Implementierung und Vergleich verschiedener Strategien zur Durchführung von Ethernet-Performancemessungen

Michael Gernoth[1] Jochen Reinwand[1] Stephan Kraft[1]
Verena Venus[2] Roland Karch[2] Ralf Kleineisel[2] Birgit König[2]

[1]Friedrich-Alexander Universität Erlangen-Nürnberg

[2]Verein zur Förderung eines Deutschen Forschungsnetzes, Berlin

1. Dezember 2006

Einsatzgebiete

- Dienstgütebestimmung in
  - lokalen Netzen
  - auf Layer-2 verbundenen Weitverkehrsnetzen
  - getunnelten Netzen
- Optimierung von Kommunikationswegen für Anwendungen im Echtzeitbereich (z.B. Videoübertragung im VIOLA-Projekt)
- Einsatz auf vorhandener Hard-/Softwareplatform

IPPM(IP Performance Metrics)-Messungen

- Layer-4-Messung mit UDP-Paketen
- Routerübergreifende Messungen möglich
- Einsatz zur Dienstgütebestimmung von Weitverkehrsnetzen (X-WiN, Internet)

Probleme

- Zeitstempelbestimmung im Userspace
- Messpaket durchläuft den kompletten Netzwerkstack mit Firewall
- Kontextwechsel verfälschen Messung
- Scheduling und Paging verfälscht Messung

→Lösung: neues Messverfahren auf Ethernet Layer-2

Layer-2-Messverfahren

- Layer-2-Messung mit eigenen Ethernet-Frames
- Nicht routbar
- Zeitstempelbestimmung auch im Systemkern
- Bestimmung des One-Way-Delays

Frameaufbau

- Ethernet Header
- Payload
  - Sequenznummer
  - Sendezeitstempel
  - Informationen über Senderzeitquelle
  - Informationen über eingesetzte Senderimplementierung
  - Platz für Empfangszeitstempel
  - Platz für Empfängerinformationen
  - Variabler Datenbereich zur Paketgrößenvariation

Frameaufbau

0	1	2	3	4	5	6	7	8	9	10	11	12	13
Zieladresse 00:11:22:33:44:55						Quelladresse 00:55:44:33:22:11						Typ 0x0815	
Sequence unsigned long				time_sent struct timespec								time_src_ time_src_	
send t		precision_send precision_t				sarimi_type_send sarimi_type_t				time_src_recv time_src_t			
precision_recv precision_t				sarimit_type_recv sarimi_type_t				time_received struct timespec					
... ...		data_len size_t				data... char[]							

Kontextwechsel

- Dauer eines Kontextwechsels im Bereich $1\mu s - 4\mu s$ auf moderner Hardware
- Mindestens 2 ,,kritische" Kontextwechsel, falls Uhrzeit im Userspace bestimmt werden muss

Auflösung der Uhrzeit

- Uhrzeitfunktionen liefern Zeitstempel mit Mikrosekundenauflösung
- Zugriff auf xtime-Kernelvariable bietet Nanosekundenauflösung, jedoch nur Millisekundengenauigkeit
- Nanosekundengenauigkeit wünschenswert, aber im Moment nicht erreichbar

Zeitsynchronisierung

- Einwegmessung, daher genaue Zeitbestimmung an beiden Messpunkten nötig
- Echtzeituhren der an der Messung beteiligten Rechner müssen so synchron wie möglich laufen
- Dies wird durch GPS-Empfänger an jeder Messstation erreicht

Implementierungen

- Bibliotheksbasierte Messung (libnet & libpcap)
- Raw-Socket-basierte Messung
- Kernelbasierte Messung

