HSR
HOCHSCHULE FÜR TECHNIK
RAPPERSWIL

FHO Fachhochschule Ostschweiz

ins institute for networked solutions

Software Defined Network

Studienarbeit

Abteilung Informatik
Hochschule für Technik Rapperswil

Frühjahrssemester 2014

Autor:	Remo Grüebler
Betreuer:	Professor Beat Stettler
Projektpartner:	Institut für Vernetzte Systeme
Experte:	
Gegenleser:	

Inhalt

1 Aufgabenstellung

Die letzten 30 Jahre Networking waren vom Prinzip der "verteilten Intelligenz" geprägt. Switches und Router entscheiden grundsätzlich selbständig, wohin sie Pakete switchen/routen und welche Informationen sie mit Nachbargeräten austauschen. Dieser Ansatz hat sich als sehr stabil, aber auch zunehmend als träge erwiesen. Träge insofern, als dass bei Netzänderungen alle Geräte einzeln umkonfiguriert werden müssen, was sehr zeitaufwändig ist.

Neue Dienste wie CloudComputing oder die zunehmende Mobilität stellen viel dynamischere Anforderungen an die Kommunikationsinfrastruktur. Kundennetze müssen innert Sekunden provisioniert mit den zugehörigen Netzwerkfunktionen wie Switching, Routing, Firewalling oder Load-Balancing ausgestattet werden können. Virtuelle Maschinen sollen automatisch vom einen Datacenter in ein anderes verschoben werden können. Auch die dazu nötigen Konfigurationsänderungen müssen automatisch erfolgen. Gerade die grossen Cloud-Anbieter wie Google oder Amazon pushen deshalb ein neuer Ansatz, genannt Software Defined Networking (SDN).

MIt SDN wird ein Teil der (bisher dezentral verteilten) Intelligenz neu auf sogenannte Controller zentralisiert. Zentrale Controller steuern ihrerseits die dezentral verteilten Netzwerkfunktionen über neuartige APIs, die einen viel grösseren Funktionsumfang aufweisen als heutige Konfigurationsprotokolle (wie CLI). Zwischen Controller und Netzwerkfunktionen kommen ebenfalls neue Protokolle wie das Openflow Protokoll zum Einsatz. Dieser Ansatz ist auch insofern revolutionär, weil er die Vorherrschaft der bisherigen Hersteller in Frage stellt. Wenn jeder eine Software zur Steuerung von günstigen Endgeräten entwickeln kann, dann ist auch nicht von vornherein klar, dass sich wieder die gleichen Player als Marktführer werden durchsetzen können.

Diese Studienarbeit ist in einen konzeptionellen und einen praktischen Teil gegliedert. Im konzeptionellen Teil soll der neuartige Ansatz "SDN" analysiert und kritisch gewürdigt werden. In einem zweiten Schritt sollen die Aktivitäten der wichtigsten Player im Markt, das heisst der Forschungsgemeinde, der Standardisierungsbehörden und der Hersteller kurz zusammengefasst und daraus Schlüsse bezüglich der Erfolgschancen der verschiedenen Umsetzungsansätze gezogen werden. Im praktischen Teil soll eine Testumgebung aufgebaut und erste Implementationen von Controllern, Protokollen und APIs auf ihren Reifegrad geprüft werden, indem mit diesen eine konkrete Anwendung (z.B. verschieben von VMs) umgesetzt wird.

Voraussetzungen: - Gutes Netzwerk Kenntnisse (z.B. CCNA)

2 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.
- das ich keine durch Copyright geschützten Materialien (z.B. Bilder) in dieser Arbeit in unerlaubter Weise genutzt habe.

Ort, Datum: Rapperswil 30.5.2014

Name, Unterschrift:

Remo Grüebler

3 Abstract

Die heutigen Computernetzwerke befinden sich im grössten Umbruch seit Beginn der IT. Netzwerke sollen nicht mehr auf Geräte Ebene konfiguriert werden, sondern als Ganzes. Dadurch ist es möglich, Netzwerke zu bauen, die flexibler, sicherer, robuster und sicherer sind als heutige Netze.

Um dies zu erreichen, wird eine weitere Abstraktionsstufe für ein Netzwerk eingefügt. Diese Stufe wird Orchestrar genannt. Dies erlaubt es einem Netzwerkadministrator, einen Service im Netzwerk zur Verfügung zu stellen, ohne sich mit jedem einzelnen Gerät zu beschäftigen.

Diese Studienarbeit zeigt, dass Software Defined Networking grosses Potential besitzt. Grad im Bereich Sicherheit kann man sehr viel erreichen. Es ist möglich, das ganze Netzwerk, mittels Flows, zu schützen und nicht nur an zentralen Punkten eine Firewall einzusetzen. Weiter kann das Netz auch dynamisch auf einen Angriff reagieren. So kann zum Beispiel ein DoS Angriff von einem Link ferngehalten oder ein Virus isoliert werden. Auch die Flexibilität von einem Netz nimmt zu mit SDN.

So kann dynamisch auf einen überlasteten Link reagiert werden und bestimmte Daten auf einen anderen Link umgeleitet werden.

Die Studienarbeit zeigt jedoch auch, dass es noch keine Lösungen auf dem Markt gibt, die produktiv einsetzbar ist.

4 Management Summary

Ausgangslage

In einem heutigen Netzwerk muss ein Netzwerkadministrator jedes einzelne Gerät von Hand konfigurieren. Dies ist zeitaufwändig und fehleranfällig. Deshalb wird versucht, ein Netzwerkmanagement System zu schaffen, das ein Netzwerk als Ganzes verwalten und konfigurieren kann. Der neuste Trend ist das Software Defined Network. Hier sollen alle Konfigurationen auf einem zentralen System erstellt werden können. Diese Konfigurationen werden dann im Netzwerk verteilt. Das Netzwerkmanagement System meldet, wenn ein Fehler passiert ist und macht die Konfiguration rückgängig. So gelangen keine fehlerhaften Einstellungen ins Netzwerk, was einen positiven Einfluss auf die Stabilität hat.

Vorgehen

In dieser Studienarbeit wird untersucht, wie weit Software Defined Network fortgeschritten ist. Dabei werden unterschiedliche Architekturen verglichen und bewertet. Anhand von verschiedenen Use Cases werden dann heutige Netzwerke mit einem Software Defined Netzwerk verglichen und bewertet.

In dem praktischen Teil der Arbeit wird versucht, die Lösungen der einzelnen Hersteller, untereinander zu vergleichen und zu bewerten.

Ergebnisse

Im ersten Teil der Studienarbeit wird Software Defined Network mit heutigen Technologien verglichen und die Vor- und Nachteile besprochen. Dabei stellt sich raus, dass SDN ein grosses Potential besitzt.

Im zweiten Teil der Arbeit werden die Produkte von vier Herstellern miteinander verglichen. Es stellt sich raus, dass noch keine der Lösungen im produktiven Einsatz verwendbar ist. Die Produkte sind noch nicht ausgereift.

Ausblick

Im praktischen Teil der Studienarbeit konnten nur wenige Tests durchgeführt werden. Sobald die Hersteller eine komplette Lösung für Software Defined Network anbieten, können weitere Tests und Vergleiche durchgeführt werden.

5 Technischer Bericht der Arbeit

5.1 Einleitung und Übersicht

Der Startschuss für das World Wide Web wurde 1989 am Forschungszentrum CERN in Genf gegeben. Damals wurde, von Berners-Lee, das Hypertext Transfer Protokoll (HTTP) entwickelt. Dieses Protokoll wird auch heute noch für die Kommunikation, zwischen dem Webbrowser und einer Website verwendet. Weiter wurde die Seitenbeschreibungssprache Hypertext Markup Language (HTML) entwickelt. Diese ist für die Darstellung von Inhalten auf einer Website zuständig. Beide Protokolle wurden ohne Lizenzgebühren zur freien Verwendung angeboten.

Die Idee bestand darin, wissenschaftliche Daten leichter zugänglich zu machen. Deshalb war das Internet zuerst nur von Universitäten und Forschungsanstalten aus erreichbar. Die Wissenschaftler konnten somit Daten mit ihren Kollegen in der ganzen Welt austauschen. Diese Kommunikation fand vorläufig nur per Email statt.

Später wurden auch kommerzielle Firmen ans Internet angeschlossen. Einige von ihnen wurden zu Service Providern. So konnte sich das Internet langsam in die Privathaushalte verbreiten. Die Anbindung der Haushalte wurde über die herkömmliche Telefonleitung realisiert. Dies war langsam und teuer. Erst mit der Einführung von Digital Subscriber Line (DSL) stand endlich eine kostengünstige und schnelle Internetverbindung zur Verfügung. Nun konnte sich das Internet schnell verbreiten. Einen grossen Einfluss auf die schnelle Verbreitung hatte auch die Tatsache, dass HTTP und HTMP frei zur Verfügung gestellt wurden. So konnten Webbrowser entwickelt werden, die mit jeder Website eine Verbindung aufbauen kann. Bis heute nehmen die Datenraten ins Internet laufend zu. Stehen doch mit Fiber To The Home (FTTH) mehrere hundert Megabit pro Sekunde (Mbit/s) zur Verfügung.

Für die Realisierung des Internets mussten Netzwerke aufgebaut werden. Diese bestehen aus Netzwerkkomponenten, wie Router und Switch. Diese Geräte können untereinander kommunizieren, um Informationen auszutauschen. Bis heute trifft jedes Gerät die Entscheidung, wohin die Daten übertragen werden müssen, für sich selbst. Das heisst es gibt keine zentrale Steuereinheit, die diese Entscheidungen trifft. Dies hat den Vorteil, dass ein Ausfall eines Gerätes nicht gleich das gesamte Netzwerk unbrauchbar macht.

In der heutigen Zeit sind die Anforderungen an die Informatiknetzwerke sehr stark gestiegen. Ein Netzwerk muss sehr schnell auf Veränderungen reagieren können. Da aber jedes Netzwerkgerät eigene Entscheidungen trifft, müssen diese Geräte auch einzeln konfiguriert werden. So ist es nicht mehr möglich in kurzer Zeit auf Veränderungen reagieren zu können. Deshalb muss ein neuer Weg gefunden werden, wie ein Informatiknetzwerk schneller um konfiguriert werden kann. Die Entwicklung geht heute in Richtung einer zentralen Verwaltung. So kann eine Änderung auf einem einzigen System vorgenommen werden und von da aus in das gesamte Netzwerk verteilt werden. Diese Idee wird von Software Defined Network (SDN) umgesetzt. An der Erforschung von SDN sind nicht nur Standardisierungs-Organisationen sondern auch einzelne Hersteller von

Netzwerkkomponenten beteiligt. Dies hat zur Folge, dass es einige unterschiedliche Ansätze, wie SDN funktionieren soll, gibt.

In dieser Studienarbeit geht es nun darum, diese verschiedenen Lösungsansätze zu vergleichen und zu bewerten. Dabei sollen unterschiedliche Use Cases definiert werden und verglichen werden, wie diese Heute gelöst werden und wie Software Defined Network diese löst. Dabei soll bewertet werden, ob SDN diese Probleme einfacher und effizienter lösen kann oder ob sie heute schon ausreichend gut gelöst werden. Braucht es SDN wirklich oder ist es nur ein Marketinghype?

5.2 Architekturüberblick

Die grösste Veränderung von SDN zu den heutigen Netzwerken, ist der Wechsel von einem dezentralen zu einem zentralen Netz. Bis jetzt hat ein Router seine direkt angeschlossenen Netze mit Hilfe von einem Routingprotokoll an seine Nachbarrouter weitergegeben. Diese konnten danach Entscheidungen treffen, welches der „beste“ Weg zum Ziel ist. Der Router konnte nur anhand von der Bandbreite, Delay oder der Anzahl Hops den „besten“ Weg finden. Der Administrator hatte jedoch keine einfache Möglichkeit, Einfluss auf diese Entscheidung zu nehmen. Mit SDN soll dies jedoch möglich sein. Auch ein Switch lernt bis heute die angeschlossenen Geräte kennen, wenn diese Daten senden. Dann speichert der Switch die Source MAC Adresse in seiner MAC Address Tabel. Die Kommunikationsdaten zwischen den Routern und die Daten der Clients werden über das gleiche Interface versendet.

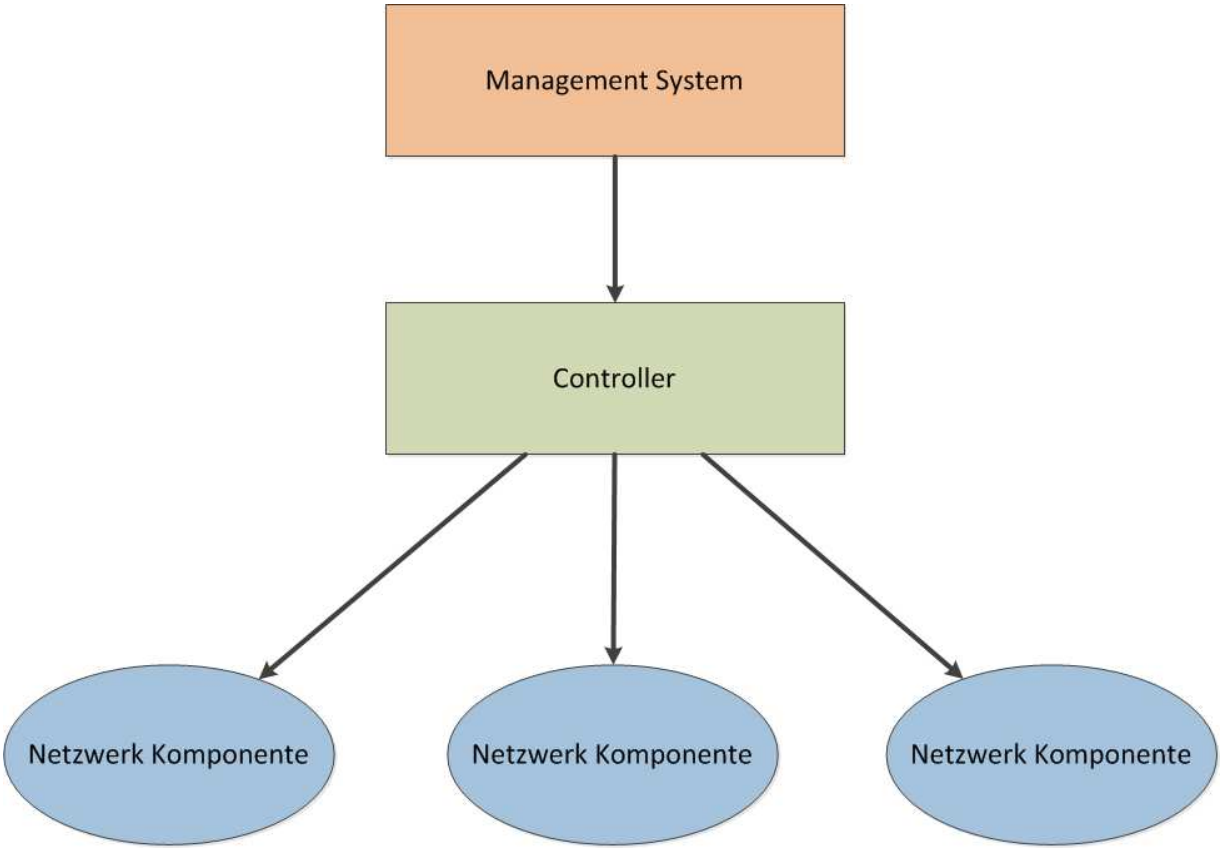
Beim Software Defined Network wir der Control Plane und der Data Plane voneinander getrennt. Als Control Plane werden alle Informationen, die benötigt werden, um den Betrieb eines Netzwerks zu ermöglichen, bezeichnet. Dazu gehören die obenerwähnten Routingprotokolle aber auch Protokolle wie das Spanning Tree Protokoll, welches ein Loop im Netz verhindert. Im Control Plane werden also keine Nutzerdaten übertragen sondern nur Steuerungsinformationen ausgetauscht.

Als Data Plane werden nun alle produktiven Daten zusammengefasst. Das heisst, alle Nutzerdaten, wie Emails, Abfragen an eine Webseite oder ein Download von einer Datei.

Die Daten des Control Planes werden über ein anderes Interface gesendet, als die Daten des Data Plane. Ein Interface ist jedoch nicht unbedingt physikalisch vorhanden sondern es kann auch nur ein logisches Interface sein. Es muss also nicht ein separates Kabel zwischen jedem Switch und dem Controller gezogen werden. Die Daten werden zwar über das gleiche Kabel gesendet, sie sind jedoch logisch voneinander getrennt. Die Trennung wird erreicht, weil der Controller die Entscheidungen, welche Daten wohin geschickt werden übernimmt. Dies wird mittels sogenannten Flows erledigt. Ein Flow ist ein Datenstrom, der von einem Punkt im Netzwerk zu einem anderen transportiert wird. Bevor die Daten übertragen werden können, muss der Controller einen Flow aufbauen. Das heisst, es wird ein Flow durch das Netzwerk durchsignalisiert. Wenn der Flow aufgebaut ist, können die Userdaten übertragen werden. Jeder Flow besitzt eine Priorität. Mit dieser kann die Reihenfolge der Flows, in der Flowtabel, bestimmt werden. Zuerst sollen die spezifischen Flows auftauchen und erst weiter unten in der Flowtabel die generellen. So kann sichergestellt werden, dass ein spezifischer Flow angewendet wird.

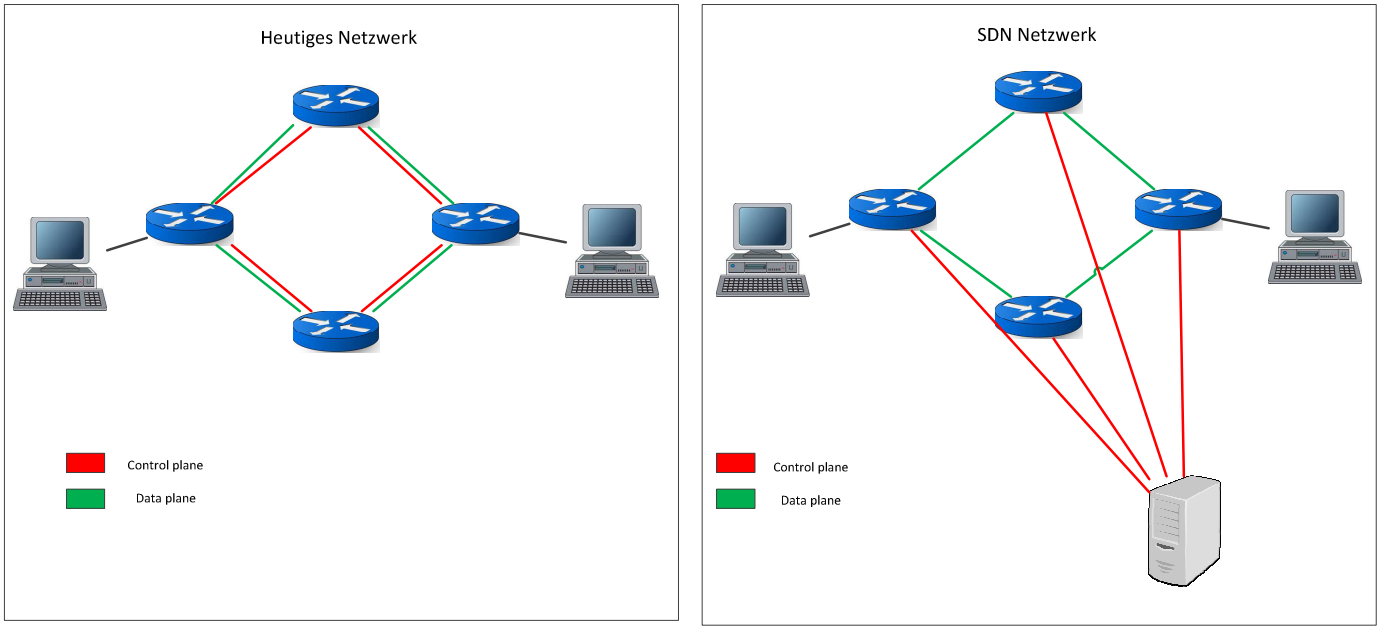
Da bei SDN der Controller eine sehr wichtige Komponente ist, muss er vor einem Ausfall geschützt werden. Dazu werden mindestens zwei Controller in einem sogenannten Cluster eingesetzt. Ein Controller wird als Master eingesetzt, der andere als Secondary. Fällt der Master aus, übernimmt der Secondary dessen Aufgaben. So entsteht kein Single Point of Failure. Weiter bleibt jedoch das Problem der langen Übertragungswege bestehen. Um Kosten zu sparen wäre eine Möglichkeit nur im Hauptquartier von einer Firma einen Controller einzusetzen und die Niederlassungen auch über diesen zu verwalten. Dabei können grosse Verzögerungen, zwischen Controller und den Switch entstehen. Eine Konfigurationsänderung braucht so sehr lange, bis sie an der entsprechenden Position Wirkung zeigt. Deshalb skaliert ein zentraler Controller nicht gut. Weiter kann die Hardware von einem Controller sehr teuer werden, da in einem grossen Netz sehr viele Anfragen auf ihn zukommen.

Im untenstehenden Bild ist eine Hierarchie von einem Software Defined Network abgebildet. Zu unterst befinden sich die Netzwerkkomponenten. Danach wird ein Controller eingesetzt, der mit den Netzwerkkomponenten kommunizieren kann. Als oberste Schicht wird ein Management System, oder auch Orchestrar genannt, eingesetzt.



Nach diesem kurzen Überblick über die Struktur von SDN wollen wir uns anschauen, wie die Kommunikation zwischen den Geräten aussieht.

Durch das Trennen von Control Plane und Data Plane kann das Netzwerkmanagement, bei SDN, an einem zentralen Ort zusammengefasst werden. Dazu besteht nun die Möglichkeit, die Konfiguration auf einem einzigen System vorzunehmen und danach im Netzwerk zu verteilen. Dadurch können Änderungen viel schneller vorgenommen werden. Dieses zentrale Management wird Orchestrierung genannt. In der nachstehenden Skizze wird den Unterschied zwischen dem verteilten Ansatz und dem zentralen Ansatz verdeutlicht.



Auf dem linken Bild führt der Control Plane und der Data Plane über dasselbe Kabel und die Intelligenz ist auf alle Geräte verteilt. Hier kann nur auf Events, die das Routingprotokoll betreffen, schnell reagiert werden. Zum Beispiel wenn ein Gerät ausfällt wird dies von den verbleibenden drei Geräten erkannt. Danach werden die Verbindungen neu berechnet. Dies findet in relativ kurzer Zeit statt. Soll jedoch eine Konfiguration im Netzwerk verändert werden, dauert dies sehr lange, da sich der Administrator auf jedes einzelne Gerät einloggen muss um die Änderung vorzunehmen.

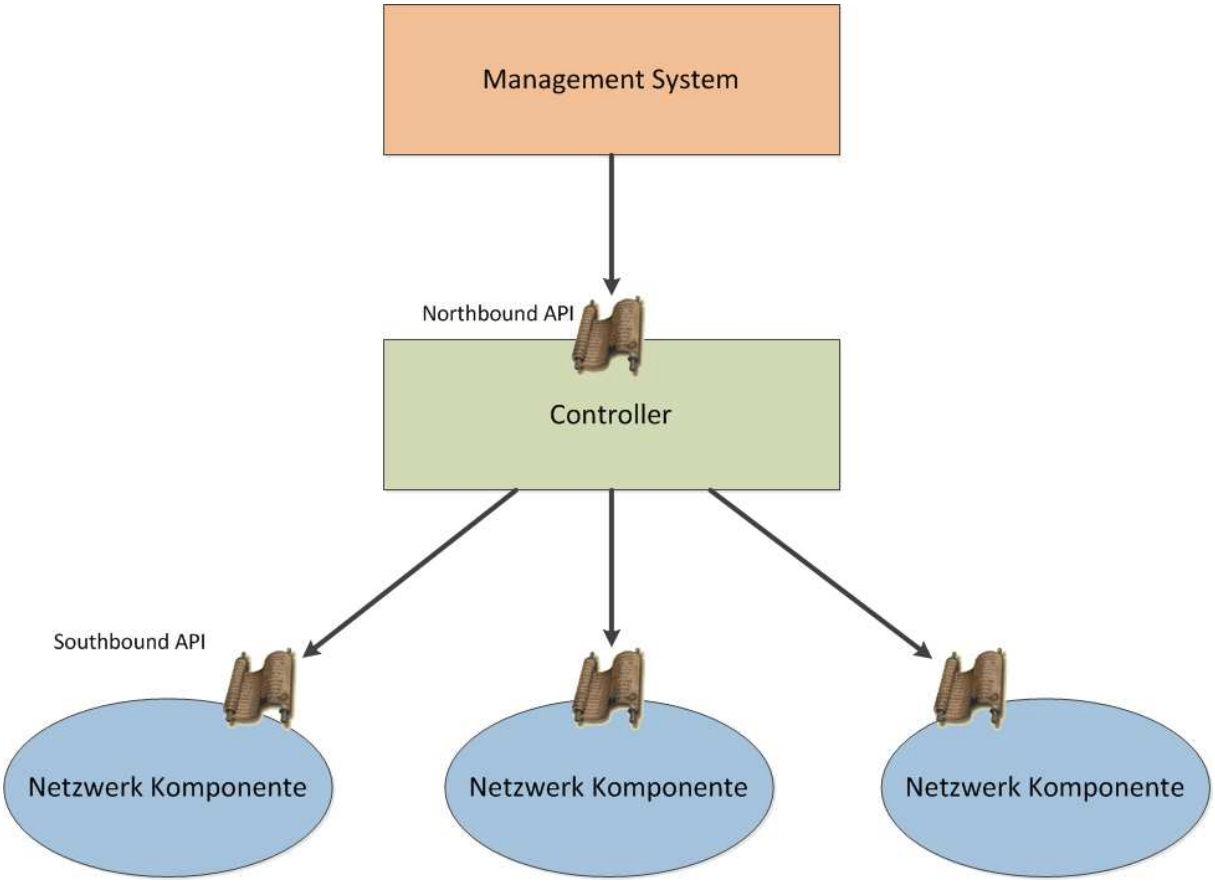
Auf dem rechten Bild führt der Control Plane zu einem zentralen Controller und nur der Data Plane wird direkt zwischen den Neztwerkgeräten transportiert. Hier kann das Netzwerk natürlich auch sehr schnell auf einen Ausfall von einem Gerät reagieren. Zusätzlich können jedoch auch Änderungen der Konfiguration schnell im Netzwerk verteilt werden. Ein Administrator loggt sich auf den Controller in und nimmt eine Konfiguration vor. Danach wird diese Änderung auf alle Geräte gleichzeitig verteilt.

Bis jetzt haben wir immer davon gesprochen, dass der Controller mit dem Netzwerkgerät kommuniziert. Heute wird im Zusammenhang mit SDN meistens von einem Switch gesprochen. Damit nun ein SDN Switch und ein Controller untereinander kommunizieren können, muss der Switch ein Application Programmable Interface (API) besitzen. Dieses API definiert die „Sprache“, die der Switch verstehen soll. Das heisst, welche Befehle ein Controller an den Switch senden kann, damit dieser sie versteht. Als Beispiel wird hier ein Openflow API verwendet. Dieses API ist ein sogenanntes Southbound API. Ein solches API wird für die Kommunikation mit einem Netzwerkgerät, das in der SDN Hierarchie, weiter unten liegt, verwendet. OpenFlow ist heute das meistverwendete Protokoll auf einem Southbound API. Fälschlicherweise wird jedoch häufig SDN mit OpenFlow gleichgesetzt. Dies ist jedoch eine falsche Annahme. SDN ist ein Konzept, wie moderne Netzwerke aufgebaut werden können. OpenFlow ist nur ein kleiner Teil von diesem Konzept.

Das API von OpenFlow ist ein offenes API. Das heisst, jeder Hersteller kann es mit seinem Controller ansteuern. Dies hat den Vorteil, dass eine beliebige Netzwerkhardware eingesetzt werden kann. Weiter kann auch der Controller ausgetauscht werden, ohne dass die Netzwerkhardware auch ersetzt werden muss. Die einzelnen Hersteller von SDN fähiger Hardware, stellen auch eigene API zur Verfügung. So kann die Netzwerkhardware nur von einem Controller des Herstellers gesteuert werden. Nun soll noch ein Blick auf das Northbound API geworfen werden.

Über ein Northbound API kommuniziert ein Gerät mit einer Komponente, die in der Hierarchie eine Schicht höher oben ist. Hier wird dieses API für die Kommunikation mit dem Management System, auch Orchestrar genannt, verwendet. So kann eine Policy auf mehrere Controller verteilt werden, ohne jeden Controller einzeln zu konfigurieren. Weiter können auch Informationen beim Controller angefragt werden, wie zum Beispiel die Flowtable. Das Zusammenspiel von einem Controller mit dem Orchestrar wird im Kapitel „Anforderungen an SDN“ genauer behandelt.

Im nachstehenden Bild wird der Unterschied zwischen Northbound und Southbound, aus Sicht des Controllers, erklärt. Für die Kommunikation mit dem Orchestrar wird das Northbound API verwendet. Meistens wird heute ein REST API verwendet. Dabei kann per HTTP eine Aktion auf dem Controller ausgeführt werden. Mit dem GET Kommando können Informationen ausgelesen werden und mit POST werden Änderungen auf den Controller gespielt. Für die Kommunikation mit den Netzwerk Switch wird das Southbound API verwendet. Dabei kann es sich um OpenFlow, XMPP, OnePK, SNMP, SSH, OpFlex oder HTTP handeln.



Northbound- und Southbound API aus Sicht des Controllers

5.3 Architekturvarianten

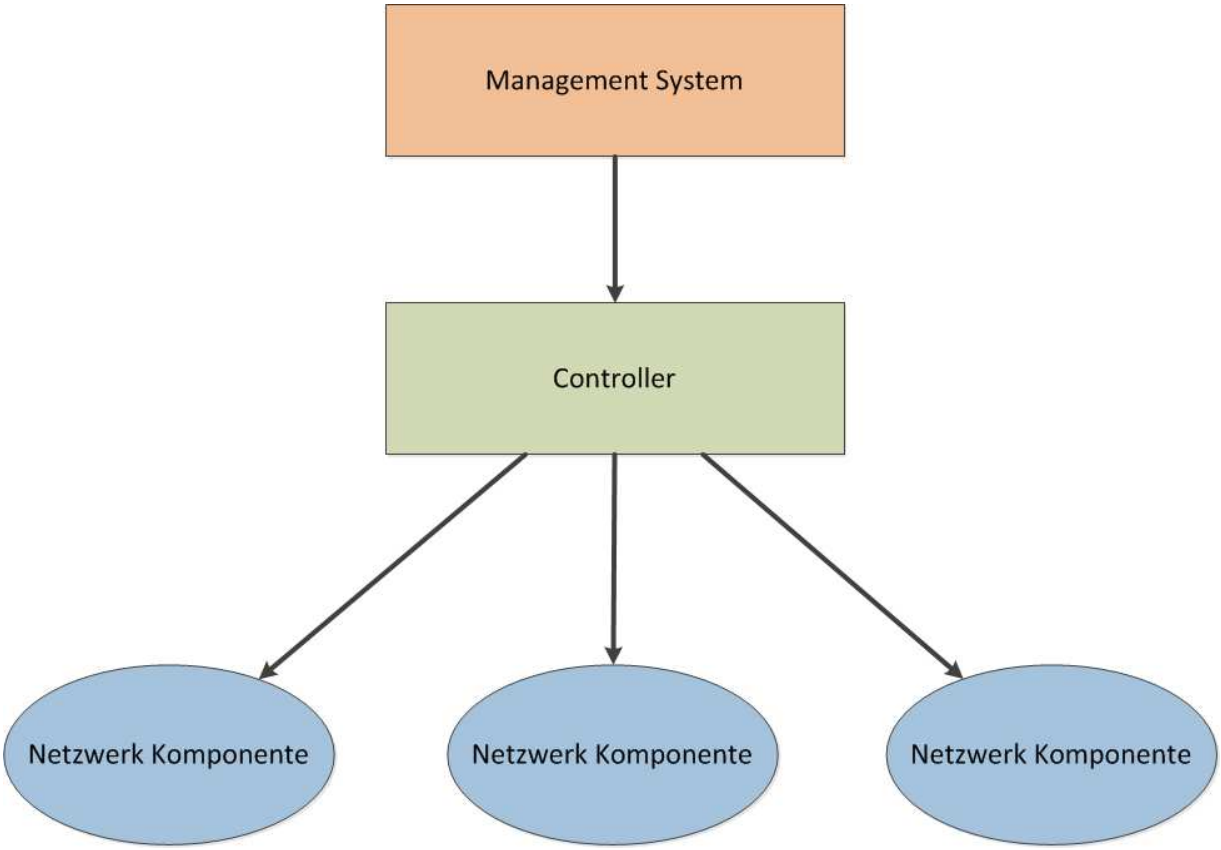
Es gibt unterschiedliche Varianten, wie ein Software Defined Network realisiert werden kann. Jedes hat seine Vor- und Nachteile. Diese Varianten werden in diesem Kapitel analysiert und verglichen.

5.3.1 Centralized SDN

Bei dem zentralen SDN geht es darum, das Netzwerk beziehungsweise die Datenübertragung einfach zu gestalten. Die Datenübertragung wurde, schon zu Beginn vom Internet, sehr einfach gestaltet. Protokolle wie das Internet Protokoll oder auch Transportprotokolle wie UDP und TCP sind sehr einfach aufgebaut. Damit ist eine Kommunikation zwischen Geräten möglich, die nicht schwierig ist aufzubauen. Als später jedoch die Anforderungen an die Netzwerke stiegen, wurde die Datenübertragung immer komplexer. Es sollte möglich sein, Daten zu übertragen, ohne dass der genaue Inhalt bekannt sein muss. Weiter wollten die Betreiber von grösseren Netzwerken auch bestimmen können, welche Daten auf welchen Wegen transportiert werden. So können teure Links gemieden werden und nur kostengünstigere verwendet werden. Eine weitere Anforderung bestand darin, dass die einzelnen Daten unterschiedlich schnell übertragen werden sollen. Voicedaten sollen also die Möglichkeit haben, weniger wichtige Daten zu „überholen“.

All diese Anforderungen machten das Netzwerk flexibler, jedoch auch sehr viel komplexer. Mit dem Centralized SDN wird versucht, die Datenübertragung wieder einfacher zu gestallten. Dabei soll sich die Netzwerkhardware wieder nur um die Datenübertragung kümmern. Die ganzen Berechnungen und Entscheidungen sollen an einem zentralen Ort getroffen werden. Die Kosten für die Netzwerkkomponenten können so sehr tief gehalten werden.

Der Vorteil bei dieser Variante ist, dass es keine Rolle spielt, von welchem Hersteller die Hardware gekauft wird. Sie muss nur das korrekte Souhtbound API besitzen, um mit dem Controller kommunizieren zu können. So können sehr flexible Netzwerke aufgebaut werden. Weiter kann ein sehr kostengünstiges Netzwerk aufgebaut werden. Dadurch ist es möglich, einige Switch auf Reserve zu kaufen. Bei einem Hardwareausfall wird der entsprechende Switch einfach ersetzt und der Controller spielt die Konfiguration wieder auf den Switch zurück. Es müssen nicht mehr teure Wartungsverträge aufgesetzt werden, damit die Hardware schnell ersetzt wird. Ein negativer Punkt bei dieser Variante ist jedoch bei einem Ausfall von einem Link zu finden. Ein Switch kann nicht selbst auf diese Situation reagieren. Er muss beim Controller nachfragen, was er machen soll. Der Controller leitet die Anfrage an den Orchestrar weiter, welcher nun die Berechnung vornimmt. Ist eine Lösung gefunden wird sie an den Switch geschickt. Die Datenübertragung kann nun fortgesetzt werden. Braucht nun die Berechnung der Lösung sehr lange, sind die Ausfallzeiten auch entsprechend gross. Dem kann entgegengewirkt werden, indem ein alternativer Pfad schon vorgängig berechnet wird. Dies wird zum Beispiel von EIGRP oder UplinkFast verwendet. Ein weiterer Nachteil ist die Skalierbarkeit. Wenn ein Netzwerk wächst, wird die Last auf dem Controller bald sehr gross. Weiter werden die Distanzen immer grösser was sich negativ auf die Antwortzeiten auswirkt. Die Performance kann auch allgemein leiden. Bei grösseren Veränderungen in einem Netz kommen auch viele Anfragen auf den Controller zu. So kommt es zu Wartezeiten.

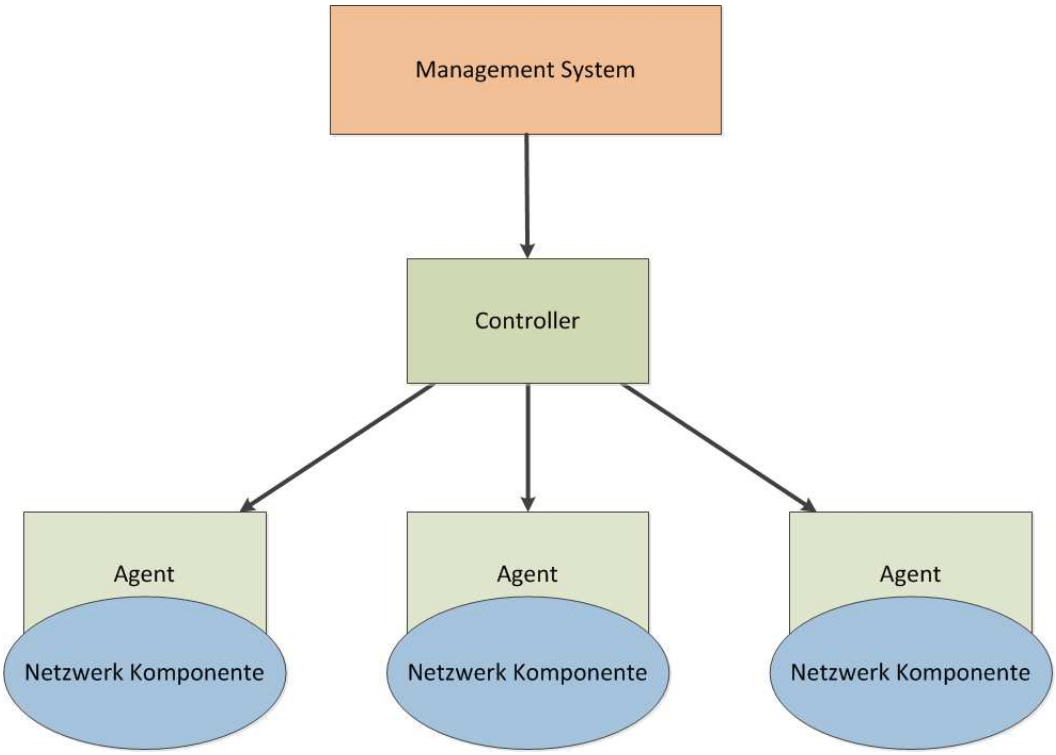


Centralized SDN

5.3.2 Distributed SDN

Bei diesem Ansatz hat jede Netzwerkkomponente einen kleinen Controller, einen sogenannten Agenten, auf der Hardware. Dieser Agent hat einige wenige Aufgaben. Das heisst, dass nicht die ganze Intelligenz an einem zentralen Ort ist. So können einige Entscheidungen schneller getroffen werden, als wenn jedes Mal der Controller angefragt werden muss. Diese Technik wird auch in heutigen WLAN Netzen angewendet. Der Access Point behandelt die Pakete, die kurze Antwortzeiten erfordern. Zum Beispiel Beacon Frames, Probe Response und ACK Pakete. Andere Pakete, wie Authentication und Association werden an den WLAN Controller gesendet und dort verarbeitet. Ein SDN Switch könnte zum Beispiel selber entscheiden, was geschehen soll, wenn ein Link ausfällt. In der Flowtable steht der alternative Pfad schon drin, jedoch mit einer geringeren Priorität. Fällt nun der Link aus, markiert der Switch alle Flows, die nun nicht mehr gültig sind. Das Weiterleiten der Pakete geht nun weiter, ohne dass auf eine Berechnung vom Orchestrar gewartet werden muss. Der Switch meldet den Ausfall nur, damit eine Statistik geführt werden kann und ein Alarm ausgelöst wird. Diese Switch haben auch eine höhere Performance als Switch ohne Agenten. So können mehr Daten verarbeitet werden.

Da sich diese Technik in WLAN Netzen bewährt hat, könnte sie auch in SDN Netzwerken erfolgreich eingesetzt werden. Einige Nachteile hat diese Technik natürlich auch und soll hier erwähnt werden. Da in jeder Netzwerkkomponente mehr Logik drin ist, wird die Hardware teurer. Weiter ist es sehr unwahrscheinlich, dass unterschiedliche Hersteller die gleichen Aufgaben an den Agenten übergeben. So kann Hardware von unterschiedlichen Anbietern nicht parallel betrieben werden. Dadurch sind diese Netze auch nicht so flexibel, wie beim Centralized SDN.



Distributed SDN

Hybrid SDN

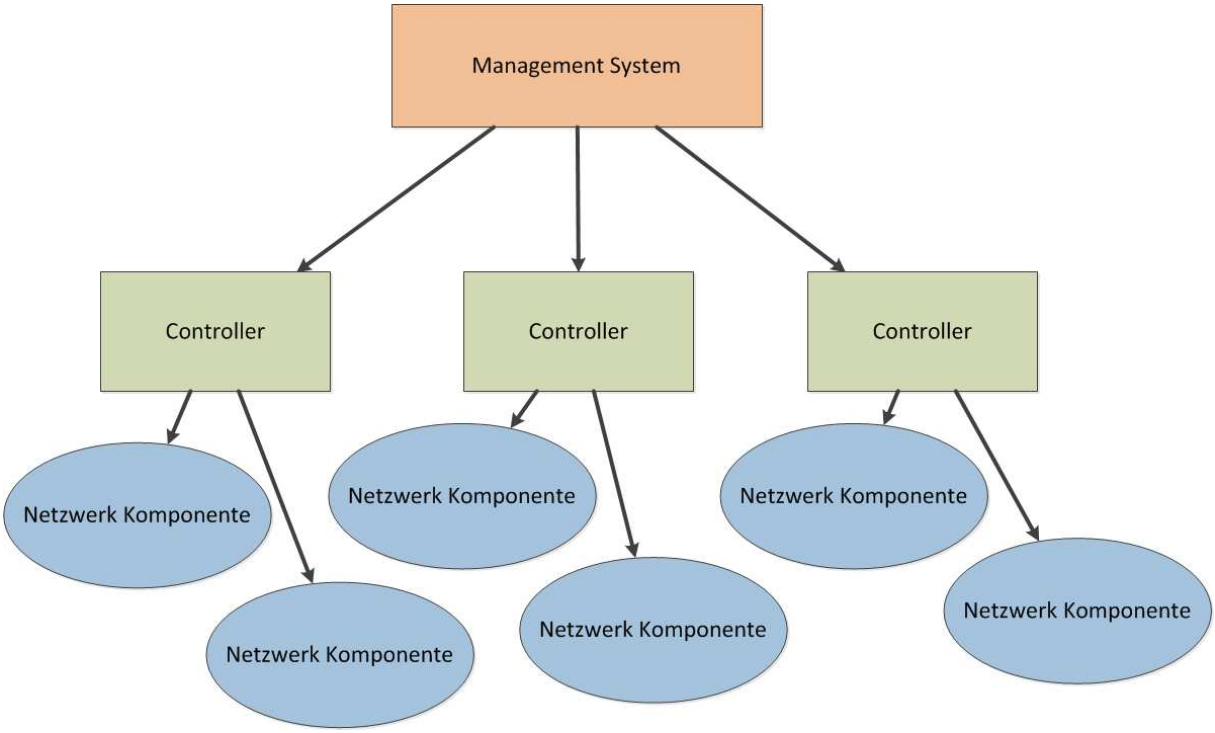
Beim hybriden SDN wird versucht, eine Mischung zwischen der zentralen und der dezentralen Intelligenz zu finden.

Es werden mehrere Controller eingesetzt, die im gesamten Netzwerk verteilt sind. Zum Beispiel kann in jedem Branche Office ein eigener Controller eingesetzt werden und noch ein Cluster in Hauptquartier der Firma. Dieser Cluster übernimmt die Aufgaben eines Controllers im Branch Office, falls dieser ausfällt. Die Konfiguration für das gesamte Netzwerk wird nun auf dem Orchestrar erledigt und an alle Controller weitergeleitet. Von da gelangen die Änderungen zu den Switch. Lokale Änderungen können auch direkt auf den Controllern gemacht werden. So können schnell lokale Änderungen gemacht werden, ohne dass der Administrator aus dem Headquarter involviert ist. Zum Beispiel kann in einem Branch Office eine bestimmte Applikation mit einer garantierten Bandbreite versorgt werden. Diese Applikation wird jedoch nur in dieser Aussenstelle eingesetzt. Vom Orchestrar aus können also bestimmte Berechtigungen an die einzelnen Controller verteilt werden. So können die Administratoren schnell reagieren, wenn eine Anpassung gemacht werden muss.

Auf der anderen Seite können auch Konfigurationen für das gesamte Firmennetz ausgeführt werden. Dafür wird eine Policy auf dem Orchestrar erstellt. Diese wird danach auf die Controller an den jeweiligen Standorten übermittelt. Als Beispiel wird hier die Bevorzugung von Voice Traffic im ganzen Netzwerk erwähnt.

