Vergleich von Hardware- und Software-Traffic-Generatoren und ihrem Einsatz in der Praxis

Tobias Weiher
Betreuer: Paul Emmerich, Daniel Raumer
Seminar Future Internet WS2015
Lehrstuhl Netzarchitekturen und Netzdienste
Fakultät für Informatik, Technische Universität München
Email: weiher@in.tum.de

KURZFASSUNG

Das Testen von Hardware für Netzwerkeinsätze wird immer wichtiger. Durch die wachsende Größe und Geschwindigkeit der Netzwerke werden auch ihre Anforderungen anspruchsvoller. Hardware wie Router und Switches müssen wie vorgesehen funktionieren und Last standhalten, um repräsentative Testergebnisse zu erhalten und auch realen Traffic zu unterstützen. Software-basierte Ansätze existieren, die unflexible hardware-orientierte Lösungen ersetzen wollen, jedoch sind jene meist unpräzise und letztere teuer. Hybride Software-Traffic-Generatoren mit Hardwareunterstützung sind eine Möglichkeit, um Präzision und geringere Kosten zu vereinbaren.

Schlüsselworte

Traffic-Generatoren, Software, Hardware, Performanz

1. EINLEITUNG

Das Internet und seine Netzwerke werden immer größer, schneller und ihre Anforderungen auf die im Hintergrund befindlichen Ressourcen anspruchsvoller. Darunter fallen Hardware wie Router und Switches, die wie vorgesehen funktionieren und Last standhalten müssen. Unter Last wird der Aufwand verstanden, den die Hardware standhalten muss, wenn sie sehr viele Pakete mit einer hohen Datenrate, dem Traffic, bearbeiten und weiterleiten muss.
Um dieses Verhalten zu testen, wird auf verschiedene Verfahren von Traffic-Generatoren zurückgegriffen. Häufig werden von größeren Unternehmen teure Hardware-Lösungen verwendet, die für gewisse Anforderungen entwickelt wurden, um gezielt die Performanz zu testen. Jedoch ist diese Herangehensweise nicht sehr flexibel, da die Hardware meist nicht selbst konfigurier- und änderbar ist. Manchmal wird jedoch ein flexibler Weg des Testens erwünscht, besonders im Umgang mit neueren Protokollen oder Netzwerkdesigns. Hierzu gibt es Traffic-Generatoren, die in Software realisiert sind. Allerdings kann bei diesen Alternativen nicht gewährleistet werden, ob ihre Anforderung auch von der zugrundeliegenden Maschine umgesetzt werden kann. Aufgrund dessen gibt es Bemühungen verschiedenster Forschungsgruppen, software-basierte Ansätze zu verbessern, um die teuren hardware-orientierten Lösungen zu ersetzen.
Im folgenden Abschnitt 2 wird zunächst auf die generelle Unterscheidung der Traffic-Generatoren eingegangen. Zudem werden einige Begrifflichkeiten im Zusammenhang zu diesen Generatoren erklärt. Im Abschnitt 3 wird der Unterschied zwischen Hardware- und Software-Generatoren verdeutlicht. Unterabschnitt 3.1 geht auf die Stärken, sowie Schwächen der Hardware-Traffic-Generatoren ein. Unterunterabschnitt 3.1.1 beschreibt mit NetFPGA eine Open Source Alternative zu proprietären Hardwaresystemen. Anschließend werden mit Unterabschnitt 3.2 die software-basierten Traffic-Generatoren betrachtet. Unterunterabschnitt 3.2.1 erwähnt die Vorteile gegenüber einfachen Software-Generatoren und beschreibt deren Funktion. Im Unterabschnitt 3.3 wird Bezug zu einigen Statistiken genommen. Zunächst werden die Traffic-Generatoren in Hinsicht auf ihrer Paketraten in Unterunterabschnitt 3.3.1 verglichen. Danach werden in Unterunterabschnitt 3.3.2 einige Strategien zur Umsetzung der Sendeintervalle zwischen Paketen, beziehungsweise den *inter-departure times* präsentiert. Der letzte Teil mit Unterabschnitt 3.4 zeigt einige Netzwerk-Tools, die in der Praxis eingesetzt werden.

2. ARTEN VON TRAFFIC-GENERATOREN

Traffic-Generatoren sind Werkzeuge, um Netzwerke hinsichtlich ihrer Performanz und Stabilität zu überprüfen, um womögliche Probleme frühzeitig in der Entwicklung zu entdecken. Hierbei werden Daten in das Netzwerk eingespeist, um angeschlossene Geräte wie Router oder Switches zu testen. Dabei wird unterschieden, welche Daten in welchem Maße zugeführt werden.
Diverse Traffic-Generatoren lassen sich grob in folgende Kategorien unterteilen.[1]

- Traffic-Generatoren auf Anwendungsebene
- Traffic-Generatoren auf Datenflussebene
- Traffic-Generatoren auf Paketebene
- Traffic-Generatoren im geschlossenen Kreis und mit mehreren Ebenen

Die Generatoren, die auf Anwendungsebene funktionieren, erlauben, das Verhalten von Programmen und ihrem Einfluss auf das Netzwerk zu emulieren. Ein Programm, das beispielsweise mehrere Spielserver emuliert und die Netzwerkdaten von vielen Clients und der Server selbst austauscht und versendet, würde zu diesen Traffic-Generatoren zählen. Ein Netzwerk-Tool, welches lediglich Pakete über eine gewisse Dauer für einen Datenfluss erzeugt, wird den Traffic-Generatoren auf Datenflussebene zugeordnet. Hierbei wird versucht, möglich realistischen Traffic zu generieren, beispielsweise die Anzahl und Größe der Pakete, die bei einem Webshop-Einkauf entstehen würden.