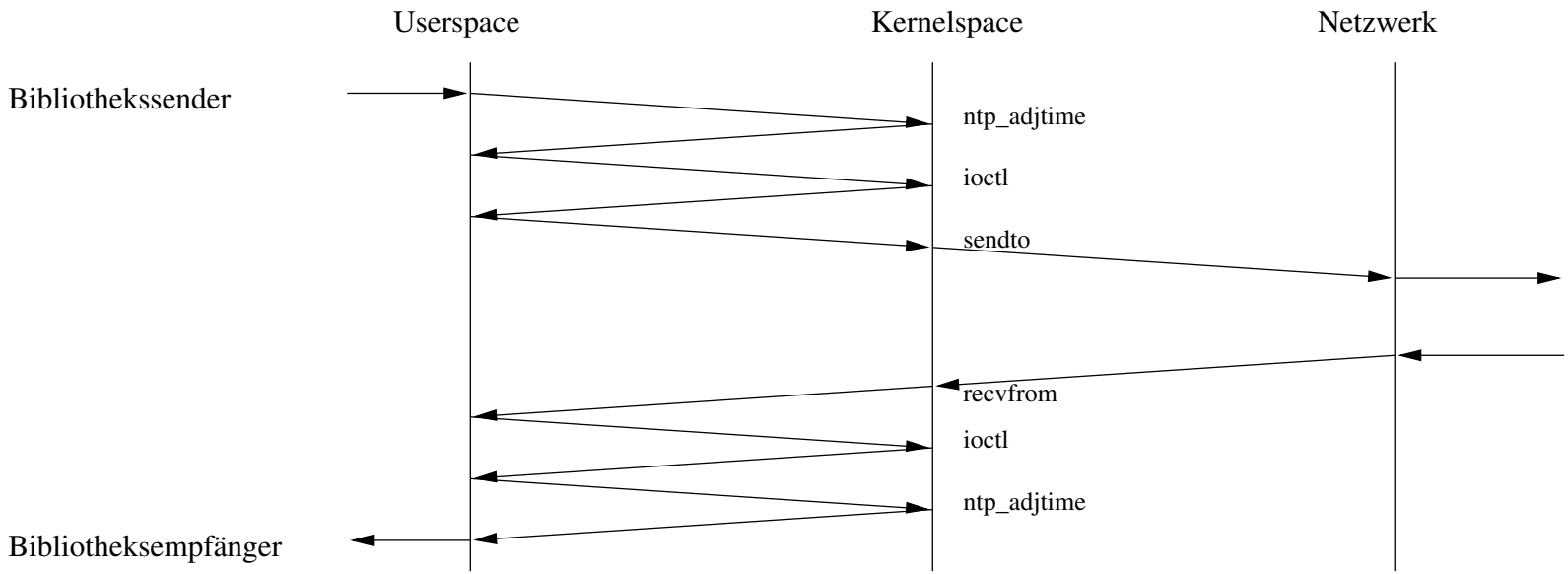
Bibliotheksbasierte Messung: Vorteile

- Einfache Implementierung
- Portabel auf andere Betriebssysteme

Bibliotheksbasierte Messung: Nachteile

- Paketgenerierung mit Zeitstempel im Userspace
- Viel (unnötiger) Programmcode in zeitkritischen Abschnitten
- Bibliotheken nicht auf zeitkritische Anwendungen ausgelegt
- Sendezeit muss bestimmt werden, bevor das Paket in der Bibliothek gebaut und gesendet wird
- Empfangszeit kann erst bestimmt werden, wenn Paket die PCAP-Bibliothek durchlaufen hat