Die Zuordnung von den Netzwerkkomponenten zu den Controllern wird normalerweise statisch ausgeführt. Dies hat jedoch einen grossen Nachteil. Der Controller muss so dimensioniert sein, dass er auch mit einer Spitzenlast umgehen kann. Fordern viele Netzwerkkomponenten vom selben Controller Flows an, kann dieser überlastet werden. In einer solchen Situation kommt es zu Wartezeiten auf dem Controller und somit zu Verzögerungen im gesamten Netzwerk. Als Lösung ist hier eine dynamische Verteilung der Netzwerkgeräte auf die Controller beschrieben. Dabei wird ein Cluster mit mehreren Controllern erstellt. Die Netzwerkgeräte verbinden sich nun auf die einzelnen Controller. Stösst dieser Cluster nun an seine Grenzen, können dynamisch weitere Controller hinzugefügt werden. Sinken die Anforderungen wieder können natürlich auch einige Controller wieder entfernt werden. Somit ist eine sehr flexible Lösung gefunden, die fast beliebig skalierbar ist.

In einer grossen Firma mit mehreren Branch Offices kann der lokale Controller als Cluster definiert werden für alle Switch in dieser Branch. Müssen nun viele Anfragen beantwortet werden, wird ein Controller aus dem Hauptquartier zum Cluster hinzugefügt. Einige Anfragen haben nun eine etwas längere Antwortzeit, da sie über einen WAN Link zum Hauptquartier geschickt werden. Dies ist jedoch schneller, als wenn sie von einem überlasteten Controller verworfen werden und nochmals geschickt werden müssen.



Eine weitere Möglichkeit vom Hybriden Ansatz ist die Zuordnung der Controller zu Einsatzgebieten. Ein Controller übernimmt die Konfiguration vom Netzwerk, ein zweiter verwaltet das Datacenter und ein dritter das WLAN Netz. Nun wird ein Management System benötigt, das mit allen drei Controllern kommunizieren kann. Der Informationsfluss kann nun von einem Controller zum Management System fliessen und danach zu einem anderen Controller weitergeleitet werden. So können Änderungen, die in einem Teil vom Netzwerk gemacht wurden, auf einen anderen einen Einfluss haben. Dieser Ansatz ist viel versprechend, weil die Last der einzelnen Controller verteilt wird.

5.3.3 Pro und Contra der Architekturen

In der nachstehenden Tabelle werden die Vor- und Nachteile der einzelnen Architekturen zusammengefasst. Die Vorteile sind grün eingefärbt, die Nachteile rot. Wenn eine Eigenschaft nicht eindeutig als Vor- oder Nachteil bewertet werden kann, wird sie orange eingefärbt.

	Preis	**Skalierbarkeit**	**Performance**	**Flexibilität**	**Abhängigkeit**
Centralized SDN	Kostengünstige Hardware	Niedrig	Niedrig	Hoch	Herstellerunabhängig
Distributed SDN	Teure Hardware	Niedrig	Hoch	Niedrig	Herstellerabhängig
Hybrid SDN	Hoch	Hoch	Mittel	Hoch	Herstellerunabhängig

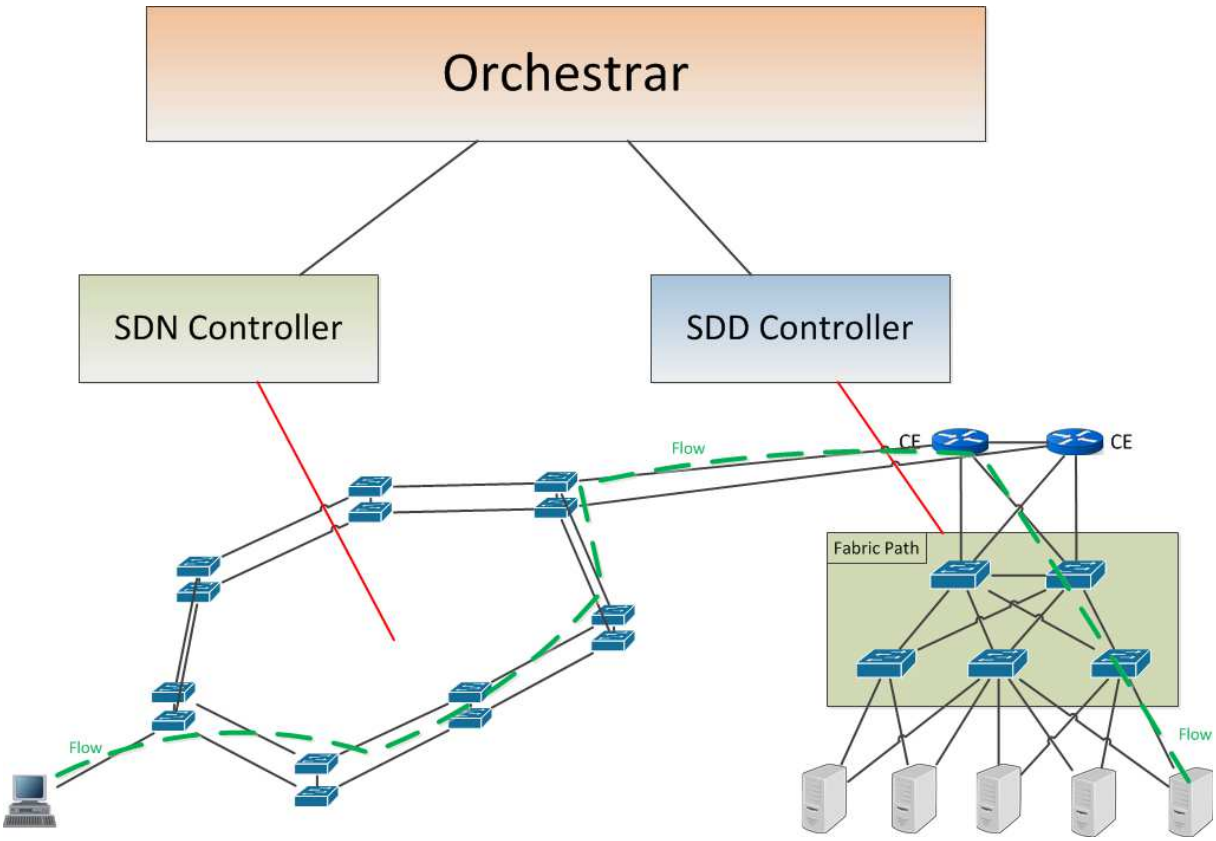
Für die Auswertung werden den Vorteilen 3 Punkte vergeben, den Nachteilen 1 Punkt und den mittleren Bewertungen 2 Punkte.

Mit dieser Skala kommt Centralized SDN auf 11 Punkte. Distributed SDN kommt auf 7 Punkte und Hybrid SDN gewinnt knapp mit 12 Punkten.

Meiner Meinung nach ist jedoch eine Mischung zwischen dem Distributed SDN und dem Hybriden SDN die Lösung mit den grössten Vorteilen. Die Hardware besitzt einen Agenten und ist dadurch sehr performant. Weiter werden mehrere Controller verwendet pro Einsatzgebiet. Das heisst für den Bereich Netzwerk werden mindestens zwei SDN Controller eingesetzt. Im Datacenter werden auch mindestens zwei SDD Controller verwendet. All diese Controller werden durch einen Orchestrar verwaltet. Dabei werden möglichst viele Konfigurationen auf die Switch gespielt. So können die Switch schon auf Veränderungen reagieren und müssen nicht den Controller um Hilfe bitten.

Definition von SDN

Nach diesem ersten Einblick in SDN soll definiert werden, wo SDN beginnt und wo dass es endet. Man hört heute häufig, von einem Datacenter Szenario. Wenn eine virtuelle Maschine verschoben wird, muss der SDN Controller dies bemerken und darauf reagieren können. Aus meiner Sicht ist dies jedoch nicht die Aufgabe von einem SDN Controller. Bei SDN geht es nur um das Netzwerk nicht um ein Datacenter. Da jedoch ein Netzwerk auch auf Veränderungen in einem Datacenter reagieren muss, braucht es noch einen Software Defined Datacenter (SDD) Controller. Dieser kann in eine Virtualisierungslösung integriert sein oder auch eine eigenständige Komponente sein. Im untenstehenden Bild ist das Zusammenspiel zwischen einem Netzwerk und einem Datacenter dargestellt.



SDN und SDD

Auf der linken Seite ist ein Netzwerk dargestellt, das von einem SDN Controller verwaltet wird. Auf der rechten Seite ist ein Datacenter abgebildet, welches vom SDD Controller verwaltet wird. Im Datacenter wird nun ein Service angeboten, welcher über einen Flow (grüne, gestrichelte Linie) zu einem Kunden transportiert wird. Der SDN Controller ist nur für das Netzwerk zuständig, nicht jedoch für das Datacenter. Im Kapitel „Networkmanagement und Automation“ wird dieses Szenario genauer behandelt. Im Moment wird nur besprochen, wie die beiden Controller zusammenspielen können. Der Orchestrar kann von einem Controller eine Information entgegennehmen und resultierende Befehle an den anderen Controller weitergeben. Die Controller kommunizieren nicht direkt miteinander.

5.4 Anforderungen an SDN

Durch die Entwicklung von Software Defined Network wird versucht die Kosten für das Management von Netzwerken zu verringern. Man spricht dabei von Kosten pro übertragenem Bit oder kurz Cost/Bit. Man könnte meinen, dass diese Kosten kleiner werden, wenn das Netzwerk grösser wird. Leider ist das Gegenteil der Fall. Ein grösseres Netz ist schwieriger zu managen, da eine Anpassung viel mehr Zeit für die Planung benötigt als in einem kleinen Netz. SDN soll nun diese Kosten reduzieren. Weiter soll, durch das zentrale Management, ein flexiblerer Einsatz des Netzwerks ermöglicht werden. So sollen zum Beispiel flexiblere Dienste angeboten werden können. Der Kunde entscheidet selbst welches Angebot er nutzen möchte. Weiter soll auf Inputs von anderen Controllern reagiert werden können. Dies kann zum Beispiel ein Software Defined Datacenter Controller oder auch ein Wireless Lan Controller sein. Ein SDN Netzwerk soll auch schneller auf eine Veränderung reagieren können. Das heisst, wenn ein Netzwerk nicht mehr erreichbar ist, soll dies innert kürzester Zeit im ganzen Netz bekannt sein und entsprechende Massnahmen ergriffen werden. Solche Massnahmen können zum Beispiel das Alarmieren oder das Umschalten auf einen anderen Link sein. Weiter sollen auch Möglichkeiten bestehen, wie bestimmten Datenverkehr umzuleiten. Bei einer Auslastung von 80 Prozent soll zum Beispiel der Voice Datenverkehr auf eine alternative Verbindung umgeleitet werden. Durch solche Umlenkung von Verkehr sollte das Netz stabiler werden, da ein Link nicht so schnell ausgelastet wird. Auch einzelne Geräte können so vor Überlast geschützt werden. Weiter wird von SDN erwartet, dass das Auffinden und Beheben von Fehlern einfach und schneller erledigt werden kann. Dazu soll der Controller oder das Management System nützliche Hilfestellung leisten. Als Beispiel soll eine ganze Verbindung zwischen Source und Destination angezeigt werden. Weiter kann das Management System auch möglich Probleme aufzeigen und einen Vorschlag für die Behebung machen.

In den Nachfolgenden Kapiteln werden diese Anforderungen überprüft und hinterfragt.

5.5 Use Cases

In diesem Kapitel wollen wir uns einzelne Use Cases anschauen. Dabei wird ein Vergleich gemacht, zwischen Software Defined Network und einem heutigen Netzwerk.

5.5.1 Netzmanagement und Automation

Beim Netzwerkmanagement geht es um die Verwaltung, Betrieb und die Überwachung von Netzwerkgeräten. Hier werden die fünf Bereiche des Netzwerkmanagements besprochen. Dabei handelt es sich um das Fehlermanagment, Konfigurationsmanagement, Abrechnungsmanagement, Leistungsmanagement und das Sicherheitsmanagement. Als erstes wird das Fehlermanagement in einem heutigen Netzwerk angeschaut.

Beim Fehlermanagement geht es darum, einen Fehler im Netzwerk zu finden und zu isolieren. In einem heutigen Netzwerk wird dies meistens über das Command Line Interface erledigt. Es gibt zwar Monitoring Tools die helfen den Fehler einzugrenzen, die exakte Lokalisierung muss jedoch mit der CLI erledigt werden. Dabei werden Informationen über die einzelnen Geräte und Prozesse angezeigt. Dies könnte der Status von einem Routingprozess oder von einem Link sein, welche Nachbarn von einem Routingprozess erkannt wurden und welche Netze ausgetauscht wurden. Weitere nützliche Hilfsmittel sind der Ping Befehl und Traceroute. Mit diesen Tools kann man die Erreichbarkeit von Routern und Clients überprüfen. Nun müssen von den einzelnen Netzwerkgeräten die Informationen zu einem Gesamtbild zusammengesucht werden. Dies erfordert, vom Systemadministrator, ein gutes Verständnis vom gesamten Netzwerk. In einem grossen Netzwerk ist dies oft schwierig zu erlangen. So ist es auch schwierig den Überblick zu bewahren und den Fehler schnell zu finden. Es existieren einige weitere Tools auf dem Markt, die dem Administrator eine Hilfe sein können bei den Fehlersuchen. Als Beispiel werden hier Observium, Icinga und Smokeping erwähnt, um nur einige zu nennen. Observium und Icinga verwenden SNMP um Abfragen an ein Netzwerkgerät zu senden. Dabei spielt es keine Rolle, ob es sich um einen Switch, Router, Firewall oder sonst ein Gerät, das SNMP unterstützt, handelt. Mit diesen Abfragen kann nun grafisch dargestellt werden, ob ein Gerät erreichbar ist, wie die Auslastung der CPU und des Memorys ist oder wie hoch die Temperatur im Innern des Gerätes ist. Eine weitere Möglichkeit, um Systemabfragen zu stellen, ist SSH. Smokeping sendet ICMP Echo Request an die Geräte im Netzwerk. Wird ein ICMP Echo Reply empfangen, wird dieses Gerät als erreichbar aufgelistet. Hier kann auch gleich die Latenz mit angegeben werden. Das heisst, wie lange das Netzwerkgerät für eine Antwort brauch. Hier eingerechnet ist die Round Trip Time (RTT), das ist die Zeit, die benötigt wird, um ein Paket zum Ziel und wieder zurück zu senden und die Reaktionszeit vom Zielsystem.

All diese Managementsysteme können auch in Kombination eingesetzt werden. So können die unterschiedlichen Vorteile ausgenutzt werden. Einige dieser Systeme könne frei eingesetzt werden. Einige nur bis zu einem bestimmten Volumen an Loggingdaten. Die Administratoren müssen nun diese Systeme in Betrieb nehmen, was nicht immer so einfach ist. Die Anforderungen an die Serversysteme, auf denen ein Management System betrieben wird, sind unterschiedlich und müssen im Vorfeld abgeklärt werden. Weiter ist das Einrichten unterschiedlich und nicht immer ganz einfach. Für Abfragen per SSH muss zum Beispiel ein Username und ein Passwort auf jedem einzelnen Gerät

eingerichtet werden. Als Alternative zum Passwort kann auch ein RSA Key erstellt werden. Der Public Key von den Netzwerkkomponenten wird nun auf das Management System gespielt. Dieses kann sich nun damit authentifizieren gegenüber dem Switch oder Router. Sollen die Abfragen per SNMP gemacht werden, muss die Version beachtet werden. Bei SNMPv2 muss ein Community String gesetzt werden. Dieser wird wie ein Passwort verwendet, jedoch wird er im Klartext über das Netzwerk übertragen. Diese Version ist also nicht geeignet wenn die Sicherheit eine Rolle spielt. Bei SNMPv3 wird die gesamte Kommunikation mittels eines Passwortes verschlüsselt. Dazu muss auf allen Netzwerkkomponenten ein Username und ein Passwort erstellt werden.

Bei all diesen Management Systemen zeigt sich wieder der typische Nachteil von der verteilten Intelligenz. Das Einrichten erfolgt auf jedem einzelnen Netzwerkgerät.

In einem Software Defined Network kann der Netzwerkadministrator eine Verbindung, vom Orchestrar, testen lassen. Wenn ein Benutzer sich beklagt, dass er nicht auf eine Webseite zugreifen kann, kann der Admin herausfinden wieso. Er loggt sich auf den Orchestrar ein und öffnet die Troubleshooting Seite. Danach wird der Benutzer ausgewählt und das „Internet“. Nun überprüft der Orchestrar als erstes, ob die Policy eine Verbindung von diesem Benutzer ins Internet erlaubt. Falls ja, generiert der Orchestrar alle nötigen Flows für diese Verbindung und fragt alle betroffenen Switch nach den Flows. Sobald die Switch ihre Flows an den Orchestrar gesendet haben, vergleicht dieser seine generierten Flows mit denen, die auf den Switch sind. Wenn sie nicht übereinstimmen, wird der Admin angefragt ob dies behoben werden soll. Stimmen sie jedoch überein, liegt der Fehler nicht im Netzwerk.

Beim Konfigurationsmanagement geht es um die eigentliche Konfiguration von einem Netzwerkgerät. Heute wird dies auch wieder über das Command Line Interface gemacht. Dabei ist es möglich, dass man von einem Textfile alle Befehle in das Terminalfenster kopiert. So müssen nicht alle Kommandos von Hand eingegeben werden. Weiter gibt es noch die Möglichkeit, ein Gerät über SNMP zu konfigurieren. Dabei wird ein Script erstellt, das dann die benötigten Schritte ausführt. Bei SNMP ist es jedoch nötig, dass die Befehle in der Reihenfolge im Script stehen, wie sie auf dem Gerät ausgeführt werden müssen. Weiter kann SNMP auch nicht zwischen Konfiguration und Nicht-Konfiguration unterscheiden. In ein Backup gehören nur die Konfigurationen. Dies macht es sehr schwierig, ein Backup von einer Konfiguration zu erstellen und später wieder einzuspielen. Ein weiteres Problem von SNMP ist, dass bei einem Fehler keine Rückmeldung gesendet wird. Die Konfiguration schlägt dann fehl ohne das der Administrator weiss weshalb. Dies macht dann die Fehlersuche sehr aufwändig. Dies ist der Hauptgrund, weshalb sich SNMP nie wirklich durchgesetzt hat und CLI Scripting am häufigsten benutzt wird. Der Nachfolger steht jedoch schon in den Startlöchern und nennt sich Network Configuration Protocol (Netconf). Netconf unterstützt Transactions. So ist es möglich, einem Gerät einige Befehle zu übergeben und das Gerät führt sie dann alle aus. Dabei spielt es keine Rolle, in welcher Reihenfolge die Befehle übergeben wurden. Auch wenn eine Aktion nicht ausgeführt werden kann, schlägt die ganze Transaktion fehl. So kann sichergestellt werden, dass keine Fehler im Netzwerk übrigbleiben. Mit Netconf ist es also möglich, ein Backup von einer Konfiguration zu erstellen und diese zu einem späteren Zeitpunkt oder auf einem anderen Gerät einzuspielen, ohne dass Fehler geschehen, die das Netzwerk beeinflussen. Weiter kann auch ein ganzer Service im Netzwerk konfiguriert werden. Erst wenn der gesamte Service erstellt ist, ist die Transaction abgeschlossen. Das heisst, dass der Service erstellt wird, oder

in einem Fehlerfall nicht erstellt wird. Jedoch gibt es keine Konfigurationsüberbleibsel. Mit Netconf können also die Kosten für das Netzwerkmanagement reduziert werden da die Netzkonfiguration automatisiert werden kann, ohne dass schwierig zu findende Fehler auftreten.

Bei Software Defined Network müssen die Geräte natürlich auch konfiguriert werden. Da OpenFlow nur auf den Data Plane Einfluss nehmen kann, jedoch nicht auf den Management Plane, braucht es auch bei SDN ein Protokoll wie Netconf. Mit OpenFlow kann zum Beispiel kein Interface erstellt werden und auch keine IP Adresse darauf konfiguriert werden. OpenFlow kann nur, mit Hilfe von einem Flow, einen Dienst zur Verfügung stellen. Mit SDN ist es jedoch möglich, Netconf in den Orchestrar zu integrieren. Dann kann von einem einzigen Managementinterface aus das Netzwerk konfiguriert werden und auch Dienste zur Verfügung gestellt werden.

Zum Netzwerkmanagement gehört auch eine Abrechnung dazu. Zum einen kann das die Verrechnung von einem Service gegenüber von einem Kunden sein aber auch das Abspeichern der Befehle, die ein Administrator ausgeführt hat. Beim Verrechnen von einem Service kann es darum gehen, wie lange ein Kunde den Dienst benutzt hat. Dabei muss natürlich die Zeit erfasst werden. Heute kann dies mit einem Radius Server erledigt werden. Wenn der Kunde den Service benutzt sendet der Server eine Start Nachricht an das Verrechnungssystem. Wird der Service nicht mehr benötigt wird eine Stopp Nachricht an das Verrechnungssystem gesendet. So kann die genaue Zeit der Benutzung erfasst werden. Beim Erfassen der Kommandos, die ein Administrator im Netzwerk ausführt, geht man ähnlich vor. Bei jedem ausgeführten Kommando wird eine Nachricht an ein System gesendet. So kann bei einer Fehlmanipulation schneller aufgeklärt werden, was schiefgelaufen ist.

Bei SDN kann das Verrechnungssystem direkt in den Orchestrar integriert werden. So muss nicht noch ein weiteres System betreut werden. Die Verrechnung von einem Service kann zum Beispiel über die Flows gemacht werden. Ein Flow erfasst die Anzahl Bytes und die Anzahl Pakete die er transportiert. So kann eine sehr detaillierte Abrechnung erstellt werden. Die Befehle, die ein Administrator über das GUI an das Netzwerk sendet müssen natürlich auch protokolliert werden. Dies kann immer dann gemacht werden, wenn die Änderungen gespeichert werden oder ins Netzwerk geschickt werden. Die Änderungen, die der Orchestrar ausführt sollen auch erfasst werden. Wenn ein neues Interface erstellt wird, oder ein neuer Flow generiert wird, wird eine Logmeldung an den Orchestrar gesendet.

In einem Netzwerk muss laufend die Leistung überwacht werden. Das bedeutet, wie viel Bandbreite wird benötigt und wie viel steht noch zur Verfügung, welche Verzögerungen sind auf den einzelnen Links und welche Paketverluste werden gemessen. Mit diesen Messresultaten können zukünftige Services geplant werden. Dabei kann festgestellt werden, ob die Bandbreite ausgebaut werden muss, ob die Latenz ausreicht für den neuen Service oder ob zu viele Pakete verloren gehen. Heute wird dies zum Beispiel mit NetFlow gelöst. Dabei senden alle Netzwerkgeräte die relevanten Informationen an einen Server. Da können sie dann ausgewertet werden.

In einem SDN Netz übernimmt der Orchestrar diese Analyse. Die relevanten Informationen können über die Meters von OpenFlow oder über Netconf ausgelesen werden. Auf dem Managementinterface können danach die Daten grafisch dargestellt werden. Bei SDN ist also die

Analyse vom Netzwerk direkt in das Managementsystem integriert. So ist der Look and Feel gleich und ein User findet sich schneller zurecht.

Als Letztes wird noch das Thema Sicherheit von den Netzwerkgeräten behandelt. Die Sicherheit vom gesamten Netzwerk wird im Unterkapitel „Sicherheit“ im Kapitel „Use Cases“ behandelt.

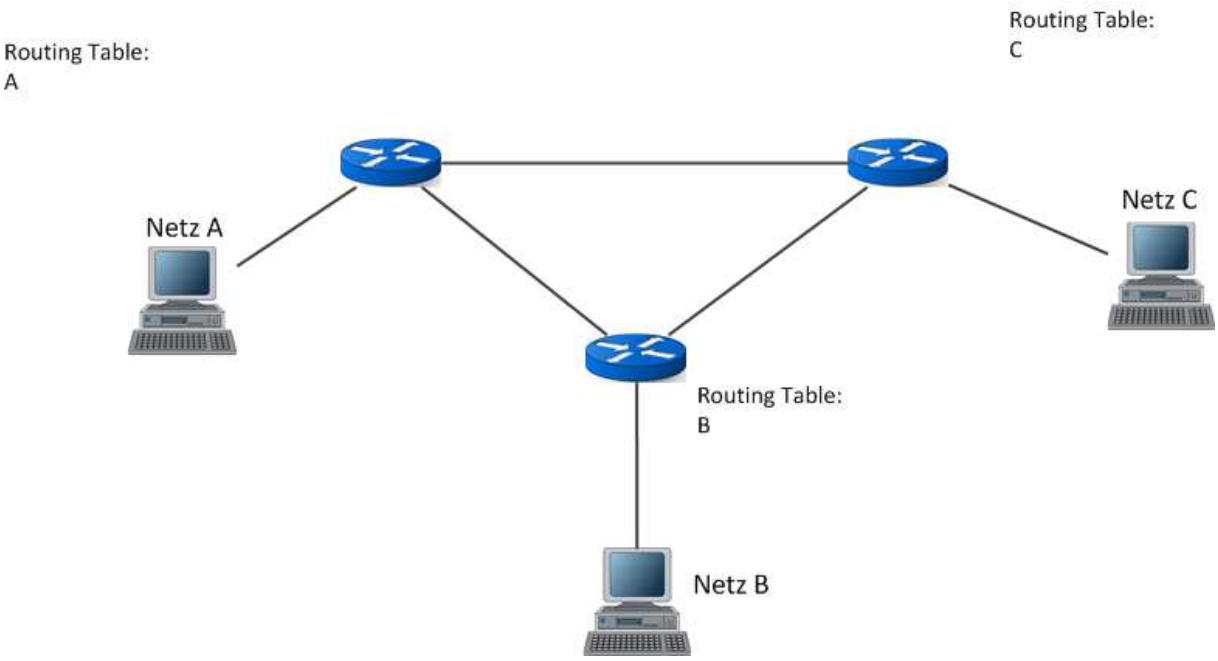
Ein Netzwerkgerät muss vor dem unerlaubten Zugriff von Drittpersonen geschützt werden. Heute muss als erstes der physische Zugang untersagt werden. Das heisst, dass ein Serverraum stets abgeschlossen sein muss und nur von berechtigten Personen betreten werden darf. Das gleiche gilt auch für die Serverschränke. Auch diese müssen immer verschlossen sein, damit niemand die Geräte beschädigen oder angreifen kann. Weitere Massnahmen sind das vergeben von einem Passwort für den Zugriff auf die Geräte, sowie das beschränken des Managementzugriffs auf bestimmte IP Adressen. Was auch nicht zu unterschätzen ist, ist ein Banner, der beim Login angezeigt wird. Dieser warnt den User, dass alle Aktionen protokolliert werden und Straftaten verfolgt werden. Auch sollen nur Protokolle für den Zugriff verwendet werden, die verschlüsselt sind. Sonst könnte ein Angreifer die Passwörter mitschneiden. In heutigen Netzwerken wird diese Konfiguration einzeln auf allen Geräten erledigt. In einem SDN Netz muss die Grundkonfiguration auch manuell und auf allen Geräten einzeln erledigt werden. Man hat jedoch die Möglichkeit, nur die IP Adresse des Controllers und das Transportprotokoll (TCP, TLS...) zu konfigurieren und den Rest vom Orchestrar aus zu machen. Dabei muss jedoch beachtet werde, dass sobald das Gerät im Netzwerk erreichbar ist, es auch angreifbar ist. Die beste Möglichkeit ist wohl, dass bei der Grundkonfiguration auch gleich die Sicherheit berücksichtigt wird. Es müssen also User und Passwörter erstellt werden, das Protokoll für den Zugriff gewählt werden und einen Login Banner erstellt werden. Weiter kann es auch noch nützlich sein, einen Flow zu erstellen, der jeglichen Verkehr auf dem Dataplane blockiert. So kann das Gerät erst Daten weiterleiten, wenn der Orchestrar Flows erstellt hat. Das Sicherheitsmanagement wird in einem SDN Netz also nicht einfacher und muss auch mit Sorgfalt erledigt werden.

5.5.2 Routing und Rerouting

In diesem Kapitel wird erklärt, wie die Pakete von den Clients den Weg durch ein Netzwerk finden.

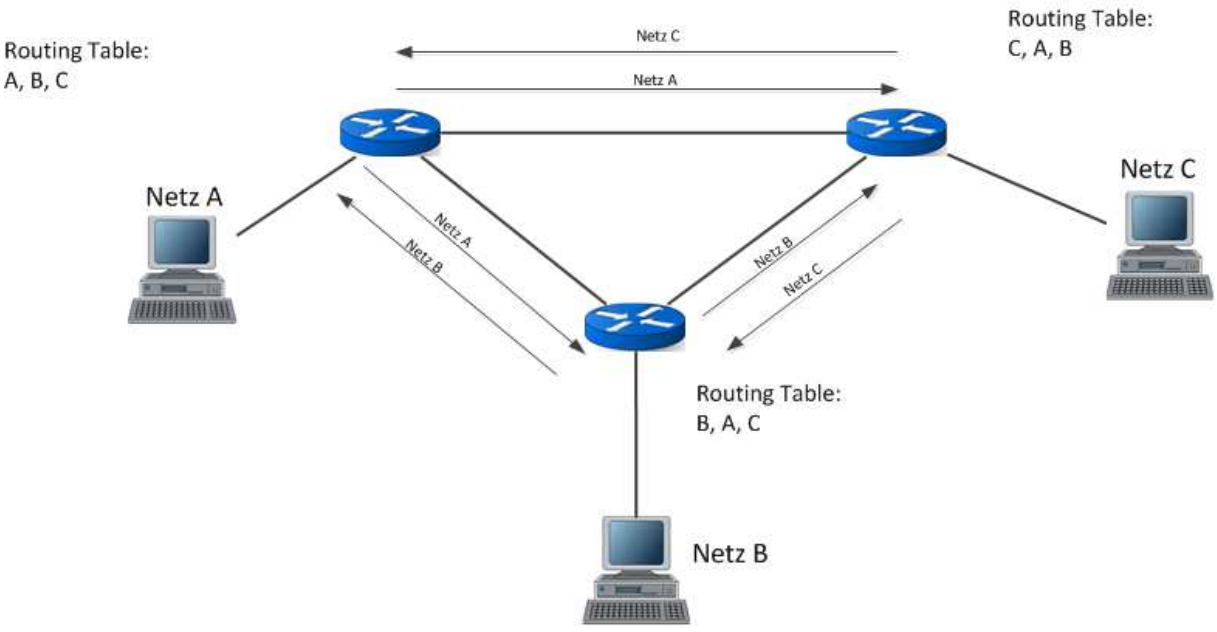
Dazu schauen wir uns zuerst an, wie ein heutiges Netzwerk funktioniert. Unterschiedliche Netzwerke werden heute mit sogenannten Routern verbunden. Diese Router haben Intelligenz, in Form von Routingprotokollen, eingebaut. Mit Hilfe dieser Protokolle können die Router miteinander kommunizieren. Ein Router teilt nun seinen Nachbarn mit, welche Netze bei ihm angeschlossen sind. Diese leiten diese Information an all ihre Nachbarn weiter und teilt gleichzeitig ihre angeschlossenen Netze mit. Alle diese Netze werden nun in der sogenannten Routingtabelle abgespeichert. Anhand dieser Tabelle entscheidet nun jeder Router, für sich selbst, wohin die Pakete weitergeleitet werden. Die Intelligenz ist also im gesamten Netzwerk verteilt. Dieser Ansatz funktioniert sehr gut. Das Problem findet sich jedoch in der Flexibilität. Soll nun Etwas an einem solchen Netz verändert werden, muss jedes einzelne Gerät um konfiguriert werden. Dies dauert natürlich, unter Umständen, sehr lange. Die Netzwerke müssen heute jedoch immer schneller auf Veränderungen reagieren

können. Ein entsprechender Ansatz wurde mit Software Defined Network (SDN) gefunden. Im nächsten Bild wird das Austauschen der Routing Informationen grafisch dargestellt.



Router kennen die direkt angeschlossenen Netze

Die Router kennen zuerst nur ihre direkt angeschlossenen Netze. Zum Beispiel kennt der Router links oben nur das Netz A.

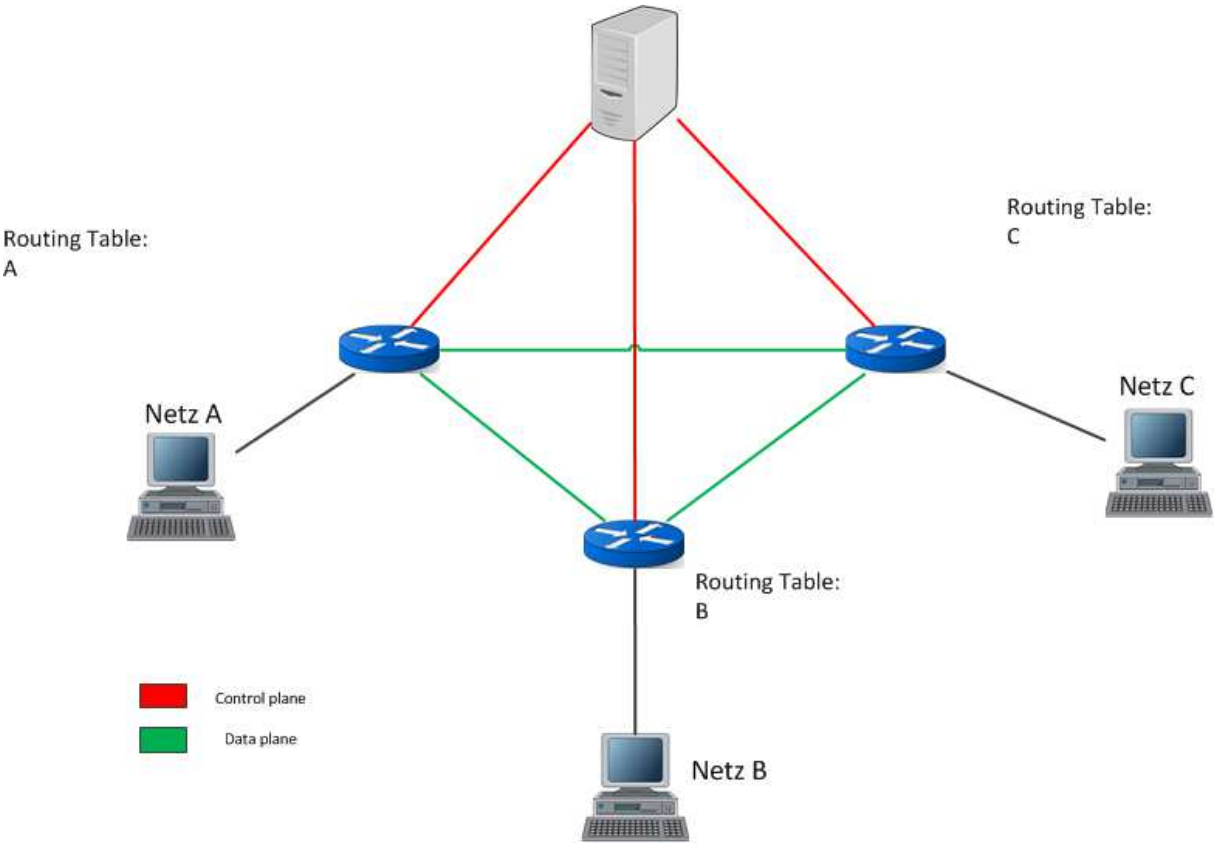


Router senden Routing Updates an ihre Nachbarn

Nun beginnen die Router Routing Updates an ihre Nachbarn zu senden. Als Beispiel wird hier wieder der Router links oben erklärt. Dieser sendet ein Routing Update an den Router unten und eines zum Router rechts. Diese zwei Router tragen nun das Netz A in ihrer Routing Tabelle ein. Gleichzeitig

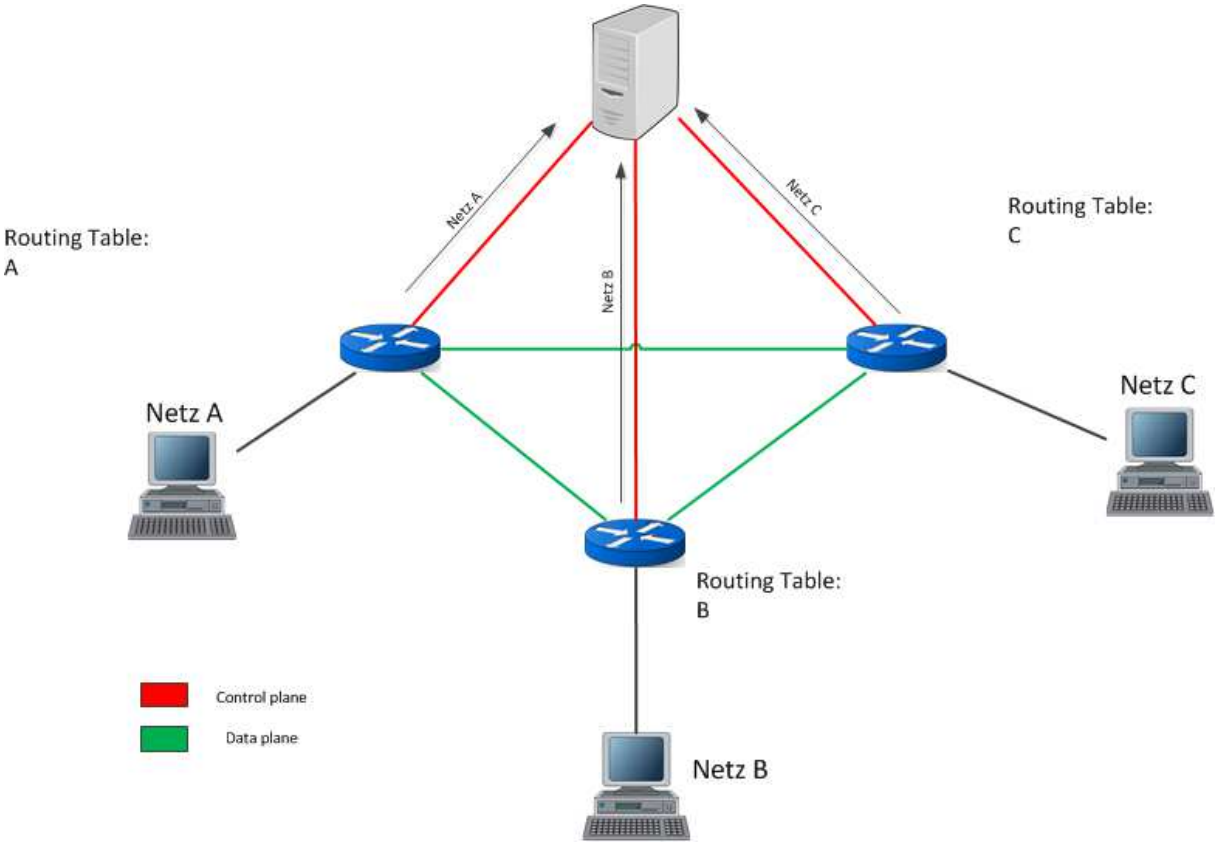
sendet der Router unten, je ein Update an seine Nachbarn. Diese können nun das Netz B in die Routing Tabelle eintragen. Auch der Router rechts sendet seinerseits zwei Updates. Nun kennen alle drei Router alle drei Netze. Bei grösseren Netzen kann dieser Vorgang einige Zeit in Anspruch nehmen.

Schauen wir uns nun an, wie das Erlernen von Netzwerken bei SDN aussieht. Dazu bedienen wir uns dem vorhergehenden Beispiel.



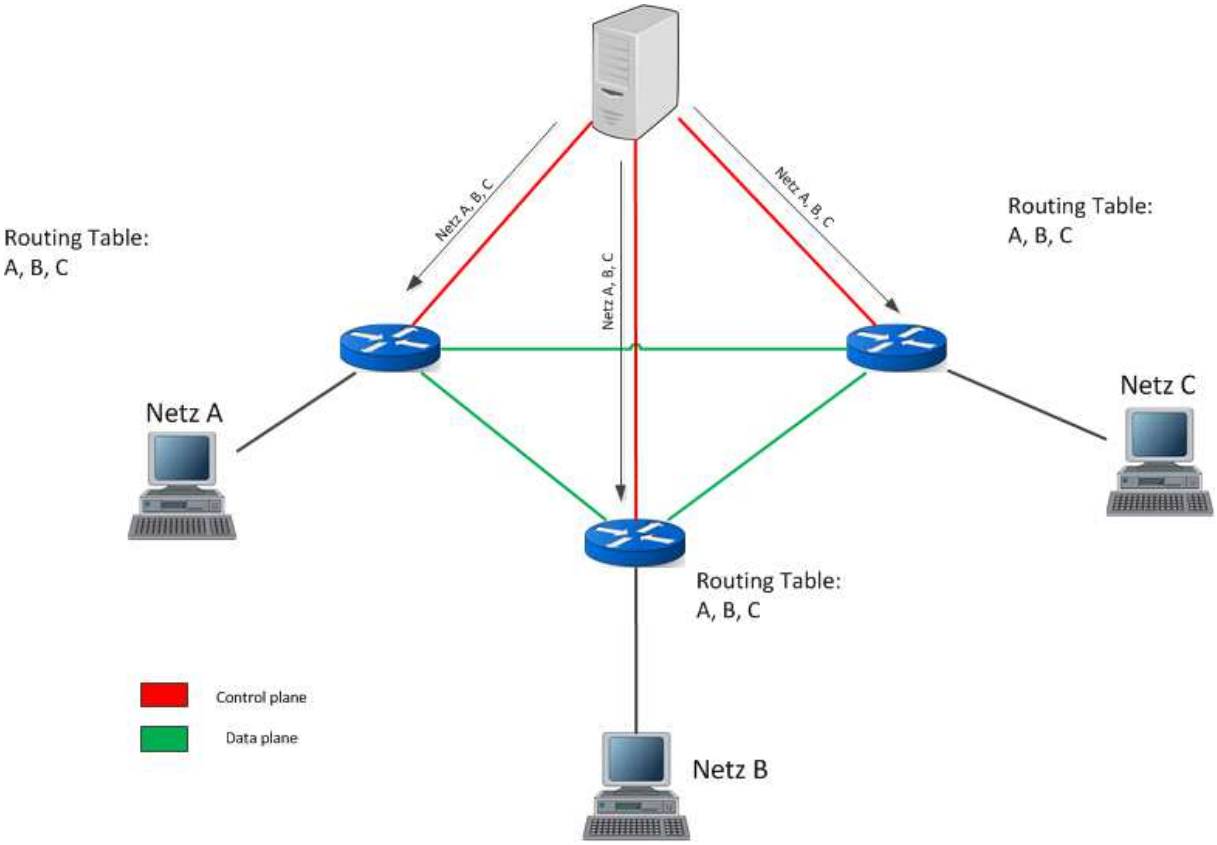
Routing mit SDN

Hier sehen wir die gleiche Topologie wie zuvor. Die drei Router sind untereinander verbunden. Dieses Mal kommt jedoch noch der zentrale Controller dazu. Dieser wird, über die roten Kabel, mit jedem Router verbunden. Diese Verbindungen können physikalisch oder auch logisch sein. In den Routing Tabellen stehen nur die direkt angeschlossenen Netze.



Routing mit SDN

Nun schicken alle Router ihre Routing Tabelle an den Controller. Da dies gleichzeitig geschieht, dauert es nicht lange.



Routing mit SDN

Der Controller entscheidet nun, welche Informationen für welchen Router sinnvoll sind. Danach werden die Routingtabellen zu den Routern geschickt. Auch dies geschieht wieder gleichzeitig. Die Router tragen nun ihre Routing Tabellen nach.

Meiner Meinung nach werden die Routingentscheidungen bei SDN schneller im Netz verteilt. Dies wird deutlich, wenn man ein Beispiel anschaut, bei dem nicht der direkt angeschlossene Link unterbrochen wird. Ein Router hat nur die Möglichkeit, anhand der fehlenden Antworten des Nachbarn, zu merken dass der Link unterbrochen ist. Dies dauert, je nach Routingprotokoll, von einigen Sekunden bis zu mehreren Minuten. Ein SDN Controller kann die Erreichbarkeit von einem Netzwerkgerät überprüfen. Sobeil eines nicht mehr erreichbar ist, wird dies im Netzwerk gemeldet. So kann schneller auf den Unterbruch reagiert werden.

In einem Software Defined Network werden heute meist nur Switch eingesetzt. Es gibt jedoch eine Möglichkeit, wie man aus einem Switch einen Router erstellen kann. Auch Google setzt auf dieses Verfahren. Es nennt sich RouteFlow.

RouteFlow ist ein Opensource Projekt mit dem Ziel, virtuelle Routing Service anzubieten. Dabei werden auf einem Server verschiedene Routing Engins betrieben. Diese werden in einer Linux virtuellen Maschine installiert. Als Routing Engine wird zum Beispiel Quagga eingesetzt. Diese wird nun wie ein Router konfiguriert. Es werden die Router ID gesetzt und alle angeschlossenen Netze eingetragen. Jede Routing Engine wird einem physikalischen Switch zugeordnet. Die Engine generiert nun die Forwarding Informatione Base für, den ihr zugewiesenen Switch. Diese Datenbank wird nun

zwischen den einzelnen Routing Instanzen via Routing Updates ausgetauscht. Der RouteFlow Server kann nun anhand der Forwarding Information Base eine Flow Tabel, für jeden Switch erstellen. Dabei wird für jede einzelne Route einen Flow erstellt. Das heisst, für jedes Source Netzwerkt wird ein Flow zu jedem Destination Netzwerk erstellt. So kann die gesamte Routing Topologie abgebildet werden. Diese Flow Tabels werden nun via Controller auf die einzelnen Switch geschickt.