doi: 10.2313/NET-2016-07-1_03

Der Großteil der Traffic-Generatoren arbeitet allerdings auf Paketebene. Hierbei werden verschiedenste, vom Benutzer definierte Parameter umgesetzt, um beispielsweise die Paketgröße, die Verzögerung von zu sendenden Pakete (auch *inter-packet delay* genannt, also die mindestens zu wartende Zeit, bevor ein Paket gesendet werden darf) oder die Paketrate, häufig gemessen in Anzahl Pakete minimaler Größe innerhalb einer Sekunde, zu konfigurieren. Auch kann das Sendeintervall bestimmt werden, welche die Zeit zwischen zwei zu sendenden Pakete so abstimmt, dass das zweite Paket erst versendet wird, nachdem eine gewisse Zeit nach dem vorherigen Versenden abgelaufen ist. Diesen Begriff versteht man unter anderem auch als *inter-departure time*, welche zu jedem Paket unterschiedlich ausfallen kann, da sie in der Regel pro Paket definiert wird. Jedoch kann dieses Sendeintervall nicht geringer sein als die minimale Verzögerung zwischen zwei Paketen, das *inter-packet delay*. Letztere wird häufig durch die physikalische Grenzleistung der Übertragung definiert, der *line rate*, im Gegensatz zur *bit rate*, die die Bitrate der Übertragung angibt.[2] Die maximale Bitrate kann nicht größer sein als die durch die physikalische *line rate* mögliche Leitung, sie kann jedoch niedriger ausfallen. Die maximale Bitrate von 10GbE, also 10 Gigabit Ethernet, ist 10.000.000.000 bps, damit 10 Milliarden Bits pro Sekunde. Um dies in einer maximalen Paketrate anzugeben, müssen die 10 Milliarden bps durch die Präambel eines jeden Paketes, der geringst möglichen Framelänge und dem sogenannten *inter-frame gap*, dem Warteabstand zwischen zwei Paketen, geteilt werden. Dadurch erhält man eine maximale Frame Rate von

$$\frac{10 \text{ Gigabits/s}}{\text{Präambel} + \text{Framelänge} + \text{inter-frame gap in bits}} = \frac{10.000.000.000 \text{ bits/s}}{8 * 8 \text{ bits} + 64 * 8 \text{ bits} + 12 * 8 \text{ bits}} = \frac{10.000.000.000 \text{ b}}{672 \text{ b} * \text{s}} =$$

$14.880.952,38$ Pakete pro Sekunde.[3]

Die letzte Kategorie umfasst Traffic-Generatoren, die die Interaktion über mehrere Schichten des Netzwerk-Protokoll-Stacks hinweg miteinbeziehen. Hier werden Benutzer-, sowie Session- und Anwendungs-Informationen extrahiert, um Netzwerkcharakteristika zu ermitteln, die ein realistischeres Traffic-Abbild erstellen sollen.
Damit immer überprüft werden kann, dass der erwünschte Traffic durch die Generatoren erzeugt wird, sollten all diese Generatoren mit sowohl Generierungs-, als auch Überwachungs-Funktionalität ausgestattet sein. Das bedeutet, dass Traffic-Generatoren nicht bloß Daten erzeugen können sollten, sondern auch eine Bestätigung über die erreichte Rate ausgeben können oder, falls diese nicht erreicht wird, dementsprechend tatsächliche Werte präsentieren. Mithilfe gewisser Zeitstempelverfahren kann dadurch sogar eine gezieltere Paketrate realisiert werden, welche in folgenden Abschnitten näher beschrieben werden.

3. VOR- UND NACHTEILE DER GENERATOREN

Traffic-Generatoren erfüllen den Zweck, Testgeräte oder Netzwerke auf Fehler und Performanz zu überprüfen. Dabei kann auf verschiedenste Realisierungen von Netzwerk-Tools zurückgegriffen werden. Es gibt sowohl hardware-basierte Ansätze wie auch Software-Traffic-Generatoren. Um zu differenzieren, welche Tools für welchen Gebrauch geeigneter erscheinen, werden generelle Lösungen betrachtet und verglichen.

3.1 Hardware-basiert

Traffic-Generatoren auf Hardware-Basis sind geschlossene, zu meist nicht näher konfigurierbare Systeme, die vordefinierte Daten nach Angaben des Herstellers erzeugen. Beispiele hierfür sind Netzwerktester von Ixia[4] oder Spirent[5], die Traffic-Generatoren für 1/10/40/100 GbE beziehungsweise sogar höhere Gigabit Ethernet Werte anbieten. Diese werben damit, Daten mit *line rate* erzeugen zu können, damit also die höchste Paketrate des jeweiligen GbE-Wertes erreichen, teilweise sogar darüber hinaus, indem sie die minimale Framelänge der kleinsten Ethernetpakete nochmals kürzen, beispielsweise auf 58 Byte. Da die Hersteller kompletten Zugriff auf ihre Hardware haben, erzielen sie sehr genaue Zeitwerte, die auch dank der präzisen Zeitstempel im Nanosekundenbereich besonders niedrig ausfallen. Die Anzahl der versendeten Bytes, unabhängig der Größe der Pakete, der Hardware-Traffic-Generatoren kann dabei nach ihren Angaben durchwegs die maximale Rate erreichen.[6]
Allerdings können nur diejenigen Protokolle verwendet werden, die zum Zeitpunkt der Erstellung der Hardware beziehungsweise ihrem Erwerb unterstützt wurden. Neuartige Protokolle werden durch die Hardware nicht erkannt. Immerhin kann bei den meisten Hardware-Generatoren der Test-Traffic durch ein Script benutzerdefiniert eingestellt werden, sodass ein Testgerät nach eigenen Bedürfnissen beansprucht und überprüft werden kann.[6]
Jedoch sind diese präzisen, hardware-basierten Traffic-Generatoren teuer und somit für Forschungseinrichtungen nicht sinnvoll erwerbbar, um beispielsweise die Daten und Spezifikationen der Hardware-Generatoren genauer zu untersuchen. Ein gebrauchter Traffic-Generator dieser Art kostet bereits 12.000$[7], neuere Geräte können damit gut das Doppelte des Preises wert sein. Das Hinzufügen weiterer unterstützter Protokolle und Funktionen für diese Module erhöht den Preis zusätzlich.[15]

