folgt beschrieben werden:



1. Das Link Layer Discovery Protocol (LLDP) sucht nach einzelnen Knoten.
2. Dem Knoten wird eine Tunnel End Point (TEP) IP Adresse zugewiesen.
3. Bei Bedarf wird ein Software Upgrade durchgeführt.
4. Mithilfe von Intra-Fabric Messaging (IFM) werden Richtlinien gesetzt.

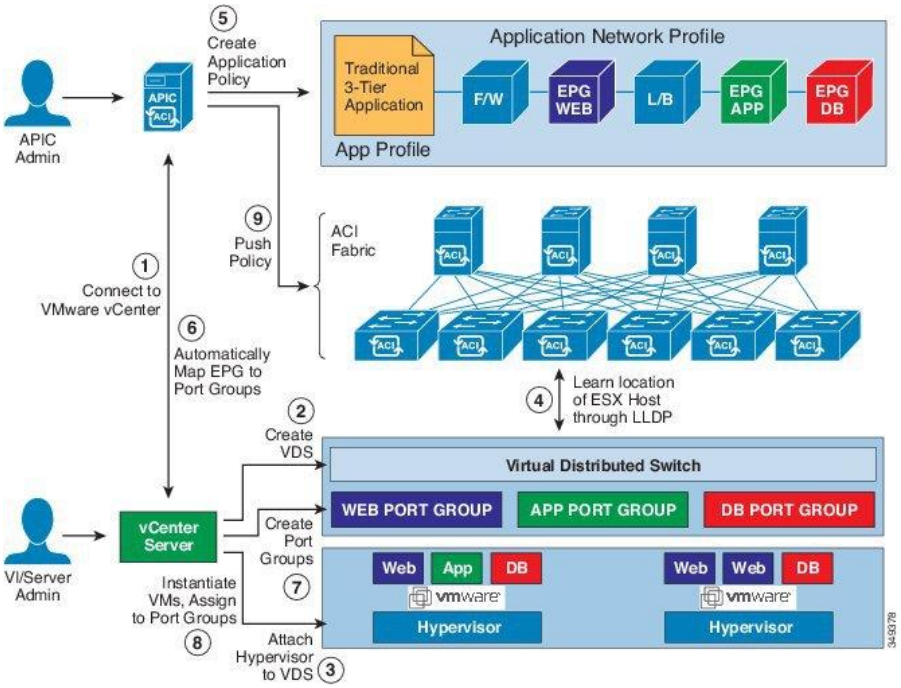
[18]*Abbildung 44: Discovery Process*

[18] http://aci-troubleshooting-book.readthedocs.org/en/latest/_images/APIC-Switch.jpg

10.3 VMM Integration

VMM steht für Virtual Machine Manger und wird von der ACI dazu benutzt, um Managementsysteme wie das VMware vCenter oder das Microsoft SCVMM zu integrieren. An dieser Stelle möchte ich kurz erwähnen, dass es mir in meiner Arbeit nicht möglich war, das vCenter mit der ACI Fabric zu verbinden. Auch mithilfe von mehreren Netzwerkspezialisten konnte der genaue Fehler nicht lokalisiert werden. Dennoch erachte ich es als wichtig, auf die Details des VMM einzugehen.

Grundsätzlich führt die Konfiguration einer VMM Domain zu einer Verbindung zwischen dem APIC und dem vCenter Server. Dabei wird automatisch ein vSphere Distributed Switch erstellt, welcher von der ACI verwaltet wird. Der genaue Workflow kann wie folgt beschrieben werden:



[19]Abbildung 45: vCenter Domain Workflow

1. Die APIC verbindet sich mit dem vCenter.
2. Es wird ein vSphere Distributed Switch erstellt.
3. Die Ports des Hypervisor werden dem neuen VDS zugewiesen. Wobei die Uplink Ports an die Leaf-Switches angeschlossen sind.
4. Mittels LLDP oder CDP werden die einzelnen Lokationen der VMs erlernt.
5. Es werden Endpoint Groups für die VMs erstellt.
6. Die EPG Policy wir der VMM Domain zugewiesen.
7. Für die einzelnen EPGs werden Port Groups auf dem VDS erstellt.
8. Die einzelnen VMs werden den Port Groups zugewiesen.
9. Nachdem die APIC über alle VM Standorte und Einstellungen Bescheid weiss, werden die Richtlinien auf die Fabric aufgespielt.

[19] http://www.cisco.com/c/dam/en/us/td/i/300001-400000/340001-350000/349001-350000/349378.eps/_jcr_content/renditions/349378.jpg

10.4 Technische Umsetzung

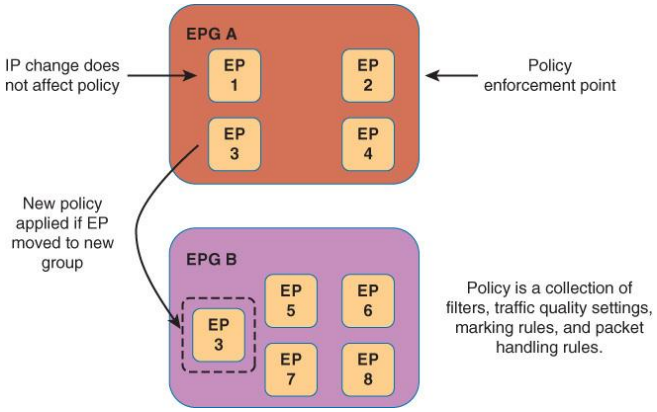
Um die Vorgänge und Richtlinien innerhalb einer Cisco ACI Fabric besser beschreiben zu können, werden nachfolgend die wichtigsten Begriffe und Funktionen genauer erläutert.

10.4.1 Endpoint Groups

Die Endpoint Groups (EPGs) stellen eine Sammlung von ähnlichen Endpunkten dar und repräsentieren ein „Application Tier" oder ein „Set of Services". Diese logische Gruppierung wurde daher gewählt, um Objekte mit ähnlichen Richtlinien effizient abbilden zu können. Dabei werden EPGs beispielweise nach folgenden Kriterien definiert:

- Objekte mit derselben VLAN-ID
- Objekte die sich im selben Subnetz befinden → 172.16.10.0/24
- Objekte mit demselben DNS Suffix → *.company.com

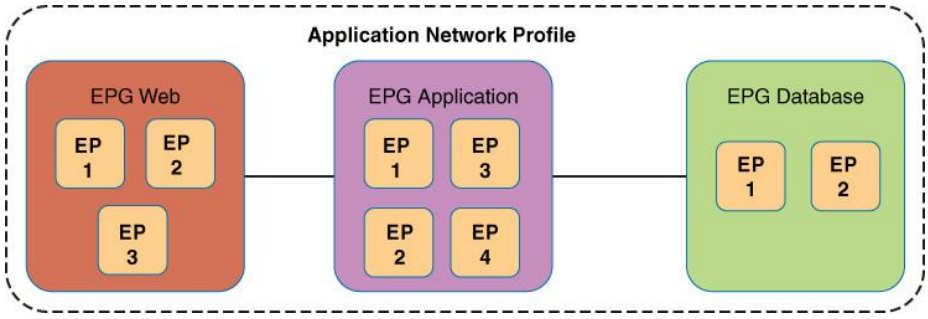
Im Endeffekt werden EPGs erstellt um später darauf Richtlinien anzuwenden. Richtlinien sind daher nicht von IP-Adressen abhängig, sondern von der EPG. Darum können problemlos IP-Adressen von Endpoints verändert werden, ohne dass zusätzliche Anpassungen an der Policy anfallen.



[20]*Abbildung 46: Relationship between EPGs and Policies*

10.4.2 Application Network Profiles

Application Network Profiles (ANP) beinhalten eine oder mehrere Endpoint Groups und formen eine Gesamtapplikation. Dabei werden Profile ausgehend von den Bedürfnissen nach Sicherheit, Performance oder bestimmten Kommunikationspfaden definiert. Als Beispiel könnte ein Onlineshop abgebildet werden, welcher aus drei Endpoint Groups besteht.



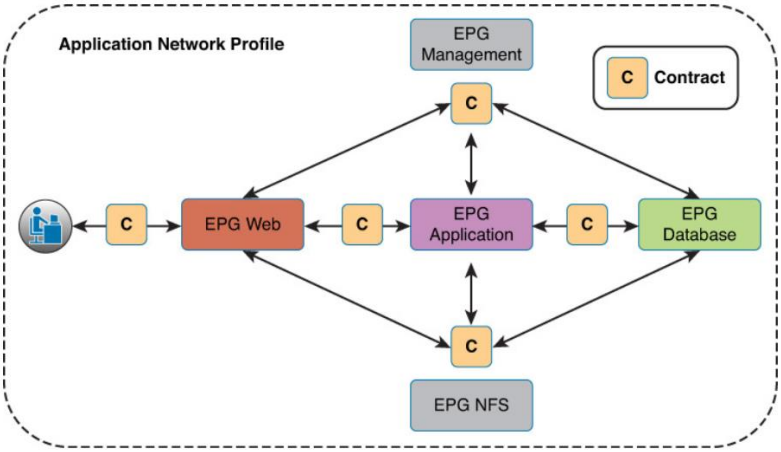
[21]*Abbildung 47: Application Network Profile*

[20] Cisco Press: The Policy Driven Data Center with ACI: Architecture, Concepts, and Methodology

[21] Cisco Press: The Policy Driven Data Center with ACI: Architecture, Concepts, and Methodology