Bibliotheksbasierte Messung: Ablauf der Messung



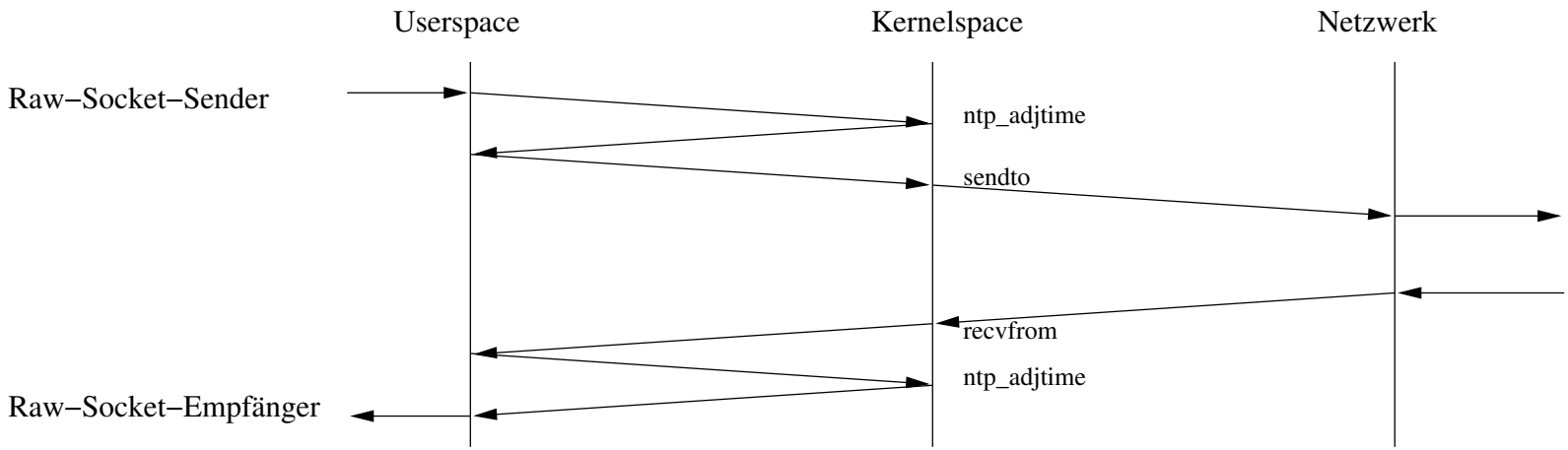
Raw-Socket-basierte Messung: Vorteile

- Empfangszeit kann bestimmt werden, sobald Paket den Userspace erreicht
- Sendezeit kann direkt vor Paketversand ermittelt werden
- 2 kritische Kontextwechsel weniger gegenüber der bibliotheksbasierten Implementierung
- Bis zu einem gewissen Grade portabel

Raw-Socket-basierte Messung: Nachteile

- Paketgenerierung mit Zeitstempel im Userspace
- Transfer des ausgefüllten Pakets vom Userspace in den Kernel

Raw-Socket-basierte Messung: Ablauf der Messung



Kernelbasierte Messung: Vorteile

- Zeitkritische Aufgaben können im Kernel durchgeführt werden
- Sehr einfache Ansteuerung des Kernel-Moduls aus dem Userspace
- Kontextwechsel bei Transfer des Pakets zwischen User- und Kernelspace wirkt sich nicht auf die Messgenauigkeit aus
- Sobald Nanosekundengenauigkeit im Kernel verfügbar, kann diese genutzt werden

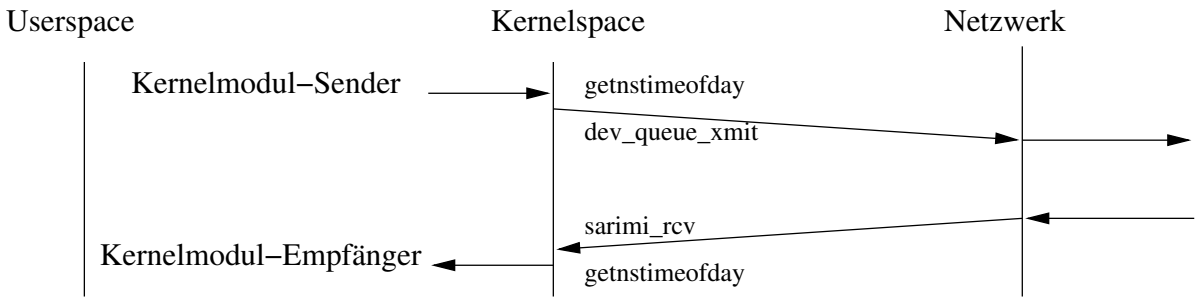
Kernelbasierte Messung: Nachteile

- Änderungen am Linux-Kernel können evtl. Änderungen des Mess-Moduls bedingen
- Nicht portabel