3.1.1 NetFPGA

Um den hohen Preisen der Hardwarehersteller entgegenzuwirken und mehr Flexibilität in das Testen durch Traffic-Generatoren zu bringen, wurde das Projekt NetFPGA[8] ins Leben gerufen. Das Projekt begann im Jahr 2001 an der Stanford University, um Studenten das Netzwerkverhalten und die benötigte Hardware zu veranschaulichen. Da das Projekt jedoch größer wurde und die Hardware immer ausgereifter, stieg die Nachfrage nach einem offizielleren Gerät speziell für Forscher an.[9]
NetFPGA bietet Software und Hardware an, um neue Designs, Simulationen und das Testen auf einer Open Source Netzwerkplattform zu vereinfachen. Der Projektname leitet sich von *network* und dem *FPGA*, einem *field-programmable gate array*, ab. Mit diesen *FPGAs* sind Architekturen gemeint, die nach der Produktion noch vom Benutzer umprogrammierbar sind. Die erste NetFPGA-Plattform, erstellt für Forschungs- und Lehrgruppen, war das NetFPGA-1G im Jahr 2007, eine kostengünstige Platine für 1 Gigabit Ethernet.[10]
Der Nachfolger von 2010 war das NetFPGA-10G (Abbildung 1), welche mit einer 40 Gigabit pro Sekunde fähigen PCIe Schnittstelle ausgestattet ist und 4 10 GbE Verbindun-

doi: 10.2313/NET-2016-07-1_03

gen mitbringt. Alle Karten des NetFPGA-Projektes sind mit *FPGAs* der Firma Xilinx[11] verbaut. Die aktuellste Karte ist das NetFPGA SUME, welche Anwendungen mit Anforderungen von 40 und 100 Gigabit pro Sekunde unterstützen können soll.[12]
Dieser Hardware-Traffic-Generator ist bereits für etwas unter 10.000$ zu erwerben, für den akademischen Gebrauch sogar unter 5.000$.[13]

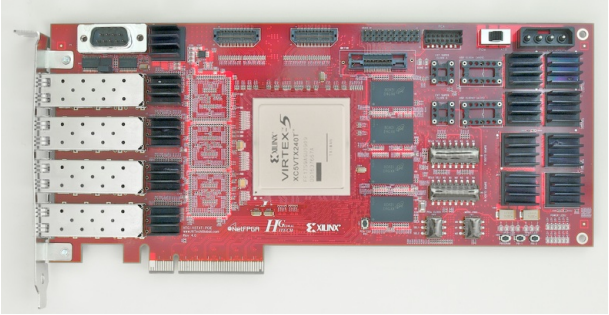

Abbildung 1: NetFPGA-10G[14]

3.2 Software-basiert

Im Gegensatz zu den hardware-basierten Traffic-Generatoren, wird für Software-Generatoren lediglich das Programm und eine ausführbare Maschine benötigt, die das jeweilige Programm umsetzen soll. Jedoch ergeben die Software-Lösungen auch unerwünschte Probleme, die nicht immer sofort zu erkennen sind. Falls nicht anders angegeben, so bezieht sich der Abschnitt der software-basierten Traffic-Generatoren auf die Quelle [1].
Bereits das Erzielen einer gewissen Paketrate mit minimalen Paketgrößen ist stark abhängig vom Softwaredesign des jeweiligen Generators. Da es auch andere Programme auf der auszuführenden Maschine gibt, die für den Ablauf des Systems benötigt werden, muss der Traffic-Generator die Ressourcen mit den anderen Anwendungen teilen. Dadurch sinkt die Performanz, die durch die CPU erreicht werden kann, um eine hohe Paketrate zu erzielen. Somit ist es möglich, dass eine angezielte Paketrate von 150.000 Paketen pro Sekunde schon bei unter 80.000 Paketen seinen Grenzwert erreicht. Dies erfolgte in einem Test mit Pentium IV Maschinen und einem Linux 2.6.15 Betriebssystem, auf denen nicht benötigte Anwendungen ausgeschaltet wurden.
Auch wenn die größtmöglichen Pakete verwendet werden, gibt es Performanzprobleme. So kann die angestrebte Bitrate von 1 Gigabit pro Sekunde bereits den maximal möglichen Durchsatz von 500 Megabit pro Sekunde erschöpfen. Dieser Wert entspricht in der kleinst möglichen Paketgröße einer Paketrate von unter 45.000 Paketen pro Sekunde, also weitaus weniger als die im vorherigen Fall erreichten 80.000 Pakete. Dies liegt an der Struktur der Software-Generatoren, die im Gegensatz zu Hardware-Traffic-Generatoren keine dedizierten Speicherbereiche besitzen, aus denen effizient gelesen oder geschrieben werden kann. Somit muss die Software diese Speicherbereiche teuer umkopieren, bis diese ausgesendet werden können, worunter die Präzision dieser Generatoren leidet.
Doch nicht nur die Konkurrenz um Ressourcen des Systems vermindern die Präzision der Software-Generatoren, sondern auch die Genauigkeit der Zeitauflösung. Dies ist besonders für die *inter-departure time* zwischen Paketen bemerkbar. Falls eine gewisse Paketrate eingestellt wird, für die das Sendeintervall zwischen zwei Paketen kleiner ausfällt als die Genauigkeit der Zeitmessung es erfassen kann, so senden einige Software-Traffic-Generatoren die gesamte Menge der Pakete in einem Schwall beziehungsweise *Burst* aus.
Ein anderer Fall, der bei einem solchen Test womöglich nicht beachtet wird, ist eine längere Wartedauer, bevor ein erneutes Paket versendet wird. Wenn die *inter-departure time* bei einigen Software-Generatoren einen Wert um die 4 Millisekunden annimmt, so könnte der Scheduler des Betriebssystems unter dieser geringen Last den Prozess des Software-Generators von der CPU entkoppeln. In diesem Augenblick wird der Prozess allerdings wieder benötigt, sodass dieser teure Kontextwechsel des Prozesses viel Zeit beansprucht und damit die Paketrate durch die Verzögerung reduziert wird. Aufgrund dessen ist es für Software-Traffic-Generatoren notwendig, diese Phänomene zu bedenken und beispielsweise eine Polling-Funktion einzuführen, die innerhalb kleinerer Zeiteinheiten durchwegs auf neue Informationen hin überprüft und damit die Ressourcen auf der CPU behält.
Obwohl die software-basierten Traffic-Generatoren flexibel sein können und durch Aktualisierungen der Versionen neuartige Protokolle unterstützt werden können, so haben diese jedoch, abhängig ihrer Implementierung, besondere Schwächen in ihrer Präzision und sind damit zumeist inakkurat.