10.4.3 Contracts, Subjects, and Filters

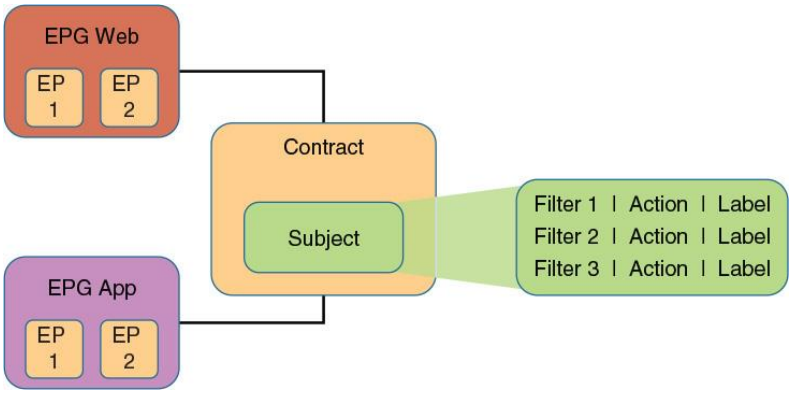
Contracts definieren Regeln für ein- und ausgehenden Datenverkehr, QoS, Weiterleitungen und Service Graphs. Abhängig von den Bedürfnissen, erlauben Contracts, wie einzelne EPGs mit anderen EPGs kommunizieren dürfen. Falls keine expliziten Contracts gesetzt wurden, darf nur innerhalb einer Endpoint Group und einem Applikation Network Profile kommuniziert werden. Alle restlichen Kommunikationsversuche werden standardmässig blockiert.



[22]*Abbildung 48: Application with Contracts*

Die Abbildung 48 zeigt eine klassische Webapplikation, die auf drei Tiers aufgeteilt ist. Dabei ist zu erkennen, dass Richtlinien sowohl unidirektional sowie bidirektional gesetzt werden können. Beispielsweise wird für das Management und das Network File System (NFS) eine Kommunikation in beide Richtungen angestrebt.

Contracts haben grundsätzlich einen Namen/Label, einen bestimmten Filter und eine Aktion. Die Kombination dieser drei Bestanteile wird dann wiederum Subject genannt. So können innerhalb eines Contracts diverse Subjects definiert werden.



[23]*Abbildung 49: Subjects Within Contract*

Filter basieren auf Layer 2 – Layer 4 Felder und klassifizieren den Datenverkehr. Bei der Aktion kann wahlweise zwischen folgenden Optionen gewählt werden: Permit, Deny, Mark, Redirect, Log und Copy. Die Labels sind optional und helfen Objekte noch besser zu gruppieren.

[22] Cisco Press: The Policy Driven Data Center with ACI: Architecture, Concepts, and Methodology

[23] Cisco Press: The Policy Driven Data Center with ACI: Architecture, Concepts, and Methodology

10.4.4 Tenants, Contexts, and Bridge Domains

Die oberste Ebene des APIC Richtlinienmodels bilden die Tenants. Tenants ermöglichen eine Trennung von Datenflüssen und der zu verwalteten Netzwerkinfrastrukturen. Im Grunde genommen handelt es sich um einen logischen Container, welcher für Kunden, Geschäftsbereiche oder andere Gruppierungen verwendet werden kann.

Tenants können entweder voneinander isoliert werden oder sie benutzen gemeinsame Ressourcen. Die wichtigsten Elemente eines Tenants sind Contexts, Filters, Contracts, Outside Networks, Bridge Domains und Application Network Profiles. Ein einzelner Tenant kann ein oder mehrere Virtual Routing and Forwarding (VRF) Instanzen, in ACI Contexts genannt, beherbergen. Wobei ein Context wiederum verschiede Bridge Domains aufnehmen kann.

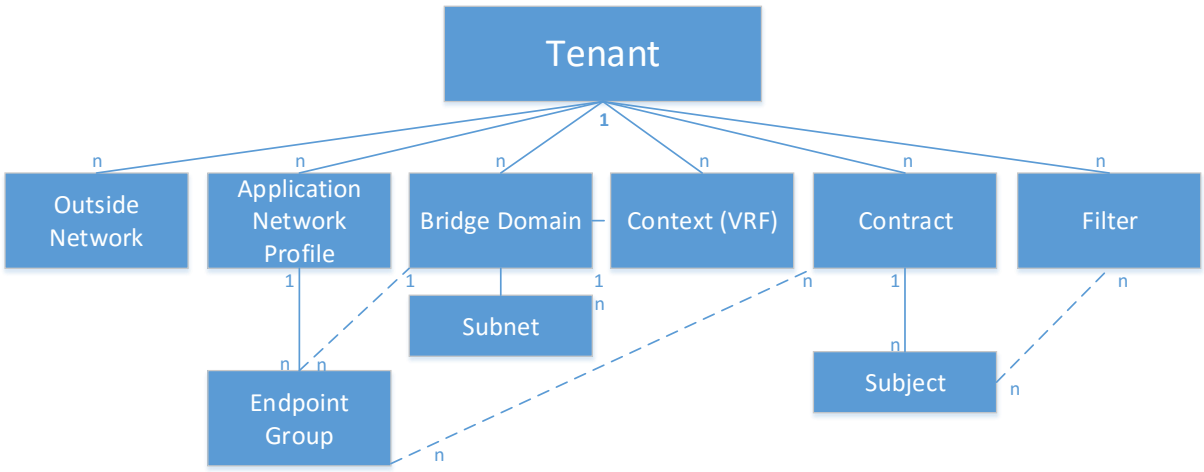


Abbildung 50: APIC Policy Model

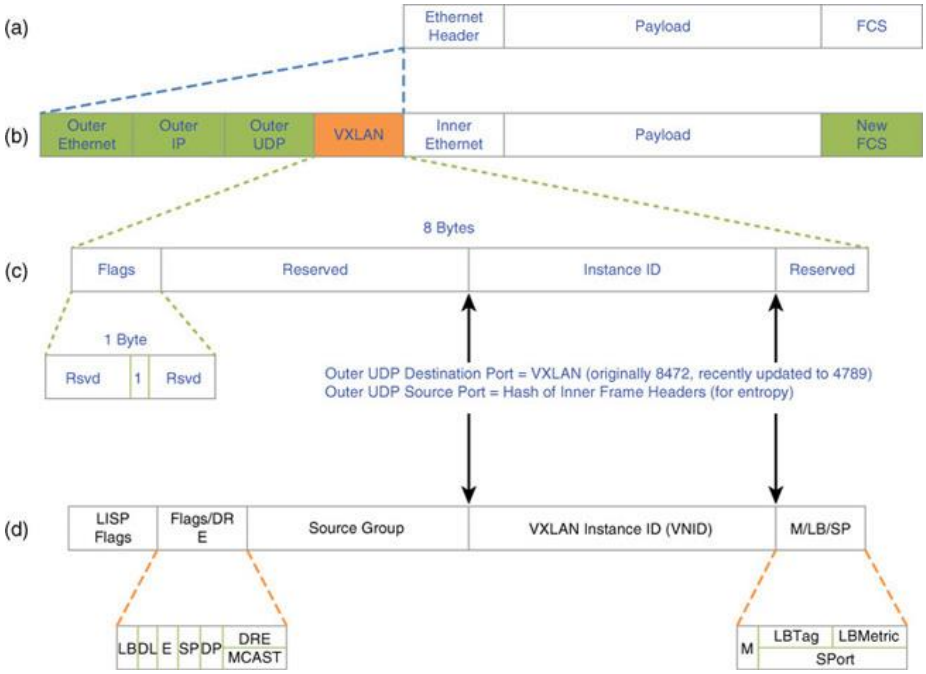
Bei einer Bridge Domain handelt es sich um etwas Ähnliches wie ein VLAN. Es stellt einen Container für ein Subnetz zur Verfügung. Dabei ist zu beachten, dass ein Container mindestens ein Subnetz besitzen muss.

10.4.5 Overlays in ACI

Die Weiterleitung in ACI basiert auf VXLAN, wobei kleine Änderungen am ursprünglichen Protokoll vorgenommen wurden. Für gewöhnlich nutzt VXLAN Multicast für Floating Pakete wie es bei Broadcast, Unknown Unicast und Multicast vorkommt. ACI dagegen möchte weitgehend auf Multicast verzichten und setzt für das Lernen und Finden von Endpunkten auf ein Konzept, dass sich ähnlich wie Locator/ID Separation Protocol (LISP) verhält. Dabei wird eine Datenbank mit Endpunkten aufgebaut.

Der VXLAN Header von ACI, eVXLAN genannt, bietet einen Tagging Mechanismus, um die Eigenschaften der ACI Fabric Frames zu identifizieren. eVXLAN erweitert das LISP Feld mit zusätzlichen Informationen wie Policy Group, Load- und Path-Metric, Counter und Encapsulation.

Die nachfolgende Abbildung zeigt das Frame-Format, wie es von der ACI Fabric verwendet wird. Teil a zeigt dabei das ursprüngliche Ethernet-Frame wie es vom Endsystems versendet wird. Teil b zeigt das ursprüngliche Frame mit dem VXLAN Header. Im Teil c wird das Format eines klassischen VXLAN Header abgebildet und Teil d zeigt die Erweiterung des ACI VXLAN Formats.



