

Bandbreite! Bandbreite! Bandbreite!



Ralf Sydekum, 2013-21-10

Sie kennen das bestimmt: Es muss immer alles schneller gehen und von Geschwindigkeit können die meisten Nutzer gar nicht genug kriegen. Manchmal ist die Bandbreite aber schlicht und einfach limitiert und man muss mit den vorhandenen Ressourcen haushalten. Wie kann man also die verfügbare Bandbreite am besten nutzen, ohne dass Services und Leistungen Einbußen hinnehmen müssen?

Der Weg zu einer optimalen Bereitstellung von Bandbreite führt meist über die Applikation. So wird SDN (in seiner klassischen, architektonischen Definition) dafür genutzt, die Qualität der Services für Anwendungen durch Priorisierung und Verteilung der Bandbreite auf einem hohen Niveau zu halten. Manchmal geschieht dies sogar dynamisch in Abhängigkeit von den herrschenden Bedingungen im Netzwerk. In diesem Szenario spricht die Applikation mit dem SDN-Controller und teilt ihm ihre Voraussetzungen hinsichtlich Bandbreite und Service-Level-Bedingungen mit. Der Controller passt das Netzwerk entsprechend an und leitet die entsprechenden Daten an die Netzwerk-Fabric weiter, um die Bedingungen der Anwendung zu erfüllen. Dies ist im Prinzip ein tolles Vorgehen, in der Realität gibt es aber einige Komplikationen.

Da wäre zum Beispiel die Herausforderung, Applikationen zur Kommunikation mit dem SDN-Controller zu bewegen. Dafür könnten Northbound APIs oder andere API-basierte Mechanismen genutzt werden. Nehmen wir einmal an, das klappt problemlos: Dann muss der SDN-Controller die Informationen immer noch an die Fabric weiterleiten. Zwei Fragen sind hier zu klären: Wie repräsentiert der SDN-Controller die Voraussetzungen der Applikation und wie gibt er diese weiter?

Wenn wir von einer klassischen SDN-Architektur und der Nutzung von OpenFlow oder einem vergleichbaren Protokoll ausgehen, kann die Flow Table in der Fabric Pakete lediglich hinsichtlich der IP/Port-Kombination unterscheiden. Da jede Anwendung über eine bestimmte IP/Port-Kombination verfügt, ist dies aber eine gute Repräsentation der gesamten Topologie. Der SDN-Controller kann Flow-Table-Regeln weiterleiten, die die Bandbreite und die Priorität festlegen.

Wie bestimmt die Applikation nun ihren Bedarf an Bandbreite? Applikationen können eine Vielzahl von Funktionen und Daten enthalten – von Texten über Bilder und Videos bis hin zu Multimedia-Inhalten. Der Bedarf all dieser Medien unterscheidet sich dabei stark. Latenz, Durchsatz und Bandbreite sind also entscheidend für die verschiedenen Inhalte, nicht für die Applikation an sich. Sollte nun eine Anwendung vom Netzwerk den Maximaldurchsatz und die niedrigste Latenz anfordern, die im Extremfall benötigt werden? Das wäre in der gesamten Zeit, in der die Anwendung nicht die volle Leistung benötigt, sehr ineffizient.

HTTP ist das neue TCP

Schwierigkeiten tauchen auf, weil HTTP zum neuen TCP aufgestiegen ist. Schon 62 Prozent der in unserer Studie erfassten Applikationen nutzen http (siehe Grafik). Ein kleinerer Teil verwendet Port 8080 und Port 443, die wir aber dennoch zu HTTP zählen. Da immer mehr Anwendungen auf die API zurückgreifen, können wir den Bedarf von Applikationen am besten auf der URI-Ebene bestimmen.



Die für uns wirklich interessanten Dinge spielen sich in Layer 7 (HTTP) ab. Um den spezifischen Bedarf an Bandbreite zu bestimmen, muss nämlich der Inhalt betrachtet werden: In den meisten Fällen ist dies mit Hilfe der URI (Datenbezeichnungen wie JPG, PNG, CSS, etc...) möglich oder es kann der HTTP Header analysiert werden, der den „Content-Type“ bestimmt. In jedem Fall müssen die HTTP-Nutzdaten analysiert werden, nicht nur die IP- und TCP-Parameter.

Schwierig wird das Verfahren durch die aktuelle Architektur des SDN, da sie sehr stark auf Paket- und Flow-basierte Prozesse fokussiert ist und die Art des Inhalts innerhalb einer Applikation nicht erfassen kann. Deshalb setzt sie für die Anwendungen Routing- und Weiterleitungs-Policies basierend auf den Anforderungen eines jeden Inhaltstyps durch. Eine Anwendung, die sowohl Video als auch Text überträgt, wird entweder für den einen oder den anderen Inhaltstyp optimiert, aber nicht für beide. Dasselbe gilt für Bilder und für unterschiedliche Auslieferungsmodelle (Push, Pull, Echtzeit, statisch) von textbasierenden Informationen.

Für eine Optimierung des Verfahrens ist also eine Sichtbarkeit in die Applikation (manchmal auch in die Nutzdaten) hinein erforderlich. Da die klassische SDN-Architektur aber nur in Layer 2-4 blicken kann, fehlen ihr schlicht die Informationen, die sich in Layer 7 verbergen. Die Lösung scheint naheliegend: Die SDN-Architektur muss mit einem Datenpfad zu Layer 7 ausgestattet werden. Auf diese Weise könnte die Standard Layer 2-3 SDN Fabric die Pakete auf Basis genereller, applikationsorientierter Netzwerkvoraussetzungen optimal durch das Netzwerk führen, während der Layer 7-Datenpfad das macht, was er am besten kann: Applikationsnachrichten inspizieren, analysieren, evaluieren und modifizieren, damit die Daten am besten zum Endnutzer gelangen.

Für den reibungslosen Ablauf im SDN ist es absolut erforderlich, dass verstanden wird, wie Netzwerkprotokolle, -verhalten und -variablen die Leistung, Sicherheit und Verfügbarkeit einer jeden Anwendung beeinflussen. Manchmal muss man dafür seinen Horizont erweitern – in unserem Fall bis zu Layer 7 und darüber hinaus!

F5 Networks, Inc. | 401 Elliot Avenue West, Seattle, WA 98119 | 888-882-4447 | f5.com

F5 Networks, Inc.
Corporate Headquarters
info@f5.com

F5 Networks
Asia-Pacific
apacinfo@f5.com

F5 Networks Ltd.
Europe/Middle-East/Africa
emeainfo@f5.com

F5 Networks
Japan K.K.
f5j-info@f5.com