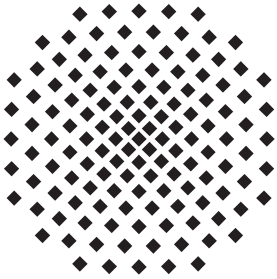


Universität Stuttgart

Fakultät Informatik, Elektrotechnik und Informationstechnik

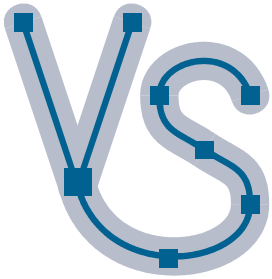
Studienarbeit Nr. 1969

Verzögerungszeiten von Vermittlungsstellen in einem Netzwerkemulationssystem

Mirko Casper

Studiengang:	Informatik
Prüfer:	Prof. Dr. K. Rothermel
Betreuer:	Dipl.-Inf. Steffen Maier
Beginn am:	15.09.2004
Beendet am:	14.03.2005
CR-Nummer:	B.8.2, C.2.2, C.4, D.4.8, I.6.0

Institut für Parallele und
Verteilte Systeme (IPVS)
Abteilung Verteilte Systeme
Universitätsstr. 38
70569 Stuttgart



Zusammenfassung

Im Rahmen des NET-Projekts (Network Emulation Testbed) wurde an der Abteilung Verteilte Systeme ein Netzwerkemulationssystem auf Basis eines PC-Clusters aufgebaut. Das System besteht aus einer Kombination von flexibel konfigurierbarer Hardware und Emulationswerkzeug-Software, die spezielle Netzwerkeigenschaften nachbildet. Es ermöglicht vergleichende Leistungsmessungen von Netzwerkprotokollen und verteilten Anwendungen. Ein Haupteinsatzgebiet des Systems ist die Leistungsbewertung von Netzwerkprotokollen in mobilen ad-hoc Netzen.

Für die Emulation großer Szenarien wird dafür eine große Anzahl kommunizierender Geräte benötigt. Um die Limitierung der Knotenanzahl auf die der verfügbaren PCs aufzubrechen, wurde das Konzept virtualisierter Ressourcen eingeführt. Dadurch lassen sich mehrere virtuelle kommunizierende Geräte auf einem Cluster-PC nachbilden. Im NET-Emulationssystem geschieht die Kommunikation über verschiedene Cluster-PCs hinweg über einen Gigabit-Ethernet-Switch als Vermittlungsstelle. Die Kommunikation zwischen virtuellen Geräten auf demselben PC jedoch über ihnen gemeinsamen Speicher.

Um Festellen zu können, ob durch die unterschiedlichen Kommunikationspfade ein unerwünschter Unterschied von Verzögerungen zum Realverhalten entsteht, werden in dieser Studienarbeit Kommunikationsverzögerungen sowohl über gemeinsamen Speicher als auch den Ethernet-Switch gezielt gemessen.

Dazu wird zunächst die Messumgebung, das Network Emulation Testbed, beschrieben. Nach Erörterung grundlegender Messmethoden, wird das zur Messung erstellte Werkzeug vorgestellt. Durch Einsatz geeigneter Szenarien und Messlasten werden dann mehrere Tests durchgeführt, deren Ergebnisse die Verzögerungen auf den unterschiedlichen Kommunikationspfaden bei verschiedenen Lastsituationen charakterisieren.

Die Messergebnisse werden anschließend mit Verzögerungen verglichen, die bei realer Funkübermittlung auftreten. Damit können Schlussfolgerungen getroffen werden, ob die im NET auftretenden Verzögerungen eine genaue Nachbildung der realen Verhältnisse erlauben. Abschließend werden Hinweise gegeben, die zu einer weiteren Verbesserung der Emulationsgenauigkeit im NET dienen können.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Kapitel 1

Einleitung

1.1 Motivation

Im Rahmen des NET-Projekts (Network Emulation Testbed) wurde an der Abteilung Verteilte Systeme ein Netzwerkemulationssystem auf Basis eines PC-Clusters aufgebaut. Das System besteht aus einer Kombination von flexibel konfigurierbarer Hardware und Emulationswerkzeug-Software, die spezielle Netzwerkeigenschaften nachbildet (emuliert). Es ermöglicht vergleichende Leistungsmessungen von Netzwerkprotokollen und verteilten Anwendungen. Für die Emulation großer Szenarien wird eine große Anzahl kommunizierender Geräte benötigt. Um die Limitierung der Knotenanzahl auf die der verfügbaren PCs aufzubrechen, wurde das Konzept virtualisierter Ressourcen eingeführt. Dadurch lassen sich mehrere „virtuelle“ kommunizierende Geräte auf einem Cluster-PC nachbilden.

Ein Haupteinsatzgebiet des Systems ist die Leistungsbewertung von Netzwerkprotokollen in mobilen ad-hoc Netzen. In diesen Netzen spielen Rundsendevorgänge eine wichtige Rolle. In der Realität erreichen diese Rundsendungen alle Empfänger mit derselben Signalausbreitungsgeschwindigkeit. Im NET-Emulationssystem geschieht die Kommunikation über verschiedene Cluster-PCs hinweg über einen Gigabit-Ethernet-Switch als Vermittlungsstelle. Zur Gewährleistung der Skalierbarkeit erfolgt die Kommunikation zwischen virtuellen Geräten auf demselben PC jedoch über ihnen gemeinsamen Speicher. Zwar ist dies wesentlich effizienter und spart Bandbreite auf dem Switch, jedoch könnte hierdurch ein unerwünschter Unterschied zum Realverhalten eingeführt werden.

1.2 Aufgabenstellung

In dieser Arbeit sollen Kommunikationsverzögerungen sowohl über den Ethernet-Switch als auch über gemeinsamen Speicher gezielt gemessen werden. Hierzu sind geeignete Szenarien unter Berücksichtigung der internen Switch-Architektur sowie Messwerkzeuge, sofern nicht vorhanden, zu entwickeln. Aus den Messergebnissen ist eine Schlussfolgerung abzuleiten, ob die Verzögerungszeiten zwischen den zwei verschiedenen Kommunikationspfaden so ähnlich sind, dass sie dem Realverhalten entsprechen. Falls dem nicht so ist, sollen Lösungshinweise aufgezeigt werden.

1.3 Aufbau der Arbeit

Insgesamt gliedert sich die Arbeit in sechs Kapitel. In Kapitel 2 wird zunächst ein kurzer Überblick über die Funkübertragung bei mobilen ad-hoc Netzwerken gegeben und die Emulationsumgebung, das Network Emulation Testbed, ausführlich beschrieben.

In Kapitel 3 werden die Verzögerungen, die bei Übermittlung von Daten über ein Netzwerk auftreten, näher erläutert und eine Klassifizierung in ihre verschiedenen Anteile vorgenommen. Zusätzlich werden Ansätze diskutiert, die für die Implementierung eines Messwerkzeuges dienen können.

Die Erkenntnisse werden in Kapitel 4 herangezogen, um einen zweckmäßigen Ansatz für das zu erstellende Messwerkzeug zu bestimmen. Die Funktionsweise der erstellten Software wird danach ebenfalls in Kapitel 4 erklärt.

Das Messwerkzeug wird anschließend für die Durchführung von Tests verwendet. Dazu werden in Kapitel 5 mehrere Tests mit verschiedenen Szenarien und Netzwerklasten vorgestellt. Die Messergebnisse werden einer eingehenden Analyse unterzogen und Ursachen für die Entstehung der Verzögerungen erläutert.

Ein Vergleich der Ergebnisse mit den Verzögerungen, die bei Funkübertragung entstehen, wird im abschließenden Kapitel 6 gezogen und Hinweise auf mögliche Lösungen zur Vermeidung von unerwünschten Effekten bei der Emulation gegeben.

Kapitel 2

Grundlagen

2.1 Mobile ad-hoc Netze

Ein mobiles ad-hoc Netzwerk (MANET) verbindet mobile Computer. Es entsteht durch eine Anzahl von Rechnern, die gleichberechtigt miteinander kommunizieren, ohne auf weitere Infrastruktur oder Koordinierungsstellen zurückgreifen zu müssen. Eine Anbindung an ein fest installiertes Netz ist nicht nötig. Die Signalübermittlung erfolgt drahtlos, meist wird Funkübertragung eingesetzt.

Dadurch unterscheidet es sich in seinen Eigenschaften deutlich von traditionellen, kabelgebundenen Rechnernetzen. Der Kommunikationskanal ist durch Funkübertragung unzuverlässig, und es werden nur geringere Übertragungsraten erreicht. Durch die Mobilität der angeschlossenen Teilnehmer ändert sich die Topologie des Netzes ständig und Partitionierungen des Netzwerkes können die Folge sein (vgl. [Rot02],[Sch00]).

Bei Funkübermittlung wird ein gemeinsames Kommunikationsmedium verwendet. Senden zwei Teilnehmer gleichzeitig, so kann der Inhalt für einen Empfänger unter Umständen nicht mehr gelesen werden. Da keine zentrale Koordinierung für den Medienzugriff besteht, muss er, um solche Kollisionen zu vermeiden, selbstständig durch die Teilnehmer geregelt werden. Zwei mögliche Lösungsansätze bieten sich dafür an: Das Verhindern von Kollisionen, bzw. zumindest das Herabsetzen der Wahrscheinlichkeit dafür, oder eine Kollisionserkennung.

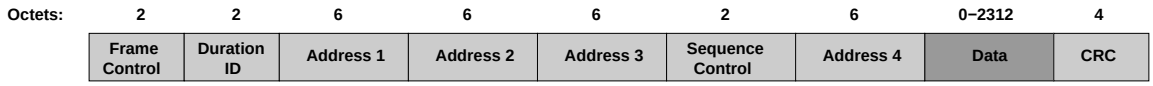


Abbildung 2.1: Grundlegende Struktur eines IEEE 802.11 MAC-Datenrahmens

Im IEEE 802.11 Standard für Wireless LANs sind drei Zugriffsverfahren beschrieben. CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) ist für eine Implementierung verbindlich vorgeschrieben und stellt Funktionen bereit, die die Wahrscheinlichkeit von Kollisionen reduzieren. Eine Erweiterung dazu stellt das CSMA/CA Verfahren mit RTS/CTS (Request To Send/Clear To Send) dar, das zusätzliche Funktionen zur Kanalbelegung enthält. Das dritte Verfahren, die Point Coordination Function (PCF), wird für den Betrieb im Infrastrukturmodus mit zentraler Koordination eingesetzt. Die ersten beiden Verfahren kommen in MANETs zum Einsatz und werden unter dem Begriff Distributed Coordination Function (DCF) zusammengefasst. Die Verfahren lassen sich nach dem OSI-Schichtenmodell der Sicherungsschicht zuordnen, die insgesamt für eine zuverlässige Rahmenübermittlung zwischen benachbarten Stationen zuständig ist (siehe [Tan96]). Der Nutzlast wird in dieser Schicht dazu ein Rahmenkopf vorangestellt, der die in Abb. 2.1 dargestellten Felder umfasst. Die Länge beträgt

inklusive der Prüfsumme am Ende des Rahmens 24 Oktette. Aus der maximalen Länge der Nutzlast von 2312 Oktetten folgt damit eine maximale Rahmenlänge von 2336 Oktetten.

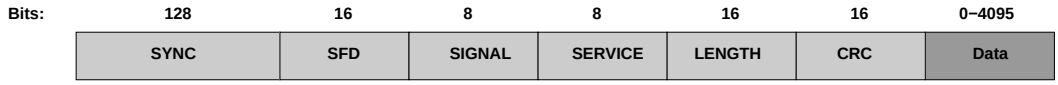


Abbildung 2.2: DSSS-Rahmenformat der Bitübertragungsschicht nach IEEE 802.11

Auf der Bitübertragungsschicht stehen zur Übertragung von Rahmen nach IEEE 802.11 zwei Varianten zur Verfügung. Frequency Hopping Spread Spectrum (FHSS) verwendet ein Frequenzsprungverfahren, um Störungen auf einer bestimmten Frequenz zu vermeiden. Häufiger eingesetzt wird Direct Sequence Spread Spectrum (DSSS), das auf Bandspreizung nach den CDMA-Verfahren (Code Division Multiple Access) basiert. Das hierzu verwendete Rahmenformat zeigt Abb. 2.2. Die maximale Länge eines Rahmens beträgt auf der Bitübertragungsschicht inklusive der Rahmenköpfe damit 2360 Oktette.

Die Reichweite der Funksignale beträgt bis zu 30 m in Gebäuden und 300 m im Freien bei Sichtverbindung der Kommunikationspartner.

Durch die Funkübertragung haben alle Sendevorgänge Rundsendecharakter. Alle Teilnehmer in Empfangsreichweite empfangen unabhängig von der Zieladresse alle gesendeten Rahmen. Der Empfangszeitpunkt eines Rahmens ist dabei, abgesehen von den Differenzen durch unterschiedliche Entfernungen vom Sender, für alle Teilnehmer gleich.

2.2 Network Emulation Testbed

Das Network Emulation Testbed (NET) der Abteilung Verteilte Systeme an der Universität Stuttgart ist ein flexibel vernetzter Linux PC-Cluster mit 64 Knoten, in dem beliebige Netztopologien und deren Eigenschaften nachbildet werden können. Damit bietet das NET eine ideale Analyseumgebung für verteilte Anwendungen und Netzprotokolle [HR02]. In diesem Abschnitt wird ein Überblick über die im Network Emulation Testbed eingesetzte Hard- und Software gegeben.

2.2.1 Hardware

Der NET-Cluster besteht aus 64 gleichen Knoten, die über zwei Ethernet Netzwerke und zwei Vermittlungsstellen miteinander gekoppelt sind. Das erste Netz dient für Verwaltungs- und Administrationsaufgaben und besitzt eine Bandbreite von 100 Mbps. Das zweite Netz ist das eigentliche Testnetz, das über einen Foundry FastIron II+ Switch alle Knoten mit einer Bandbreite vom 1 Gbps vollduplex verbindet (siehe Abb. 2.3).

Knoten

Jeder Knoten des Clusters besitzt die gleiche Ausstattung. Ausgerüstet ist er mit einem 2,4 GHz Intel Pentium 4 Prozessor, einem Hauptspeicher von 512 MB und einem Frontside-Bus mit 133MHz Taktrate. Das Administrationsnetz wird über eine Intel EtherPro Netzwerkkarte angesprochen. Die Anbindung an das Testnetz findet über eine Altima AC9100 Gigabit Ethernetkarte statt, die über den 33MHz, 32bit PCI-Bus and den Rechner angeschlossen ist.

Die Umsetzung einer Netzwerktopologie für ein Emulationsszenario wird mit Hilfe des Switch erreicht [HM04]. Verschiedene VLANs bilden dabei die Kommunikationskanäle der beteiligten Knoten. Ein VLAN mit nur zwei Knoten stellt somit eine exklusive Verbindung zwischen den Knoten dar. Ein

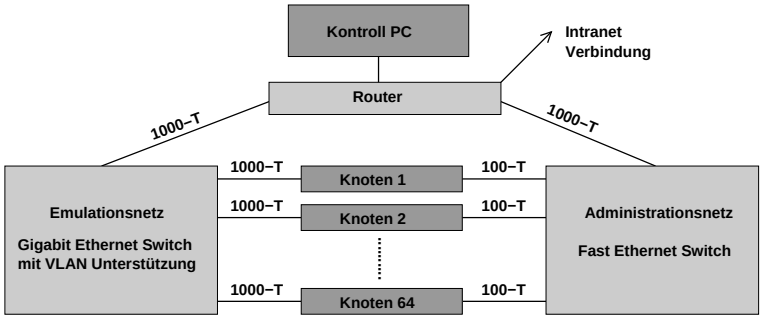


Abbildung 2.3: NET Hardware-Architektur

gemeinsamer Kommunikationskanal wird durch ein VLAN gebildet, das mehrere Knoten umfasst. Jeder physische Knoten entspricht dabei genau einem emulierten Knoten.

Switch

Die Knoten des NET sind gekoppelt über einen Foundry FastIron II+ Switch von Foundry Networks, der in [Fou03], [Fou04b], [Fou04a] und [Fou04c] beschrieben wird. Der Switch ist modular in einer zweistufigen Architektur aufgebaut. Eine Backplane dient zur Aufnahme von bis zu acht Steckplätzen, die über die Chassis Crosspoint Switch Fabric verbunden werden. Die Steckplätze können verschiedene Interface Module aufnehmen. Im NET werden 1-Gbps-Module verwendet, die über acht Ports verfügen und im Folgenden auch als Portgruppe bezeichnet werden. Alle acht Portgruppen mit jeweils acht Ports verbinden die 64 Knoten des Clusters.

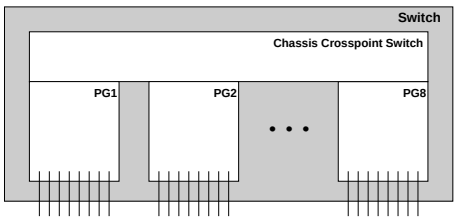


Abbildung 2.4: Grundsätzlicher Aufbau des Switch

Der Switch arbeitet nach dem store-and-forward Prinzip. Alle Rahmen werden nach dem Empfang zunächst in einen Zwischenspeicher abgelegt und bei freier Verbindung zum Zielport wieder ausgelesen und weitergeleitet.

Interface Modul Ein Interface Modul dient zur Verbindung zwischen der Chassis Crosspoint Switch Fabric und den physisch angeschlossenen Knoten. Der logische Aufbau eines Interface Moduls wird beim FastIron II+ Switch als IronCore-Architektur bezeichnet. Abb. 2.5 stellt die Architektur eines Interface Moduls dar.

Physical Ports & Multiport-MAC Die Physical Ports stellen die physische Verbindung zwischen den Knoten und dem Switch her und sind an die Multiport-MACs angebunden. Der Multiport-MAC (Media Access Controller) dient dazu, einen kontinuierlichen Datenfluss zwischen den Physical Ports und dem Packet Processor herzustellen und übernimmt zusätzlich die Aufgabe des Kodierens und Dekodierens

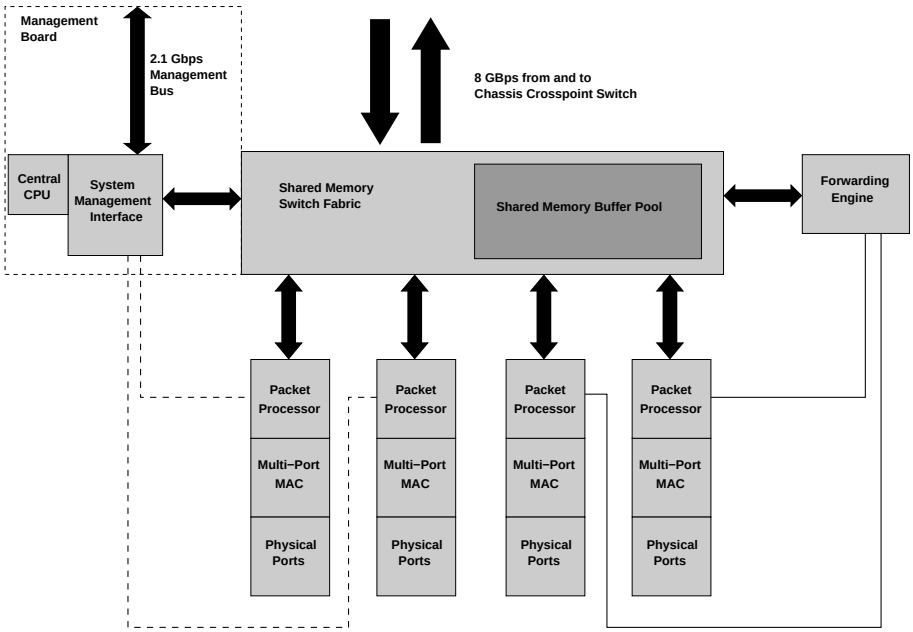


Abbildung 2.5: Architektur einer Portgruppe

von Rahmen und der Verifikation der Prüfsummen ankommender Rahmen. Für ausgehende Rahmen wird die Prüfsumme berechnet und dem Rahmen angehängt.

Packet Processor Angeschlossen an den Multiport-MAC ist der Packet Processor. Der Packet Processor übernimmt die Aufgabe, ankommende Rahmen zu analysieren und für die Weiterleitung vorzubereiten. Für die weitere Bearbeitung im Switch wird dem Rahmen ein interner Header vorangestellt. Teil des Headers ist der „forwarding identifier“ (FID). Neben Prioritätsinformationen beinhaltet der FID alle Ausgangsports, an die der Rahmen im weiteren Verlauf ausgeliefert werden muss, sowie evtl. benötigte VLAN IDs. Nach Anfügen des zusätzlichen Headers wird ein ankommender Rahmen an die Shared Memory Switch Fabric übergeben.

Shared Memory Switch Fabric Die Shared Memory Switch Fabric hat die Aufgabe, Rahmen im Shared Memory Buffer Pool zu speichern und von dort auszulesen. Die Shared Memory Switch Fabric ist verbunden mit der Forwarding Engine, dem Packet Processor und der Chassis Crosspoint Switch Fabric. Wird ein Rahmen empfangen, legt die Shared Memory Switch Fabric ihn im Shared Memory Buffer Pool ab. Sie informiert die Forwarding Engine über die Speicherung und übergibt folgende Informationen:

- Speichernummer des Rahmens
- Forwarding Identifier
- Priorität des Rahmens
- Auslieferung einer Kopie an das Management Modul benötigt

Die Forwarding Engine verwendet die Informationen, um den Rahmen eindeutig identifizieren zu können und alle Zielports, und damit die Ausgangsqueue, bestimmen zu können. Unter Kontrolle der Forwarding Engine sendet die Shared Memory Switch Fabric ausgehende Rahmen an die Zielports und löscht anschließend die Rahmen aus dem Shared Memory Buffer Pool.

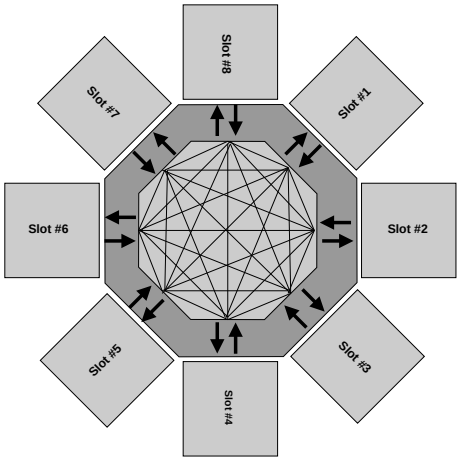


Abbildung 2.6: Chassis Crosspoint Switch Fabric

Shared Memory Buffer Pool Der Shared Memory Buffer Pool stellt Speicherplatz für Rahmen bereit. Er verfügt über eine Bandbreite, die es ermöglicht, Rahmen mit einer Rate zu speichern und auszulesen, die der Gesamtsumme der Bandbreite aller angeschlossenen Packet Processors und der Chassis Crosspoints Switch Fabric entspricht.

Forwarding Engine Die Forwarding Engine ist verantwortlich für die Speicherverwaltung innerhalb der Portgruppe und übernimmt zusätzlich weitere Aufgaben:

- Verwaltung der Warteschlangen
- Verwaltung von QoS (Quality of Services)
- Prioritätsverwaltung für ein- und ausgehende Rahmen
- Ratenlimitierung

Nach Erhalt eines Rahmens wird die Forwarding Engine von der Shared Memory Switch Fabric informiert und verwendet die übergebenen Informationen, um die Zieladressen und damit die Ausgangswarteschlange bestimmen zu können. Die übergebene Speichernummer eines Rahmens wird dann der Ausgangswarteschlange angehängt. Ist der Kopf der Warteschlange erreicht, instruiert die Forwarding Engine die Shared Memory Switch Fabric, welcher Rahmen auszuliefern ist und an welche Ports er weitergeleitet werden muss.

System Management Interface Das System Management Interface verbindet das Interface Module mit dem Management Bus. Über den Management Bus sind alle Interface Module mit dem Management Module verbunden, das Verwaltungsaufgaben übernimmt. Dazu gehören zum Beispiel Konfigurationsänderungen, die über den Management Bus an das Interface Modul weitergegeben werden. Weiterhin können über diese Schnittstelle Statistiken über den Netzwerkverkehr und Rahmen zur Analyse an das Management Module weitergegeben werden.

Chassis Crosspoint Switch Fabric Die Chassis Crosspoint Switch Fabric stellt die Verbindung zwischen den Portgruppen dar und ist in Abb. 2.6 abgebildet. Sie ermöglicht die Weiterleitung von Rahmen,

die nicht allein in einer Portgruppe ausgeliefert werden können. Dazu verbindet sie alle Portgruppen voll vermascht mit einer Bandbreite von 8 Gbps in beiden Übertragungsrichtungen. Dadurch wird eine nichtblockierende Weiterleitung zwischen den Portgruppen ermöglicht.

Exemplarischer Weiterleitungsweg Bei Empfang eines Rahmens am physischen Port wird nach Analyse im Packet Processor dem Rahmen ein FID angehängt und über die Shared Memory Switch Fabric der Rahmen im Shared Memory Buffer Pool abgelegt. Aufgrund der Informationen im FID wird in der Forwarding Engine die Speichernummer in die entsprechenden Ausgangswarteschlangen gelegt. Sobald der Rahmen ausgeliefert werden muss, beauftragt die Forwarding Engine die Shared Memory Switch Farbic, den Rahmen an alle Zielports zu übertragen. Liegen Ports in anderen Portgruppen, so wird er über die Chassis Crosspoint Switch Fabric in die Portgruppe übertragen. Der Rahmen wird „on the fly“ an die Chassis Crosspoint Switch Fabric übergeben und von der Shared Memory Switch Fabric der Zielgruppe direkt im Shared Memory Buffer Pool gespeichert. Dort wird er bei Erreichen des Warteschlangenkopfes an den physischen Port übertragen.

Bei Broadcast- und Multicast-Rahmen werden alle Ausgangsports im FID gespeichert und die Speichernummer in alle zugehörigen Warteschlangen eingetragen. Liegen Ports in einer entfernten Portgruppe, wird der Rahmen an jede Zielportgruppe nur einmal übertragen. Die Aufgabe der mehrfachen Weiterleitung an die physischen Ports wird dann von der Forwarding Engine der Zielportgruppe übernommen. Die mehrfache Weiterleitung ist durch einen Zähler realisiert, der die Anzahl der physischen Ports bzw. Portgruppen enthält. Ist der Rahmen an alle erfolgreich versendet worden, ist der Zähler Null, und der Rahmen wird aus dem Shared Memory Buffer Pool gelöscht.

2.2.2 Software

Alle Knoten im Network Emulation Testbed besitzen die gleiche Softwareausstattung. Als Betriebssystem wird Linux verwendet. Die derzeit laufende Distribution ist RedHat 7.3. Um Emulationen durchführen zu können, ist der Kernel um verschiedene Module und Patches erweitert worden und es werden Emulationswerkzeuge verwendet, die im Folgenden näher betrachtet werden.

NETShaper

Der NETShaper dient zur Emulation von Eigenschaften eines Kommunikationskanals. Als mögliche Parameter kann die Bandbreite des Kommunikationskanals verändert, feste und variable Verzögerungen bestimmt und Rahmenverluste emuliert werden.

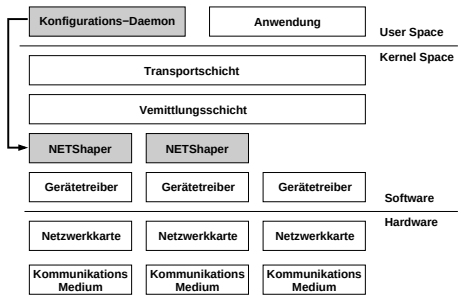


Abbildung 2.7: Einordnung des NETShaper im Protokollstack

Die Implementierung beinhaltet zwei Programme. Ein Kernel-Modul und einen Konfigurations-Dämon. Das netshaper-Modul ist im Kernel als Netzwerkschnittstelle implementiert. Die Emulations-

schicht ist wie in Abb. 2.7 dargestellt im Protokollstapel verankert.

Zur Kommunikation über ein Netzwerk wird eine netshaper-Instanz an eine real existierende Schnittstelle gebunden. Von höheren Schichten wird sie ebenfalls wie eine reale Netzwerkschnittstelle angesprochen und ist somit transparent eingebunden.

Die möglichen Eigenschaften sind dabei folgendermaßen organisiert: Die Bandbreitenlimitierung und Verzögerungen werden im ausgehenden Pfad nachgebildet. Rahmenverluste können auf zwei Arten emuliert werden. Auf dem ausgehenden Pfad kann unabhängig von der Zieladresse ein Rahmenverlust emuliert werden. Bei Empfang besteht zudem die Möglichkeit Rahmen in Abhängigkeit von der Senderadresse zu verwerfen.

Die Kommunikationseigenschaften können während der Initialisierung des Kernel-Moduls oder dynamisch zur Laufzeit geändert werden. Dynamische Änderungen werden von einem Steuerungs-PC, der den Ablauf einer Emulation steuert, per UDP über das Verwaltungsnetzwerk zu allen an der Emulation beteiligten Knoten übertragen. Ein Konfigurations-Dämon empfängt dort die Nachricht und ändert über eine Verwaltungsschnittstelle die Eigenschaften. Die mögliche Änderung der Parameter während der Emulation ist nötig, da sich bei Emulation von mobilen ad-hoc Netzwerken die Eigenschaften des Kommunikationskanals fortlaufend ändern.

VNMux

Der VNMux (virtual node multiplexer) stellt eine weitere Abstraktionsebene innerhalb eines physischen Knotens dar. Bei der Emulation von ad-hoc Netzwerken ist die Zahl der beteiligten Knoten oft größer als die im NET zur Verfügung stehenden 64 Knoten.

Die Skalierbarkeit der emulierbaren Szenarien wird erhöht, indem eine Ressourcenvirtualisierung vorgenommen wird. Dazu werden mehrere Werkzeuge eingesetzt, für Schicht 3 zum Beispiel eine Komponente zur Verwaltung virtueller Routing-Tabellen. Der VNMux implementiert als Teil der Ressourcenvirtualisierung die Funktionalität eines virtuellen Switches auf einem Knoten. Auf einem physischen Knoten (pnode) kann die Emulation mehrerer virtueller Knoten (vnodes) erfolgen, die über ein gemeinsames virtuelles Netz kommunizieren. Mehrere vnmux-Instanzen ermöglichen es, auf einem Knoten mehrere Netze zu emulieren.