[24]Abbildung 51: ACI VXLAN Frame

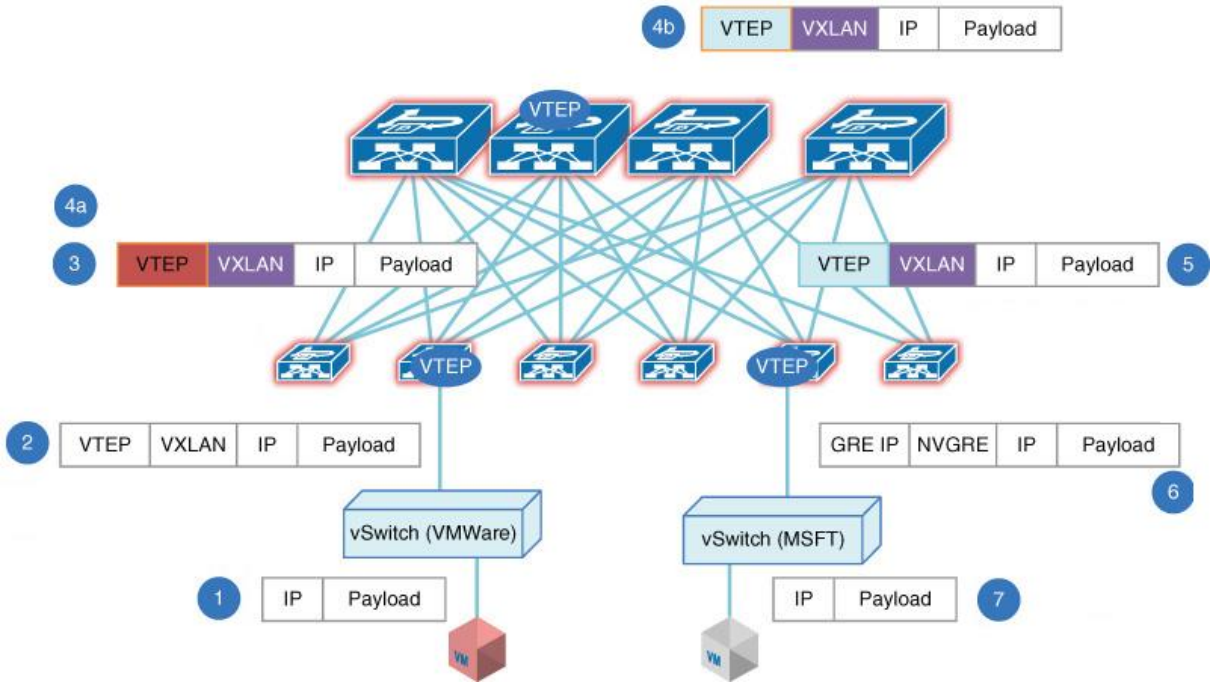
10.4.6 Pervasive Gateway

Die Konfiguration von Hot Standby Router Protocol (HSRP) oder Virtual Router Redundancy Protocol (VRRP) wird mit ACI hinfällig. Die ACI Fabric benutzt ein Konzept namens Pervasive Gateway. Das Default Gateway eines Subnetzes wird auf allen Leaf-Switches zur Verfügung gestellt. Der Vorteil ist eine vereinfachte Kommunikation, da jedes Top of Rack (ToR) Gerät die Rolle des Default Gateways übernimmt.

[24] Cisco Press: The Policy Driven Data Center with ACI: Architecture, Concepts, and Methodology

10.4.7 Packet Flow

ACI lernt fortlaufend die MAC oder IP Adresse der einzelnen Endpunkte und kann sie anhand der VXLAN Network Identifier (VNID) den einzelnen VTEP zuordnen. Anhand des folgenden Beispiels soll der Ablauf eines Datenflusses detailliert erläutert werden:



[25]*Abbildung 52: ACI Packet Forwarding*

1. Es wird ein Paket versendet, dass entweder von einer VM oder direkt von einem physikalischen Server erstellt wurde.
2. Falls ein vSwitch an der Paketweitergabe beteiligt ist, wird dementsprechend das Paket mit einem VXLAN Header versehen und an den Leaf-Switch weitegeleitet.
3. Der Leaf-Switch ist in der Lage VLAN, VXLAN oder NVGRE gekapselte Pakete entgegenzunehmen und die wichtigsten Bestanteile in das eigene eVXLAN einzubinden.

4a. Falls für die Ziel IP-Adresse ein passender VTEP bekannt ist, wird das Paket direkt an den entsprechenden Spine-Switch bzw. Leaf-Switch weitergesendet.

4b. Falls der Leaf-Switch in seiner lokalen Datenbank keinen Eintrag für das Ziel findet, wird als Ziel ein Spine-Switch angegeben. Der Spine-Switch erhält das Paket und findet dank seiner globalen Datenbank die Ziel VTEP Adresse. Der Paketheader wird demensprechend angepasst, wobei die Quell VTEP Adresse nicht verändert wird.

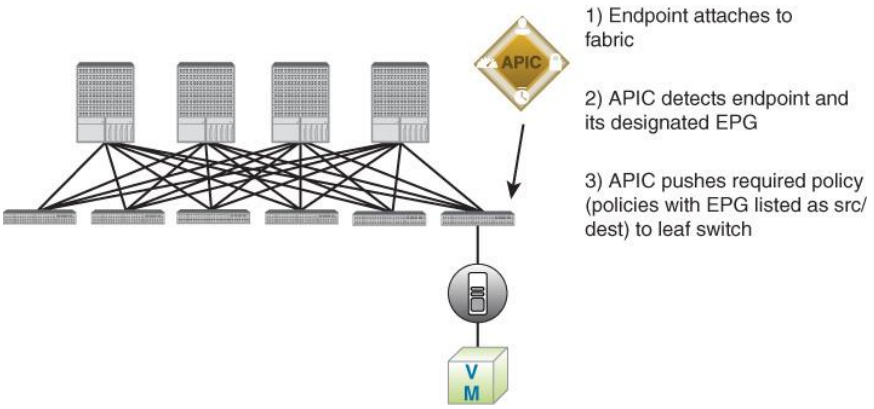
5. Der Leaf-Switch entfernt die eVXLAN Informationen und setzt die passende NVGRE Einstellungen.
6. Das Paket wird dem vSwitch weitergeleitet.
7. Der vSwitch sendet das Paket an die VM oder an den physikalischen Server.

[25] Cisco Press: The Policy Driven Data Center with ACI: Architecture, Concepts, and Methodology

10.4.8 Policy Enforcement

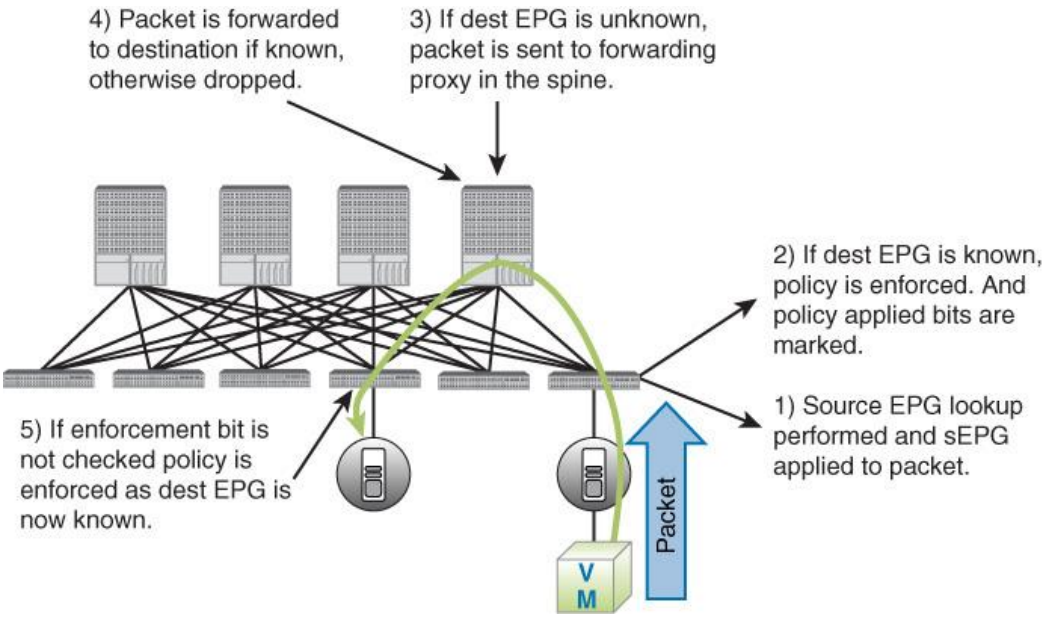
Richtlinien können grundsätzlich an zwei Orten platziert werden. Entweder am Ein - oder Ausgang eines Leaf-Knotens. Dabei ist die erste Variante nur möglich, falls die Endpoint Group des Ziels bekannt ist.

Dadurch, dass sich die aktuellen Richtlinien immer auf den Leaf-Knoten befinden, muss für eine Entscheidung niemals der Kontroller angefragt werden.



[26]*Abbildung 53: Applying Policy to Leaf Nodes*

Falls die Ziel EPG nicht bekannt ist, wird das Paket speziell markiert und dem Forwarding Proxy, dabei handelt es sich um einen Spine-Switch, weitergeleitet. Der Spine-Proxy hat Kenntnis über alle Netze und leitet nach Möglichkeit das Paket an den entsprechenden Leaf-Switch weiter, andernfalls wird das Paket verworfen. Am Ziel Switch angekommen überprüft der Leaf-Knoten das Paket nach einer Markierung und kann feststellen, ob eine Überprüfung der Richtlinien bereits stattgefunden hat. Falls dem nicht so ist, konsultiert der Leaf-Knoten seine eigene Richtliniendatenbank und entscheidet was mit dem Paket geschieht.