3.2.1 Hardware-unterstützt

Werden Software-Traffic-Generatoren jedoch auf Basis gewisser Netzwerkkarten entwickelt, so gewinnen sie eine höhere Präzision durch unterstützte Funktionen der Hardware. Hierzu zählen genauere Zeitstempel durch Taktraten der Chips auf den Netzwerkkarten oder auch eine Zeitkorrektur im Laufe der Ausführung, die sich besonders bei reinen Software-Generatoren von der tatsächlichen Zeit durch Fehler, einem *Clock Drift*, entfernt.
Ein bekanntes Beispiel in der Open Source Community ist OSNT, ein *Open Source Network Tester*, welcher auf das NetFPGA-10G (Abbildung 1) mit 4 10GbE Schnittstellen basiert. OSNT ermöglicht durch eine Virtualisierung der zugrundeliegenden Hardware, dem *NetV* (Abbildung 2), eine Aufteilung des NetFPGA-10G für sowohl Traffic-Generierung, als auch Traffic-Überwachung.[15]

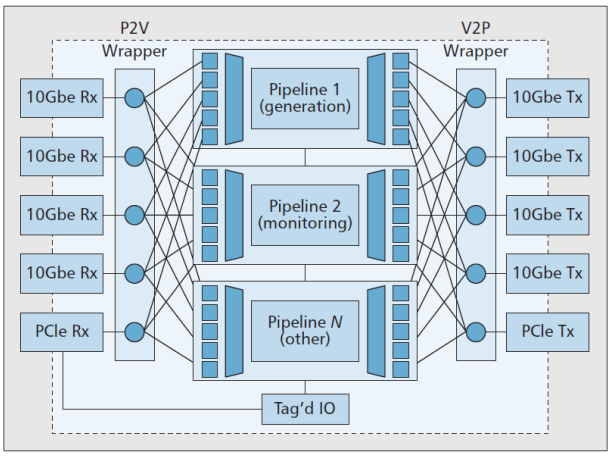

Abbildung 2: NetV-Virtualisierung[15]

Diese Software besteht aus einem *OSNT Traffic-Generator*,

doi: 10.2313/NET-2016-07-1_03

welcher Pakete an allen 4 10GbE Schnittstellen erstellen und empfangen kann. Indem ausgehende Pakete mit Zeitstempel versehen werden, können Informationen bezüglich Verzögerungen und Verlust der Pakete ermittelt werden. Damit kann ein Netzwerkelement wie ein Router oder Switch getestet werden oder ein kleineres Netzwerk, da Ein- und Ausgang der Pakete an derselben NetFPGA-Karte verbunden werden können.
Ebenso ermöglicht OSNT durch seinen *OSNT Traffic-Monitor*, ankommende Pakete an den 10GbE Schnittstellen aufzuzeichnen und an die überliegende Software weiterzuleiten, um diese zu analysieren und weiterzuverwenden. Diese Datenrate kann sehr hoch ausfallen, wenn alle oder die meisten der eingehenden Pakete aufgezeichnet werden sollen. Da die Aufzeichnung einen Flaschenhals für die PCIe Bandbreite verursacht, werden Pakete optional gehashed, also effizient tabellarisiert, und in der Hardware bereits gekürzt. Damit soll eine geringe Verlustrate mit einer hohen Auflösung sowie Präzision des Zeitstempels unterstützt werden.
Durch das *hybride OSNT System* kann auf derselben NetFPGA-Karte mit ihrer *NetV*-Virtualisierungstechnik und den Zeitstempeln eine Charakterisierung des Netzwerkes beziehungsweise des Messgerätes auf *full line-rate* pro Datenfluss erzielt werden. Mit dem Zeitstempel werden ein- und ausgehende Pakete markiert, sodass OSNT im Netzwerk diese Pakete analysieren kann. Da die Zeit jeweils im Paket vermerkt wurde, lässt sich eine Aussage bezüglich des Netzwerkes treffen.
Außerdem bemüht sich die Software als ein *skalierbares OSNT System*, eine große Anzahl von mehreren Traffic-Generatoren und -Überwachungen zu koordinieren. Durch eine Zeitsynchronisierung aller OSNT-Geräte soll ein größeres Netzwerk zuverlässig auf beispielsweise Latenz, Paketschwankungen oder -verlust überprüft werden können.[15]
Eine weitere Software, die die Hardware eines Systems, beispielsweise eines NetFPGAs, ausnutzen kann, ist das OFLOPS, ein offenes Framework für OpenFlow Switch Evaluationen.[16] OpenFlow ist ein System, mit welchem Router durch eine Konfiguration lediglich Regeln ausführen, die ihm durch einen Kontrollrechner beigebracht wurden. Somit wird seine Routing-Logik ersetzt und aus dem System ausgegliedert. Dieses Verfahren kommt in software-definierten Netzwerken, auch *Software Defined Networks*, kurz SDN, vor, indem die Logik, auch *Control Plane*, ausgelagert wird und der Router als einfache Weiterleitungsplattform, auch *Data/Forwarding Plane*, fungiert. Wenn eine höhere Zeitpräzision verlangt wird, so kann OFLOPS die NetFPGA-Hardware nutzen, andernfalls kann diese auch mit Performanzeinschränkungen auf üblicher Hardware mit der Softwarelösung *OpenVSwitch* genutzt werden.[16]
Ebenso ist eine Verbindung der Hardware NetFPGA-10G mit der Virtualisierung durch OSNT und der Host-Software OFLOPS für einen OpenFlow-Einsatz denkbar. Mit diesem sogenannten OFLOPS-Turbo-Host können mehrere Switches in unterschiedlichen Netzwerkstrukturen verbunden werden, um gewisse Aspekte der Netzwerkarchitektur mit hoher Präzision zu messen. Dies ist sowohl mit der *Data Plane* der Switches oder Router möglich, als auch mit der ausgegliederten *Control Plane*.[17]
Ein weiterer hardware-unterstützter Software-Traffic-Generator ist BRUNO, ein Traffic-Generator für einen Netzwerkprozessor, im Speziellen dem Intel IXP2400. BRUNO, welcher für *BRUte on Network prOcessor* steht, basiert auf einer modifizierten BRUTE (Browny and RobUst Traffic Engine) Version. BRUTE wurde dafür designt, um Sendezeiten von Paketen abhängig von gegebenen Traffic-Modellen zu ermitteln. Das System schreibt diese Informationen in einen Speicherbereich, der mit der Paket-verarbeitenden Einheit des Netzwerkprozessors geteilt wird. Der Netzwerkprozessor nutzt diese Daten für die Erstellung der Pakete und sendet diese mit einer geeigneten Zeitschranke, der *inter-departure time*, los. BRUNO soll für eine 1GbE Verbindung die *line rate* ausnutzen können und versendet mit einer hohen Präzision mit einer Paketrate von bis zu 1.488.000 Pakete pro Sekunde.[18]
Um höhere Flexibilität bei gewohnter Präzision durch Hardware zu erhalten, verwendet MoonGen einen etwas anderen Ansatz. MoonGen ist ein Hochgeschwindigkeits-Paket-Generator, welcher 2014 an der Technischen Universität München entwickelt wurde. Durch MoonGen wird die gesamte Paketerstellungslogik zu Benutzer-kontrollierbaren Lua-Scripts übertragen, um ein Höchstmaß an Flexibilität zu erzielen. Dies wird durch die Einbindung von LuaJIT ermöglicht, einem *Just-In-Time*-Compiler für Lua, womit direkt mit Bibliotheken der Sprache C und dessen Structs gearbeitet werden kann. Dadurch können Pakete effizient mit MoonGen erzeugt werden. Außerdem wird das Paketverarbeitungs-Framework DPDK, das *Data Plane Development Kit*, verwendet, um schnell und präzise den Input und Output der Pakete auf unterstützter Hardware durchzuführen. Dadurch wird eine Rate von 14,88 Megapakete pro Sekunde ermöglicht, einer *line rate* von 10GbE mit minimaler Paketgröße (vergleiche Abschnitt 2). Derzeit werden durch DPDK und MoonGen die Hardware-Funktionen auf den Intel-Karten 82599, X540 und 82580 unterstützt. Es können zwar andere Netzwerkkarten verwendet werden, die von DPDK unterstützt werden, jedoch kann dabei die Zeitstempelfunktion und die Datenraten-Kontrolle der Hardware nicht genutzt werden. Durch die Lua-Scripts ist es außerdem möglich, jedes einzelne Paket, das versendet werden soll, zu manipulieren. Dies kann ohne große Performanz-Einbußen stattfinden, da durch Lua eine kleinere Schleife durchlaufen wird, wenn nicht alle Bereiche des Paketes verändert werden. Durch MoonGen wird nur dann mit Performanz bezahlt, wenn es durch aufwändigere Aktionen benötigt wird, beispielsweise dem Ver- oder Entschlüsseln einzelner Felder in eigen definierten Protokollen eines Paketes. Wenn hingegen nur die IP-Adresse vieler vordefinierter Pakete im Puffer geändert werden soll, bevor diese versendet werden, so kann MoonGen mehrere Pakete erstellen und nur das nötige IP-Feld bearbeiten. Andere Paket-Generatoren, die ebenfalls auf DPDK aufbauen wie *Pktgen-DPDK*, können dabei nur langsamer Pakete versenden als MoonGen, da jene in einer komplexeren Ausführung alle möglichen Konfigurationen durchlaufen müssen, obwohl nur ein Feld im Paket bearbeitet werden müsste.[19]