Virtuelle Netzwerkschnittstellen werden dabei durch netshaper-Instanzen realisiert. Diese können innerhalb des physischen Knotens (intra-pnode-Kommunikation) über das virtuelle Netzwerk Daten austauschen. Zur Kommunikation mit anderen Knoten, der inter-pnode-Kommunikation, wird der VNMux an eine reale Schnittstelle bzw. eine VLAN-Schnittstelle gebunden.

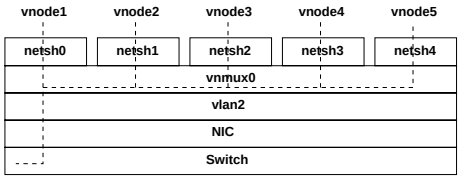


Abbildung 2.8: Einordnung des VNMux im Protokollstack

Implementiert ist der VNMux als Kernel-Modul. Abb. 2.8 zeigt, wie sich das Modul in den Protokollstapel einfügt. Alle netshaper-Instanzen eines virtuellen Netzes sind an eine vnmux-Instanz gebunden. Sind virtuelle Knoten von verschiedenen physischen Knoten in einem Netz zusammengeschlossen, kann eine vnmux-Instanz an eine reale Netzwerkschnittstelle bzw. ein VLAN gebunden werden. Die reale Netzwerkschnittstelle wird dazu in den „promiscuous“-Modus gesetzt, so dass alle Rahmen für die virtuellen Knoten unabhängig von deren MAC-Adresse empfangen werden können.

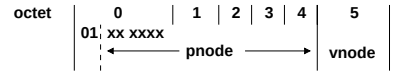


Abbildung 2.9: Aufteilung der MAC-Adressen

Für virtuelle Knoten ist ein Schema zur Vergabe von „privaten“ MAC-Adressen vorgegeben (Abb. 2.9). Alle MAC-Adressen haben die obersten Bits auf 01 gesetzt. Das Adressierungsschema teilt die restlichen Oktette ähnlich zu IP-Subnetzen in zwei Teile auf. Die oberen 5 Oktette bestimmen den physischen Knoten, das letzte Oktett den virtuellen Knoten. Damit ist die Anzahl der virtuellen Knoten derzeit auf 256 pro physischem Knoten beschränkt. Im VNMux wird eine Tabelle fester Länge verwaltet, die eine Umsetzung zwischen der virtuellen Netzwerkadresse und der Netzwerkschnittstelle der netshaper-Instanz ermöglicht. Eine Switching-Entscheidung ist dadurch mit minimalem, konstantem Aufwand unabhängig von der Anzahl angebundener netshaper-Instanzen möglich.

Das vnmux-Modul kann Unicast- und Broadcast-Rahmen weiterleiten. Für die Weiterleitung von Unicasts muss ein Rahmen lediglich in die Eingangswarteschlange der empfangenden netshaper-Instanz eingeordnet werden. Dabei finden keine aufwändigen Kopieraktionen statt, und die Verarbeitungsgeschwindigkeit ist weitgehend unabhängig von der Rahmenlänge. Liegt ein virtueller Zielknoten in einem anderen physischen Knoten, wird der Rahmen über die reale Schnittstelle übertragen.

Die Verarbeitung von Broadcasts ist aufwändiger. Ein Broadcast in einem realen Netz erfordert nur einen Sendevorgang und die benötigten Kopien für alle Knoten werden vom Switch ausgeliefert. Bei einem gemeinsamen Kommunikationsmedium, wie zum Beispiel bei Funkübertragung, erhalten alle Kommunikationspartner den Rahmen durch nur einen Sendevorgang. Die Virtualisierung in VNMux erfordert jedoch eine Auslieferung aller Kopien innerhalb eines Knotens. Dazu müssen Kopien erstellt werden. Rahmenkopien werden im Kernel vom VNMux als Klone erstellt. Es handelt sich dabei um Kopien der Verwaltungsinformationen, die zur Bearbeitung und Auslieferung eines Rahmens im Kernel benötigt werden. Die eigentliche Nutzlast wird nur einmal im Speicher abgelegt und von den Verwaltungsinformationen referenziert. Für alle virtuellen Knoten werden bei einem Broadcast Klone erstellt, und an alle virtuellen Knoten, die an die vnmux-Instanz angeschlossen sind, ausgeliefert. Da die physischen Knoten im NET nur über eine CPU verfügen, findet der Vorgang sequentiell statt.

Die Konfiguration des VNMux findet auf zwei Stufen statt. Beim Laden des Moduls können Optionen angegeben werden, die zur Initialisierung dienen. Eine Änderung der angebundenen netshaper-Instanzen und der realen Netzwerkschnittstelle ermöglicht das Programm „vnmuxcfg“.

Kapitel 3

Messen von Übertragungsverzögerungen

3.1 Anteile der Verzögerung

Bei Übertragung eines Rahmens treten im Verlauf Verzögerungen an diversen Stellen auf, die bei der Messung von Laufzeiten beachtet werden müssen. Dabei lassen sich die Verzögerungen in zwei Klassen einteilen: die Verzögerung ist entweder abhängig von der Rahmenlänge oder unabhängig davon.

Die physische Übertragung über ein Kommunikationsmedium beinhaltet beide Arten. Die Übertragungszeit eines Rahmens bestimmt sich durch die Zeit, die benötigt wird um alle Bits eines Rahmens auf das Kommunikationsmedium zu übertragen, und zusätzlich der Zeit, die das Signal benötigt, um die Distanz zwischen Sender und Empfänger zu durchlaufen.

Der erste Anteil ist die Serialisierungsverzögerung. Sie ist abhängig von der Bandbreite des Kanals und der Anzahl der Bits eines Rahmens. Berechnet werden kann sie nach Gleichung 3.1.

$$\textit{Serialisierungsverzögerung} = \frac{\textit{Anzahl der Bits}}{\textit{Bandbreite des Kanals}} \tag{3.1}$$

Die Signalausbreitungsverzögerung wird durch die zu durchlaufende Strecke zwischen Sender und Empfänger und der Geschwindigkeit, mit der sich ein Signal im Medium ausbreitet, bestimmt. Die Ausbreitungsgeschwindigkeit der Signale wird durch das verwendetete Übertragungsmedium bestimmt. In Tab. 3.1 sind Geschwindigkeiten relativ zur Lichtgeschwindigkeit c im Vakuum (ca. 300.000 km/sec) angegeben. Die Signalausbreitungsverzögerung berechnet sich dann nach Gleichung 3.2 und ist unabhängig von der Rahmenlänge.

$$\textit{Signalausbreitungsverzögerung} = \frac{\textit{Distanz zwischen Sender und Empfänger}}{\textit{Signalausbreitungsgeschwindigkeit}} \tag{3.2}$$

Beide Verzögerungsanteile zusammen geben den Zeitraum an, der für die Übertragung eines Rahmens über einen Kommunikationskanal benötigt wird.

Weitere Verzögerungen entstehen durch die Bearbeitung der Rahmen in den an der Kommunikation beteiligten Komponenten. So werden im Sender und Empfänger und für eine eventuelle Weiterleitung im Switch die Rahmen analysiert und zwischengespeichert.

Die Anteile lassen sich wieder in rahmenlängenabhängige und -unabhängige Teile unterscheiden (vgl. [BNK97]). Kopiervorgänge sind für lange Rahmen aufwändiger und bedeuten einen höheren Rechenaufwand. Andererseits muss für jeden Rahmen zumindest ein bestimmter Anteil Software unabhängig von der Rahmenlänge durchlaufen werden. Die Rahmen werden weiterhin in den Komponenten in Warteschlangen zwischengespeichert bis sie gesendet, empfangen bzw. weitergeleitet werden können. Die Warteschlangenlänge ist dabei von der Last im Netz und der Verarbeitungsgeschwindigkeit eines

Medium	Ausbreitungsgeschwindigkeit	
Thick Coax	0,77c	(231.000 km/sec)
Thin Coax	0,65c	(195.000 km/sec)
Twisted Pair	0,59c	(177.000 km/sec)
Fiber	0,66c	(198.000 km/sec)
AUI Cable	0,65c	(195.000 km/sec)
Luft	c	(300.000 km/sec)

Tabelle 3.1: Signalausbreitungsgeschwindigkeit in diversen Medien

Knotens abhängig. Ist zum Beispiel die Rate der zu empfangenden Rahmen höher als die Bearbeitungsgeschwindigkeit, wird die Warteschlange länger und die Bearbeitung der folgenden Rahmen verzögert sich.

Insgesamt kann die Verzögerung für einen Rahmen bei zwei physischen Knoten über einen Switch durch folgende Anteile angegeben werden. Im Sender wird der Rahmen durch Ablage in eine Ausgangswarteschlange abhängig von der Last im Knoten verzögert. Der Rahmen wird zweimal physisch übertragen, vom Sender zum Switch und vom Switch zum Empfänger. Die Serialisierungs- und Signalausbreitungsverzögerung treten also abhängig von der Distanz und der Übertragungsrate zweimal auf. Bei der Bearbeitung des Rahmens im Switch entsteht eine Verzögerung durch Zwischenspeichern vor der Weiterleitung zum Empfänger. Schließlich wird der Rahmen bei Empfang im Zielknoten zunächst in eine Eingangswarteschlange gelegt bis er ausgeliefert werden kann.

3.2 Konzepte

aktives / passives Messen

Für die Analyse von Laufzeiten müssen Messdaten gesammelt werden. Je nach Art der Messdatengewinnung unterscheidet man die aktive Messung von der passiven Messung. Bei der aktiven Messung werden Rahmen für Messzwecke generiert, um Messdaten zu erhalten.

Eine zweite Möglichkeit ist die passive Messung. Bei der passiven Messung wird vorhandener Netzwerkverkehr verwendet, um Daten für Auswertungszwecke zu gewinnen. Die Messdaten stellen damit einen Mitschnitt von Rahmen eines Netzwerkes dar.

Einweglaufzeit / Umlaufzeit

Die Verzögerung kann, je nach Messweg den ein Rahmen durchläuft, auf zwei Arten angegeben werden. Unterschieden wird zwischen der Umlaufzeit (round-trip-time, RTT) und der Einweglaufzeit (one-trip-time, OTT).

Die Einweglaufzeit wird durch einen Rahmen ermittelt, der von einem Sender zum Empfänger übertragen wird. Der Sendezeitpunkt wird dabei im Rahmen an dem Empfänger übermittelt (Abb. 3.1). Durch Bilden der Differenz des Sendezeitpunkts mit dem Empfangszeitpunkt, kann im Empfänger die Laufzeit ermittelt werden. Eine genaue Messung ist aber nur dann möglich, wenn beide Kommunikationspartner über synchronisierte Uhren verfügen. Sollen kurze Laufzeiten über geringe Distanzen gemessen werden, kann dies nur mit Hilfe spezieller Hardware zur Synchronisation der Uhren erfolgen. Das NTP-Protokoll, bei dem Uhren mit einem genauen Zeitserver abgeglichen werden, reicht in diesem Fall nicht aus [Pax98].

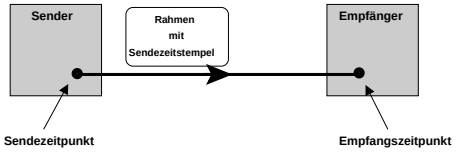


Abbildung 3.1: Messen der Einweglaufzeit

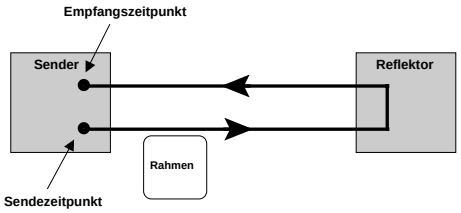


Abbildung 3.2: Messen der Umlaufzeit

Bei der Ermittlung der Umlaufzeit wird ein Testrahmen von einem Sender an einen Zielknoten übermittelt, und von dort wieder an den ursprünglichen Knoten zurückgesendet (reflektiert).

Die Ermittlung der Laufzeit findet dabei durch Bilden der Differenz von Sende- und Empfangszeitpunkt im Sender statt (Abb. 3.2). Synchronisierte Uhren sind also nicht erforderlich. Die Güte der Messung bestimmt sich aus der Genauigkeit der Uhr im Sender. Es wird also die gesamte Laufzeit, der Hin- und Rückweg eines Rahmens, berechnet. Die Laufzeit des Rahmens für einen Weg kann durch Halbieren der Umlaufzeit abgeschätzt werden. Als Voraussetzung für genaue Messungen ergibt sich daraus die Notwendigkeit, dass auf dem Hin- und Rückweg möglichst gleiche Voraussetzungen gegeben sein müssen. Für einen Testrahmen sollte die Last im Netz auf dem Hin- und Rückweg möglichst symmetrisch verteilt sein, um Ungenauigkeiten der Ergebnisse zu vermeiden.

Eine weitere Verfeinerung des Verfahrens kann erreicht werden, indem zusätzlich die Zeit für die Bearbeitung beim Zurücksenden im Testrahmen vermerkt wird, und dadurch genauere Ergebnisse erreicht werden. Die Einweglaufzeit kann dann aus der Umlaufzeit durch Gleichung 3.3 berechnet werden [CPB93].

$$< \textit{Rahmenlaufzeit} > = \frac{< \textit{gesamte Laufzeit} > - < \textit{Zeitdauer zur Reflektion} >}{2} \tag{3.3}$$

Messkonzepte

Aus den Rahmenwegen und der Art der Lastgenerierung lassen sich verschiedene Konzepte ableiten, die als Grundlage für die Messung der Rahmenlaufzeit dienen könnten.

- *Protokollierung des Netzwerkverkehrs im Switch* (passives Messen)
  Protokolle des Switch, die evtl. entsprechende Informationen über Zeitpunkte von Rahmen haben, werden zur Auswertung verwendet. Die Möglichkeit besteht zum Beispiel bei dem in NET eingesetzten FastIron II+ Switch, der über das System Management Interface eine Protokollierung der Netzlast ermöglicht und Kopien einzelner Rahmen zur Analyse ausliefern kann.

- *Round-trip-time allgemein* (RTT/aktives Messen)
  Die Rahmenlaufzeit wird durch Messen der gesamten Zeit, die ein Rahmen benötigt bestimmt. Ein Lastgenerator sendet einen Rahmen, der von einem zweiten Knoten, dem Reflektor, zurückgesendet wird. Die Laufzeit des Rahmens ergibt sich, halbiert man die Gesamtlaufzeit.

- *Minimale Laufzeit + Rahmenlaufzeitunterschiede* (OTT/aktives Messen)
  Kann die minimale Laufzeit eines Rahmens bestimmt werden, so können die Ergebnisse weiter verfeinert werden. Der Lastgenerator sendet dabei Rahmen in einem genau definierten Zeitabstand. Im Empfänger kann aufgrund der minimalen Laufzeit und der Unterschiede zwischen den Rahmen die genaue Laufzeit abgeleitet werden. Lediglich die Rahmenrate und die minimale Laufzeit eines Rahmens muss dem Empfänger bekannt sein.

- *Differentielles Messen über zwei parallele Netzwerke* (OTT/aktives Messen)
  Eine weitere Messung ist durch Laufzeitunterschiede zweier parallel liegender Netze möglich. Wird ein Rahmen von einem Sender gleichzeitig über zwei Netzwerke übertragen, so können aufgrund der Laufzeitunterschiede Aussagen über die unterschiedlichen Weiterleitungseigenschaften gemacht werden.

3.3 Verwandte Arbeiten

Das bekannteste Programm zur Messung von Laufzeiten ist „ping“. Es misst die Umlaufzeit von Paketen. Als Protokoll verwendet es das im IP-Standard festgelegte Internet Control Message Protocol (ICMP). ICMP ist in RFC 793 genauer beschrieben [Pos81].

Zur Messung werden ICMP-Pakete des Typs „Echo Request“ versendet, die von einem Zielrechner mit „Echo Reply“ beantwortet werden. Die Berechnung der Laufzeit findet durch Differenzbildung des Sendezeitpunkts zum Empfangszeitunkt statt. Es wird also die Umlaufzeit berechnet. Die Übermittlung der Nachrichten wird mit IP-Paketen vorgenommen. Daher können Laufzeiten zwischen Knoten gemessen werden, die über mehrere Vermittlungsstellen miteinander verbunden sind.

Die erreichbaren Messgenauigkeiten sind jedoch begrenzt. Problematisch ist vor allem, dass die Wegewahl auf Hin- und Rückweg eines Pakets entkoppelt ist und zu unterschiedlichen Pfaden durch das Netzwerk führen kann. Aufgrund der Implementierung des ICMP-Protokolls als Teil der IP-Schicht ist dieses Verfahren weit verbreitet.

Eine Weiterentwicklung zu „ping“ ist das ebenfalls von Van Jacobson veröffentlichte „pathchar“. Es ermöglicht zusätzlich, die Bandbreiten und Verlustraten auf allen Teilstrecken zu bestimmen [AP99].

Das Programm „traceroute“ kann die Umlaufzeiten zu allen Zwischenstationen zwischen zwei Knoten bestimmen. Dazu werden UDP-Pakete an einen Zielknoten gesendet. Um die Verzögerungen zu den einzelnen Zwischenstationen zu bestimmen, wird dabei die Lebenszeit (Time to live, TTL) im Header des IP-Paketes variiert. Die Lebenszeit gibt die maximale Anzahl von Zwischenstationen an, die ein Paket durchlaufen darf, bevor es nicht mehr weitergeleitet und verworfen wird. Das TTL-Feld ist als ein Zähler implementiert, der bei jeder Zwischenstation dekrememtiert wird. Ist der Zähler abgelaufen wird das Paket verworfen und die ICMP-Nachricht „Time Exceeded“ an den Sender geschickt.

Zur Ermittlung der Laufzeiten zu allen Zwischenstationen werden dann mehrere Pakete versendet, deren Lebenszeit beginnend mit eins kontinuierlich erhöht wird, bis der Zielknoten erreicht wird. Aus den Sendezeitpunkten und den erhaltenen „Time Exceeded“-Nachrichten können dann sowohl alle Zwischenstationen, als auch die Umlaufzeiten bestimmt werden.

Das IP Monitoring System (IPMON) [FML$^+$03] implementiert ein verteiltes Verfahren zur Ermittlung von Laufzeiten. Das System besteht aus drei Komponenten: Mehreren Rechnern zum Sammeln von Paketen, einer Einheit zur Speicherung der Daten und einer Plattform zur späteren Analyse.

Das Sammeln der Pakete findet durch Rechner (monitoring entities) statt, die eine spezielle Hardware (DAG-Karte,[GPMD97]) besitzen. Die Hardware ermöglicht es, Pakete aus einem Netzwerk auszulesen und mit Zeitstempeln zu versehen. Durch Synchronisation mit dem Global Positioning System

(GPS), wird dabei eine Genauigkeit der Zeitstempel von 20 ns erreicht. Um die entstehenden Datenmengen zu beschränken, wird jedoch nicht das komplette Paket gespeichert, sondern lediglich die zur Analyse benötigten Informationen des Paketkopfes.

Die Daten von allen monitoring entities werden dann in einer Datenbank gesammelt und können zu einem späteren Zeitpunkt analysiert werden. Durch Vergleichen der an verschiedenen Stellen gesammelten Daten kann der Weg eines Paketes nachvollzogen werden, und durch die gespeicherten Zeitstempel wird die Berechnung der Einweglaufzeit möglich. Insgesamt wird dabei eine Genauigkeit von 5 μs erreicht. Der Aufwand und der benötigte Speicherplatz sind jedoch aufgrund der hohen Datenmengen erheblich.

Kapitel 4

Messkonzept

4.1 Ansatz

Bisherige Arbeiten befassen sich meist mit Laufzeiten bei Übermittlung über weitere Entfernungen und mehrere Koppelelemente. Die dabei entstehenden Verzögerungen sind größer und variieren durch mehrmaliges Zwischenspeichern. In dieser Arbeit liegt das Augenmerk jedoch auf den sehr kurzen Verzögerungszeiten der Datensicherungsschicht.

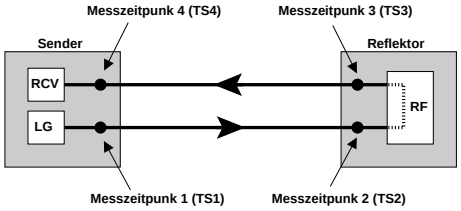


Abbildung 4.1: Messvorgang

Die Ausstattung der NET-Knoten umfasst keine spezielle Hardware zur Uhrensynchronisation. Eine Messung der Einweglaufzeit ist deshalb nicht möglich. Aus den in Abschnitt 3.2 beschriebenen Ansätzen wird daher die Messung mit verfeinerter Umlaufzeit als Ansatz gewählt. Die Zeitstempel werden, wie in Abb. 4.1 dargestellt, bei Umlauf eines Rahmens ermittelt und im Rahmen gespeichert.

$$T = \frac{(TS4 - TS1) - (TS3 - TS2)}{2} \tag{4.1}$$

Die Berechnung der Rahmenlaufzeit T ist dann nach Gleichung 4.1 im Empfänger möglich. Die Laufzeit gibt die Hälfte der Verzögerung abzüglich der im Reflektor aufgewendeten Zeit zum Zurücksenden an. Der Wert ist also der Mittelwert der Einweglaufzeit „im Netz“ für Hin- und Rückweg.

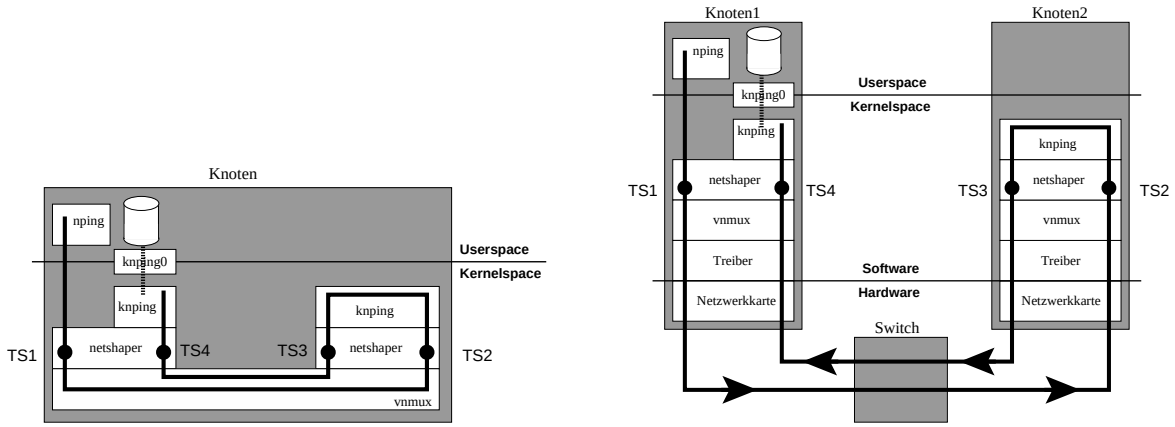
Aus Abb. 4.1 ist bereits zu erkennen, welche Komponenten benötig werden, um Messungen durchführen zu können: ein Lastgenerator (LG), ein Reflektor (RF), ein Empfänger (RCV), sowie eine Komponente zur Zeitnahme und zum Eintragen der Zeitstempel in Rahmen.

4.2 Messwerkzeuge

Für die Messung von Laufzeiten lassen sich grundsätzlich zwei Kommunikationspfade im NET unterscheiden. Zum einen werden Rahmen innerhalb eines physischen Knotens gesendet und über das

vnmux-Modul an den Empfänger geleitet. Im zweiten Fall werden Rahmen über den Switch zwischen physischen Knoten ausgetauscht.

Die bei den Messungen im NET entstehenden Rahmenwege sind in Abb. 4.2 dargestellt. Um für beide Pfade eine Vergleichbarkeit der Ergebnisse zu erhalten, werden die Zeitstempel (TS1–TS4) im netshaper-Modul genommen. Dazu werden für beide Fälle sowohl ein netshaper-Modul als auch ein vnmux-Modul bei jeder Messung verwendet. Dies entspricht auch der Situation bei Durchführung von Emulationen, bei denen bei Kommunikation über den Switch ebenfalls zunächst das vnmux-Modul durchlaufen werden muss.



(a) Virtualisierter Fall (b) Kommunikation über Switch

Abbildung 4.2: Rahmenweg

Die entwickelte Software umfasst mehrere Teile, die die erwähnten Komponenten des vorigen Abschnitts umfassen und in Abb. 4.2 bereits eingetragen sind. Die Erstellung und das Versenden übernimmt der Lastgenerator „nping“. „Nping“ versendet Rahmen auf Schicht 2 des OSI-Modells (MAC-Rahmen) über das Netzwerk. Die anderen Komponenten sind im Kernel-Modul „knping“ zusammengefasst. Die Implementierung wurde mit Hilfe von Informationen aus [WPR+02], [BC02], [RC01], [Her02] und [Ste90] vorgenommen.

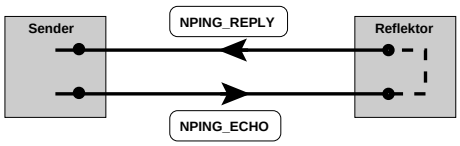


Abbildung 4.3: Protokolltypen

Der Messvorgang findet durch Übermittlung von Rahmen statt, die mit derzeit unverwendeten Protokollkennungen im MAC-Header gesendet werden und im Protokollstack des Betriebssystems für die Verarbeitung an die zuständige Komponente weitergeleitet werden können. Um den Hin- und Rückweg unterscheiden zu können, werden zwei verschiedene Protokollkennungen benötigt. Die NPING_ECHO-Kennung (0x8888) wird für den Hinweg verwendet, der Reflektor kennzeichnet die Rahmen mit NPING_REPLY (0x8889) (Abb. 4.3).

Bei Umlauf eines Rahmens werden mehrere Zeitstempel (TS1–TS4) genommen, die in der Nutzlast des Rahmens gespeichert werden. Im Empfänger werden die Daten in einem Ringpuffer zwischengespeichert und können über ein Zeichengerät (character device) ausgelesen werden (knping0). Die Zeit-

stempel werden durch den Timestamp Counter (TSC) des Pentium 4 Prozessors generiert. Er stellt einen Zähler dar, der mit der Taktgeschwindigkeit des Prozessors inkrementiert wird. Damit erreicht er bei den im NET verwendeten Knoten eine Auflösung von 0,417 ns (bei 2,4 GHz Taktrate der CPU).

$$T\ (\mu s) = \frac{(TS4 - TS1)\ /\ CPU1\_MHZ\ -\ (TS3 - TS2)\ /\ CPU2\_MHZ}{2} \tag{4.2}$$

Die Berechnung der Laufzeit wird damit nach Gleichung 4.2 möglich. Um aus den Werten des Timestamp Counter die Zeitdauer bestimmen zu können, werden die Taktraten *CPUx_MHZ* der beiden Messknoten verwendet.

4.2.1 nping Rahmenformat

Die genommenen Zeitstempel bei Umlauf eines Rahmens werden in der Nutzlast des MAC-Rahmens gespeichert. Die Länge eines Rahmens wird beim Erstellen vom Lastgenerator für eine Messung fest vorgegeben und die Nutzlast wird durch die Zeitstempel lediglich „gefüllt“. Damit ist die Anzahl der maximal erfassbaren Zeitstempel von der festgelegten Rahmenlänge abhängig. Strukturiert wird die Speicherung durch das nping-Rahmenformat. Der grundsätzliche Aufbau des nping-Rahmenformats ist in Abb. 4.4 dargestellt. Der Rahmen setzt sich aus einem Kopf (HEADER) und daran angehängten Zeitstempeln (TIMESTAMPx) zusammen.

Abbildung 4.4: Aufbau eines nping-Rahmens

Rahmenkopf

Der Kopf dient zur Steuerung des Messablaufs und enthält zusätzlich Informationen, die bei Hinzufügen von Zeitstempeln benötigt werden.

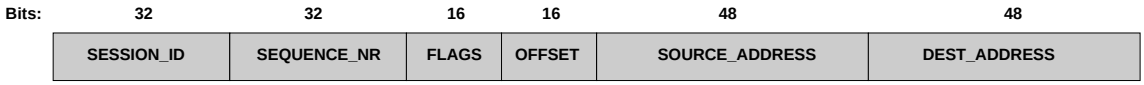


Abbildung 4.5: Aufbau des Rahmenkopfes eines nping-Rahmens

Die Struktur des Rahmenkopfes ist in Abb. 4.5 dargestellt. Das Feld SESSION_ID dient zur Unterscheidung mehrerer gleichzeitiger Messvorgänge. Als Kennung wird derzeit die Prozessnummer verwendet. Das Feld SEQUENCE_NR enthält die Sequenznummer des Rahmens des laufenden Tests. Das FLAGS-Feld wird derzeit nur teilweise verwendet. Es enthält Informationen zur Steuerung des Messablaufs. Die Bits 0–1 geben die Art des Rahmens an. Drei mögliche Rahmenarten sind definiert: ein „Start of Test“-Rahmen (SOT), ein „In-Test“-Rahmen (INT) und „End of Test“-Rahmen (EOT). Der SOT-Rahmen dient dazu, bereits vor dem eigentlichen Test eine gewisse Netzlast zu erzeugen. Diese Rahmen werden im Empfänger nicht gespeichert. „In-Test“-Rahmen werden während des Messvorgangs versendet. Nach Ende des Tests können zudem „End of Test“-Rahmen übermittelt werden, die das Ende des Tests anzeigen. Das Feld OFFSET enthält die Anzahl der Zeitstempel eines Rahmens und wird zur Berechnung bei Eintragen eines neuen Zeitstempels verwendet. Die SOURCE_ADDRESS- und DEST_ADDRESS-Felder enthalten eine MAC-Quelladresse und -Zieladresse. SOURCE_ADDRESS

und DEST_ADDRESS werden zusätzlich zu den im MAC-Header geführten Adressen verwendet und enthalten teilweise unterschiedliche Adressen gegenüber dem MAC-Header. Die Notwendigkeit dafür wird im Abschnitt „Ablauf von Unicast und Broadcast“ erläutert.

