



Fachgebiet 3-3 – Alternative Transportprotokolle

10 Gbit/s-Performance-Messungen

im VIOLA-Netz

Autor

Franz Petri (ZAM, Forschungszentrum Jülich)

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01AK800B gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt bei den Autoren.

Inhaltsverzeichnis

1 Einleitung.....	4
2 Aufbau der Messumgebung.....	5
2.1 Streckenführung im VIOLA-Netz.....	5
2.2 Konfiguration der Endsysteme.....	6
2.2.1 Hardware.....	6
2.2.2 Anpassung von Kernel- und TCP-Parametern.....	6
2.2.3 UDT-Version.....	6
3 Messergebnisse.....	7
3.1 tcppp.....	7
3.2 udtp.....	8
3.3 vtspp.....	9
3.4 visitpp.....	10
3.5 iperf.....	11
3.6 gridftp.....	12
4 Zusammenfassung und Ausblick.....	12
4.1 Tabellarischer Überblick.....	12
4.2 Paketverlust.....	14
4.3 TCP vs. UDT.....	14
4.4 Neterion vs. Myricom.....	14
4.5 iperf -P vs. gridftp.....	14
Literatur.....	15

1 Einleitung

Der Fachbereich 3-3 des D-Grid-Projektes ist dafür verantwortlich, existierende alternative Transportprotokolle für den Einsatz in D-Grid-Anwendungen hinsichtlich Performance und Stabilität zu bewerten und Empfehlungen für die Entwickler in den D-Grid-Communities zu formulieren. Um eine hinreichend praxisnahe Evaluation von Transportprotokollen jenseits von Bandbreiten von 1 Gbit/s durchführen zu können, werden darüber hinaus verschiedene 10-Gbit/s-NICs in die Test-szenarien mit einbezogen. Ziel hierbei ist, durch geeignete Wahl von Parametern in den Betriebssystemen oder in den Applikationen einen hohen Datendurchsatz auf der D-Grid-Infrastruktur zu erreichen und so eine optimale Nutzung der verfügbaren Ressourcen zu gewährleisten. Die Notwendigkeit für diese Untersuchungen wurde in vorausgegangenen Reports festgestellt (vgl. [SG06], [PE06]).

Für diesen Report wurde die Netz-Infrastruktur des VIOLA-Projektes herangezogen; VIOLA bietet als DFN-Testbed für neuen Netztechniken und neuen Formen der Netzintelligenz eine Infrastruktur mit hoher Bandbreite, auf der sich 10 Gbit/s-NICs und alternative Transportprotokolle unter WAN-Bedingungen jenseits der 1 Gbit/s testen lassen (siehe [Viola]).

Es wurden Messungen zu Datendurchsatz und Nachrichtenlaufzeit durchgeführt. Hierfür kam das Benchmarking-Tool Iperf, der GridFTP-Server und -Client des Globus Toolkit sowie die Pingpong-Tools der VISIT-Suite zum Einsatz (siehe auch [Iperf], [Gtk], [Visit]).

Die für die Teststellung eingerichteten Netzwerkschleifen im VIOLA-Netz wurden überdies für eine Demonstration der vom FZJ entwickelten und um das alternative Transportprotokoll UDT erweiterten VIOLA-Anwendung zur Kollaborativen Datenvisualisierung [Kodavis] auf dem D-Grid-All-Hands-Meeting (23.-24. November 2006) im DESY in Hamburg genutzt.

2 Aufbau der Messumgebung

Wie in vorausgegangen Berichten festgestellt wurde, werden die Schwächen der Standard-TCP-Implementierungen insbesondere auf breitbandigen Netzwerkverbindungen mit Paketverlust in Verbindung mit hoher Latenz offensichtlich: aufgrund seines AIMD-Verhaltens reduziert TCP-RENO bei Paketverlust seine Übertragungsrate drastisch und erhöht diese dann nur additiv und indirekt proportional zur RTT. Konkret bedeutet das beispielsweise, dass bei einer Verzehnfachung der Bandbreite (z.B. 1 Gbit/s -> 10 Gbit/s) und der RTT (z.B. 1 ms LAN -> 10 ms WAN) eine Verhundertfachung der Zeitspanne eintritt, die Standard-TCP benötigt, um nach einem einmaligen Paketverlust die ursprüngliche Durchsatzrate wieder herzustellen (vgl. [SG06]).

In [PE06] wurde weiterhin festgestellt, dass auf dem verwendeten Opteron-Testsystem Paketverlust und Latenz ab einer Bandbreite größer als 1 Gbit/s mit Hilfe von Emulatoren wie [Netem] oder [Nistnet] lokal nicht oder nur unter Verfälschung der Testergebnisse erzeugt werden können.

Aus diesem Grund wurde mit Hilfe einer Schleife im VIOLA-Testbed die Teststrecke verlängert, um so "echte" (im Gegensatz zu emulierter) Laufzeiterhöhung der Datenpakete zu erhalten.