3.3 Eigenschaften im Vergleich

Um nun einige der hier erwähnten Traffic-Generatoren zu vergleichen, wird zunächst nur auf eine der vielen denkbaren Charakteristika eingegangen. Allerdings ist dieser Vergleich nicht einfach, da es keine direkt übereinstimmenden Metriken gibt, auf die Traffic-Generatoren sich reduzieren lassen.[20] Aufgrund dessen werden die durch die Forschungsgruppen spezifizierten Statistiken bei Übereinstimmung aufgezählt und bewertet.

3.3.1 Paketraten

Unter dem Begriff der Paketrate wird der maximale Durchsatz an Paketen mit minimaler Größe verstanden. Dies entspricht in der Regel einer Paketgröße von 64 Bytes. Zusätzlich mit dem Abstand zweier Frames und der Ankündigung eines weiteren Paketes ergibt dies eine Länge von 84 Byte (Abschnitt 2).
Hierbei erzielen die hardware-basierten Traffic-Generatoren wie beispielsweise von Ixia[4] oder Spirent[5] den meisten Durchsatz bis hin zur maximal möglichen *line rate* der jeweiligen GbE-Anbindung. Jedoch sind diese auch die teuersten Ableger in ihrem Bereich.
Die einfachen Software-Traffic-Generatoren erzielen weitaus weniger Pakete in der Sekunde, ob nun die kleinstmögliche Länge oder nicht, da diese Ressourcen mit dem Betriebssystem teilen müssen und sonst weniger optimiert auf die jeweilige Hardware der Rechner sind.
Im Gegenteil dazu sind die neueren Software-Generatoren basierend auf spezieller Hardware, wie zum Beispiel aus dem NetFPGA-Projekt (Unterunterabschnitt 3.1.1) oder unterstützten DPDK-Modellen, wesentlich effizienter und erreichen zudem Paketraten bis zur *line rate*.[19]
Jedoch sind die meisten dieser hardware-unterstützten Software-Traffic-Generatoren für 10GbE-Schnittstellen entwickelt, während die Hardware-Generatoren schon mit 100GbE oder teilweise sogar 400GbE für Datenzentren werben.[21] Einige Testversuche zu Software-Traffic-Generatoren für höhere Raten scheinen aber zuversichtliche Ergebnisse zu liefern. So erreicht MoonGen mit einem Test von 120 Gigabit pro Sekunde bis zu 178,5 Megapakete pro Sekunde.[19] Ein Maximalwert bei dieser Anbindung wäre eine Paketrate von

$$\frac{120 \text{ Gigabits/s}}{84 * 8 \text{ bits}} = \frac{120.000.000.000 \text{ bits/s}}{672 \text{ bits}} =$$

178.571.428, 57 Pakete pro Sekunde.