Ein Vorteil dieser Technik ist, dass ein Fehler in einem Routing Prozess, also der Absturz einer Routing Engine, noch keinen Einfluss auf die Flow Tabels hat. Der Controller schickt erst Änderungen auf einen Switch, wenn die Routing Engine explizit eine Änderung erzeugt.

Ein Nachteil ist jedoch, dass ein Switch nicht feststellen kann, an welche Netze er angeschlossen ist. Daher ist es möglich, einer Routing Engine ein Netz hinzuzufügen, das gar nicht erreichbar ist. Dies ist auch bei heutigen BGP Installationen der Fall.

5.5.3 Explizites Routing

Beim expliziten Routing geht es darum, unterschiedliche Daten anders zu behandeln. Diese Daten können anhand von dem Application Layer unterschieden werden. So ist es möglich Voice Daten von Web Daten zu trennen. Dies wird nun anhand von einem Beispiel genauer erklärt.

In einem grösseren Netzwerk spielt die Ausfallsicherheit eine grosse Rolle. Dazu muss Redundanz geschaffen werden, indem jeder Link doppelt geführt wird. Dies gilt natürlich auch für den Link ins Internet. Dazu werden zwei Leitungen zu je einem Provider gemietet. Dabei muss beachtet werden, dass diese beiden Provider nicht denselben Upstream Provider verwenden. Weil wenn dieser einen Netzausfall verursacht beide Provider auch betroffen sind. Weiter muss auch darauf geachtet werden, dass die beiden Links nicht im selben Trasse geführt werden. Dies nennt sich Trasse Redundanz und soll verhindern, dass eine Unachtsamkeit auf einer Baustelle oder ein Unwetter, nicht beide Links zerstören kann.

Ein Problem, das nun auftaucht, ist das nur ein Link voll ausgelastet werden darf. Weil wenn der primäre Datenpfad unterbrochen wird, muss der sekundäre die ganze Last transportieren können. Das bedeutet jedoch, dass ein Link bezahlt wird, obwohl er nicht benutzt wird.

Sinnvoller scheint es, wenn beide Links benutzt werden, also mehr als 50 Prozent der maximalen Bandbreite. Beim Erstellen eines Backups auf ein entferntes System oder beim Verschieben einer virtuellen Maschine kann dies sehr nützlich sein.

Was geschieht nun wenn mehr als 50 Prozent der Bandbreite benutzt wird und ein Link ausfällt? Wie kann nun entschieden werden, welche Daten weniger wichtig sind und verworfen werden? Als erstes müssen nun die Daten klassifiziert werden. Es muss unterschieden werden, welche Daten Voice enthalten, welche Daten für Backups sind oder durch das Surfen im Internet entstehen. Dazu muss ein Switch eine Layer 7 Inspection durchführen. Das heisst, der Application Layer muss untersucht werden. Weiter muss noch unterschieden werden zwischen Daten, die im internen Netz bleiben und Daten, die in ein externes Netz weitergeleitet werden. Da interne Daten durch den Linkausfall zum Provider nicht betroffen sind, muss auch keine andere Policy darauf angewendet werden. Sie werden

wie bisher behandelt. Nur auf Daten, die das interne Netz verlassen, muss eine andere Policy angewendet werden. Mit dieser Policy kann nun entschieden werden, welche Daten Business critical sind und welche nicht. Als Business critical werden Daten bezeichnet, die das Unternehmen unbedingt benötigt. Das können zum Beispiel Voice Daten sein, oder Zugriffe von Kunden auf die Webserver. Diese kritischen Daten werden nun mit einer höheren Quality of Service (QoS) versehen. Sie werden also bevorzugt über den verbleibenden Link gesendet. Ist dieser ausgelastet, werden die weniger wichtigen Daten verworfen. Der Betrieb kann somit aufrechterhalten werden jedoch stehen nicht mehr alle Funktionen zur Verfügung. Als Beispiel kann der Datenverkehr zu YouTube oder Facebook eingeschränkt sein.

Stellt sich nun jedoch die Frage, wo diese Policy angewendet wird. Eine Möglichkeit ist, auf den Border Routern. Dabei kann es vorkommen, dass die Eingangswarteschlange voll wird. Alle Daten, die danach beim Router ankommen, werden verworfen (Taildrop). Dies gilt auch für Daten mit einem hohen QoS, weil ein Paket zuerst eingelesen werden muss, bevor der QoS bestimmt werden kann. Mit den heutigen Mitteln, wie Performance Routing und Policy Routing ist dieses Problem nicht in den Griff zu bekommen.

In einem SDN Netz kann nun aber diese Policy bis zu den Access Layer Switch angewendet werden. So ist es möglich, weniger kritische Daten schon beim Client zu blockieren. Die Border Router werden dadurch entlastet. Eine solche Policy kann natürlich auch sehr flexibel angewendet werden. Als erstes wird die Last auf dem Link zum Provider dauernd überwacht. Erst wenn die maximale Last beinahe erreicht ist, wird die Policy auf den Access Layer Switch angepasst. Das Anpassen kann stufenweise durchgeführt werden. Zuerst wird zum Beispiel der gesamte Datenverkehr, der nicht so wichtig ist, blockiert. Hat sich nun die Last auf dem Provider Link stabilisiert, kann der Webtraffic zu einem Content Filter geleitet werden. Dort werden dann bestimmte Kategorien, die im Geschäftsalltag, nicht benötigt werden, blockiert. So wird die Policy schrittweise gelockert oder wieder verschärft.

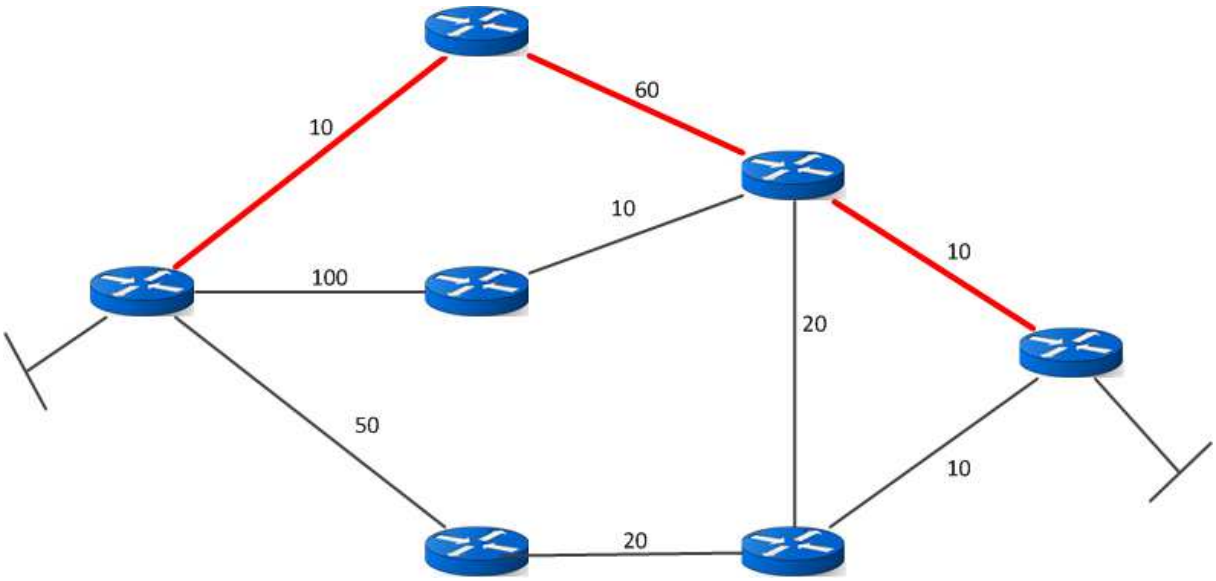
Als Letztes muss noch besprochen werden, wie eine solche Policy in einem SDN Netz angewendet werden kann. Ein Flowtable Eintrag leitet zum Beispiel jeglichen Webtraffig zu einem Router weiter. Dieser schickt die Daten dann ins Internet. So muss nicht für jeden einzelnen Client ein eigener Flowtable Eintrag erstellt werden. Sollen nun einzelne Webseiten gesperrt werden, wird einfach ein neuer Flowtable Eintrag generiert. Dieser leitet den Webtraffig nicht mehr zum Border Router sondern zu einem Content Filter.

Mit SDN kann also auch mit dem Uplink Geld gespart werden. Es können zwei Links gemietet werden mit kleineren Bandbreiten, die somit günstiger sind.

Das zweite Beispiel soll aufzeigen, wie ein Service Provider, mit Traffic Engineering, sein Netzwerk optimieren kann.

Bei der Optimierung geht es darum, bestehende Bandbreite besser ausnutzen zu können. Ein Service Provider möchte möglichst viele Kunden mit demselben Link bedienen. Mit einem Routingprotokoll kann normalerweise jedoch nicht bestimmt werden, welche Daten wie transportiert werden. Im

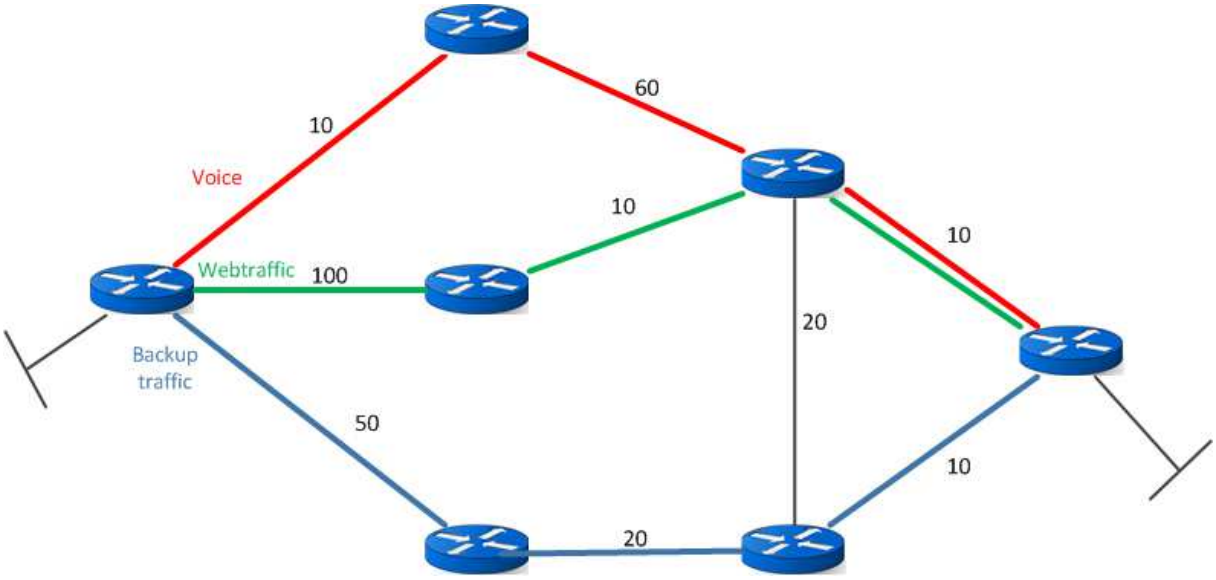
nachstehenden Bild wird ein Providernetz aufgezeigt. Dabei sieht man, dass immer dieselben Links belastet werden.



ISP Netz ohne Optimierung

Die rote Linie zeigt den Shortest Path, der von einem Routingprotokoll berechnet wurde. Es gibt mehrere mögliche Wege durch das Netz, jedoch wird nur einer verwendet. So kann nun natürlich nicht die maximale Bandbreite ausgenutzt werden. Es muss eine Möglichkeit gefunden werden, wie bestimmte Daten auf einen anderen Link umgeleitet werden können. Eine Variante, die hier besprochen wird, ist das Trafficengineering von MPLS. Dabei werden die Labels nicht per Label Distribution Protocol (LDP) ausgetauscht, sondern über das Ressource Reservation Protocol (RSVP). Als erstes wird nun die benötigte Dienstgüte reserviert. Das kann die Bandbreite, der Delay oder auch der Jitter sein. Wenn das Netz den geforderten Dienst zur Verfügung stellen kann, wird ein MPLS Label ausgetauscht, und so einen Weg signalisiert. Es kann nun auch entschieden werden, dass ein bestimmter Link nicht verwendet werden darf. Ein Service Provider kann nun einzelne Trafficklassen unterschiedlich behandelt. Voiceverkehr wird dann zum Beispiel nie über einen Satelitenlink geschickt.

Im nachstehenden Bild ist das gleiche Netz nochmals aufgezeigt, diesmal jedoch mit Trafficengineering.



ISP Netz mit Optimierung

Auf dem roten Link werden nun Sprachdaten übertragen, weil hier der Delay klein ist. Webtraffic wiederum kann über eine langsame Verbindung gesendet werden, weil diese Daten nicht zeitkritisch sind. Für einen Link, der benutzt wird, um ein Backup in ein anderes Datacenter zu spielen, ist die Bandbreite die kritische Grösse. Auf dem blauen Link kann nun die benötigte Bandbreite reserviert werden.

Mit Software Defined Network kann ein Service Provider das gesamte Netz als eine Einheit verwalten. Als Transportprotokoll kann immer noch MPLS verwendet werden. Die Labels werden nun jedoch durch den SDN Controller ausgetauscht. Zuerst werden die Switch, über NetFlow oder einem ähnlichen Protokoll abgefragt, ob die benötigten Ressourcen zur Verfügung stehen. Stehen die Ressourcen zur Verfügung, werden die MPLS Labels ausgetauscht und so einen logischen Kanal durch das Netz signalisiert. Der SDN Controller kann nun auch sehr flexibel auf Änderungen reagieren. Es kann zum Beispiel ein Flow, auf einen Link der die Anforderungen auch erfüllt, umgeleitet werden, damit wieder Ressourcen frei sind für anderen Traffic. So können die einzelnen Links optimaler ausgelastet werden. Ein weiterer Vorteil von SDN ist, dass in einem Datacenter die Daten von den einzelnen virtuellen Maschinen durch VLANs und IP Adressen voneinander getrennt werden können. Bevor nun die Daten auf einen WAN Link geleitet werden, wird anhand vom VLAN Tag und der Source IP Adresse ein MPLS Label an die Pakete angehängt. So wird nun der Traffic von den einzelnen virtuellen Maschinen durch MPLS Labels voneinander getrennt. So können virtuelle Dienste in einem Netzwerk zur Verfügung gestellt werden.

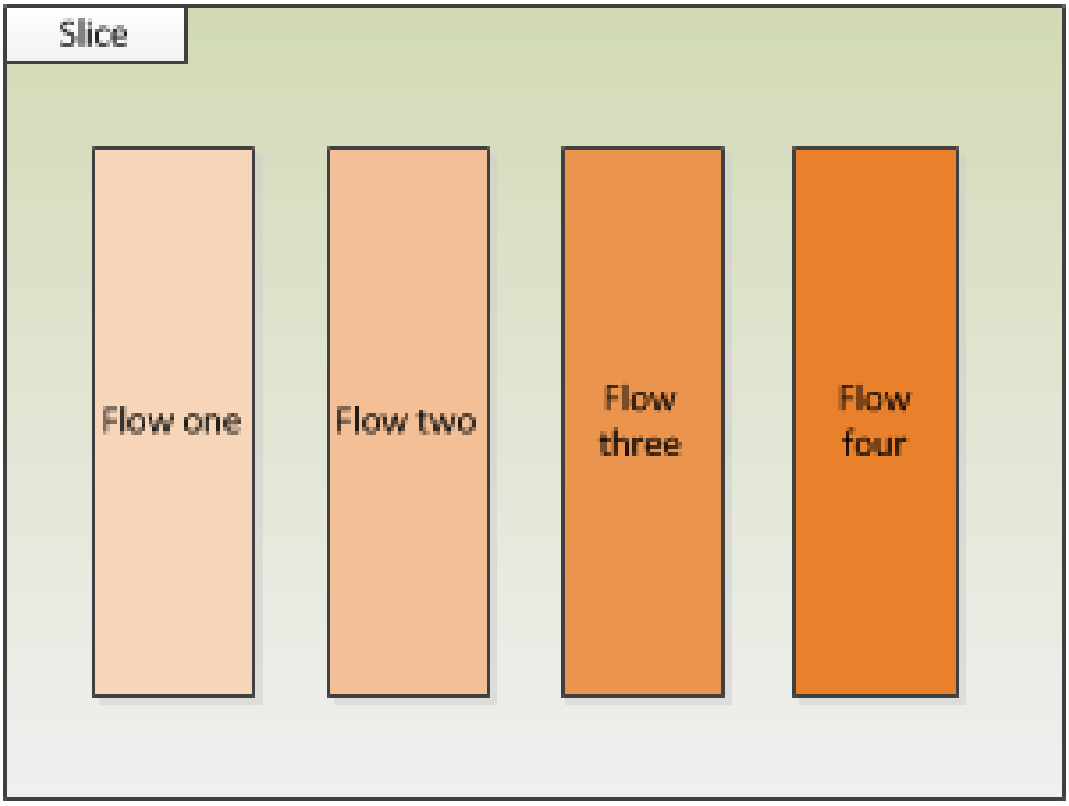
5.5.4 Virtualisierte Netzwerke

Ein Service Provider hat normalerweise zwei Netzwerke, die er betreuen muss. Ein Access Netzwerk und ein Backbone Netzwerk. Das Access Netzwerk holt die Daten bei den Kunden ab, und leitet sie an das Backbone Netz weiter. Dies können Daten von ADSL, VDSL, FTTH, oder was der Provider sonst noch für Dienste anbietet, sein. Das Backbone Netz leitet diese Daten dann weiter, zum Beispiel an einen Transit Provider. Das Problem, dass nun ein Service Provider hat, sind die Cost of Ownership. Das heisst, die Kosten, die entstehen durch das Betreiben eines Netzwerks. Die Kosten setzen sich zu etwa 30% Anschaffungskosten und etwa 70% Unterhaltskosten zusammen. Unter Anschaffungskosten sind die Kosten gemeint, die entstehen, wenn neue Hardware gekauft wird, um ein Netz auszubauen oder alte Hardware zu ersetzen. Dies sind einmalige Beträge und so einfach vorauszuberechnen. Die höchsten Kosten entstehen jedoch durch den Unterhalt der Hardware. Es müssen Logfiles ausgewertet werden, was in einem so grossen Netz mit grossem Aufwand verbunden ist. Weiter müssen auch Konfigurationsänderunen eingespielt werden. Dazu ist meist ein Change Request nötig. Diese Anfrage beinhaltet alle Konfigurationen und auch ein Rollback. Das heisst wenn die Konfiguration nicht so funktioniert, wie erwartet, muss sie rückgängig gemacht werden. All das muss vom Change Manager überprüft und freigegeben werden. Erst nach Bestätigung können die Änderungen im produktiven Netzwerk ausgeführt werden. Da bei diesem Prozess mehrere Personen beteiligt sind, schiessen auch die Kosten in die Höhe.

Es müssen noch weitere Überwachungsaufgaben wahrgenommen werden. Ein Administrator muss zum Beispiel auch die Auslastungen von den einzelnen Links messen und beobachtet. Nur so kann beurteilt werden, wo ein Ausbau der Kapazität nötig ist und wo nicht. Weiter können so auch Ausfälle von Hardware besser vorausgesagt werden. Wenn immer mehr Bitfehler auf einem Interface auftreten, ist dies ein Indiz, dass dieses Interface bald den Dienst aufgeben könnte. Weiter kann auch die Temperatur der Hardware mittels einer Wärmebildkamera überwacht werden. Ist die Temperatur an einer Stelle ungewöhnlich hoch, ist auch das ein Anzeichen auf einen bevorstehenden Defekt.

All diese Massnahmen treiben die Kosten pro Bit (Cost/Bit) in die Höhe. Ein Service Provider möchte jedoch diese Kosten möglichst tief halten. Mit Software Defined Network könnte dies erreicht werden. Es besteht die Möglichkeit, ein physikalisches Netz in virtuelle Netze, sogenannte Slices, zu unterteilen. Damit ist es möglich, für jeden grösseren Kunden, ein eigenes Netz zur Verfügung zu stellen.

Im untenstehenden Bild sieht man beispielhaft dargestellt, verschiedene Flows, die auf einem Switch definiert wurden. Diese vier Flows wurden zu einem Slice zusammengefasst. So muss ein Service Provider nicht für jeden Kunden hunderte von Flows erstellen. Es reicht, wenn er einen oder mehrere Slices erstellt. Die Slices werden so abstrakt wie möglich definiert. So können sie wiederverwendet werden. Zum Beispiel ist es sinnvoll, einen Slice zu definieren, der den gesamten Webtraffic behandelt. Dieser Slice besteht dann aus zwei Flows. Der eine für HTTP, der anderen für HTTPS. Dieser Slice kann nun für verschiedene Kunden verwendet werden. Einen konkreten Slice könnte man nicht wiederverwenden. Würde man zum Beispiel die Flows noch mit einer IP Adresse ergänzen, könnte man den Slice nur für einen einzigen Kunden verwenden.

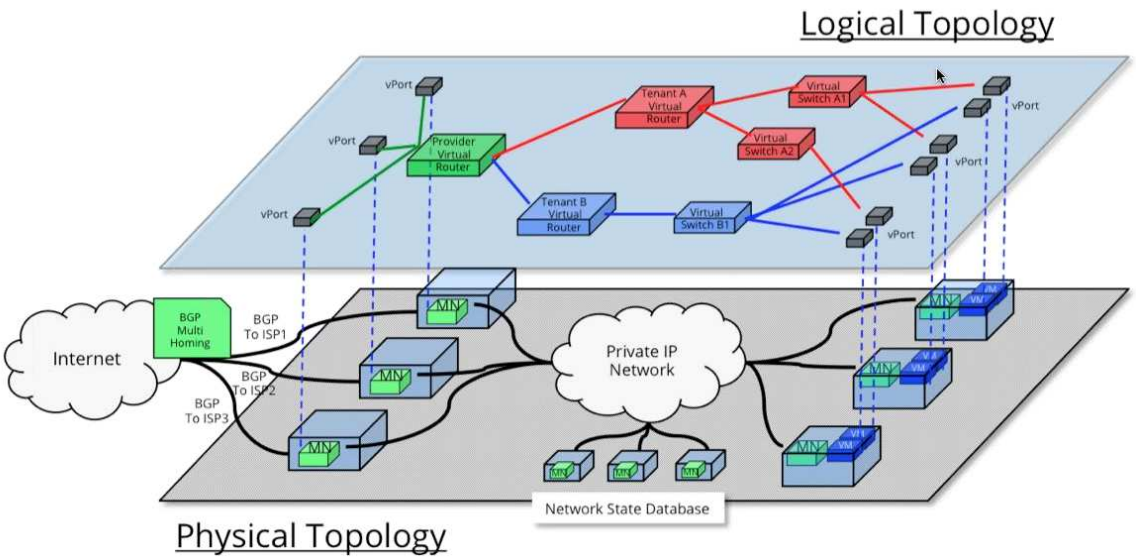


Slice mit unterschiedlichen Flows.

In heutigen Netzwerken kann diese Unterteilung mit Virtual Routing and Forwarding Instanzen (VRF) und Multiprotocol Label Switching (MPLS) Labels erreicht werden. Hier müssen dann jedoch zwei Technologien angewendet werden. Weiter müssen die Konfigurationen natürlich auf jedem einzelnen Gerät angepasst werden

Mit SDN kann nun ein Slice auf dem Controller definiert und im Netzwerk verteilt werden. Ein SDN Switch überprüft nun als erstes, zu welchem Slice ein Packet gehört. Dies kann er in der Hardware erledigen. So muss die CPU nicht belastet werden und viel höhere Performance ist möglich. Nun muss nur noch die Flowtable von diesem einen Slice überprüft werden. In dieser Flowtable können nun wieder Flows definiert werden, die auf den Source- oder Destination Port, IP Adressen oder auf das Protokoll reagieren.

In der folgenden Grafik wird ein Overlay Network dargestellt. In der unteren Hälfte ist die physikalische Sicht aufgezeichnet. Dies ist das reale Netzwerk. Darüber ist die logische Sicht dargestellt. Mit dem roten Slice wird ein Kunde bedient und ein Anderer wird mit dem blauen Slice bedient. Von den einzelnen Slices werden die gleichen physikalischen Geräte verwendet. Nur die Daten sind voneinander getrennt.



Overlay Network, wie es in einem Service Provider Netz angewendet werden kann.

So kann auch dynamisch auf einen Kundenwunsch reagiert werden. Möchte ein Kunde eine neue Niederlassung eröffnen, wird vom Provider einfach ein neuer Slice erstellt, oder ein bestehender erweitert. So kann sehr schnell reagiert werden. Dies ist auch mit heutigen Technologien, wie Frame Relay, möglich. Dies wird im entsprechenden Kapitel genauer behandelt.

5.5.5 Netzwerkzugriffs Kontrolle

Heute kann die Zugriffskontrolle in einem Netzwerk mit einem Network Access Controll Server durchgeführt werden. Verbindet sich nun ein Client mit dem Netzwerk, muss er sich zuerst authentisieren. Dies geschieht meistens über IEEE 802.1x. Dabei fragt der Authenticator (dies kann ein Switch sein) den Client nach Usernamen und Passwort. Sobald der User seine Daten eingegeben hat, werden diese an den Authentication Server weitergeleitet. Dieser überprüft die Zugriffsberechtigung vom User und meldet dem Authenticator, ob der Zugriff erlaubt ist oder nicht. Der Authentication Server kann auch eine VLAN ID mitschicken. So kann der Client in ein anderes Netz weitergeleitet werden. Weiter kann auch überprüft werden, ob der Client die neuste Virendatenbank und die Betriebsystemupdates eingespielt hat. Erst nach erfolgreicher Überprüfung wird der Client in das interne Netzwerk verschoben.

In einem Software Defined Network muss es natürlich auch möglich sein, einen Client zu authentisieren und den Systemstatus zu überprüfen. Beim Verbinden von einem Client mit einem Switch wird eine Meldung vom Switch zum SDN Controller geschickt. Die Meldung besagt dass sich der Status eines Ports verändert hat. Der Controller leitet diese Meldung an den Orchestrar weiter. Dieser kann nun einen Flow auf den Switch schicken, der nur EAP over Ethernet erlaubt. Dieses Protokoll wird verwendet, um eine Authentisierung von einem Client durchzuführen. Der SDN Switch fragt nun den User nach dem Usernamen und dem Passwort. Dieser Prozess ist gleich wie in einem heutigen Netzwerk. Nachdem der User seine Daten eingegeben hat, schickt sie der SDN Switch zum Orchestrar. Dieser fragt nun beim Authentication Server nach, ob der User Zugriff erhalten soll. Diese

Kommunikation kann zum Beispiel über Radius stattfinden. Wenn der Zugriff erlaubt ist, kann der Orchestrar einen Slice generieren, der dem Client die nötigen Zugriffsberechtigungen gibt.

Der Authentication Server kann natürlich auch gleich in den Orchestrar integriert werden. Dies hat den Vorteil, dass kein externer Radius Server betrieben werden muss. Der Orchestrar muss jedoch die Möglichkeit bieten, den User extern zu authentisieren. Ansonsten müsste eine Firma die gesamte User Authentifizierung umstellen, nur weil das Netzwerk auf SDN umgestellt wird.

Die Authentisierung von einem Client vereinfacht sich also nicht mit SDN. Die Zugriffsberechtigung kann pro User vergeben werden und ist so sehr flexibel einsetzbar. Dies ist in heutigen Netzen auch schon möglich. Auch hier werden nicht grosse Vorteile entstehen durch SDN.

5.5.6 Überwachung

In heutigen Netzwerken ist das Monitoring eine Herausforderung. Man kann zwar Informationen von Netzwerkgeräten mit SNMP empfangen, nur muss auf allen Geräten ein Username und Passwort gesetzt werden. Dies ist wieder ein grosser Administrationsaufwand. Wird sogar nur SNMPv2 eingesetzt, sind die Informationen nicht verschlüsselt und ein Angreifer kann auch auf die Netzwerkgeräte zugreifen. Dies ist natürlich eine riesige Sicherheitslücke. Eine andere Möglichkeit ist, dass auf einem Switch ein sogenannter Mirrorport konfiguriert wird. Dabei wird der gesamte Verkehr von einem Sourceport auf den Mirrorport gespiegelt. An diesem Mirrorport befindet sich dann das Monitoring Tool. Diese Möglichkeit ist nicht flexibel, weil für jedes Gerät, das überwacht werden soll, eine Monitoring Session konfiguriert werden muss. Weiter kann das Monitoring System schnell an seine Grenzen stossen, wenn viele Daten empfangen werden sollen.

Weiter gibt es Monitoring Systeme, die sich per SSH auf die entsprechenden Geräte einloggen und so Informationen auslesen können. So sind die Informationen geschützt und nicht von Dritten auslesbar. Auch hier werden wieder ein Username und ein Passwort benötigt auf allen Netzwerkgeräten. SSH bietet noch eine weitere Möglichkeit, um den Zugriff zu schützen. Es kann, auf dem Monitoring Tool, ein Schlüsselpaar generiert werden. Den Public Key spielt man danach auf alle Netzwerkgeräte. Nun kann sich das Monitoring Tool auf die Netzwerkgeräte einloggen und sich mit dem Private Key authentisieren.

Ein anderer Ansatz ist, die Verfügbarkeit von Geräten mittels Pings zu überprüfen. Dabei kann aber nicht getestet werden, ob auch alle Dienste auf dem Gerät funktionieren. Es wird nur überprüft, ob das Gerät noch läuft und ob die Netzwerkverbindung noch steht.

Alle diese Systeme haben eine gemeinsame Schwäche. Es können nur Informationen von einzelnen Geräten beschaffen. Es ist jedoch nicht möglich, die Informationen in Relation zu einander zu bringen. Es kann nur anhand von Zeitangaben, die Ereignisse aufbereitet werden.

In einem SDN Netzwerk kann nun jedoch ein gesamter Flow überprüft werden. Dabei kann festgestellt werden, wie viele Daten pro Flow übertragen werden. So können Statistiken über den End-to-End Datenverkehr erstellt werden. Es gibt sogar die Möglichkeit, die Daten anhand des Applikationslayers zu unterscheiden. Dies ist hilfreich, da so eine Auswertung erstellt werden kann, wie viele Daten zu Facebook oder zu YouTube übertragen wird. Anhand solcher Statistiken kann das

Surfverhalten der Benutzer erfasst werden und gegebenenfalls die Nutzungspolicy angepasst werden.

In einem SDN Netzwerk müssen natürlich auch die einzelnen Geräte überwacht werden. Wichtige Werte sind dabei die Auslastung des Prozessors, der verfügbare Speicher oder die Temperatur des Gerätes. Mit OpenFlow kann dies nicht realisiert werden, weil das Protokoll nicht dafür gedacht ist. In einem SDN Netz muss weiterhin SNMP oder als Alternative Netconf eingesetzt werden. Dabei entstehen die gleichen Probleme, wie in einem heutigen Netzwerk. Alle Konfigurationen müssen auf jedem Gerät einzeln gemacht werden. Das Überwachen von einzelnen Geräten wird also nicht einfacher mit SDN.

5.5.7 Priorisierung von Daten

In der heutigen Datenübertragung sind nicht alle Daten gleich wichtig. Sprachpakete müssen zum Beispiel schneller an ihr Ziel übertragen werden, damit ein Gespräch flüssig empfangen wird. Solche wichtigen Daten bekommen eine höhere Quality of Service (QoS). So kann ein Gerät, egal ob Router oder Switch, solche Pakete bevorzugt behandelt. Technisch wird das so gelöst, dass mehrere Verarbeitung Queues implementiert sind. Jede Queue hat eine andere Wichtigkeit. Je nach Queuing Theorie wird nun zuerst die Queue mit der höchsten Priorität abgearbeitet und erst danach die anderen. Dies kann jedoch dazu führen, dass die Queues mit niedriger Priorität nicht abgearbeitet werden. Sie „verhungern“ also. Es kann auch eine Round Robin Strategie angewendet werden, bei der jede Queue nacheinander abgearbeitet wird. Von der Queue mit der höchsten Priorität werden jedoch mehr Pakete abgearbeitet als von den anderen. Bei dieser Strategie kann es nicht vorkommen, dass eine Queue „verhungert“.

Beim Quality of Service ist das Problem auch wieder, dass jedes Gerät konfiguriert werden muss. Wenn nur ein Gerät QoS nicht implementiert, kann das die gesamte QoS Policy zu nichte machen. Ein weiteres Problem besteht darin, dass das Netzwerk nicht immer den QoS Wert bestimmen kann. Dies ist zum einen dann der Fall, wenn das Netzwerk die Applikation nicht erkennt oder wenn die Kommunikation verschlüsselt ist. Dieses Problem kann gelöst werden, indem die Applikation selbstständig den QoS Wert setzt. Das Netzwerk kann nun einfach auf die DSCP Werte im IP Paket reagieren. Nun taucht jedoch ein neues Problem auf. Die „lügenden Endgeräte“. Eine Applikation könnte sich einfach eine höhere Priorität vergeben, um die eigenen Daten schneller ins Netz schicken zu können. Die sicherste Möglichkeit ist, dass das Netzwerk die Applikation erkennt. Ein Switch muss ein gutes Application Fingerprinting besitzen. So kann die Applikation möglichst nahe beim Client bestimmt werden. Dies ist jedoch erst in den letzten Jahren möglich.

Mit OpenFlow 1.3 sind die sogenannten Meters in den Standard aufgenommen worden. Ein Meter besteht aus einem oder mehreren Meter Bands. Mit einem Meter Band wird ein Schwellwert definiert, der den Meter auslöst. Als Beispiel kann ein Schwellwert definiert werden, um einen Broadcaststorm zu verhindern. Das Gleiche wird auch noch für den Multicastverkehr benötigt. Hier können nun zwei Meter Bands definiert werden, die im Ereignisfall denselben Meter auslösen. Wird der Meter ausgelöst, können alle Pakete, die höher als der Schwellwert sind, verworfen werden.

Ein Meter hat jedoch auch noch eine andere Möglichkeit, um Pakete zu behandeln. Es können die DSCP Werte angepasst werden und somit den Quality of Service beeinflusst werden. Alle Pakete von einem bestimmten Flow werden nun mit einem Meter bearbeitet. So können alle Daten von einer Applikation einer QoS Klasse zugeordnet werden.

Mit den Meters hat OpenFlow also eine gute Möglichkeit um einen Ratelimitter zu erstellen oder auch eine QoS Policy in einem Netzwerk anzuwenden. Ein Meter kann auch vom Controller dynamisch erstellt werden. Wenn zum Beispiel ein Link ausfällt und so weniger Bandbreite zur Verfügung steht, kann eine QoS Policy im Netzwerk verteilt werden um den kritischen Traffic zu schützen. Dies wird auch schon im Kapitel „Explizites Routing“ behandelt.

5.5.8 Sicherheit

Bei dem Thema Sicherheit geht es darum, sicherzustellen, dass die IT Infrastruktur nicht für Etwas verwendet wird, für das sie nicht gedacht ist. Diese, etwas kompliziertere, Definition bringt es jedoch ziemlich genau auf den Punkt. Es muss verhindert werden, dass unerlaubte Daten im Netzwerk transportiert werden. Ob diese Daten böswillig oder versehentlich erzeugt wurden, ist nicht relevant. Weiter müssen auch unerlaubte Zugriffe auf Systeme unterbunden werden.

In einem heutigen Netzwerk versucht man mehrere Schichten von Sicherheitssystemen hintereinander zu schalten. Für die Verteidigung gegen Angriffe aus dem Internet wird meistens ein Router installiert als erste Abwehrschicht. Dieser soll eine erste Filterung von den Daten vornehmen und Denial-of-Service Angriffe abwehren. Der Vorteil von einem Router als erste Abwehr ist, dass eine Inspektion von einem Paket nicht so viele Ressourcen benötigt wie bei einer Firewall. So kann ein Router einfacher einen DoS Angriff abwehren, ohne selbst Opfer davon zu werden. Wenn durch einen DoS Angriff sehr viele Daten generiert werden und den Uplink drohen zu überfüllen, kann auch das Opfersystem aus dem Routing genommen werden. Das bedeutet, dass die Routen zum Opfersystem nicht mehr an den Provider annaunced werden. So ist zwar das System im Internet auch nicht mehr erreichbar, jedoch kann der Traffic vom Uplink ferngehalten werden.

Die nächste Schicht wird durch eine Firewall gebildet. Sie ist für die feine Filterung von den Daten zuständig. Dabei kann auch in den Applicationslayer geschaut werden um zu überprüfen, ob der Header dem Standard entspricht. Eine Firewall kann auch weitere Angriffe abwehren, die durch einen Router nicht entdeckt werden. Hier soll der SYN-Flooding Angriff erwähnt werden. Dabei schickt ein Angreifer TCP Pakete, die das SYN Flag gesetzt haben. Das Opfer muss nun Speicher reservieren um auf eine eingehende Verbindung zu warten. Der Angreifer erzeugt nun laufend neue, halboffene Verbindungen, bis das Opfersystem kein Speicher mehr zur Verfügung hat und den Dienst nicht mehr erbringen kann. Eine Firewall lässt nun nur eine bestimmte Zahl von halboffenen Verbindungen zu. So können die Server geschützt werden.

Um die Sicherheit weiter zu erhöhen kann ein Intrusion Detection System oder ein Intrusion Prevention System eingesetzt werden. Diese erkennen einen Angriff anhand von Signaturen und können zum Beispiel eine Firewall um konfigurieren. Wie bei einem Virenscanner müssen die Datenbanken auch stets auf dem aktuellen Stand gehalten werden.

Natürlich müssen nicht nur Angriffe aus dem Internet abgewehrt werden. Die meisten Angriffe erfolgen von innerhalb des Netzwerkes. Die Meisten sind harmlos, weil ein Mitarbeiter einfach versucht ein Sicherheitssystem zu umgehen. Diese Angriffe sind technisch nicht so anspruchsvoll und stellen so keine grosse Gefahr dar. Trotzdem müssen auch solche Gefahren gebannt werden. Dazu werden auch innerhalb vom Netzwerk Firewalls installiert.

Ein grosser Nachteil in herkömmlichen Netzen ist nun jedoch die Tatsache, dass Pakete nur gefiltert werden können, wenn sie durch eine Firewall fliessen. Jeglicher anderer Verkehr wird nicht überprüft. Mit Software Defined Network ist es nun möglich eine Security Policy im ganzen Netz zu verteilen. Dies wird mit Hilfe von Flows erreicht, die bestimmten Verkehr zulassen und den Rest blockieren. So wird auch gleich die Strategie von der kleinsten Berechtigung umgesetzt. Diese Flows werden nun natürlich auf jedem Switch definiert und die Security Policy wird auf alle Pakete angewendet. So ist es nicht mehr möglich, die Security Policy zu untergraben. Ein Switch darf nun natürlich nicht eigenen Entscheidungen treffen. Es dürfen keine Pakete anhand von der MAC Adresse weitergeleitet werden. Das sogenannte Ethernetswitching muss also verhindert werden ansonsten wird die Policy untergraben. Für jeden neuen Flow muss der Controller angefragt werden. Dieser entscheidet nun was mit dem Flow geschehen soll. Er kann erlaubt, verworfen oder angepasst werden. Als Beispiel für das Anpassen von einem Flow kann Webtraffic erwähnt werden. Ein Client fragt einen Flow an, um auf einen Webserver im Internet zuzugreifen. Der Controller erlaubt dies zwar, leitet den Verkehr jedoch über einen Content Filter weiter.

Beim Übergang ins Internet werden immer noch die Gleichen Sicherheitsmassnahmen getroffen wie in heutigen Netzwerken. Es wird also immer noch ein Router, eine Firewall und ein Intrusion Detection System benötigt, um die Angriffe aus dem Internet abzuwehren.

Durch das Verteilen der Security Policy ins ganze Netzwerk kann die Sicherheit erheblich erhöht werden. Auch wird die Policy an einem zentralen Ort verwaltet. Dadurch ist sie einfacher zu warten und zu erweitern. Mit Software Defined Networking kann also die Sicherheit im internen Netz erheblich erhöht werden.

5.6 Beispiele zu SDN

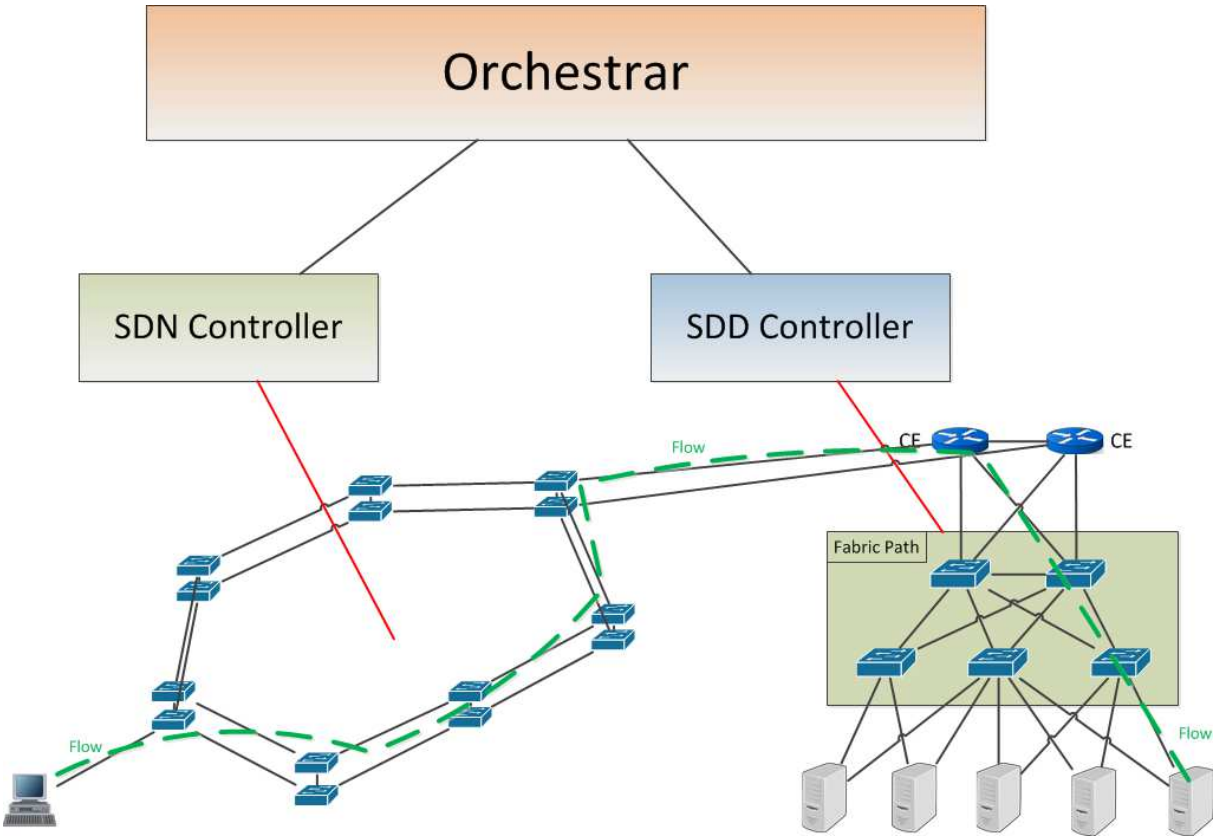
5.6.1 Firma mit zwei Datacenter

Beim ersten Szenario geht es um eine Firma, die zwei Datacenter betreibt. Zwischen den Datacentern wird ein Layer 2 Link benötigt, welcher von einem Service Provider gemietet wird. Im Datacenter werden virtuelle Maschinen betrieben, die einen Service für die Kunden zur Verfügung stellen. Der Vorteil davon ist, dass eine virtuelle Maschine (virtueller Server) einfach an einem anderen Ort abgespeichert werden kann und somit vor einem Verlust geschützt wird (Backup). Weiter kann auch eine virtuelle Maschine verschoben werden. Das bedeutet, von einem physikalischen Server auf einen anderen. Eine solche Verschiebung wird durch den Software Defined Datacenter Controller übernommen. Die virtuelle Maschine muss, nach der Verschiebung, die gleiche IP Adresse behalten, da sonst die Kunden nicht mehr darauf zugreifen können. Die zwei Datacenter können zum Beispiel über einen Layer 2 Link untereinander verbunden werden. Dann befinden sie sich jedoch in derselben Broadcastdomain. Bei grossen Datacentern kann dies zu grossen Problemen führen. Befinden sich die Datacenter in unterschiedlichen Netzwerken können nun Techniken wie VXLAN oder OTV eingesetzt werden. Damit können virtuelle Layer 2 Verbindungen über einen Layer 3 Link angeboten werden. Die virtuellen Maschinen können so im selben Subnetz sein, die Broadcast werden jedoch nicht an alle Maschinen im anderen Datacenter geschickt. Ist dies nun ein Fall für Software Defined Network? Der Service Provider setzt in diesem Fall ein Software Defined Network ein, muss jedoch nicht auf das Verschieben der virtuellen Maschine reagieren können. Der Provider stellt nur einen Link zwischen den Datacentern zur Verfügung. Dies kann zum Beispiel mit einem MPLS VPN realisiert werden. Der Service Provider konfiguriert auf dem Orchestrar den Link zwischen den Datacentern und die SDN Controller übernehmen die Konfiguration der Switch. Ein Switch fügt nun ein MPLS Label in ein Paket ein und sendet es durch das Backbone Netz des Providers in das andere Datacenter. Den Traffic, der über diesen Link geschickt wird, interessieren den Provider nicht. Bezahlt der Kunde den Link jedoch pro Megabyte, kann SDN wieder sehr interessant sein. Der Orchestrar misst nun die übertragene Datenmenge und leitet die Auswertung an ein System weiter, das die Abrechnung erstellt.