2.1 Streckenführung im VIOLA-Netz

Für die Untersuchungen wurden zwei Testserver mit eigenem privaten Subnetz über eine Schleife im VIOLA-Netz miteinander verbunden. Die Daten wurden hierbei über zwei Gateways (Riverstone 15008 und Alcatel 7750) ins VIOLA-WDM-Equipment eingespeist (Vgl. Abbildung 1: Streckenführung VIOLA-Netz).

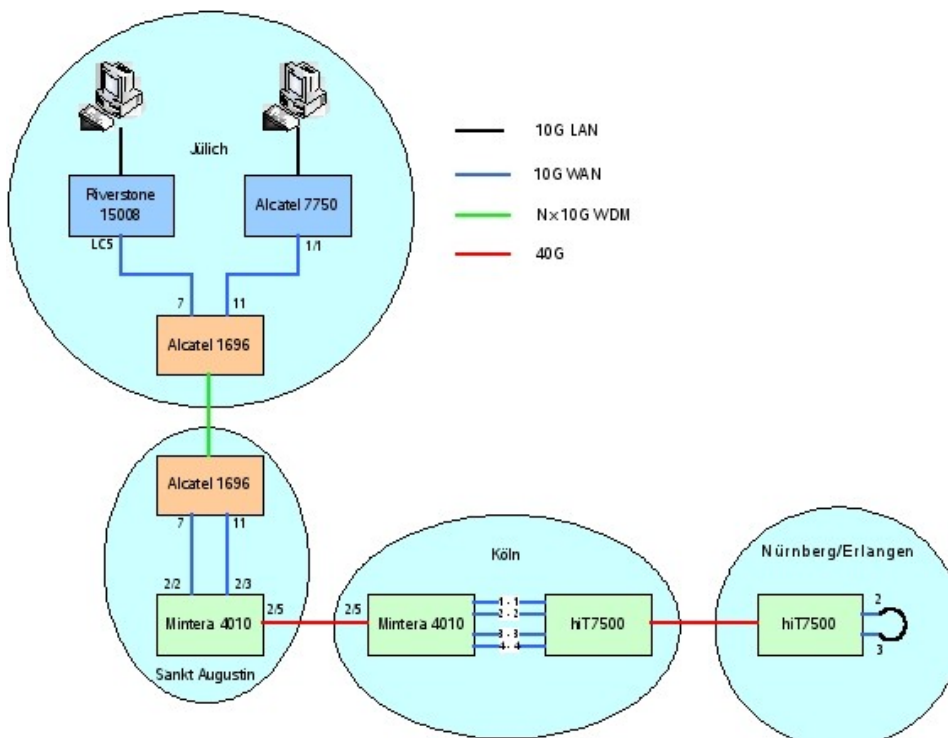


Abbildung 1: Streckenführung VIOLA-Netz

Durch die Streckenführung über Nürnberg/Erlangen wurde eine durchschnittliche Round Trip Time von 20 ms erzeugt.

2.2 Konfiguration der Endsysteme

2.2.1 Hardware

Auf den Endsystemen kamen pro Maschine folgende Komponenten zum Einsatz:

- Tyan Thunder K8WE (S2895), 4x 512 MB DDR400
- Chipsatz: AMD 8131, Nvidia Professional 2200&2050
- CPU: 2x AMD Opteron Dual Core 265 (1800 Mhz)

Der Speicherzugriff war als NUMA konfiguriert. Das Node-Interleaving wurde per BIOS deaktiviert.

Als Betriebssystem war OpenSuse 10.1 installiert, Kernelversion 2.6.16.21-0.25-smp.

Folgende 10 GbE-NICs wurden eingesetzt:

- 2x Neterion Xframe E
 - PCI-Express 4x
 - Treiber: s2io, Version 2.0.9.5:
`modprobe s2io lro=1`
- 2x Myricom 10G-PCIE-8A-R
 - PCI-Express 8x
 - Treiber myri10ge, Version 1.1.0:
`modprobe myri10ge; ethtool -K <ethxxx> tso off`

2.2.2 Anpassung von Kernel- und TCP-Parametern

Die Kernelparameter wurden per sysctl-Skript wie folgt gesetzt:

```
net.core.rmem_max = 33554432
net.core.wmem_max = 33554432
net.ipv4.tcp_rmem = 4096 8388608 33554432
net.ipv4.tcp_wmem = 4096 8388608 33554432
net.core.netdev_max_backlog = 100000
net.ipv4.tcp_timestamps = 1
net.ipv4.tcp_sack = 0
net.ipv4.tcp_no_metrics_save = 1
```

Die Transmit-Queue wurde folgendermaßen angepasst:

```
ifconfig <ethxxx> txqueuelen 100000
```

Für weiter Informationen zum Thema TCP- und Socket-Tuning siehe [MR06].