[27]*Abbildung 54: Enforcing Policy on Fabric*

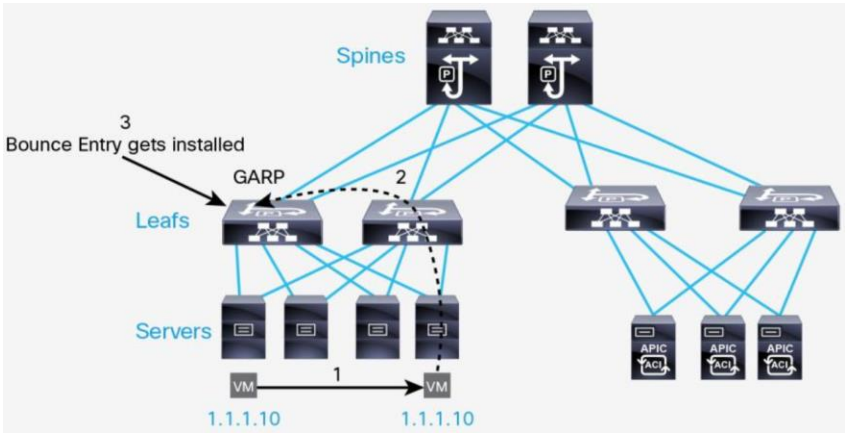
[26] Cisco Press: The Policy Driven Data Center with ACI: Architecture, Concepts, and Methodology

[27] Cisco Press: The Policy Driven Data Center with ACI: Architecture, Concepts, and Methodology

10.4.9 VM Migration

Dieser Abschnitt soll aufzeigen, wie ACI bei einer VM Migration vorgeht. Dabei muss einerseits sichergestellt werden, dass es zu keinem Verbindungsunterbruch kommt und dass der Verkehrsfluss anschliessend weiterhin optimal gewählt wird.

Das Verhalten bei einer VM Migration von einem Leaf-Switch(VTEP) zu einem anderen Leaf-Switch(VTEP) kann wie folgt beschrieben werden:



[28]*Abbildung 55: VM Migration in ACI*

1. Sobald die VM erfolgreich migriert wurde, wird eine GARP Nachricht versendet.
2. Der direkt angeschlossene Leaf-Switch leitet die GARP Nachricht an den ursprünglichen Leaf-Switch weiter und sendet ein Update an die zentralisierte Mapping Datenbank (Spine-Proxy).
3. Der ursprüngliche Switch markierte die VM IP-Adresse als Bounce-Eintrag.
4. Jeglicher Traffic an die VM kann nun vom ursprünglichen Switch an den neuen Switch weitergeleitet werden.
5. Mit der Zeit aktualisieren alle Leaf-Switches ihre Forwarding Tabelle. Das kann entweder mit einer Abfrage an den Spine-Proxy geschehen oder sobald der Leaf-Switch das erste Mal ein Antwortpaket der migrierten VM bekommt.

[28] http://www.cisco.com/c/dam/en/us/solutions/collateral/data-center-virtualization/application-centric-infrastructure/guide-c07-736077.doc/_jcr_content/renditions/guide-c07-736077_15.jpg

10.5 Testszenarien

Die Testszenarien konnten nicht wie gewünscht durchgeführt werden. Es war nicht möglich eine ACI Fabric samt vCenter Integration aufzusetzen. Aus diesem Grund wird ausschliesslich das zu erwartende Ergebnis beschrieben.

10.5.1 Testfall 1

Test # 1	Das Implementieren von Multi-Tier Services kann innert wenigen Minuten anstelle von Tagen erfolgen.
Ausgangslage	Für einen Neukunden soll eine 3-Tier Applikation bereitgestellt werden. Dabei werden bereits vorkonfigurierte Server verwendet. Folgende Komponenten müssen neu konfiguriert werden: • 1x Tenant • 1x Private Network • 3x Bridge Domains • 1x Physical Domain • 1x Application Profile
Wie wird getestet	Sämtliche Einstellungen werden über das Web-GUI vorgenommen. Der Ablauf erfolgt sequenziell.
Erwartetes Ergebnis	Die Implementierung dauert nicht länger als 20 Minuten.
Erläuterung	Der zeitliche Aufwand für einen Neukunden konnte mittels ACI bereits drastisch reduziert werden. Dennoch fallen noch zu viele manuelle Schritte an. Besser wäre es, man würde über die zur Verfügung gestellte API nutzen und die Prozesse optimieren. Passende Lösungen werden beispielsweise von OpenStack angeboten.

10.5.2 Testfall 2

Test # 2	Das Verschieben einer VM Instanz mit 4GB RAM und zentralisiertem Datenspeicher, darf nicht länger als 300 Sekunden dauern.
Ausgangslage	Es existieren zwei unterschiedliche Compute Cluster die sich in einem dedizierten vMotion Netzwerk befinden. Die Datenrate beträgt max. 10Gbit/s.
Wie wird getestet	Eine aktive VM Instanz wird via vCenter auf einen anderen Cluster migriert. Es werden die Standardeinstellungen des Assistenten verwendet. Lediglich die Switchangabe für die VM muss neu gesetzt werden.
Mögliche Auswirkungen	Es gibt einen Verbindungsunterbruch bei der VM. -> Test #3 Der Verkehrsfluss ist nicht mehr optimal. -> Test #4 Zugriffsregeln werden nicht migriert. -> Test #5
Erwartetes Ergebnis	Die Migrationszeit beträgt weniger als 300 Sekunden.
Erläuterung	vMotion benutzt für gewöhnlich einen eigenen VMkernel Adapter, der sich in einem dedizierten Subnetz befindet. Dies kann auch weiterhin so genutzt werden. ACI erstellt bei der Integration des vCenters einen Distributed Switch (VDS) bereit. Anschliessend wird eine neue Port-Group definiert und einer EPG zugewiesen. Die Geschwindigkeit von vMotion ist im Endeffekt von den Bandbreiten der Fabric abhängig.

10.5.3 Testfall 3

Test # 3	Das Verschieben einer VM Instanz führt nicht zu einem Unterbruch des Services.
Ausgangslage	Der Mandant besitzt einen Webserver, der einen Service nach Aussen anbietet. Aus Performance- oder Wartungsgründen kann es vorkommen, dass die VM Instanz von einem Compute Cluster auf einen anderen Cluster verschoben wird. Der Unterbruch des Services darf für einen aussenstehenden Benutzer nicht spürbar sein.
Wie wird getestet	Eine aktive VM Instanz wird via vCenter auf einen anderen Cluster migriert. Während der gesamten Migration wird sowohl die interne sowie die öffentliche IP-Adresse gepingt.
Erwartetes Ergebnis	Es gibt keinen Unterbruch.
Erläuterung	Sobald die Migration erfolgreich abgeschlossen ist, sendet die VM eine GARP Nachricht an seinen neuen Leaf-Switch. Dieser informiert wiederum den alten Leaf-Switch über die Änderung und richtet temporär eine Weiterleitung ein. Erhält der alte Leaf-Switch aufgrund von veralteten Cache-Einträgen Pakete für den Webserver, wird das Paket an den neuen Leaf-Switch weitergeleitet. Somit kommt es zu keinem Unterbruch des Services.

10.5.4 Testfall 4

Test # 4	Der Traffic Flow bleibt auch nachdem verschieben einer VM optimal.
Ausgangslage	VMs werden aus Wartungs- oder Performancegründen regelmässig auf unterschiedliche ESXi Hosts verschoben. Dabei muss weiterhin der bestmögliche Pfad für den Verkehrsfluss gewähreistet sein. Die Testumgebung besteht aus VM1 (172.16.10.10) auf ESXi-4 (192.168.250.101) und VM2 (172.16.10.11) auf ESXi-3 (192.168.250.100).
Wie wird getestet	Eine aktive VM Instanz wird via vCenter auf einen anderen Cluster migriert. Dabei wird der Traffic Flow vor und nach der Migration analysiert. Der Test wird mittels ICMP Ping durchgeführt. Der Ziel Host von VM1 wird ESXi-2 (192.168.150.101) sein.
Erwartetes Ergebnis	Für einen kurzen Moment werden die Pakete an einen falschen Leaf-Switch geschickt, wobei sich das Problem mit dem ersten Response Paket löst.
Erläuterung	Nach der VM Migration verwenden die Leaf-Switches weiterhin die veralteten Cache-Einträge.

10.5.5 Testfall 5

Test # 5	Zugriffsregeln werden mit vMotion problemlos mitmigriert.
Ausgangslage	Es existieren 2 VMs (Web01 und Web02), die sich in einer Endpoint Group befinden. Die Contracts wurden so konfiguriert, dass ein ICMP Ping von Web01 zu Web02 nicht erlaubt ist.
Wie wird getestet	Die Contracts werden gemäss Ausgangslage vorkonfiguriert. Anschliessend wird die VM Web01 vom einem ESXi Cluster auf einen anderen Cluster verschoben. Dabei sollte auf dem ESXi Host ersichtlich sein, dass die Zugriffsregeln erhalten bleiben.
Erwartetes Ergebnis	Die Zugriffregeln bleiben bestehen.
Erläuterung	Contracts werden zwischen unterschiedlichen EPGs definiert. Dabei spielt es keine Rolle, wo sich eine VM befindet.