Kernelbasierte Messung: Funktionsweise

- Implementierung als neue Ethernet-Adressfamilie
- Senden:
  - Übergabe eines teilweise ausgefüllten (Sequenznummer) Mess-Pakets und Ziel-MAC an den Kernel
  - Ausfüllen der Sendezeit/-parameter im Kernel direkt vor dem Senden (optional bei gesperrten Interrupts)
- Empfangen:
  - Ausfüllen der Empfangszeit/-parameter im Kernel bei gesperrten Interrupts direkt nach dem Empfang
  - Übergabe des vollständig ausgefüllten Mess-Pakets an den Userspace

Kernelbasierte Messung: Ablauf der Messung

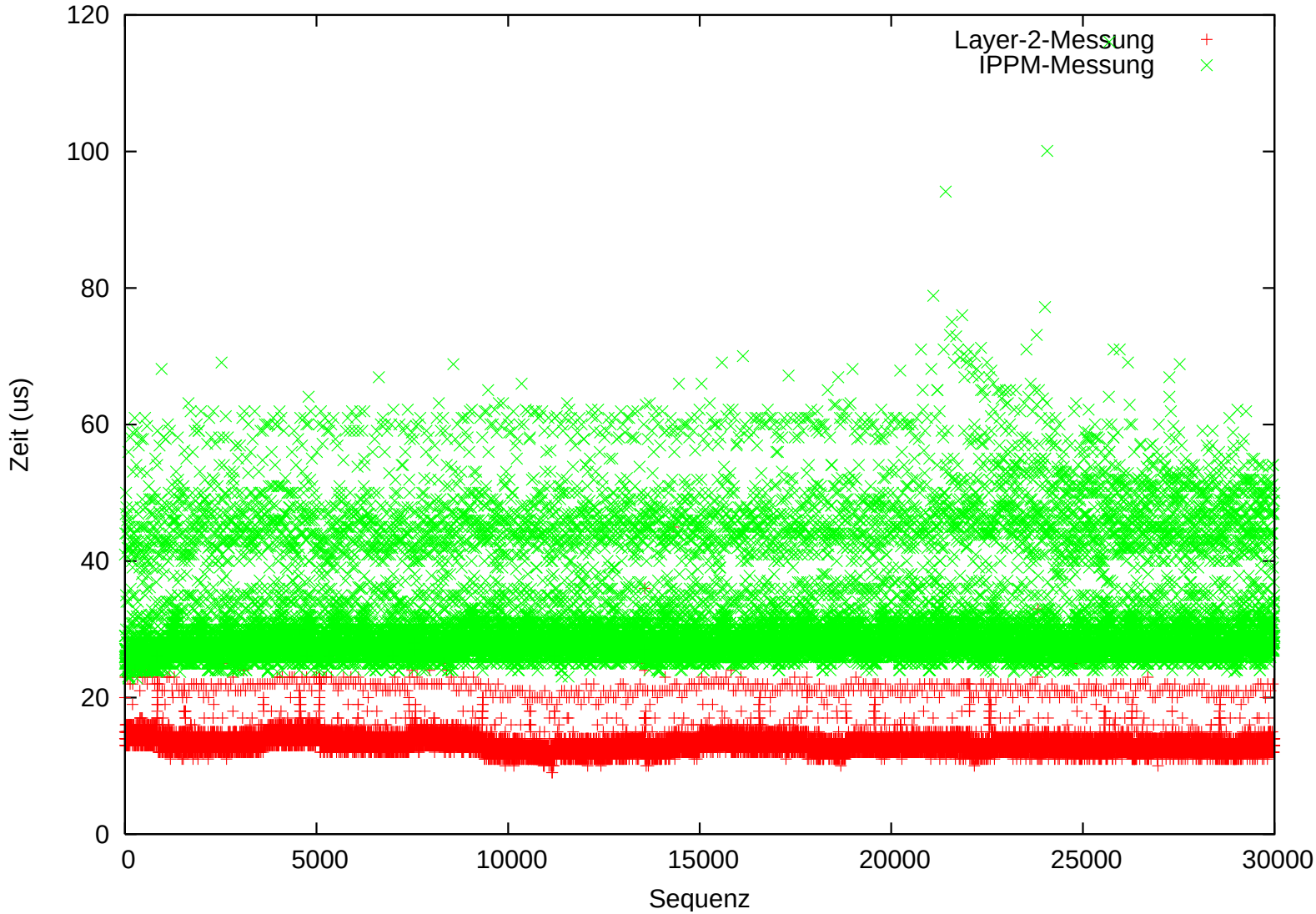


Vergleich von Layer-2- mit IPPM

Layer-2 Ergebnisse

- haben schärfer begrenzten Hauptbereich
- streuen weniger
- haben kleinere Ausreißer
- liegen auf der gleichen Messstrecke deutlich unter den IPPM-Ergebnissen

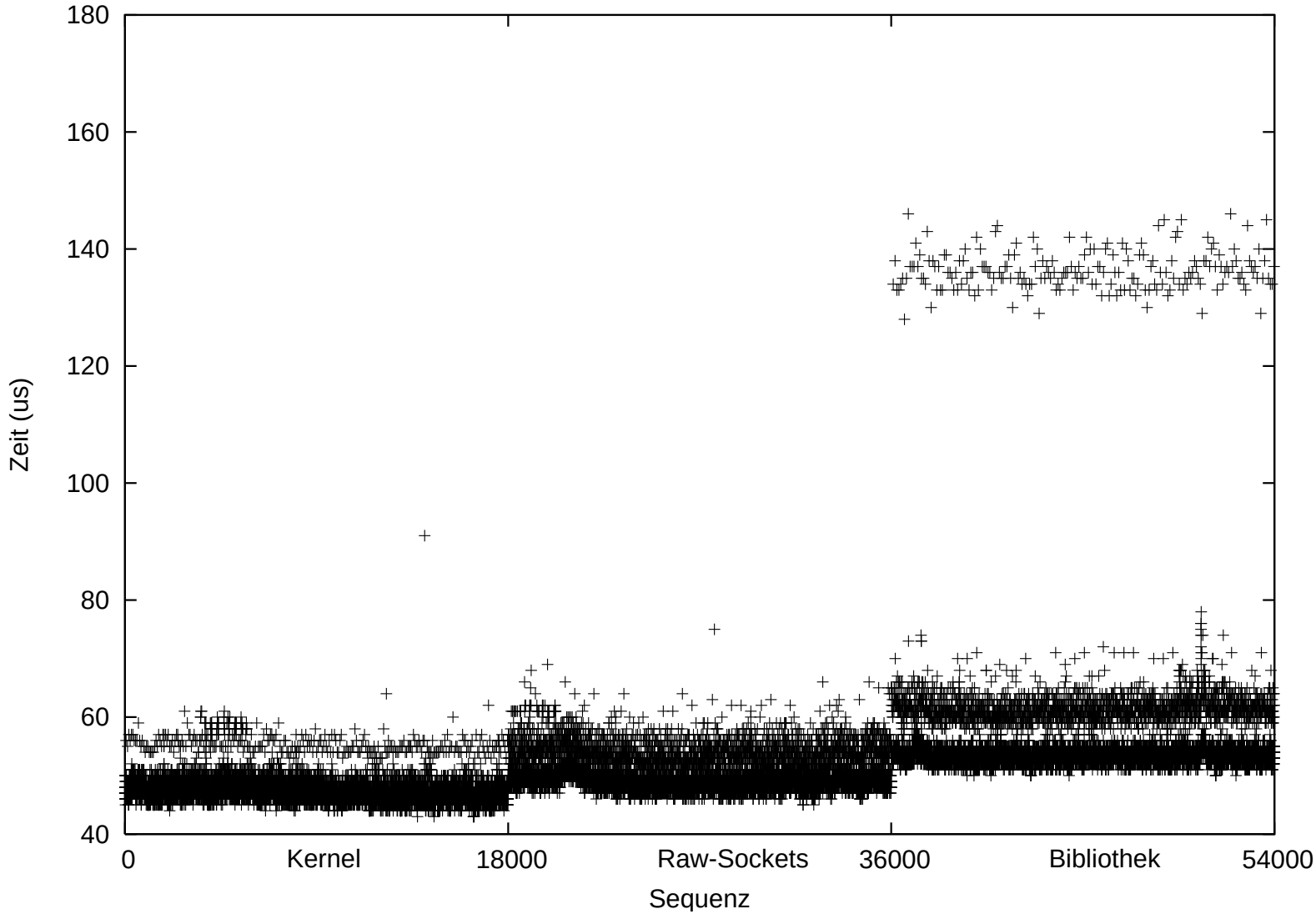
Typ der Messung	Hauptband	Streuung	maximale Ausreißer
Layer-2	$11\mu s - 14\mu s$	$11\mu s - 25\mu s$	$40\mu s$
IPPM	$24\mu s - 30\mu s$	$24\mu s - 62\mu s$	$120\mu s$


