Hochauflösende Verkehrsformung für Dienstgüteunterstützung in Software-Routern

cand. Inform. Uwe Walter, Dipl.-Inform. Klaus Wehrle
{walter,wehrle}@tm.uka.de

Institut für Telematik
Universität Karlsruhe
D–76128 Karlsruhe

Betreuer der Arbeit: Prof. Dr. Martina Zitterbart
Art der Arbeit: Forschungsprojekt, Studienarbeit
Fachbereich GI: 3

1 Einführung

Die bisher vom Internet zur Verfügung gestellte Qualitätsgarantie namens Best-Effort wird modernen Multimedia-Protokollen und gewachsenen Kundenanforderungen in steigendem Ausmaß nicht mehr gerecht. Aus diesem Grund ist die Unterstützung von Dienstgüte (Quality of Service - QoS) ein Thema, das zunehmend an Bedeutung gewinnt.

Zum Aufbau dieser leistungsfähigeren Dienstgütegarantien sind verschiedene Mechanismen erforderlich, von denen ein wesentlicher Baustein die Möglichkeit zur Verkehrsformung darstellt. Verkehrsformer glätten die Bandbreitennutzung eines Datenstroms auf eine einstellbare Rate, indem sie zu schnell aufeinanderfolgende Datenpakete um eine entsprechende Zeit verzögern.

Auf diesem Weg reduzieren Verkehrsformer die Pakethäufungen (Bursts) einer Datenübertragung und sorgen so für die bessere Einhaltung von Dienstgütevereinbarungen. Ebenso werden die Schwankungen im Abstand zweier Datenpakete verringert, was Protokollen und Anwendungen zu Gute kommt, die auf einen gleichmäßig eintreffenden Paketstrom angewiesen sind (bspw. Multimedia-Anwendungen).

Um Datenverkehr effektiv formen zu können, werden vor allem zwei Dinge benötigt:

- Es muss eine *hochauflösende Zeitmessung* möglich sein, z.B. um den Ankunftszeitpunkt eines Pakets genau erfassen zu können.
- Die Möglichkeit, zu ebenso präzise definierten Zeitpunkten, Aktionen auszulösen, muss gegeben sein, um bspw. ein Paket pünktlich abzusenden. Hierfür werden hochgenaue Unterbrechungsgeneratoren (Timer) benötigt, da die Paketabstände in Hochleistungsnetzwerken mit Bandbreiten im Mbps- oder gar Gbps-Bereich, sehr klein sind.

Im stark anwachsenden Internet werden zunehmend Standard-PC-Komponenten als sogenannte Software-Router für den Netzzugang verwendet. Somit stellt sich die Frage, in welchem Ausmaß diese Endgeräte Möglichkeiten zur Verkehrsformung unterstützen und wie gut sie die dafür notwendigen Bedingungen erfüllen.

Sehr häufig wird das freie Betriebssystem Linux auf solchen Software-Router eingesetzt, da es sich dank seiner Stabilität, einer umfassenden Netzwerkunterstützung und der einfachen Erweiterbarkeit in diesem Bereich bewährt hat. Aus diesen Gründen basiert auch die folgende Arbeit auf diesem Betriebssystem.

2 Die Zeit im Linux-Kern

Im Linux-Kern „*ticken*" die Uhren etwas anders als in der realen Welt. Die Zeit schreitet nicht kontinuierlich voran, sondern in Schritten von 10 ms, die auch als *Ticks* oder *jiffies* bezeichnet werden. Zwischen zwei Ticks bleibt die Zeit sozusagen stehen.

Die Auslösefrequenz des Timer-Interrupts wird mit dem Wert der Variable HZ initialisiert und aktualisiert alle $\frac{1}{HZ}s$ die jiffies-Variable[1]. Für normale Anwendungen ist diese Zeitspanne völlig ausreichend, weil eine höhere Interrupt-Frequenz zur Inkrementierung der *Ticks* nur eine höhere Belastung des Systems durch zu viele, meist unnötige Unterbrechungen bedeuten würde. Es gibt aber bestimmte Situationen, in denen eine hohe Timerauflösung benötigt wird, vor allem beim Messen kleiner Zeitabstände [RiWe00]. Für solche Aufgaben werden Uhren mit einer Auflösung benötigt, die mindestens im μs-Bereich liegen. So muss beispielsweise für die Implementierung eines Verkehrsformers die Anzahl der Bytes berechnet werden, die innerhalb einer Zeitspanne gesendet werden dürfen. Dafür ist die jiffies-Zeitmessung mit einer Auflösung von 100 Hz nicht geeignet. Bei einer Rate von 2 Mbit/s entspricht eine Zeitspanne von $\frac{1}{100}s$ schon einem Paket der Länge 2500 byte.