Dieses Szenario soll zeigen, dass SDN nicht immer ein Datacenter und das Netzwerk verwaltet. Es kann ein Orchestrar eingesetzt werden, der nur das Netzwerk verwaltet. In diesem Szenario verwendet der Service Provider SDN, um sein Netzwerk billiger und flexibler zu gestallten. Da er jedoch kein Datacenter besitzt muss er auch keinen SDD Controller einsetzen.

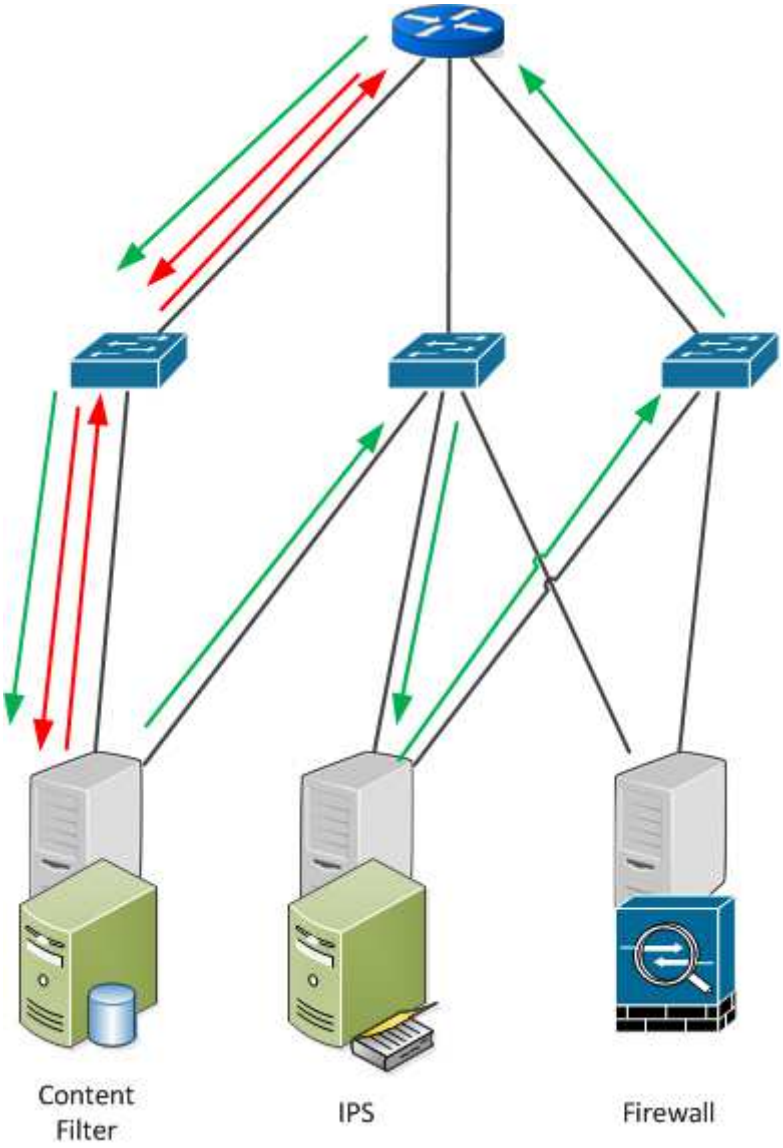
5.6.2 Service Provider

Beim zweiten Beispiel geht es um einen Service Provider, der seinen Kunden die Möglichkeit geben möchte, ihren Service selbst zusammenstellen zu können. Ein Kunde kann sich auf die Website des Providers einloggen und wählen, ob er seinen Internetverkehr durch eine Firewall, IPS oder einen Content Filter leiten möchte. Im untenstehenden Bild ist das Netzwerk des Service Providers abgebildet.



Service Provider Netz mit SDN und SDD

Der gesamte Datenverkehr von dem Kunden wird zu einem Datacenter des Providers geleitet (grüne Linie). Dort werden die Daten durch eine transparente Firewall geleitet und gefiltert. So ist das Netzwerk des Kunden vor Angriffen aus dem Internet geschützt. Nach der Firewall werden die Daten zurück ins Netzwerk des Providers geschickt und über den Coreswitch ins Internet weitergeleitet. Der Rückweg der Daten gelangt wieder über einen Flow ins Datacenter zur virtuellen Maschine und wird dort wieder gefiltert. Im nachfolgenden Bild wird der Fluss der Daten, zwischen den virtuellen Maschinen, etwas genauer angeschaut.



Datenfluss im Datacenter

Die roten Pfeile zeigen den Datenfluss eines Kunden, der einen Content Filter einsetzt. Die Daten werden über den Border Router des Datacenters zu der virtuellen Maschine, die den Content Filter betreibt, geleitet. Der Rückweg wird durch einen zweiten Flow sichergestellt. Dieser leitet die Daten von der virtuellen Maschine wieder zurück zum Border Relay. Die grünen Pfeile zeigen denn Datenfluss an, wenn ein Kunde einen Content Filter, ein Intrusion Prevention System und eine Firewall einsetzt. Es wird wieder ein Flow erzeugt, der die Daten beim Border Relay abholt und an die Virtualisierungsumgebung sendet. Dort werden die Pakete nun durch den Content Filter geschickt. Danach gelangen die Daten zu den anderen virtuellen Maschinen. Ob das Weiterleiten zu den anderen VMs durch das Netzwerk, oder durch die Virtualisierungslösung erledigt wird, spielt dabei keine Rolle.

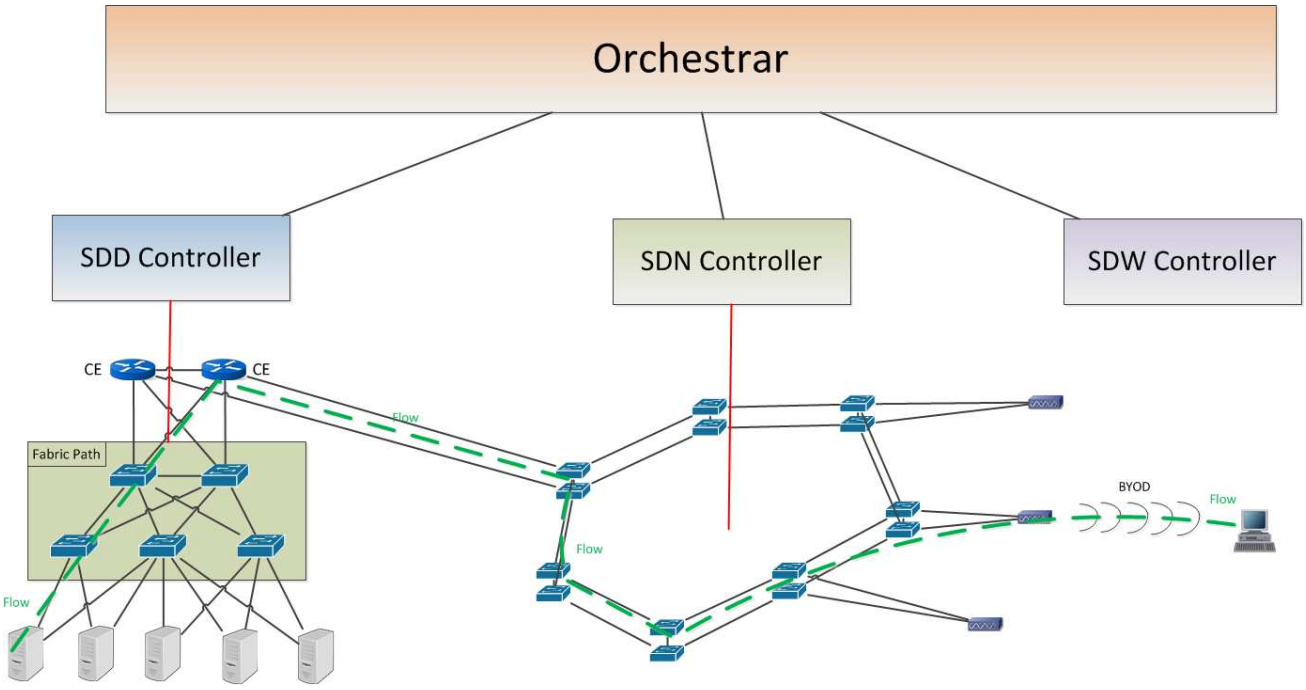
Wird nun ein Datacenter bis an die Belastungsgrenze ausgelastet entscheidet sich der SDD Controller dazu, eine virtuelle Maschine zu verschieben. Dies meldet er nun dem Orchestrar. Dieser teilt dem

Controller mit, welche Maschinen er verschieben kann. So ist es möglich, virtuelle Maschinen zu bündeln und nur ganze Bündel zu verschieben. Werden nicht ganze Bündel verschoben, kann die Situation auftreten, dass der Content Filter von einem Kunden im ersten Datacenter betrieben wird, während sich die Firewall im zweiten Datacenter befindet. Dies würde das Backbonenetzwerk des Providers sehr stark belasten.

Werden nun die VMs verschoben, gibt der Orchestrar dem SDN Controller den Befehl, die Flows der Kunden so anzupassen, dass die Daten in das andere Datacenter umgeleitet werden. Da die angebotenen Services alle transparent sind, spielt es keine Rolle, wenn sich die IP Adresse der VMs ändert. Diese werden nur für das Management verwendet.

5.6.3 Firma mit WLAN Netz

Das dritte Szenario dreht sich um Wireless Lan. Über den Orchestrar erstellt ein Administrator einen neuen Service für das WLAN Netz. Dies könne ein Bring your own device (BYOD) Netzwerk sein. Der Orchestrar meldet nun dem SDD Controller, er solle zwei neue virtuelle Maschinen in Betrieb nehmen und ein Captive Portal darauf starten. Die zweite Meldung geht an den Software Defined WLAN (SDW) Controller. Dieser nimmt nun eine neue SSID in Betrieb mit dem Namen, den der Administrator auf dem Orchestrar konfiguriert hat. Die dritte Nachricht wird an den SDN Controller gesendet. Dieser soll nun die Daten, die über die BYOD SSID gesendet werden, über Flows zu dem Captive Portal weiterleiten. Wenn sich der Benutzer registriert hat, werden seine Daten direkt ins Internet geleitet. Im untenstehenden Bild wird dieses Szenario dargestellt.



SDN und SDW

Die grüne Linie stellt den Flow dar, der die Daten der Clients über das WLAN Netz in das Datacenter leitet.

Wie die letzten drei Szenarien gezeigt haben, können mit Hilfe von einem Orchestrar sehr flexibel Services angeboten werden. Da die Situationen jedoch sehr unterschiedlich sind, müssen auch unterschiedliche Controller angeboten werden. Ein Controller, der alle Szenarien abdeckt würde viel zu teuer werden, da er sehr viel Anfragen beantworten müsste.

5.7 OpenFlow

Nachdem die Anforderungen an SDN und die Use Cases besprochen wurden, wird ein Blick auf die Kommunikation zwischen den SDN Switch und dem SDN Controller geworfen. Ein mögliches Kommunikationsprotokoll ist OpenFlow. Dieses Protokoll transportiert die Nachrichten zwischen den Netzwerkkomponenten und den Controllern. Heute ist OpenFlow das am häufigsten eingesetzte Protokoll in einem Software Defined Network.

OpenFlow hat, im Jahre 2008, als Forschungsprojekt an der Standford Universität begonnen. Erst im Jahre 2011 fasste OpenFlow langsam Fuss in der Wirtschaft. Google hat eigene SDN Switch entwickelt, die OpenFlow fähig sind. Damit ist der Internetriese die erste Firma, die SDN produktiv einsetzt. Google hat zwei grosse Backbone Netzwerke. Das eine ist für den Userdatenverkehr, das andere für den Datacenterverkehr. Das heisst für den Verkehr, der auf den Servern von Google verarbeitet wird. So grosse Netzwerke sind schwierig zu verwalten und auch teuer. Die Idee von Google war nun, die Kosten pro Bit zu reduzieren. Normalerweise steigen diese Kosten an bei grösseren Netzen. Um die Cost/Bit zu senken wollte Google die Administration der Backbones vereinfachen. Dazu muss ein Netzwerk als Gesamtes verwaltet werden nicht mehr pro Gerät.

In einer ersten Phase wurden OpenFlow Switch eingesetzt, die aber konfiguriert waren wie normale Router. Somit hatte SDN noch keinen Einfluss auf den Backbone und die Gefahr für einen Ausfall war gering. Natürlich wurde nicht die gesamte Hardware gemeinsam ersetzt. Zuerst wurde die eine Hälfte der Redundanz ersetz, danach die Zweite. So hatte Google zu jedem Zeitpunkt ein funktionierendes Backbone Netz.

In der zweiten Phase wurden einfache SDN Funktionalitäten aktiviert. Dazu setzt Google das open soucre Produkt RouteFlow ein. RouteFlow ist ein Linux basierter Dienst, der virtuelles IP Routing zur Verfügung stellt. Das heisst, es werden verschiedene virtuelle Routinginstanzen auf einem zentralen Server betrieben. Diese Routinginstanzen beziehen ihre Routinginformationen von den OpenFlow Switch. So kann jeder Switch ein Routingprotokoll verstehen. Ein Switch wird also zu einem virtuellen Router. Der RouteFlow Server wandelt die Routinginformationen in OpenFlow Nachrichten um und sendet diese zu den Switch. So kann ein Switch auf ein Routing Update reagieren. Im Kapitel „RouteFlow“ wird dieser Dienst genauer unter die Lupe genommen.

In der dritten Phase wurde ein gesamtes Datacenter an das SDN Netz übergeben. Jeglicher Datenverkehr wurde über das neue Netz übertragen. Weiter wurde auch eine Priorisierung von

bestimmten Daten, anhand von Layer 7 Informationen, implementiert. Dies geschah Anfang 2012. Google brauchte also nur gut ein Jahr, um den Backbone auf SDN umzustellen.

Nach diesem Ausflug in die Umsetzung von einem SDN Netz, wollen wir das OpenFlow Protokoll ein bisschen genauer anschauen. OpenFlow trennt den Dataplane vom Managementplane. Als Dataplane werden die Daten, die von einem Benutzer gesendet werden, bezeichnet. Das heisst, das sind die produktiven Daten. Auf der anderen Seite werden die Daten, die für die Verwaltung von einem Netzwerk benötigt werden, als Managementplane bezeichnet. Eine Trennung vom Managementplane und vom Dataplane hat zum Vorteil, dass die Konfiguration vom Netzwerk an einer zentralen Stelle erledigt werden kann. Somit ist die Verwaltung viel einfacher und auch billiger. Dieses zentrale System muss natürlich ausfallsicher ausgelegt werden, weil sonst ein ganzes Netz ausfallen kann.

Die kleinste Konfigurationseinheit bei OpenFlow ist ein sogenannter Flow. Ein Flow ist ein Datenstrom von logisch zusammengehörenden Daten. Zum Beispiel ist eine TCP Session ein Flow. Dieser Flow wird anhand von Source- und Destination Port sowie von Source- und Destination IP Adresse identifiziert. Weiter kann ein Flow auch anhand von Source- und Destination MAC Adresse erkannt werden. Ein OpenFlow fähiger Switch besitzt eine Flowtable. In dieser Tabelle wird festgelegt, anhand welcher Merkmale ein Flow erkannt wird. Weiter wird jedem Flow mindestens eine Aktion zugewiesen. So kann der Switch entscheiden, was mit den Paketen geschehen soll.

Flowtable von einem OpenFlow fähigen Switch.

Jeder Eintrag in der Flowtable hat eine Priorität. Dies stellt sicher, dass immer eine eindeutige Entscheidung getroffen werden kann. Dies wird deutlich, wenn eine Flowtable die folgenden zwei Einträge besitzt: Pakete werden anhand vom TCP Quell-Port 2000 identifiziert und einen weiteren Eintrag, der die Pakete anhand vom TCP-Ziel Port 80 identifiziert. So kann keine eindeutige Entscheidung getroffen werden, wenn ein Paket vom TCP Quell Port 2000 zum TCP Ziel Port 80 gesendet wird. In einem solchen Fall ist das Verhalten undefiniert.

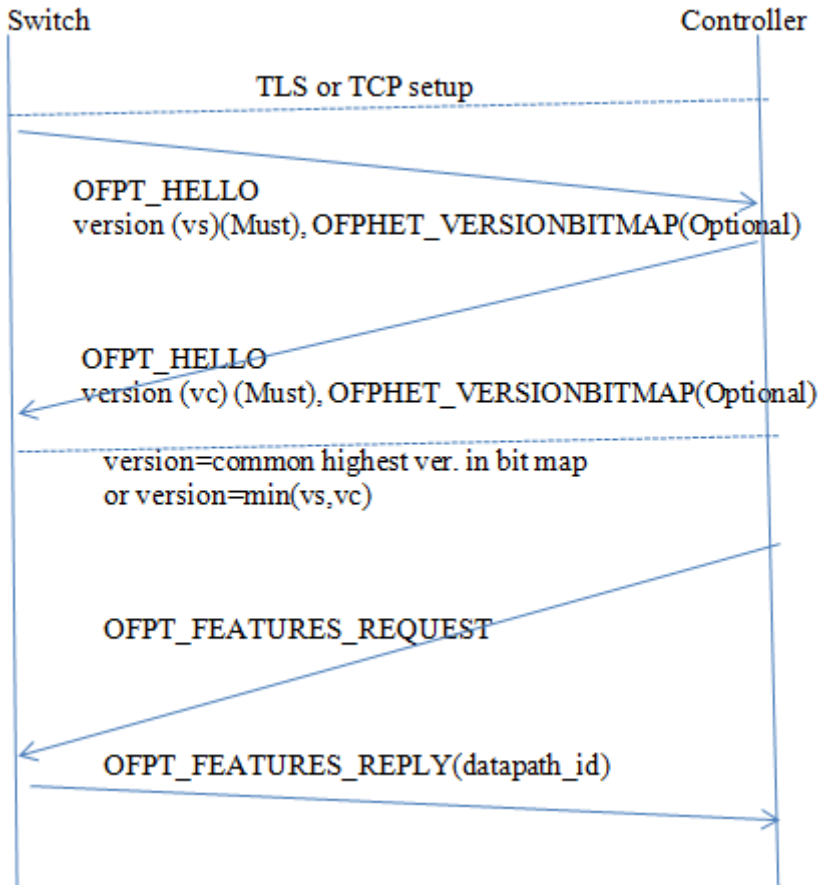
Ein Switch besitzt mehrere solcher Flowtable, welche nacheinander abgearbeitet werden. Wird ein Treffer gefunden, werden alle zugewiesenen Aktionen durchgeführt. Wird kein Treffer gefunden, wird versucht in der nächsten Flowtable einen zu finden. Falls auch in der letzten Tabelle kein Eintrag existiert, wird das Paket an den Controller gesendet. Dieser generiert daraufhin einen neuen Flow und übermittelt ihn per OpenFlow an den Switch. Dieser speichert den Eintrag in der entsprechenden Flowtable. Es existieren mehrere Tabellen, damit ein Hersteller eigene Optimierungen in OpenFlow einbauen kann, ohne seine streng gehüteten Architekturdeteils preis zu geben. Es können zum Beispiel L2- und L3 Lookups in separaten Tabellen realisiert werden. Findet nun ein Hersteller eine schnellere Möglichkeit für den L2 Lookup, wird diese in der entsprechenden Tabelle implementiert.

Nach Aussen sind jedoch die zwei unterschiedlichen Flowtable Lookups nicht sichtbar. So muss sich dieser Herstellen nicht in die Karten schauen lassen.

Ein weiterer Eintrag in der Flowtable zählt die Treffer, die ein Flow erzielt hat und die vergangene Zeit seit dem letzten Treffer. So kann eine Statistik erstellt werden, wie viele Daten pro Flow übertragen wurden.

Für die Kommunikation mit dem Controller, baut jeder OpenFlow fähige Switch eine SSL-gesicherte Verbindung auf. So ist der Managementplane vor unerlaubter Veränderung oder Ausspähung geschützt. Weiter muss jeder Switch einige Basisaktionen unterstützen. Dazu gehören das Weiterleiten von Paketen auf einen bestimmten Port, das Fluten von Daten auf alle Ports und das Verwerfen von einzelnen Paketen. Einige Hersteller implementieren auch noch weitere Aktionen wie das Hinzufügen von VLAN Tags oder MPLS Labels. So können diese Switch sehr flexibel eingesetzt werden.

Beim Verbindungsaufbau mit dem Controller wird als erstes eine TCP oder eine TLS Session aufgebaut. Danach sendet der Switch eine OFPT_HELLO Nachricht an den Controller. Dabei wird die höchste OpenFlow Version (vs), die der Switch unterstütz, übermittelt. In der Versions Bitmap können mehrere, unterstützte Versionen mitgeschickt werden. Der Controller antwortet seinerseits mit einer OFPT_HELLO Nachricht. Nun wird in der Versions Bitmap nach der höchsten OpenFlow Version gesucht, die beide Seiten unterstützen. Als Nächstes sendet der Controller einen Feature Request an den Switch. Dabei fragt der Controller, ob der Switch die verlangten Features unterstützt. Der Switch muss mit einem Feature Reply antworten. Werden alle verlangten Features unterstützt, kann die Verbindung aufgebaut werden. Sollte nun ein Switch die Verbindung zu all seinen Controllern verlieren, wechselt er entweder in den „Fail secure mode“ oder in den „Fail standalone mode“. Im „Fail secure mode“ verwirft der Switch alle Pakete, die an den Controller adressiert sind. Im „Fail standalone mode“ reagiert der SDN Switch wie ein normaler Ethernet Switch. Dieser Modus ist nur auf einem Hybrid Switch verfügbar. Also auf einem Switch, der als Ethernet Switch und als SDN Switch eingesetzt werden kann. Im nachstehenden Bild ist der Ablauf vom Verbindungsaufbau grafisch dargestellt.



5.8 Orchestrar

Bis jetzt wurde immer vom Orchestrar gesprochen und dass alle Einstellungen darauf gemacht werden. Nun stellt sich jedoch die Frage, was ist ein Orchestrar? Kann ein Orchestrar überhaupt alle Anforderungen erfüllen, die an ihn gestellt werden? In diesem Kapitel möchte ich meine Sicht vom Orchestrar auflisten. Da bis heute noch kein Orchestrar auf dem Markt verfügbar ist, stellt dieses Kapitel meine persönliche Ansicht dar.

Der Orchestrar ist das Management System in einem SDN Netzwerk. Er muss alle Aufgaben erfüllen können, die im Kapitel 3.6.1 Netzmanagement und Automation beschrieben wurden. Dazu gehört die Konfiguration von einem Netzwerk beziehungsweise die Konfiguration von Diensten, die im Netzwerk zur Verfügung gestellt werden. Wie könnte nun die Konfiguration von einem Dienst aussehen? Ein Dienst setzt sich aus mehreren Teilen zusammen. Ein Teil beschreibt das Netzwerk, der zweite Teil das Datacenter und der dritte Teil definiert den WLAN Teil. Im Netzwerkteil wird zum Beispiel definiert, welche Daten dem Dienst zugeordnet werden sollen. Dabei spielt die Source und Destination eine Rolle sowie das verwendete Protokoll. Danach wird definiert, was mit den Daten geschehen soll. Werden sie auf eine virtuelle Maschine weitergeleitet oder einfach durch das Netzwerk transportiert. Dem Dienst soll auch eine Security Policy zugeordnet werden. Dort wird bestimmt, ob ein Network Access Controll benötigt wird und welche Daten erlaubt sind. Es könne

auch mehrere Policy definiert werden. So können unterschiedliche Daten auch anders behandelt werden.

Im Datacenter Teil kann eine virtuelle Maschine erstellt werden und die Daten, die im Netzwerkteil definiert wurden, darauf umgeleitet werden. So kann eine VM von aussen zugänglich gemacht werden oder der gesamte Datenverkehr von den Usern wird über die virtuelle Maschine geführt. So kann zum Beispiel eine Filterung realisiert oder einfach eine Statistik erstellt werden.

Im letzten Teil kann der Service in einem WLAN Netz verfügbar gemacht werden. Dazu muss eine SSID definiert werden, und die Verschlüsselung. So werden nun alle Daten, die über diese SSID verschickt werden, auf die virtuelle Maschine umgeleitet. Dazu werden die Flows verwendet, die im Netzwerk Teil definiert wurden. So kann aus den drei Teilen ein Dienst zusammengestellt werden. Das Konfigurationsmanagement ist also mit einem Orchestrar lösbar.

Wie kann nun ein Orchestrar den Administrator bei der Fehlersuche unterstützen? Auf dem Userinterface vom Orchestrar soll eine Verbindung getestet werden können. Dabei sollen eine Source und eine Destination ausgewählt werden können. Dies können einzelne Hosts sein oder auch ein Raum, oder ein Gebäude. Die Zuordnung von Switch Port zu einem Raum wird über einen Dictionary Dienst sichergestellt. So kann zum Beispiel eine Netzwerk Policy auf einen Raum angewendet werden und muss nicht für jeden einzelnen Port definiert werden.

Danach fragt der Orchestrar die Controller nach den Floweinträgen für die zu testende Verbindung ab. Wenn Flows von der Source bis zu der Destination definiert sind und auch das Protokoll richtig gewählt ist, funktioniert die Verbindung. So kann in kurzer Zeit die Verfügbarkeit von einem Netzwerk überprüft werden. Der Orchestrar kann also auch die Fehlersuche vereinfachen.

Als Nächstes wird das User Management besprochen. Ein Orchestrar kann alle User, die berechtigt sind, auf das Netzwerk zuzugreifen, in einer Datenbank abspeichern. Ein User kann einer Gruppe zugeordnet werden. Weiter wird einer Gruppe oder einem einzelnen User eine Policy zugewiesen. Mit dieser werden die Berechtigungen definiert. Auf welche Dienste darf zugegriffen werden, von wo darf sich ein User ins Netz verbinden und welche Applikationen darf er verwenden. Der Orchestrar muss auch eine Schnittstelle zu einer externen Userverwaltung besitzen. So muss bei der Umstellung, von einem Netzwerk, nicht gleich die gesamte Userverwaltung ersetzt werden. Die User und Usergruppen müssen einfach in den Orchestrar importiert werden können. So muss nicht jeder einzelne User neu erfasst werden.

Damit ein Orchestrar seine Logging Dateien sicher abspeichern kann, benötigt er eine Schnittstelle zu einem Logging System. Dies kann ein Syslog Server sein. Nun kann der Orchestrar erfolgreiche und auch erfolglose Anmeldeversuche protokollieren. Weiter können auch alle Befehle, die ein Administrator ausführt, protokolliert werden. Bei der Verrechnung von einem Dienst, den ein Kunde benutzt, wird es ein bisschen komplizierter. In der Datenbank vom Orchestrar werden alle Flows abgespeichert und einem Dienst zugeordnet. So können nun alle Switch abgefragt werden, wie viele Daten sie für einen Dienst transportiert haben. Dies wird dann dem Kunden in Rechnung gestellt.

Ein Dienst kann jedoch auch von unterschiedlichen Benutzern verwendet werden. Als Beispiel wird hier ein Hotspot erwähnt. Ein Provider bietet öffentliche Hotspots an. Ein Kunde kann sich anmeldet

und für ein kleines End Geld im Internet surfen. Hier kann die Abrechnung nicht mehr für den ganzen Dienst gemacht werden, sondern muss auf die einzelnen Benutzer aufgeteilt werden. Dies kann gelöst werden, indem ein neuer Flow erstellt wird, sobald sich ein Benutzer anmeldet. Sobald sich der Benutzer wieder abmeldet, wird der Flow abgefragt, wie viel Daten übertragen wurden. Dies wird, in der Datenbank, dem Benutzer zugewiesen. So kann das Datenvolumen dem Benutzer in Rechnung gestellt werden.

Mit einem Orchestrar ist es auch möglich, die Leistung von einem Netzwerk zu überwachen. Dazu werden die verwendeten Bandbreiten der einzelnen Ports ausgelesen. Dies kann zum Beispiel mit NetFlow gelöst werden. So kann eine Statistik erstellt werden, wie stark eine Verbindung ausgelastet ist. Diese Statistik kann in einem PDF File zusammengefasst oder auch ausgedruckt werden. So können Berichte, für die Management Etage, erstellt werden.

Falls eine Verbindung zu stark ausglastet ist, wird analysiert, um was für Daten es sich handelt. Der Orchestrar entscheidet nun, welche Daten businesscritical sind und welche nicht. Alle Daten, die nicht so wichtig sind, werden mit Hilfe von Policy auf einen anderen Link verschoben. Diese Policy sind in der Datenbank des Orchestrar gespeichert. Das Verschieben, von den weniger wichtigen Daten, wird mittels Flows gelöst. Der Orchestrar generiert mehrere Flows, die auf diese Daten passen. Danach werden die Flows im Netzwerk so verteilt, dass ein alternativer Link verwendet wird. So kann die Belastung auf einem Link flexibel angepasst werden. Dies kann für einen Service Provider genauso interessant sein wie für eine Firma mit mehreren Uplinks.

Zum Schluss des Kapitels wird noch das Sicherheitsmanagement besprochen. Ein Orchestrar muss die Möglichkeit haben, periodisch die Sicherheit im Netzwerk zu überprüfen. Dazu überprüft er, ob die Sicherheitspolicy richtig umgesetzt ist im Netzwerk. Ein Eintrag könnte wie folgt lauten: Aus dem Netz A dürfen nur ICMP Echo Request Pakete in das Netz B geschickt werden. Andere Daten sind nicht erlaubt zwischen diesen zwei Netzwerken. Der Orchestrar kann dies nun anhand der Flowtabel der einzelnen Switch überprüfen. Hat nun ein Angreifer einen zusätzlichen Flow auf einem Switch installiert, kann der Orchestrar dies erkennen und beheben. Weiter soll ein Orchestrar auch mit einem IDS zusammenarbeiten können. Erkennt ein IDS einen Angriff oder einen Virus, kann der Angreifer isoliert werden. Dies wird auch wieder über die Flowtable gelöst.

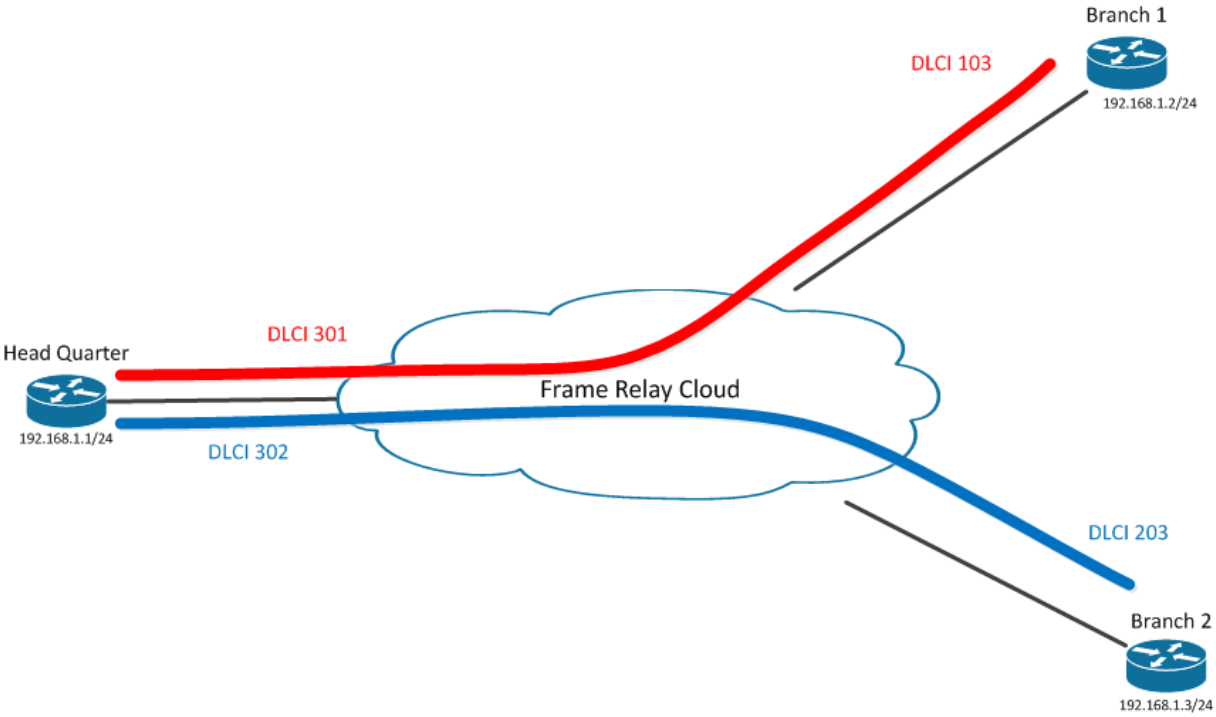
Der höchste Schutz muss jedoch für den Orchestrar selbst gewährleistet sein. Wenn ein Angreifer die Kontrolle über ihn erlangen würde, fällt die ganze Sicherheit vom Netzwerk. Dies kann nur gewährleistet werden, wenn aus einem Clientnetz keine Daten an den Orchestrar geschickt werden dürfen. Deshalb muss die 802.1x Authentisierung von einem Switch ausgeführt werden und nicht direkt vom Orchestrar. Auch der Zugriff von einem Administrator muss auf ein Netz begrenzt werden.

5.9 Vergleich zu herkömmlichen Technologien

5.9.1 Frame Relay

Um einen Vergleich zwischen Software Defined Network und Frame Relay machen zu können, müssen wir uns zuerst die Funktionsweise von Frame Relay anschauen.

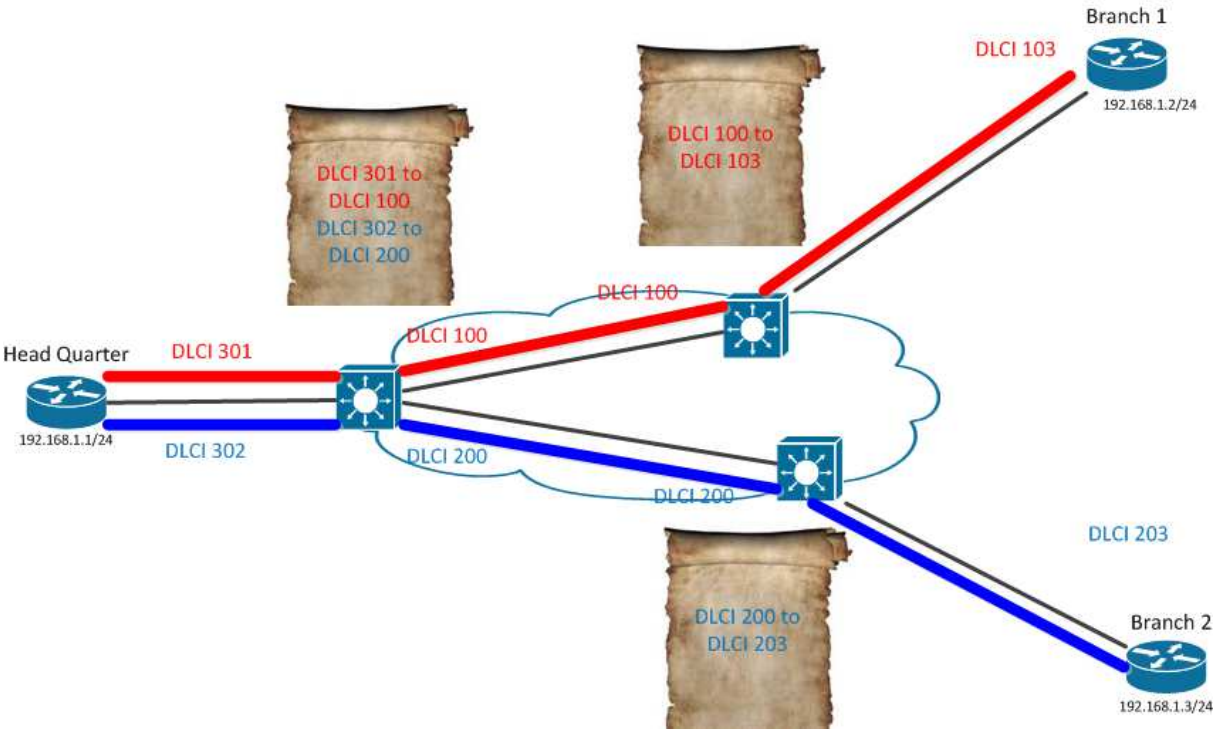
Frame Relay ist ein circuit switched Netzwerk. Das heisst, dass nicht in jedem Paket die Empfänger- und die Absenderadressen eingetragen sind. So „weiss“ also nicht das Paket wohin es muss, sondern das Netzwerk muss diese Entscheidungen treffen. Dies geschieht anhand von sogenannten Permanent virtual circuit (PVC). Im nachstehenden Bild wird dies verdeutlicht.



PVC in einem Frame Relay Netz

Man sieht einen roten PVC und eine Blauen. Der Rote Permanent virtual Circuit verbindet das Head Quarter mit der Branche1. Der Kunde konfiguriert auf dem Router im Head Quarter einen Data-Link Connection Identifier (DLCI) 301. In der Branche1 wird nun ein DLCI 103 konfiguriert für den Rückweg. Werden nun Daten vom Head Quarter in die Branch1 geschickt, werden sie vom Router, in ein Frame Relay Paket verpackt und mit dem DLCI 301 versehen. Der Service Provider schickt das Paket nun durch das Frame Relay Netz zur Branch1. Der Router in dem Branch Office entpackt das Paket und schickt es als normales Ethernet Frame ins interne Netzwerk. Beim Rückweg, also von der Branch1 ins Head Quarter, wird das Ethernet Frame in ein Frame Relay Paket verpackt und mit dem DLCI 103 versehen. Auch dieses Paket wird nun vom Service Provider zurück ins Head Quarter geschickt. Dort entpackt der Router das Paket und schickt es ins interne Netz.

Sauen wir uns nun den Konfigurationsaufwand für den Service Provider ein bisschen genauer an. Im nachfolgenden Bild sieht man, dass der Service Provider zwischen seinen Frame Relay Switch unterschiedliche DLCIs konfigurieren kann. Weiter sind die Frame Relay Tabellen auf den Frame Relay Switch dargestellt. Diese werden für die Weiterleitung von Paketen benötigt.



Genauere Betrachtung von einem Frame Relay Netzwerk

Schauen wir uns wieder den roten Permanent virtual circuit an. Sollen Daten vom Head Quarter in das Branch1 Office geschickt werden, wird das Ethernet Frame wieder in ein Frame Relay Paket verpackt und mit dem Data-Link Connection Identifier versehen. Nun wird das Packet an einen Frame Relay Switch vom Service Provider geschickt. Dieser schaut nun in der Frame Relay Tabelle nach was mit diesem Paket geschehen soll. In unserem Fall steht in der Tabelle, dass der DLCI 301 in den DLCI 100 umgewandelt werden soll. Also wird der DLCI im Paket verändert und weiter zum nächsten Frame Relay Switch geschickt. Bei diesem angekommen wird wieder in der Frame Relay Tabelle nachgeschaut und festgestellt, dass der DLCI 100 in den DLCI 103 umgewandelt werden soll. Nach der Änderung des DLCI kann das Paket an den Router im Branch1 Office weitergeleitet werden.

An diesem Beispiel sieht man nun, dass ein Service Provider für jede Verbindung viele DLCIs konfigurieren muss. Das ist mit grossem Aufwand verbunden. Nun wollen wir den Konfigurationsaufwand mit einem Software Defined Network vergleichen.

Ein Netzwerk kann in Slices unterteilt werden. Das heisst, ein physikalisches Netz wird in verschiedene logische Netze unterteilt. Dies wurde im Kapitle „SDN im Provider Netz“ genauer behandelt. Mit einem Slice kann nun das Gleiche erreicht werden, wie mit einem PVC. Es werden logische Verbindungen zwischen zwei Netzwerken hergestellt. Ein Slice muss jedoch nicht zwischen

zwei SDN Switch konfiguriert werden sondern auf dem Controller. So ist der Konfigurationsaufwand nur ein Bruchteil von dem Aufwand bei Frame Relay. Das bedeutet zwar, dass mit SDN Kosten gespart werden können gegenüber Frame Relay, aber die Anschaffungen für SDN fähige Switch sind hoch. Weiter ist SDN eine relativ neue Technologie. Das heisst es sind noch einige „Kinderkrankheiten“ vorhanden. Solche kleine Fehler können in einem produktiven Netzwerk fatale Auswirkungen haben. Wenn also ein Netzwerk von Frame Relay auf SDN umgestellt werden soll, muss dies sehr genau geplant und getestet werden. Eine Umstellung findet am besten Schrittweise statt, wie am Beispiel von Google schon erklärt wurde.

5.9.2 SDN vs Performance Routing

Wenn von Software Defined Network die Rede ist, hört man immer wieder, dass es möglich sei, bestimmten Datenverkehr von einem Link auf einen anderen umzuleiten. Dies macht Sinn, wenn die Bandbreite bald erschöpft ist. So kann zum Beispiel Voice Datenverkehr auf einen Backup Link umgeleitet werden, der noch mehr Bandbreite zur Verfügung hat. So wird der primäre Link entlastet und der Voice Verkehr wird nicht beeinträchtigt. Ein weiterer Anwendungsfall ist, wenn eine Firma sehr zeitkritische Anwendungen betreibt. Das kann zum Beispiel eine Trading Plattform sein. Hier müssen die Daten in Echtzeit übertragen werden. Hat nun ein Service Provider ein Problem im Netz, hat dies meinst Auswirkungen auf die Latenz. Das heisst die Daten können nicht mehr in Echtzeit übertragen werden. In einem solchen Fall soll das Netzwerk reagieren und die zeitkritischen Daten auf einen anderen Link weiterleiten. Heute ist dies mit dem sogenannten Performance Routing möglich. Dabei wird ein Maser Controller und mehrere Border Router benötigt. Der Master Controller trifft die Entscheidungen in einem Performance Routing Netzwerk, muss jedoch nicht im Datenpfad sein. Die Border Router bauen eine TCP Verbindung zum Master Controller auf. Dabei wird als Destination Port 3949 verwendet. Die Border Router verwenden nun IP SLA um die Performance auf einem Link zu messen. Diese Messwerte werden nun mittels NetFlow an den Master Controller übermittelt. Dieser kann nun, wenn ein vordefinierter Wert überschritten wird, eine Default Route an eine Border Router senden. So können die Daten auf einen anderen Link geleitet werden. Diese neue Route wird nicht in die Running Configuration übernommen. Das heisst, dass nach einem Reload des Gerätes wieder der ursprüngliche Link verwendet wird.

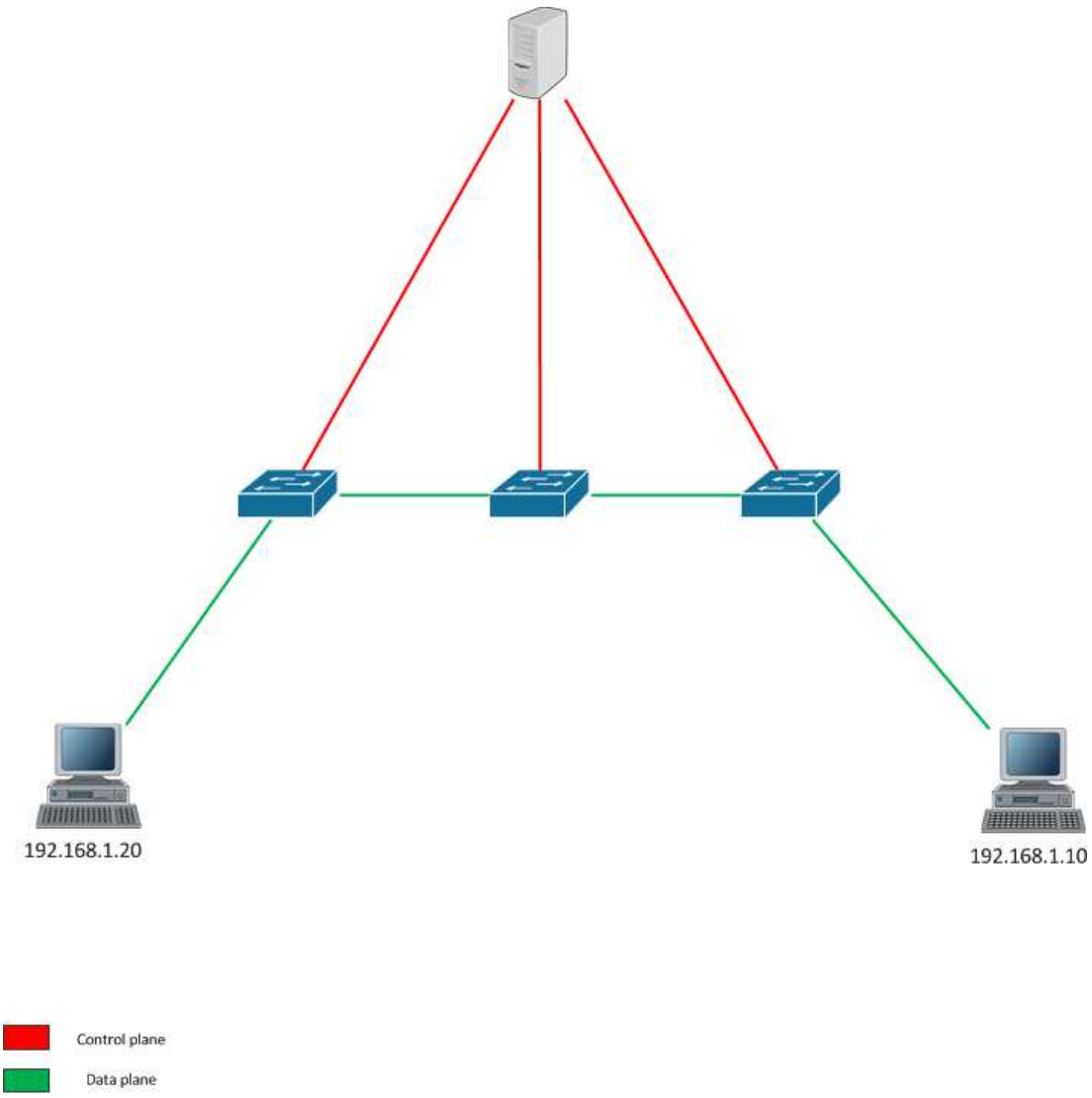
5.10 Testlab

Um einen Vergleich zwischen den Herstellern durchführen zu können, wird ein Test für jeden Use Case aus Kapitel 3.5 erstellt. Falls das benötigte Material zur Verfügung steht werden die Tests auch praktisch durchgeführt.