3.3.2 Sendeintervall

Das Sendeintervall zwischen zwei zu sendenden Paketen wird unterschiedlich gebraucht. Zum einen gibt es den *inter-packet delay*, welcher den minimalen Zeitabstand zwischen allen Paketen entspricht. Hier darf nur dann ein Paket versendet werden, wenn diese Zeit erreicht wird. Damit kann eine gewisse Paketrate eingestellt werden, die durch den Testverlauf eingehalten wird. Dieser Zeitabstand kann nicht kleiner ausfallen als es die maximale Paketrate durch die *line rate*, der physikalischen Grenze, erlaubt, durch welche sie häufig definiert wird. Dieses *inter-packet delay* ist abhängig von der Fähigkeit des Generators, der angebundenen Verbindung und dem dahinterliegenden Netzwerk.
Zum anderen gibt es den Begriff der *inter-departure time*, welche für jedes Paket unterschiedlich konfiguriert werden kann. Diese entspricht der Zeit, nachdem ein Paket versendet wurde, bis ein nächstes Paket gesendet werden darf. Diese kann nicht kleiner ausfallen als das *inter-packet delay*.
Einfache Software-Traffic-Generatoren können eine *inter-departure time* von wenigen Milli- und Mikrosekunden schon nicht mehr garantieren, wodurch bei einem sehr kleinen Sendeintervall die Software zu einem *burst* neigt, also Pakete in einem Schwall so schnell wie möglich versendet werden. Da die Genauigkeit in der Zeitstempelauflösung fehlt, können gezielte Tests für Netzwerkgeräte kaum ausgeführt werden, wenn die Ankunfts- und Sendezeiten der Pakete nicht genau gemessen werden kann.[1] Hardware-unterstützte Software-Generatoren haben diesbezüglich eine wesentlich bessere Chance, die Zeiten bei der Paketmarkierung einzuhalten. Das NetFPGA-10G besitzt eine 6,25 Nanosekunde Zeitstempel-Auflösung mit einer Koordination der Zeitfehler, verursacht durch *Clock Drifts*, durch GPS.[15] Somit dürften die Zeitgrenzen zwischen Paketen bei unterschiedlichen *inter-departure time*-Verteilungen mit hoher Genauigkeit erkannt werden. Lediglich die Hardware-Traffic-Generatoren werben mit einem höchst akkuraten Zeitstempelmodul für Zeitsynchronisationen direkt über Kabel oder ebenfalls GPS von einer Auflösung von 2,5 Nanosekunden für Generierung und Analyse.[6]

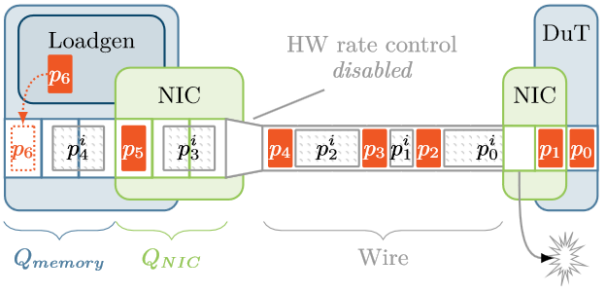

Abbildung 3: MoonGen's Paketverzögerung[19]

MoonGen bietet für gewisse *inter-departure times* eine weitere Strategie an. Anstatt eine Zeit lang zu warten, bis Pakete gesendet werden dürfen, was Fehler in der Zeitabweichung verursachen kann, werden die Lücken zwischen regulären Paketen mit ungültigen Paketen gefüllt. Durch die Veränderung der Größe der ungültigen Pakete kann genau bestimmt werden, wann Pakete versendet werden. Zudem können beliebig komplexe Traffic-Muster dadurch erzeugt werden. Jedoch muss das Testgerät im Netzwerk dazu in der Lage sein, ungültige Pakete zu erkennen und diese, ohne die Paketverarbeitung negativ zu beeinflussen, zu verwerfen. Diese Pakete besitzen eine falsche CRC-Summe im Ethernetframe und, wenn nötig, eine ungültige Länge für kleinere Lücken. Obwohl in der Theorie willkürliche Lücken zwischen Paketen möglich sein sollten, füllen einige Netzwerkkarten Pakete kleiner als 76 Bytes, inklusive Präambel und inter-frame gap (siehe Abschnitt 2), auf, also 8 Byte weniger als das übliche Minimum von 84 Bytes (Unterunterabschnitt 3.3.1). Daher können Lücken von 1 bis 75 Bytes, also 0,8 bis 60 Nanosekunden bei 10 Gigabit Ethernet, nicht erzeugt werden. Die generierte Paketrate bei dieser Alternative muss der jeweiligen *line rate* entsprechen, es müssen also durchwegs Pakete generiert und von der Hardware versendet werden. Die transparenten Pakete p^i in Abbildung 3 sind die ungültigen Pakete, die von der Testhardware frühzeitig verworfen werden.[19]

3.4 Werkzeuge (Tools) in Verwendung

Traffic-Generatoren, die vermehrt in der Praxis zum Einsatz kommen, werden in diesem Abschnitt beschrieben. Sie entsprechen weit verbreiteten Software-Generatoren, die unterschiedliche Einsatzbereiche haben, da sie verschiedenen Arten von Traffic-Generatoren (Abschnitt 2) entsprechen. Viele von ihnen existieren bereits seit einiger Zeit, einige sind etwas moderner.