10.5.6 Testfall 6

Test # 6	Der Datenverkehr zwischen unterschiedlichen EPGs können priorisiert werden.
Ausgangslage	Es existieren zwei EPGs, welche ihrerseits mehrere VMs auf verschiedenen Clustern haben. Dabei wurde im Contract ein QoS Wert gesetzt.
Wie wird getestet	Der Datenverkehr zwischen unterschiedlichen EPGs wird mittels Wireshark analysiert.
Erwartetes Ergebnis	Die Datenpakete erhalten im Header eine QoS Markierung
Erläuterung	ACI bietet die Möglichkeit mithilfe von Contracts L2 oder L3 Markierungen zwischen EPGs zu setzten.

10.5.7 Testfall 7

Test # 7	Es spielt keine Rolle wo eine VM instanziiert wird. Es kann durchgängig dasselbe logische Layer 2 Netzwerk angeboten werden.
Ausgangslage	ESXi-1 und ESXi-2 werden an dieselbe Physical Domain angeschlossen und besitzen eine VM namens VM1 bzw. VM2.
Wie wird getestet	Es wird ein ICMP Ping von VM1 zu VM2 und umgekehrt abgesetzt. Dabei sind keine Default Gateways konfiguriert.
Erwartetes Ergebnis	Die Kommunikation zwischen VM1 und VM2 ist problemlos möglich.
Erläuterung	Mithilfe von VXLAN kann ein logisches Layer 2 Netz über ein Layer 3 Netz realisiert werden.

10.5.8 Testfall 8

Test # 8	VMs im selben Subnetz lassen sich segmentieren.
Ausgangslage	Web-01a und Web-02a befinden sich beide im selben Subnetz, 172.16.10.0/24. Die beiden Maschinen dürfen nicht miteinander kommunizieren. Dafür wurden unterschiedliche EPGs eingerichtet und mit Contracts abgesichert.
Wie wird getestet	Web-01a versucht auf den Webservice von Web-02a zuzugreifen.
Erwartetes Ergebnis	VMs lassen sich segmentieren.
Erläuterung	Mithilfe von Contracts lassen sich Zugriffe steuern. Leider müssen die VMs auf unterschiedliche EPGs aufgeteilt werden, da Regeln nur zwischen EPGs definieret werden können.

10.5.9 Testfall 9

Test # 9	Der Datenflow kann nachverfolgt werden.
Ausgangslage	Der Client win8-01a (ESXi-3) greift auf den Webservice auf web-01a (ESXi-1) zu.
Wie wird getestet	Es wird nach einem Log im APIC Manager gesucht, der den Traffic Flow angeben kann.
Erwartetes Ergebnis	Es gibt ein passendes Log.
Erläuterung	-

10.5.10 Testfall 10

Test # 10	Ein Ausfall eines redundanten APIC Controller hat keinen Einfluss auf den laufenden Betrieb.
Ausgangslage	ESXi-1 bzw. ESXi-3 besitzen je eine laufende VM im selben Subnetz.
Wie wird getestet	Ein aktiver APIC Kontroller wird auf „shutdown“ gesetzt. Anschliessend soll Web-01a (ESXi-1) ein ICMP Paket an Web-02a (ESXi-3) senden.
Erwartetes Ergebnis	Der Ausfall eines Kontrollers hat keinen Einfluss auf den laufenden Betrieb.
Erläuterung	Die Kontroller werden lediglich dazu gebraucht, um Änderungen auf die Fabric aufzuspielen. Bestehende Regeln sind stets lokal auf der Fabric und arbeiten unabhängig vom APIC.

11 Ergebnisse

11.1 Funktioneller Vergleich

VMware NSX ist eine rein softwarebasierte Overlay Netzwerk Lösung. Alle Komponenten sind Softwareelemente und laufen entweder im Hypervisor Kernel oder als virtuelle Maschine. Aus diesem Grund muss das physikalische Netzwerk separat betrachtet werden. Es ist daher wichtig anzumerken, dass die Überwachung und Administration der physikalischen Netzwerkkomponenten nicht von NSX übernommen werden kann.

Auf der anderen Seite haben wir Cisco ACI. Mit der ACI wird eine vollständige SDN Lösung angeboten. Die RESTful API kann sowohl Einstellungen an der Private Cloud sowie an einzelnen Netzwerkelementen vornehmen. Daher ist ein direkter Vergleich zwischen den beiden Lösungen nicht möglich. Dennoch besitzen beide Produkte überlappende Funktionen. So werden in ihrem Overlay Netzwerk Services von L2 bis L7 angeboten. Teilweise müssen sie aber in Verbindung mit einem Drittthersteller realisiert werden. Konkret bietet NSX eine integrierte Firewall-, VPN-Terminierung, NAT- und Load Balancer Funktion, die es bei der ACI so nicht gibt. Dafür unterstützt die aktuelle Version von NSX nur virtuelle Service Appliance von ganz bestimmten zertifizierten Partnern.

Um externen Geräten in NSX den Zugriff ins Overlay Netzwerk zu ermöglichen, muss zwingenderweise ein Edge Service in Form eines Edge Service Gateways erstellt werden. Das löst ACI eleganter und benötigt dafür keine speziellen VMs. Es können direkt Komponenten an die Leaf-Switches angeschlossen werden.

11.2 Operative Aspekte

11.2.1 Einfachheit

Für die Bewertung der Einfachheit haben beide Produkte dieselbe Ausgangslage. Ich kannte zuvor weder die ACI Lösung von Cisco noch NSX von VMware. Ebenfalls fehlten mir grossartige Erfahrungen mit den vSphere Produkten. Ich versuchte mein Wissen grösstenteils von den Onlinedokumentationen der Hersteller sowie der Online Community anzueignen.

Beim ersten Produkt VMware NSX gelang die Informationsbeschaffung erstaunlich gut. Nebst vielen Marketingbroschüren werden detaillierte Design Guides angeboten. Weiter werden technische Aspekte ausführlich erläutert und es werden detaillierte Installations- und Konfigurationsanleitungen in mehreren Sprachen zur Verfügung gestellt. Die spätere graphische Menüführung, die ins vCenter integriert wurde, überzeugt durch eine klare und intuitive Struktur.

Ganz anders sieht es mit dem Produkt von Cisco aus. Zu Beginn wurden viele Versprechungen gemacht, leider konnten nur die wenigsten eingehalten werden. Obwohl die zur Verfügung stehenden Dokumentationen ziemlich zahlreich sind, gelang es nicht, eigenständig eine komplette ACI Umgebung samt vCenter-Einbindung in Betrieb zu nehmen. Das Vorhaben scheiterte anfänglich bereits an der ACI Fabric. So einfach wie angepriesen werden die Leaf und Spine Switches bei der Initialisierung des APIC Kontrollers nicht gefunden. Einzelne Knoten mussten unzählige Male neugestartet werden, bis sich die Fabric selbst aufbaute. Der nächste Schwachpunkt ist das sehr komplexe und überhaupt nicht benutzerfreundliche Web-GUI. Viele Konfigurationen sind derart komplex verschachtelt, dass einem ein schneller Einstieg ins Produkt verwehrt bleibt. Ebenfalls sind die Begrifflichkeiten oftmals schlecht gewählt und weitgehend nicht selbsterklärend.

11.3 Skalierung

Beide Systeme sind dazu ausgelegt hunderte von Mandanten zu verwalten. Die Skalierung ist zum grössten Teil eine reine Kosten- bzw. Architekturfrage. Dennoch möchte ich die wichtigsten Eckdaten aufzeigen:

	[29]Cisco ACI Release 1.1(1j)
APIC Kontroller	3
Leaf-Switches	80
Spine-Switches	6
Physische Server	3'600
Tenants	1'000
Layer 3 Contexts	1'000
Contracts/Filters	1'000 Contracts 10'000 Filters
Endpunkte	180'000

[30]*Tabelle 4: Cisco ACI Facts*

	VMware NSX 6.1
Kontroller	3
Hosts pro Cluster	32
Virtuelle Maschinen	>100'000
Hosts pro Transport Zone	256
Logical Switches	10'000
Regeln pro NSX Manager	50'000

Tabelle 5: VMware NSX Facts

11.4 Performance

Eine Aussage über die Performance kann aus mehreren Gründen nur sehr schwer gemacht werden. Zum einen wurden keine spezifischen Performancemessungen durchgeführt und zum anderen Unterschied sich die Hardware zu stark voneinander. Subjektiv würde ich aber behaupten, dass sich im Punkto Bedienung das Cisco APIC Web-GUI besser geschlagen hat als das vCenter Web-GUI. Das könnte unter anderem auch daran liegen, dass das eine auf HTML5 basiert und das andere auf Flash.

11.5 Einsatzgebiete

Beide Produkte sind darauf spezialisiert, Arbeitsabläufe innerhalb eines Data Centers zu beschleunigen und zu vereinfachen. Das kann entweder für interne Zwecke genutzt werden oder man bietet seine Infrastruktur als Dienstleitung in Form eines Cloud Dienstes an. Beliebt sind dabei Services im Bereich IaaS oder PaaS.