Zeitstempel

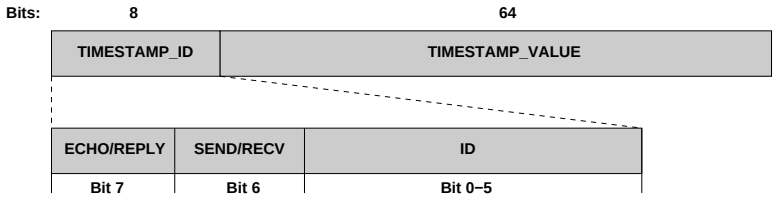


Abbildung 4.6: Aufbau der Zeitstempel eines nping-Rahmens

Die Speicherung der Zeitstempel im Rahmen wird ebenfalls durch eine Struktur festgelegt, die in Abb. 4.6 dargestellt ist. Das TIMESTAMP_VALUE Feld enthält den vom TSC erhaltenen Wert. Das Feld TIMESTAMP_ID dient zu einer eindeutigen Kennung eines Zeitstempels.

Ort der Messung	Symbol	Wert
Lastgenerator	NPING_CLIENT	1
NETShaper	NPING_NETSHAPER	2
Netzwerkkartentreiber	NPING_NETIF	3
Netzwerkkartentreiber (IRQ)	NPING_NETIF_IRQ	4
Reflektor	NPING_REFLECTOR	5

Tabelle 4.1: Mögliche Zeitpunkte der Messung

Die Kennung des Zeitstempels TIMESTAMP_ID ist in mehrere Bereiche aufgespaltet. Bit 7 ist 0 bei Zeitnahme während des ECHO-Rahmens (NPING_ECHO), und 1 bei Zeitnahme während des REPLY-Rahmens (NPING_REPLY). Bit 6 ist 0 bei Zeitnahme während eines Sendevorgangs (NPING_SEND), und 1 bei Zeitnahme während des Empfangsvorgangs (NPING_RECV). Die Bits 0–5 enthalten eine festgelegte Nummer, die den Zeitpunkt der Zeitnahme bestimmt (siehe Abb. 4.1). Die möglichen Werte sind in Tab. 4.1 angegeben. Insgesamt wird durch das Nummerierungsschema eine eindeutige Identifizierung eines Zeitstempel auch nach Testende ermöglicht. Für die im NETShaper genommenenen Zeitstempel TS1–TS4 aus Abb. 4.2 ergeben sich damit die in Tab. 4.2 angegeben Werte für das TIMESTAMP_ID-Feld.

Zeitstempel	Symbole	Kennung (ID)
TS1	NPING_NETSHAPER | NPING_ECHO | NPING_SEND	2
TS2	NPING_NETSHAPER | NPING_ECHO | NPING_RECV	66
TS3	NPING_NETSHAPER | NPING_REPLY | NPING_SEND	130
TS4	NPING_NETSHAPER | NPING_REPLY | NPING_RECV	194

Tabelle 4.2: Kennungen der Zeitstempel für den NETShaper

Ablauf von Unicast und Broadcast

Durch die Berechnung der Einweglaufzeit aus dem Mittel der Umlaufzeit ergibt sich die Forderung, dass die Verzögerungen auf dem Hin- und Rückweg möglichst gleich sein sollten. Erreicht wird dies durch eine konstante Last für den Hin- und Rückweg.

In den Tests in Kapitel 5 werden Unicast- und Broadcast-Rahmen versendet. Für Unicast-Rahmen ist die Forderung immer erfüllt. Bei Broadcast-Rahmen ist das nicht der Fall. Wird ein Rahmen an alle Reflektoren gesendet, können die Rahmen ohne weitere Informationen nur von allen Reflektoren zurückgesendet werden, da der MAC-Header keine Unicast-Zieladresse enthält. Die Last wäre unsymmetrisch für den Hin- und Rückweg, und ungenaue Messergebnisse wären die Folge.

Deshalb wird für Broadcast-Rahmen ein Kommunikationsschema verwendet, das die Forderung erfüllt. Anstatt alle Reflektoren den Rahmen zurücksenden zu lassen, wird das Feld DEST_ADDRESS des nping-Headers bei Generierung eines Rahmens mit der MAC-Adresse des Reflektors belegt, der den Rahmen reflektieren muss. Das Feld wird bei jedem Rahmen neu gesetzt und alle Reflektoren senden bei den Folgerahmen reihum jeweils genau eine Antwort, ebenfalls als Broadcast. Das Verfahren wird im Folgenden als Round-Robin-Broadcast (RRBC) bezeichnet. Um dennoch alle Reflektoren auf einen Rahmen antworten zu lassen, ist es auch möglich das Feld DEST_ADDRESS mit der MAC-Broadcast-Adresse (FF:FF:FF:FF:FF:FF) zu belegen.

4.2.2 nping

Der Lastgenerator „nping“ ist für die Erstellung der Messlast zuständig. Er kann Rahmen mit verschiedenen Längen erzeugen und sie mit verschiedenen Rahmenraten bzw. Bitraten versenden. Der Sendevorgang ist dabei vom Empfangsvorgang im knping-Modul entkoppelt, so dass sich mehrere Rahmen zur gleichen Zeit im Umlauf befinden können.

Zudem können Unicast- und Broadcast-Rahmen versendet werden. Letztere ebenfalls mit dem beschriebenen RRBC-Schema. Die Zielrechner werden dabei durch Angabe der MAC-Adresse(n) bestimmt, die zu verwendende Netzwerkschnittstelle kann ebenfalls angegeben werden.

Die Rahmenraten werden dabei erzeugt, indem zwischen Sendevorgängen der Prozess schlafen gelegt wird. Bei hohen Rahmenraten führt die Granularität der Prozessverwaltung des Linux-Kernels mit 10 ms aber dazu, dass der Prozess zu spät reaktiviert wird. Deshalb wird bei Erzeugen hoher Rahmenraten ein Token-Verfahren eingesetzt. Zunächst wird ein Rahmen versendet und der Prozess die berechnete Zeit zwischen diesem und dem folgenden Rahmen schlafen gelegt. Der Prozess wird dann frühestens nach 10 ms, unabhängig von der angegebenen Zeit, wieder geweckt. Als Folge wäre die Rahmenrate zu niedrig. Deshalb wird die Anzahl der Rahmen berechnet, die während dieser Zeit hätten versendet werden sollen. Diese werden dann umgehend so schnell wie möglich nachgeholt. Das entstehende Muster der Rahmensendevorgänge ergibt insgesamt die geforderte Rahmen bzw. Bitrate.

Des Programm verfügt ebenfalls über die Möglichkeit einen Zeitstempel vor dem Senden in den Rahmen einzufügen.

4.2.3 knping

„Knping“ ist ein Kernel-Modul, das die Komponenten für den Reflektor, den Empfänger, sowie eine Komponente zur Zeitnahme und Eintragen von Zeitstempeln in Testrahmen umfasst.

Komponente zur Zeitnahme

Das Erfassen der Zeitstempel kann im Kernel an verschiedenen Stellen der Netzwerkimplementierung vorgenommen werden. Das Eintragen der Zeitstempel wird dabei durch eine Funktion im knping-Modul vorgenommen. Der Aufruf der Funktion kann aus beliebigen anderen Kernelteilen bzw. -Modulen erfolgen. Dazu wurde ein globaler Funktionszeiger statisch im Kernel definiert. Er ermöglicht es, die Funktion zur Zeitnahme in das knping-Modul zu verlagern und gleichzeitig den Aufruf aus anderen Teilen des Kernels zu entkoppeln. Ist das knping-Modul nicht geladen, ist der Funktionszeiger auf *NULL* gesetzt und wird nicht dereferenziert, d.h. es findet keine Eintragung in den Rahmen statt. Beim Laden des knping-Moduls wird der Funktionszeiger umgesetzt. Durch Dereferenzieren des Zeigers wird die Funktion im knping-Modul aufgerufen und das Eintragen eines Zeitstempels in den Rahmen vorgenommen.

Bei Aufruf wird der übergebene Zeitstempel in den Rahmen eingetragen, wenn die Zieladresse im nping-Header mit der Adresse der Netzwerkschnittstelle übereinstimmt. Wird festgestellt, dass im Rahmen nicht genügend Platz zur Verfügung steht, um einen weiteren Zeitstempel aufzunehmen, wird der Wert verworfen.

Der Typ des Zeitstempels wird festgelegt, in dem aus dem MAC-Protokoll des Rahmens der Protokolltyp ausgelesen und mit den übergebenen Parametern der Funktion verknüpft wird. Die Parameter beinhalten, ob es sich um einen Sende- oder Empfangsvorgang handelt und die Komponente, in der der Zeitstempel genommen wird.

Reflektor

Der Reflektor wird an Netzwerkschnittstellen gebunden, die beim Laden des Moduls angegeben werden können (maximal 32). Daraufhin wird der Protokolltyp NPING_ECHO im Kernel registriert, so dass alle Rahmen des Typs empfangen und an die Reflektoren weitergeleitet werden können.

Die Auslieferung von Broadcast-Rahmen wird im vnmux-Modul mittels Rahmenklonen implementiert. Da der Reflektor ebenfalls im Kernel arbeitet, ist zu diesem Zeitpunkt die Nutzlast für alle ausgelieferten Rahmen in einer gemeinsamen Struktur gespeichert. Deshalb wird im Reflektor zunächst durch den Aufruf der Kernelfunktion *skb_unshare()* eine exklusive Kopie der Nutzlast des MAC-Rahmens angelegt, um weitere Zeitstempel eintragen zu können.

Bei Empfang eines Rahmens werden die Adressen des MAC-Headers umgesetzt. Bei Broadcast-Rahmen wird als Absenderadresse die MAC-Adresse der Netzwerkschnittstelle eingesetzt. Als Zieladresse wird die MAC-Broadcast-Adresse gesetzt. Bei Unicast-Rahmen werden lediglich Quell- und Zieladresse vertauscht. Zusätzlich wird der Protokolltyp des MAC-Headers auf NPING_REPLY (0x8889) gesetzt. Im nping-Header werden lediglich die Quelladresse und die Zieladresse vertauscht.

Beim Laden des Moduls kann über eine Option angegeben werden, ob Zeitstempel beim Empfang und vor dem Senden in den Rahmen eingetragen werden sollen.

Empfänger

Alle Rahmen werden nach Durchlaufen des Netzwerkes im Empfänger gesammelt. Die Empfangs-Funktionalität ist ebenfalls im Kernel-Modul knping implementiert. Die Empfängerfunktion wird beim Initialisieren des Moduls im Kernel eingetragen, indem der Protokolltyp ECHO_REPLY registriert wird. Pro physischem Knoten wird derzeit nur ein Empfänger unterstützt, der beim Laden an eine Netzwerkschnittstelle gebunden wird.

Empfangene Rahmen werden in einem Ringpuffer zwischengespeichert. Dabei wird lediglich der nping-Header und alle Zeitstempel des Rahmens abgelegt, so dass der Speicherbedarf unabhängig von

der tatsächlichen Rahmenlänge bleibt. Der Speicherbedarf bestimmt sich nur aus der Zahl der genommenen Zeitstempel.

Der Ringpuffer wird bei der Initialisierung des Moduls angelegt, und kann durch Verwenden der Kernelfunktion *vmalloc()* eine nahezu beliebige Größe haben, die durch eine Option beim Laden des Moduls festgelegt wird. Die maximale Größe ist derzeit jedoch auf 200 MB begrenzt.

Um die Messdaten aus dem Puffer auslesen zu können, wird als Schnittstelle für Programme ein Character-Device zur Verfügung gestellt. Es kann lediglich gelesen werden, und der Zugriff ist exklusiv auf einen Prozess beschränkt. Das Ergebnis einer Messung ist dann eine Datei, in der alle nping-Rahmen binär als Dump gespeichert sind.

Die Daten können während des laufenden Tests oder gesammelt nach Testende ausgelesen werden. Die Größe von 200 MB ermöglicht es über 2 Millionen nping-Rahmen mit je zwei Zeitstempeln im Ringpuffer zu speichern. Ist der Ringpuffer komplett gefüllt, werden die Daten der nachfolgenden Rahmen nicht gespeichert und gehen verloren.

Der Empfänger bietet ebenfalls eine Option zur Erfassung eines Zeitstempels, die beim Laden des Moduls aktiviert werden kann.

4.3 Auswertungswerkzeuge

Um die Daten im weiteren Verlauf bearbeiten und analysieren zu können, dienen die folgenden Werkzeuge zur Transformation und Aggregation vor der eigentlichen Analyse, die mit Hilfe einer Tabellenkalkulation durchgeführt wird.

nping2ascii

„nping2ascii“ wandelt den nping-Dump in eine Textdatei um. Das erste Feld enthält die Sequenznummer des Rahmens. Getrennt durch ein Leerzeichen enthält die zweite Spalte die MAC-Adresse der Netzwerkschnittstelle des Reflektors. Daran schließen sich, sortiert nach der TIMESTAMP_ID jedes Zeitstempels, alle 256 möglichen Zeitstempel an.

nping2rtt

Das Programm „nping2rtt“ wird zur Berechnung der Rahmenlaufzeit eingesetzt. Als Eingabe wird ebenfalls der nping-Dump verwendet.

Das Programm wird mit 8 Optionen aufgerufen. Die ersten vier geben die Kennungen der Zeitstempel an, die im Verlauf der Messung genommen wurden, die folgenden zwei sind die Taktraten des Lastgenerators und des Reflektors. Daran schließen sich die Namen der Eingabe- und der Ausgabedatei an.

Das Programm berecht dann die Rahmenlaufzeit für alle Rahmen nach Gleichung 4.2 und gibt sie aus. Die erste Spalte enthält die Sequenznummer, die zweite die berechnete Laufzeit.

eval

Das Programm „eval“ kann die berechneten Laufzeiten einer Messung voraggregieren. Das Programm berechnet das arithmetische Mittel aller Rahmenlaufzeiten und die Standardabweichung. Anschließend werden alle Laufzeiten sortiert und der Median und die 1%-, 2%- ,5%- , 95%-, 98%- und 99%-Perzentilen ermittelt. Das Format der Eingabe entspricht dem Ausgabeformat des Programms „nping2rtt“.

eval2_nping.pl

Das perl-Skript „eval2_nping.pl“ wurde für die Auswertung von Test 9 und 10 geschrieben. Es berechnet die Zeitabschnitte bei Messung mit 12 Zeitstempeln aus einer Datei, die mit „nping2ascii“ erzeugt wurde.

Kapitel 5

Messergebnisse

5.1 Messablauf und Messlasten

Um die Kommunikationseigenschaften und Laufzeiten im NET zu bestimmen, werden mehrere Tests durchgeführt. Der Aufbau eines Tests wird durch ein vorher festgelegtes Szenario bestimmt und führt durch Messlasten zu mehreren Messungen, die ausgewertet und analysiert werden.

Allgemeiner Ablauf

Der Ablauf und die Analyse der Messungen gliedern sich in mehrere Phasen. In der ersten Phase werden alle Tests durchgeführt und die Daten auf dem Frontend gespeichert. In einem zweiten Schritt werden alle Messergebnisse voraggregiert, die dann in einem dritten Schritt analysiert und verglichen werden.

Szenarien

Grundsätzlich lassen sich zwei Kommunikationspfade unterscheiden. Durch die Verwendung des vnmux-Moduls wird die Emulation eines virtuellen Netzwerkes ermöglicht. Die Rahmen werden auf nur einem physischen Knoten versendet bzw. empfangen.

Ein zweiter Kommunikationspfad ergibt sich durch Senden von Rahmen über das physische Netz und den Switch. Im Gegensatz zum virtuellen Fall sind dann mehrere physische Knoten an der Kommunikation beteiligt.

Für beide Pfade werden Rahmen von einem Lastgenerator versendet und über den Switch an den Zielknoten bzw. Reflektor ausgeliefert, der sie zurücksendet.

Um eine möglichst genaue Charakterisierung der Kommunikationseigenschaften zu ermöglichen, werden mehrere Szenarien erstellt, die sich in folgenden Eigenschaften unterscheiden:

- Anzahl der beteiligten virtuellen bzw. realen Knoten
- Kommunikationsweg – virtuell vs. physisch ; Portgruppe vs. Chassis Crosspoint Switch
- Kommunikationsform – Unicast bzw. Broadcast
- Hintergrundnetzlast – ja / nein

Im virtuellen Fall werden Rahmenlaufzeiten von Unicasts zunächst ohne weitere Beeinflussung gemessen. Ein zweiter Test wird mit zusätzlicher Hintergrundnetzlast durchgeführt. Der anschließende Test mit Broadcast-Rahmen misst die Einflüsse auf die Laufzeiten bei mehrfacher Auslieferung der Rahmen.

Die drei Szenarien werden analog für eine Kommunikation über das physische Netzwerk wiederverwendet. Aufgrund der zweistufigen Architektur des Switch ergeben sich jedoch zwei mögliche Kommunikationswege und die Anzahl verdoppelt sich. Die Rahmenlaufzeiten werden jeweils innerhalb einer Portgruppe und über den Chassis Crosspoint Switch gemessen.

Insgesamt führen die Szenarien und Kommunikationswege zu neun Tests.

Messlasten

Die Laufzeiten von Rahmen werden weitgehend von der Belastung der Knoten und der Last im Netz bestimmt. Um verschiedene Lastsituationen zu erzeugen, werden in jedem Test mehrere Messungen mit unterschiedlichen Messlasten vorgenommen.

Die Messlasten werden durch zwei Parameter, die Bitrate und Rahmenlänge, variiert. Tabelle 5.1 stellt die verwendeten Messlasten dar, die mit sieben Rahmenlängen und acht Bitraten zu 56 Messungen bei jedem Test führen. Um Messungen bei minimaler und maximaler Last zu erhalten, werden zwei Messlasten verwendet, die nicht über eine Bitrate spezifiziert werden. Als untere Grenze wird eine Rahmenrate von zwei Rahmen pro Sekunde festgelegt. Die sich ergebenden Bitraten sind in Tab. 5.1 in Klammern angegeben. Weiterhin wird eine Messung vorgenommen, bei der mit maximaler Rate gesendet wird.

	Rahmenlänge (Oktette, nur Nutzlast)						
	64	128	256	512	1024	1500	2342
2 Rahmen/s	500 (1024 bps)	500 (2048 bps)	500 (4096 bps)	500 (8192 bps)	500 (16384 bps)	500 (24000 bps)	500 (37472 bps)
10 kbps	1200	1000	1000	1000	300	300	250
100 kbps	12000	5900	3000	1500	1000	1000	500
1 Mbps	120000	59000	30000	15000	7500	5000	3200
10 Mbps	1200000	590000	300000	150000	75000	50000	32000
100 Mbps	2000000	2000000	2000000	1500000	750000	500000	320000
500 Mbps	2000000	2000000	2000000	2000000	2000000	2000000	1600000
max. Rate	2000000	2000000	2000000	2000000	2000000	2000000	2000000

Tabelle 5.1: Anzahl der Testrahmen für die verschiedenen Messlasten

In Tab. 5.1 ist zusätzlich die Anzahl der gesendeten Testrahmen in Abhängigkeit der Bitraten und Rahmenlängen angegeben. Bei der Berechnung wird von einer Messdauer von ca. 60 Sekunden ausgegangen. Da bei geringen Bitraten zu wenig Messdaten gesammelt würden, ist die Anzahl auf mindestens 250 Rahmen erhöht worden. Bei hohen Bitraten ist die Anzahl auf 2 Millionen Rahmen begrenzt, um die entstehende Datenmenge einzuschränken.

Um eine Vergleichbarkeit aller Messergebnisse zu ermöglichen, gibt die Rahmenlänge die Anzahl der Oktette der Nutzlast an, da sich die insgesamt übertragenen Oktette im virtuellen und realen Fall unterscheiden. Im virtuellen Fall ist die Größe des gesamten Rahmens durch Addition der 14 Oktette für den MAC-Header zu erhalten. Im realen Fall steigt die Länge gegenüber dem angegebenen Wert um 30 Oktette. Die Länge des MAC-Headers beträgt 18 Oktette, da zusätzlich 4 Oktette für die Verwaltung der VLANs benötigt werden. Für die physische Übertragung wird dem Rahmen zusätzlich ein Header von 8 Oktetten vorangestellt und 4 Oktette CRC-Informationen angehängt.

Die Rahmenlänge von 2342 Oktetten stellt einen Sonderfall dar. Sie wurde zusätzlich aufgenommen,

da bei Funkübertragung längere Rahmen als bei Ethernet möglich sind. Die Messung mit dieser Rahmenlänge erfolgt jedoch nur im virtuellen Fall. Über das physische Netz ist dies nicht möglich, da der verwendete Switch nicht für eine Übertragung der für Ethernet überlangen Rahmen konfiguriert werden kann (entgegen der Aussage in [Fou04b]).

Messzeitpunkt & Messgröße

Um vergleichbare Werte zu erhalten, ist der Messzeitpunkt für alle Tests gleich. Zeitstempel werden zum erstmöglichen Zeitpunkt genommen, an dem die Rahmen bei allen Tests einen gemeinsamen Verarbeitungspunkt durchlaufen. Bei allen Tests wird eine vnmux-Instanz konfiguriert, an die mindestens eine netshaper-Instanz gebunden ist. Dies gilt auch für die Kommunikation über den Switch. Die Messwerte können damit in allen Tests im netshaper-Modul genommen werden und stellen somit ein normiertes Maß für beide Fälle dar. Weitere Zeitstempel erhöhen die Last des Knotens und führen zu Ungenauigkeiten späterer Messzeitpunkte, weshalb darauf verzichtet wird.

Alle Messwerte geben die halbe Rahmenumlaufzeit wieder, die wie in Abschnitt 4.1 angegeben berechnet und in Mikrosekunden angegeben wird.

Messablauf

Für jeden Test steuert ein eigenes Shell-Skript den Ablauf. Die Messungen eines Tests werden mit allen Messlasten sequentiell ausgeführt. Um Randeffekte zu vermeiden, werden die Knoten für jede Messung neu konfiguriert und die Kernel-Module neu geladen und initialisiert.

Das knping-Modul erlaubt es, die Größe des Datenpuffers beim Laden zu bestimmen. Um Einflüsse der Messung auf die Ergebnisse möglichst gering zu halten, werden 200 MB Hauptspeicher reserviert, so dass alle während einer Messung gesammelten Daten dort abgelegt werden können. Ausgelesen werden die Daten erst nach Messende und komprimiert in einer Datei gespeichert. Nach Beendigung eines Tests werden alle gesammelten Daten auf das Frontend zur späteren Auswertung übertragen.

Methodik der Analyse

Die Auswertung erfolgt in der zweiten Phase nach Beendigung aller Tests. Aus den gesammelten Daten werden die Laufzeiten aller Rahmen einer Messung bestimmt und folgende Werte berechnet:

- Mittelwert
- Standardabweichung
- Median
- Minimum
- Maximum
- Perzentilen: 1%, 2%, 5%, 95%, 98%, 99%

Zusätzlich wird für jeden Messvorgang die erreichte Bitrate, die Anzahl der empfangenen Rahmen und die Messdauer aus den Messprotokollen extrahiert.

In einem dritten Schritt werden die voraraggregierten Daten in eine Tabellenkalkulation importiert und weitere Merkmale abgeleitet. Berechnet werden neben weiteren Mittelwerten die Rahmenverluste, erreichte Rahmenraten, Serialisierungs- und Übertragungsverzögerungen und die Verteilung der Laufzeiten für jede Bitrate und Rahmenlänge.

Mit Hilfe aller Informationen wird schließlich die Analyse durchgeführt und jeder Test auf folgende Merkmale untersucht:

- Auffälligkeiten in der Verteilung der Messwerte
- Maximale und minimale Laufzeiten einer Messung
- Messwertverlauf bei einer Rahmenlänge über alle Bitraten
- Messwertunterschiede bei einer Bitrate und verschiedenen Rahmenlängen
- Verteilung der Laufzeiten einzelner Rahmen innerhalb einer Messung
- Maximal erreichte Übertragungsraten im Vergleich zu den geforderten Raten der Messlasten
- Rahmenverluste durch Überlastung
- Einfluss der Anzahl der beteiligten Kommunikationspartner
- Einfluss der Serialisierungs- und Übertragungsverzögerung
- Einfluss von Rahmenkopie/-klon auf die Laufzeit bei Broadcasts
- Einfluss der höheren Belastung im Netz bei Broadcasts und Tests mit Hintergrundnetzlast

5.2 Virtualisierter Fall

In diesem Abschnitt werden zunächst die Messergebnisse der Tests über das virtuelle Netzwerk ausgewertet.

Da die Rahmen nur auf einem Knoten verarbeitet werden, ergeben sich Rahmenbedingungen, die für alle virtuellen Tests gleich sind. Die Verarbeitung der Rahmen findet zum großen Teil nur im Kernel statt. Insbesondere wird durch den virtuellen Fall kein Aufwand für die Serialisierung und die Übertragung benötigt. Rahmenverluste sind ebenfalls nicht möglich. Bei Broadcast-Rahmen müssen Kopien lokal erstellt und ausgeliefert werden. Im Kernel wird dies durch Klone realisiert, die eine Kopie der Verwaltungsdaten für die Verarbeitung im Kernel darstellen. Die eigentliche Nutzlast ist in einer gesonderten Struktur nur einmal gespeichert.

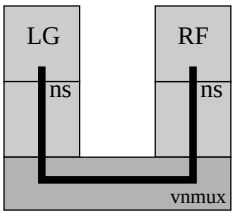
Der Aufbau der Messumgebung ist in den Grundzügen für alle Tests gleich. Es wird nur ein virtuelles Netz mittels einer vnmux-Instanz angelegt. Daran angebunden ist jeweils die für den Test benötigte Anzahl von netshaper-Instanzen, die als virtuelle Netzwerkschnittstellen zum Senden und Empfangen verwendet werden.

5.2.1 Test 1: Unicast

Ziel

Der erste Test über das virtuelle Netzwerk soll die minimal erreichbaren Rahmenlaufzeiten ermitteln.

Aufbau/Ablauf



Das Netzwerk besteht aus zwei netshaper-Instanzen (ns), die über eine vnmux-Instanz (vnmux) miteinander kommunizieren. Gesendet werden von einem Lastgenerator (LG) Unicast-Rahmen, die von einem zweiten Knoten (RF) reflektiert werden. Die Anzahl der virtuellen Knoten (netshaper-Instanzen) ist für die Messergebnisse nicht relevant, da die Switching-Entscheidung mit konstantem Aufwand getroffen wird. Weitere (nicht sendende) virtuelle Knoten würden die Testergebnisse somit nicht beeinflussen.

Ergebnisse

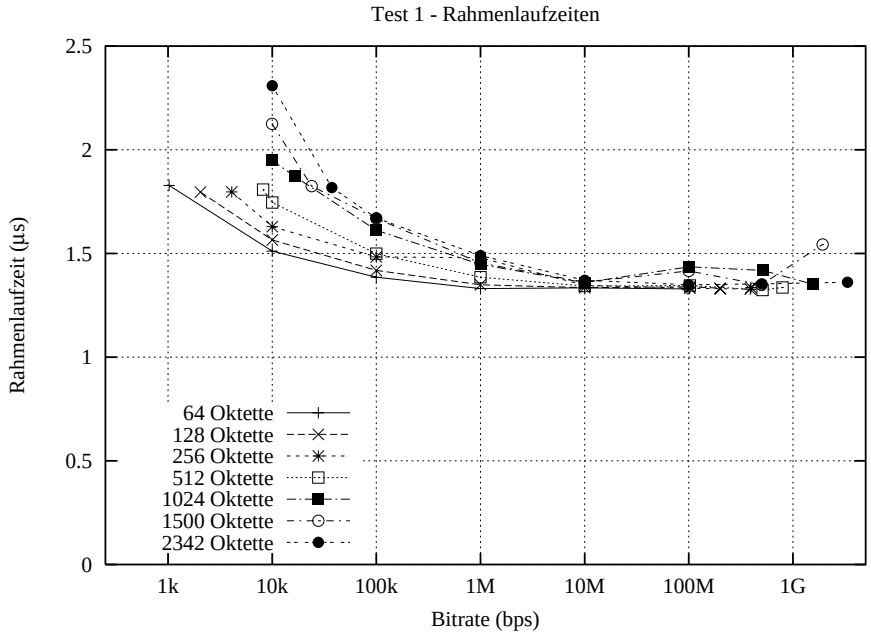


Abbildung 5.1: Test 1 - Rahmenlaufzeiten

In Abb. 5.1 sind die Mittelwerte der gemessenen Rahmenlaufzeiten graphisch dargestellt. Für jede Rahmenlänge ergibt sich aus den Messungen eine Kurve, die den Verlauf bei einer Rahmenlänge mit steigender Bitrate darstellt. Die Bitraten entsprechen dabei den tatsächlich gemessenen Bitraten, die aus den Messprotokollen extrahiert wurden. Im Idealfall sollten alle Messpunkte einer Bitrate vertikal zueinander liegen. Die Messlast mit 2 Rahmen pro Sekunde führt jedoch, abhängig von der Rahmenlänge, zu verschiedenen Bitraten, so dass deren Werte in der Abbildung als nahezu horizontale Anordnung bei 1,82 μs am linken Rand zu erkennen sind. Ferner wird eine Messung mit maximaler Übertragungsrate vorgenommen, die je nach Szenario und Rahmenlänge zu unterschiedlichen Bitraten führt. Damit erklärt

sich auch die nicht vertikale Anordnung der Werte am rechten Rand der Abbildung, die in diesem Test Bitraten bis zu 3 Gbps zeigt.

Betrachtet man die Laufzeiten in Abb. 5.1, so sind bei geringeren Bitraten größere Werte zu erkennen, die mit steigender Rahmenlänge immer deutlicher werden. Über alle Messungen ergibt sich ein mittlerer Wert von 1,51 μs, mit einem Minimum von 1,32 μs bei 512 Oktetten und 500 Mbps, und einem Maximum von 2,31 μs bei 2342 Oktetten und 10 kbps.

Die Laufzeiten sind insgesamt sehr kurz und die Werte über die einzelnen Messungen homogen. Betrachtet man die prozentuale Abweichung ergeben sich zwar deutliche Unterschiede, die jedoch aufgrund der geringen absoluten Werte relativiert werden. Die Latenzzeiten der beiden Messungen mit 2 Rahmen pro Sekunde und 10 kbps sind größer und weisen außerdem, betrachtet man den Unterschied zwischen den Rahmengrößen, eine breitere Streuung der Werte auf. Lange Rahmen führen zu überproportional längeren Laufzeiten. Bei 10 kbps steigen die Werte von 1,51 μs bei 64 Oktetten bis auf 2,31 μs bei 2342 Oktetten. Analog ist dies, jedoch in geringerem Umfang, bei 100 kbps zu erkennen. Bei höheren Bitraten ist diese Streuung nicht mehr vorhanden. Eventuelle Laufzeitdifferenzen über 1 Mbps sind auf Messungenauigkeiten zurückzuführen.