Um diese Problematik zu umgehen, verfügen modernen Prozessoren (Pentium, Alpha, etc.) über entsprechende Register. Im Falle des Pentium-Prozessors und seinen Nachfolgern (sowie den meisten Clones) bietet der Prozessor ein sogenanntes 64 bit breites TSC-Register (*Time Stamp Counter*), dessen Inhalt bei jedem Prozessortaktschritt um Eins inkrementiert wird. Am Inhalt dieses Registers kann die Anzahl der vergangenen Taktzyklen seit dem Systemstart abgelesen werden.

Das TSC-Register ist eigentlich nichts anderes als eine Hardware-Variante von jiffies – nur mit einer um den Faktor 10^6 bis 10^8 größeren Auflösung. Bei einem Pentium-Prozessor mit einer Taktrate von 1 GHz wird es dadurch möglich, Zeitabstände mit einer Genauigkeit im Bereich von 0.001 μs zu messen.

Bei Messungen mit dem TSC-Register entsteht trotzdem eine gewisse Ungenauigkeit, weil das Auslesen des Registers auch einige Takte benötigt (ca. 10 Takte=0.01 μs benötigt. Der Grund hierfür liegt im Hauptspeicherzugriff, welcher nach dem Auslesen des Registerwerts erfolgt (1-2 Takte). Für normale Messungen ist der Fehler, welchen das TSC-Register erzeugt jedoch bedeutungslos, weil oft nur, für diese Zeitauflösung relativ große Zeitabschnitte (im Bereich von 1 μs) gemessen werden.

2.1 Standard Timer

Für die Implementierung eines Verkehrsformer (Traffic-Shapers), welcher Pakete zu bestimmten Zeitpunkten versendet, wird neben der Messung von Zeitabständen im Mikrosekundenbreich auch die Möglichkeit benötigt, eine Funktion zu einem bestimmten Zeitpunkt ausführen zu können. Die Auflösung des *Timers* sollte mindestens im 100 μs-Bereich liegen. Weil ein PC jedoch nur über *einen* Timerbaustein verfügt, kann nur dieser verwendet werden. Er wird, wie oben beschrieben, HZ-mal in der Sekunde ausgelöst. Neben der Aktualisierung der jiffies nutzt Linux den Timer-Interrupt auch für die Ausführung von Funktionen zu bestimmten Zeitpunkten.

2.2 Hochauflösende Timer mit APIC

Der Linux Kern verfügte bisher über keinerlei Unterstützug für frei programmierbare Timer mit einer Genauigkeit im Mikrosekundenbereich.

Bisher wurden Zeitgeber nur auf Basis des Timer-Interrupts zur Verfügung gestellt und besitzen deshalb nur eine Genauigkeit von $\frac{1}{\mathsf{HZ}}$Sekunden. Die nachfolgend vorgestellte Timerfunktionalität auf Basis des APIC-Bausteins (kurz *APIC-Timer*) versucht diese Probleme zu umgehen. Der APIC-Timer wurde am Institut für Telematik der Universität Karlsruhe entwickelt [RiWe00] und kann unter [APIC01] geladen werden.

Die x86-Prozessor-Familie von Intel benutzte ursprünglich den Programmable Interrupt Controller PIC 8259A für die Verwaltung ausgelöster Interrupts. Er wurde seit dem ersten Personal Computer Anfang der achtziger verwendet und erfüllte seine Aufgabe ohne Probleme. Jedoch wurde für die Multiprozessor-Fähigkeit die Möglichkeit benötigt, ausgelöste Interrupts zwischen verschiedenen Prozessoren eines SMP-Rechners zu verteilen. Aus diesem Grund führte Intel den sogenannten APIC (Advanced Programmable Interrupt Controller) ein.

Der *Local APIC* ist (seit dem Pentium P54C) in alle Pentium-Prozessoren integriert. Jeder Local APIC verfügt über mehrere 32-bit-Register, eine interne Uhr, einen internen Timer, 240 Interrupt-Vektoren und zwei zusätzliche Interrupt-Leitungen, die für lokal generierte Interrupts verwendet werden können.

[1] HZ ist architekturabhängig - bei Alpha-Prozessoren beträgt HZ = 1024, bei den meisten anderen Architekturen gilt HZ = 100.