5.10.1 Fehlermanagement

Bei dem Test des Fehlermanagements wird überprüft, ob ein Fehler im Netz gefunden werden kann. Dabei wird ein Linkausfall simuliert.

5.10.1.1 Testinfrastruktur



5.10.1.2 Test Idee

Auf dem Controller wird überprüft, ob die Testinfrastruktur korrekt angezeigt wird. Danach wird ein Link zwischen den Switch unterbrochen. Erwartet wird dass der Controller den Ausfall erkennt und grafisch anzeigt. So kann der Fehler lokalisiert werden. Das beheben des Fehlers ist nicht Bestandteil des Tests.

5.10.1.3 Testdurchführung

HP 5900 Switch

Von HP stehen drei Switch und ein Controller zur Verfügung. Die Testinfrastruktur wird aufgebaut. Auf dem Webinterface des Controllers wird die Infrastruktur korrekt angezeigt. Danach wird ein Link unterbrochen.

Test	Erwartetes Ergebnis	Ergebnis	Schlussfolgerung
Link Unterbruch	Der Switch meldet dem Controller sofort dass der Link down ist. Auf dem GUI wird der Ausfall sofort angezeigt.	Nach 2 Sekunden wird der Ausfall angezeigt.	Der Switch meldet den Ausfall sofort. Funktioniert nur bei Out-of-Band Management.
Link verbinden	Der Switch meldet dem Controller sofort dass der Link up ist. Auf dem GUI wird der Link sofort angezeigt.	Nach 2 Sekunden wird der Link angezeigt.	Der Switch meldet den Link sofort. Dies funktioniert auch bei in-Band Management.
Unterbruch Mgmt Port	Der Controller bemerkt den Ausfall erst nach einigen Sekunden.	Nach 30 Sekunden wird der Ausfall erkannt.	Der Controller überprüft periodisch, ob die Switch erreichbar sind.

Opendaylight

Der OpenDaylight Controller wird mit mininet getestet. Auf dem Webinterface des Controllers wird die Infrastruktur korrekt angezeigt. Danach wird ein Link unterbrochen.

Test	**Erwartetes Ergebnis**	**Ergebnis**	**Schlussfolgerung**
Link Unterbruch	Der Switch meldet dem Controller sofort dass der Link down ist. Auf dem GUI wird der Ausfall sofort angezeigt.	Der Link ist nach einem reload des GUIs nicht mehr sichtbar.	Der OpenVSwitch meldet sofort, wenn ein Link ausfällt
Link verbinden	Der Switch meldet dem Controller sofort dass der Link up ist. Auf dem GUI wird der Link sofort angezeigt.	Der Link ist nach einem reload des GUIs wieder sichtbar	Der OpenVSwitch meldet sofort, wenn der Link wieder verfügbar ist
Unterbruch Mgmt Port	Kann mit mininet nicht simuliert werden	n/a	n/a

Arista 7150 Switch

Da Arista keinen Controller hat und die Switch OpenFlow noch nicht unterstützen, kann dieser Test nicht durchgeführt werden.

Test	Erwartetes Ergebnis	Ergebnis	Schlussfolgerung
Link Unterbruch	Der Switch meldet dem Controller sofort dass der Link down ist. Auf dem GUI wird der Ausfall sofort angezeigt.	n/a	Der Switch unterstützt Openflow 1.3 im Quartal 4 2014. Deshalb ist anzunehmen, dass der Linkausfall erkannt wird.
Link verbinden	Der Switch meldet dem Controller sofort dass der Link up ist. Auf dem GUI wird der Link sofort angezeigt.	n/a	Es ist anzunehmen, dass der Link wieder angezeigt wird.
Unterbruch Mgmt Port	Kann mit mininet nicht simuliert werden	n/a	n/a

Cisco

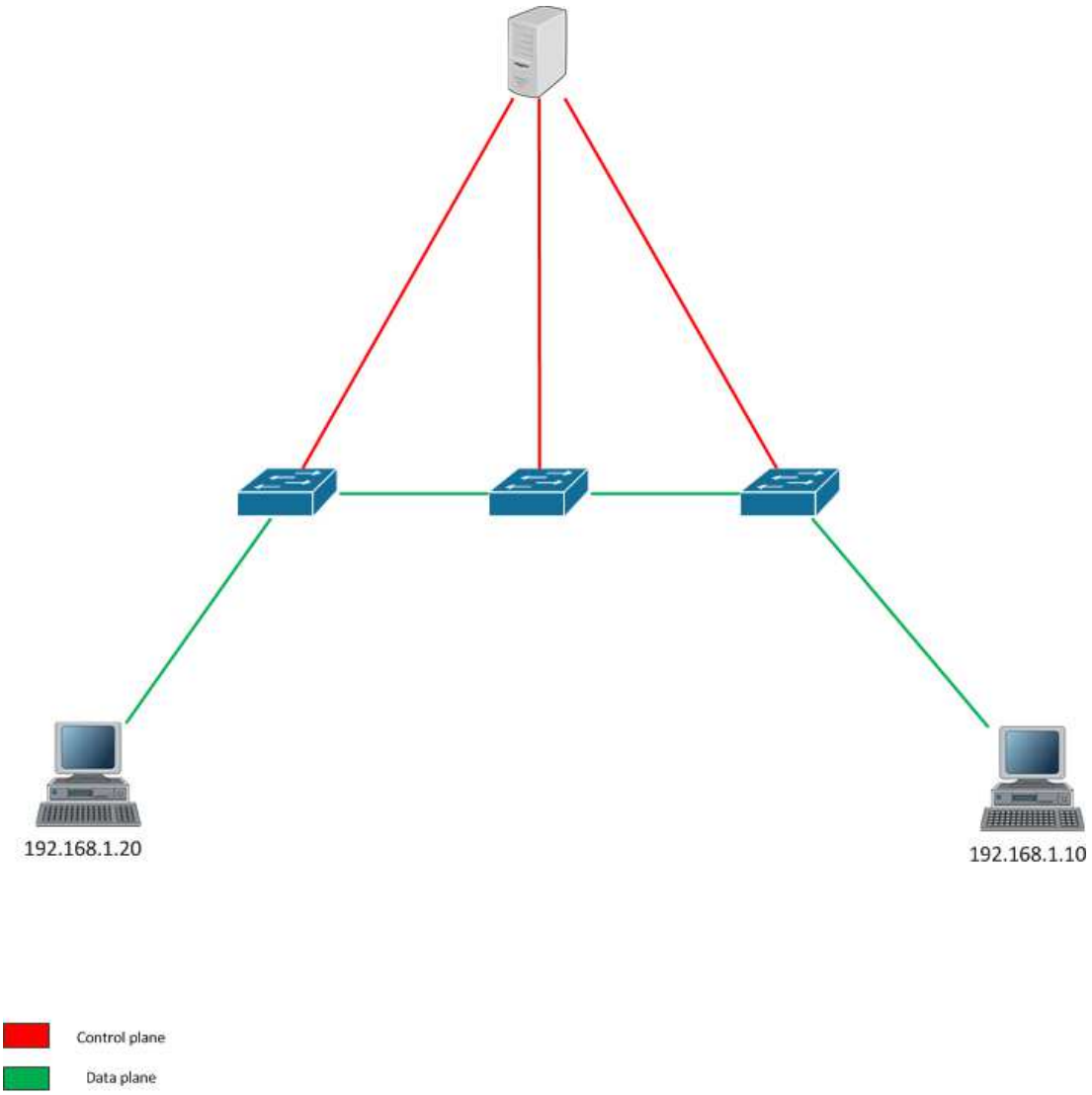
Der Cisco APIC ist noch nicht Verfügbar deshalb wird getestet, welche Möglichkeiten OnePK bietet.

Test	**Erwartetes Ergebnis**	**Ergebnis**	**Schlussfolgerung**
Link Unterbruch	Der Switch meldet dem Controller sofort dass der Link down ist. Auf dem GUI wird der Ausfall sofort angezeigt.	n/a	n/a
Link verbinden	Der Switch meldet dem Controller sofort dass der Link up ist. Auf dem GUI wird der Link sofort angezeigt.	n/a	n/a
Unterbruch Mgmt Port	Kann mit mininet nicht simuliert werden	n/a	n/a

5.10.2 Konfigurationsmanagement

Bei dem Test des Konfigurationsmanagement wird überprüft, ob eine Konfiguration auf ein Device gespielt werden kann.

5.10.2.1 Testinfrastruktur



5.10.2.2 Test Idee

Es wird überprüft, ob eine Konfiguration auf einen Switch gespielt werden kann. Dies kann über Netconf, SNMP oder XMPP erledigt werden.

5.10.2.3 Testdurchführung

HP 5900 Switch

Das Northbound API vom Switch wird auf die vorhandenen Konfigurationsmöglichkeiten überprüft.

Test	**Erwartetes Ergebnis**	**Ergebnis**	**Schlussfolgerung**
Netconf		Wird nicht unterstützt	
SNMP		V1,v2c,v3	Es kann eine Konfiguration auf den Switch gespielt werden. Es können jedoch Fehler passieren und Konfigurationen im Netz übrig bleiben.
XMPP		Wird nicht unterstützt	
SSH		Wird unterstützt	So kann über das Netzwerk auf einen einzelnen Switch zugegriffen werden. Für die Fehlersuche praktisch.
DHCP Autoconfig		Wird unterstützt	Der DHCP Server gibt einen Server an, von dem sich der Switch die Konfiguration runterlädt. So ist eine Autokonfiguration möglich.
CLI		Wird unterstützt	Für die Initiale Konfiguration.

OpenVSwitch

Der OpenDaylight Controller wird mit mininet getestet. Auf dem Webinterface des Controllers wird die Infrastruktur korrekt angezeigt. Danach wird ein Link unterbrochen.

Test	**Erwartetes Ergebnis**	**Ergebnis**	**Schlussfolgerung**
Netconf		Wird nicht unterstützt	OpenVSwitch wird nur über die CLI konfiguriert.
SNMP		Wird nicht unterstützt	
XMPP		Wird nicht unterstützt	
SSH		Wird nicht unterstützt	
CLI		Wird unterstützt	In mininet gibt es spezielle Kommandos, um den OpenVSwitch zu konfigurieren. Z.B. „link s1 s2 down“ stellt den Link zwischen s1 und s2 ab.

Arista Switch 7150

Test	Erwartetes Ergebnis	Ergebnis	Schlussfolgerung
Netconf		Wird nicht unterstützt	
SNMP		Wird unterstützt	Es kann eine Konfiguration auf den Switch gespielt werden. Es können jedoch Fehler passieren und Konfigurationen im Netz übrig bleiben.
XMPP		Wird unterstützt	Es können mehrere Switch gleichzeitig per CLI Kommandos konfiguriert werden. So kann ein Dienst im Netzwerk vorbereitet werden.
SSH		Wird unterstützt	So kann über das Netzwerk auf einen einzelnen Switch zugegriffen werden. Für die Fehlersuche praktisch.
CLI		Wird unterstützt	Für die Initiale Konfiguration.

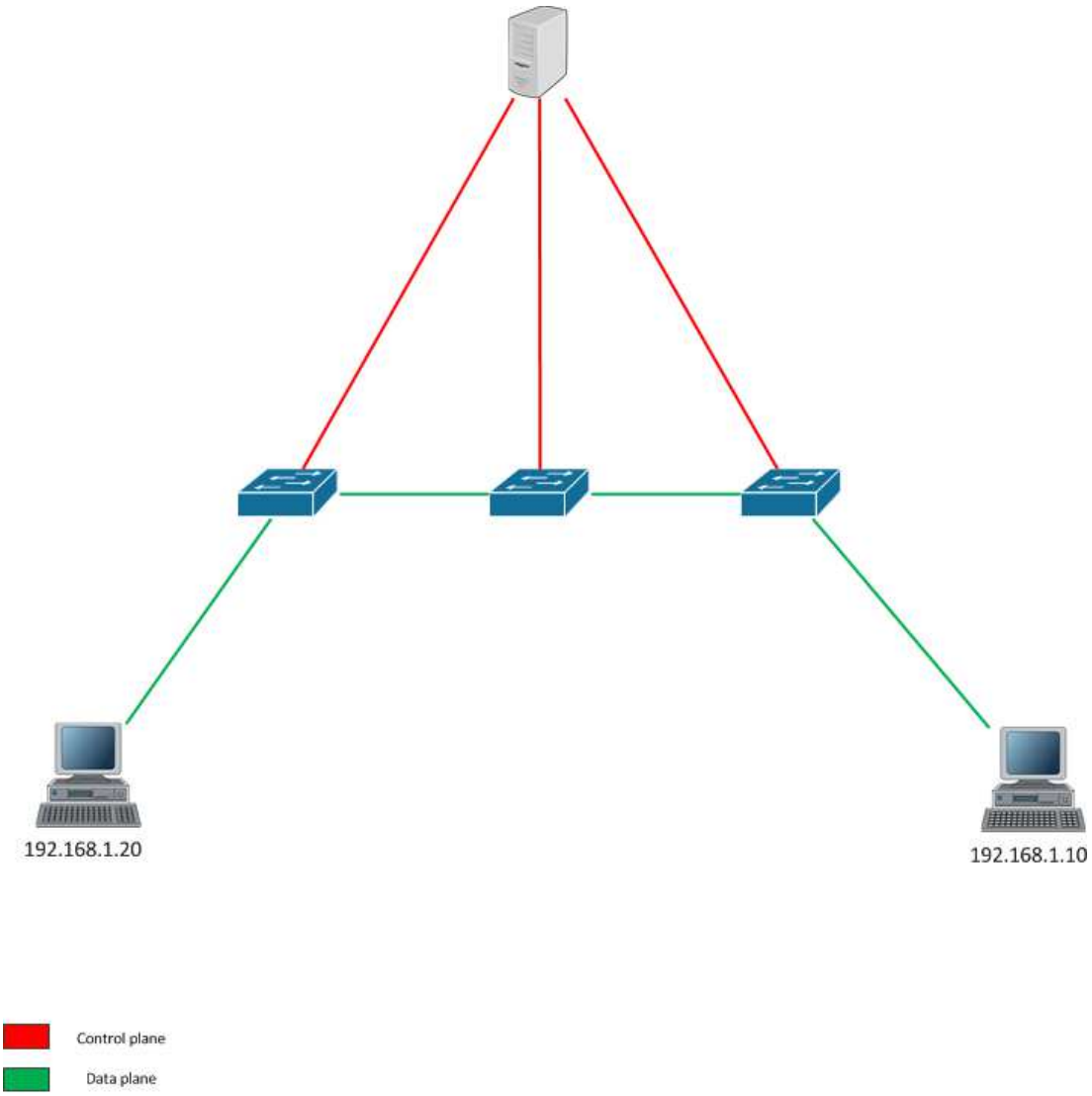
Cisco Nexus 1000V

Test	Erwartetes Ergebnis	Ergebnis	Schlussfolgerung
Netconf		Wird unterstützt	Es kann eine Konfiguration eingespielt werden. Der Switch meldet zurück, ob es geklappt hat oder nicht. So können keine fehlerhaften Konfigurationen im Netz zurückbleiben.
SNMP		Wird unterstützt	Es kann eine Konfiguration auf den Switch gespielt werden. Es können jedoch Fehler passieren und Konfigurationen im Netz übrig bleiben.
XMPP		Wird nicht unterstützt	
SSH		Wird unterstützt	So kann über das Netzwerk auf einen einzelnen Switch zugegriffen werden. Für die Fehlersuche praktisch.
CLI		Wird unterstützt	Für die Initiale Konfiguration.

5.10.3 Abrechnungsmanagement

Bei dem Test des Abrechnungsmanagement wird überprüft, ob die Datenmenge, die über einen Flow gesendet wurde, ausgelesen werden kann.

5.10.3.1 Testinfrastruktur



5.10.3.2 Test Idee

Es wird überprüft, ob erfasst werden kann, wie viele Daten pro Flow transportiert wurden.

5.10.3.3 Testdurchführung

HP 5900 Switch

Das Northbound API vom Switch wird auf die vorhandenen Konfigurationsmöglichkeiten überprüft.

Test	**Erwartetes Ergebnis**	**Ergebnis**	**Schlussfolgerung**
Loggen von Daten per Syslog	Es kann ein Syslog Server angegeben werden.	Es kann ein Syslog Server konfiguriert werden.	Die Logging Daten können sicher abgespeichert werden.
Loggen von Daten per SNMP Traps	Bei einem Ereignis werden SNMP Traps an den Server gesendet	Der Switch kann SNMP Traps senden.	Der Switch kann einen Alarm auslösen bei einem bestimmten Ereignis.
Melden von Ereignissen per OpenFlow	Die Counters der Flowtable können ausgelesen werden.	Die Counters können ausgelesen werden.	Es kann eine Statistik erstellt werden, wie viele Daten pro Flow übertragen wurden.

OpenVSwitch

Der OpenDaylight Controller wird mit mininet getestet. Auf dem Webinterface des Controllers wird die Infrastruktur korrekt angezeigt. Danach wird ein Link unterbrochen.

Test	Erwartetes Ergebnis	Ergebnis	Schlussfolgerung
Loggen von Daten per Syslog	Es kann ein Syslog Server angegeben werden.	Es kann ein Syslog Server konfiguriert werden.	Die Logging Daten können sicher abgespeichert werden.
Loggen von Daten per SNMP Traps	Bei einem Ereignis werden SNMP Traps an den Server gesendet	Der Switch kann keine SNMP Traps senden.	Da mininet nur für Testumgebungen gedacht ist, ist SNMP nicht zwingend notwendig.
Melden von Ereignissen per OpenFlow	Die Counters der Flowtable können ausgelesen werden.	Die Counters können ausgelesen werden.	Es kann eine Statistik erstellt werden, wie viele Daten pro Flow übertragen wurden.

Arista 7150 Switch

Da Arista keinen Controller hat und die Switch OpenFlow noch nicht unterstützen, kann dieser Test nicht durchgeführt werden.

Test	Erwartetes Ergebnis	Ergebnis	Schlussfolgerung
Loggen von Daten per Syslog	Es kann ein Syslog Server angegeben werden.	Es kann ein Syslog Server konfiguriert werden.	Die Logging Daten können sicher abgespeichert werden.
Loggen von Daten per SNMP Traps	Bei einem Ereignis werden SNMP Traps an den Server gesendet	Der Switch kann SNMP Traps senden.	Der Switch kann einen Alarm auslösen bei einem bestimmten Ereignis.
Melden von Ereignissen per OpenFlow	Die Counters der Flowtable können ausgelesen werden.	Arista unterstütz noch kein OpenFlow.	

Cisco Nexus 1000V

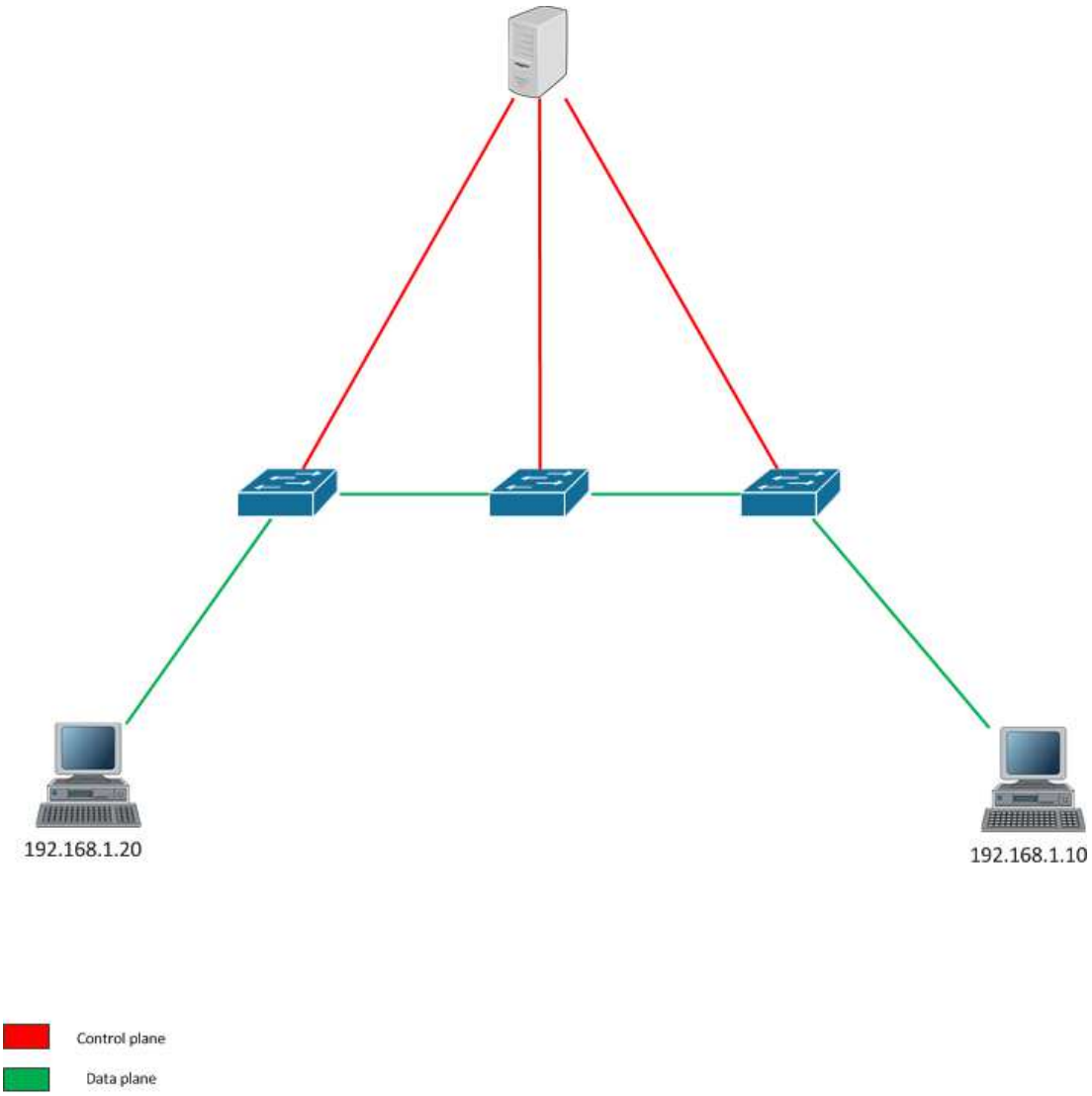
Der Cisco APIC ist noch nicht Verfügbar deshalb wird getestet, welche Möglichkeiten OnePK bietet.

Test	**Erwartetes Ergebnis**	**Ergebnis**	**Schlussfolgerung**
Loggen von Daten per Syslog	Es kann ein Syslog Server angegeben werden.	Es kann ein Syslog Server konfiguriert werden.	Die Logging Daten können sicher abgespeichert werden.
Loggen von Daten per SNMP Traps	Bei einem Ereignis werden SNMP Traps an den Server gesendet	Der Switch kann SNMP Traps senden.	Der Switch kann einen Alarm auslösen bei einem bestimmten Ereignis.
Melden von Ereignissen per OpenFlow	Die Counters der Flowtable können ausgelesen werden.	Die Counters können ausgelesen werden.	Es kann eine Statistik erstellt werden, wie viele Daten pro Flow übertragen wurden.

5.10.4 Leistungsmanagement

Bei dem Test des Leistungsmanagement wird überprüft, ob die Auslastung von einem Link ausgelesen werden kann.

5.10.4.1 Testinfrastruktur



5.10.4.2 Test Idee

Es wird überprüft, ob erfasst werden kann, wie stark ein Link ausgelastet ist.

5.10.4.3 Testdurchführung

HP 5900 Switch

Das Northbound API vom Switch wird auf die vorhandenen Konfigurationsmöglichkeiten überprüft.

Test	Erwartetes Ergebnis	Ergebnis	Schlussfolgerung
Auslesen der Datenmenge per Openflow	Es wird die Datenmenge für einen Flow angezeigt	Ist möglich mit OpenFlow 1.3	
Auslesen der Datenmenge per NetFlow	Es wird die Datenmenge für einen Flow oder für ein Interface angezeigt.	Wird nicht unterstützt	
Auslesen der Datenmenge per SNMP	Es wird die Datenmenge für einen Flow oder für ein Interface angezeigt.	Ist möglich	
Auslesen der Datenmenge per SSH	Es wird die Datenmenge für einen Flow oder für ein Interface angezeigt.	Ist möglich	

Opendaylight

Der OpenDaylight Controller wird mit mininet getestet. Auf dem Webinterface des Controllers wird die Infrastruktur korrekt angezeigt. Danach wird ein Link unterbrochen.

Test	**Erwartetes Ergebnis**	**Ergebnis**	**Schlussfolgerung**
Auslesen der Datenmenge per Openflow	Es wird die Datenmenge für einen Flow angezeigt	Ist möglich mit OpenFlow 1.3	
Auslesen der Datenmenge per NetFlow	Es wird die Datenmenge für einen Flow oder für ein Interface angezeigt.	Nicht möglich	
Auslesen der Datenmenge per SNMP	Es wird die Datenmenge für einen Flow oder für ein Interface angezeigt.	Nicht möglich	
Auslesen der Datenmenge per SSH	Es wird die Datenmenge für einen Flow oder für ein Interface angezeigt.	Nicht möglich	

Arista 7150 Switch

Da Arista keinen Controller hat und die Switch OpenFlow noch nicht unterstützen, kann dieser Test nicht durchgeführt werden.

Test	**Erwartetes Ergebnis**	**Ergebnis**	**Schlussfolgerung**
Auslesen der Datenmenge per Openflow	Es wird die Datenmenge für einen Flow angezeigt	Nicht möglich	
Auslesen der Datenmenge per NetFlow	Es wird die Datenmenge für einen Flow oder für ein Interface angezeigt.	Ist möglich	
Auslesen der Datenmenge per SNMP	Es wird die Datenmenge für einen Flow oder für ein Interface angezeigt.	Ist möglich	
Auslesen der Datenmenge per SSH	Es wird die Datenmenge für einen Flow oder für ein Interface angezeigt.	Ist möglich	

Cisco Nexus 1000V

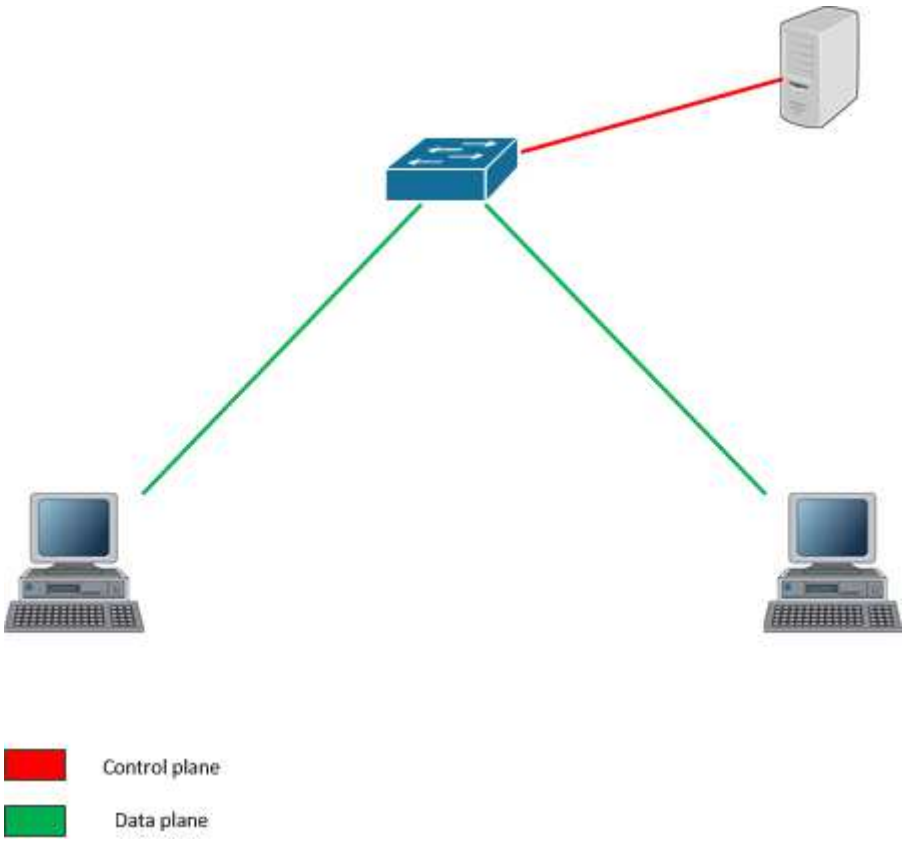
Der Cisco APIC ist noch nicht Verfügbar deshalb wird getestet, welche Möglichkeiten OnePK bietet.

Test	**Erwartetes Ergebnis**	**Ergebnis**	**Schlussfolgerung**
Auslesen der Datenmenge per Openflow	Es wird die Datenmenge für einen Flow angezeigt	Ist möglich	
Auslesen der Datenmenge per NetFlow	Es wird die Datenmenge für einen Flow oder für ein Interface angezeigt.	Ist möglich	
Auslesen der Datenmenge per SNMP	Es wird die Datenmenge für einen Flow oder für ein Interface angezeigt.	Ist möglich	
Auslesen der Datenmenge per SSH	Es wird die Datenmenge für einen Flow oder für ein Interface angezeigt.	Ist möglich	

5.10.5 Sicherheitsmanagement

Bei dem Test des Sicherheitsmanagement wird überprüft, ob ein Gerät vor unerlaubtem Zugriff geschützt werden kann.

5.10.5.1 Testinfrastruktur



5.10.5.2 Test Idee

Es wird überprüft, ob ein Flow generiert werden kann, der bestimmte Daten dropt.

5.10.5.3 Testdurchführung

HP 5900 Switch

Das Northbound API vom Switch wird auf die vorhandenen Konfigurationsmöglichkeiten überprüft.

Test	Erwartetes Ergebnis	Ergebnis	Schlussfolgerung
Es wird ein Flow erstellt, der eine MAC Adresse blockiert	Es wird nur die definierte MAC Adresse blockiert	Der Switch unterstützt die Drop Aktion. Der Controller jedoch nicht.	Der Switch erlaubt Firewall Funktionen.
Es wird ein Flow erstellt, der eine IP Adresse blockiert	Es wird nur die definierte IP Adresse blockiert	Der Switch unterstützt die Drop Aktion. Der Controller jedoch nicht.	Der Switch erlaubt Firewall Funktionen.
Es wird ein Flow erstellt, der HTTP blockiert	Es wird nur Webtraffic blockiert	Der Switch unterstützt die Drop Aktion. Der Controller jedoch nicht.	Der Switch erlaubt Firewall Funktionen.

Opendaylight

Der OpenDaylight Controller wird mit mininet getestet. Auf dem Webinterface des Controllers wird die Infrastruktur korrekt angezeigt. Danach wird ein Link unterbrochen.

Test	**Erwartetes Ergebnis**	**Ergebnis**	**Schlussfolgerung**
Es wird ein Flow erstellt, der eine MAC Adresse blockiert	Es wird nur die definierte MAC Adresse blockiert	Ist möglich	Es können Firewall Funktionen implementiert werden.
Es wird ein Flow erstellt, der eine IP Adresse blockiert	Es wird nur die definierte IP Adresse blockiert	Ist möglich	Es können Firewall Funktionen implementiert werden.
Es wird ein Flow erstellt, der HTTP blockiert	Es wird nur Webtraffic blockiert	Ist möglich	Es können Firewall Funktionen implementiert werden.

Arista 7150 Switch

Da Arista keinen Controller hat und die Switch OpenFlow noch nicht unterstützen, kann dieser Test nicht durchgeführt werden.

Test	Erwartetes Ergebnis	Ergebnis	Schlussfolgerung
Es wird ein Flow erstellt, der eine MAC Adresse blockiert	Es wird nur die definierte MAC Adresse blockiert	Open Flow wird nicht unterstützt.	
Es wird ein Flow erstellt, der eine IP Adresse blockiert	Es wird nur die definierte IP Adresse blockiert	Open Flow wird nicht unterstützt.	
Es wird ein Flow erstellt, der HTTP blockiert	Es wird nur Webtraffic blockiert	Open Flow wird nicht unterstützt.	

Cisco Nexus 1000V

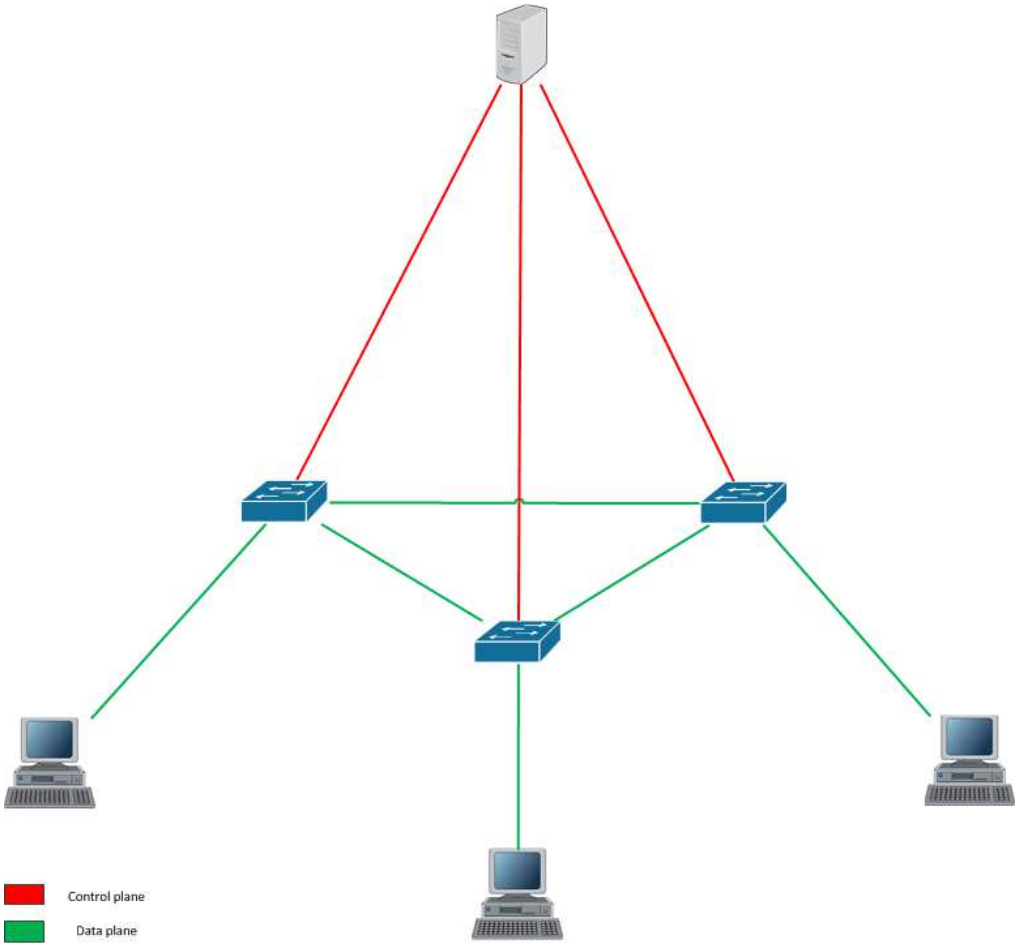
Der Cisco APIC ist noch nicht Verfügbar deshalb wird getestet, welche Möglichkeiten OnePK bietet.

Test	**Erwartetes Ergebnis**	**Ergebnis**	**Schlussfolgerung**
Es wird ein Flow erstellt, der eine MAC Adresse blockiert	Es wird nur die definierte MAC Adresse blockiert	Der Switch unterstützt die Drop Aktion. Der Controller existiert jedoch noch nicht.	
Es wird ein Flow erstellt, der eine IP Adresse blockiert	Es wird nur die definierte IP Adresse blockiert	Der Switch unterstützt die Drop Aktion. Der Controller existiert jedoch noch nicht.	
Es wird ein Flow erstellt, der HTTP blockiert	Es wird nur Webtraffic blockiert	Der Switch unterstützt die Drop Aktion. Der Controller existiert jedoch noch nicht.	

5.10.6 Routing und Rerouting

Bei dem Test des Routings wird überprüft, ob ein alternativer Pfad gefunden wird und auch genutzt werden kann.

5.10.6.1 Testinfrastruktur



5.10.6.2 Test Idee

Das Netzwerk besitzt zwei Datenpfade. Der Controller wählt den „besseren“ aus und verwendet diesen. Wird nun der aktive Link unterbrochen, muss der Controller ein Failover einleiten.

5.10.6.3 Testdurchführung

HP 5900 Switch

Es werden Daten gesendet über den aktiven Link. Danach wird dieser Link unterbrochen. Die Zeit für das Rerouting wird gemessen.

Test	Erwartetes Ergebnis	Ergebnis	Schlussfolgerung
Wird der beste Pfad gefunden?	Die Daten fliessen nur über zwei Switch.	Ja	
Funktioniert das Rerouting?	Nach dem Unterbruch wird innerhalb von Sekunden auf den sekundären Link umgeschaltet.	Ja innerhalb von 2 Sekunden wird auf den Backuplink umgeschalten.	Das Failover funktioniert gut

Opendaylight

Der OpenDaylight Controller wird mit mininet getestet. Auf dem Webinterface des Controllers wird die Infrastruktur korrekt angezeigt. Danach wird ein Link unterbrochen.

Test	Erwartetes Ergebnis	Ergebnis	Schlussfolgerung
Wird der beste Pfad gefunden?	Die Daten fliessen nur über zwei Switch.	Ja	
Funktioniert das Rerouting?	Nach dem Unterbruch wird innerhalb von Sekunden auf den sekundären Link umgeschaltet.	Ja innerhalb von 2 Sekunden wird auf den Backuplink umgeschalten.	Das Failover funktioniert gut

Arista 7150 Switch

Da Arista keinen Controller hat und die Switch OpenFlow noch nicht unterstützen, kann dieser Test nicht durchgeführt werden.

Test	Erwartetes Ergebnis	Ergebnis	Schlussfolgerung
Wird der beste Pfad gefunden?	Die Daten fliessen nur über zwei Switch.	n/a	
Funktioniert das Rerouting?	Nach dem Unterbruch wird innerhalb von Sekunden auf den sekundären Link umgeschaltet.	n/a	

Cisco Nexus 1000V

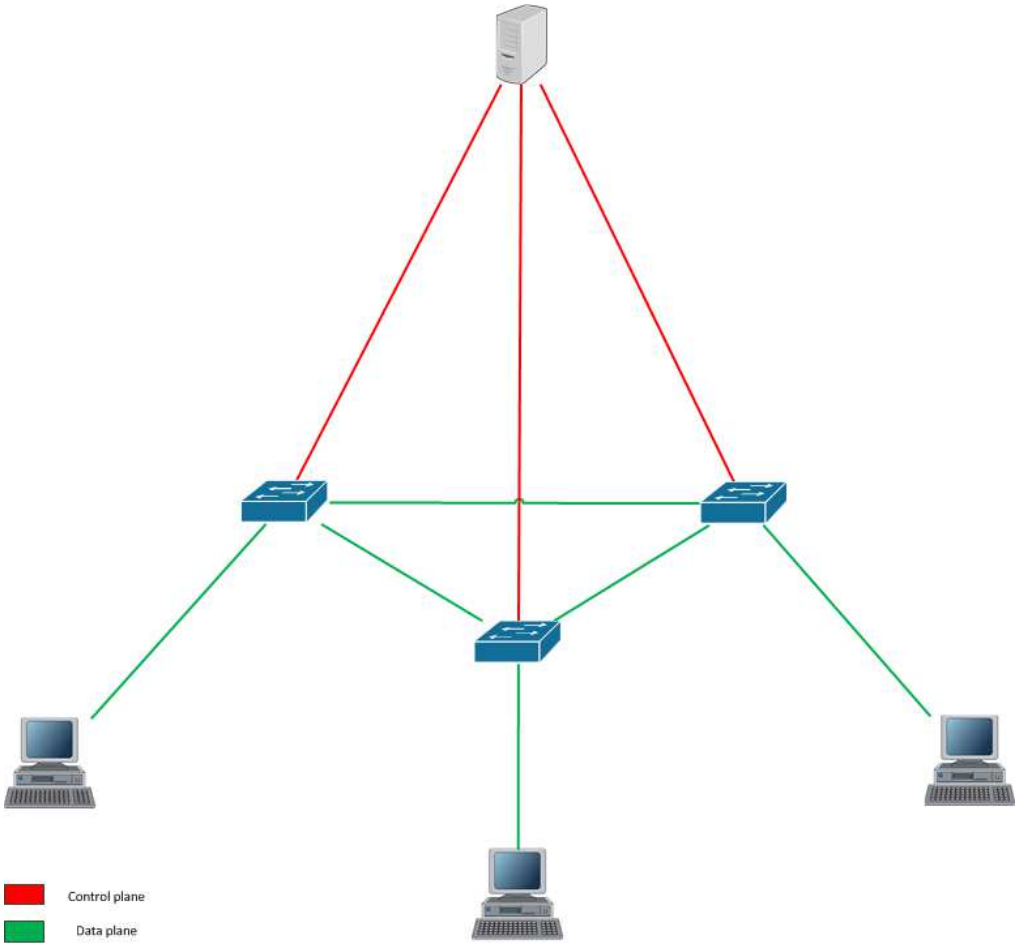
Der Cisco APIC ist noch nicht Verfügbar deshalb wird getestet, welche Möglichkeiten OnePK bietet.

Test	**Erwartetes Ergebnis**	**Ergebnis**	**Schlussfolgerung**
Wird der beste Pfad gefunden?	Die Daten fliessen nur über zwei Switch.	n/a	
Funktioniert das Rerouting?	Nach dem Unterbruch wird innerhalb von Sekunden auf den sekundären Link umgeschaltet.	n/a	

5.10.7 Explizites Routing

Beim Test des expliziten Routings wird überprüft, ob bestimmte Daten auf einen anderen Link umgeleitet werden können.

5.10.7.1 Testinfrastruktur



5.10.7.2 Test Idee

Das Netzwerk besitzt zwei Datenpfade. Der Controller wählt den „besseren“ aus und verwendet diesen. Nun wird ein Flow generiert, der den gesamten HTTP Traffic auf den sekundären Link umleitet.

5.10.7.3 Testdurchführung

HP 5900 Switch

Es werden Daten gesendet über den aktiven Link. Danach wird der HTTP Traffic auf den sekundären Link umgeleitet. Dies wird anhand der Datenmenge, die über diesen Flow geschickt wird, überprüft.

Test	**Erwartetes Ergebnis**	**Ergebnis**	**Schlussfolgerung**
Anlegen eines Flows für HTTP.	Es kann ein Flow für HTTP angelegt werden.	Der Switch unterstützt keine Layer 7 Inspection	Explizites Routing ist nicht möglich.
Der Flow zeigt auf den sekundären Link.	Der HTTP Traffic fliesst auf dem sekundären Link.	Der Switch unterstützt keine Layer 7 Inspection	Explizites Routing ist nicht möglich.

Opendaylight

Der OpenDaylight Controller wird mit mininet getestet. Auf dem Webinterface des Controllers wird die Infrastruktur korrekt angezeigt. Danach wird ein Link unterbrochen.

Test	Erwartetes Ergebnis	Ergebnis	Schlussfolgerung
Anlegen eines Flows für HTTP.	Es kann ein Flow für HTTP angelegt werden.	Der Switch unterstützt keine Layer 7 Inspection	Explizites Routing ist nicht möglich.
Der Flow zeigt auf den sekundären Link.	Der HTTP Traffic fliesst auf dem sekundären Link.	Der Switch unterstützt keine Layer 7 Inspection	Explizites Routing ist nicht möglich.

Arista 7150 Switch

Da Arista keinen Controller hat und die Switch OpenFlow noch nicht unterstützen, kann dieser Test nicht durchgeführt werden.

Test	Erwartetes Ergebnis	Ergebnis	Schlussfolgerung
Anlegen eines Flows für HTTP.	Es kann ein Flow für HTTP angelegt werden.	Der Switch unterstützt keine Layer 7 Inspection	Explizites Routing ist nicht möglich.
Der Flow zeigt auf den sekundären Link.	Der HTTP Traffic fliesst auf dem sekundären Link.	Der Switch unterstützt keine Layer 7 Inspection	Explizites Routing ist nicht möglich.