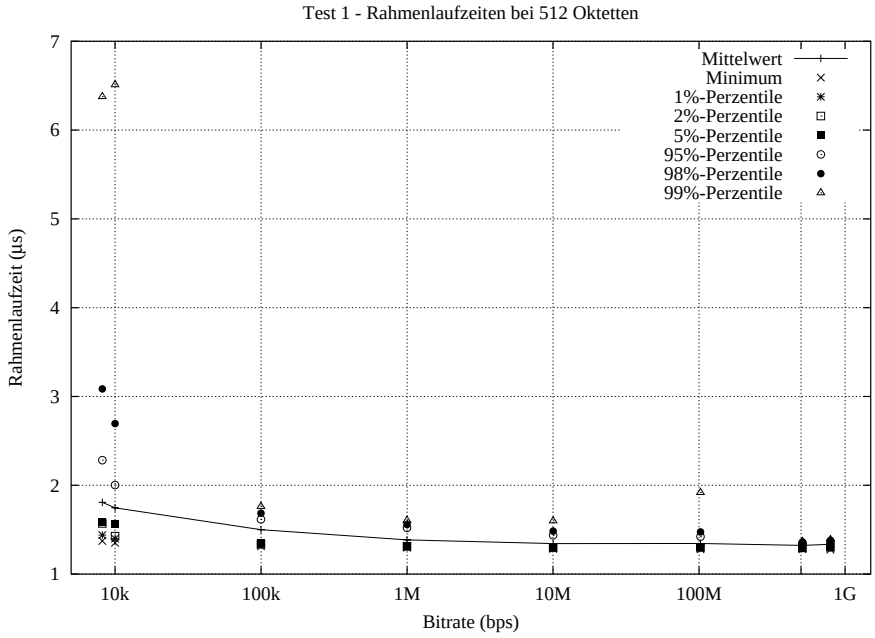


Abbildung 5.2: Test 1 - Verteilung einzelner Rahmenlaufzeiten bei 512 Oktetten

In Abb. 5.2 sind für eine Rahmenlänge von 512 Oktetten der berechnete Mittelwert und ausgewählte Laufzeiten einzelner Rahmen mittels der Perzentilen und des Minimums für alle Bitraten dargestellt. Während bei geringen Raten einzelne Laufzeiten deutliche Abweichungen zeigen, ist ab ca. 1 Mbps fast keine Streuung mehr vorhanden. Die minimale Laufzeit ist nahezu konstant bei 1,31 μs bei allen Bitraten. Die Perzentilen nahe des Minimums (1%) und Maximums (99%) weichen nur gering vom mittleren Wert ab. Die Messwerte weisen also nur eine geringe Streuung auf. Das Maximum ist nicht dargestellt, da die Werte deutlich höher sind und damit die Aussagekraft der Abbildung einschränken würden. Es handelt sich jedoch um wenige Ausreißerwerte, die nur einen geringen Einfluss auf die Ergebnisse haben.

Die in Test 1 gemessenen Bitraten sind in Tab. 5.2 aufgeführt. Die minimale Bitrate bestimmt sich entweder aus der Messlast mit 10 kbps, oder der eventuell niedrigeren Bitrate bei 2 Rahmen pro Sekunde.

Die maximal erreichbare Bitrate ist vom Test abhängig. Die geforderten 500 Mbps können in diesem Test erst ab einer Rahmenlänge von 512 Oktetten erreicht werden. Die Messung mit maximaler Bitrate führt aber zu Raten bis über 3.3 Gbps für lange Rahmen.

	Rahmenlänge (Oktette, nur Nutzlast)						
	64	128	256	512	1024	1500	2342
Minimum (bps)	1022	2044	4089	8187	9670	9980	9993
Maximum (Mbps)	101,8	201,4	396,1	794,4	1559,2	1933,3	3344,9

Tabelle 5.2: Test 1 - Gemessene Bitraten

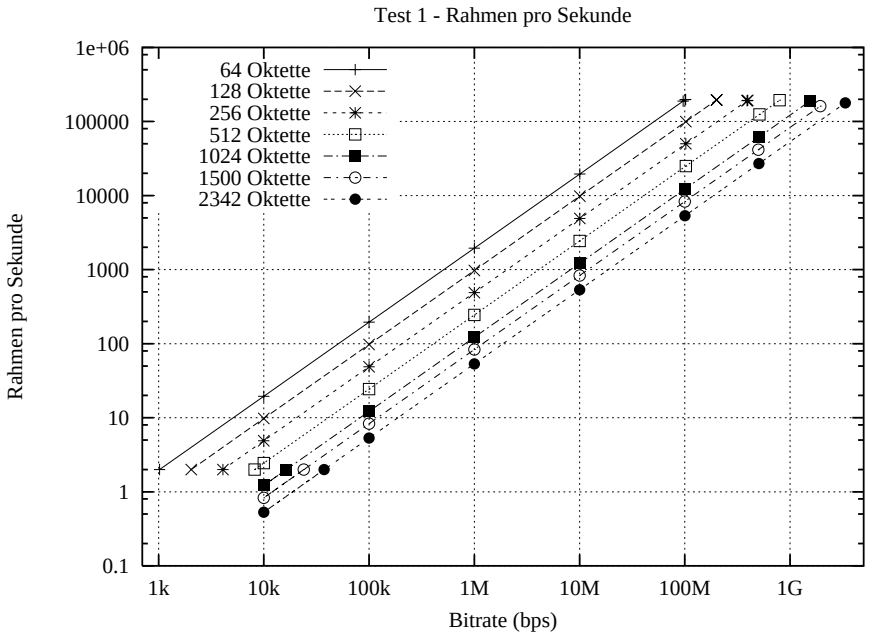


Abbildung 5.3: Test 1 - Rahmen pro Sekunde

Der fehlende Durchsatz bei hohen Bitraten und kurzen Rahmen kann ebenfalls erklärt werden. Berechnet man die Anzahl der ausgelieferten Rahmen pro Sekunde, so ist in Abb. 5.3 zu erkennen, dass sie durch die Rechenkapazität des Knotens beschränkt wird. Unabhängig von der Rahmenlänge können maximal ca. 200.000 Rahmen pro Sekunde gesendet bzw. empfangen werden. Demzufolge werden nicht alle Bitraten der Messlasten bei kurzen Rahmen erreicht. Rahmenverluste durch Überlastung des Knotens treten aufgrund des virtuellen Szenarios nicht auf.

Trägt man die Laufzeiten über die Rahmenrate auf, ergibt sich ein Messwertverlauf nach Abb. 5.4.

Schlussfolgerungen

Die Messwerte zeigen insgesamt konstante Rahmenlaufzeiten an. Geringe Bitraten führen jedoch zu leicht höheren Werten, die auf die Nebenläufigkeit von Linux zurückzuführen sind. Da das System nicht konstant im Netzwerkbereich arbeitet, sondern zwischen den Rahmen andere Aufgaben abarbeitet, entsteht ein höherer Aufwand durch den Taskwechsel, der die auftretenden Effekte zur Folge hat.

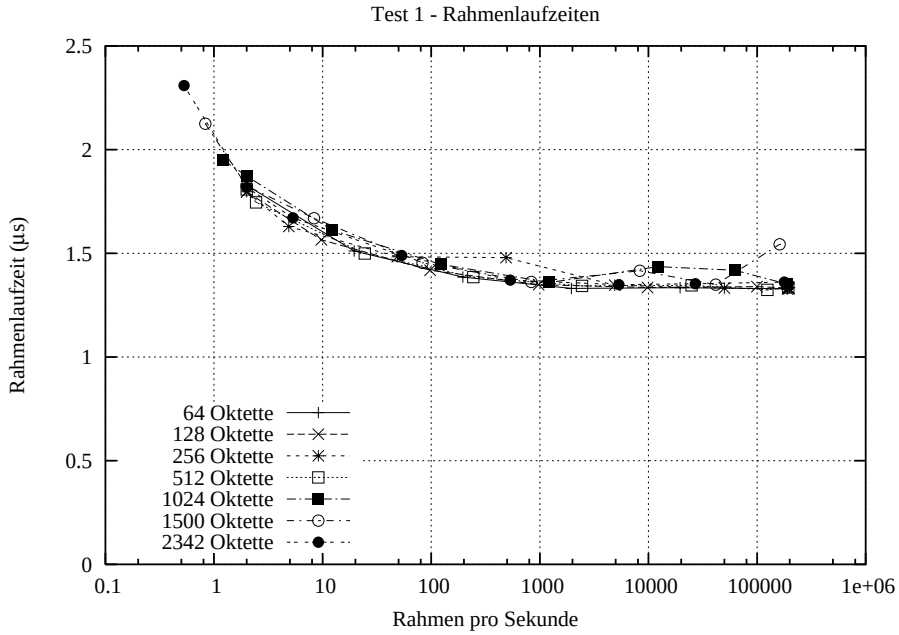


Abbildung 5.4: Test 1 - Rahmenlaufzeiten (über Rahmenrate)

Der geringe Einfluss der Rahmenlänge auf die Laufzeit kann ebenfalls begründet werden. Für jeden Rahmen wird nur einmal Speicher beim Senden im Kernel reserviert, und in der folgenden Verarbeitung finden kaum aufwändige Kopiervorgänge statt.

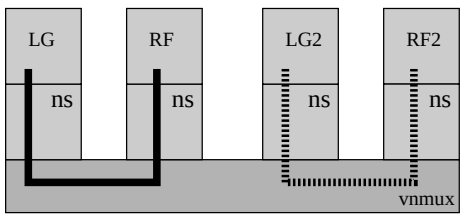
Eine Überlastung des Rechners durch hohe Bitraten beeinflusst die Laufzeit der Rahmen nicht. Als Folge einer Überlastung sinkt die Senderate und damit die maximal erreichbare Bitrate. Rahmenverluste treten nicht auf, da die Rahmen im virtuellen Fall den Knoten (genauer Linux-Kernel) nicht verlassen.

5.2.2 Test 2: Unicast mit Hintergrundnetzlast

Ziel

Zusätzlich zu den Messlasten wird die Last im Netz durch eine Hintergrundlast erhöht und der Einfluss auf die Laufzeiten ermittelt.

Aufbau/Ablauf



Analog zu Test 1 werden Rahmen von einem Lastgenerator an den Reflektor gesendet. Zusätzlich wird unabhängig davon eine weitere Last im virtuellen Netz erzeugt, indem Rahmen von einem zweiten Lastgenerator an einen weiteren Reflektor gesendet werden. Diese Hintergrundnetzlast verwendet bei

der Messung jeweils die gleichen Rahmenlängen und Bitraten zeitgleich. Die Bitrate bzw. Rahmenrate verdoppelt sich demnach im virtuellen Netz.

Ergebnisse

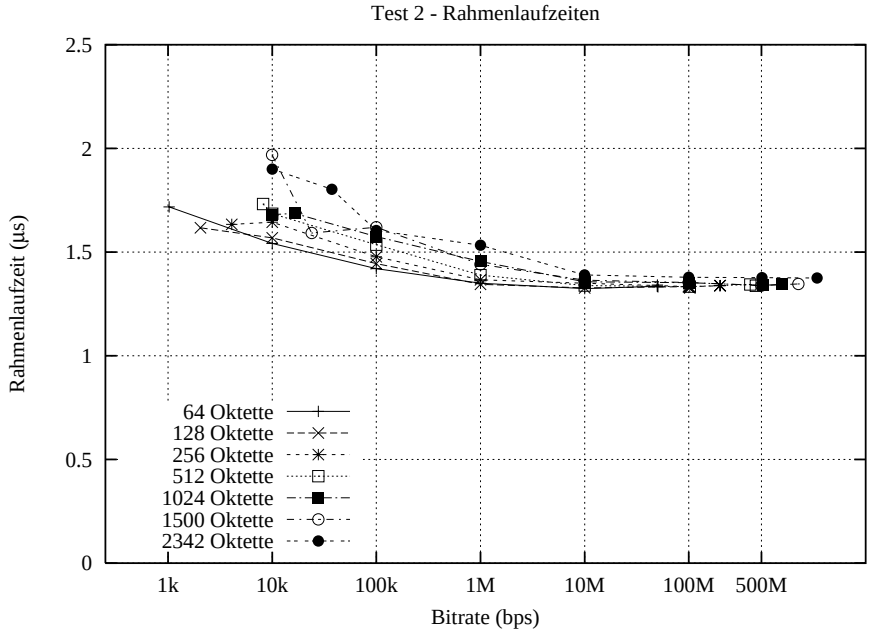


Abbildung 5.5: Test 2 - Rahmenlaufzeiten

Die Rahmenlaufzeiten der Messungen entsprechen weitgehend denen des vorigen Tests. Das Mittel aller Messungen ist 1,46 μs, das Minimum liegt bei 64 Oktetten Rahmenlänge bei 10 Mbps bei einer mittleren Laufzeit von 1,32 μs. Bei 1500 Oktetten und 10 kbps liegt das Maximum von 1,97 μs.

Vergleicht man den Verlauf der Werte in Abb. 5.5 mit denen aus Test 1, so ähnelt sich dieser weitgehend. Längere Laufzeiten und der Einfluss der Rahmenlänge sind bei geringen Bitraten ebenfalls erkennbar. Der Effekt tritt jedoch nur in abgeschwächter Form auf, und die Streuung bei verschiedenen Rahmenlängen bei gleicher Bitrate ist geringer. Ab 10 Mbps ist die Laufzeit erneut unabhängig von der Rahmenlänge.

Die Verteilung einzelner Laufzeiten innerhalb einer Messung ist gering. Der Median und der Mittelwert über alle Laufzeiten einer Messung weichen kaum voneinander ab und liegen nahe am Minimum.

Die größere Belastung des Knotens aufgrund der zusätzlichen Netzlast führt zu einem geringeren Durchsatz bei höheren Bitraten. Die maximale Rate der versendeten Rahmen sinkt auf den halben Wert, so dass maximal 100.000 Rahmen pro Sekunde versendet werden (Abb. 5.6). Die beiden Lastgeneratoren erhalten die verfügbare Kapazität also zu gleichen teilen. Als Folge werden die geforderten Bitraten der Messlast ebenfalls nicht durchgehend erreicht. 100 Mbps werden erst ab 128 Oktetten erreicht, 500 Mbps erst ab 1024 Oktetten.

Schlussfolgerungen

Die Ergebnisse dieses Tests lassen sich weitgehend aus den schon im vorigen Test abgeleiteten Schlüssen erklären. Die Einführung einer Hintergrundnetzlast beeinflusst die Ergebnisse kaum, und die Laufzeiten

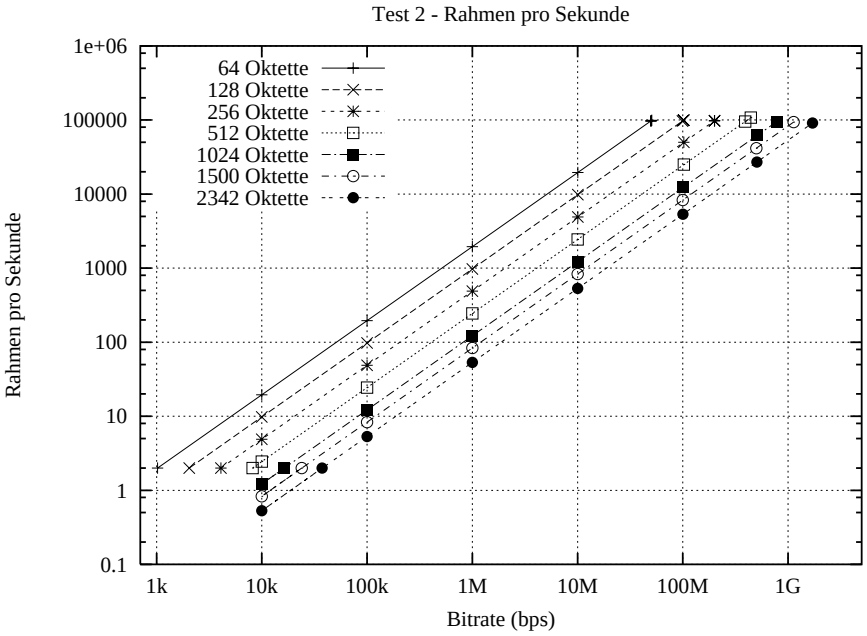


Abbildung 5.6: Test 2 - Rahmen pro Sekunde

stimmen weitgehend mit Test 1 überein.

Die höheren Laufzeiten und die größere Streuung bei geringen Bitraten sind ebenfalls vorhanden, der Effekt wird jedoch durch die zusätzliche Last positiv beeinflusst, da eine insgesamt größere Rahmenrate durch den zusätzlichen Lastgenerator schon bei geringen Bitraten vorliegt. Als Folge finden weniger Taskwechsel statt, so dass weniger „Overhead“ entsteht. Das Mittel aller Messungen ist mit 1,46 μs geringfügig kleiner als der in Test 1 errechnete Wert von 1,51 μs.

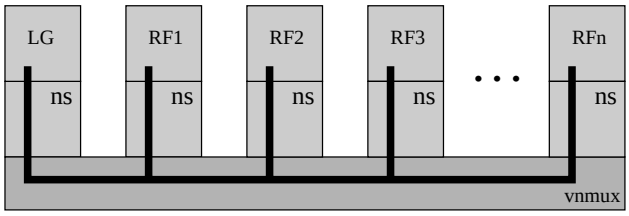
Weiterhin haben, wie bereits in Test 1 festgestellt, sowohl die Bitrate als auch die Rahmengröße insgesamt kaum Einfluss auf die Laufzeit. Die Messergebnisse sind daher bis auf Randeffekte nahezu identisch.

5.2.3 Test 3: Broadcast

Ziel

Mittels des VNMux können mehrere Netzwerkschnittstellen auf einem Knoten emuliert werden. Bei Broadcasts müssen Rahmenkopien im vnmux-Modul erzeugt und an alle virtuellen Netzwerkschnittstellen ausliefert werden. Das Senden von Broadcasts erfordert demnach mehr Ressourcen. Im folgenden Test soll der Mehraufwand gemessen und ausgewertet werden.

Aufbau/Ablauf

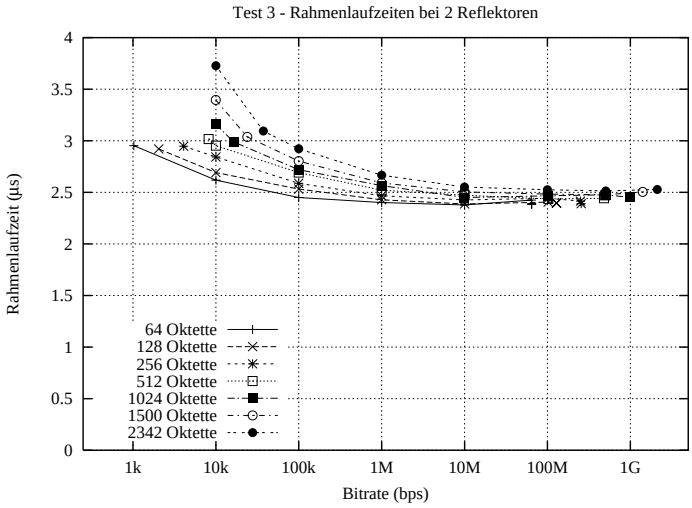


Bei diesem Test sind mehrere netshaper-Instanzen an eine einzige vnmux-Instanz gebunden, und die Testrahmen werden als Broadcast versendet. Als Folge der mehrfachen Auslieferung eines Rahmens an alle netshaper-Instanzen, müssen in der vnmux-Instanz mehrere Rahmenkopien erstellt werden. Die Rahmen werden dann sequentiell ausgeliefert, und die Anzahl der Reflektoren hat demnach einen Einfluss auf die Laufzeit.

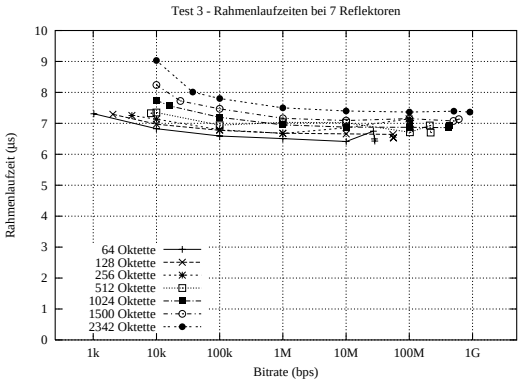
Um den Einfluss des Mehraufwandes zu analysieren, wird der Test mit steigender Anzahl von Reflektoren mehrmals mit allen Messlasten durchlaufen. Die Anzahl wird beginnend mit 2 Reflektoren kontinuierlich bis auf 31 Reflektoren erhöht.

Um möglichst genaue Messwerte zu erhalten, wird das Round-Robin-Broadcast-Schema (RRBC-Schema) verwendet, das eine symmetrische Netzlast für den Hin- und Rückweg eines Broadcast-Rahmens erzeugt (siehe Abschnitt 4.2.1).

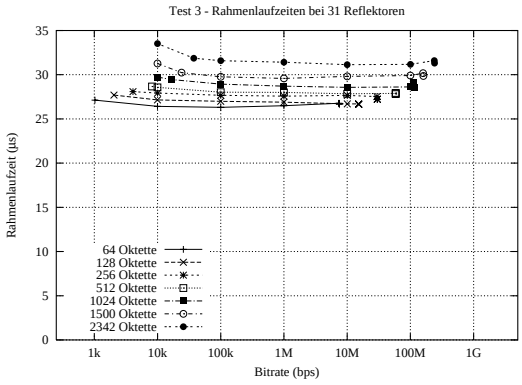
Ergebnisse



(a) 2 Reflektoren



(b) 7 Reflektoren



(c) 31 Reflektoren

Abbildung 5.7: Test 3 - Rahmenlaufzeiten

Exemplarisch für alle durchgeführten Messungen sind in Abb. 5.7(a), Abb. 5.7(b) und Abb. 5.7(c)

die Messwertverläufe für zwei, sieben bzw. 31 Reflektoren dargestellt.

Der Verlauf der Kurven bei zwei Reflektoren in Abb. 5.7(a) ähnelt der Charakteristik der Messergebnisse aus Test 1 (Abb. 5.1), die Laufzeiten sind jedoch länger. Der Mittelwert aller Messungen bei zwei Reflektoren beträgt 2,63 μs. Das Minimum von 2,38 μs liegt bei 64 Oktetten bei 10 Mbps, das Maximum von 3,73 μs bei 2342 Oktetten und 10 kbps.

Die Streuung einzelner Rahmenlaufzeiten innerhalb einer Messung ist gering. Die einzelnen Kurven für jede Rahmenlänge zeigen ähnliche Ergebnisse wie Test 1, eine Abschwächung des bereits in Abschnitt 5.2.1 erklärten Effekts bei geringen Bitraten ist analog zu Test 2 auf die höhere Rate der insgesamt zu bearbeitenden Rahmen zurückzuführen. Weiterhin führt die erhöhte Rechenlast dazu, dass auch die von der Messlast geforderten Bitraten nicht durchgängig erreicht werden. Es treten zwar keine Rahmenverluste auf, 500 Mbps werden jedoch erst ab einer Rahmenlänge von 512 Oktetten erreicht. 100 Mbps werden bei 64 Oktetten nur zu 64 % erreicht. Ursache ist die auf ca. 125.000 sinkende Rate der gesendeten Rahmen pro Sekunde.

Vergleicht man die Resultate mit den in Abb. 5.7(b) dargestellten Messwerten mit 7 Reflektoren, so ist neben den weiter anwachsenden Laufzeiten zu erkennen, dass die gemessenen Werte für eine Rahmenlänge nahezu konstant sind. Es besteht also nur eine geringe Beziehung zwischen der Bitrate und der Rahmenlänge. Auffallend ist jedoch, dass mit wachsender Rahmenlänge die Laufzeiten ebenfalls ansteigen, die Rahmenlänge also mit steigender Reflektorzahl die Laufzeit beeinflusst. Die Rahmenrate ist mit maximal 51.000 Rahmen pro Sekunde weiter gesunken, ebenso sinken die maximal erreichten Bitraten. Erhöht man die Anzahl weiter auf 31 Reflektoren, setzten sich diese Effekte weiter durch, so dass die gemessenen Werte für eine Rahmenlänge nahezu konstant sind, jedoch bei längeren Rahmen höhere Werte gemessen werden.

Aufgrund der Ähnlichkeit der Ergebnisse aller 31 Tests mit Broadcast-Rahmen über das virtuelle Netzwerk werden sie nicht im Einzelnen aufgeführt und es wird im Folgenden ein Vergleich zwischen allen Tests gezogen.

Vergleicht man alle Ergebnisse, so ist zunächst eine wachsende Laufzeit auffällig. Die Rahmen werden in der vnmux-Instanz empfangen und dort ein Klon für jede angebundene netshaper-Instanz erstellt. Da die im NET verwendeten Knoten nur eine CPU besitzen, findet der Vorgang sequentiell statt und führt durch den steigenden Bearbeitungsaufwand zu wachsenden Laufzeiten.

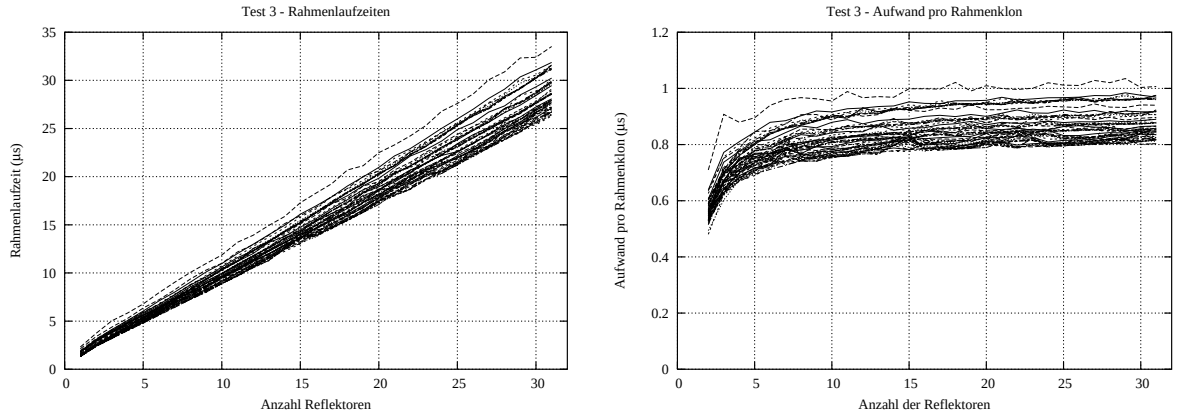
Der Einfluss der Rahmenlänge bei steigender Reflektorzahl ist ebenfalls durch den steigenden Aufwand in der Verarbeitung zu erklären. Die Ursache liegt jedoch nicht in der Verarbeitung der vnmux-Instanz, sondern ist auf den Reflektor zurückzuführen. Die Kopien der Rahmen werden als Klone angelegt. Da jedoch Zeitstempel in die Rahmen eingefügt werden, muss im Reflektor auch eine Kopie der eigentlichen Nutzlast erstellt werden. Erst dann können die Messdaten eindeutig einem Rahmen zugewiesen werden. Für jeden Rahmen muss dafür neuer Speicher reserviert werden, was mit wachsender Rahmenlänge einen größeren Aufwand darstellt.

Dadurch sinken mit steigender Anzahl der Reflektoren ebenfalls die erreichbaren Bit- bzw. Rahmenraten.

Abb. 5.8(a) stellt die Ergebnisse aller Tests dar. Jede Kurve gibt die Laufzeit bei einer Bitrate und einer Rahmenlänge für alle Tests an. Die Werte zeigen einen nahezu linear steigenden Verlauf. Das „Auseinanderdriften“ der Kurven zeigt die Abhängigkeit der Laufzeit von der Rahmenlänge bei steigender Knotenzahl.

$$< \textit{Aufwand pro Klon} > = \frac{< \textit{Broadcast Latenz} > - < \textit{Unicast Latenz} >}{< \textit{Anzahl der Reflektoren} >} \tag{5.1}$$

Die Homogenität der gesamten Daten ermöglicht es, den Aufwand für die Auslieferung eines Rahmens bei n Reflektoren zu berechnen. Werden die Mittelwerte aus Test 1 dazu verwendet eine minimale Laufzeit festzulegen, so stellt die Differenz zum gemessenen Wert näherungsweise den Mehraufwand dar,



(a) Rahmenlaufzeiten (b) Aufwand pro Klon

Abbildung 5.8: Test 3 - Rahmenlaufzeiten und Aufwand pro Klone

der benötigt wird, um alle Klone auszuliefern. Teilt man das Ergebnis durch die Anzahl der Reflektoren, so ergibt sich gemäß Gleichung 5.1 der Aufwand pro Klon. In Abb. 5.8(b) sind alle berechneten Werte dargestellt, die im Mittel bei ca. 0,8223 μs liegen.

Schlussfolgerungen

Die ermittelten Laufzeiten von Broadcast-Rahmen über das virtuelle Netzwerk zeigen ein insgesamt sehr homogenes Bild. Im Vergleich zur Laufzeit mit Unicast-Rahmen aus Test 1 ist ein Anstieg zu erkennen, der abhängig von der Zahl der virtuellen Netzwerkschnittstellen zu deutlich höheren Werten führen kann. Aufgrund der Verarbeitung zeigen die Laufzeiten ein lineares Verhalten. Da der Aufwand für die Erstellung von Rahmenkopien nahezu unabhängig von der Bitrate und Rahmenlänge ist, kann er berechnet werden.

5.3 Kommunikation über Switch

Nach den Messungen der Laufzeiten über das virtuelle Netzwerk, werden in den folgenden Tests Rahmenlaufzeiten über das reale Netz zwischen Knoten gemessen.

Zusätzlich zum bisherigen Aufwand verursachen also die Kommunikation des Kernels mit der Netzwerkkarte, die physische Übertragung an den Switch, im Switch selbst und vom Switch zum Zielknoten weitere Verzögerungen. Dabei sind auch Signalausbreitungs- und Serialisierungsverzögerung und die Verzögerung im Switch einzubeziehen.