2.2.3 UDT-Version

Es kam die UDT-Library in der Version 3.1 zum Einsatz (siehe [Udt]). Die Library wurde übersetzt mit:

```
make -e arch=AMD64
```

Ansonsten wurden die UDT-Standard-Einstellungen übernommen.

3 Messergebnisse

Es wurden Messungen mit den Pingpong-Tools der VISIT-Suite gemäß [PE06] (vgl. Abbildung 2: Schematische Darstellung eines Datenpingongs mit Zeitmessung), sowie Durchsatztests mit iperf (Version 2.0.2) und Grid-FTP (globus_gridftp_server: 2.3, globus-url-copy: 3.20) durchgeführt.

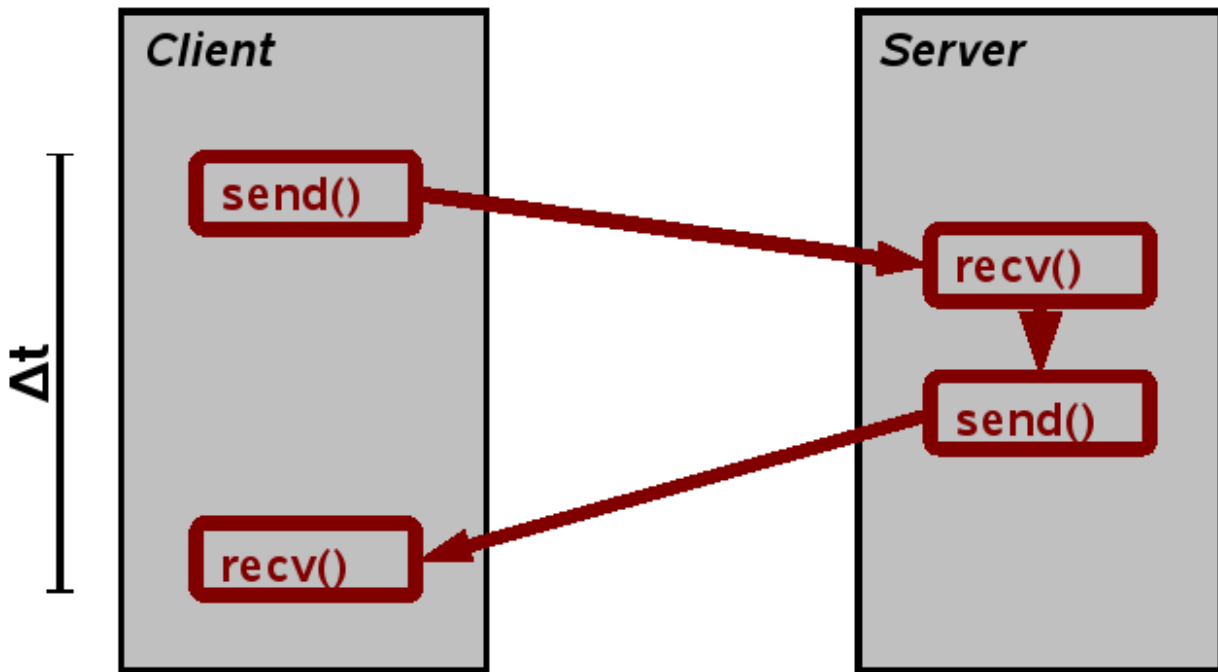