Für die schon beschriebenen Aufgaben ist vor allem der interne Timer interessant. Er arbeitet mit Bustaktgenauigkeit und kann mit einem bestimmten Wert initialisiert werden. Anschließend wird bei jedem Bustakt der Wert des Timers dekrementiert und bei Erreichen des Wertes Null wird ein Interrupt ausgelöst. Somit lässt sich mit dem internen Timer des APIC-Bausteins ein hochauflösender Timer mit nahezu Bustaktgenauigkeit realisieren.

In Uniprozessorsystemen werden die Local APICs von den meisten Betriebssystemen beim Systemstart nicht aktiviert. Bei älteren P5-Prozessoren ließ sich der Local-APIC-Baustein nur zu diesem Zeitpunkt aktivieren, eine spätere Initialisierung konnte nur durch Hardware-Manipulation erreicht werden. Seit der P6-Prozessor-Generation (Pentium Pro und Nachfolger) kann der integrierte Local APIC auch während des Betriebs per Software-Befehle aktiviert werden. Er lässt sich somit für die Realisierung eines hochauflösenden Timers verwenden.

Arbeitsweise des APIC-Timers Dabei sei aber nochmals darauf hingewiesen, dass sich der Local APIC nur in Uniprozessor-Rechnern für einen frei programmierbaren Timer einsetzen lässt, da der Timer des Local-APIC in Multiprozessor-Systemen für die Interprozessor-Synchronisation verwendet wird.

In einigen Versionen des Linux-Kerns 2.3 konnte der Local-APIC-Baustein über ein Modul wieder aktiviert werden. Leider ist dieses Modul in den 2.4er Versionen nicht mehr enthalten. Es gibt jedoch einen Patch, der eine Aktivierung des Local-APIC in Uniprozessor-Systemen zur Laufzeit ermöglicht. Basierend auf einem aktivierten Local APIC und dessen integriertem Timer wurde ein hochauflösender Timer entwickelt, der eine ähnliche Programmierschnittstelle aufweist, wie die Standard-Timer des Linux-Kerns [RiWe00,APIC01].

3 Timer-Anwendung für die Verkehrsformung

Im Zuge einer Studienarbeit wurde am Institut für Telematik der Universität Karlsruhe die Linux-Ausprägung der Dienstgüte-Architektur KIDS (*Karlsruhe Implementation Architecture for Differentiated Services* [Wehr01]) um die notwendigen Mechanismen zur Verkehrsformung erweitert [Walt01].

Dabei wurde die Performance des implementierten Verkehrsformers unter Verwendung der folgenden Timer untersucht:

1. Der Standard-PC-Timer. Seine Auslösefrequenz liegt standardmäßig bei 100 HZ, die erreichbare Genauigkeit somit bei 10ms.
2. Ein modifizierter Standard-PC-Timer. Die Auslösefrequenz wurde manuell auf 5000 HZ erhöht. Die dadurch erreichbare Genauigkeit von 0,2ms wird durch eine deutlich höhere Systembelastung durch die vielen Interrupts erkauft.
3. Der UKA-APIC-Timer, aus Abschnitt 2.2 (kurz: APIC-Timer).

Als Testsystem wurde ein Standard-PC mit einer 700 MHz CPU und einer 100 Mbps Ethernet-Verbindung verwendet.

3.1 Timer-Raster und Abweichung

In Abbildung 1 ist für alle drei verwendeten Timer dargestellt, wie weit der tatsächliche Sendezeitpunkt im Durchschnitt vom geplanten abweicht, wenn der Sendemechanismus Pakete um eine gleichverteilte Zeit verzögern soll.

Bei Verwendung des Standard-PC-Timers beträgt die Abweichung im Durchschnitt 4,75ms, was praktisch genau dem halben Intervall zwischen zwei Timer-Interrupts entspricht. Das gleiche gilt für den modifizierten Standard-PC-Timer, der mit einem durchschnittlichen Fehler von $84\mu s$ ebenfalls in der Mitte seines 200 Mikrosekunden breiten Timer-Rasters liegt. Erst die Nutzung des APIC-Timers erlaubt das Absenden von Datenpaketen mit einer durchschnittlichen Abweichung von nur $1,9\mu s$. Genauere Histogramme der Abweichungen der verschiedenen Timer finden sich in Abbildung 2.