Cisco Nexus 1000V

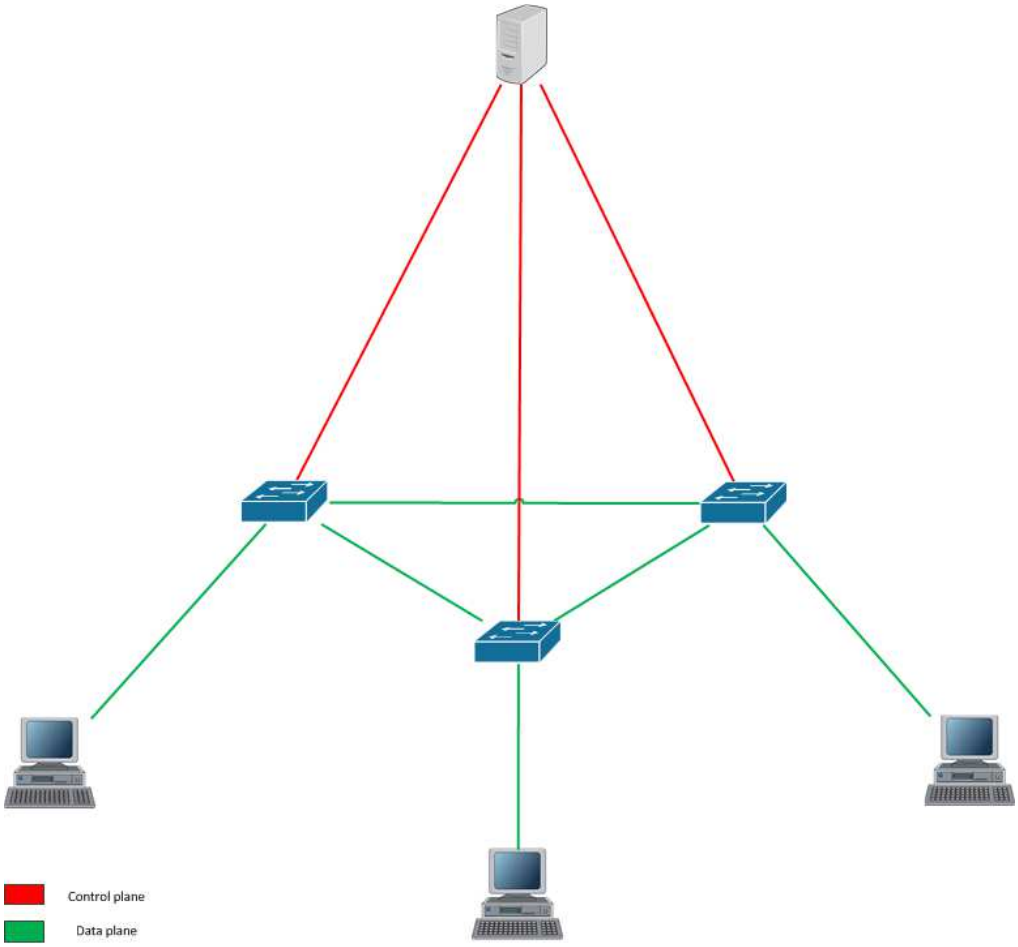
Der Cisco APIC ist noch nicht Verfügbar deshalb wird getestet, welche Möglichkeiten OnePK bietet.

Test	**Erwartetes Ergebnis**	**Ergebnis**	**Schlussfolgerung**
Anlegen eines Flows für HTTP.	Es kann ein Flow für HTTP angelegt werden.	Layer 7 Inspection wird unterstützt. Da jedoch kein Controller existiert, dann dieser Test nicht durchgeführt werden.	Im Moment nicht möglich.
Der Flow zeigt auf den sekundären Link.	Der HTTP Traffic fliesst auf dem sekundären Link.	Layer 7 Inspection wird unterstützt. Da jedoch kein Controller existiert, dann dieser Test nicht durchgeführt werden.	Im Moment nicht möglich.

5.10.8 Virtualisierte Netzwerke

Beim Test der virtualisieren Netzwerke wird versucht, mehrere Netze zu erstellen. Diese befinden sich alle auf derselben Hardware.

5.10.8.1 Testinfrastruktur



5.10.8.2 Test Idee

Das Netzwerk besitzt zwei Datenpfade. Nun werden zwei Netzwerke erstellt. Ein Netz ist für die Kommunikation zwischen zwei PCs. Zum Beispiel zwischen PC1 und PC2 wird ein virtuelles Netz verwendet und zwischen PC1 und PC3 wird ein anderes verwendet.

5.10.8.3 Testdurchführung

HP Controller

Es werden verschiedene Flows erstellt um zwei Netzwerke zu erstellen.

Test	**Erwartetes Ergebnis**	**Ergebnis**	**Schlussfolgerung**
Es können zwei getrennte Netze erstellt werden.	Mit Flows werden zwei Netzwerke erstellt.	Die Flows können auf die Source IP konfiguriert werden.	Virtualisierte Netzwerke können erstellt werden.
Die Flows können in Slices zusammengefasst werden.	Es kann ein Slice erstellt und Flows zugeordnet werden.	Flows können in Gruppen zusammengefasst werden.	Virtualisierte Netzwerke können erstellt werden.

Opendaylight

Der OpenDaylight Controller wird mit mininet getestet. Auf dem Webinterface des Controllers wird die Infrastruktur korrekt angezeigt. Danach wird ein Link unterbrochen.

Test	Erwartetes Ergebnis	Ergebnis	Schlussfolgerung
Es können zwei getrennte Netze erstellt werden.	Mit Flows werden zwei Netzwerke erstellt.	Die Flows können auf die Source IP konfiguriert werden.	Virtualisierte Netzwerke können erstellt werden.
Die Flows können in Slices zusammengefasst werden.	Es kann ein Slice erstellt und Flows zugeordnet werden.	Flows können nicht zusammengefasst werden.	

Arista Controller

Da Arista keinen Controller hat und die Switch OpenFlow noch nicht unterstützen, kann dieser Test nicht durchgeführt werden.

Test	Erwartetes Ergebnis	Ergebnis	Schlussfolgerung
Es können zwei getrennte Netze erstellt werden.	Mit Flows werden zwei Netzwerke erstellt.	n/a	
Die Flows können in Slices zusammengefasst werden.	Es kann ein Slice erstellt und Flows zugeordnet werden.	n/a	

Cisco APIC

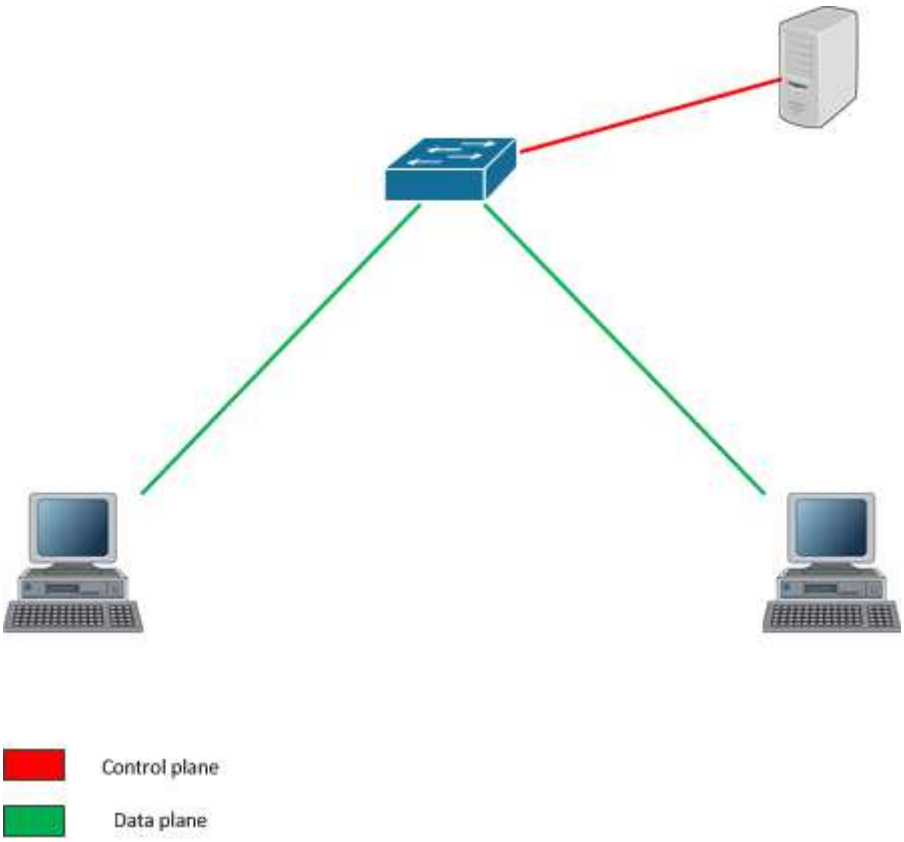
Der Cisco APIC ist noch nicht Verfügbar deshalb wird dieser Test nicht durchgeführt.

Test	Erwartetes Ergebnis	Ergebnis	Schlussfolgerung
Es können zwei getrennte Netze erstellt werden.	Mit Flows werden zwei Netzwerke erstellt.	n/a	
Die Flows können in Slices zusammengefasst werden.	Es kann ein Slice erstellt und Flows zugeordnet werden.	n/a	

5.10.9 Netzwerkzugriffs Kontrolle

Beim Test der Zugriffskontrolle wird überprüft, ob der Switch eine Möglichkeit hat, einen User zu authentisieren.

5.10.9.1 Testinfrastruktur



5.10.9.2 Test Idee

Der Switch wird so eingerichtet, dass sich ein User authentisieren muss, bevor er Daten senden darf.

5.10.9.3 Testdurchführung

HP 5900 Switch

Es werden verschiedene Flows erstellt um zwei Netzwerke zu erstellen.

Test	Erwartetes Ergebnis	Ergebnis	Schlussfolgerung
Es kann ein Authentisierungs Server angegeben werden.	Es kann ein Radius oder Tacacs Server angegeben werden.	Es kann ein Radius Server angegeben werden.	NAC ist möglich.
Ein Switch Port kann so eingestellt werden, dass sich ein Client authentisieren muss.	Der Switch Port kann so eingestellt werden, dass sich der Client über IEEE 802.1X authentisieren muss.	Dot1x kann eingestellt werden.	NAC ist möglich.

OpenVSwitch

Der OpenDaylight Controller wird mit mininet getestet. Auf dem Webinterface des Controllers wird die Infrastruktur korrekt angezeigt. Danach wird ein Link unterbrochen.

Test	**Erwartetes Ergebnis**	**Ergebnis**	**Schlussfolgerung**
Es kann ein Authentisierungs Server angegeben werden.	Es kann ein Radius oder Tacacs Server angegeben werden.	Dot1x wird nicht unterstützt.	NAC ist nicht möglich.
Ein Switch Port kann so eingestellt werden, dass sich ein Client authentisieren muss.	Der Switch Port kann so eingestellt werden, dass sich der Client über IEEE 802.1X authentisieren muss.	Dot1x wird nicht unterstützt.	NAC ist nicht möglich.

Arista 7150 Switch

Da Arista keinen Controller hat und die Switch OpenFlow noch nicht unterstützen, kann dieser Test nicht durchgeführt werden.

Test	Erwartetes Ergebnis	Ergebnis	Schlussfolgerung
Es kann ein Authentisierungs Server angegeben werden.	Es kann ein Radius oder Tacacs Server angegeben werden.	Es kann ein Radius Server angegeben werden.	NAC ist möglich.
Ein Switch Port kann so eingestellt werden, dass sich ein Client authentisieren muss.	Der Switch Port kann so eingestellt werden, dass sich der Client über IEEE 802.1X authentisieren muss.	Dot1x kann eingestellt werden.	NAC ist möglich.

Cisco Nexus 1000V

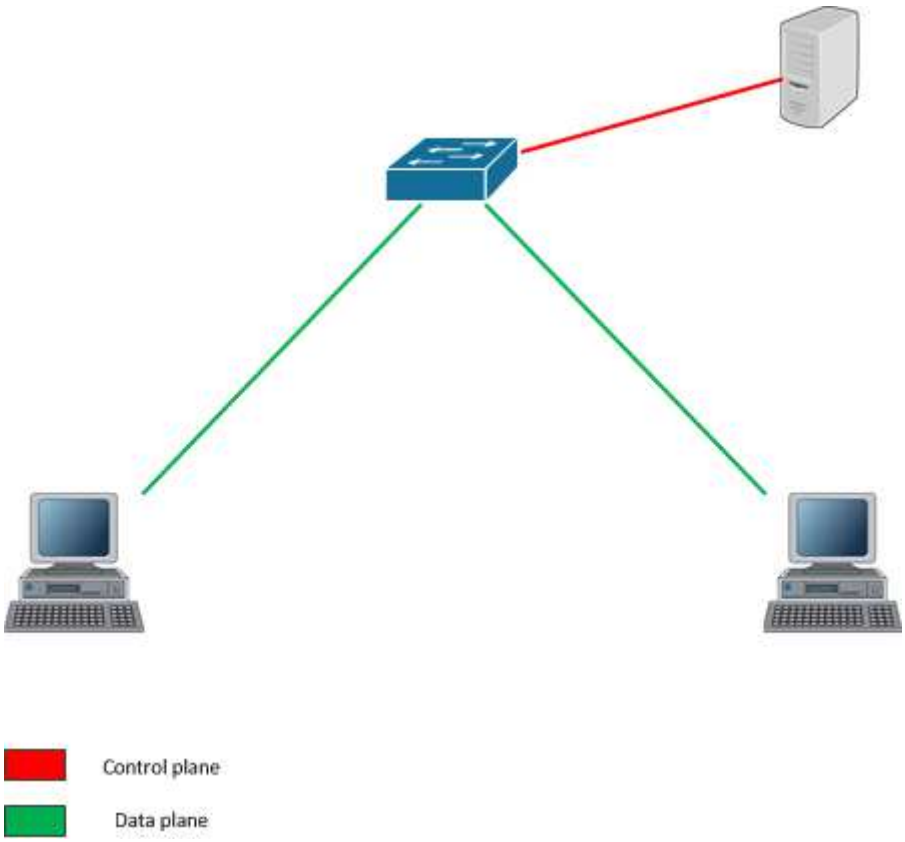
Der Cisco APIC ist noch nicht Verfügbar deshalb wird getestet, welche Möglichkeiten OnePK bietet.

Test	Erwartetes Ergebnis	Ergebnis	Schlussfolgerung
Es kann ein Authentisierungs Server angegeben werden.	Es kann ein Radius oder Tacacs Server angegeben werden.	Es kann ein Radius Server angegeben werden.	NAC ist möglich.
Ein Switch Port kann so eingestellt werden, dass sich ein Client authentisieren muss.	Der Switch Port kann so eingestellt werden, dass sich der Client über IEEE 802.1X authentisieren muss.	Dot1x kann eingestellt werden.	NAC ist möglich.

5.10.10 Überwachung

Beim Test der Überwachung wird überprüft, ob ein Monitoring Tool Daten auslesen kann vom Switch.

5.10.10.1 Testinfrastruktur



5.10.10.2 Test Idee

Der Switch wird so eingerichtet, dass per SNMP, SSH, NetFlow oder XMPP Daten ausgelesen werden kann.

5.10.10.3 Testdurchführung

HP 5900 Switch

Test	Erwartetes Ergebnis	Ergebnis	Schlussfolgerung
Es wird SNMP eingerichtet.	Es kann SNMP v3 eingerichtet werden.	Ja	
Es wird SSH eingerichtet.	Es kann SSH v2 eingerichtet werden.	Ja	
Es wird NetFlow eingerichtet.	Es kann NetFlow eingerichtet werden.	Ja	
Es wird XMPP eingerichtet.	Es kann XMPP eingerichtet werden.	Nein	

OpenVSwitch

Der OpenDaylight Controller wird mit mininet getestet. Auf dem Webinterface des Controllers wird die Infrastruktur korrekt angezeigt. Danach wird ein Link unterbrochen.

Test	Erwartetes Ergebnis	Ergebnis	Schlussfolgerung
Es wird SNMP eingerichtet.	Es kann SNMP v3 eingerichtet werden.	Nein	
Es wird SSH eingerichtet.	Es kann SSH v2 eingerichtet werden.	Nein	
Es wird NetFlow eingerichtet.	Es kann NetFlow eingerichtet werden.	Ja	
Es wird XMPP eingerichtet.	Es kann XMPP eingerichtet werden.	Nein	

Arista 7150 Switch

Da Arista keinen Controller hat und die Switch OpenFlow noch nicht unterstützen, kann dieser Test nicht durchgeführt werden.

Test	Erwartetes Ergebnis	Ergebnis	Schlussfolgerung
Es wird SNMP eingerichtet.	Es kann SNMP v3 eingerichtet werden.	Ja	
Es wird SSH eingerichtet.	Es kann SSH v2 eingerichtet werden.	Ja	
Es wird NetFlow eingerichtet.	Es kann NetFlow eingerichtet werden.	Ja	
Es wird XMPP eingerichtet.	Es kann XMPP eingerichtet werden.	Ja	

Cisco Nexus 1000V

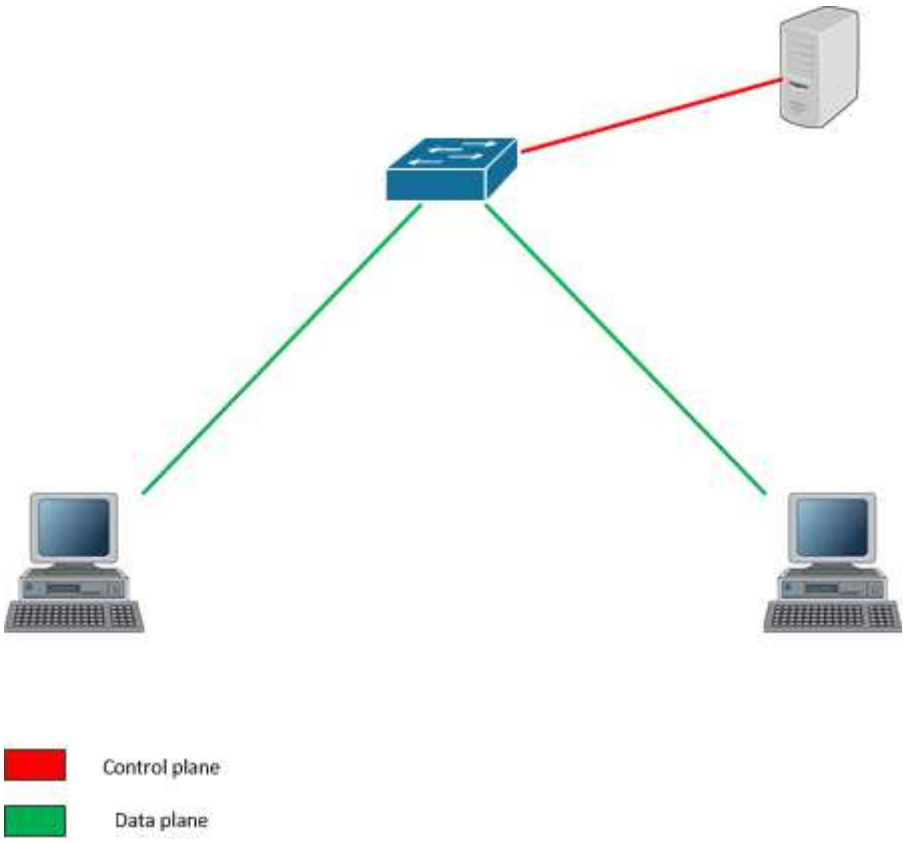
Der Cisco APIC ist noch nicht Verfügbar deshalb wird getestet, welche Möglichkeiten OnePK bietet.

Test	Erwartetes Ergebnis	Ergebnis	Schlussfolgerung
Es wird SNMP eingerichtet.	Es kann SNMP v3 eingerichtet werden.	Ja	
Es wird SSH eingerichtet.	Es kann SSH v2 eingerichtet werden.	Ja	
Es wird NetFlow eingerichtet.	Es kann NetFlow eingerichtet werden.	Ja	
Es wird XMPP eingerichtet.	Es kann XMPP eingerichtet werden.	Nein	

5.10.11 Priorisierung von Daten

Beim Test der Priorisierung wird getestet, ob ein QoS Label gesetzt werden kann.

5.10.11.1 Testinfrastruktur



5.10.11.2 Test Idee

Der Switch wird so eingerichtet, dass für HTTP Traffic einen höheren QoS Wert gesetzt wird.

5.10.11.3 Testdurchführung

HP 5900 Switch

Test	Erwartetes Ergebnis	Ergebnis	Schlussfolgerung
Anpassen des QoS Wertes.	Der QoS Wert kann angepasst werden.	QoS wird unterstützt.	QoS kann im Netzwerk verwendet werden.
Anpassen des QoS für bestimmte Daten.	Der QoS Wert kann für HTTP Traffic gesetzt werden.	Ja. QoS wird anhand von ACLs gesetzt.	
Anpassen des QoS für einen Layer 4 Port.	Der QoS Wert kann für den Port 80 gesetzt werden.	Ja. QoS wird anhand von ACLs gesetzt.	
Anpassen des QoS für einen Switch Port	Der QoS Wert kann für den Switch Port 1 gesetzt werden.	Ja mit OpenFlwo kann der DSCP Wert gesetzt werden.	

OpenVSwitch

Der OpenDaylight Controller wird mit mininet getestet. Auf dem Webinterface des Controllers wird die Infrastruktur korrekt angezeigt. Danach wird ein Link unterbrochen.

Test	Erwartetes Ergebnis	Ergebnis	Schlussfolgerung
Anpassen des QoS Wertes.	Der QoS Wert kann angepasst werden.	Wird nicht unterstützt	
Anpassen des QoS für bestimmte Daten.	Der QoS Wert kann für HTTP Traffic gesetzt werden.	Wird nicht unterstützt	
Anpassen des QoS für einen Layer 4 Port.	Der QoS Wert kann für den Port 80 gesetzt werden.	Wird nicht unterstützt	
Anpassen des QoS für einen Switch Port	Der QoS Wert kann für den Switch Port 1 gesetzt werden.	Wird nicht unterstützt	

Arista 7150 Switch

Da Arista keinen Controller hat und die Switch OpenFlow noch nicht unterstützen, kann dieser Test nicht durchgeführt werden.

Test	Erwartetes Ergebnis	Ergebnis	Schlussfolgerung
Anpassen des QoS Wertes.	Der QoS Wert kann angepasst werden.	QoS wird unterstützt.	
Anpassen des QoS für bestimmte Daten.	Der QoS Wert kann für HTTP Traffic gesetzt werden.	Layer 7 QoS ist nicht möglich.	
Anpassen des QoS für einen Layer 4 Port.	Der QoS Wert kann für den Port 80 gesetzt werden.	Layer 4 QoS ist möglich.	
Anpassen des QoS für einen Switch Port	Der QoS Wert kann für den Switch Port 1 gesetzt werden.	QoS kann pro Port definiert werden.	

Cisco Nexus 1000V

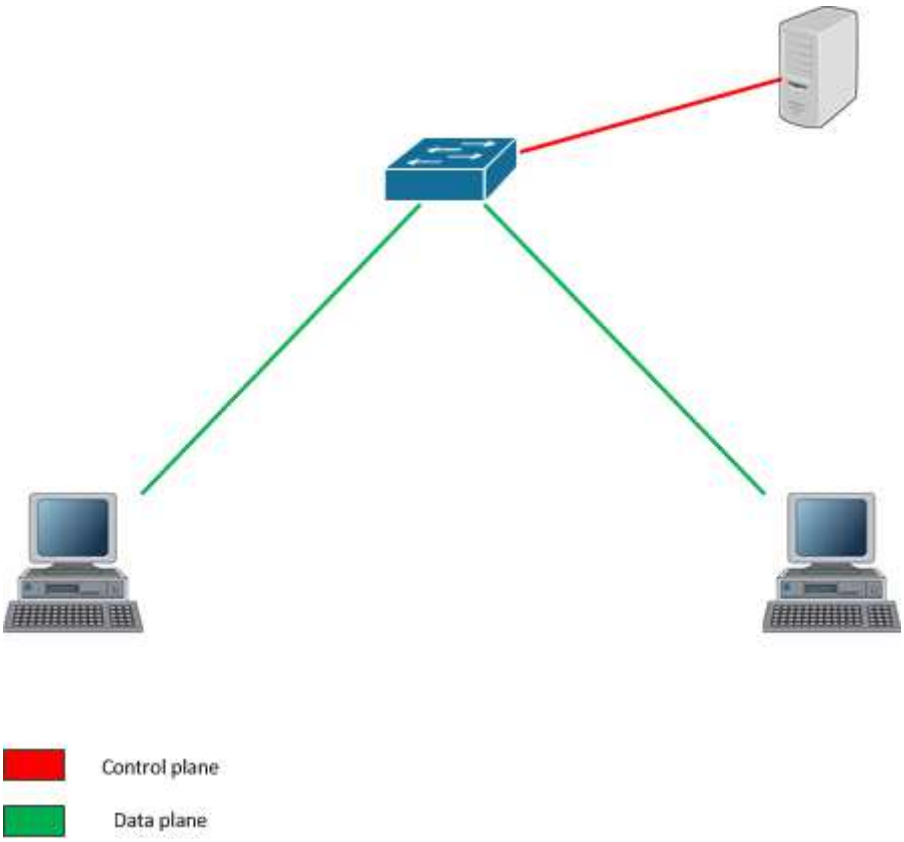
Der Cisco APIC ist noch nicht Verfügbar deshalb wird getestet, welche Möglichkeiten OnePK bietet.

Test	**Erwartetes Ergebnis**	**Ergebnis**	**Schlussfolgerung**
Anpassen des QoS Wertes.	Der QoS Wert kann angepasst werden.	Wird unterstützt	
Anpassen des QoS für bestimmte Daten.	Der QoS Wert kann für HTTP Traffic gesetzt werden.	Wird unterstützt	
Anpassen des QoS für einen Layer 4 Port.	Der QoS Wert kann für den Port 80 gesetzt werden.	Wird unterstützt	
Anpassen des QoS für einen Switch Port	Der QoS Wert kann für den Switch Port 1 gesetzt werden.	Wird unterstützt	

5.10.12 Sicherheit

Beim Test der Sicherheit wird überprüft, ob ein Flow erstellt werden kann, der bestimmte Daten blockiert.

5.10.12.1 Testinfrastruktur



5.10.12.2 Test Idee

Der Switch wird so eingerichtet, dass für HTTP Traffic blockiert wird.

5.10.12.3 Testdurchführung

HP Controller

Test	Erwartetes Ergebnis	Ergebnis	Schlussfolgerung
Es wird ein Flow erstellt, der http Daten blockiert.	http Daten werden blockiert.	Es kann kein Flow erstellt werden mit der Aktion Drop	Daten können nicht blockiert werden.

Opendaylight

Der OpenDaylight Controller wird mit mininet getestet. Auf dem Webinterface des Controllers wird die Infrastruktur korrekt angezeigt. Danach wird ein Link unterbrochen.

Test	Erwartetes Ergebnis	Ergebnis	Schlussfolgerung
Es wird ein Flow erstellt, der http Daten blockiert.	http Daten werden blockiert.	Es kann ein Flow erstellt werden, mit dem Destination Port 80. http kann nicht ausgewählt werden.	Bestimmte Daten können blockiert werden.

Arista Controller

Da Arista keinen Controller hat und die Switch OpenFlow noch nicht unterstützen, kann dieser Test nicht durchgeführt werden.

Test	Erwartetes Ergebnis	Ergebnis	Schlussfolgerung
Es wird ein Flow erstellt, der http Daten blockiert.	http Daten werden blockiert.	n/a	

Cisco APIC

Der Cisco APIC ist noch nicht Verfügbar deshalb wird getestet, welche Möglichkeiten OnePK bietet.

Test	Erwartetes Ergebnis	Ergebnis	Schlussfolgerung
Es wird ein Flow erstellt, der http Daten blockiert.	http Daten werden blockiert.	n/a	

5.10.13 Reaktion der Controller

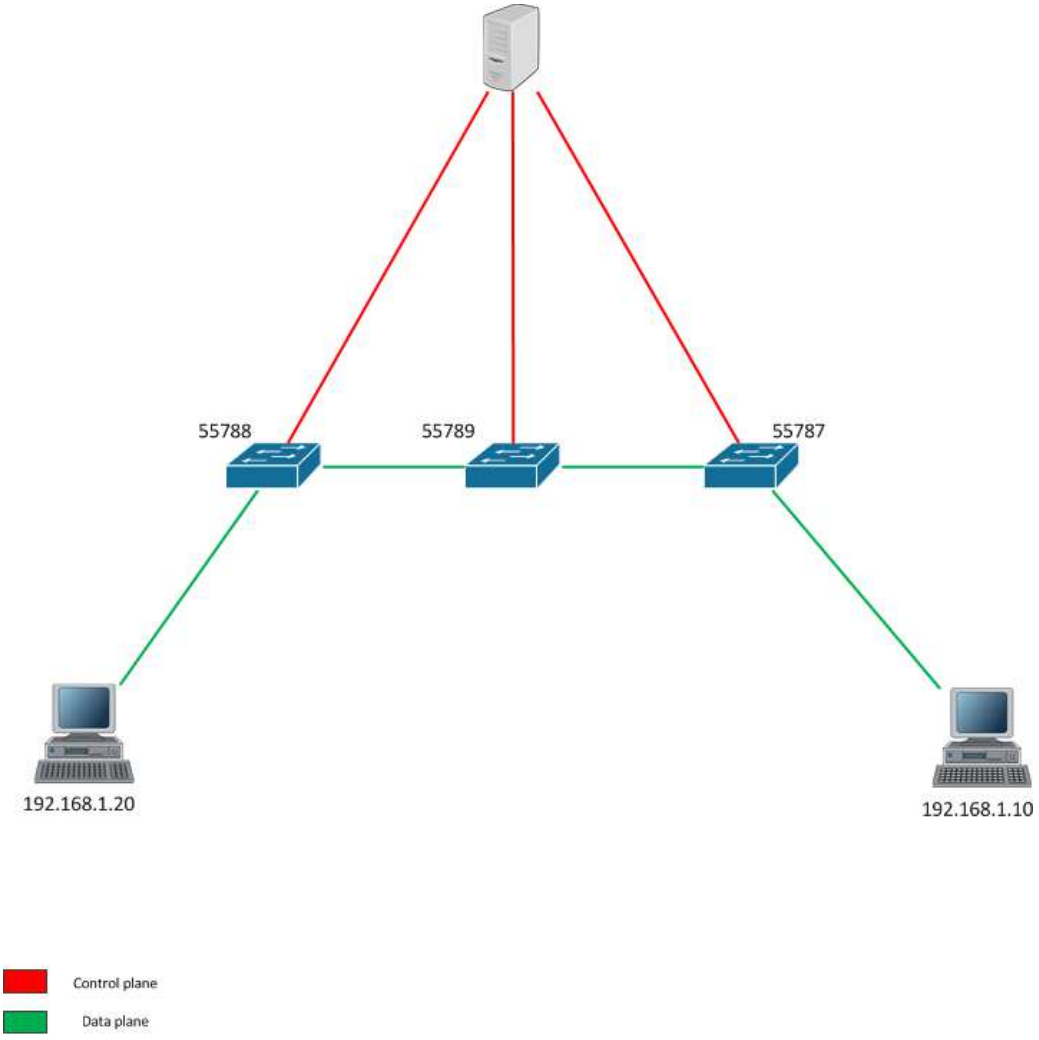
5.10.13.1 Test Idee

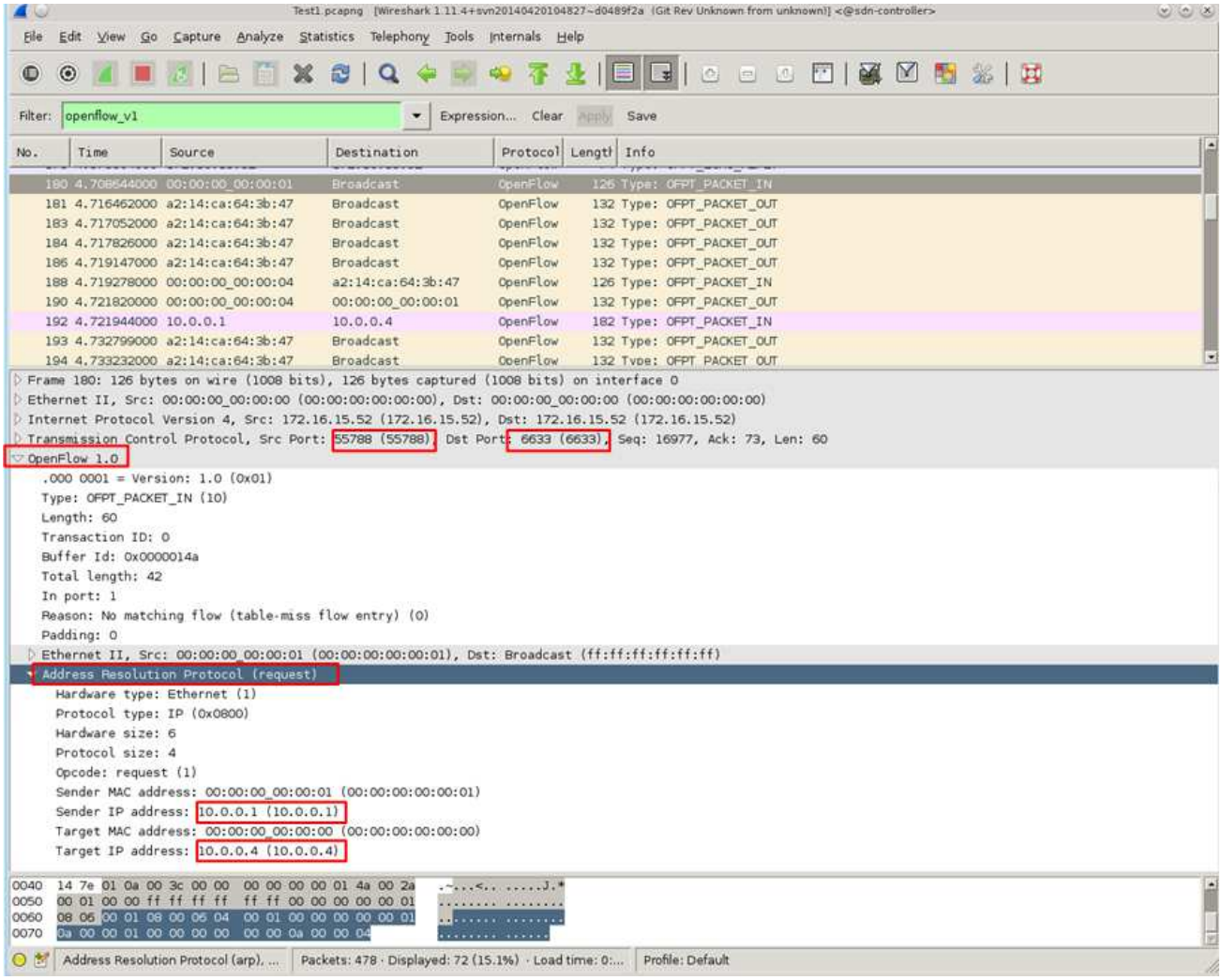
Bei diesem Test soll die Frage geklärt werden, wie der Controller auf eine Flow Anfrage reagiert. Es gibt zwei Möglichkeiten. Wenn der erste Switch einen Flow anfragt, kann der Controller auf allen drei Switch gleich einen Flow anlegen. Bei der zweiten Möglichkeit muss jeder Switch einzeln den Flow anfragen.

5.10.13.2 Testdurchführung

Opendaylight

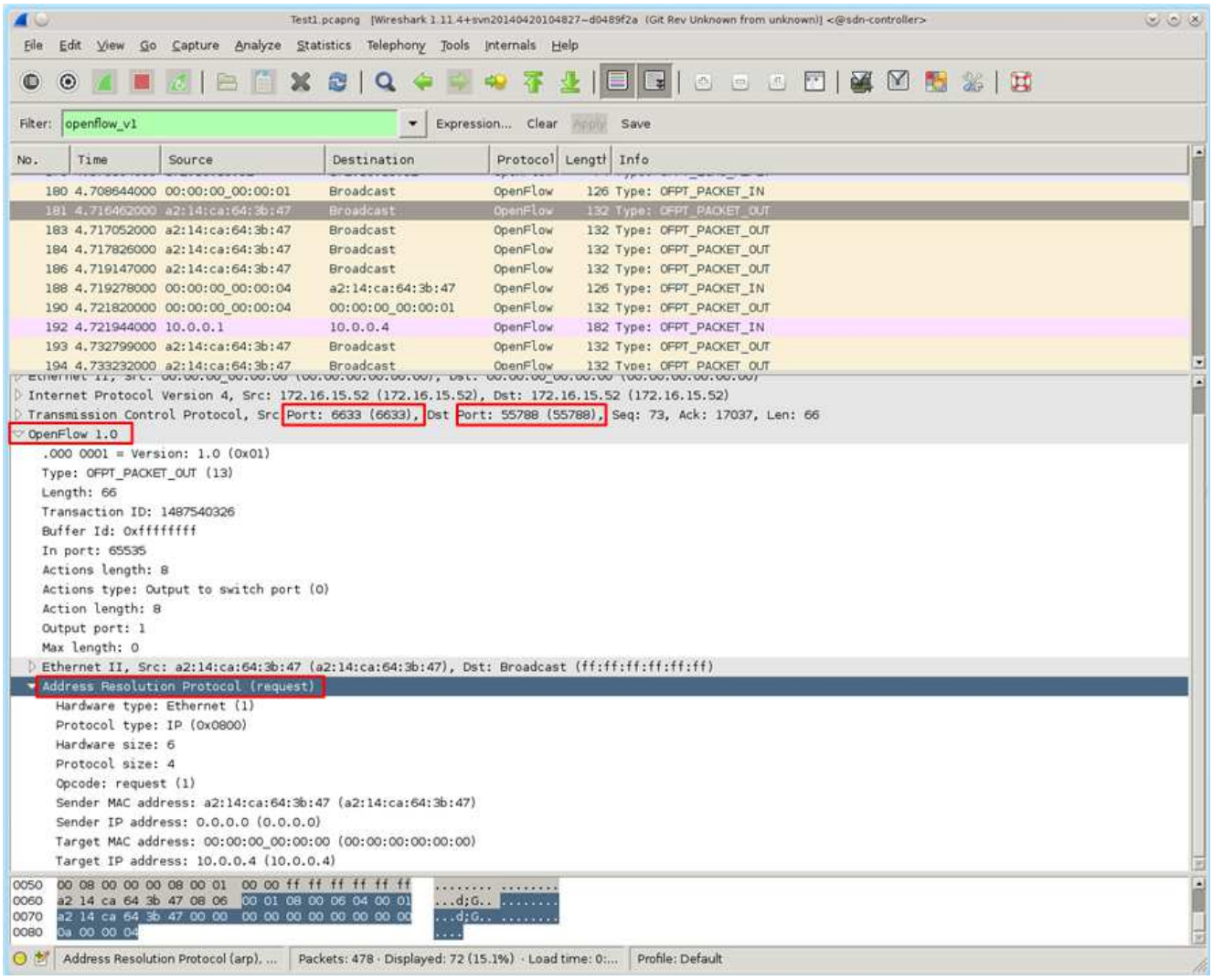
Der erste Test wurde mit dem Opensource Controller OpenDaylight durchgeführt. Dabei wurden mit der Testumgebung mininet drei Switch und zwei Hosts simuliert. Danach wurde auf dem OpenDaylight Controller nach OpenFlow Paketen gesnifft. Nun wurde ein Ping von Host1 zu Host2 ausgeführt.


55788
55789
55787
192.168.1.20
192.168.1.10
Control plane
Data plane

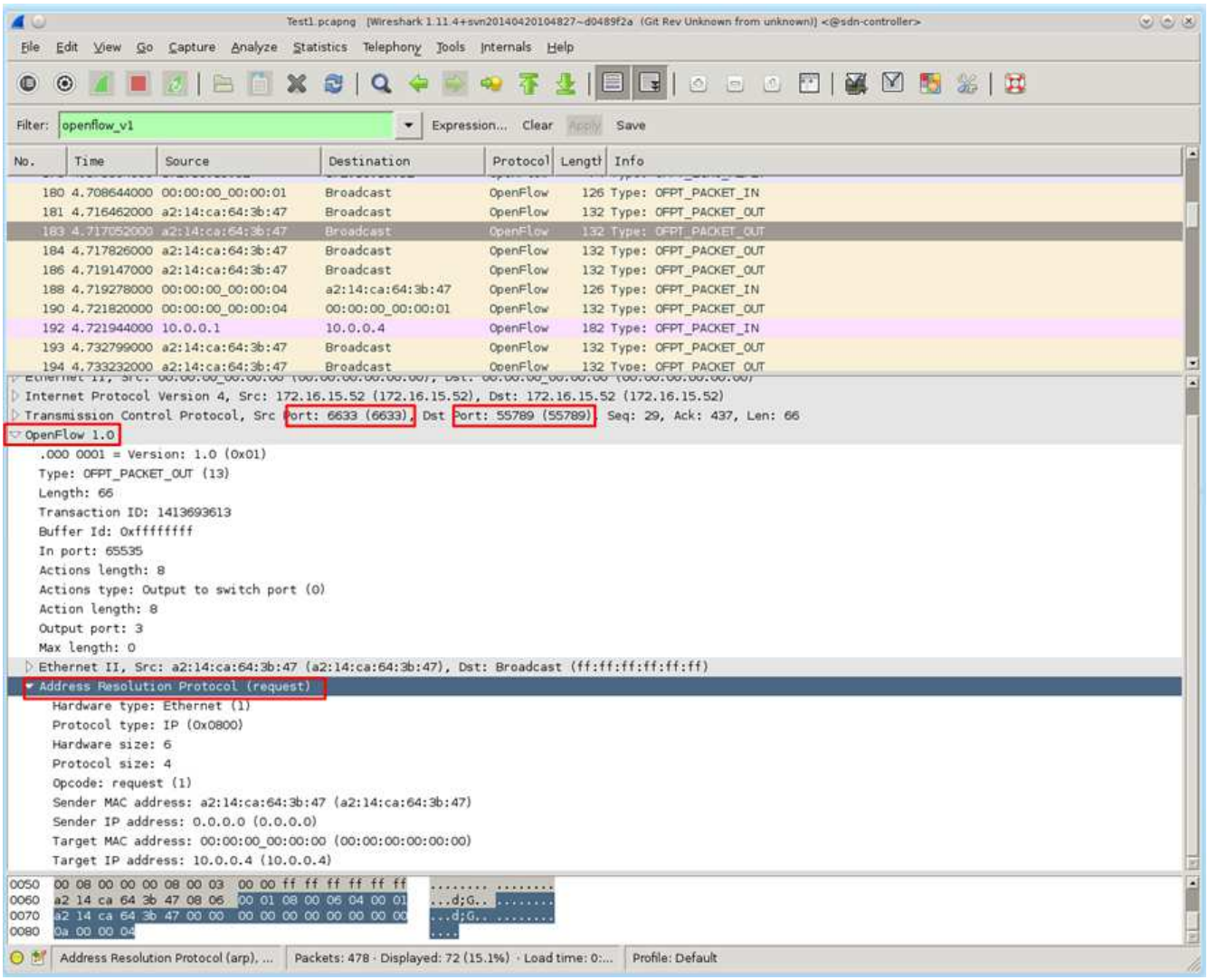


Das erste Paket wird vom Switch1 (Port 55788) zum Controller (Port 6633) gesendet. Dabei handelt es sich um ein OpenFlow 1.0 Paket. Als Payload ist ein ARP Request enthalten. Der ARP Request wurde vom Host1 gesendet, um die MAC Adresse von Host2 ausfindig zu machen.

Der Switch hat jedoch keinen Eintrag in der Flowtable und fragt den Controller an.

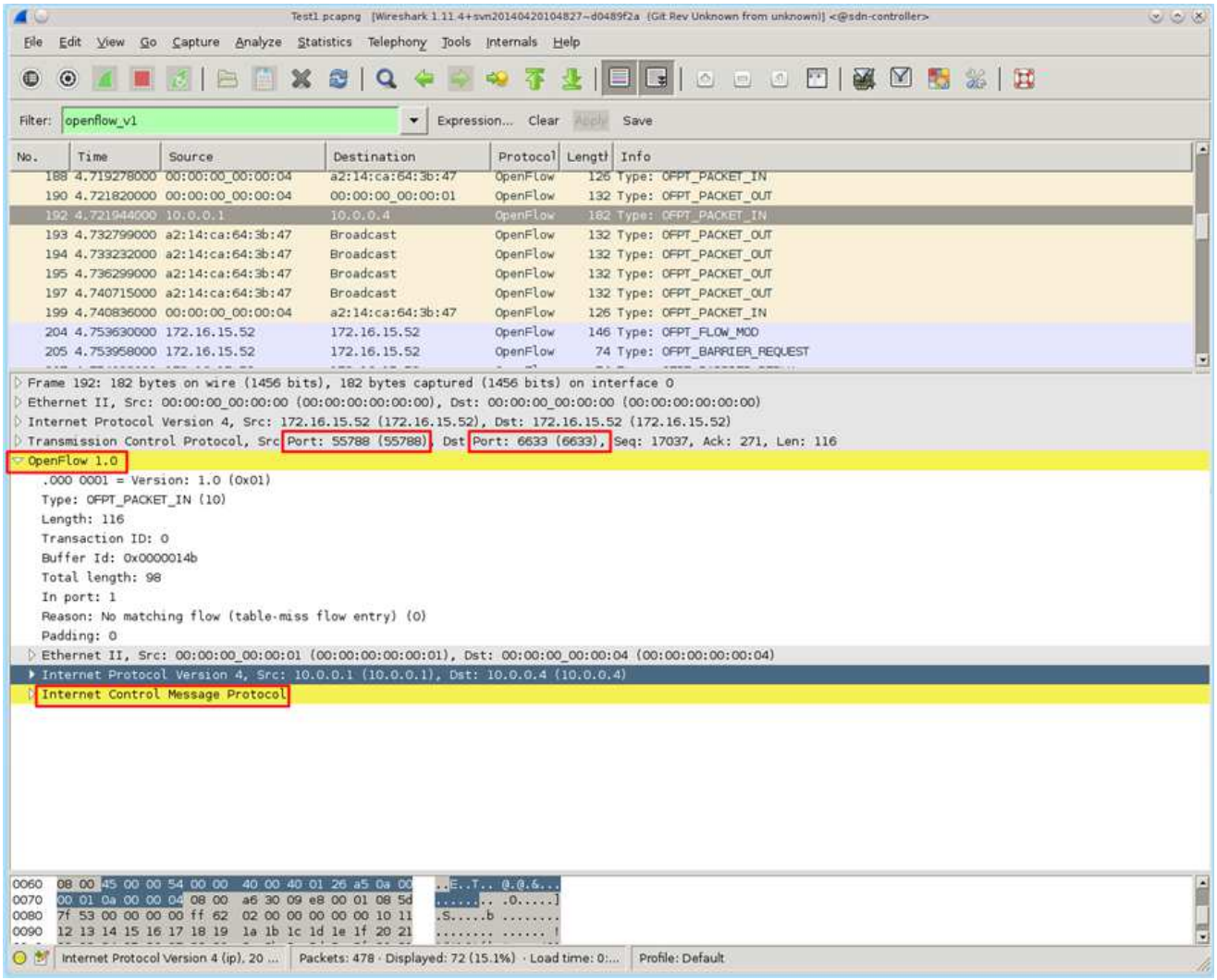


Der Controller (Port 6633) antwortet mit einem OpenFlow Paket, das dem Switch mitteilt, er solle den ARP Request auf Port 1 weiterleiten.