Abbildung 2: Schematische Darstellung eines Datenpingongs mit Zeitmessung

3.1 tcppp

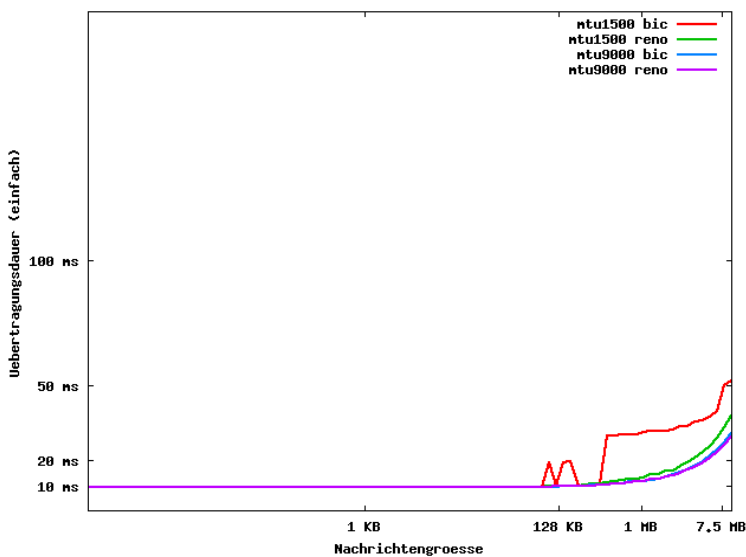


Abbildung 3: TCP-Pingongs, Laufzeit in Abhängigkeit von Nachrichtengröße

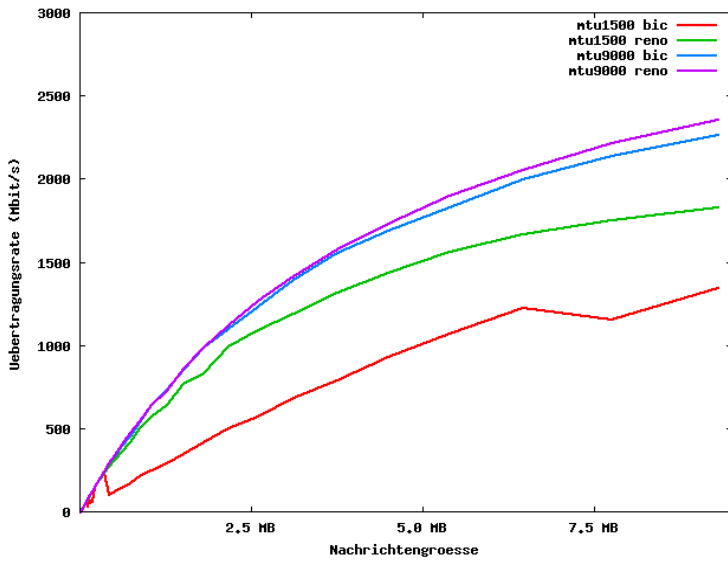


Abbildung 4: TCP-Pingpongs, Übertragungsrate in Abhängigkeit von Nachrichtengröße

3.2 udtp

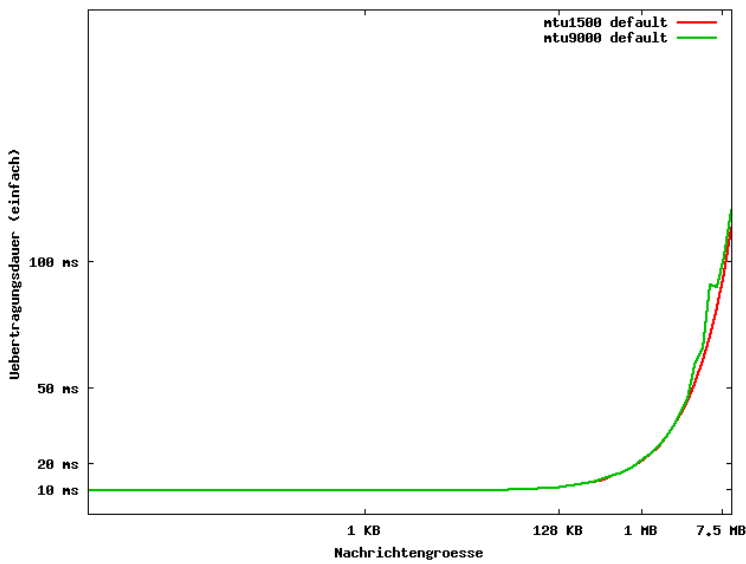


Abbildung 5: UDT-Pingpongs, Laufzeit in Abhängigkeit von Nachrichtengröße

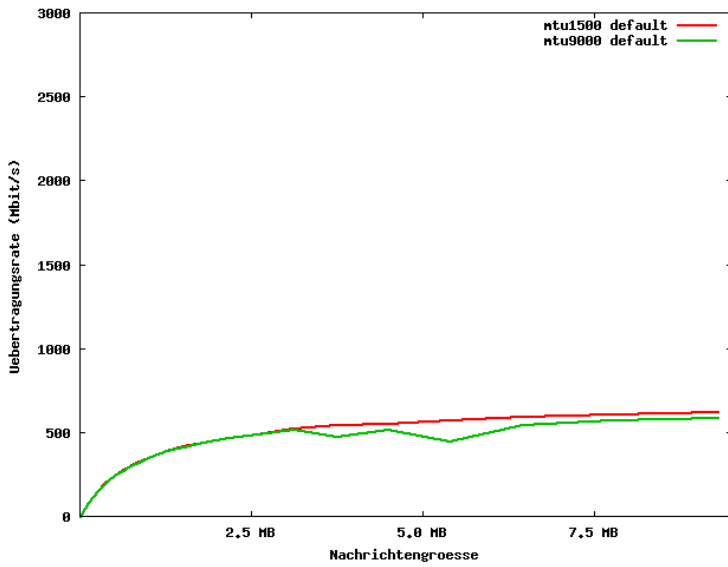


Abbildung 6: UDT-Pingpongs, Übertragungsrate in Abhängigkeit von Nachrichtengröße