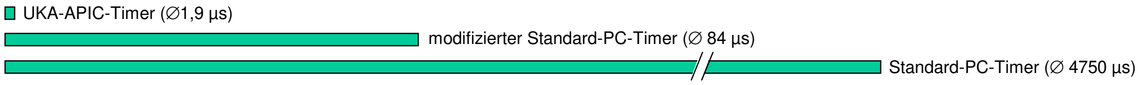

Abbildung1. Durchschnittliche Abweichung des tatsächlichen vom geplanten Sendezeitpunkt unter Verwendung der verschiedenen Timer.

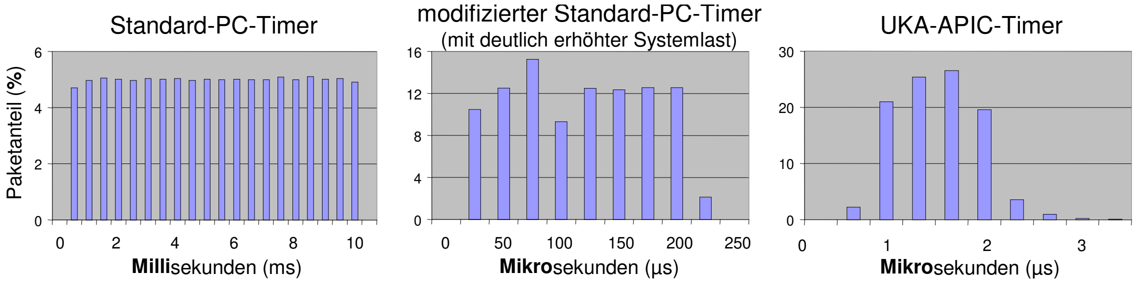

Abbildung2. Abweichung des tatsächlichen vom geplanten Sendezeitpunkt bei der Verkehrsformung unter Verwendung der verschiedenen Timer.

3.2 Verkehrsformung

Das Verhalten des Verkehrsformers ist direkt von der Auslösegenauigkeit und dem Raster des verwendeten Timers abhängig. Zur Evaluierung wurde ein konstanter UDP-Paketstrom durch den Verkehrsformer geleitet und danach die Zwischenankunftszeit benachbarter Pakete betrachtet. In allen Versuchen war die Datenrate des eingehenden Verkehrs etwas größer als die konfigurierte Ausgangsrate des Verkehrsformers, so dass immer Datenpakete zum Verschicken zur Verfügung standen.

In Abbildung 3 ist das Ergebnis eines solchen Versuchs dargestellt, in dem ein Datenstrom, bestehend aus 1292 byte großen UDP-Paketen, auf eine Bandbreite von 1 Mbps geformt werden sollte. Bei diesem Szenario musste somit im Idealfall alle 10,3ms ein Datenpaket versandt werden.

Unter Verwendung des Standard-PC-Timers bleibt aufgrund seines 10ms großen Auslöseintervalls nur die Möglichkeit, ein Datenpaket entweder genau 10ms oder genau 20ms zu verzögern. Den Soll-Mittelwert von 10,3ms erreicht dieser Timer durch sporadische Verzögerung einzelner Datenpakete um 20ms, während er den Rest mit 10ms etwas zu früh versendet.

Der modifizierte Standard-PC-Timer kann mit seinen im 0,2ms-Raster auslösbaren Timern diese Aufgabe deutlich besser lösen. Auch er pendelt im Normalfall zwischen den zwei naheliegendsten Timer-Zeitpunkten.

Bei Verwendung des APIC-Timers ergibt sich ein glatter Datenstrom, da dessen Ungenauigkeit von etwa $2\mu s$ in diesem Beispiel im dargestellten Maßstab nicht ersichtlich ist.

Während der Standard-PC-Timer schon mit dieser Aufgabe über den Grenzen seiner Leistungsfähigkeit operiert, können die anderen zwei Zeitgeber noch deutlich höher belastet werden. In Abbildung 4 ist das Resultat eines weiteren Versuches dargestellt, in dem ein Datenstrom aus 667 byte großen Paketen auf eine Datenrate von 50 Mbps geglättet werden sollte. Um dies zu erreichen, musste alle $107\mu s$ ein Paket versandt werden.

Bei diesem Versuch gerät auch der modifizierte Standard-PC-Timer an die Grenzen seiner maximale Auslösefrequenz. Er kann den Sollabstand nicht gleichmäßig einhalten, sondern verschickt Pakete entweder unverzögert oder um einen vollen 0,2ms-Timer verspätet.

Bei Verwendung des APIC-Timers lässt sich auch bei dieser Bandbreite noch keine störende Ungenauigkeit feststellen.