Die verwendeten Szenarien orientieren sich an den bereits aus den ersten Tests bekannten. Da der Switch eine zweistufige Architektur aufweist, verdoppelt sich die Anzahl jedoch, da alle Kommunikationspfade in Betracht gezogen werden sollen. Jeder Test wird einmal mit Knoten einer Portgruppe durchgeführt, ein zweiter, analoger Test dient zur Evaluierung der Laufzeiten des Kommunikationspfades über den Chassis Crosspoint Switch.

Um die Szenarien abzubilden, wird für jeden Test ein VLAN angelegt, das jeweils nur die am Test beteiligten Knoten umfasst. Dies ist vor allem bei Broadcasts von Bedeutung, da der Switch für jeden beteiligten Knoten eine Kopie des Rahmens zustellt.

Die Knoten sind analog zu den vorigen Tests konfiguriert. Alle Rahmen werden über eine netshaper-Instanz versendet, die an eine vnmux-Instanz gebunden ist. Erst im vnmux wird entschieden, dass der Rahmen an einen anderen Knoten gesendet werden muss und an den Treiber der Netzwerkkarte ausgeliefert.

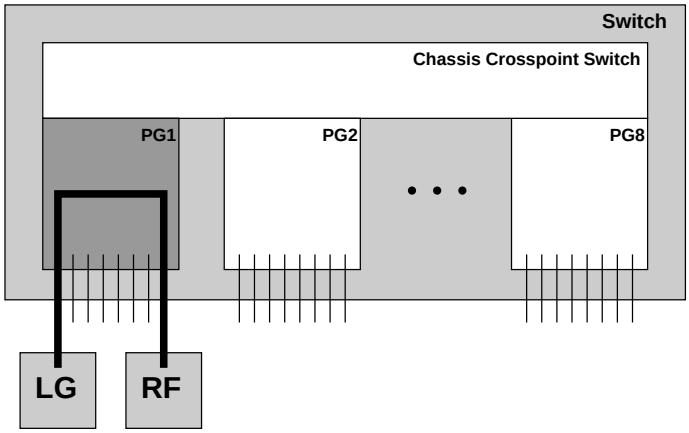
Die Messlasten werden nahezu unverändert übernommen. Überlange Rahmen mit 2342 Oktetten können jedoch nicht versendet werden und werden als Messlast nicht verwendet.

5.3.1 Test 4: Unicast innerhalb einer Portgruppe

Ziel

Der erste Test über den Switch dient dazu, die minimalen Rahmenlaufzeiten über das physische Netzwerk zu messen.

Aufbau/Ablauf



In diesem Test wird ein möglichst einfacher Aufbau verwendet. Beteiligt sind nur zwei Knoten, ein Lastgenerator und ein Reflektor. Die Kommunikation findet über die Shared Memory Switch Fabric statt,

der Knoten innerhalb einer Portgruppe miteinander verbindet. Die Verbindung stellt also den kürzesten und schnellsten Weg zwischen den Knoten dar.

Ergebnisse

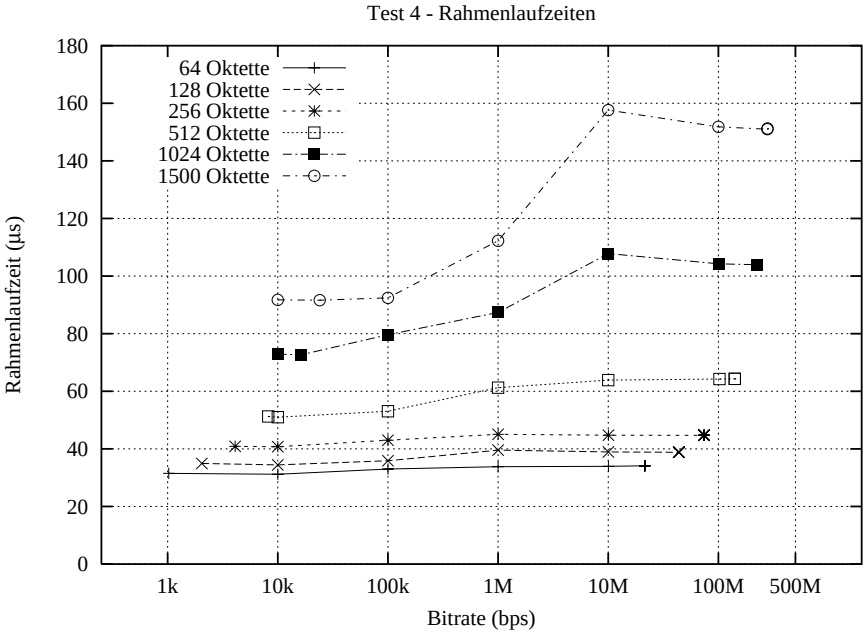


Abbildung 5.9: Test 4 - Rahmenlaufzeiten

Die Messwerte zeigen, wie aufgrund des gestiegenen Aufwandes gegenüber Test 1 zu vermuten, deutlich längere Laufzeiten. Der Mittelwert über alle Messungen von diesem Test beträgt 65,01 μs mit einer Schwankung von 31,17 μs bei 64 Oktetten mit 10 kbps bis zu 157,613 μs bei 1500 Oktetten mit 10 Mbps.

In Abb. 5.9 werden die gemessenen Laufzeiten dargestellt. Bei kurzen Rahmen ist die Laufzeit annähernd konstant und beträgt bei 64 Oktetten im Mittel 33,21 μs. Rahmen mit 128 Oktetten benötigen 37,55 μs, bei 256 Oktetten wird ein mittlerer Wert von 43,60 μs gemessen. Bei längeren Rahmen werden deutlich höhere Werte gemessen. Für 512 Oktette beträgt der Mittelwert 59,18 μs, bei 1500 Oktetten steigt er auf 124,96 μs. Die Verteilung der Werte unterscheidet sich demnach deutlich zu Test 1 im virtuellen Fall.

Die Verteilung der Messwerte einzelner Rahmen weist ebenfalls eine andere Charakteristik auf. In Abb. 5.10 sind diese Werte graphisch für Rahmen mit 512 Oktetten dargestellt. Die Perzentilen verteilen sich gleichmäßig um den berechneten Mittelwert. Der Median liegt immer nahe am Mittelwert, was auf eine gleichmäßige Verteilung aller Messwerte deutet. Die Streuung wird mit zunehmender Bitrate jedoch größer. Berechnet man die Differenz zwischen der 5%-Perzentile und der 95%-Perzentile, in denen 90 % der Messwerte liegen, liegt eine Streuung von ca. 50 % vor. Die Messwerte weichen also 25 % nach oben und 25 % nach unten ab. Die Messungen anderer Rahmenlängen zeigen eine vergleichbare Verteilung.

Die in der Messlast festgelegten Bitraten werden nicht durchgehend erreicht. Die erreichten Bitraten sind in Tab. 5.3 aufgeführt. 100 Mbps werden erst bei einer Rahmenlänge von 512 Oktetten erreicht. Eine Bitrate von 500 Mbps wird überhaupt nicht erreicht, bei einer Rahmenlänge von 64 Oktetten wird sie nur zu 4,2 % erreicht, bei 1500 Oktetten steigt der Anteil auf 55,7 %. Die nicht erreichbaren Bitraten

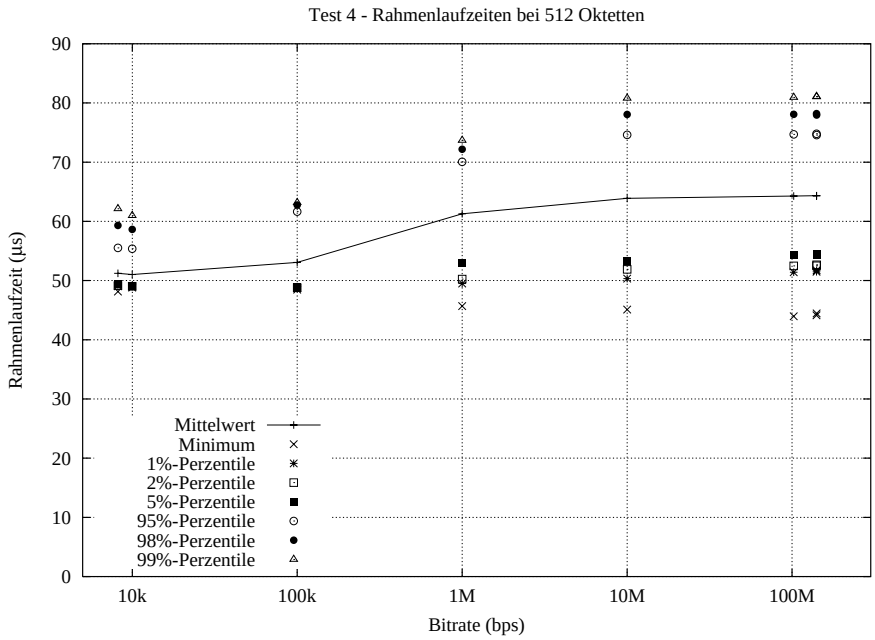


Abbildung 5.10: Test 4 - Verteilung einzelner Rahmenlaufzeiten bei 512 Oktetten

lassen auf eine Auslastung des Knotens schließen. Rahmenverluste durch Überlastung treten jedoch nicht auf.

	Rahmenlänge (Oktette, nur Nutzlast)					
	64	128	256	512	1024	1500
Minimum (bps)	1022	2044	4089	8187	9670	9980
Maximum (Mbps)	21,6	44,0	74,2	140,9	222,3	279,4

Tabelle 5.3: Test 4 - Gemessene Bitraten

In diesem Test werden Rahmen erstmals über eine physische Verbindung übertragen. Der entstehende Aufwand für die Serialisierungs- und die Signalausbreitungsverzögerung führt zu einer berechenbaren, konstanten Laufzeiterhöhung. Die bisher angegebenen Rahmenlängen bezogen sich, um vergleichbare Messwerte zu erhalten, auf die übertragene Nutzlast. Um die reale Anzahl der übertragenen Oktette zu erhalten sind insgesamt 30 Oktette zu addieren. Die benötigte Zeit zur Serialisierung berechnet sich dann nach Gleichung 5.2. Dabei ergeben sich die in Tab. 5.4 berechneten Werte.

$$< Serialisierungsverzögerung/s > = \frac{(< Rahmenlänge/Bytes > + 30) \cdot 8}{1000} \tag{5.2}$$

Eine qualitative Änderung ist aufgrund des konstanten Wertes bei einer Rahmenlänge nicht erkennbar. Die Laufzeiten bei verschiedenen Längen rücken durch herausrechnen der Serialisierungsverzögerung näher zusammen, der Verlauf zeigt aber keine Änderung.

Die Verzögerung durch die Signalausbreitung ist noch geringer. Nimmt man 10 m als Kabellänge an, so benötigt das Signal 57,123 ns um die Strecke zu durchlaufen. Da der Rahmen zweimal übertragen wird, ergibt sich ein Gesamtwert von 114,29 ns. Die entstehenden Verzögerungen sind gegenüber

	Rahmenlänge (Oktette, nur Nutzlast)					
	64	128	256	512	1024	1500
Serialisierungsverzögerung (μs)	0,752	1,264	2,288	4,336	8,432	12,24

Tabelle 5.4: Test 4 - Serialisierungsverzögerung

dem Einfluss der durch die Bearbeitung im Knoten entstehenden Effekte jedoch von untergeordneter Bedeutung.

Der Einfluss des Knotens am Laufzeitverhalten tritt an den Kurven in Abb. 5.9 deutlich zu Tage. Neben den insgesamt höheren Laufzeiten ist ein überproportionaler Anstieg auffällig, der für eine Rahmenlänge ab bestimmten Bitraten auftritt. Nach einer anfangs beinahe konstanten Laufzeit steigen die Werte sprunghaft an, um im weiteren Verlauf dann wieder leicht zu sinken. Die Stelle des Anstiegs verschiebt sich dabei mit wachsender Länge des Rahmens nach rechts zu größeren Bitraten.

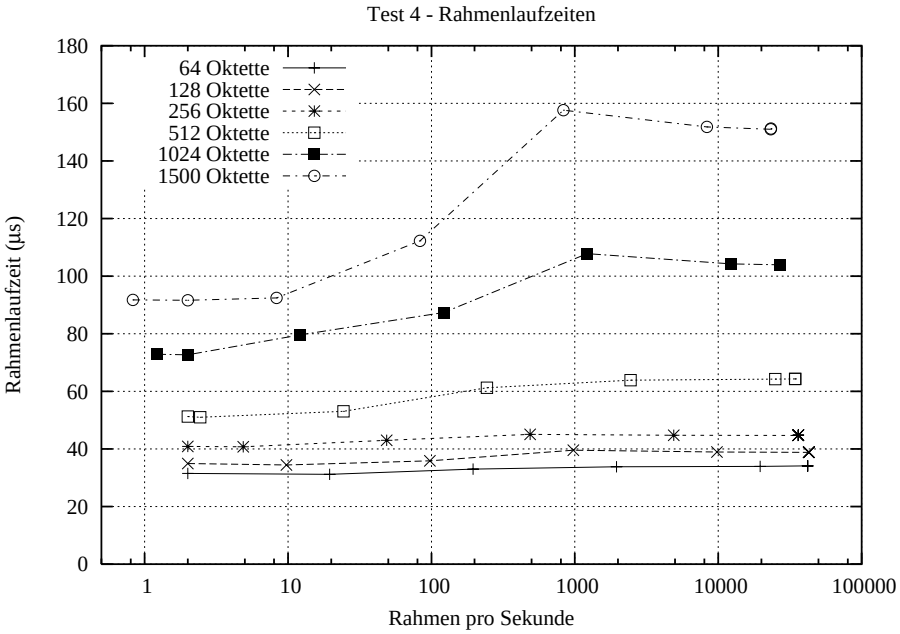


Abbildung 5.11: Test 4 - Rahmenlaufzeiten (über Rahmenrate)

Wählt man eine andere Darstellung der Messergebnisse, so ist der Effekt zu erklären. Abb. 5.11 stellt die Laufzeit über die Anzahl der Rahmen pro Sekunde dar. Der überproportionale Anstieg ist bei dieser Darstellung an nahezu konstanter Stelle ca. zwischen 500–1000 Rahmen/s zu erkennen, der durch eine veränderte Verarbeitungsweise der Rahmen im Knoten entstehen könnte.

Schlussfolgerungen

Der erste Test über das physische Netz zeigt eine gänzlich andere Charakteristik der Messergebnisse. Der Bearbeitungsaufwand für die Rahmen steigt im Knoten an, da zusätzlich eine Kommunikation mit der Netzwerkkarte erfolgt. Bei der physischen Übertragung treten weitere Laufzeitverlängerungen durch die Serialisierungsverzögerung und die begrenzte Signalausbreitungsgeschwindigkeit auf. Zusätzlich muss

der Rahmen im Switch verarbeitet und weitergeleitet werden.

In der Folge unterscheiden sich die gemessenen Werte deutlich. Die Charakteristik des Messwertverlaufes einer Rahmenlänge wird weitgehend durch die Bearbeitungszeit im Knoten bestimmt. Grund dafür ist, dass beim Empfang eines Rahmens vom Kernel jedes Mal neuer Speicherplatz bereitgestellt werden muss, um die weitere Verarbeitung im Knoten zu ermöglichen.

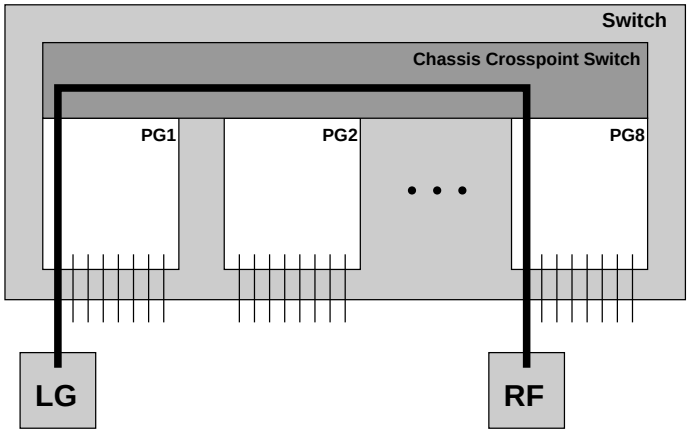
Im Vergleich zu Test 1, dem entsprechenden Szenario im virtuellen Fall, sind klare Unterschiede zu erkennen. Der Weg des Rahmens über den Switch, sowie die zusätzliche Verarbeitung in den Netzwerkkarten, führen zu deutlich höheren Werten.

5.3.2 Test 5: Unicast über Chassis Crosspoint Switch

Ziel

Die Übertragung zwischen verschiedenen Portgruppen erfordert im Switch einen erhöhten Bearbeitungsaufwand. Der Einfluss des Kommunikationspfades über den Crosspoint Switch soll in diesem Test ermittelt werden.

Aufbau/Ablauf



Der Aufbau unterscheidet sich vom vorigen Test durch den Kommunikationspfad über den Switch. Die beiden Messknoten befinden sich in verschiedenen Portgruppen, so dass ein Rahmen zusätzlich über das Verbindungsnetzwerk zwischen den Portgruppen im Switch übertragen werden muss.

Ergebnisse

Vergleicht man Abb. 5.12 mit den Ergebnissen des vorigen Tests (Abb. 5.9), fällt die offenkundige Ähnlichkeit auf. Der Verlauf der Kurven ist identisch, und die gemessenen Werte unterscheiden sich nur unwesentlich. Der Mittelwert aller Messwerte ist mit 67,02 μs nur gering höher, das Maximum wird 161,92 μs bei einer Rahmenlänge von 1500 Oktetten gemessen. Die kürzeste Laufzeit liegt mit 64 Oktetten und einer Bitrate von 10 kbps bei einem Wert von 32,81 μs.

Auch die weitere Analyse bestätigt die nur geringfügig abweichenden Ergebnisse. Die Verteilung der Laufzeiten einzelner Rahmen einer Messung stimmt ebenso überein, wie die minimal und maximal gemessenen Rahmen- und Bitraten. Rahmenverluste treten ebenfalls nicht auf.

Gerade deshalb lassen sich die Messergebnisse mit denen des vorigen Tests gut vergleichen.

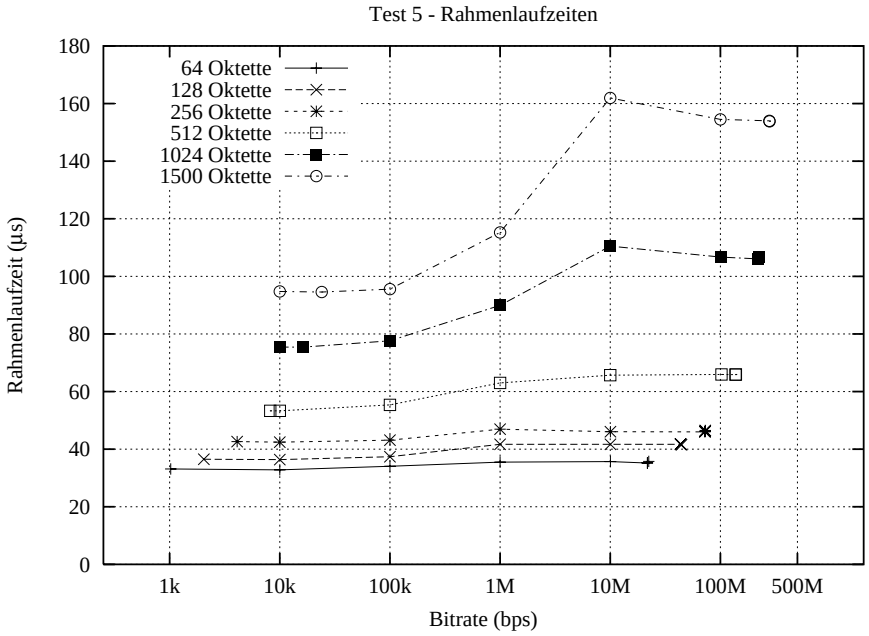


Abbildung 5.12: Test 5 - Rahmenlaufzeiten

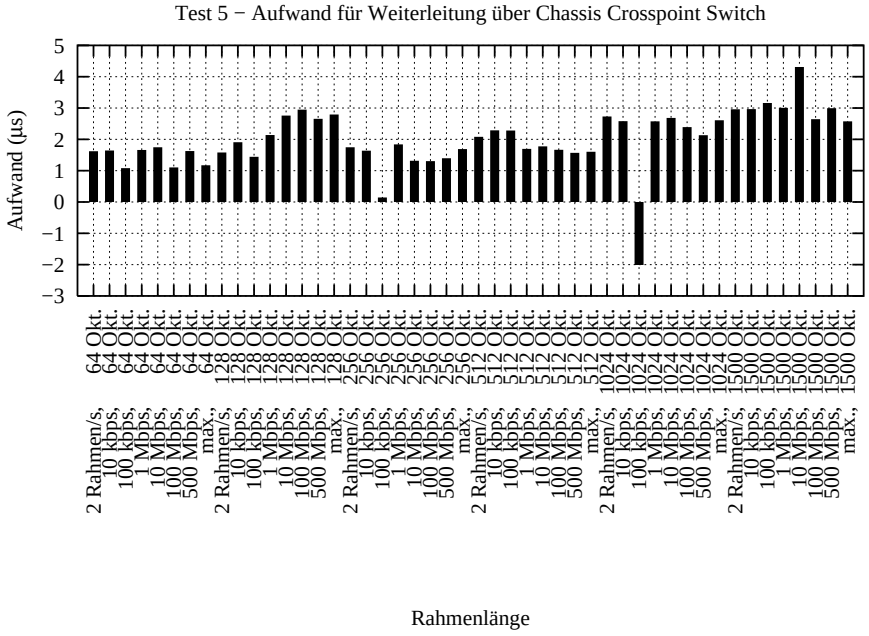


Abbildung 5.13: Test 4 und 5 - Laufzeitdifferenzen

In Abb. 5.13 sind die Differenzen der Laufzeiten zwischen Test 4 und diesem Test berechnet. Der Mehraufwand des Kommunikationspfades, der durch die zusätzliche bitserielle Übertragung über das Verbindungsnetzwerk im Switch entsteht, lässt sich gut erkennen. Durch Messungenauigkeiten zeigen die sehr kleinen Werte zwar nicht ausnahmslos ein eindeutiges Bild, jedoch lassen sich allgemeine Schlüsse ziehen. Die Werte sind insgesamt nahezu konstant, und der Mittelwert aller Differenzen be-

trägt 2,005 μs. Leicht höhere Werte für längere Rahmen lassen sogar den wachsenden Serialisierungsaufwand bei Übertragung des Rahmens über den Chassis Crosspoint Switch erkennen. Ein Einfluss der steigenden Bitrate für eine Rahmenlänge auf die Laufzeit ist nicht zu erkennen. Der Switch kann also die Messlasten verarbeiten, ohne merklich ausgelastet zu werden.

Schlussfolgerungen

Die Ergebnisse sind mit den Werten aus Test 4 fast identisch. Der zusätzliche Aufwand für die Weiterleitung über den Chassis Crosspoint Switch zur gesamten Rahmenlaufzeit beträgt im Mittel 2 μs. Die Auslastung des Switch ist bei diesem Szenario gering.

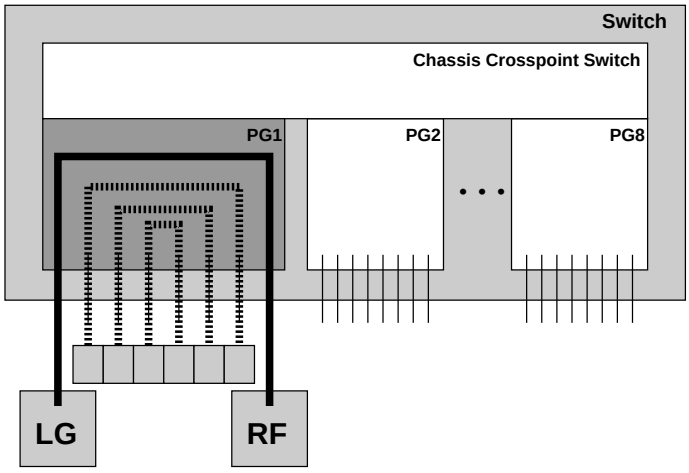
Als weiteres Ergebnis stellt sich die hohe Genauigkeit und Reproduzierbarkeit des Messvorgangs heraus.

5.3.3 Test 6: Unicast innerhalb einer Portgruppe mit Hintergrundnetzlast

Ziel

Ziel dieses Tests ist es, den Einfluss höherer Last im Switch auf die Rahmenlaufzeiten zu messen.

Aufbau/Ablauf



Um eine hohe Netzlast zu erzeugen, senden alle acht Knoten der Portgruppe. Der Messvorgang findet wieder zwischen zwei Knoten statt, die durch den zusätzlichen Netzwerkverkehr nicht beeinträchtigt werden. Die Hintergrundnetzlast wird durch nping erzeugt. Im Gegensatz zum Messvorgang werden die Rahmen jedoch nicht reflektiert, sondern im Empfänger verworfen. Dieser sendet ebenfalls Rahmen, jedoch an einen anderen Knoten. Das entstehende Kommunikationsmuster ist durch dieses Schema weitgehend verschachtelt, während einer Messung wird es aber nicht verändert. Aufgrund der Art der Weiterleitungsfindung im Switch entsteht jedoch kein Nachteil, da die MAC-Adressen der Knoten durch die begrenzte Anzahl komplett im „Lookup-Speicher“ gehalten werden können, so dass Verzögerungen aufgrund unbekannter Zieladressen nicht auftreten. Tab. 5.5 zeigt die verwendete Kommunikationsmatrix.

Quelle		Ziel	
Knoten	Portgruppe-Slot	Knoten	Portgruppe-Slot
c02	1-2	c07	1-7
c03	1-3	c06	1-6
c04	1-4	c05	1-5
c05	1-5	c02	1-2
c06	1-6	c03	1-3
c07	1-7	c04	1-4

Tabelle 5.5: Test 6 - Kommunikationsmatrix für die Hintergrundnetzlast

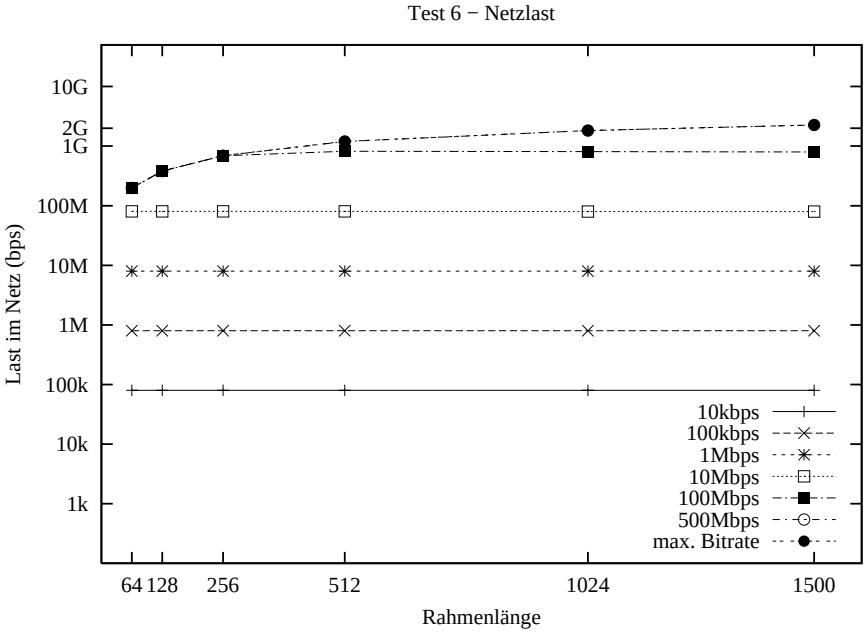


Abbildung 5.14: Test 6 - Netzlast

Ergebnisse

Die Messergebnisse sind analog zu den vorigen Tests ausgewertet worden und in Abb. 5.15 graphisch dargestellt. Die Messung findet aus Sicht der Messknoten unter gleichen Bedingungen wie in Test 4 statt, da sie durch den zusätzlichen Netzwerkverkehr nicht beeinflusst werden. Lediglich im Switch entsteht eine deutliche höhere Belastung durch die Weiterleitung zusätzlicher Rahmen. Die gesamte vom Switch zu verarbeitende Netzlast ist in Abb. 5.14 dargestellt. Die begrenzte Anzahl von Knoten führt bei langen Rahmen zu einer Last von bis zu 2 Gbps, die von der Logik der Portgruppe verarbeitet werden muss. Vergleicht man die Ergebnisse dieses Tests mit den in Test 4 ohne zusätzliche Netzlast ermittelten Werten, ist eine hohe Übereinstimmung zu erkennen. Der Mittelwert aller Messungen berechnet sich zu 67,02 μs, das Minimum von 32,81 μs wird bei 10 kbps und einer Rahmenlänge von 64 Oktetten gemessen. Die maximale Laufzeit von 161,92 μs entsteht bei 1500 Oktetten und 10 Mbps. Die Verteilung einzelner Rahmenlaufzeiten innerhalb einer Messung ist nur unwesentlich abweichend gegenüber Test 4. Weitere Unterschiede in den Auswertungskriterien, wie die Erfüllung der in der Messlast geforderten

Bitraten oder Rahmenverluste können ebenfalls nicht festgestellt werden. Die Ergebnisse sind daher als gleich zu betrachten.