Mausezahn ist ein Traffic-Generator auf Paketebene, mit welchem beinahe jedes denkbare Paket versendet werden kann. Es wird hauptsächlich für *Voice-over-IP*-Programme, aber auch für Sicherheitstests gegen bestimmte Angriffe, wie einem *Denial of Service*, kurz *DoS*, durch einen *TCP SYN-Flood*, verwendet. Hierbei können Verzögerungen von Paketen zwischen zwei Endgeräten präzise überprüft werden, Tests und Angriffe auf Firewalls, Einbruchserkennungssystemen für Netzwerke oder Netzwerke selbst ausgeübt werden, sowie das Netzwerkverhalten unter besonderen Bedingungen, wie Lasttests oder defekte Pakete, getestet werden. Mausezahn existiert seit 2007, wurde aber im Juli 2013 in das *netsniff-ng toolkit* übernommen.[23] *Netsniff-ng* ist ein Netzwerkkit, welches für Linux frei zur Verfügung steht.[24] Dadurch, dass bei Paketempfang und -versand keine Daten umkopiert werden müssen aufgrund von *Zero-Copy-Buffers*, also gemeinsam verwendetem Speicher (zwischen Kernel- und Userspace), erzielt netsniff-ng eine bessere Performanz als einfache Software-Traffic-Generatoren.[22]
Iperf ist ein weiterer Software-Traffic-Generator auf Paketebene.[25] Iperf ist ein Tool, um die maximale TCP-Bandbreite zu messen. Ebenso können diverse Parameter und UDP-Charakteristika eingestellt werden. Iperf gibt die gemessene Bandbreite, den Delay und Paketverlust wieder, nachdem so viel Traffic wie möglich über eine gewisse Zeit versendet wurde. Da durch Iperf jedoch nicht speziell definiert werden kann, welche Pakete wie versendet werden, werden diese Bandbreiten-Mess-Werkzeuge nicht direkt als Traffic-Generatoren bezeichnet.[1] 2003 wurde Iperf der Version 1.7.0 veröffentlich. Seit 2014 existiert Iperf3, eine Reimplementierung, um eine kleinere, einfachere Basis zu bieten, mit der nun Funktionalitäten verwendbar sind, die es in der ersten Iperf-Version nicht gab[26], wie TCP Retransmit-Informationen, welche nun standardmäßig aktiviert sind, und eine genauere Ausgabe bezüglich CPU-Verbrauch.[27]
Ein weiterer Traffic-Generator, der seit 2014 durch Hardware mit DPDK unterstützt werden kann, ist *Ostinato*, eine Paket-basierte Generator- und Analyse-Software mit benutzerfreundlicher Oberfläche. Mit dieser Open Source Software können Pakete von verschiedenen Datenströmen mit unterschiedlichen Protokollen und Raten versendet werden. Ostinato wurde im April 2010 veröffentlicht.[28]
Zudem gibt es Paket-Traffic-Generatoren, die schon 2007 versuchten, Hardware zur Unterstützung einzusetzen. *D-ITG*, ein *Distributed Internet Traffic Generator*, welcher zu damaliger Zeit neuartige Protokolle unterstützte und auf dem Intel IXP-425 Netzwerkprozessor aufbaut. D-ITG erzeugt IPv4 und IPv6 Traffic, welche durch die *inter-departure time* und Paketgrößen in einem stochastischen Prozess mit mehreren zur Verfügung stehenden Mustern erzeugt wird. Damit kann die Verzögerung, die Hin- und Rückzeit eines Paketes, also die *Round Trip Time*, Paketverlust, Schwankungen und Durchsatz gemessen werden. Zudem ist es möglich, jedes Experiment mit einem *Random Seed* zu versehen, welche die Reproduzierbarkeit eines zufälligen Traffic-Musters garantiert. Ebenso können viele andere Bereiche des Paketes verändert werden.[29]
Ein Traffic-Generator auf Datenflussebene ist Harpoon. Es verwendet eine Menge von Verteilungseigenschaften, die automatisch aus Netzwerkverläufen erfasst werden können, um Datenflüsse mit denselben statistischen Angaben, wie aus den Internetverläufen, zu erstellen. Ebenso kann Hintergrund-Traffic generiert werden, welcher zum Testen von Anwendungen, Protokollen oder Router und Switches genutzt wird. Im Juni 2004 wurde Harpoon für die Öffentlichkeit zur Verfügung gestellt.[30]
Ein Software-Traffic-Generator, welcher auf mehreren Protokoll-Ebenen arbeitet, ist *Swing*. Dieses Programm zeichnet akkurat die Paketinteraktionen von Anwendungen auf und extrahiert Verteilungen von Nutzer-, Anwendungs- und Netzwerkverhalten. Damit kann anschließend sofort Paket-Traffic generiert werden, welcher auf den zugrundeliegenden Modellen der Netzwerkemulations-Umgebung mit gewöhnlichen Protokollstacks arbeitet. Durch die extrahierten Verhaltensweisen kann detailliert *Burst* im Traffic über zahlreiche Zeiträume reproduziert werden. Zudem kann der Benutzer in Swing Annahmen über den Traffic ändern, wie beispielsweise Paketgrößen oder zusätzliche Anwendungen, um einen neuen Traffic zu erzeugen.[31]

4. ZUSAMMENFASSUNG

In dieser Arbeit wurden die Stärken der hardware-basierten und -unterstützten Traffic-Generatoren bezüglich Performanz und Präzision gegenüber den einfachen, aber günstigen Software-Generatoren aufgezeigt. Besonders Generatoren mit hoher Zeitstempelauflösung sind für Delay-Messungen wichtig, da sie akkurat ausgehende, sowie eintreffende Pakete bezeichnen können und damit eine detaillierte und genaue Zeitinformation erhalten. Ebenso sind Generatoren wichtig, die ungewollte Burst-Situationen beim Umsetzen einer gewissen Paketrate durch Einhalten der *inter-departure times* vermeiden. Letztendlich sollte der Traffic-Generator auch im Stande sein zu überprüfen, welcher Traffic durch ihn verursacht und ob das angestrebte Ziel erreicht wurde, also Daten auch überwachen, beziehungsweise aufzeichnen können. Auch der Trend der immer schnelleren Anbindung von 100 GbE oder 400 Gigabit Ethernet sollte von moderneren Software-Traffic-Generatoren präzise auf *line rate* erreicht werden, um eine Alternative zu geschlossenen und meist unflexiblen Hardware-Traffic-Systemen zu bieten.