3.3 vtsp

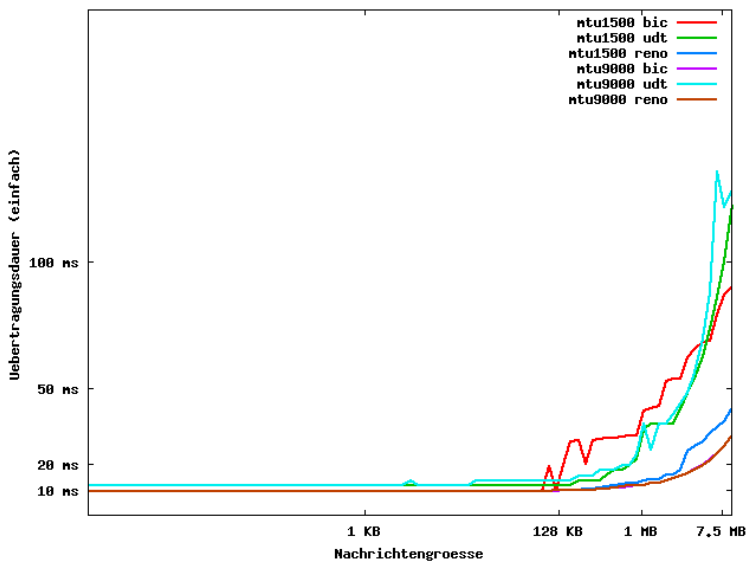


Abbildung 7: VTS-Pingpongs, Laufzeit in Abhängigkeit von Nachrichtengröße

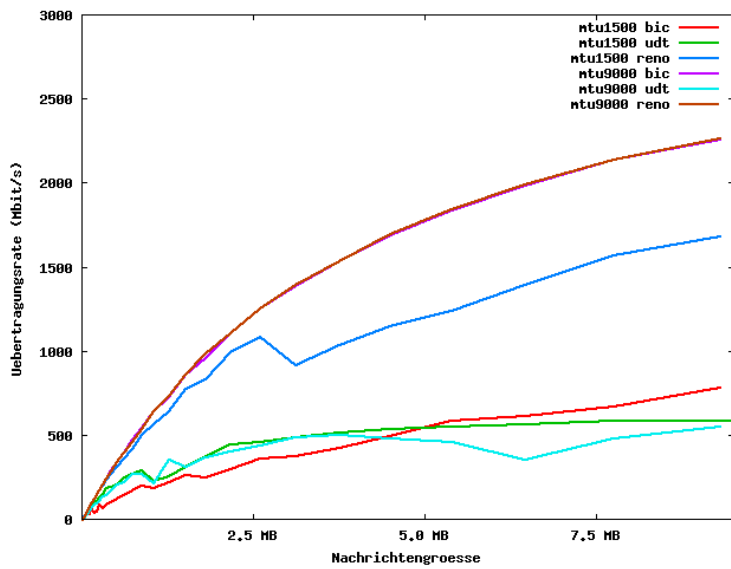


Abbildung 8: VTS-Pingpongs, Übertragungsrate in Abhängigkeit von Nachrichtengröße