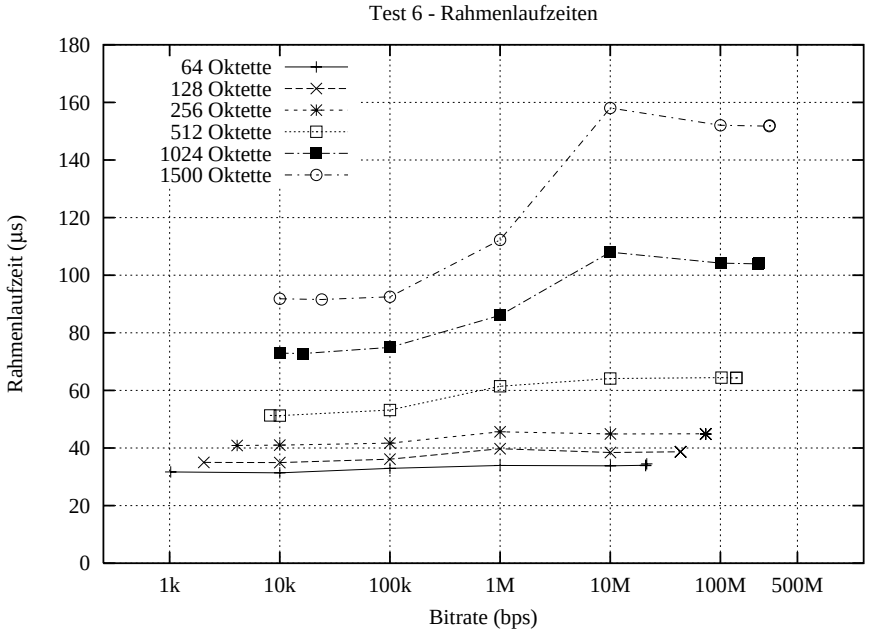


Abbildung 5.15: Test 6 - Rahmenlaufzeiten

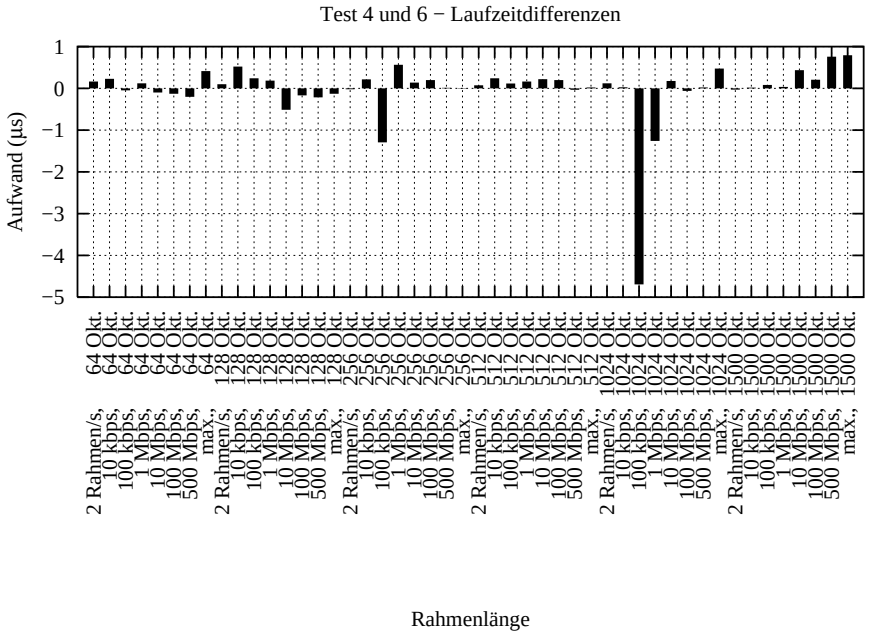


Abbildung 5.16: Test 4 und 6 - Laufzeitdifferenzen

Die Beobachtung bestätigt sich, berechnet man die Differenz der Laufzeiten zwischen Test 4 und diesem Test analog zum vorigen Abschnitt. Der Mittelwert aller Differenzen aus Abb. 5.16 errechnet

sich zu $-0{,}0336\ \mu$s. Der negative Wert bei 100 kbps und 1024 Oktetten ist auf einen Messfehler in Test 4 zurückzuführen, der zu einer Differenz von $-4{,}694\ \mu$s führt. Alle anderen Werte liegen Nahe bei Null, so dass der geringe Mittelwert aller Differenzen nicht nur durch eine günstige Verteilung der Differenzen entsteht.

Schlussfolgerungen

Die Messknoten werden durch die zusätzliche Last im Netz in ihrem Verhalten nicht beeinflusst.

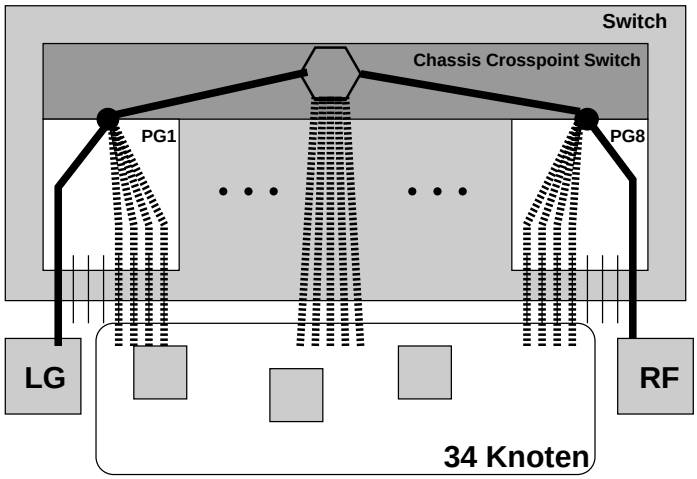
Aufgrund der begrenzten Anzahl von Knoten einer Portgruppe, ist die Last der zusätzlichen Knoten zu gering, um den Switch merklich zu belasten und damit einen Einfluss auf die Messergebnisse herbeizuführen. Minimale Differenzen sind durch Messungenauigkeiten zu erklären. Die Effekte, die durch die Messknoten verursacht werden, überwiegen den Einfluss der zusätzlichen Last bei weitem.

5.3.4 Test 7: Unicast über Chassis Crosspoint Switch mit Hintergrundnetzlast

Ziel

Der Kommunikationspfad zwischen unterschiedlichen Portgruppen führt zu einer Übermittlung der Daten über den Chassis Crosspoint Switch. Während in Test 5 kein weiterer Netzwerkverkehr auf dem Switch vorhanden ist, wird durch das Einführen einer Hintergrundnetzlast ein erhöhter Verarbeitungsaufwand im Switch verursacht. Die entstehenden Auswirkungen auf die Laufzeiten sind Gegenstand dieses Tests.

Aufbau/Ablauf



Im Unterschied zum vorigen Test befinden sich die Messknoten in verschiedenen Portgruppen und die Rahmen durchlaufen den Chassis Crosspoint Switch. Der Chassis Crosspoint Switch verbindet alle acht Portgruppen voll vermascht mit Punkt-zu-Punkt-Verbindungen, die eine Kapazität von 8 Gbps haben. Diese Verbindung kann nicht überlastet werden, da die lediglich acht angeschlossenen Knoten einer Portgruppe zu wenig Last erzeugen. Die Verbindung zwischen dem Chassis Crosspoint Switch und der Portgruppe wird jedoch von nur einer Komponente, der Shared Memory Switch Fabric, hergestellt. Senden viele Knoten an eine Portgruppe, so besteht die Möglichkeit einer Überlastung an dieser Stelle. Um einen Netzwerkverkehr von über 8 Gbps zu erzeugen, sind insgesamt 36 Knoten an dem Test beteiligt.

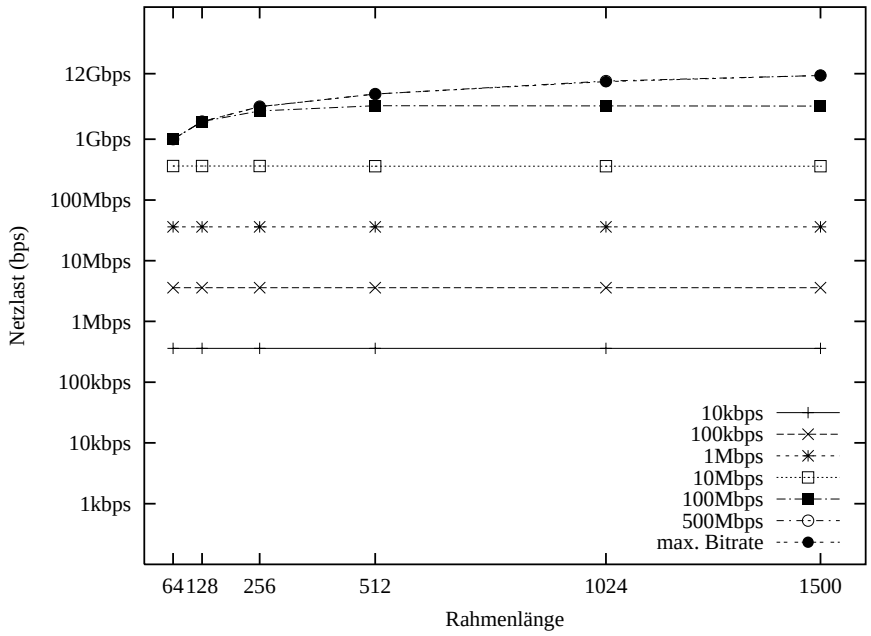


Abbildung 5.17: Test 7 - Netzlast

Die Testlast ist wie folgt organisiert: In Portgruppe 1 befindet sich der Lastgenerator, in Portgruppe 8 der Reflektor. In 4 anderen Gruppen sendet jeweils die Hälfte der Knoten nach Portgruppe 1, die andere Hälfte nach Portgruppe 8. Zielknoten sind dabei jeweils die restlichen, nicht an der Messung beteiligten Knoten. Die Knoten in Portgruppe 1 und 8 senden in eine freie Gruppe, die keine weitere Last erzeugt. Die Anzahl der Knoten ist gegenüber Test 6 deutlich erhöht und die insgesamt erreichbare Netzwerklast steigt damit deutlich an. Das genaue Kommunikationsmuster der Knoten findet sich im Anhang. Die Rahmenlängen und Bitraten der Hintergrundnetzlast entsprechen während einer Messung den vom Lastgenerator verwendeten Werten.

Ergebnisse

Die gesamte Netzlast ist in Abb. 5.17 aus den Messprotokollen aller am Test beteiligten Knoten berechnet worden. Die Messknoten werden von der Hintergrundnetzlast jedoch nicht beeinflusst, eventuelle Laufzeitunterschiede bedeuten also ein verändertes Zeitverhalten der Weiterleitung im Switch. Die Laufzeiten der Rahmen sind, wie in Abb. 5.18 zu erkennen, bis zu mittlerer Last gleich zu den Ergebnissen aus Test 5. Die minimale Laufzeit wird bei 2 Rahmen pro Sekunde und 64 Oktetten Länge 30,72 μs gemessen, die längste Laufzeit wird bei 1500 Oktetten bei 100 MBps Senderate mit 247,48 μs erreicht. Der Mittelwert aller Messungen beträgt 73,74 μs.

Für lange Rahmen und hohe Bitraten steigen die Werte deutlich an, und die Ergebnisse weichen von den gemessenen Werten aus Test 5 ab. Die Differenz der Laufzeiten zu Test 5 ist in Abb. 5.19 berechnet worden. Bei 1024 Oktetten ist ab einer Bitrate von 100 Mbps der Mittelwert um 12,68 μs höher. Die Last im Switch beträgt dabei 3,544 Gbps. Die größte Rahmenlänge mit 1500 Oktetten zeigt ab 10 Mbps deutliche höhere Werte bei einer insgesamt vom Switch zu bearbeitenden Last von 3,60 Gbps.

Unterschiede zeigen sich auch bei den erreichbaren Bitraten. Schon in Test 5 wurden 100 Mbps erst ab einer Länge von 512 Oktetten erreicht. Bei diesem Test werden Raten bis 10 Mbps immer erreicht, Senden mit 100 Mbps Bitrate ist jedoch bei keiner Messung möglich. Die maximalen Raten steigen

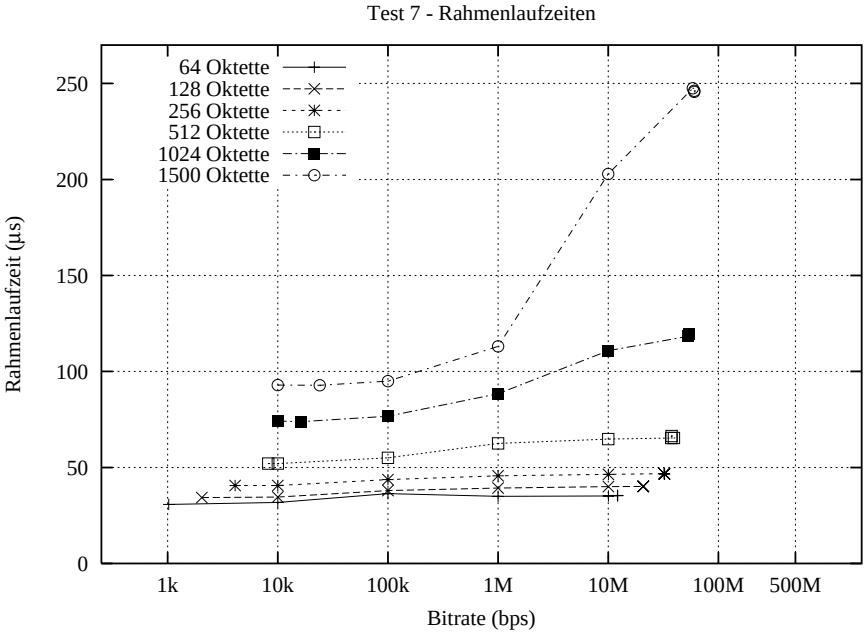


Abbildung 5.18: Test 7 - Rahmenlaufzeiten

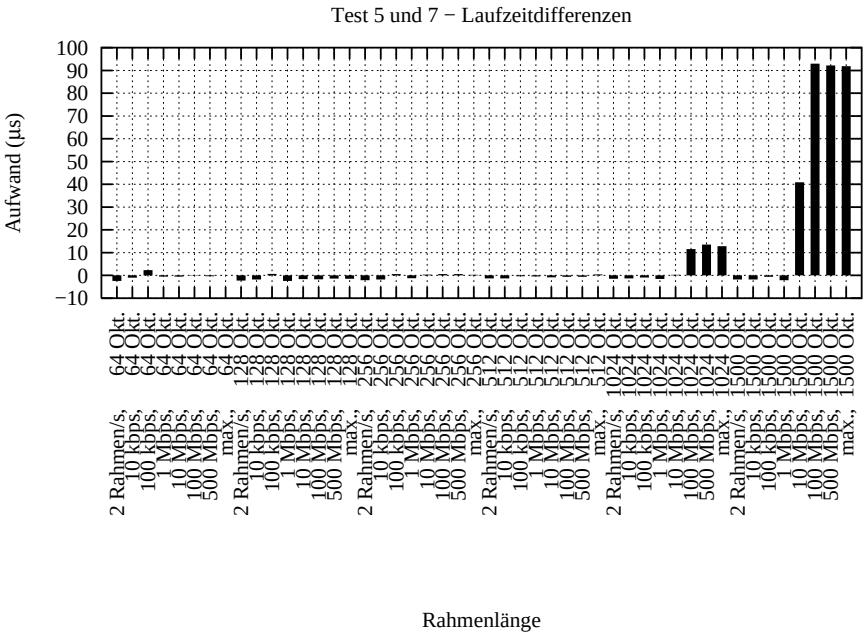


Abbildung 5.19: Test 5 und 7 - Laufzeitdifferenzen

linear von 12 Mbps bei 64 Oktetten auf 60 Mbps bei 1500 Oktetten und liegen damit deutlich unter den in den Messlasten geforderten Werten. Rahmenverluste treten erneut keine auf, und die Verteilung einzelner Laufzeiten einer Messung stimmt meist überein.

Schlussfolgerungen

Während im vorigen Test die erzeugte Last in der Portgruppe nicht zur Auslastung des Switch führt, werden durch die hohe Anzahl sendender Knoten erstmals Bit- bzw. Rahmenraten erreicht, die deutliche Laufzeitänderungen hervorrufen.

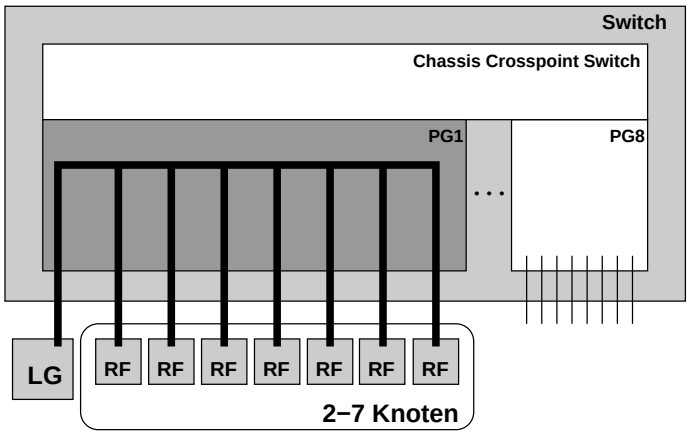
Im Vergleich zu Test 5 werden ähnliche Ergebnisse gemessen. Die Latenzzeiten entsprechend bei niedrigen und mittleren Bitraten denen des Tests in einer Portgruppe. Allerdings ist bei 100 Mbps und 500 Mbps ein überproportionaler Anstieg zu erkennen, der sich auf die hohe Belastung des Switch zurückführen lässt. Da in diesen Test die Anzahl der Knoten deutlich erhöht wird, steigt die Last auf dem Switch bis auf 12 Gbps. Die hohe Belastung bei großen Rahmen führt im Switch wahrscheinlich zu einer batchartigen Verarbeitung, die den Sprung in den Werten verursacht.

5.3.5 Test 8: Broadcast innerhalb einer Portgruppe

Ziel

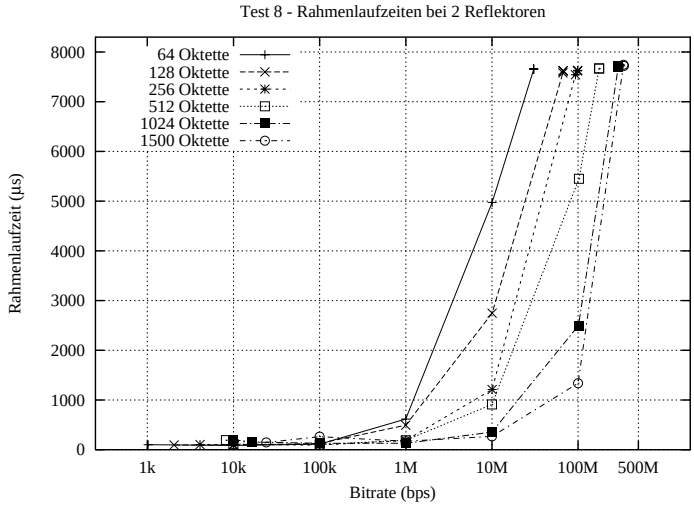
Das Laufzeitverhalten der Unicast-Rahmen wurde in den vorigen Tests analysiert. Bei den folgenden Tests werden Broadcasts versendet. Die Knoten versenden dabei über die Netzwerkkarte nur einen Rahmen, der im Switch für alle Empfänger vervielfacht und ausgeliefert werden muss. Ziel des Tests ist es diesen Aufwand zu ermitteln.

Aufbau/Ablauf

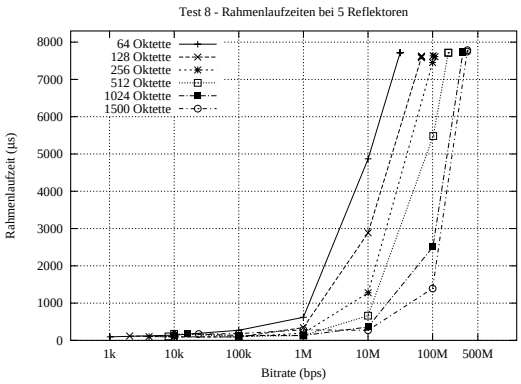


Die Rahmen werden wie dargestellt innerhalb der Portgruppe versendet. Das Kommunikationsmuster entspricht dem des virtuellen Falls in Test 3. Der Lastgenerator sendet ein Broadcast an alle Zielknoten, die in einem VLAN zusammengefasst sind. Jeweils ein Knoten beantwortet die Anfrage und sendet einen Broadcast-Rahmen zurück. Der Test wird ebenfalls mehrmals mit steigender Knotenzahl durchlaufen. Begonnen wird mit zwei Reflektoren, die maximale Anzahl ist durch die acht Knoten einer Portgruppe auf sieben beschränkt.

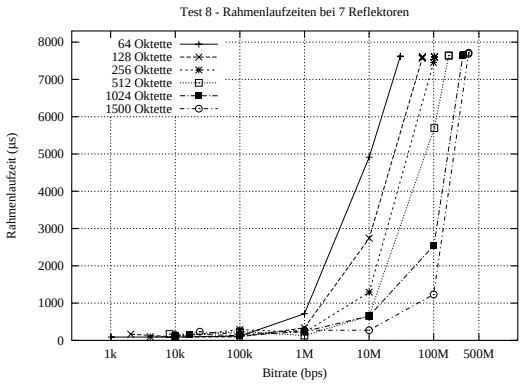
Ergebnisse



(a) 2 Reflektoren



(b) 5 Reflektoren



(c) 7 Reflektoren

Abbildung 5.20: Test 8 - Rahmenlaufzeiten

Die Abbildung 5.20 zeigt die Messergebnisse für zwei, fünf bzw. sieben Reflektoren. Alle zeigen einen gleichartigen Verlauf, der auch bei den anderen Tests nur unwesentlich abweicht. Eine detaillierte Auswertung erfolgt zunächst für den Test mit zwei Reflektoren.

Auffällig ist der starke Anstieg der Rahmenlaufzeit ab 100 Mbps unabhängig von der Rahmenlänge. Im Vergleich zu Test 4 zeigt sich demnach ein gänzlich anderes Laufzeitverhalten. Der Mittelwert steigt auf einen Durchschnittswert von 2889,78 μs an, und liegt deutlich über den im Unicast Fall gemessenen Werten. Der maximale Wert der Laufzeit beträgt 7743,77 μs und wird bei einer Rahmenlänge von 1500 Oktetten bei maximaler Bitrate gemessen. Das Minimum mit 87,47 μs bei 64 Oktetten und 10kbit liegt jedoch deutlich niedriger. Die Anteile der Serialisierungs- und Übertragungsverzögerung können aufgrund der geringen Werte unberücksichtig bleiben und haben den gleichen konstanten Beitrag an der Laufzeit wie bei Messung mit Unicast-Rahmen (vgl. Abschnitt 5.3.1).

Der Verlauf der Messkurve einer Bitrate zeigt starke Schwankungen. Schon bei geringen Bitraten können keine homogenen Werte ermittelt werden, und die Messwerte schwanken im Verhältnis zur noch

insgesamt geringen Laufzeit deutlich. Bei Bitraten ab 100 Mbps steigt der Wert sprunghaft an.

Unstetigkeiten finden sich auch, unabhängig von Bitrate und Rahmenlänge, in der Verteilung der Laufzeiten einzelner Testrahmen. So sind bei geringen Bitraten große prozentuale Unterschiede in den Werten erkennbar, die, keinem Muster folgend, zwischen 30 % und 600 % für die 90 % der Rahmen zwischen der 5%-Perzentile und der 95%-Perzentile liegen. Mit steigender Bitrate und hoher Belastung der Knoten sinkt der Wert auf ca. 22 %.

Erstmals sind auch Rahmenverluste erkennbar, die ab einer Bitrate von 10 Mbps von den Knoten verursacht werden. Nur 83 % der ursprünglich gesendeten Rahmen werden bei 64 Oktetten Länge vom Lastgenerator wieder empfangen. Bei höheren Bitraten sinkt der Wert weiter. Im schlechtesten Fall werden bei 128 Oktetten Rahmenlänge und 100 Mbps nur 10 % der Rahmen wieder vom Lastgenerator empfangen (Abb. 5.21).

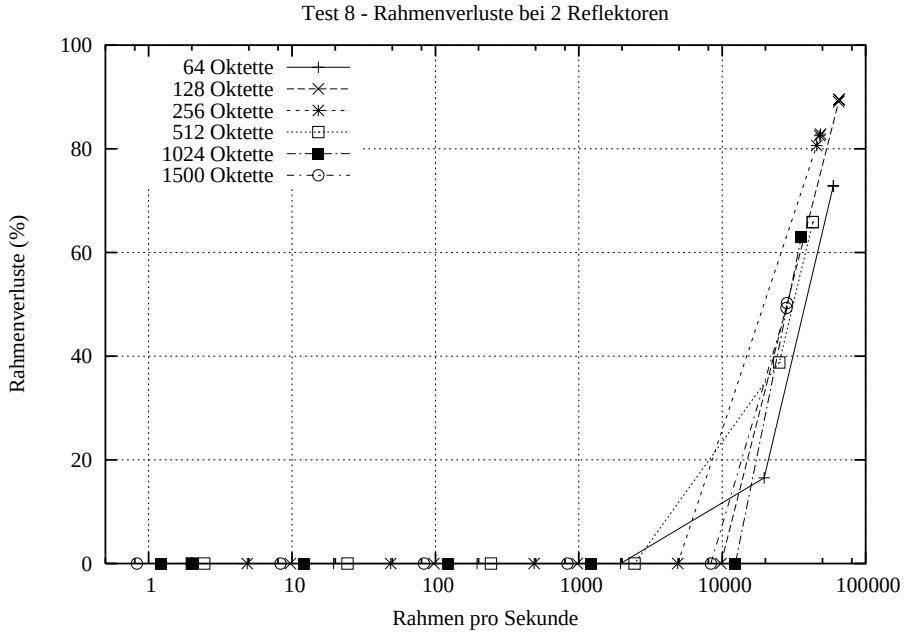


Abbildung 5.21: Test 8 - Rahmenverlust bei 2 Reflektoren

Neben den hohen Rahmenverlusten werden auch die in der Messlast festgelegten Bitraten nicht erreicht. Überraschenderweise werden jedoch höhere Werte gemessen als im Unicast Fall, der ebenfalls in der Tab. 5.6 zu finden ist.

	Rahmenlänge (Oktette, nur Nutzlast)					
	64	128	256	512	1024	1500
Test 4 (Unicast)	21,6	44,0	74,2	140,9	222,3	279,4
Test 8 mit 2 Refl.	30,5	67,0	99,1	175,3	292,0	339,5
Test 8 mit 7 Refl.	30,5	66,8	102,8	170,7	283,9	345,8

Tabelle 5.6: Test 8 - Gemessene Bitraten

Der Vergleich zwischen den Bitraten mit zwei und sieben Reflektoren zeigt ebenfalls die Ähnlichkeit der Messergebnisse mit steigender Anzahl der Reflektoren. Auch alle anderen Aspekte, wie zum Beispiel

die Höhe der Rahmenverluste und die Verteilung der Laufzeiten einzelner Rahmen unterscheiden sich nur in sehr geringem Umfang.

Test	Minimum		Maximum		Mittelwert
2 Reflektoren	87,47	(64,10k)	7743,77	(1500,max)	2889,78
3 Reflektoren	90,50	(256,100k)	7652,42	(1500,max)	2845,56
4 Reflektoren	90,91	(256,100k)	7670,88	(1500,max)	2849,30
5 Reflektoren	92,47	(128,10k)	7784,29	(1500,max)	2894,17
6 Reflektoren	92,46	(128,100k)	7674,70	(1500,max)	2851,21
7 Reflektoren	88,69	(64,10k)	7718,47	(1500,max)	2892,46
Mittelwert	90,41		7707,42		2870,41

Tabelle 5.7: Test 8 - Minimal und maximal gemessene Laufzeiten

Die hohen Laufzeiten zeigen, dass eine Überlastung der Knoten vorliegt. Das ursprüngliche Ziel, den zusätzlichen Aufwand für die Rahmenkopien zu ermitteln, ist nur schwer zu erreichen. Das Verhalten der Knoten beeinflusst die Messwerte so nachhaltig, dass die analoge Berechnung des Mehraufwandes zum vergleichbaren virtuellen Fall nicht erfolgen kann. Die in diesen Test erreichten Belastungen des Switch sind aber, betrachtet man die Ergebnisse der vorigen Tests, so gering, dass vermutlich nur geringe Verzögerungen auftreten.

Die Ursache der Überlastung der Knoten kann aus dem verwendeten Kommunikationsmuster abgeleitet werden. Ein Testrahmen wird als Broadcast an alle Knoten versendet und von genau einem Reflektor – jeweils alternierend – zum Lastgenerator zurückgeschickt. Alle anderen Knoten empfangen jedoch beide Rahmen und verwerfen sie im Reflektor, da die Entscheidung erst dort erfolgen kann. Das Senden im Lastgenerator erfolgt unabhängig vom Empfang. Bei hohen Bitraten befinden sich somit mehrere Rahmen im Umlauf. Während bisher die Rahmenrate durch die maximale Sendeleistung des Lastgenerators beschränkt war, müssen in diesem Szenario auch Rahmen anderer Knoten im Reflektor verarbeitet werden. Das verwendete Kommunikationsmuster führt insgesamt zu einer doppelten Rahmenrate und zur Überlastung der Reflektoren. In der Folge werden nicht mehr alle Rahmen empfangen, was zu hohen Verlusten der Testrahmen führt.

Ein Vergleich zwischen den Messwerten von Unicasts und Broadcasts ist nur eingeschränkt möglich. Lediglich in Bereichen, in denen der Knoten nicht überlastet ist, können aussagefähige Schlüsse getroffen werden. Die Laufzeit aller Rahmen erhöht sich im Mittel um 2824,77 μs gegenüber Test 4. Werden nur Bitraten bis 100 kbps betrachtet, die keine Überlast erzeugen, sinkt der zusätzliche Aufwand auf 78,22 μs.

Schlussfolgerungen

Das Kommunikationsmuster erzeugt eine konstante Last im Netz. Die verdoppelte Rahmenrate für die Reflektoren führt jedoch schon früh zu einer Überlastsituation in den Knoten. Der Einfluss der Knoten auf die Messergebnisse überwiegt den zusätzlichen Aufwand zur Auslieferung der Rahmenkopien im Switch bei weitem, und die entstehenden Laufzeiten führen zu gänzlich unterschiedlichen Messkurven.

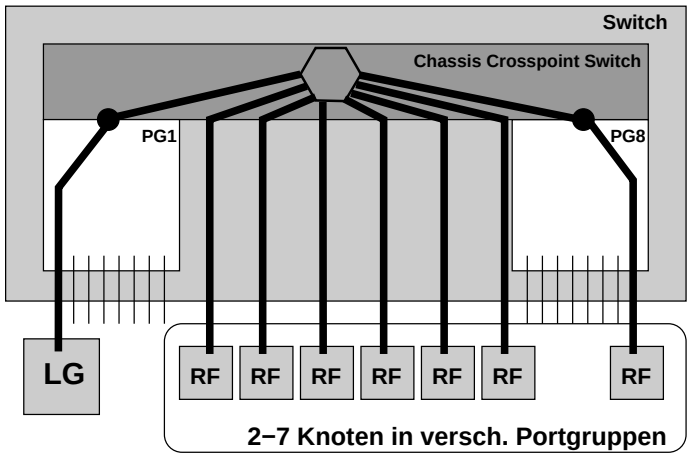
Ein Vergleich mit Test 5, um den Mehraufwand für Rahmenkopien zu ermitteln, ist nicht möglich.