11.6 Technische Umsetzung

VMware NSX überzeugt mit ihrer Lösung der Micro Segmentierung von VMs. Der Verkehr kann bereits vor dem Verlassen des ESXi-Host überprüft werden und verbraucht dank ESXi-Kernel Implementierung fast keine Ressourcen. Auf Basis von zahlreichen Objekten lässt sich eine VM problemlos von einer anderen VM segmentieren. Schaut man in Richtung ACI stellt man fest, dass die Überprüfung zu einem späteren Zeitpunkt stattfindet und nur auf Basis von Endpoint Groups

[29] Mittelgrosse L3 Fabric; die Angaben stellen nicht das theoretische Maximum dar.

[30] http://www.cisco.com/c/en/us/td/docs/switches/datacenter/aci/apic/sw/1-x/verified-scalability/b_Verified_Scalability_Release_1_1_1j.html

gemacht werden kann. Das schränkt die Flexibilität ein und ist mit einem höheren Aufwand verbunden. Wünschenswert wäre ebenfalls eine Segmentierung auf VM-Ebene.

Betrachtet man die Aktualisierung der VTEP Tabelle nach einer VM Migration, ist interessant, wie beide Produkte dabei vorgehen. Dadurch dass NSX keinen Einfluss auf das physikalische Netzwerk ausübt, muss nach einer Änderung zwingenderweise sofort der ESXi-Host reagieren. Anders wie gewohnt, wird der NSX-Controller in diesem Prozess nicht involviert. Man könnte erwarten, dass eine Änderung via Kontroller an alle relevanten Geräte forciert wird. Dem ist aber nicht so. Der ESXi-Host informiert direkt seine anderen ESXi-Host. Das ist wahrscheinlich darauf zurückzuführen, dass die Information sehr zeitkritisch ist und daher ein Umweg vermieden wird. Analysiert man das Vorgehen bei der ACI, stellt sich für mich die Frage, wieso bei der VTEP Aktualisierung eher passiv vorgegangen wird. Bei einer Änderung wird lediglich der alte Leaf-Switch und der Spine-Proxy informiert. Das hat zur Folge, dass alle Leaf-Switches mit einem alten Cache Eintrag das Paket zuerst an einen falschen Switch versenden. Erst bei einer Antwort wird der VTEP Forwarding Cache auf den neuen Leaf-Switch aktualisiert. Unter Umständen wäre es vielleicht besser, direkt alle Leaf-Switches über die Änderung zu informieren, damit es bei der Zustellung zu keiner Verzögerung kommt.

12 Schlussfolgerung

Software Defined Networking bleibt nach wie vor ein grosser Trend in der gegenwärtigen Zeit. Die Studienarbeit hat während der Einführung klar verdeutlicht, dass SDN ein sehr grosses Potenzial besitzt aber sehr unterschiedlich wahrgenommen wird. Je nach Hersteller wird eine andere Ideologie verfolgt und dementsprechend vermarktet. Mit den jeweiligen Lösungen von Cisco und VMware wurde versucht, eine erste Analyse zwischen zwei unterschiedlichen Ansätze aufzuzeigen. Beide verfolgen dabei das Ziel, das Rechenzentrum zu automatisieren, Cloud Dienste auf Basis eines Self-Service anzubieten und darüber hinaus enorme Kosten einzusparen. Dazu braucht es aber immer die Hilfe von Zusatzprodukten oder eine Eigenentwicklung. Standardmässig wird lediglich eine API Schnittstelle angeboten.

Das erste Produkt das getestet wurde, VMware NSX, stellt mit seiner Softwarelösung ein Overlay Netzwerk zur Verfügung, dass in vielen Punkten überzeugen kann. Setzt das Unternehmen bereits mehrheitlich virtuelle Server ein und besitzt weitere VMware Produkte, ist NSX sicherlich eine mögliche Option. Leider wird aber das zugrundeliegende Netzwerk grossenteils vernachlässigt. Zum jetzigen Zeitpunkt ist es nicht möglich, das Underlay Netzwerk mit zu verwalten. So entsteht weiterhin ein gewisser Mehraufwand und erfüllt nicht alle Kundenwünsche.

Mit der Cisco ACI wird eine vollständige SDN Lösung angeboten. Das beinhaltet auch, dass zwangsläufig auf das Produktsortiment von Cisco zurückgegriffen werden muss. Die ACI Fabric setzt auf eine Spine/Leaf Topologie, basierend auf der Nexus 9000 Serie, was enorme Performance mit sich bringt. Anders als mit den NSX-Kontrollern, möchte mit dem APIC Kontroller nicht ein Teil der Control Plane ausgelagert werden, sondern dient als Automatisierungs- und Managementpunkt. Das bedeutet auch, dass die gesamte Logik auf den Leaf- und Spine-Knoten bestehen bleibt.

Vergleicht man die Installation und Konfiguration beider Produkte in der Praxis, schneidet Cisco massiv schlechter ab. Das Einrichten einer Testumgebung konnte durch mangelnde Dokumentationen nicht erfolgreich abgeschlossen werden. Beschriebene Konfigurationsvorgänge wurden später von Experten als Designfehler interpretiert. Ebenfalls nicht überzeugend ist die angebotene Weboberfläche zur Verwaltung der Fabric. Die Konfigurationen sind im Standard Web-GUI einfach zu verschachtelt und zu intransparent.

Möchte man an dieser Stelle eine Produkteempfehlung erhalten, so kann diese auch nach einer ersten Analyse nicht abgegeben werden. Dazu sind die Produkte im Kern zu verschieden. Wie im Abschnitt „Evaluationskatalog“ angemerkt, müssen zuerst die eigenen Bedürfnisse erfasst werden. Erst dann lässt sich in einem zweiten Schritt eine mögliche SDN Lösung evaluieren.

13 Glossar

ACI – *Application Centric Infrastructure*
ANP – *Application Network Profiles*
API – *Application Programming Interface*
APIC – *Application Policy Infrastructure Controller*
AVS – *Application Virtual Switch*
BUM – *Broadcast, Unknown Unicast and Multicast*
BYOD – *Bring Your Own Device*
CLI – *Command Line Interface*
CoS – *Class of Service*
DFW – *NSX Distributed Firewall*
EPG – *Endpoint Group*
FIB – *Forwarding Information Base*
HSRP – *Hot Standby Router Protocol*
IFM – *Intra-Fabric Messaging*
LISP – *Locator/ID Separation Protocol*
LLDP – *Link Layer Discovery Protocol*
NAC – *Network Access Control*
NFS – *Network File System*
onePK – *Open Network Environment Platform Kit*
ONF – *Open Networking Fundation*
QoS – *Quality of Service*
RIB – *Routing Information Base*
SDN – *Software Defined Networking*
TEP – *Tunnel End Point*
ToR – *Top of Rack*
UTEP – *Unicast Tunnel End Point*
VDS – *VMware vSphere Distributed Switch*
VIB – *vSphere Installation Bundle*
VIO – *VMware Integrated OpenStack*
VLAN – *Virtual Local Area Network*
VNI – *VXLAN Network Identifier*
vNIC – *Virtual Network Interface Card*
VNID – *VXLAN Network Identifier*
VRF – *Virtual Routing and Forwarding*
VRRP – *Virtual Router Redundancy Protocol*
VTEP – *VXLAN Tunnel End Point*
VXLAN – *Virtual Extensible LAN*
XNC – *Extensible Network Controller*

14 Literaturverzeichnis

Chuck Black Paul Goransson Software Defined Networks [Buch]. - [s.l.] : Elsevier Science, 2014.

Cisco ACI and VMware Integration [Online]. - 15. 11 2015. - http://www.cisco.com/c/en/us/solutions/collateral/data-center-virtualization/application-centric-infrastructure/white-paper-c11-731961.html.

Cisco ACI Fabric Forwarding In a Nutshell [Online]. - 03. 12 2015. - http://ethancbanks.com/2014/10/16/cisco-aci-fabric-forwarding-in-a-nutshell/.

Documentation VMware NSX 6.2 for vSphere [Online]. - 02. 11 2015. - http://pubs.vmware.com/NSX-62/index.jsp?lang=en.

Dr. Jim Metzler, Ashton Metzler The 2015 Guide to SDN and NFV [Online]. - 01. 10 2015. - http://www.nuagenetworks.net/wp-content/uploads/2015/02/2015Ebook-Nuage.pdf.

Firefly Educate Cisco Nexus 9000 Switches in ACI Mode Test Drive [Buch]. - [s.l.] : FIREFLY, 2014. - Bd. 1.1.

Guide Cisco Application Centric Infrastructure Design [Online]. - 02. 12 2015. - http://www.cisco.com/c/en/us/solutions/collateral/data-center-virtualization/application-centric-infrastructure/white-paper-c11-731960.html.

Installation VMware NSX [Online]. - 27. 10 2015. - http://www.vmwarearena.com/2015/01/vmware-nsx-installation-part-1-nsx-overview-installation-prerequistes.html.

Ken Gray Thomas D. Nadeau SDN: Software Defined Networks [Buch]. - [s.l.] : O'Reilly Media, Inc., 2013.

Maurizio Portolani Lucien Avramov The Policy Driven Data Center with ACI: Architecture, [Buch]. - [s.l.] : Cisco Press, 2014.

Pressemeldung [Online]. - 28. September 2015. - http://globalnewsroom.cisco.com/de/de/press-releases/cisco-prasentiert-mit-aci-komplett-anwendungszentr-nasdaq-csco-1066713.

Raffe Adam [Online]. - 28. 11 2015. - http://adamraffe.com/.