3.4 visitpp

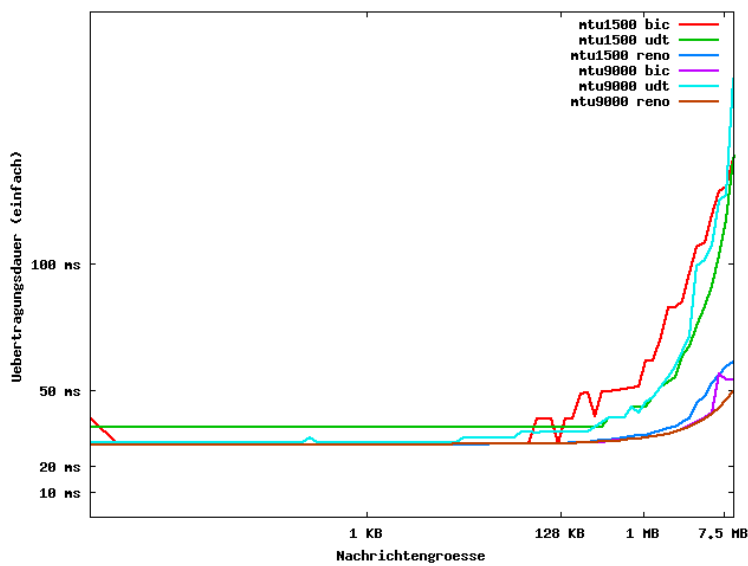


Abbildung 9: VISIT-Pingpongs, Laufzeit in Abhängigkeit von Nachrichtengröße

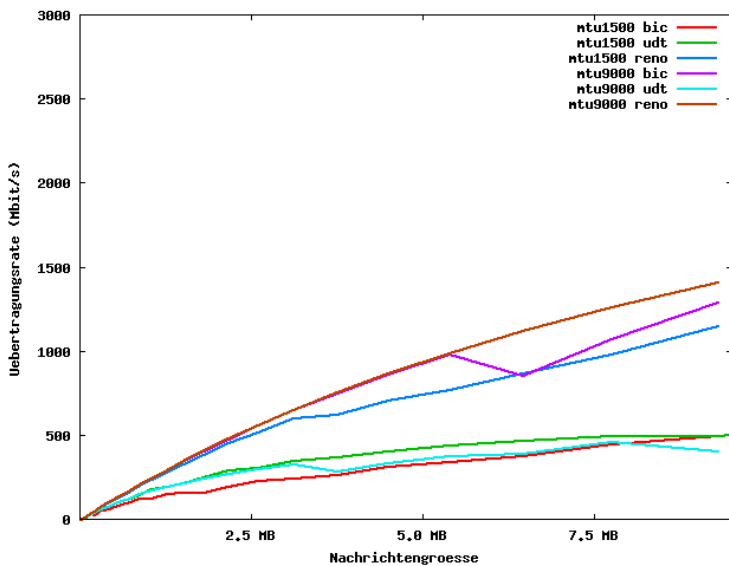


Abbildung 10: VISIT-Pingpongs, Übertragungsrate in Abhängigkeit von Nachrichtengröße

3.5 iperf

Die Iperf-Tests wurden mit TCP RENO durchgeführt. Die Werte sind über 60 Sekunden Messdauer gemittelt.

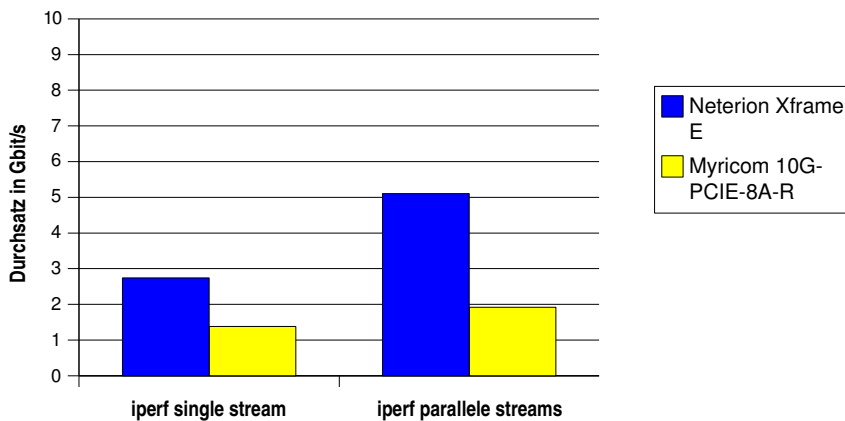


Abbildung 11: Vergleich Neterion-Myricom, iperf, mtu 1500

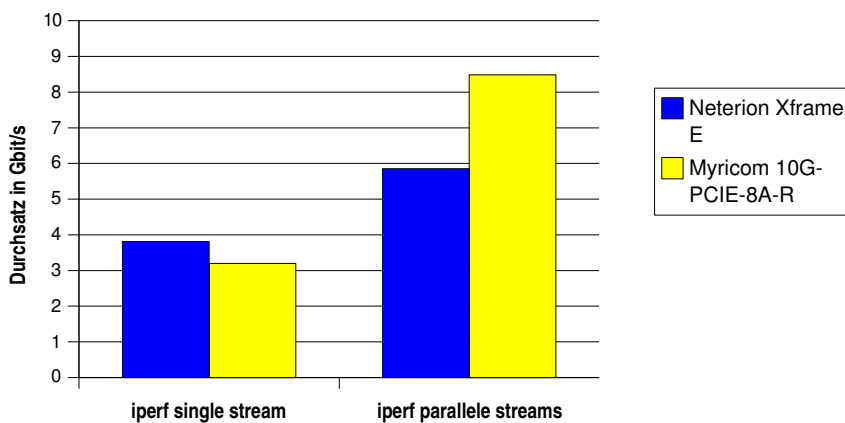


Abbildung 12: Vergleich Neterion-Myricom, iperf, mtu 9000

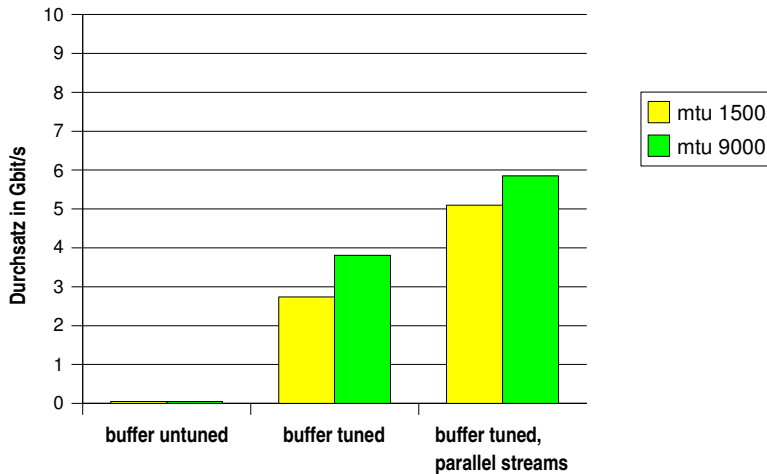


Abbildung 13: Auswirkungen von Buffer Tuning und parallelen Streams

3.6 gridftp

Die gridftp-Tests wurden mit TCP RENO durchgeführt. Es wurde von /dev/zero nach /dev/null übertragen.