5.3.6 Test 9: Broadcast über Chassis Crosspoint Switch

Ziel

Neben zusätzlichem Aufwand für die Auslieferung von Rahmenkopien in der Portgruppe, müssen die Rahmen in diesem Test zusätzlich über den Chassis Crosspoint Switch übertragen werden.

Aufbau/Ablauf



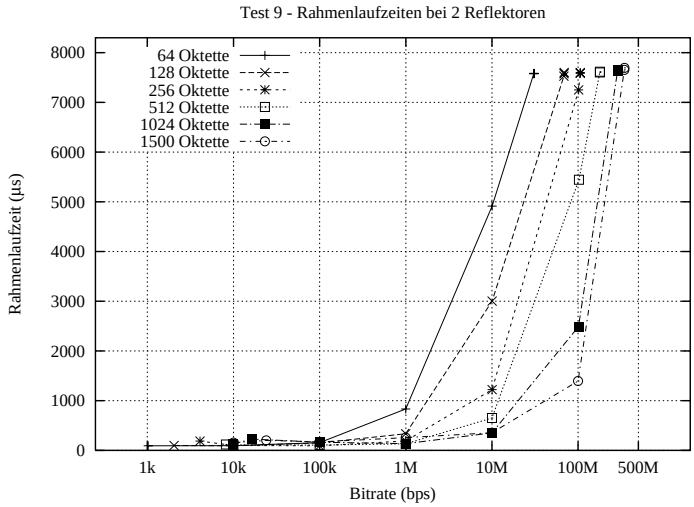
Der Testaufbau entspricht weitgehend dem vorigen, die Kommunikation erfolgt jedoch über den Chassis Crosspoint Switch. Der Switch überträgt eine Kopie in jede Portgruppe einmal. Alle Reflektoren sind daher in unterschiedlichen Portgruppen. Das Kommunikationsmuster wird aus dem vorigen Test übernommen, und der Test wird ebenfalls mehrmals mit steigender Anzahl von Reflektoren durchgeführt. Der erste Test wird mit zwei Reflektoren durchgeführt und bis zu sieben Reflektoren in unterschiedlichen Portgruppen gesteigert.

Ergebnisse

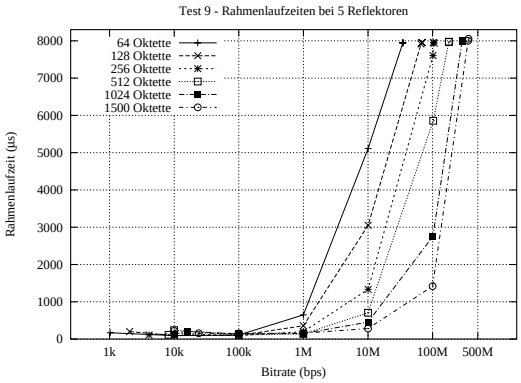
Die Messkurven ähneln den in Test 8 mit Broadcasts innerhalb der Portgruppe ermittelten Ergebnissen. Die Anzahl der Reflektoren hat auch bei diesem Test demnach kaum Einfluss auf die Ergebnisse, wie die Messkurven in Abb. 5.22 zeigen. Nach anfangs, unabhängig von Bitrate und Rahmenlänge verteilten, relativ geringen Laufzeiten, steigen die Werte deutlich an. Die Minima und Maxima der Tests sind in Tab. 5.8 angegeben. Die Werte entsprechen weitgehend den schon in Test 8 festgestellten. Auffällig ist jedoch, dass Tests mit einer geraden Anzahl von Reflektoren durchgehend kleinere Mittelwerte zeigen.

Die Streuung einzelner Laufzeiten einer Messung ist mit den aus Test 8 ermittelten Ergebnissen zu vergleichen. Bei geringen Bitraten werden teilweise Werte bis 564 % erreicht (2 Reflektoren, 64 Oktette, 10 kbps), ein Zusammenhang zur Rahmenlänge lässt sich jedoch nicht erkennen. Bei hohen Bitraten sinkt die prozentuale Abweichung wieder auf Werte um 22 %. Gleiches gilt für die erreichten Bitraten und Rahmenverluste. Die Werte sind unabhängig von der Anzahl der Reflektoren für alle Tests annähernd gleich. Ebenfalls nur unwesentliche Unterschiede sind im Vergleich zu den Ergebnissen von Test 8 festzustellen.

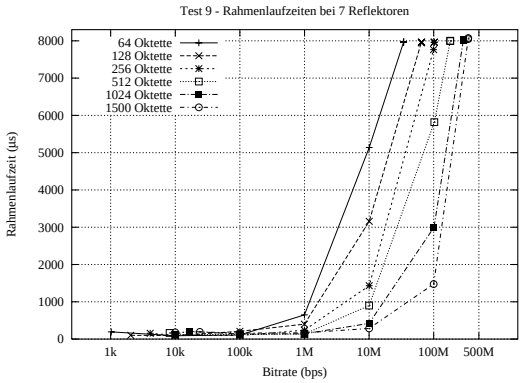
Der Aufwand des mehrfachen Versendens über den Chassis Crosspoint Switch ist in diesem Test jedoch größer. Der Rahmen muss von der Portgruppe des sendenden Knotens einzeln an alle Portgruppen der Zielknoten übertragen werden. Die Belastung im Switch steigt dabei bis auf das Siebenfache



(a) 2 Reflektoren



(b) 5 Reflektoren



(c) 7 Reflektoren

Abbildung 5.22: Test 9 - Rahmenlaufzeiten

bei steigender Reflektorzahl an. Aufgrund der starken Streuung der Werte und der hohen Laufzeit, die durch die Art der Bearbeitung im Knoten entstehen, kann der absolute Zusatzaufwand aber nicht in den Messwerten erfasst werden.

Schlussfolgerungen

Das veränderte Kommunikationsschema führt im Vergleich zu Test 8 nur zu unwesentlichen Veränderungen der Laufzeiten. Der Einfluss der Knoten bestimmt weitgehend die Laufzeiten, so dass der veränderte Aufwand zur Weiterleitung im Switch nicht an den Ergebnissen erkennbar ist.

5.3.7 Test 10 und 11: Unicast & Broadcast mit weiteren Zeitstempeln

Ziel

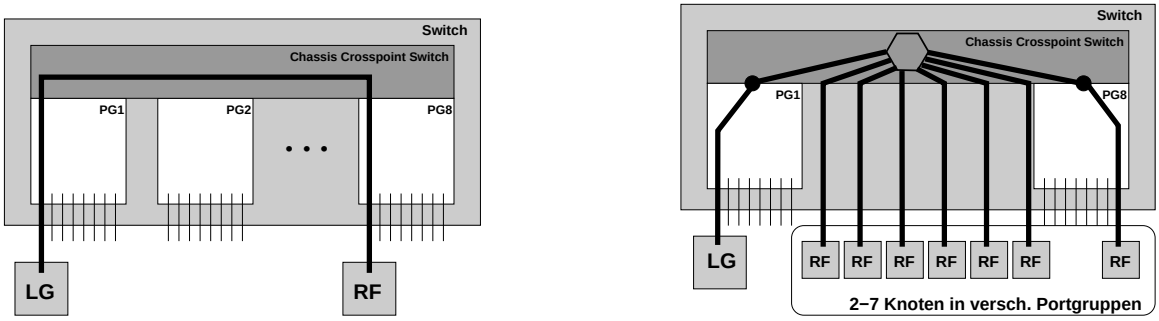
Um die Auswirkungen der Überlast bei der Bearbeitung eines Rahmens im Knoten zu analysieren, werden Test 4 und Test 8 mit 2 Reflektoren nochmals durchgeführt. Im Unterschied zu allen vorigen

Test	Minimum		Maximum		Mittelwert
2 Reflektoren	92,33	(256,10k)	7693,80	(1500,max)	2873,03
3 Reflektoren	93,90	(64,10k)	7855,29	(1500,max)	2935,27
4 Reflektoren	90,57	(64,10k)	7701,51	(1500,max)	2877,96
5 Reflektoren	90,94	(64,10k)	8053,89	(1500,max)	3008,09
6 Reflektoren	94,29	(128,100k)	7743,09	(1500,max)	2892,38
7 Reflektoren	89,93	(64,10k)	8074,06	(1500,max)	3032,22
Mittelwert	91,99		7853,61		2936,49

Tabelle 5.8: Test 9 - Minimal und maximal gemessene Laufzeiten

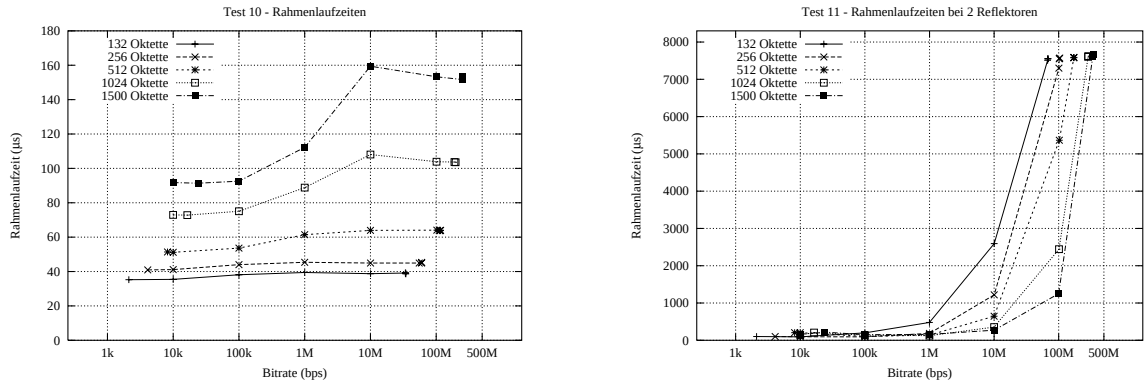
Messvorgängen werden allerdings mehr Zeitstempel genommen. Zusätzlich zu den vier Messzeitpunkten in den netshaper-Instanzen werden auch Werte im Lastgenerator bzw. den Reflektoren sowie im Treiber der Netzwerkkarte genommen. Ziel ist die Aufschlüsselung der gesamten Laufzeit in die einzelnen Phasen der Bearbeitung eines Rahmens. Es können damit Aussagen über die Zeitpunkte der Verzögerung getroffen werden, um den auffälligen Anstieg der Laufzeiten bei Überlast weiter zu klären. Test 10 entspricht dabei Test 4, Test 11 ist analog zu Test 8 mit 2 Reflektoren.

Aufbau/Ablauf



Der Aufbau wird unverändert aus den Tests 4 und 8 übernommen. Es werden insgesamt 12 Zeitstempel gegenüber bisher vier genommen. Da in den Rahmen entsprechend mehr Platz zur Speicherung benötigt wird, müssen die Messlasten angepasst werden. Die 12 Zeitstempel können erst ab einer Länge von 132 Bytes komplett im Rahmen gespeichert werden. Rahmenlängen von 64 Oktetten und 128 Oktetten können also nicht ausgewertet werden, die anderen Rahmenlängen und alle Bitraten werden unverändert übernommen. Zusätzlich wird eine Messung mit 132 Oktetten Rahmenlänge vorgenommen.

Eine Vergleichbarkeit zu den bisherigen Ergebnissen ist nur in begrenztem Umfang möglich, da sich der Aufwand zur Zeitnahme und Speicherung der Daten im Rahmen vergrößert. Messwerte die zu einem späteren Zeitpunkt genommen werden, vergrößern sich mindestens um die Zeit, die benötigt wird, um alle zusätzlich genommenen, früheren Zeitstempel zu erfassen und in die Rahmen einzutragen.



(a) Test10 (b) Test11

Abbildung 5.23: Test 10 und 11 - Rahmenlaufzeiten

Ergebnisse

Die beiden Tests zeigen insgesamt sehr ähnliche Ergebnisse gegenüber den vorigen Messungen mit weniger Zeitstempeln. Betrachtet man die Laufzeiten die aus den Zeitstempeln des netshaper-Moduls berechnet werden können, zeigen die in Abb. 5.23(a) dargestellten Messkurven vergleichbare Ergebnisse wie Test 4, Abb. 5.23(b) zeigt ebenfalls einen insgesamt gleichen Verlauf wie Test 8.

Für Test 10 ergibt sich ein Mittelwert von 71,58 μs. Das Maximum liegt mit 35,28 μs bei 132 Oktetten und 2 Rahmen pro Sekunde, das Minimum mit 159,41 μs bei 1500 Oktetten und 10 Mbps. Die veränderten Messlasten lassen den direkten Vergleich zum Mittelwert aus Test 4 nicht zu. Der vergleichbare Mittelwert aus Test 4 für Rahmenlängen zwischen 128 und 1500 Oktetten liegt jedoch mit 71,37 μs fast gleich. Die anderen Analyseergebnisse zeigen ebenfalls meist keine merklichen Unterschiede.

Der gestiegene Messaufwand ist jedoch bei hohen Bitraten zu erkennen. Die erreichbaren Bitraten bzw. Rahmenraten sinken ab 100 Mbps um bis zu 20 % ab. Der Sendevorgang wird ausgebremst und führt deshalb auch auf dem weiteren Weg zum Reflektor und zurück nicht zu einer Überlastung der Knoten.

Test 11 zeigt im Vergleich zu Test 8 ebenfalls nur geringe Unterschiede. Die minimale Laufzeit wird bei 132 Oktetten und 10kbit mit 91,45 μs gemessen, das Maximum von 7662,53 μs wird bei maximaler Bitrate für Rahmen mit 1500 Oktetten erreicht. Der Mittelwert liegt mit 2703,75 μs nahe am vergleichbaren Mittelwert von 2746,30 μs von Test 8. Der weitere Vergleich zwischen Test 11 und 8 ergibt ein uneinheitliches Bild. Je nach Bitrate und Rahmenlänge differieren die Mittelwerte der Messpunkte um bis zu über 30 %. Test 11 zeigt einen leicht höheren Rahmenverlust, die maximalen Bitraten steigen dagegen leicht an. Insgesamt beeinflussen die zusätzlichen Zeitstempel die Laufzeit dennoch trotz des gestiegenen Messaufwandes nur gering.

Abb. 5.24 verdeutlicht nochmals die Zeitpunkte, an denen Zeitstempel im Verlauf genommen werden (TS1–TS12). Aus den Differenzen zweier aufeinander folgender Zeitstempel kann dann der Bearbeitungsaufwand für jeden Arbeitsschritt berechnet werden (t1–t9). Für den Lastgenerator und Reflektor lassen sich aus den gemessenen Werten also neun Abschnitte ableiten. Die Berechnung der Übertragungszeit ist analog zu den bisherigen Tests, es werden aber die Zeitstempel, die im Treiber der Netzwerkkarte gemessen werden, verwendet.

Die neue Zeitnahme im Treiber der Netzwerkkarte führt zu genaueren Laufzeitmessungen, da der Anteil der beteiligten Kernel-Komponenten sinkt und Effekte durch die Nebenläufigkeit minimiert werden. Die Ergebnisse sind aber nicht mehr direkt mit den virtuellen Tests zu vergleichen.

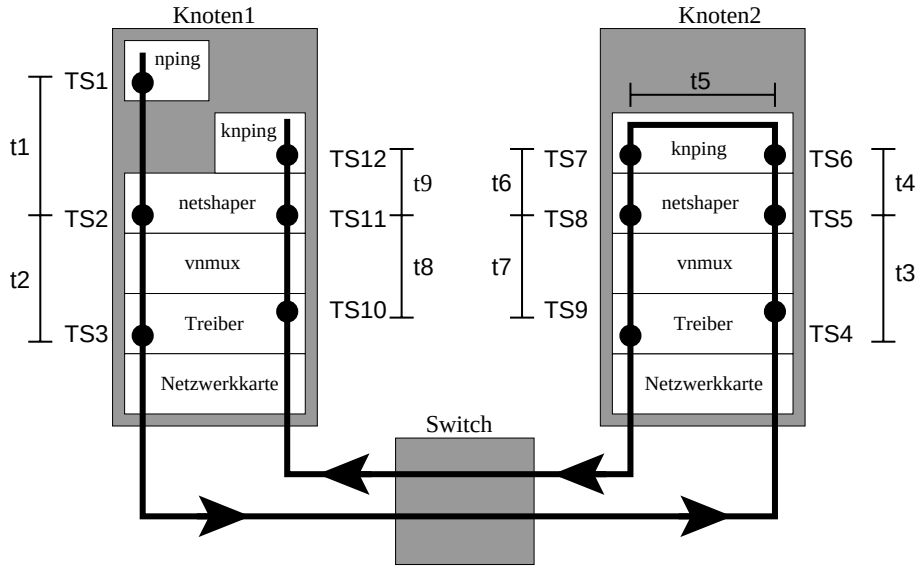
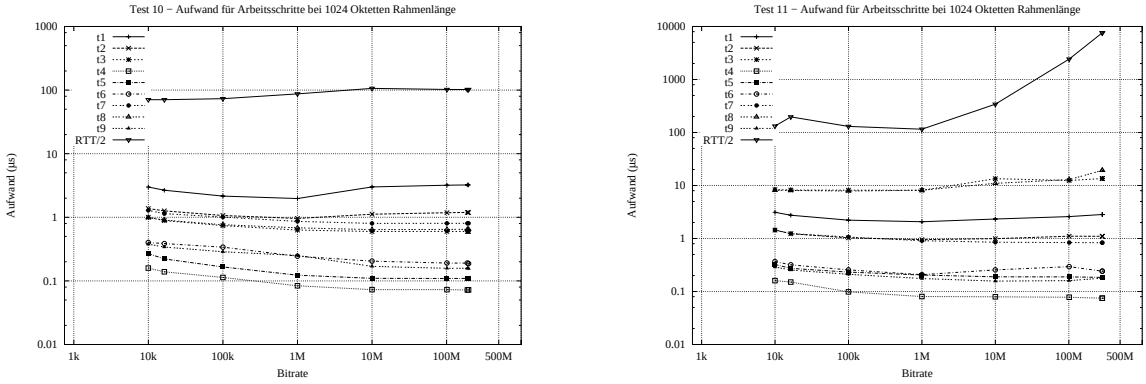


Abbildung 5.24: Test 10 und 11 - Erfasste Zeitstempel und Zeitabschnitte

Der Messzeitpunkt im Treiber der Netzwerkkarte stellt den ersten Zeitpunkt dar, an dem Messwerte genommen und im Rahmen gespeichert werden können. Der Messzeitpunkt ist beim Senden direkt vor der Übergabe an die Netzwerkkarte implementiert. Das Empfangen von Rahmen wird im Treiber durch ein zweistufiges Konzept realisiert. Wird ein Rahmen vom Netzwerk entgegengenommen, so wird er durch einen DMA-Transfer im Speicher abgelegt, und durch einen Interrupt der Empfang dem Betriebssystem gemeldet. Treffen mehrere Rahmen kurz hintereinander ein, so wird ebenfalls nur ein Interrupt ausgelöst. Die Rahmen werden dann durch eine weitere Funktion, der Polling-Funktion, zur weiteren Verarbeitung gesammelt. Der Zeitpunkt der Messdatenerfassung findet beim Empfang in der Polling-Funktion statt. Während der Bearbeitung des Interrupts sind die Rahmen noch nicht verfügbar, so dass keine Zeitstempel eingetragen werden können. Der Zeitpunkt des Pollings kann sich jedoch bei hoher Last um bis zu 10 ms verzögern. Schon die in Test 8 berechneten Durchschnittswerte legen ein solches Verhalten bereits nahe.

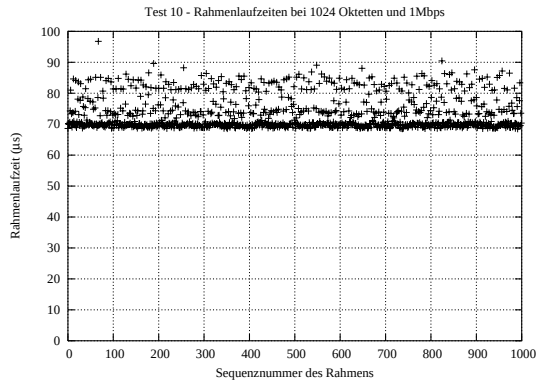


(a) Test10: Phasenlängen

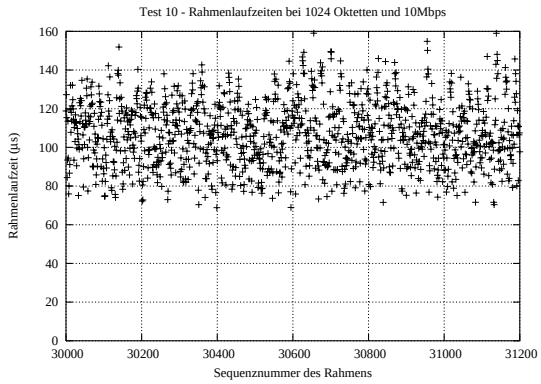
(b) Test11: Phasenlängen

Abbildung 5.25: Test 10 und 11 - Dauer einzelner Verarbeitungsschritte

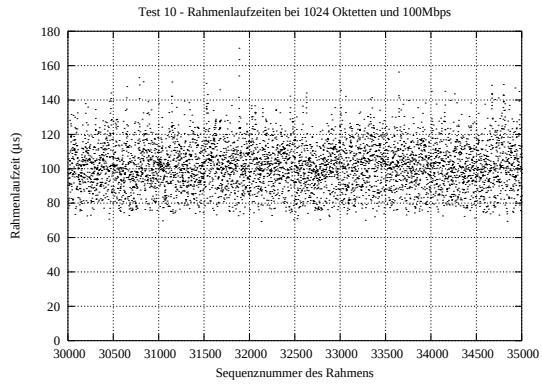
Diese Zeitabschnitte werden im Folgenden dazu verwendet, den Anteil an der Gesamtlaufzeit aufzuschlüsseln. Die sich daraus ergebenden Phasenlängen sind für Test 10 in Abb. 5.25(a) und für Test 11 in Abb. 5.25(b) für eine Rahmenlänge von 1024 Oktetten dargestellt. Die Zeitabschnitte t1 ,t2, t4, t5, t6, t7 und t9 sind immer annähernd konstant und weisen nur sehr geringe Werte auf. Eine Abhängigkeit von der Auslastung zeigen jedoch die Schritte t3 und t8, die den Bearbeitungsaufwand vom Empfang des Rahmens bis zur Auslieferung an das netshaper-Modul im Lastgenerator und Reflektor umfassen. Bei geringer Last (Test10) liegt der Wert bei ca. 0,7234 μs , bei hoher Last (Test11) steigt der Wert auf 11,28 μs im Mittel.



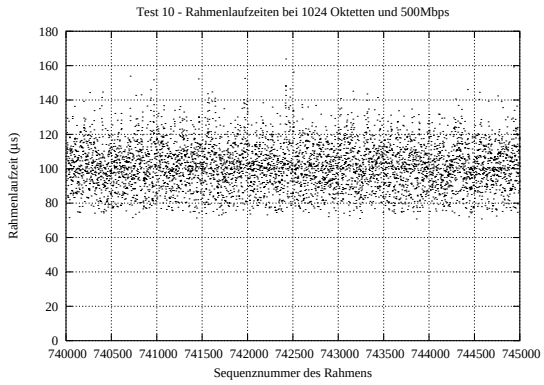
(a) 1 Mbps



(b) 10 Mbps



(c) 100 Mbps

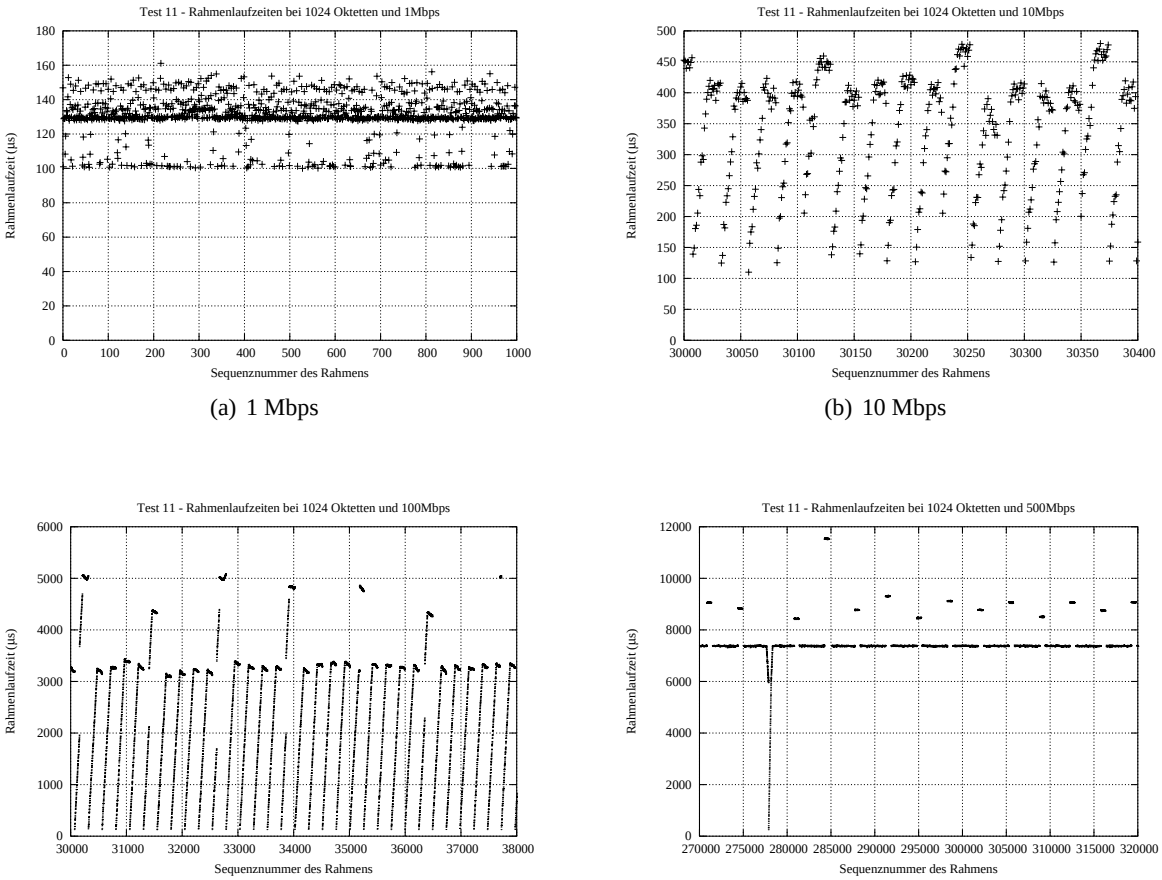


(d) 500 Mbps

Abbildung 5.26: Test 10 - Laufzeiten einzelner Rahmen bei 1024 Oktetten

Der wesentliche Anteil an der Gesamtlaufzeit entsteht zwischen Übergabe an die Netzwerkkarte bis zum Empfang im Linux-Kernel durch die Polling-Funktion (in Abb. 5.25 zwischen TS3 und TS4, bzw. TS9 und TS10). Bei entsprechender Auslastung entstehen an dieser Stelle Verzögerungen, wenn die Daten nicht umgehend abgeholt werden können, sondern im ungünstigen Fall erst 10 ms später beim nächsten Aufruf des SoftIRQ.

Die im Folgenden angegebenen Messwerte beziehen sich auf die im Treiber der Netzwerkkarte genommenen Zeitstempel. Abb. 5.26 stellt die Laufzeiten einzelner Rahmen für eine Länge von 1024 Oktetten bei Test 10 dar. Die Rate der gesendeten Unicasts führt zu keiner Überlastung. Bei allen dargestellten Bitraten ist eine gleichmäßige Verteilung zu erkennen. In Abb. 5.26(a) liegen viele Werte sogar



(a) 1 Mbps

(b) 10 Mbps

(c) 100 Mbps

(d) 500 Mbps

Abbildung 5.27: Test 11 - Laufzeiten einzelner Rahmen bei 1024 Oktetten

nahe der minimalen Laufzeit. Bei höheren Bitraten verteilen sich die Laufzeiten gleichmäßig um den Mittelwert.

Bei einer zu hohen Rahmenrate und der Überlastung des Knotens sind wie in Abb. 5.26 für Test 11 jedoch Unterschiede zu erkennen. Während bei geringen Rahmenraten die Rechenkapazität ausreichend ist, wird mit steigender Rate die Empfangsverzögerung immer größer. Ursache ist, dass einzelne Rahmen nicht bis zum Eintreffen des nächsten vollständig verarbeitet werden können. Dadurch verzögert sich der Ablauf für den folgenden Rahmen, da erst die Bearbeitung des vorigen Rahmens abgeschlossen wird. In der Abbildung wird dies durch die steigenden Laufzeiten der Rahmen ersichtlich. Eine gleich bleibende, hohe Verzögerung entsteht in der Folge, wenn mehrere Rahmen durch einen einzigen Aufruf der Polling-Funktion verarbeitet werden.

Schlussfolgerungen

Die weitere Aufschlüsselung der Laufzeit auf die einzelnen Bearbeitungsphasen zeigt deutlich den starken Einfluss der Verarbeitung im Knoten auf die Laufzeit. Den Hauptanteil trägt dabei die im Treiber implementierte Empfangsmethode bei. Können die Rahmen beim Empfang nicht schnell genug vom Kernel abgeholt werden, entstehen Verzögerungen, die alle anderen bei weitem übertreffen. Alle anderen Zeitabschnitte weisen nur geringe Werte auf, die meist unabhängig von der Rahmenlänge und -rate

sind.

5.4 Zusammenfassung der Ergebnisse

Nach der ausführlichen Analyse der einzelnen Tests, werden in diesem Abschnitt die Ergebnisse zusammengefasst und ein Vergleich zwischen virtuellem und realem Fall gezogen.

Im virtuellen Fall werden die Rahmen ausschließlich auf einem Knoten bearbeitet. Daraus ergeben sich Charakteristika, die bei allen Tests im virtuellen Fall festgestellt werden können. Die Laufzeiten sind insgesamt sehr kurz, die Rahmenlänge hat nur geringen Einfluss auf die Laufzeit. Eine Hintergrundnetzlast und die daraus mögliche Überlast des Knotens führt nicht zu Rahmenverlusten. Als Auswirkung sinkt jedoch die Anzahl der maximal versendbaren Rahmen pro Sekunde. Unicasts zeigen ein fast konstantes Laufzeitverhalten. Ein Mittelwert kann mit 1,485 μs angegeben werden. Das Minimum der Laufzeit eines Rahmens ist 1,27 μs, die längste gemessene Laufzeit eines Rahmens beträgt 66,86 μs.