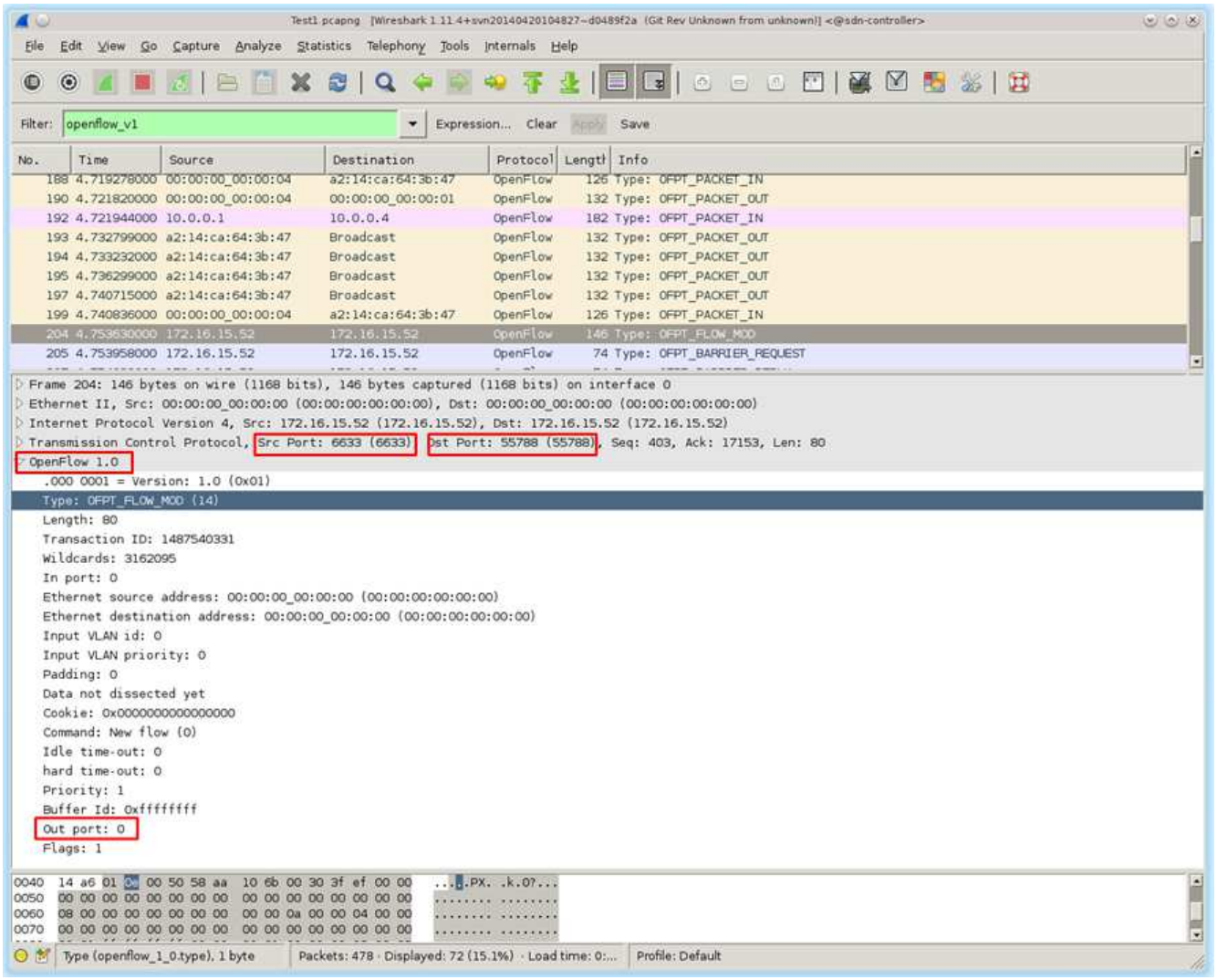


Dasselbe Paket schickt der Controller auch an den Switch2 (Port 55789). So muss dieser nicht selbst eine Anfrage senden.

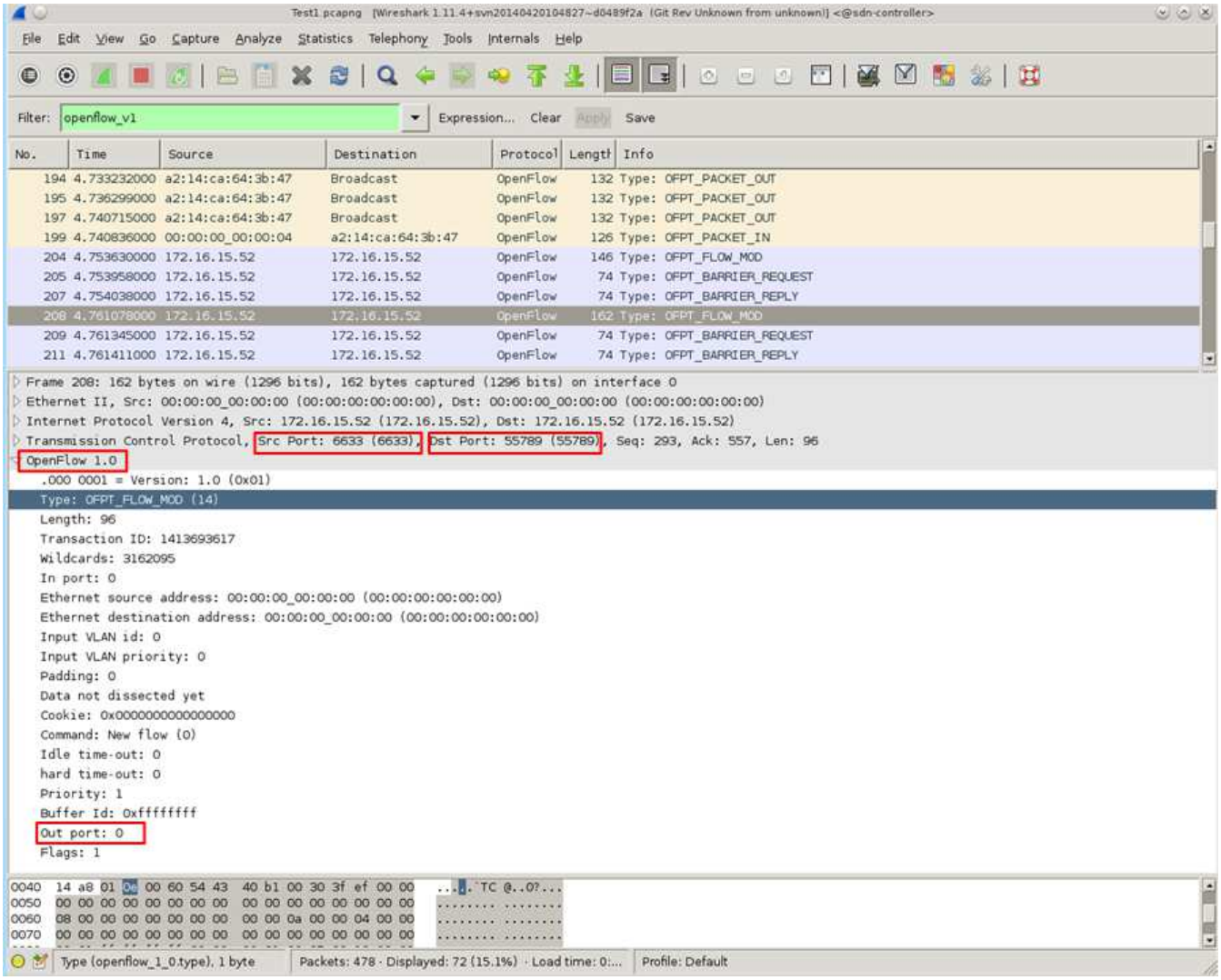
Der Rückweg, mit dem ARP Reply, sieht gleich aus. Der Switch3 macht eine Anfrage und der Controller teilt allen drei Switch mit, wie das Paket zu behandeln ist.



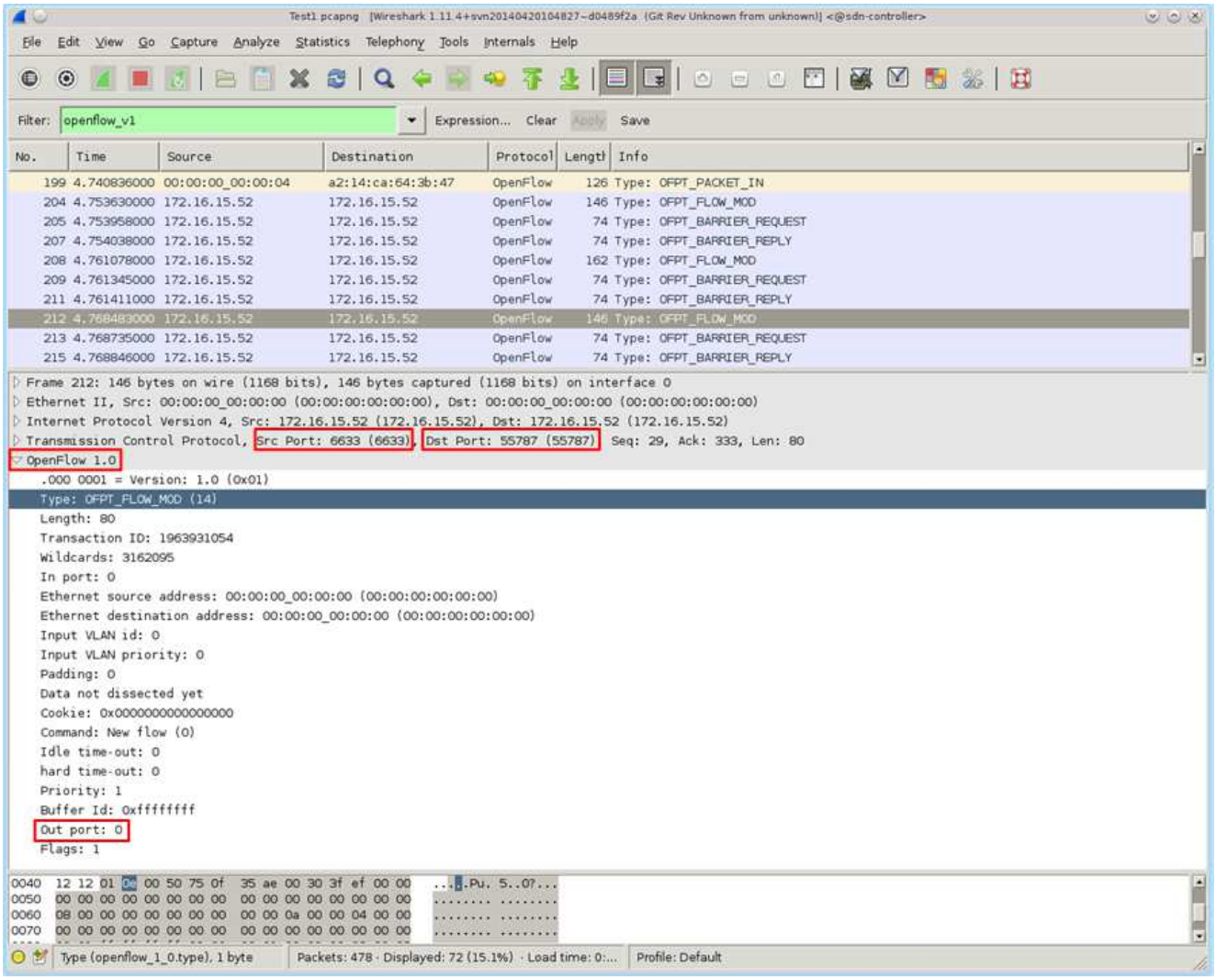
Als nächstes taucht der ICMP Echo Request auf. Dieser wird von Host1 an Host2 gesendet. Da der Switch1 nur einen Floweintrag für ARP hat, muss er den Controller nachfragen, was mit einem ICMP Paket geschehen soll.



Der Controller teilt dem Switch1 mit, er soll den ICMP Echo Request auf Port 0 weiterleiten.



Dasselbe Paket schickt der Controller auch an den Switch2...



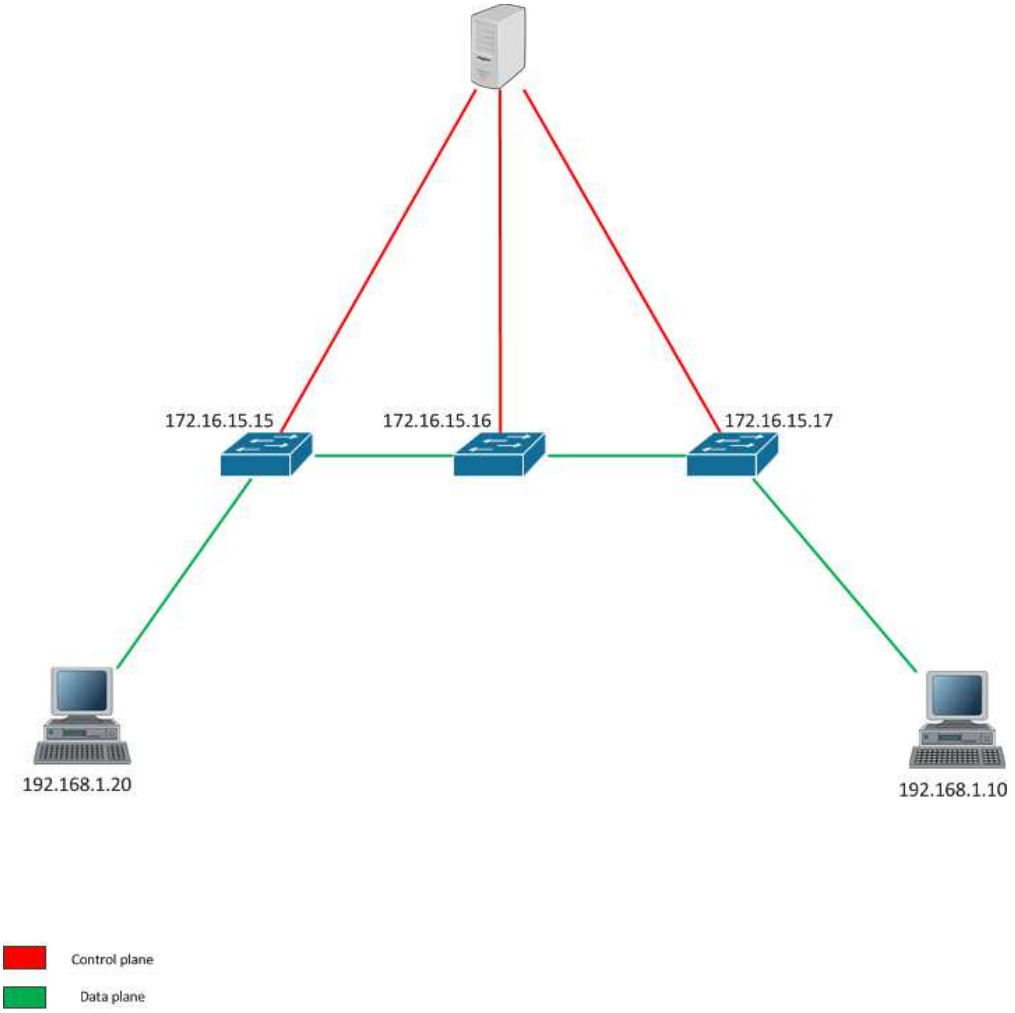
...und an Switch3.

Arista 7150 Switch

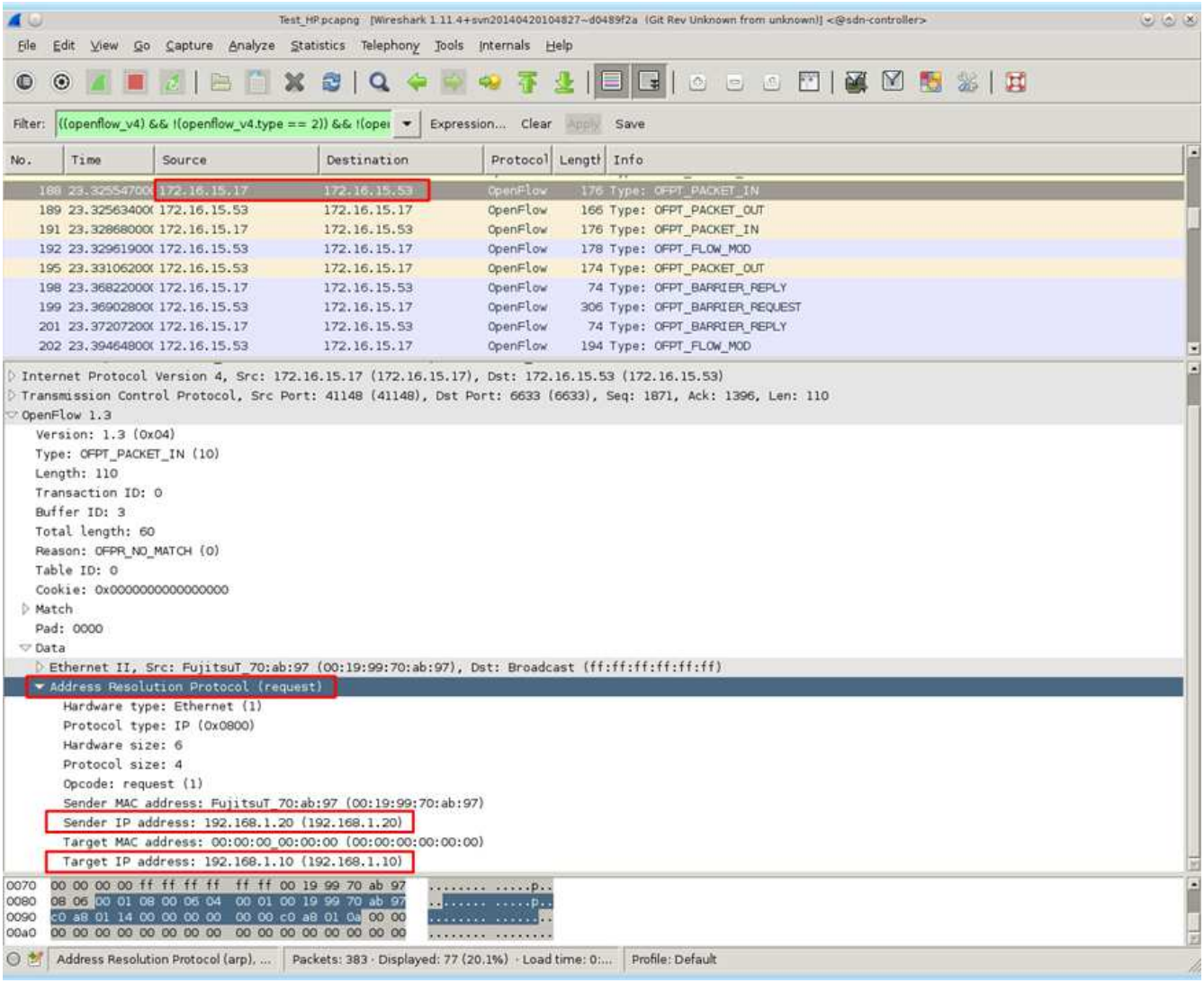
Arista produziert keinen eigenen SDN Controller. Da die Switch von Arista jedoch OpenFlow fähig sind, kann ein Opensource Controller eingesetzt werden. Der vorhergehende Test vom OpenDaylight Controller ist deshalb auch auf Arista Switch anwendbar.

HP 5900 Switch

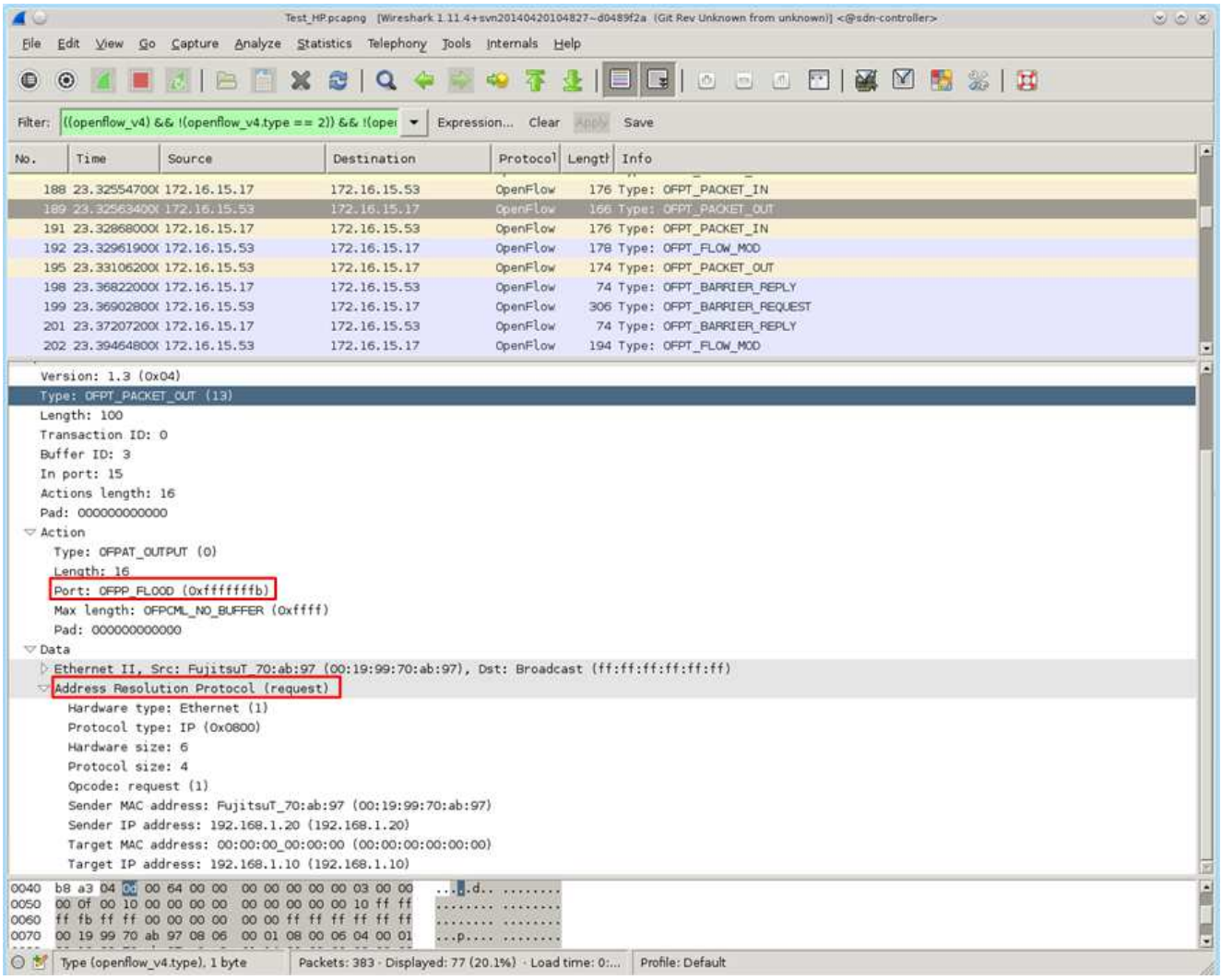
Der zweite Test wurde mit einer Umgebung von HP durchgeführt. Dabei wurden drei Switch aus der 5900er Serie und der HP VAN SDN Controller verwendet. Für den Test wurden die drei Switch in Serie geschalten und mit dem Controller verbunden.



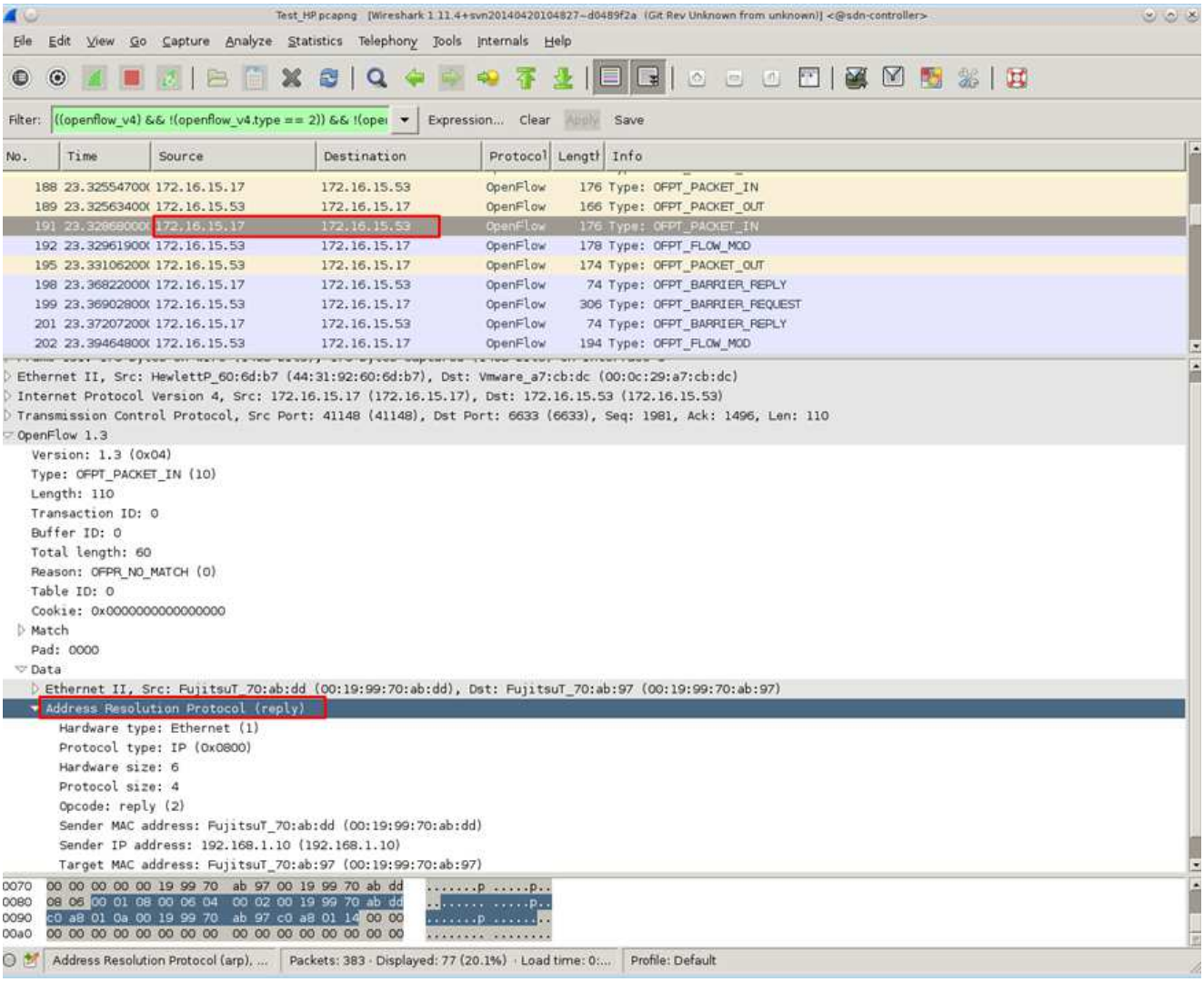
Auch bei diesem Test wurde ein Ping von Host1 (192.168.1.20) zu Host2 (192.168.1.10) ausgeführt. Dabei wurde am Management Interface von Switch3 die Kommunikation mit dem Controller mitgeschnitten.



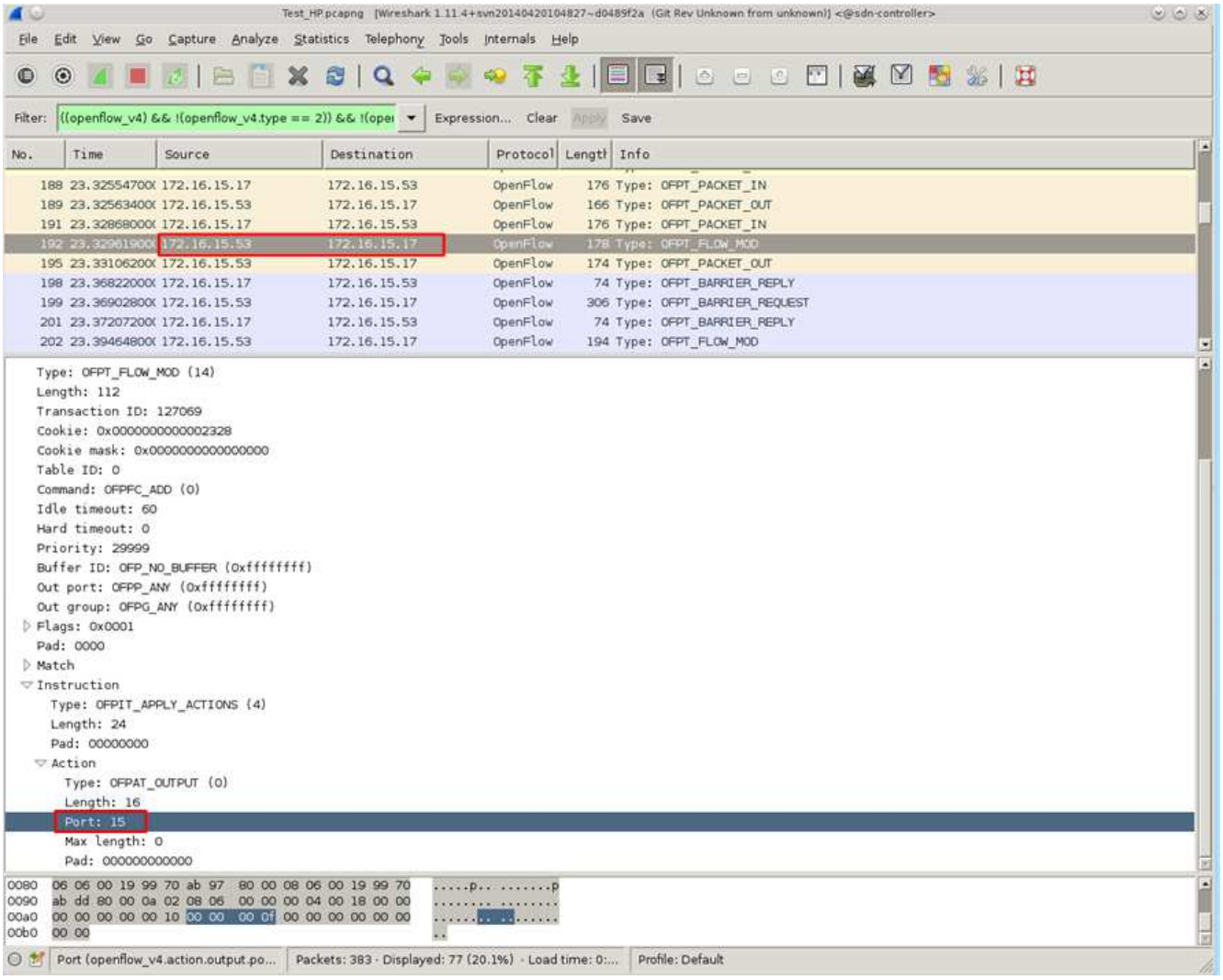
Hier wird ein ARP Request dargestellt. Dieser wurde von Host1 gesendet, um die MAC Adresse von Host2 anzufragen. Da der Switch3 der Letzte der drei Switch ist, kann daraus geschlossen werden, dass jeder Switch einzeln die Flows anfragen muss.



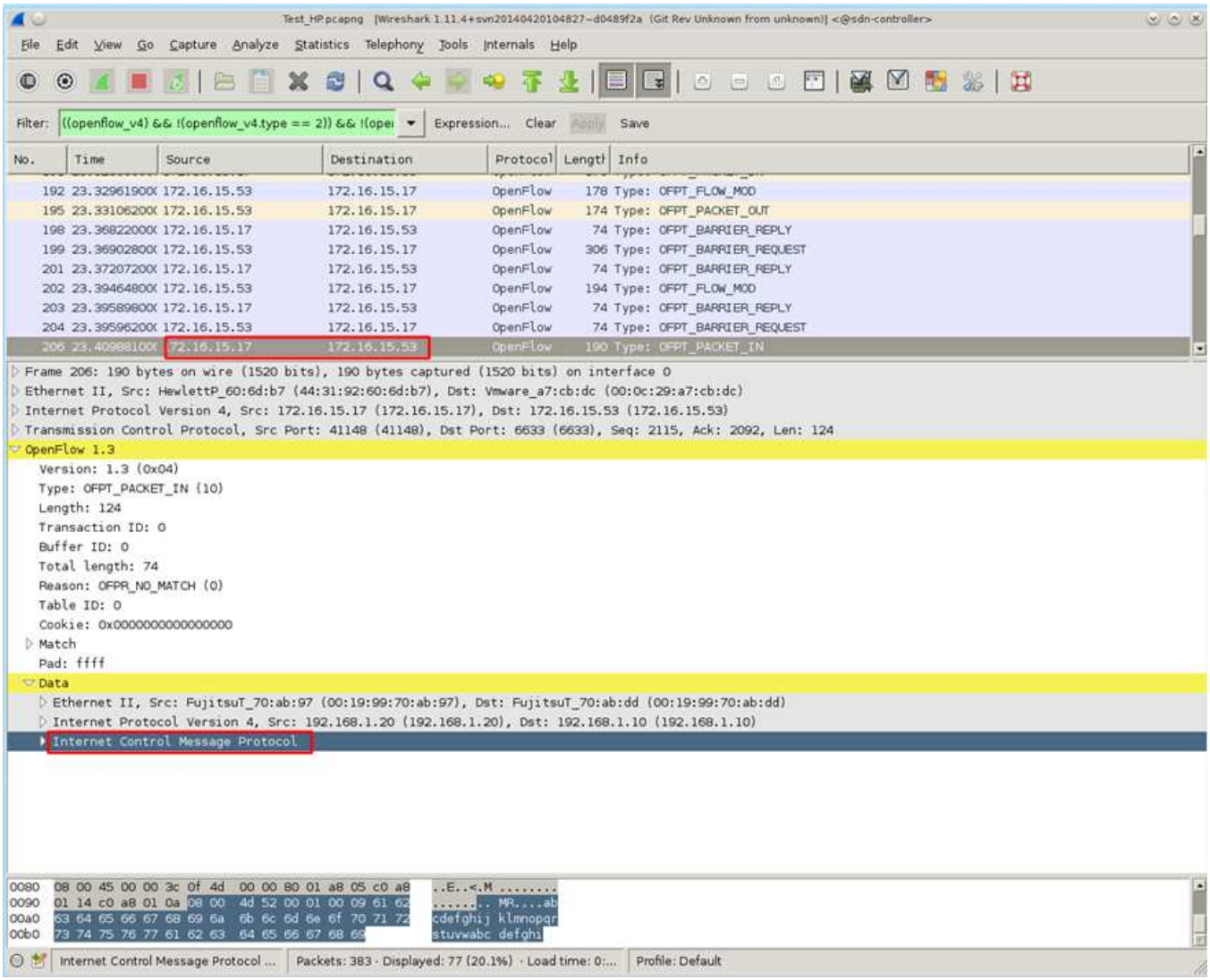
Die Antwort vom Controller bedeutet, dass der ARP Request auf alle Ports geschickt werden soll.



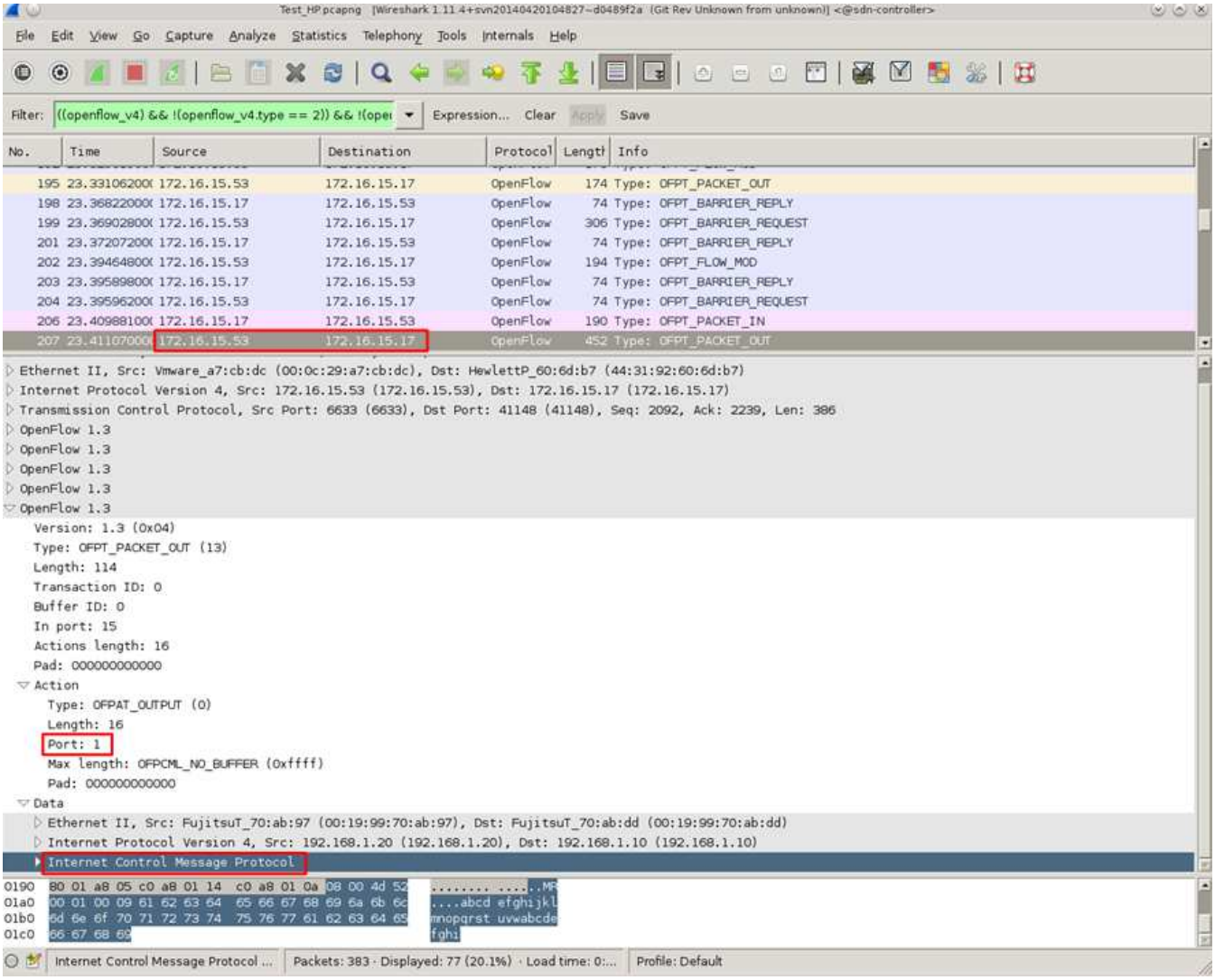
Beim ARP Reply, das von Host2 zu Host1 gesendet wird, muss der Switch3 auch wieder den Controller anfragen.



Dieses Mal „weiss“ der Controller, wo sich der Host1 befindet. Der ARP Reply muss nicht mehr auf alle Ports rausgeschickt werden, sondern nur noch an den Port 15. Dies ist die Verbindung zwischen Switch3 und Switch2.



Hier wird noch die Anfrage für das ICMP Echo Request Paket an den Controller gesendet.



Das ICMP Echo Request Paket soll an den Port 1 geschickt werden. An diesem befindet sich der Host2. Für das ICMP Echo Reply Paket muss auch wieder der Controller angefragt werden. Dies sieht aus wie beim ARP Reply und wird deshalb hier nicht weiter beschrieben.

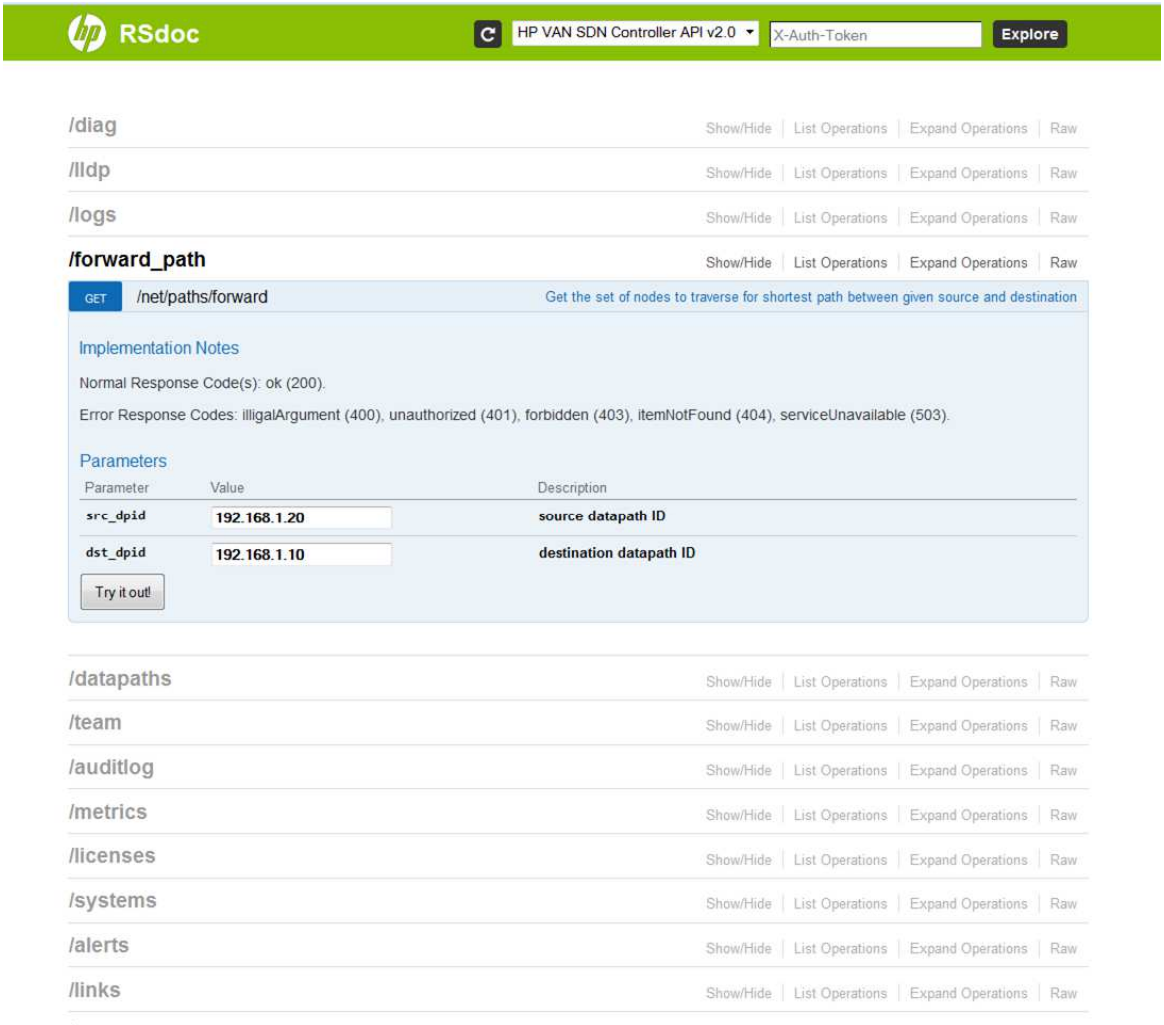
5.11 Vergleich von Hersteller

5.11.1 HP

Die Switch von HP können mittels OpenFlow 1.3 mit Flows versorgt werden. Dies ermöglicht es dem Controller ein ganzes Netz zu kontrollieren.