5. LITERATUR

[1] Alessio Botta, Alberto Dainotti, Antonio Pescapé: *Do You Trust Your Software-Based Traffic Generator?*, in IEEE Communications Magazine, Seite 158-165, September 2010

[2] *Line rate and bit rate*, `http://blog.ipspace.net/2009/03/line-rate-and-bit-rate.html`, zuletzt besucht am 21.09.2015

[3] Spirent: *HOW TO TEST 10 GIGABIT ETHERNET PERFORMANCE*, Rev. B 03/12, März 2012

[4] ImpairNet, `http://www.ixiacom.com/products/impairnet`, zuletzt besucht am 22.09.2015

[5] Spirent TestCenter, `http://www.spirent.com/Ethernet_Testing/Software/TestCenter?docfilter={FF146BE3-9E89-476F-AB1E-3C176C0AB3A4}#Overview`, zuletzt besucht am 22.09.2015

[6] SPIRENT: *MX 100 GIGABIT TEST MODULES*, `http://www.spirent.com/~/media/Datasheets/Broadband/PAB/SpirentTestCenter/Spirent_mX_100G_CFP2_Datasheet.pdf`, Seite 3, Rev. A 05/13, Mai 2013

[7] Spirent TestCenter Module,

http://www.smartechconsulting.com/NG-100G-F2-HyperMetrics-40G-100G-Ethernet, zuletzt besucht am 22.09.2015

[8] NetFPGA, www.netfpga.org, zuletzt besucht am 22.09.2015

[9] Greg Watson, Nick McKeown, Martin Casado: *NetFPGA: A Tool for Network Research and Education*, in 2nd workshop on Architectural Research using FPGA Platforms (WARFP), 2006.

[10] John W. Lockwood, Nick McKeown, Greg Watson, Glen Gibb, Paul Hartke, Jad Naous, Ramanan Raghuraman, Jianying Luo: *NetFPGA - An Open Platform for Gigabit-rate Network Switching and Routing*, in IEEE International Conference on Microelectronic Systems Education (MSE'07), Seite 160-161, Juni 2007

[11] All Programmable FPGAs and 3D ICs, http://www.xilinx.com/products/silicon-devices/fpga.html, zuletzt besucht am 22.09.2015

[12] Noa Zilberman, Yury Audzevich, G. Adam Covington, Andrew W. Moore: 'NetFPGA SUME: Toward 100 Gbps as Research Commodity,' IEEE Micro, vol.34, no.5, Seite 32-41, September-October 2014

[13] NETFPGA-SUME, http://digilentinc.com/Products/Detail.cfm?NavPath=2,1301,1311&Prod=NETFPGA-10G-SUME, zuletzt besucht am 22.09.2015

[14] NetFPGA 10G Board, https://github.com/NetFPGA/NetFPGA-public/wiki/NetFPGA-10G-Board, zuletzt besucht 23.09.2015

[15] Gianni Antichi, Muhammad Shahbaz, Yilong Geng, Noa Zilberman, Adam Covington, Marc Bruyere, Nick McKeown, Nick Feamster, Bob Felderman, Michaela Blott, Andrew W. Moore, Philippe Owezarski: *OSNT: Open Source Network Tester*, in IEEE Network, Seite 6-12, September/Oktober 2014

[16] Charalampos Rotsos, Nadi Sarrar, Steve Uhlig, Rob Sherwood, Andrew W. Moore: *OFLOPS: An Open Framework for OpenFlow Switch Evaluation*, in Volume 7192 der Serie Lecture Notes in Computer Science, Springer-Verlag GmbH Berlin Heidelberg, Seite 85-95, März 2012

[17] Charalampos Rotsos, Gianni Antichi, Marc Bruyère, Philippe Owezarski, Andrew Moore: *OFLOPS-Turbo: Testing the Next-Generation OpenFlow switch*, European Workshop on Software Defined Networks (EWSDN), Budapest, Hungary, 2 Seiten, September 2014

[18] Gianni Antichi, Andrea Di Pietro, Domenico Ficara, Stefano Giordano, Gregorio Procissi, Fabio Vitucci: *BRUNO: A High Performance Traffic Generator for Network Processor*, in SPECTS 2008, Seite 526-533, Edinburgh, UK, Juni 2008

[19] Paul Emmerich, Sebastian Gallenmüller, Daniel Raumer, Florian Wohlfart, Georg Carle: *MoonGen: A Scriptable High-Speed Packet Generator*, http://arxiv.org/abs/1410.3322, 13 Oktober 2014, zuletzt besucht am 23.09.2015

[20] Sándor Molnár, Péter Megyesi, Géza Szabó: *How to Validate Traffic Generators?*, in IEEE International Conference on Communications 2013: IEEE ICC'13 - 1st IEEE Workshop on Traffic Identification and Classification for Advanced Network Services and Scenarios, Seite 1340-1344, Juni 2013

[21] 400GbE Load Modules, http://www.ixiacom.com/products/400gbe-load-modules, zuletzt besucht am 24.09.2015

[22] Mausezahn, http://www.perihel.at/sec/mz/, zuletzt besucht am 24.09.2015

[23] netsniff-ng v0.5.8-rc1, https://github.com/netsniff-ng/netsniff-ng/releases/tag/v0.5.8-rc1, zuletzt besucht am 24.09.2015

[24] netsniff-ng toolkit, http://netsniff-ng.org/, zuletzt besucht am 24.09.2015

[25] Iperf - The TCP/UDP Bandwidth Measurement Tool, http://web.archive.org/web/20081012013349/http://dast.nlanr.net/projects/Iperf/, zuletzt besucht am 24.09.2015

[26] iperf / iperf3, https://fasterdata.es.net/performance-testing/network-troubleshooting-tools/iperf-and-iperf3/, zuletzt besucht am 07.11.2015

[27] iperf3, http://software.es.net/iperf/, zuletzt besucht am 24.09.2015

[28] Ostinato, http://ostinato.org/, zuletzt besucht am 24.09.2015

[29] Alessio Botta, Alberto Dainotti, Antonio Pescapè: *Multi-protocol and Multi-platform Traffic Generation and Measurement*, in INFOCOM 2007 DEMO Session, 2007

[30] Harpoon: A Flow-level Traffic Generator, http://cs.colgate.edu/~jsommers/harpoon/, zuletzt besucht am 24.09.2015

[31] The Swing Traffic Generator, http://cseweb.ucsd.edu/~kvishwanath/Swing/, zuletzt besucht am 24.09.2015