Broadcasts werden im Knoten mittels Klonen mehrfach ausgeliefert. Die sequentielle Abfolge führt dabei mit wachsender Zahl der angebundenen Netzwerkschnittstellen zu einer linear ansteigenden Laufzeit. Dabei entsteht für jeden Klon ein zusätzlicher Aufwand von ca. 0,8223 μs. Die Auslieferung des ersten Klons erfolgt also nach 1,485 μs, der zweite Klon wird nach 2,307 μs ausgeliefert.

Die Kommunikation über das physische Netzwerk zeigt eine gänzlich andere Charakteristik und es werden deutlich längere Laufzeiten gemessen. Der Unterschied ergibt sich hauptsächlich durch die aufwändigere Verarbeitung der Rahmen im Knoten, die Serialisierungsverzögerungen und Verzögerungen durch die Weiterleitung über den Switch sind dagegen gering. Ein deutlicher Unterschied zeigt sich in den Laufzeiten bei hoher Belastung der Knoten.

Sind die Knoten nicht überlastet und können die Messlasten verarbeiten, ergeben sich im Unicast Fall Laufzeiten zwischen 31,52 μs und 247,49 μs, abhängig von der Rahmenlänge. Der Mittelwert aller Laufzeiten von Test 4 bis 7 beträgt 74,445 μs, die kürzeste Laufzeit eines Rahmens liegt bei 26,26 μs, die Längste bei 10029,49 μs. In Tabelle Tab. 5.9 sind die Mittelwerte von Test 4 bis 7 aufgeführt. Zusätzlich ist die Abweichung angegeben, die sich aus dem Minimum und dem Maximum der Mittelwerte der Tests einer Rahmenlänge zum gesamten Mittelwert ergibt.

Rahmenlänge	Mittelwert	Abweichung nach unten	Abweichung nach oben
64	33,87	9,35 %	7,38 %
128	38,31	10,51 %	9,18 %
256	44,20	8,17 %	6,14 %
512	59,99	14,94 %	10,60 %
1024	93,42	22,21 %	28,09 %
1500	136,31	32,82 %	81,56 %

Tabelle 5.9: Test 4-7 - Rahmenlaufzeiten

In den durchgeführten Broadcast-Tests führt die schnell steigende Last dazu, dass die Knoten nicht alle Rahmen verarbeiten können. Die zusätzliche Kommunikation des Kernels mit der Netzwerkkarte führt bei hoher Last und dem verwendeten Polling zu konstant hohen Werten von ca. 7670 μs.

Die unterschiedlichen Kommunikationswege über den Switch haben nur einen geringen Anteil. Der zusätzliche Aufwand liegt bei 2 μs über den Chassis Crosspoint Switch gegenüber der Kommunikation innerhalb einer Portgruppe.

Die Anteile der Verzögerungen lassen sich in zwei Klassen einteilen. Ein Teil der gesamten Verzögerung ist unabhängig von der Rahmenlänge und umfasst den Software-Anteil, der bei der Bearbeitung im Knoten immer durchlaufen werden muss, jedoch keine Kopiervorgänge der Daten beinhaltet. Ebenso

ist Signalausbreitungsverzögerung unabhängig von der Rahmenlänge. Der zweite Teil ist abhängig von der Rahmenlänge. Im Kernel werden Kopiervorgänge und die Speicherreservierung bei langen Rahmen aufwändiger. Bei der Übertragung ist die Serialisierungsverzögerung von der Rahmenlänge abhängig. Der Versuch einer algorithmischen Ableitung der Messergebnisse ist mit den vorliegenden Daten jedoch nicht möglich, und die Werte bei Kommunikation über ein reales Netzwerk sind daher nur schwer vorab ermittelbar.

Der Vergleich der Kommunikationsverzögerungen zwischen virtuellem und realem Fall ist nur eingeschränkt möglich. Betrachtet man jedoch die gemessenen Werte, so sind schon für kurze Rahmen bei geringer Bitrate (und damit geringer Belastung des Knotens) deutliche Unterschiede mit 1,27 μs gegenüber 33,87 μs zu erkennen. Während die Laufzeiten im virtuellen Fall konstant bleiben, steigen sie mit zunehmender Rahmenrate und Rahmenlänge bei Kommunikation über das physische Netz auf Werte von 136,31 μs mit im Einzelfall deutlichen Schwankungen. Eine Überlastung des Knotens, die vor allem bei Broadcasts schnell erreicht wird, führt zu Laufzeiten die um Größenordnungen höher liegen und im Mittel Werte bis zu 7670 μs aufweisen. Demgegenüber ist bei virtueller Kommunikation nur ein konstanter Anstieg zu erkennen, der zu Laufzeiten von maximal 33,52 μs bei 31 Reflektoren führt.

Kapitel 6

Ausblick

6.1 Vergleich zwischen Emulation und Funkübertragung

In diesem Abschnitt werden die Messergebnisse der vorangegangenen Tests mit den bei Funkübermittlung entstehen Verzögerungen verglichen. Um den Vergleich durchführen zu können, werden zunächst die bei Funkübermittlung auftretenden Verzögerungen abgeschätzt.

Danach wird der Vergleich mit beiden möglichen Kommunikationspfaden bei Emulation im NET durchgeführt. Zunächst werden die Verzögerungen zwischen der virtualisierten Kommunikation und Funkübertragung verglichen, dann der Vergleich zwischen den Verzögerungen bei Kommunikation über den Switch und bei Funkübertragung gezogen. Für beide Fälle werden die Verzögerungen für Unicast- und Broadcast-Rahmen betrachtet.

Die Verzögerungen von Emulation und Funkübertragung können dabei auf zwei Merkmale untersucht werden. Sind die entstehenden Verzögerungen bei einer Emulation im Mittel kleiner als bei Funkübermittlung, so kann eine Emulation die Laufzeiten nur im Mittel und über den gesamten Zeitraum nachbilden. Der Vergleich wird dazu zwischen den Mittelwerten der Messergebnisse und den Verzögerungen bei Funkübermittlung gezogen.

Eine durchgehend zeitgenaue Emulation ist aber nur möglich, wenn die Verzögerungen bei einer Emulation immer geringer als die bei Funkübertragung entstehenden Verzögerungen sind. Dazu wird der ungünstigste Fall betrachtet, und die längste Laufzeit eines Rahmens aus den Messergebnissen mit der Laufzeit bei Funkübertragung verglichen.

Verzögerungen bei Funkübermittlung

Die Übertragungsverzögerungen bei Funkübermittlung setzen sich aus der Serialisierungsverzögerung, der Signalausbreitungsverzögerung und einer in den Kommunikationspartnern aufgewendeten Bearbeitungszeit zusammen.

In Tab. 6.1 ist die Serialisierungsverzögerung für DSSS-Rahmen der Bitübertragungsschicht berechnet. Die Rahmenlänge gibt die Länge der Nutzlast an. Bei DSSS-Rahmen sind insgesamt 48 Oktette zu addieren, um die real übertragenen Oktette zu erhalten. Aufgrund der geringeren Bandbreite sind die entstehenden Verzögerungen deutlich höher als bei dem im NET eingesetzten kabelgebundenen Netz mit 1 Gbps Bandbreite.

Die Signalausbreitungsgeschwindigkeit bei Funkübertragung ist die Lichtgeschwindigkeit. Die Signalausbreitungsverzögerungen für verschiedene Distanzen zwischen Sender und Empfänger sind in Tab. 6.2 angegeben.

		Rahmenlänge (Nutzlast in Oktetten)						
		64	128	256	512	1024	1500	2336
DSSS	1 Mbps	896,00	1408,00	2432,00	4480,00	8576,00	12384,00	18880,00
	2 Mbps	448,00	704,00	1216,00	2240,00	4288,00	6192,00	9440,00
	10 Mbps	89,60	140,80	243,20	448,00	857,60	1238,40	1888,00
	54 Mbps	16,59	26,07	45,04	82,96	158,81	229,33	349,63

Tabelle 6.1: Serialisierungsverzögerung bei Funkübermittlung (in μs)

Distanz	Verzögerung
3m	10 ns
10m	33,3 ns
30m	100 ns
100m	333 ns
300m	1000 ns

Tabelle 6.2: Signalausbreitungsverzögerung bei Funkübermittlung

Zusätzlich entsteht in den Kommunikationspartnern ebenfalls Aufwand für die Bearbeitung der Rahmen, der allerdings nur geschätzt werden kann. Aus den Messergebnissen aus Abschnitt 5.3 bei Kommunikation über das reale Netz werden in Test 10 und 11 die Laufzeiten in einzelne Phasen aufgeteilt. Dabei werden nur Werte von einigen Mikrosekunden für die Verzögerung zwischen der netshaper-Instanz und der Übergabe an die Netzwerkkarte festgestellt. Die Zeit zwischen der Übergabe an die Netzwerkkarte und dem Start des Sendevorgangs, ist jedoch nicht zu erkennen. Geht man von einer Rahmenlänge von 64 Oktetten aus, so beträgt die kürzeste gemessene Laufzeit 26,6 μs. Nach [Gro02] wird ein Rahmen innerhalb von 6 μs über einen zum FastIron II+ vergleichbaren Switch weitergeleitet. Die zweimalige Serialisierungsverzögerung vom Sender zum Empfänger beträgt ca. 1,4 μs. Betrachtet man die kürzeste Laufzeit als optimalen Fall mit geringstem Einfluss der Verarbeitung auf die Laufzeit, so kann aus den Anteilen ein minimaler Aufwand von 20 μs abgeschätzt werden (vgl. [Yan04]).

		Rahmenlänge (Nutzlast in Oktetten)						
		64	128	256	512	1024	1500	2336
DSSS	1 Mbps	936,10	1448,10	2472,10	4520,10	8616,10	12424,10	18920,10
	2 Mbps	488,10	744,10	1256,10	2280,10	4328,10	6232,10	9480,10
	10 Mbps	129,70	180,90	283,30	488,10	897,70	1278,50	1928,10
	54 Mbps	56,69	66,17	85,14	123,06	198,92	269,43	389,73

Tabelle 6.3: Übertragungsverzögerung bei Funkübermittlung (30 m Distanz, in μs)

Da in Funknetzen die Kommunikationspartner oftmals nur über geringe Rechenkapazitäten verfügen, wird der Schätzwert verdoppelt und führt zu 40 μs. Der geschätzte Bearbeitungsaufwand wird für die folgenden Betrachtungen unabhängig von der Rahmenlänge als konstant angenommen, und die Distanz zwischen Sender und Empfänger auf 30 m festgelegt. Daraus ergeben sich die Laufzeitverzöge-

rungen bei Funkübermittlung nach Tab. 6.3.

Vergleich virtualisierte Kommunikation und Funkübermittlung

Für Unicast-Rahmen wird im virtualisierten Fall eine mittlere Verzögerung von 1,485 μs gemessen. Vergleicht man den Wert mit den in Tab. 6.3 angegebenen Verzögerungen bei Funkübermittlung, so ist zu erkennen, dass die Verzögerung für alle Bitraten und Rahmenlängen immer deutlich geringer ist. Die Verzögerungen bei Funkübermittlung können also nachgebildet werden, und eine den realen Laufzeiten entsprechende, zeitgenaue Emulation ist daher im Mittel über die gesamte Laufzeit möglich.

Der ungünstige Fall liegt vor, vergleicht man die längste Laufzeit eines Rahmens aus den Messergebissen mit der kürzesten Laufzeit eines Rahmens gleicher Länge bei Funkübermittlung. In den Messergebnissen der Tests bei virtualisierter Kommunikation beträgt die längste gemessene Laufzeit eines Rahmens 66,76 μs (Test 2, 64 Oktette, 10 Mbps). Es handelt sich dabei allerdings um einen seltenen Ausreißerwert, wie die Verteilung der Rahmenlaufzeiten aus Test 1 in Abb. 5.2 zeigt. Die kürzeste Verzögerung bei Funkübertragung beträgt nach Tab. 6.3 geschätzte 56,69 μs bei 64 Oktetten Rahmenlänge und 54 Mbps. Eine Emulation ist in diesem Fall also nicht schnell genug, um eine zeitgenaue Nachbildung durchgehend zu ermöglichen.

Für Broadcast-Rahmen unterscheidet sich die Verarbeitungsweise von Rahmen bei Emulation gegenüber der Charakteristik bei realer Funkübertragung. Im virtualisierten Fall wird ein Rahmen geklont und sequentiell an die Empfänger ausgeliefert. Der Aufwand pro Klon ist mit ca. 0,8223 μs für jeden zusätzlichen virtuellen Knoten berechenbar. Geht man von 31 virtuellen Knoten aus, so liegen zwischen der Auslieferung des ersten und des letzten Rahmens ca. 25,49 μs. Der letzte Rahmen wird also im Mittel nach einer Zeit von insgesamt 26,98 μs ausgeliefert. Bei Funkübertragung erhalten alle Empfänger einen Broadcast-Rahmen nahezu gleichzeitig. Die Laufzeiten entsprechen also den bei Übermittlung von Unicast-Rahmen aus Tab. 6.3 geschätzten Werten. Eine zeitgenaue Emulation ist im Mittel also möglich, da die Laufzeiten im virtualisierten Fall unter den bei Funkübertragung geschätzten Werten liegen. Steigt die Anzahl der virtuellen Knoten jedoch, erhalten nicht alle Knoten den Rahmen innerhalb der Verzögerungszeit bei Funkübermittlung. Die zeitliche Genauigkeit einer Emulation ist bei Broadcast-Rahmen also abhängig von der Anzahl der virtuellen Knoten.

Im ungünstigsten Fall beträgt die Laufzeit eines Broadcast-Rahmens mit 64 Oktetten bei 31 virtuellen Knoten 90,93 μs. Die Laufzeit ist also für Rahmenlängen von 64–256 Oktetten bei 54 Mbps länger als bei Funkübermittlung. Bei großer Anzahl virtueller Knoten und hohen Bitraten ist eine zeitgenaue Emulation also nicht immer möglich.

Vergleich Kommunikation über den Switch und Funkübermittlung

Bei Kommunikation über den Switch entstehen längere Laufzeiten, wie die Messergebnisse der vorangegangenen Tests zeigen. Bei Unicast-Rahmen wird eine mittlere Laufzeit von 67,88 μs für Test 4–7 gemessen.

In Abb. 6.1 sind die Messwertkurven bei Versenden von Unicast-Rahmen über den Switch nochmals am Beispiel von Test 7 dargestellt. Die Messwerte geben die mittlere Laufzeit bei einer Bitrate und Rahmenlänge an. Bei Test 7 werden durch die hohe Hintergrundnetzlast und den Kommunikationspfad über den Switch lange Laufzeiten gemessen. Zusätzlich sind in der Abbildung die in Tab. 6.3 berechneten Verzögerungen bei Funkübertragung für alle Rahmenlängen abgebildet. Vergleicht man die Messkurven einer Rahmenlänge von Test 7 mit den Werten bei Funkübermittlung, so ist zu erkennen, dass die Laufzeiten bei Kommunikation über den Switch immer unter den Verzögerungen bei Funkübermittlung liegen. Eine zeitgenaue Emulation ist also im Mittel möglich.

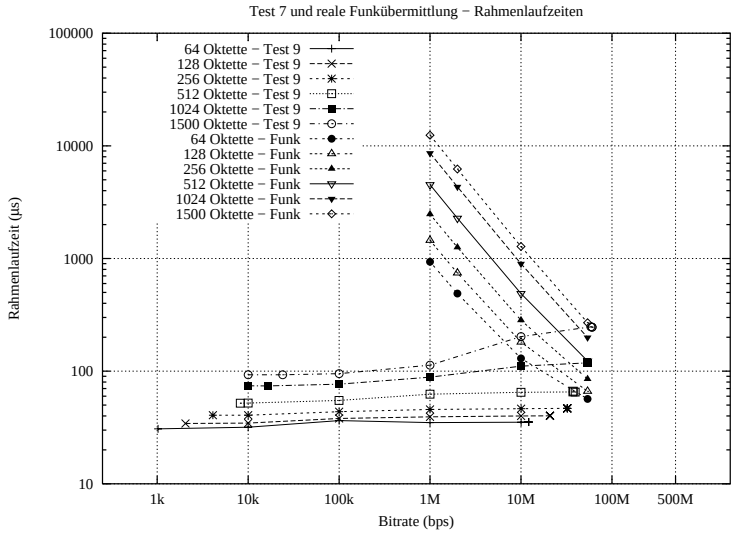


Abbildung 6.1: Test 7 - Rahmenlaufzeiten im Vergleich zu Funkübermittlung

Die maximale Laufzeit eines Unicast-Rahmens liegt mit 10029,49 μs jedoch erheblich höher. Im ungünstigsten Fall können die Laufzeiten bei Funkübermittlung also nicht in allen Fällen zeitgenau nachgebildet werden.

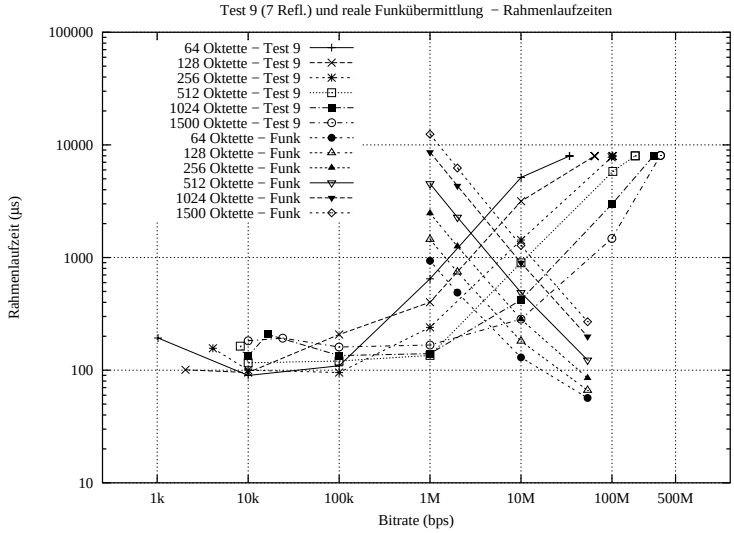


Abbildung 6.2: Test 9 mit 7 Reflektoren - Rahmenlaufzeiten im Vergleich zu Funkübermittlung

Bei Versenden von Broadcast-Rahmen entstehen bei Kommunikation über den Switch noch längere Verzögerungen. Die Messwerte von Test 9 mit 7 Reflektoren sind in Abb. 6.1 dargestellt und zusätzlich ebenfalls die geschätzten Laufzeiten bei Funkübermittlung. Der Vergleich der Kurven einer Rahmenlänge zeigt, dass die Laufzeiten bei Kommunikation über den Switch nicht durchgängig unter denen bei Funkübermittlung liegen.

Dabei lassen sich zwei Fälle unterscheiden. Bei niedrigen Bitraten bis zu 1 Mbps sind die physischen Knoten nicht überlastet, und die Laufzeit ist mit im Mittel 192,85 μs geringer als die bei Funkübertra-

gung entstehenden Verzögerungen. Dies gilt ebenfalls für 2 Mbps.

Bei höheren Bitraten steigen die Laufzeiten jedoch stark an. Der physische Knoten ist überlastet, und es entstehen Verzögerungen von im Mittel 7218,11 μs mit einer maximalen Laufzeit einzelner Rahmen von bis zu 12936,09 μs. Die Emulation ist damit langsamer als Funkübermittlung. Zu beachten ist jedoch, dass die bei der Bearbeitung eines Rahmens entstehenden Verzögerung bei Funkübermittlung für den Vergleich als konstant angenommen wurden, und damit für höhere Bitraten wahrscheinlich zu niedrig liegen. Dennoch ist bei Überlastung der Knoten anzunehmen, dass die Laufzeiten im NET bei Kommunikation über den Switch länger sind als bei Funkübermittlung.

Für Unicast- und Broadcast-Rahmen lässt sich also bei Kommunikation über den Switch feststellen, dass der Einfluss der Verarbeitung von Rahmen in den physischen Knoten so stark ist, dass bei Überlast die Laufzeiten länger als bei Funkübermittlung einzuschätzen sind. Ist der Knoten dagegen nicht überlastet, kann eine zeitgenaue Emulation im Mittel durchgeführt werden. Die Werte liegen, abgesehen von überlangen Laufzeiten einzelner Rahmen, immer unter denen bei Funkübermittlung.

Die Güte der Emulation ist also entscheidend von der Auslastung der physischen Knoten abhängig. Die Belastung steigt dabei mit Anzahl der emulierten Knoten in einem Szenario und der verwendeten Bitrate.

6.2 Lösungsansätze

Für die zeitliche Genauigkeit der Rahmenauslieferung bei einer Emulation lassen sich zwei Strategien verfolgen. Zum einen kann versucht werden, die Ankunftszeiten über die gesamte Emulationszeit im Mittel den bei der Funkübermittlung entstehenden Verzögerungen entsprechen zu lassen. Eine zweite Möglichkeit stellt die Betonung der Gleichzeitigkeit von Ankunftszeiten unabhängig vom absoluten Wert der Verzögerung dar. Da das Laufzeitverhalten weitgehend von den physischen Knoten beim Empfang eines Rahmens bestimmt wird, ist in beiden Fällen die Lastsituation im Empfänger entscheidend.

Die erste Strategie kann im Network Emulation Testbed in Szenarien verwirklicht werden, in denen die Zahl der emulierten Knoten nicht zu einer Überlastung der physischen Knoten führt. Die Laufzeiten im NET sind meist länger und eine entsprechende, berechenbare Verzögerung vor der Auslieferung im Empfänger kann die Laufzeitdifferenzen zwischen den beiden Kommunikationspfaden an die realen Werte bei Funkübertragung angleichen. Bei geringen Bitraten können also größere Szenarien zum Einsatz kommen.

Soll eine gleiche Ankunftszeit für alle Lastsituationen im Netz bei allen emulierten Knoten garantiert werden, ist vom ungünstigsten Fall der Verzögerung im NET auszugehen. Die Verzögerung ist dann mit 12 ms abzuschätzen, die durch das Polling des Kernels beim Empfang von Rahmen über die Netzwerkkarte bei hoher Belastung entsteht. Die Verzögerung eines Rahmens müsste für alle Knoten bei Empfang vor der Auslieferung eingefügt werden.

Die derzeitige Implementierung des Netshaper-Moduls enthält bereits Mechanismen zur Verzögerung von Rahmen. Allerdings werden die Rahmen im Sender verzögert. Die Ankunftszeit wird aber weitgehend durch die Last des Empfängers bestimmt, so dass eine Verzögerung an dieser Stelle erfolgen muss. Ferner ist die Abbildung kurzer Verzögerungen im Linux-Kernel nur bedingt möglich. Durch die Granularität der Prozessverwaltung von 10 ms können kurze Verzögerungen bisher nicht genau genug implementiert werden. Eine Erweiterung, die den APIC-Timer verwendet und eine höhere Genauigkeit aufweist, ist jedoch in Planung. Damit wäre es möglich, die Rahmen im Empfänger mit entsprechender Genauigkeit zu verzögern, und somit ein realistisches Verhalten zu erzeugen.

6.3 Fazit

Ziel bei Emulationen im Network Emulation Testbed ist es, eine möglichst genaue Übereinstimmung zwischen dem Verhalten eines emulierten Netzwerkes und den realen Eigenschaften eines Netzwerks zu erreichen. Ein Aspekt der genauen Emulation von mobilen ad-hoc Netzwerken sind dabei Verzögerungszeiten, die bei der Kommunikation auftreten.

Zur Ermittlung der Verzögerungszeiten im NET wurde in Kapitel 1 zunächst die Emulationsumgebung beschrieben. In Kapitel 2 wurden die grundlegenden Zusammenhänge dargestellt, die bei Übermittlung von Rahmen zu einer Verzögerung führen, und Ansätze vorgestellt, die für die Implementierung eines Messwerkzeuges herangezogen werden können. Darauf aufbauend wurde in Kapitel 3 die erstellte Messsoftware vorgestellt. Die Software wird in Kapitel 4 eingesetzt, um Verzögerungszeiten im NET bei unterschiedlichen Lasten und Szenarien zu ermitteln. Die Messergebnisse der durchgeführten Tests zeigen, dass sich die Verzögerungen zwischen virtueller Kommunikation und dem Versenden von Rahmen über ein physisches Netzwerk deutlich unterscheiden. Neben den Laufzeitunterschieden bei verschiedenen Rahmenlängen und Bitraten, ist vor allem die unterschiedliche Charakteristik für die verschiedenen Kommunikationspfade festzustellen.

Der Vergleich zwischen Emulation und realer Funkübertragung im abschließenden Kapitel führt zum Schluss, dass realistische Emulationsergebnisse erreicht werden können, wenn die Belastung der Knoten nicht zu hoch ist. Die bei Überlastung überproportional ansteigenden Verzögerungen liegen jedoch bei hoher Netzlast über denen in MANETs. Um dem entgegenwirken zu können, werden abschließend Lösungshinweise gegeben, die bei zukünftigen Erweiterungen der Emulationsumgebung in Betracht gezogen werden können.

Anhang A

Kommunikationsmatrix von Test 7

Quelle		Ziel	
Knoten	Portgruppe-Slot	Knoten	Portgruppe-Slot
c02	1-2	c25	4-1
c03	1-3	c26	4-2
c04	1-4	c27	4-3
c05	1-5	c28	4-4
c06	1-6	c29	4-5
c07	1-7	c30	4-6
c08	1-8	c31	4-7
c09	2-1	c02	1-2
c10	2-2	c57	8-1
c11	2-3	c03	1-3
c13	2-5	c60	8-3
c14	2-6	c04	1-4
c15	2-7	c61	8-4
c17	3-1	c57	8-1
c20	3-4	c08	1-8
c21	3-5	c60	8-3
c22	3-6	c02	1-2
c23	3-7	c61	8-4
c24	3-8	c03	1-3
c41	6-1	c62	8-5
c43	6-3	c04	1-4
c44	6-4	c63	8-6
c46	6-6	c05	1-5
c47	6-7	c57	8-1
c48	6-8	c06	1-6
c51	7-3	c05	1-5
c52	7-4	c62	8-6
c53	7-5	c06	1-6
c54	7-6	c63	8-7
c55	7-7	c07	1-7
c57	8-1	c33	5-1
c60	8-4	c34	5-2
c61	8-5	c36	5-3
c62	8-6	c37	5-4
c63	8-7	c29	5-5

Literaturverzeichnis

[AP99] Mark Allman and Vern Paxson. On estimating end-to-end network path properties. In *SIGCOMM*, pages 263–274, 1999.

[BC02] Daniel P. Bovet and Marco Cesati. *Understanding the Linux Kernel*. O'Reilly, 2nd edition, 2002.

[BNK97] G. Nguyen B. Nobel, M. Satyanarayanan and R. Katz. Trace-Based Mobile Network Emulation. In *Proceedings of the ACM SIGCOMM, Cannes, France*, 1997.

[CPB93] Kimberly Claffy, George Polyzos, and Hans-Werner Braun. Measurement Considerations for Assessing Unidirectional Latencies. In *Journal of Internetworking*, 1993.

[FML+03] Chuck Fraleigh, Sue Moon, Bryan Lyles, Chase Cotton, Mujahid Khan, Deb Moll, Rob Rockell, and Ted Seely. Packet-Level Traffic Measurements from the Sprint IP Backbone. Technical report, IEEE Network, November 2003.

[Fou03] Foundry Networks, Inc. *JetCore Base Chassis Systems - An Architecture Brief on NetIron BigIron, and FastIron Systems*, 2003.

[Fou04a] Foundry Networks, Inc. *Foundry Diagnostic Guide*, July 2004.

[Fou04b] Foundry Networks, Inc. *Foundry Switch and Router Command Line Interface Reference*, July 2004.

[Fou04c] Foundry Networks, Inc. *Foundry Switch and Router Installation and Basic Configuration Guide*, July 2004.

[GPMD97] Ian D. Graham, Murray Pearson, Jed Martens, and Stephen Donnelly. Dag - a cell capture board for ATM measurement systems. April 1997.

[Gro02] The Tolly Group. FastIron400/FastIron Edge 9604 - Performance, High Aviability & Voice Quality. Technical report, Foundry Networks, Inc., December 2002.

[Her02] Helmut Herold. *Linux-Unix-Systemprogrammierung*. Addison-Wesley, 4th edition, 2002.

[HM04] Daniel Herrscher and Steffen Maier. *NET: The Network Emulation Testbed Manual*. Institut für parallele und verteilte Systeme, Universität Stuttgart, July 2004.

[HR02] Daniel Herrscher and Kurt Rothermel. A Dynamic Network Scenario Emulation Tool. In *Proceedings of the 11th International Conference on Computer Communications and Networks (ICCCN 2002)*, pages 262–267, Miami, October 2002.

[Pax98] Vern Paxson. On Calibrating Measurements of Packet Transit Times. In *Proceedings of SIGMETRICS*, June 1998.

[Pos81] J. Postel. Internet Control Message Protocol. RFC 792 (Standard), September 1981. Updated by RFC 950.

[RC01] Alessandro Rubini and Jonathan Corbet. *Linux Device Drivers*. O'Reilly, 2nd edition, June 2001.

[Rot02] Jörg Roth. *Mobile Computing*. dpunkt Verlag, 2002.

[Sch00] Jochen Schiller. *Mobilkommunikation - Techniken für das allgegenwärtige Internet*. Addison-Wesley, 2000.

[Ste90] Richard Stevens. *Unix Network Programming Volume 1*. Prentice Hall, 2nd edition, 1990.

[Tan96] Andrew S. Tanenbaum. *Computer Networks*. Prentice Hall, 3rd edition, 1996.

[WPR+02] Klaus Wehrle, Frank Pählke, Hartmut Ritter, Daniel Müller, and Marc Bechler. *Linux Netzwerkarchitektur: Design und Implementierung von Netzwerkprotokollen im Linux-Kern*. Addison-Wesley, 2002.

[Yan04] Zhenxiang Yang. Entwicklung eines Verfahrens zur Emulation der Medienzugriffssteuerung in Wireless LAN. Diplomarbeit, Universität Stuttgart, May 2004.