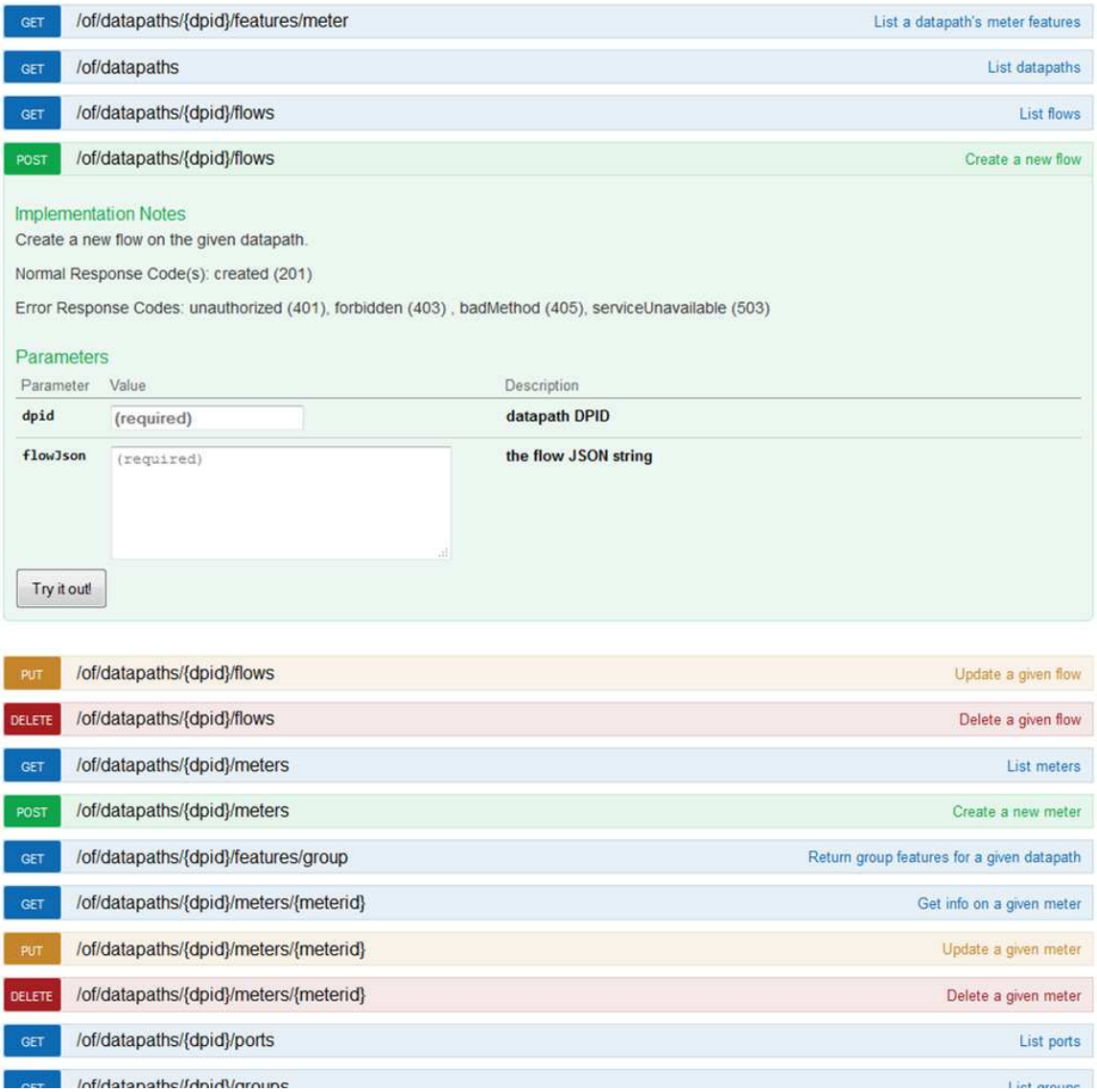
Die einzelnen Switch können mit mehreren OpenFlow Instanzen konfiguriert werden. Eine Instanz ist wie ein virtueller Switch. Ihr werden ein oder mehrere VLANs zugeordnet. So kann ein Netzwerk sehr flexibel verwaltet werden. Diese Konfiguration muss jedoch auf dem Switch erledigt werden. Es ist nicht möglich dies via Controller zu konfigurieren. Auf dem Controller können noch Apps installiert werden. Dadurch ist es möglich, weitere Funktionen hinzuzufügen. Diese Apps können auch wieder deinstalliert oder auch deaktiviert werden.

Der Controller besitzt ein REST API. Darüber können JSON Objekte geschickt werden um den Controller Befehle ausführen zu lassen. Nachstehend sind ein paar Beispiele abgebildet.



REST API vom HP Controller

Hier kann der Controller nach dem kürzesten Weg zwischen zwei Hosts gefragt werden. Das GUI erstellt dann ein JSON Objekt und zeigt die Antwort vom Controller an. Ein JSON Objekt kann natürlich direkt an den Controller gesendet werden. Dies muss nicht manuell über das GUI gemacht werden. So kann ein Orchestrar mehrere Controller verwalten.



REST API vom HP Controller

Natürlich können auch einzelne Flows abgefragt und erstellt werden. Eine weitere Möglichkeit, die das REST API bietet, ist das erstellen, löschen und ändern von Meter bands.

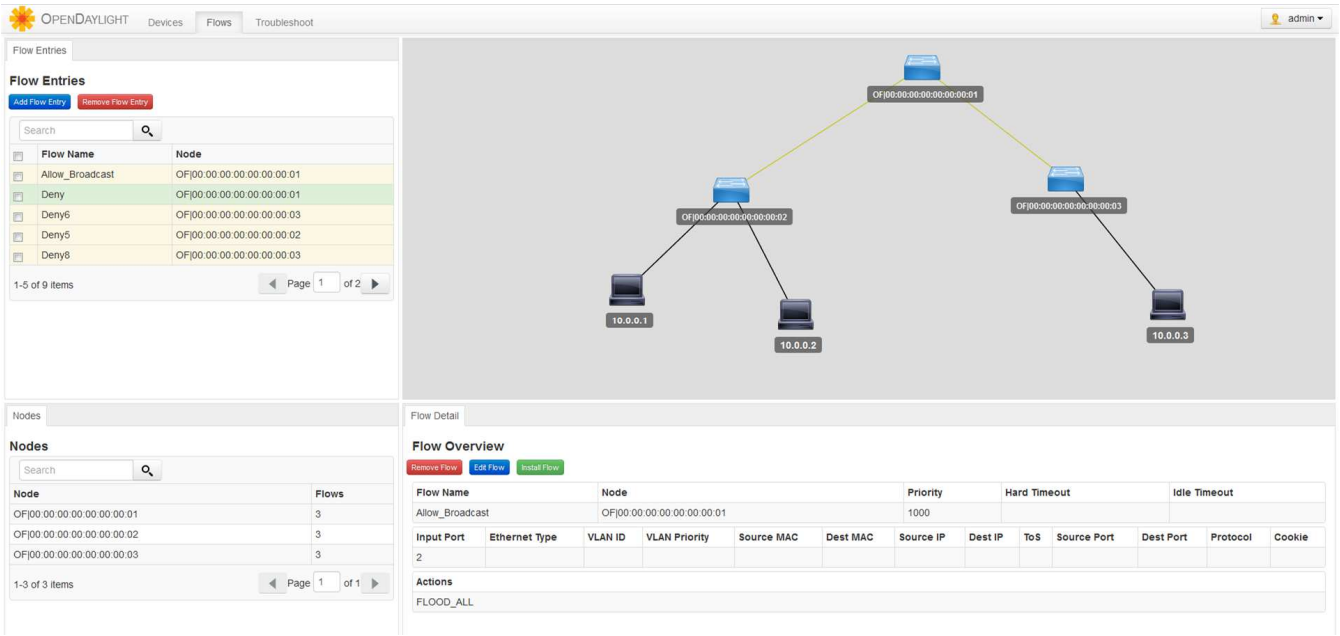
HP besitzt noch keinen Orchestrar, der einen ganzen Service im Netzwerk konfigurieren kann. Dank des REST API ist es jedoch möglich, einen Orchestrar zu entwickeln für den HP SDN Controller. So ist HP sehr flexibel und bereit, in der Zukunft von SDN mitzuwirken.

5.11.2 Arista

Die Switch von Arista bieten noch keine Möglichkeit, um per OpenFlow einen Pfad zu konfigurieren. Dies wird voraussichtlich im vierten Quartal 2014 möglich sein. Es besteht jedoch die Möglichkeit, mehrere Switch gleichzeitig, mit XMPP, zu konfigurieren. So können CLI Befehle an mehrere Switch gleichzeitig gesendet werden. So könnte ein Orchestrar einen Dienst in einem Netzwerk vorbereiten und danach mit OpenFlow zur Verfügung stellen. Arista vertritt jedoch die Meinung, dass sich OpenFlow nie durchsetzen wird. Deshalb wird auch kein eigener Controller entwickelt. OpenFlow wird nur auf den Switch unterstützt, dass ein Controller von einem Drittanbieter oder sogar ein Opensource Controller eingesetzt werden kann.

5.11.3 OpenDaylight

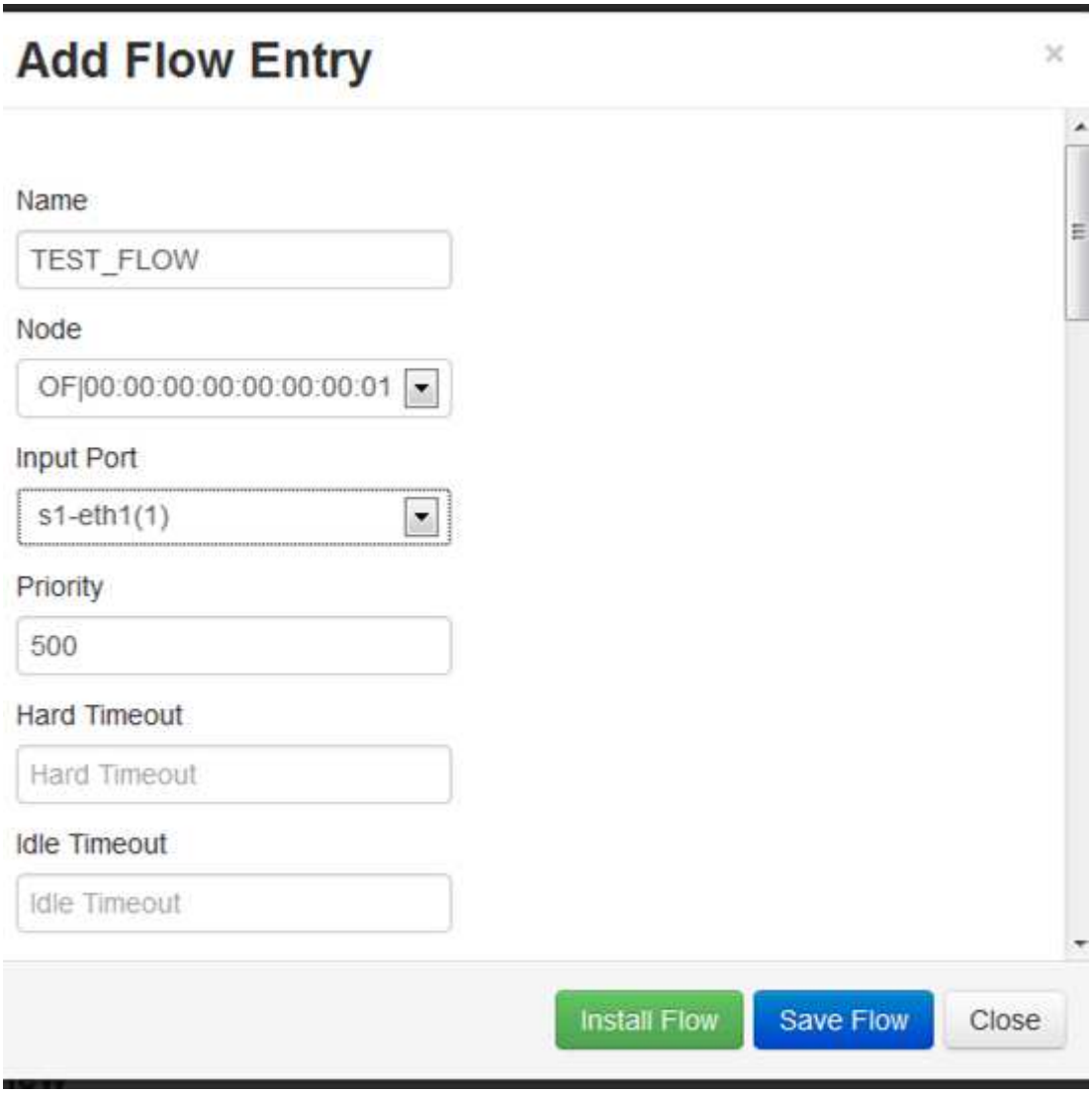
Der OpenDaylight Controller ist eine Opensource Software. Mit ihm kann die Netzwerktopologie angezeigt werden und Flows auf einzelne Switch konfiguriert werden. Dies wird mit OpenFlow gelöst. Weiter verfügt der Controller noch über ein REST API. Damit kann ein Orchestrar ein Netzwerk verwalten. Nachstehend ist die Netzwerktopologie vom Test dargestellt.



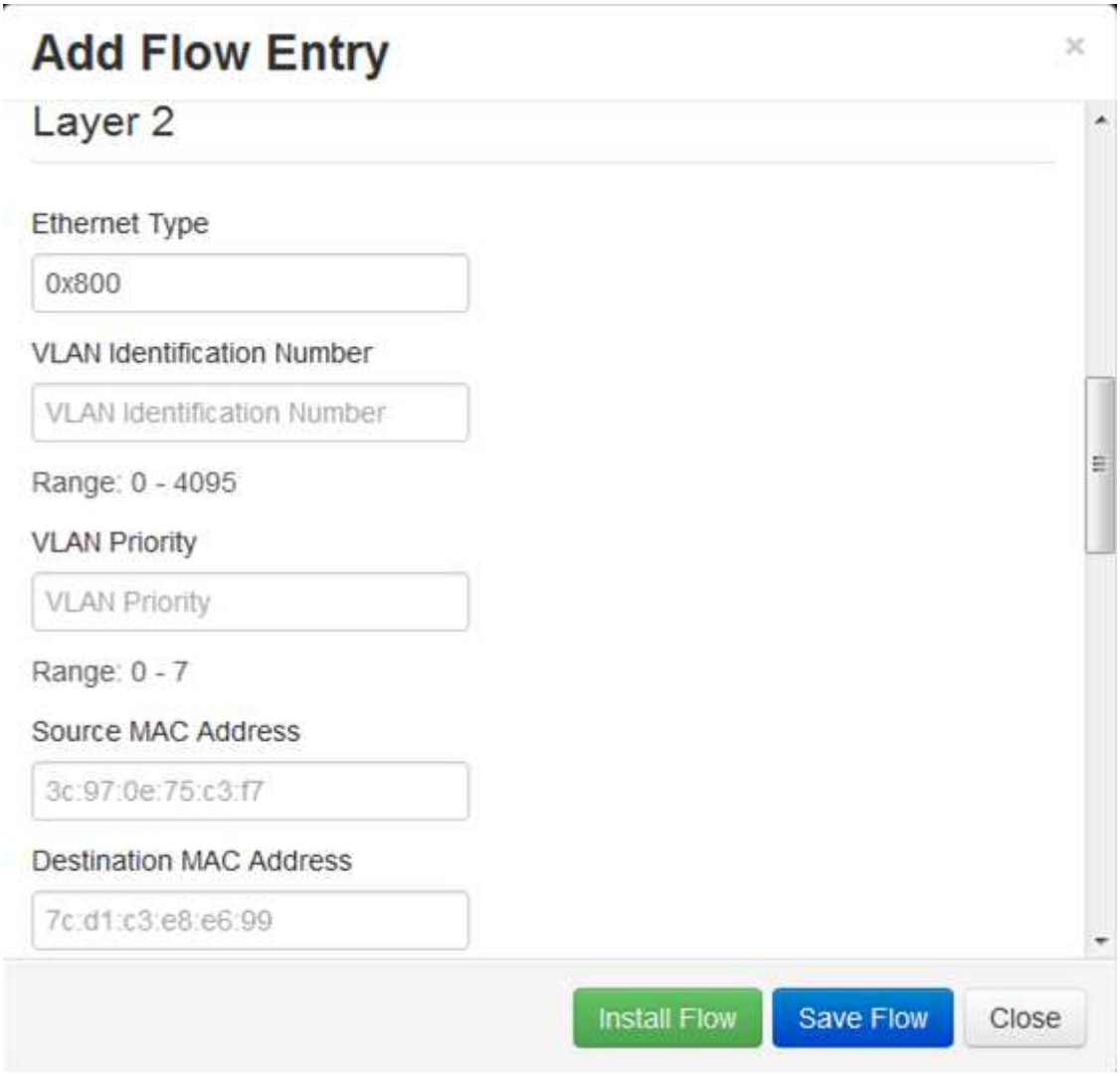
Netzwerktopologie vom Test Lab. Angezeigt von OpenDaylight

Für das Erstellen von einem Flow steht eine Eingabemaske zur Verfügung. Unter „Node“ kann ein Switch ausgewählt werden. Auf diesem wird dann der Flow generiert. Als „Input Port“ sind nur Ports wählbar, die auch aktiv sind. Das heisst eine Verbindung zu einem Host oder Netzwerkgerät haben.

Mit einem Timeout kann festgelegt werden, wie lange ein Flow in der Flowtable gespeichert wird.

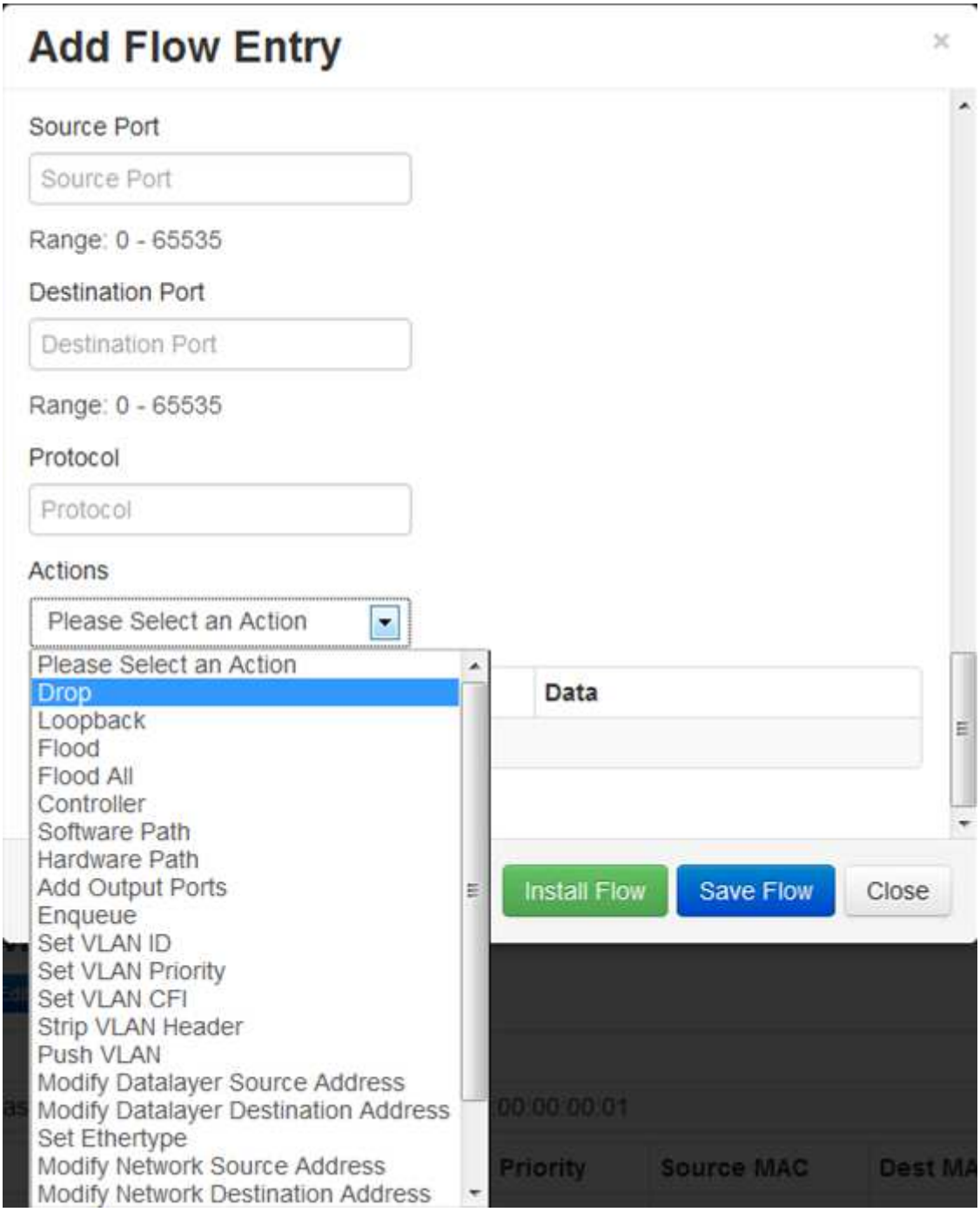


Flowkonfiguration auf OpenDaylight



Flowkonfiguration auf OpenDaylight

Danach kann der Ethernet Typ und ein VLAN angegeben werden. Diese Pakete werden dann dem Flow zugeordnet. Weiter können auch einzelne Hosts, mittels MAC Adresse, angegeben werden.



Flowkonfiguration auf OpenDaylight

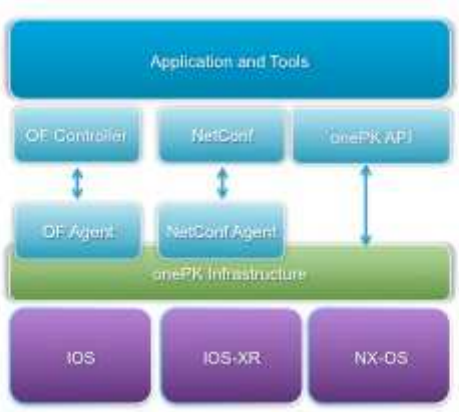
Zum Schluss wird noch definiert, was mit den Paketen geschehen soll. Sie können verworfen werden, auf alle oder auf einen bestimmten Port weitergeleitet werden, das VLAN Tag, oder MAC –und IP Adressen können verändert werden oder das VLAN Tag kann entfernt werden.

So muss jedoch ein Flow auf jedem einzelnen Switch erstellt werden. Dies macht natürlich keinen Sinn. Sonst könnte sich ein Netzwerkadministrator auch gleich per SSH auf jeden Switch einloggen

und die Konfiguration so erledigen. Über das REST API ist OpenDaylight jedoch bereit mit einem Orchestrar zusammenzuarbeiten.

5.11.4 Cisco

Cisco entwickelt einen eigenen SDN Controller. Er wird Application Policy Infrastructure Controller Enterprise Module (APIC EM) genannt. Dieser Controller basiert auf dem OpenDaylight Controller. Für die Kommunikation mit den Switch verwendet Cisco OnePK. Auf jedem IOS von Cisco wird dabei ein OnePK Agent installiert. Dieser nimmt die Befehle, die der Controller schickt, entgegen und wandelt sie in Befehle um, die die darunterliegende Hardware interpretieren kann. Im untenstehenden Bild sieht man zwei Agenten. Der eine nimmt Befehle über OpenFlow entgegen, der andere via Netconf. So können dieselben Befehle verwendet werden, egal ob die Hardware ein IOS, IOS-XR oder ein NX-OS verwendet.



OnePK Infrastruktur. http://blogs.cisco.com/wp-content/uploads/onepk-2.jpg

Der Controller ist auch über ein REST API ansprechbar. So kann mit einer Cisco Enterprise Application der Controller verwaltet werden. Über der Enterprise Application wird dann noch ein Orchestrar eingesetzt.

5.12 Ergebnisse

Zu Beginn der Arbeit wurden die drei verschiedenen Architekturen von Software Defined Networking analysiert. Dabei wurde festgestellt, dass der zentrale Ansatz am kostengünstigsten umgesetzt werden kann. Dies ist möglich, da die Netzwerkhardware keine eigene Intelligenz haben muss. Alle Entscheidungen werden zentral von einem Controller getroffen. Diese werden dann zum Beispiel mit OpenFlow an die Netzwerkhardware gesendet. Mit dieser Architektur können Netzwerke aufgebaut werden, die Herstellerunabhängig sind. Die Hardware muss nur das verwendete Kommunikationsprotokoll und alle benötigten Features unterstützen. Die Hardware kann nun von dem günstigsten Hersteller besorgt werden. Dies macht das Netzwerk auch sehr flexibel. Das centralized SDN hat natürlich auch einige Negative Punkte. Wenn alle Entscheidungen zentral getroffen werden, kann das Netzwerk nicht gut skalieren. Sobald das Netz über grössere Distanzen vergrössert wird, wird die Latenz zwischen dem Controller und der Netzwerkhardware zu gross. So können keine schnellen Entscheidungen getroffen werden. Weiter ist die Performance niedrig. Dies resultiert aus der fehlenden Intelligenz der Hardware. Das Weiterleiten der Pakete wird von der CPU erledigt und nicht in der Hardware.

Beim distributed SDN besitzt die Netzwerkhardware einen Agenten. Dieser kann bestimmte Entscheidungen selbst treffen ohne den Controller Anfragen zu müssen. So kann auch das Weiterleiten der Pakete in der Hardware erledigt werden was einen grossen Performance Vorteil hat. Leider wird jedoch die Hardware dadurch auch teuer und ist nicht mehr Herstellerunabhängig. Der Controller muss mit dem Agenten kommunizieren können. Da jeder Hersteller jedoch einen eigenen Agenten entwickelt, kann die Netzwerkhardware nicht mit einem Controller von einem anderen Hersteller kommunizieren. So können keine flexiblen Netzwerke aufgebaut werden und man bleibt abhängig von einem Hersteller. Bei dieser Architektur wird auch zentral ein Controller eingesetzt. Dadurch ist diese Variante auch nicht gut skalierbar.

Die dritte Architektur ist das hybride SDN und versucht Vorteile aus dem centralized SDN und dem distributed SDN zu ziehen. Dabei werden mehrere Controller eingesetzt damit das Netz gut skalieren kann. Die eingesetzte Netzwerkhardware besitzt keine eigene Intelligenz und ist dadurch auch wieder günstig und Herstellerunabhängig. Die verschiedenen Controller werden über das ganze Netzwerk verteilt. So können die Latenzen, für alle Regionen im Netz, niedrig gehalten werden. Dies hat natürlich einen positiven Einfluss auf die Performance. Als Nachteil ist hier der Preis zu erwähnen. Es werden mehrere Controller eingesetzt was den Preis nach oben treibt.

In der nachstehenden Tabelle werden die einzelnen Bewertungen nochmals zusammengefasst.

	Preis	Skalierbarkeit	Performance	Flexibilität	Abhängigkeit
Centralized SDN	Kostengünstige Hardware	Niedrig	Niedrig	Hoch	Herstellerunabhängig
Distributed SDN	Teure Hardware	Niedrig	Hoch	Niedrig	Herstellerabhängig
Hybrid SDN	Hoch	Hoch	Mittel	Hoch	Herstellerunabhängig

Aus meiner Sicht gibt es noch eine vierte Variante. Sie ist eine Kombination zwischen dem hybriden SDN und dem distributed SDN. Dabei werden auch mehrere Controller eingesetzt, die von einem zentralen Orchestrar verwaltet werden. Die Netwerkhardware hat einen Agenten eingebaut. Dieser ist auf das Kommunikationsprotokoll zugeschnitten. Das heisst, dass der Agent alle Features, die das Kommunikationsprotokoll unterstützt, auch unterstützen muss. So kann die Hardware herstellerunabhängig eingesetzt werden, hat jedoch trotzdem den Vorteil der besseren Performance. Durch den Agenten sind die Anschaffungskosten der Hardware jedoch hoch, was ein Nachteil bleibt dieser Variante.

Nach dem Überblick über die verschiedenen Architekturen werden die Anforderungen an SDN definiert. Dabei stehen die Kosten im Mittelpunkt. SDN soll es ermöglichen, ein kostengünstiges Netzwerk aufzubauen. Wie die Auswertung der Architekturen zeigt, wird dies mit den Anschaffungskosten nicht gelingen. Die höchsten Kosten werden jedoch durch das Netzmanagement verursacht. SDN soll nun diese Kosten möglichst weit nach unten drücken. Dies soll durch das zentrale Management und die einfachere Konfiguration des Netzwerks gelingen. Ein Administrator soll die Möglichkeit besitzen, einen Dienst im Netzwerk zu erstellen, ohne sich um die Konfiguration der einzelnen Netzwerkgeräte kümmern zu müssen. Ein Software Defined Netzwerk soll auch schneller auf einen Ausfall reagieren können. Sobald ein Gerät einen Ausfall meldet, teilt der Controller dies im ganzen Netzwerk mit. Weiter soll auch ein Link vor Überlast geschützt werden können. So soll SDN bestimmte Daten von einem Link auf einen anderen verschieben können.

Der Controller muss den Netzwerkadministrator bei der Fehlersuche unterstützen können. Es muss eine ganze Verbindung getestet werden können ohne dass sich der Admin um jedes einzelne Gerät kümmern muss.

Anhand der Anforderungen an SDN können Use Cases definiert werden. Beim ersten Use Case geht es um das Netzmanagement. Dabei wurde festgestellt, dass in einem heutigen Netzwerk das Management auf ein einzelnes Device abzielt. Man kann ein Überwachungstool, wie Icinga oder Smokeping, einsetzen, um einen Ausfall zu erkennen. Um den exakten Grund für den Ausfall zu erkennen, ist es oft nötig sich auf ein Gerät einzuloggen und nach der Ursache zu suchen. Auch bei der Konfiguration von einem Netz sieht es nicht anders aus. Es können zwar mit einem SNMP Script mehrere Geräte gleichzeitig Konfiguriert werden, wenn jedoch ein Fehler auftritt bekommt der Admin keine Rückmeldung. Der gewünschte Dienst funktioniert einfach nicht und es muss eine zeitaufwändige Fehlersuche gestartet werden. SDN kann hier Abhilfe schaffen. Bei der Fehlersuche werden ganze Verbindungen getestet und bei einem erkannten Fehler kann dieser sofort behoben

werden. Der Administrator muss sich nicht auf ein Netzwerkgerät einloggen. Die Konfiguration der Netzwerkhardware kann bei SDN via Netconf erledigt werden. Der Vorteil gegenüber von SNMP ist, dass Transactions verwendet werden. So werden nur ganze Konfigurationen auf ein Gerät geschrieben. Falls ein Fehler auftaucht, werden alle Änderungen wieder rückgängig gemacht. So bleiben keine Konfigurationsresten im Netzwerk übrig.

Bei der Analyse des Abrechnungsmanagements und des Leistungsmanagements wurde festgestellt, dass dies in einem Software Defined Netzwerk mit den Bordmitteln erledigt werden kann. Der Orchestrar kann einen Switch abfragen, wie viele Daten er pro Flow gesendet hat. Daraus kann eine Abrechnung für einen Kunden erstellt werden. Weiter kann auch die Auslastung von einem Link festgestellt werden. Dies kann auch wieder über die Flows geschehen, oder über NetFlow. So kann der Orchestrar feststellen, wenn ein Link zu stark ausgelastet ist. Bei SDN muss also nicht noch ein externes Tool eingesetzt werden, um einen Leistungsreport oder eine Abrechnung zu generieren.

Die Sicherheit von einem Netzwerkgerät muss natürlich auch bei SDN sichergestellt werden. Der physikalische Schutz spielt dabei eine wichtige Rolle. Ein Serverraum oder ein Datacenter muss immer vor unerlaubtem Zugriff geschützt werden. Andernfalls kann ein Angreifer ein Gerät zerstören, umgehen oder den Datenverkehr abhören. Weiter müssen alle Geräte mit einem Passwort geschützt werden, und alle Managementdaten müssen verschlüsselt übertragen werden.

Der zweite Use Case behandelt das Routing und das Rerouting. Es wurde untersucht, ob das Routing in einem SDN Netz schneller ist als in einem herkömmlichen Netz. Dies ist dann der Fall, wenn nicht ein direkte angeschlossener Link davon betroffen ist. In einem heutigen Netz wird anhand der fehlenden Antworten des Nachbarn erkannt, dass ein Unterbruch vorliegt. Dies kann bis zu Minuten dauern. Ein SDN Controller kann die Verfügbarkeit von den Geräten überprüfen und gegebenenfalls einen Ausfall im Netzwerk melden. So kann viel Zeit gespart werden.

Zum Thema Routing gehört auch das explizite Routing dazu. Dabei geht es darum, bestimmte Daten auf einen anderen Link umzuleiten. Der Switch muss dann jedoch eine Layer 7 Inspektion der Pakete durchführen können. Explizites Routing ermöglicht es nun einer Firma mit zwei Internet Uplinks, beide Links zu benutzen. Es können dann auch mehr als 50% der maximalen Bandbreite verwendet werden. Fällt nun ein Link aus, muss entschieden werden, welche Daten verworfen werden können. Dies wird anhand verschiedener Policy realisiert, die vom Orchestrar im Netzwerk verteilt werden. Diese Policy können nun bis zu den Access Switch gesendet werden. Dadurch können die weniger wichtigen Daten sehr nahe beim Client verworfen, oder niedriger priorisiert werden. Mit SDN ist es also möglich nur noch die wichtigen Daten über den verbleibenden Link zu senden.

Als nächstes wurde untersucht, wie ein Netzwerk virtualisiert werden kann. Das Ziel besteht darin, einem Kunden ein eigenes Netz zur Verfügung zu stellen. So muss er keine Angst haben, dass seine Daten mit denen der anderen Kunden vermischt werden. In einem heutigen Netzwerk wird dies zum Beispiel mittels MPLS Labels und VRF gelöst. Bei SDN können mehrere Flows zu einem sogenannten Slice zusammengefasst werden. So ist es möglich, dass für ähnliche Anforderungen nicht alle Flows nochmals definiert werden müssen. Es kann einfach ein Slice verwendet werden. Ein Service Provider kann so also einem Neukunden einfach einen Slice zuordnen und schon sind die richtigen Policy dem Kunden zugeordnet. Das reduziert den Konfigurationsaufwand erheblich.

Beim Use Case Netzwerkzugriffs Kontrolle geht es um das authentisieren von einem Benutzer. Sobald man sich mit dem Netzwerk verbindet, muss man sich mit dem Usernamen und Passwort anmelden. Dabei fragt der Switch nach den Userdaten und leitet diese dann an den Orchestrar weiter. Dieser überprüft, ob der Benutzer die Erlaubnis hat, sich mit diesem Netz zu verbinden. Danach teilt der Orchestrar dem Switch mit, ob er den Benutzer zulassen soll oder nicht. Der Benutzer kann auch zuerst in ein Quarantäne Netz umgeleitet werden. Da muss er sich zuerst die neuste Virendatenbank runterladen. Erst danach wird er in das produktive Netz weitergeleitet. Heute funktioniert eine solche Authentisierung meist über einen Radius Server. Statt dass der Switch den Orchestrar anfragt, fragt er den Radius Server an. Damit nicht gleich die ganze Benutzerauthentisierung umgestellt werden muss, wenn ein Netzwerk auf SDN umgestellt wird, muss der Orchestrar die Möglichkeit haben, eine Anfrage an den Radius Server zu stellen.

Ein Software Defined Network muss natürlich auch überwacht werden. Dabei können die SDN Controller die Verfügbarkeit der einzelnen Netzwerkgeräte periodisch überprüfen. Am einfachsten wird dies mit OpenFlow Hello Paket erledigt. Sobald ein Gerät keine Antwort mehr gibt, wird ein Alarm ausgelöst. OpenFlow bietet jedoch noch die Möglichkeit, die Datenmenge anzuzeigen, die über einen Flow gesendet wurde. Daraus können dann Statistiken erstellt werden, um das Surfverhalten der Benutzer zu analysieren. Diese Statistiken können dann für die Verbesserung der Policy verwendet werden. In heutigen Netzwerken wird eine solche Überwachung mit externen Systemen durchgeführt. Dies erfordern natürlich wieder Know How und ist deshalb nicht so einfach einzusetzen wie in einem SDN Netz.

Beim Thema Sicherheit stellt sich die Frage, ob SDN die Netzwerksicherheit erhöhen kann. In einem heutigen Netz wird beim Übergang zum Internet ein Wall von Sicherheitssystemen aufgebaut. Das kann ein Router sein, der DoS Angriffe abwehren soll. Dahinter wird meist eine Firewall eingesetzt, die eine genauere Analyse der Daten machen kann. In der dritten Verteidigung stufe kann ein Intrusion Prevension System eingesetzt werden. Dieses System erkennt einen Virus oder einen Angriff anhand einer Signaturdatenbank. Diese erste Verteidigungslinie kann durch SDN nicht ersetzt werden und muss deshalb auch in einem SDN Netz vorhanden sein. SDN kann jedoch die internen Firewalls und Filterregeln ersetzen. Eine Policy wird im gesamten Netz verteilt. So sind alle Daten von der Security Policy betroffen, nicht nur solche, die über eine Firewall gesendet werden. So wird die Sicherheit in einem Software Defined Network erhöht.

Nach dem Use Cases wurden drei Beispiele von SDN Netzwerken beschrieben. Beim ersten geht es um eine Firma, die zwei Datacenter betreibt. Um ein Verschieben von virtuellen Maschinen zu ermöglichen, mietet die Firma einen Layer 2 Link zwischen den Datacentern. Der Service Provider, der diesen Link zur Verfügung stellt, verwendet ein Software Defined Network. Die Frage stellt sich nun, ob der Service Provider etwas davon wissen muss, wenn der Kunde eine VM verschiebt. Die Antwort lautet nein. Den Provider interessieren die Daten nicht, die über den Link gesendet werden. Er stellt nur das Transportnetz zur Verfügung. Damit die virtuelle Maschine, nach dem Verschieben wieder erreichbar ist, muss der Kunde besorgt sein. Dieses Beispiel zeigt, dass Software Defined Network und Software Defined Datacenter nicht immer zusammengehören.

Im nächsten Beispiel wird ein Netzwerk von einem Service Provider etwas genauer angeschaut. Dieser betreibt mehrere Datacenter und möchte verschiedene Dienste für seine Kunden zur

Verfügung stellen. Ein Dienst wird in einem Datacenter betrieben und im Netzwerk über Flows angeboten. Ein Kunde greift nun über den Flow auf diesen Dienst zu. Wird nun die virtuelle Maschine, auf der dieser Dienst betrieben wird, in ein anderes Datacenter verschoben, muss das Netzwerk darauf reagieren können. Nach dem Verschieben wird also der Flow des Kunden so umgeleitet, dass er in das andere Datacenter zeigt. So ist der Dienst weiterhin erreichbar und der Kunde muss keine Änderungen vornehmen. Dies ist nun ein Beispiel, bei dem Software Defined Network und Software Defined Datacenter zusammenspielen müssen.

Beim dritten Beispiel geht es um eine Firma, die das Netzwerk, das Datacenter und das WLAN Netz von einem Orchestrar verwalten lässt. So ist es möglich, einen Dienst im WLAN Netz anzubieten, der in einer virtuellen Maschine im Datacenter läuft. Die Daten von den Benutzern werden nun über das WLAN Netz ins SDN Netz übertragen. Da ist wieder ein Flow definiert, der den Dienst zur Verfügung stellt. Der Orchestrar muss mit allen drei Controllern kommunizieren können.

Da es bis heute noch keinen Orchestrar gibt auf dem Markt, wurde in dieser Arbeit eine mögliche Implementation behandelt. Dabei wurde festgestellt, dass das Erstellen von einem Dienst aufwändig sein wird bei der Umsetzung. Der Orchestrar muss einen Dictionary Dienst betreiben, damit ein Administrator einen Dienst für ein Gebäude oder für einen Raum zur Verfügung stellen kann. Sonst müsste man für jeden einzelnen Switch Port den Dienst erstellen. Weiter ist so die Fehlersuche einfacher. Es kann überprüft werden, ob aus einem bestimmten Raum auf das Internet zugegriffen werden darf oder nicht. Das Usermanagement soll auch gleich in den Orchestrar integriert werden. So kann, bei der Anmeldung von einem Benutzer, gleich ein Eintrag im Dictionary gemacht werden. So kann ein Switch Port einem User zugeordnet werden, was das Netzmanagement vereinfacht. Ein Orchestrar muss auch protokollieren können, welcher Admin welche Änderungen auf dem System durchführt. Damit diese Daten gut geschützt sind, werden sie auf ein weiterers System abgelegt. Dabei kann es sich um einen Syslog Server handeln. Der Orchestrar muss also Syslog Nachrichten versenden können. Ein Orchestrar soll einen Leistungsreport für ein Netzwerk durchführen können. Also muss er die Auslastung von einem Link messen können. Am besten gelingt dies mit NetFlow. Diese Statistik kann nun als PDF abgespeichert oder ausgedruckt werden. Die Analyse der Aufgaben von einem Orchestrar hat gezeigt, dass die Umsetzung aufwändig aber realistisch ist.

Um zu überprüfen, ob Software Defined Network überhaupt benötigt wird, wird ein Vergleich mit Frame Relay und Performance Routing durchgeführt. In einem Frame Relay Netz werden sogenannte Permanent Virtual Circuits (PVC) konfiguriert. Dies ist eine logische Verbindung zwischen zwei Punkten im Netz. Dieser PVC wird mit Data-Link Connection Identifier (DLCI) erstellt. Ein DLCI definiert eine Verbindung zwischen zwei Frame Relay Switch. Dies ist vergleichbar mit einem Flow bei SDN. Das Erstellen der einzelnen DLCI ist sehr zeitaufwändig. Bei SDN werden die einzelnen Flows von einem Controller im Netz konfiguriert. Somit ist der Konfigurationsaufwand bei SDN erheblich kleiner.

Wie sieht es nun bei Performance Routing aus? Beim Performance Routing wird ein Master Controller und ein oder mehrere Border Router konfiguriert. Die Border Router messen mit IP SLA die Dienstgüte von einem Link. Diese Auswertung senden sie dann mit NetFlow an den Master Controller. Wird nun ein vordefinierter Wert überschritten, sendet der Master Controller eine Default Route an den Border Router. So können die Daten auf einen anderen Link umgeleitet

werden. Dies ist jedoch nicht möglich für einzelne Daten. Ein SDN Orchestrar hat die Möglichkeit, nur bestimmte Daten auf einen anderen Link umzuleiten und ist somit flexibler einsetzbar als Performance Routing.

Im praktischen Teil geht es nun darum, die definierten Use Cases in der Praxis zu testen. Es werden vier unterschiedliche Lösungen getestet. Die Lösungen sind von HP, Arista, Cisco und von der Opensource Community. HP ist der einzige Hersteller, der eine komplette Lösung anbietet. Arista und Cisco haben noch keinen Controller auf dem Markt. Deshalb können nicht alle Tests durchgeführt werden.

5.13 Schlussfolgerung

Laut Gartner befindet sich Software Defined Networking in einem Hype. Dies merkt man auch schnell, wenn man sich mit dem Thema beschäftigt. Für jeden Hersteller ist SDN etwas anderes. HP verwendet OpenFlow und Switch, die keine eigenen Entscheidungen treffen, Arista behauptet, dass sich OpenFlow nie durchsetzen wird und entwickelt keinen eigenen Controller, Cisco setzt auf intelligente Switch und entwickelt eine Alternative zu OpenFlow und zum Schluss haben wir noch die Opensouce Community, die auf OpenFlow setzen. Das Einzige, bei dem sich die Hersteller einig sind, ist, dass man mit Software Defined Netzworking Geld verdienen kann.

Die meisten Informationen, die man findet, sind Marketing. Es wird erklärt, was alles möglich ist und dass vieles einfacher wird. Die Produkte, die auf dem Markt verfügbar sind können jedoch nicht produktiv eingesetzt werden. Dafür haben sie einfach zu wenige Funktionen und sind zu aufwändig bei der Verwendung.

Dadurch ist es kaum möglich, einen Test durchzuführen, um die einzelnen Hersteller zu vergleichen. Von Cisco und Arista ist, zum jetzigen Zeitpunkt, kein Controller verfügbar. Der einzige Vergleich konnte zwischen dem Opensource Controller OpenDaylight und dem Controller von HP gemacht werden. Dabei stellte sich heraus, dass sich der OpenDaylight intelligenter verhält. Wenn ein Switch einen Flow anfragt, spielt der Controller den Flow auf alle betroffenen Switch. So muss nicht jeder einzelne Switch denselben Flow anfragen, was Zeit spart. Geplant war, dass Tests gemacht werden mit einer virtuellen Maschine, die verschoben wird über die Layer 3 Grenze hinweg. Der Controller sollte dies dann erkennen und darauf reagieren. Es stellte sich jedoch schnell heraus, dass die heutigen Controller nicht dafür gedacht sind, ein Verschieben zu erkennen. Solche Tests wurden deshalb nicht durchgeführt.

Die Studienarbeit hat gezeigt, dass Software Defined Network ein grosses Potential besitzt. Die Hersteller haben jedoch noch viel Arbeit vor sich, bis SDN wirklich produktiv eingesetzt werden kann.

5.13.1.1 Quellenangaben

http://searchsdn.techtarget.com/resources/SDN-management-applications

http://searchsdn.techtarget.com/feature/The-programmable-WAN-Applications-are-boss-and-networks-bend

http://searchsdn.techtarget.com/tip/Centralized-vs-decentralized-SDN-architecture-Which-works-for-you

http://searchsdn.techtarget.com/guides/Northbound-API-guide-The-rise-of-the-network-applications

http://searchsdn.techtarget.com/feature/A-primer-on-northbound-APIs-Their-role-in-a-software-defined-network

http://searchsdn.techtarget.com/feature/Software-defined-networking-applications-move-beyond-the-data-center

http://etherealmind.com/tech-review-netconf-and-yang/

http://searchsdn.techtarget.com/essentialguide/SDN-basics-for-service-providers

http://www.avaya.com/uk/resource/assets/whitepapers/dn4469%20-%20network%20virtual%20using%20spb%20white%20paper.pdf

http://blog.ipspace.net/2012/06/we-need-both-openflow-and-netconf.html

http://searchsdn.techtarget.com/feature/Border-Gateway-Protocol-as-a-hybrid-SDN-protocol