SDN: Hype oder Muss? [Online]. - 13. 12 2015. - http://www.computerwoche.de/a/sdn-hype-oder-muss,2554649.

VMware NSX 6.x Configuration Maximums [Online]. - 10. 12 2015. - http://www.vmwareguruz.com/cloud-e2e/vmware-nsx-6-x-configuration-maximums/.

15 Abbildungsverzeichnis

II. Anhänge

16 Projektmanagement

16.1 Management Summary

16.1.1 Ausgangslage

Grosse Unternehmen, wie Amazon und Google, investieren enorme Geldsummen in ihre Infrastruktur. Es braucht ständig neue Server, neuen Speicher und mehr Netzwerkleistung. Der ständige Ausbau der Kapazitäten skalierte aber ergonomisch nicht überall gleich gut. Im Bereich Netzwerk wurde lange Zeit der Trend in Richtung Virtualisierung und Automatisierung verschlafen. Erst in den letzten paar Jahren haben auch grosse Firmen wie Cisco, HP und VMware das Potenzial erkannt und setzen vermehrt auf Software Defined Networking (SDN). Ziel ist es, die einst unabhängigen Netzwerkkomponenten zentral zu Steuern. Das ergibt den Vorteil, dass man flexibler und dynamischer auf Anforderungen reagieren kann. Es ist auch die optimale Grundlage für Cloud Computing.

Die Studienarbeit soll aufzeigen, ob die grossen Marketingversprechen auch wirklich eingehalten werden. Dazu werden zwei Produkte implementiert, getestet und analysiert. Basierend auf vorab definierten Anforderungen werden neben funktionellen Vergleiche auch operative Aspekte untersucht.

16.1.2 Vorgehen, Technologien

Unter dem Begriff SDN wird sehr vieles zusammengefasst. Im ersten Teil möchte ich mich mit der Frage auseinandersetzen, um was es sich bei SDN genau handelt und welche Use Cases davon abgedeckt werden. In einem zweiten Schritt werden detaillierte Anforderungen ausgearbeitet, um anschliessend die Produkte hinsichtlich Funktionalität und Bedienbarkeit zu testen. Konkret wird eine kleine Private Cloud Umgebung mit einem einzelnen Mandat abgebildet. Dazu wird eine Variante mit Cisco ACI und eine Variante mit VMware NSX realisiert.

16.1.3 Ergebnisse

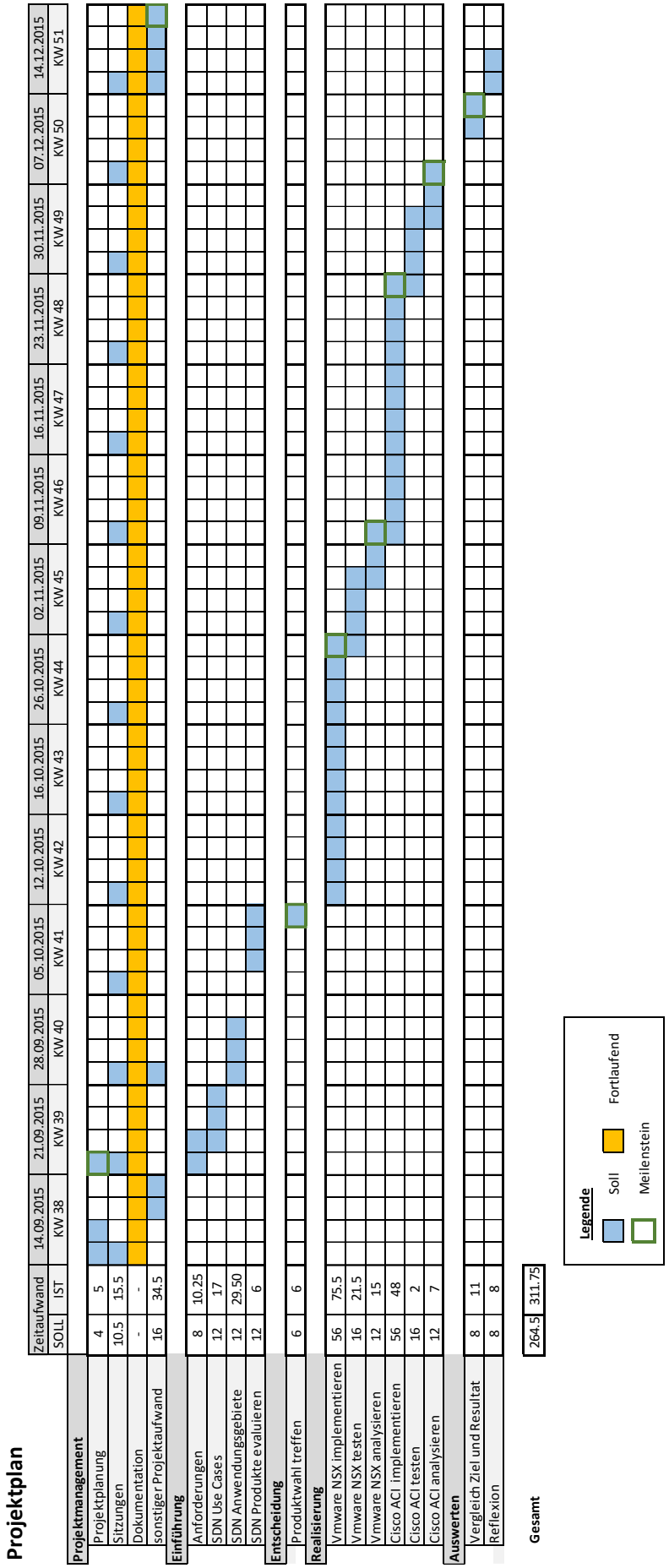
Software Defined Networking bleibt nach wie vor ein grosser Trend in der gegenwärtigen Zeit. Die Studienarbeit hat während der Einführung klar verdeutlicht, dass SDN ein sehr grosses Potenzial besitzt aber sehr unterschiedlich wahrgenommen wird. Je nach Hersteller wird eine andere Ideologie verfolgt und dementsprechend vermarktet.

Mit VMware NSX wird einem eine komplett softwarebasierte Lösung angeboten. Das zugrundeliegende physikalische Netzwerk ist dabei nicht Bestandteil von SDN. Das hat zur Folge, dass einem trotz ausgereiften Overlay Netzwerk, was viele nützliche Funktionen mit sich bringt, nicht alle Wünsche erfüllt werden. Der Mehraufwand für die Verwaltung aller Komponenten bleibt weiterhin bestehen.

Cisco ACI bietet hingegen eine vollständige SDN Lösung. Dies beinhaltet aber auch, dass für die Kernelemente auf das Produktesegment von Cisco zugrückgegriffen werden muss. Dafür bekommt man für teures Geld viel Leistung. Leider hat sich während der Testphase gezeigt, dass die Bedienung ziemlich verbesserungswürdig ist. Zurzeit sind die Prozesse noch zu benutzerunfreundlich und zu verschachtelt.

Es handelt sich bei beiden Produkte um sehr interessante und sicherlich brauchbare Lösungen, die ihre Vor- und Nachteile haben. Es gilt die persönlichen Anforderungen zu definieren und dementsprechend zu evaluieren. Eine direkte Kaufempfehlung kann aufgrund ihrer Komplexität und Verschiedenheit nicht gemacht werden.

16.2 Projektplan



16.3 Projektorganisation

Das Projekt wird als Einzelarbeit von Pascal Meier durchgeführt. Die Betreuung wird von Prof. Beat Stettler übernommen, dabei steht sein Assistent Urs Baumann ebenfalls zur Verfügung.

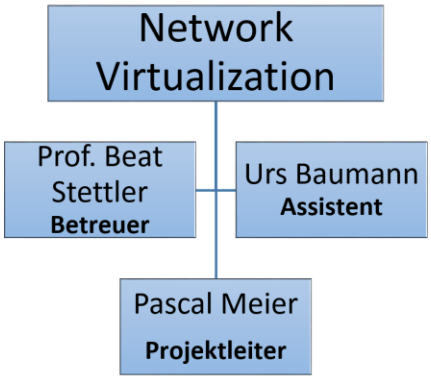


Abbildung 56: Organigramm

16.4 Zeitaufwand

Mit dem Startstuss am 14.09.2015 stehen dem Kandidaten 14 Wochen zur Verfügung. Dabei soll der Aufwand von mind. 240 Stunden nicht unterschritten werden. Die Arbeit findet parallel zum Unterricht der HSR statt und kann frei eingeteilt werden.

16.5 Besprechungen

Es wird eine wöchentliche Sitzung mit dem Betreuer angestrebt, um den aktuellen Stand zu besprechen. Dabei kann der genaue Wochentag variieren. Alle wichtigen Ereignisse und Entscheidungen werden in einem Sitzungsprotokoll festgehalten.

16.6 Infrastruktur

Dem Kandidaten wird ein Arbeitsplatz mit Computer zur Verfügung gestellt. Zusätzlich erhält der Kandidat Zugang zu einem Testlabor um die einzelnen Produkte zu testen.