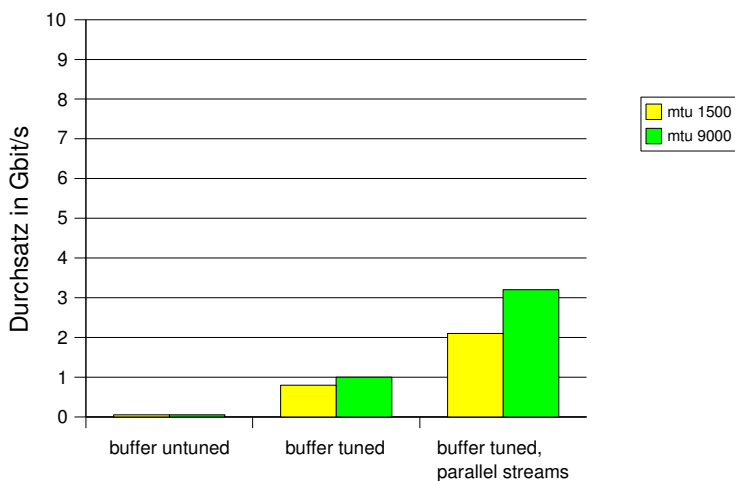


Abbildung 14: Gridftp, Single- und Parallele Streams

4 Zusammenfassung und Ausblick

4.1 Tabellarischer Überblick

Tabelle 1 fasst die Messergebnisse der Pingpong-Benchmarks zusammen, wobei gilt:

- $\text{Übertragungszeit}[1B] = \Delta t/2$ (gemessen für eine Nachricht mit der Größe 1 Byte)

- $\text{Übertragungsrate}[\max] = \text{Nachrichtengröße}/\Delta t$ (gemessen für eine Nachricht mit der Größe 9288586 Byte)

Zur Bedeutung von Δt , siehe Abbildung 2: Schematische Darstellung eines Datenpingpongs mit Zeitmessung.

Benchmark, MTU, Protokoll	Übertragungszeit[1B] [ms]	Übertragungsrate[max] [Mbit/s]
tcppp, mtu1500, bic	9694	1346
tcppp, mtu1500, reno	9697	1836
tcppp, mtu9000, bic	9697	2269
tcppp, mtu9000, reno	9702	2362
udtpp, mtu1500, default	9727	623
udtpp, mtu9000, default	9730	587
vtsp, mtu1500, bic	9704	783
vtsp, mtu1500, udt	11999	358
vtsp, mtu1500, reno	9704	1686
vtsp, mtu9000, bic	9700	2260
vtsp, mtu9000, udt	11996	551
vtsp, mtu9000, reno	9700	2270
visitpp, mtu1500, bic	39425	500
visitpp, mtu1500, udt	35968	657
visitpp, mtu1500, reno	29189	1150
visitpp, mtu9000, bic	29175	1293
visitpp, mtu9000, udt	29971	407
visitpp, mtu9000, reno	29175	1414

Tabelle 1: Nachrichtenlaufzeiten und Übertragungsraten

4.2 Paketverlust

Zu Beginn der Tests kam es auf der Teststrecke zu vereinzelt Paketverlusten. Im Schnitt ging aufgrund von Bitfehlern jedes vierzigmillionste Paket verloren. Diese auf den ersten Blick gering erscheinende Verlustrate in der Größenordnung von $<10^7$ führte zu deutlichen Einbrüchen in der durchschnittlichen Übertragungsrate und reduzierte diese auf weniger als 1 Gbit/s. Der Paketverlust konnte durch diverse Anpassungen an den optischen Komponenten auf der Strecke eliminiert werden, was allerdings einen erheblichen Mehraufwand bedeutete. Diese Beobachtung unterstreicht die Notwendigkeit von aggressiveren Congestion-Controll-Mechanismen in Reaktion auf Paketverlust, insbesondere auf dedizierten Verbindungen, will man die theoretischen Möglichkeiten der zur Verfügung stehenden Hardware ausnutzen.

4.3 TCP vs. UDT

Im direkten Vergleich (siehe Abbildung 2: TCP-Pingpongs, Laufzeit in Abhängigkeit von Nachrichtengröße und Abbildung 4: UDT-Pingpongs, Laufzeit in Abhängigkeit von Nachrichtengröße) wird deutlich, dass die in [PE06] festgestellte höhere Latenz von UDT gegenüber TCP bei höherer RTT an Bedeutung verliert, insbesondere für Nachrichtengrößen $< 128\text{KByte}$.