Layer-2-Messung
IPPM-Messung
Zeit (us)
Sequenz
0
20
40
60
80
100
120
0
5000
10000
15000
20000
25000
30000

Vergleich der Layer-2-Implementierungen

- Kernelmodul bietet beste Genauigkeit
- Bibliotheksimplementierung hat größte Streuung und ein „Ausreißerband" um 140μs

Implementierung	Hauptband	Streuung	maximale Ausreißer
Kernelmodul	45μs – 50μs	45μs – 61μs	90μs
Raw-Socket	47μs – 53μs	46μs – 63μs	70μs
Bibliothek	51μs – 55μs	50μs – 69μs	131μs – 146μs


180
160
140
120
100
80
60
40
Zeit (us)
0
Kernel
18000
Raw-Sockets
36000
Bibliothek
54000
Sequenz

Weitere Ergebnisse

- Bestimmung der
  - im Netzwerkstack auftretenden Verzögerungen
  - Länge eines Kontextwechsels
  - Eignung spezieller Hardware für gewisse Aufgaben

Im Netzwerkstack verbrachte Zeit

Prozessor	Kernelmodul		Raw-Socket	libnet/libpcap
	NIC-Modul	NIC-Userspace	NIC-Userspace	NIC-Userspace
Freescale 7447A 1,33GHz (ppc)	$1\mu s - 3\mu s$	$6\mu s - 20\mu s$	$6\mu s - 17\mu s$	$7\mu s - 24\mu s$
Intel CoreDuo 1,66GHz (x86)	$1\mu s - 3\mu s$	$14\mu s - 30\mu s$	$12\mu s - 27\mu s$	$16\mu s - 33\mu s$
Intel P4 1,6GHz (x86)	$5\mu s - 31\mu s$	$17\mu s - 104\mu s$	$16\mu s - 98\mu s$	$20\mu s - 106\mu s$
AMD Geode 266MHz (x86)	$25\mu s - 51\mu s$	$115\mu s - 219\mu s$	$106\mu s - 216\mu s$	$113\mu s - 810\mu s$

NIC = Zeitstempel im Netzwerkkartentreiber

- Kontextwechsel haben Einfluß auf die Messgenauigkeit
- Codegröße hat Einfluß auf die Messgenauigkeit (Cache-misses)

Fazit

- Genaue Messung auf Ethernet Layer-2 möglich
- Verfügbar auf verbreiteter Hard-/Softwarekombination (x86/Linux 2.6)
- 3 Implementierungen mit unterschiedlicher Genauigkeit
- Architekturspezifische Eigenheiten haben Einfluss auf das Ergebnis

Vielen Dank für Ihre Aufmerksamkeit!