Hardware:

- 1x IBM System x3650 (2x 2.66GHz Intel Xeon CPU, 32GB RAM)
- 1x IBM System x3550 (1x 2.66GHz Intel Xeon CPU, 16GB RAM)
- 2x IBM System x3550 (2x 2.00GHz Intel Xeon CPU, 5GB RAM)
- 1x Cisco Catalyst 4948
- 2x N9K-C9396PX
- 1x N9K-C9336PQ
- 1x APIC Controller

Software:

- VMware ESXi 6.0
- VMware vCenter 6.0
- VMware NSX-Manager 6.2.0
- Cisco APIC Controller 1.0(3n)
- Leaf-Switches n9000-11.1(4e)
- Spine-Switch n9000-11.1(4e)

16.7 Risikomanagement

ID	Risiko	Auswirkung	Massnahme	Eintrittswahr-scheinlichkeit
R01	Ausfall durch Krankheit	Verzögerung des Projektes	-	Niedrig
R02	Projektumfang zu gross	Geplante Arbeitspakete können nicht vollständig umgesetzt werden	Rechtzeitige Priorisierung der Arbeitspakete	Mittel
R03	Datenverlust	Daten müssen wieder hergestellt/neuerstellt werden	Versionierung und regelmässige Backups	Niedrig
R04	Hardware/Software kann nicht wie geplant beschaffen werden	Neues Produkt muss evaluiert werden	Abklärungen und Zugeständnisse rechtzeitig einholen	Niedrig - Mittel

Tabelle 6: Risiko Management

16.8 Meilensteine

16.8.1 MS1 – Projektplan

Review des Projektplans

16.8.2 MS2 – Produktwahl für die Realisierung getroffen

Produkt wurde evaluiert und steht zur Projektarbeit zur Verfügung

16.8.3 MS3 – Produkt 1 implementiert

Produkt 1 wurde implementiert und ist funktionsfähig

16.8.4 MS4 – Produkt 1 analysiert

Produkt 1 wurde hinsichtlich operativer Aspekte untersucht.

16.8.5 MS5 – Produkt 2 implementiert

Produkt 1 wurde implementiert und ist funktionsfähig

16.8.6 MS6 – Produkt 2 analysiert

Produkt 2 wurde hinsichtlich operativer Aspekte untersucht.

16.8.7 MS7 – Auswertung

Es wird ein Fazit über alle Erkenntnisse gezogen.

16.8.8 MS8 – Projektabgabe

Abgabe des Gesamtprojekts auf CD.

16.9 Zeitplan

Es wurde 265 Stunden budgetiert und letztendlich 312 Stunden gearbeitet. Die verlangten 240 Stunden wurden deutlich erreicht und mit 72 Stunden überschritten. Der Mehraufwand lässt sich einerseits durch unvorhergesehene Ereignisse und anderseits durch eine zu offene Projektplanung begründen.

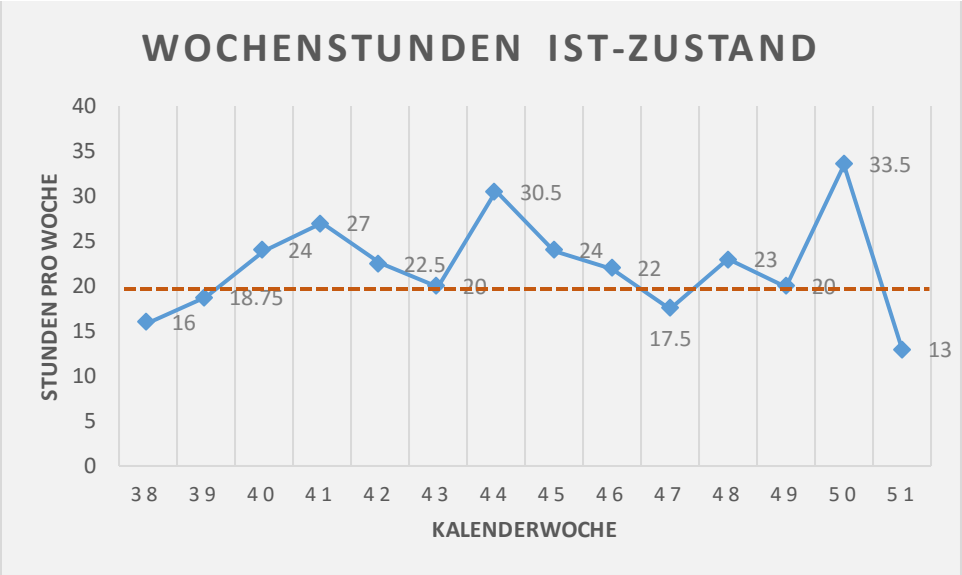


Abbildung 57: Wochenstunden

Wie unschwer zu erkennen ist, wurde die meiste Zeit in die Realisierung investiert. Dies beinhaltet ebenfalls die Einarbeitung und das Testen der Umgebung.

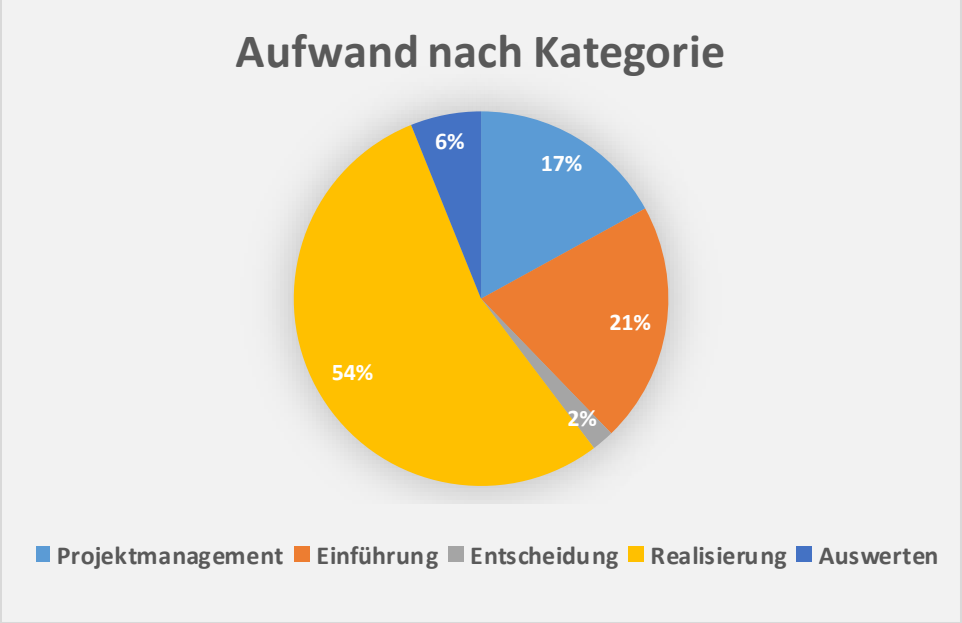


Abbildung 58: Aufwand nach Kategorie

16.10 Persönlicher Bericht

Ich blicke auf 14 spannende und intensive Wochen zurück in denen es Höhen und Tiefen gab. Direkt zu Beginn wurde ich mit der Tatsache konfrontiert, die Arbeit im Alleingang durchzuführen. Mein Arbeitspartner wurde im letzten Moment nicht zur Semesterarbeit zugelassen und musste das geplante 2er Team verlassen. Angespornt von der interessanten Aufgabenstellung habe ich mich, nach Rücksprache mit meinem Dozenten, dazu entschlossen, die Arbeit alleine zu schreiben.

Während der Planungsphase wurde die Aufgabenstellung leicht angepasst und das Ziel auf zwei zu testende Produkte gelegt. Wobei in den ersten paar Wochen noch unklar war, ob die Cisco Hardware rechtzeitig beschafft werden konnte. Nichts desto trotz habe ich mich im ersten Teil recht intensiv mit dem Thema SDN auseinander gesetzt. Dabei musste ich immer wieder feststellen, dass viele Informationen rein marketingtechnisch formuliert wurden. Handfeste Performancemessungen oder detaillierte Use Cases wurden nicht gefunden. Das ist unter anderem einer der Gründe, wieso viel mehr Zeit als geplant in die Einführung investiert wurde.

Das mit viel Spannung erwartete VMware Produkt konnte im Labor erfolgreich aufgebaut werden. Die Informationsbeschaffung gelang dabei erstaunlich gut. VMware legt sehr viel Wert, eine umfangreiche Unterstützung anzubieten. Leider war es ein wenig Schade, dass mir im Labor nur bedingt gute Hardware zur Verfügung stand. Einzelne Arbeitsschritte zogen sich unnötig in die Länge und stellten meine Geduld arg auf die Probe.

Was leider überhaupt nicht erfolgreich verlief, war die Arbeit mit der ACI. Nebst lückenhafter Dokumentation, stand mir ebenfalls keine Expertenhilfe zur Verfügung. Über 3 Wochen hinweg war es mir nicht möglich, eine funktionierte Testumgebung aufzubauen. Dabei scheiterte es bereits daran, das vCenter in die ACI Fabric zu integrieren. Das hatte zur Folge, dass keinerlei Tests gemacht werden konnten und dass mir daher die nötige Erfahrung fehlte, eine spannende und aussagekräftige Meinung zu bilden, die es mir ermöglichte, ein gutes Fazit zu ziehen.

Rückblickend denke ich, dass sich ein Gedankenaustausch mit einem Teamkollegen sehr gelohnt hätte. Trotzdem bedauere ich meine Endscheidung nicht, die Arbeit alleine gemacht zu haben. Ich bin stolz auf meine geleistete Arbeit und würde diesen Weg wieder gehen.

Abschliessend möchte ich mich bei Prof. Dr. Beat Stettler für seine gute und kompetente Betreuung bedanken. Ein besonderer Dank geht ebenfalls an Urs Baumgartner, der mich während meiner Arbeit mit Rat und Tat zur Seite stand.