Desweiteren fällt auf, dass mit der UDT-Library in den vorliegenden Tests keine Übertragungsraten über 1 Gbit/s erreicht werden konnten. Auch von Jumbo-Frames konnte UDT in diesem Zusammenhang nicht profitieren. Ob diese Schwäche konzeptioneller Natur ist, also der Implementierung von Congestion Control und Zuverlässigkeit bei UDT im User- anstatt im Kernspace (wie bei TCP) geschuldet ist und inwieweit sich durch Code-Optimierung noch Leistungsreserven erschließen lassen, ist Gegenstand weiterer Untersuchungen.

4.4 Neterion vs. Myricom

Die NICs von Myricom zeigten unter Default-Einstellungen deutlich schlechtere Performance-Werte als die Neterion-Adapter (siehe Abbildung 10: Vergleich Neterion-Myricom, iperf, mtu 1500 und Abbildung 11: Vergleich Neterion-Myricom, iperf, mtu 9000). Insbesondere das standardmäßig aktivierte TCP-Segmentation Offloading (TSO) drückte die durchschnittliche Durchsatzrate der Myricom-NICs auf unter 1 Mbit/s (!).

Die Vorteile der Busanbindung der Myricom-Karten (PCI-E 8x) bzw. die Limitierung der Neterion Xframe E durch die PCI-E 4x-Anbindung machen sich erst durch den Einsatz von Jumbo-Frames und parallelen Streams (iperf -P) bemerkbar. Ob zukünftige Treiberversionen bzw. Veränderungen an den Standardeinstellungen der Treiber im Zusammenspiel mit neueren Kernelversionen eine Leistungssteigerung möglich machen, soll in weiteren Untersuchungen geklärt werden.

4.5 iperf -P vs. gridftp

Der Einsatz von parallelen TCP-Streams führt zu einer deutlichen Steigerung der Gesamt-Durchsatzrate (siehe Abbildung 12: Auswirkungen von Buffer Tuning und parallelen Streams und Abbildung 13: Gridftp, Single- und Parallele Streams). Bei den Tests fiel auf, dass bei Bandbreiten über 1Gbit/s, also in Bereichen, in denen die CPU-Leistung als begrenzender Faktor an Bedeutung gewinnt, eine Parallelisierung der Datenübertragung durch den Einsatz von parallelen Streams nicht nur Vorteile hinsichtlich des Ausgleichs bekannter Schwächen der Standard-TCP-Implementierung, also größere Robustheit gegenüber Paketverlusten, bringt, sondern dass auch die vorhandene Prozessorleistung effizienter ausgenutzt werden kann. Insbesondere Iperf konnte in diesem Zusammenhang durch den Einsatz von mehreren parallelen Send- und Receive-Threads von der vorhandenen Multicore-Architektur profitieren. Dies legt weitere Untersuchungen nahe, in denen geklärt werden soll, inwieweit durch parallele TCP-Streams neben den bekannten Benchmark- und Dateitransfer-Werkzeugen wie iperf oder gridftp auch allgemeinere TCP-Datentransfers auf

Übertragungsraten von mehr als 1 Gbit/s beschleunigt werden können.

Literatur

- [SG06] H. Schwier, C. Grimm. *Analyse von TCP-Varianten*. <http://www.d-grid.de>.
https://www.d-grid.de/fileadmin/user_upload/documents/DGI-FG3-3/FG3-3_Analyse-TCP.pdf, 2006.
- [PE06] F. Petri, Th. Eickermann. *Alternative Transportprotokolle im Einsatz: Laufzeittests mit dem Visualisation Interface Toolkit VISIT*. <http://www.d-grid.de>.
http://dgi.d-grid.de/fileadmin/user_upload/documents/DGI-FG3-3/FG3-3_Laufzeittests_VISIT_v1_0.pdf, 2006.
- [Viola] *Viola - optische Netze: Projekt VIOLA*. <http://www.viola-testbed.de>, (2006).
- [Iperf] *NLANR/DAST : Iperf 2.0.2 - The TCP/UDP Bandwidth Measurement Tool*.
<http://dast.nlanr.net/Projects/Iperf/>, (2005).
- [Gtk] *Globus Toolkit 4.0.3*. <http://www.globus.org/toolkit/>, (2006).
- [Visit] *VISIT - a Visualization Toolkit*. <http://www.fz-juelich.de/zam/visit/>, (2006).
- [Kodavis] *KoDaVis: Collaborative Visualisation of Huge Atmospheric Data in a Heterogeneous Environment*. <http://www.viola-testbed.de/content/index.php?id=kodavis>, (2006).
- [Netem] *Netem - LinuxNet*. <http://linux-net.osdl.org/index.php/Netem>, (2006).
- [Nistnet] *NIST Net Home Page*. <http://www-x.antd.nist.gov/nistnet/>, (2005).
- [MR06] *Enabling High Performance Data Transfers [PSC]*.
<http://www.psc.edu/networking/projects/tcptune/>, (2006).
- [Udt] *UDT: UDP-based Data Transfer Protocol*. <http://udt.sourceforge.net/>, (2006).