In anschließenden Belastungstests wurde mit dem APIC-Timer Verkehrsformung bis zu einem minimalen Paketabstand von $17\mu s$ betrieben, bevor andere Grenzen des Testsystems (Verkehrsgenerierung, Datenversand)

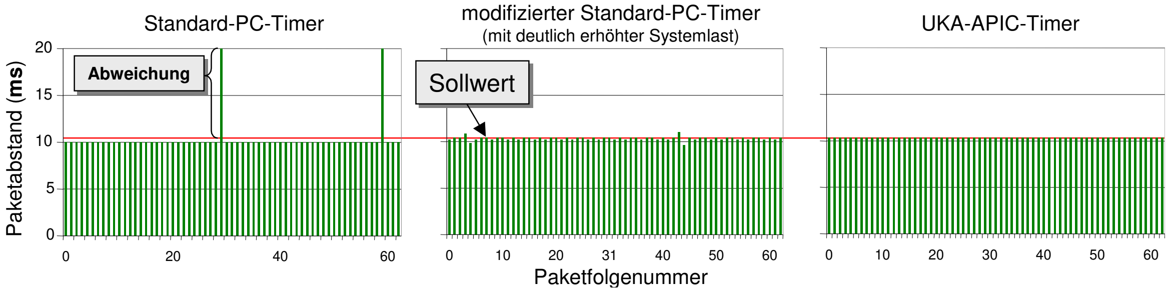

Abbildung3. Paketabstände eines auf 1 Mbps geformten Datenstroms, bestehend aus 1292 byte großen Paketen, entsprechend einer Soll-Abstandszeit von 10,3ms.

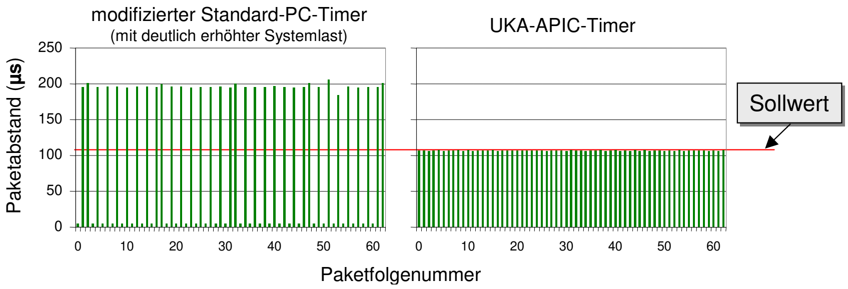

Abbildung4. Paketabstände eines auf 50 Mbps geformten Datenstroms, bestehend aus 667 byte großen Paketen, entsprechend einer Soll-Abstandszeit von $107\mu s$.

erreicht wurden. Dies ergibt eine Paketfrequenz von 58800 Paketen pro Sekunde, was selbst bei Verwendung von kleinen 100 byte großen Paketen einer Übertragungsrate von über 47 Mbps entspricht.

4 Zusammenfassung

In der vorliegenden Arbeit wurde unter Verwendung eines neu entwickelten und hochgenauen Zeitgebers die Möglichkeit geschaffen, auf einem Software-Router effizient hochauflösende Verkehrsformung zu betreiben. Die erreichte Genauigkeit für den Sendezeitpunkt eines Datenpaketes liegt dabei auch bei hohen Bandbreiten im Mikrosekundenbereich.

Literatur

APIC01. UKA-APIC-Timer Homepage. http://www.tm.uka.de/Forschung/apic/, Institut für Telematik, Universität Karlsruhe (TH), Januar 2001.

BlWe99. Roland Bless und Klaus Wehrle. Evaluation of Differentiated Services using an Implementation under Linux. In *Proceedings of the 7th IFIP Workshop on Quality of Service, London*. IEEE, Juni 1999.

RiWe00. Hartmut Ritter und Klaus Wehrle. Traffic Shaping in ATM and IP networks using standard end systems. In *Proceedings of Conference on High Performance Switching & Routing, Joint IEEE/ATM Workshop 2000 and 3rd International Conference on ATM, Heidelberg*, Juni 2000.

Walt01. Uwe Walter. Implementierung von Verkehrsformern unter Linux. Studienarbeit am Institut für Telematik, Universität Karlsruhe (TH), Januar 2001.

Wehr01. Klaus Wehrle. An open Architecture for Evaluating Arbitrary Quality of Service Mechanisms in Software Routers. In *Proceedings of IEEE International Conference on Networking (ICN 2001), Colmar, France*. Springer, Juli 2001.