Architektur, Implementierung und Leistungsbewertung eines parallelen Transportsubsystems für zellenbasierte Hochgeschwindigkeitsnetze

Torsten Braun

Institut für Telematik, Universität Karlsruhe, Zirkel 2, 76128 Karlsruhe

Telefon: 0721/608-3982, Fax: 0721/388097

E-mail: braun@telematik.informatik.uni-karlsruhe.de

Zusammenfassung

Moderne Anwendungen und Netzwerktechnologien stellen zunehmend höhere Anforderungen an geeignete Transportsysteme. Das Transportsubsystem PATROCLOS wurde entworfen, um leistungsfähige parallele Implementierungen auf Multiprozessorarchitekturen mit dedizierten Hardwarekomponenten für zeitkritische Funktionen zu unterstützen. Weitere Entwurfsaspekte bestanden in der Bereitstellung von im Umfeld von Hochgeschwindigkeitsnetzen notwendigen Protokollfunktionen und in der Unterstützung einer flexiblen Kommunikationsarchitektur durch eine modulare, in funktionale Einheiten unterteilte Protokollarchitektur. Die Leistungsbewertung einer Implementierung auf einer hybriden Multiprozessorarchitektur ergaben dabei deutlich bessere Durchsatzergebnisse und Speed-up-Werte verglichen mit anderen parallelen Transportsubsystem-Implementierungen auf ähnlichen Implementierungsplattformen.

1. Einleitung

Derzeit entwickeln sich neue Anwendungen mit hohen quantitativen und qualitativen Anforderungen an Kommunikationssysteme und Netzwerke in Form von hohem Durchsatz, geringen Verzögerungen und niedrigen Fehlerraten. Enorme Fortschritte in der optischen Übertragungstechnik ermöglichen die notwendigen Datenraten, welche sich bei modernen Hochgeschwindigkeitsnetzen bereits im Gigabit/s-Bereich befinden. Allerdings sind die als Bindeglied zwischen Rechnersystem und Netzwerk agierenden Kommunikationssysteme derzeit nicht in der Lage, die vorhandene Leistungsfähigkeit moderner Hochgeschwindigkeitsnetze den Anwendungen direkt verfügbar zu machen. Einerseits können die zur Unterstützung einer sicheren Rechnerkommunikation notwendigen Kommunikationsprotokolle nicht schnell genug verarbeitet werden, andererseits stellt die Kopplung der Kommunikationssysteme an Netzwerke und Rechnersysteme einen gravierenden Engpaß dar. Insgesamt lassen sich die Leistungsschwächen auf so unterschiedliche Ursachen wie für die Hochgeschwindigkeitskommunikation

ungeeigneten Protokollentwurf, ungünstige Implementierungsplattformen und die fehlende Abstimmung der Kommunikationssysteme auf unterliegende Netzwerke zurückführen. Außerdem wird häufig die Integration der Kommunikationssysteme in die Architektur und in die Betriebssystemumgebung von Rechnersystemen erschwert.

Die bisherigen Arbeiten im Forschungsgebiet der Hochgeschwindigkeitskommunikation befaßten sich im wesentlichen mit der Beseitigung des Leistungsengpasses bei der Protokollverarbeitung. Dazu entwickelten sich zwei klassische Ansätze. Zum einen können Kommunikationsprotokolle durch verbesserte Implementierungsstrategien, die Verwendung von Spezialhardware oder durch die Unterstützung von parallelen Implementierungsarchitekturen leistungsfähiger implementiert werden. Der zweite Ansatz versucht die Protokollverarbeitung durch den Einsatz sogenannter Hochgeschwindigkeitsprotokolle zu beschleunigen. Diese Protokolle wurden auf die Anwendung in modernen Netzwerkumgebungen hin optimiert, z.B. werden aufwendige Fehlerbehandlungsfunktionen vermieden, die in modernen Netzwerken mit extrem niedrigen Fehlerraten teilweise nicht mehr erforderlich sind.

Mit dem Transportsubsystem PATROCLOS (Parallel Transport Subsystem for Cell-Based High-Speed Networks) werden die klassischen Lösungswege kombiniert, d.h. es werden verschiedene Aspekte wie die Eigenschaften zellenbasierter Netzwerke und die zur Realisierung auf parallelen Implementierungsarchitekturen geeignete Protokollarchitektur integrierend betrachtet. Es erbringt unter Berücksichtigung zukünftiger Erfordernisse hinsichtlich neuer Dienste und Dienstqualitätsmerkmale einen erweiterten Transportdienst und besitzt dabei nicht nur die Funktionalität konventioneller Transportprotokolle, sondern übernimmt auch Funktionen der Sicherungs- und Vermittlungsschicht. Ein wesentlicher Entwurfsaspekt der entwickelten Protokollarchitektur besteht in der Berücksichtigung von Eigenschaften zellenbasierter Hochgeschwindigkeitsnetze wie DQDB und Breitband-ISDN auf der Basis von ATM, die zur Unterstützung verschiedenartiger Dienste kleine Übertragungseinheiten fester Größe, sogenannte Zellen, verwenden. Die Verarbeitung dieser Zellen stellt zwar hohe Leistungsanforderungen an Kommunikationssubsysteme, erlaubt aber auch eine effizientere Übertragung kleiner Datenmengen.

2. Architektur des parallelen Transportsubsystems

Ausgehend von Erfahrungen und Ergebnissen aus der Analyse des Hochgeschwindigkeitsprotokolls XTP (Xpress Transfer Protocol) [1] sowie aus Leistungsuntersuchungen an einer parallelen XTP-Implementierung auf einer Multiprozessorarchitektur [2] wurden Konzepte für den Entwurf eines parallelen Transportsubsystems erarbeitet. Die entwickelten parallelen

Entwurfskonzepte [3] resultierten im parallelen Transportsubsystem PATROCLOS [4], welches in orthogonale, funktionale Einheiten in Form von autonomen Automaten unterteilt ist.

2.1. Parallele Protokollarchitektur

Die Dekomposition in weitgehend voneinander unabhängige und damit parallel ausführbare funktionsbasierte Einheiten unterstützt direkt effiziente Implementierungen auf parallelen Architekturen wie hybriden Multiprozessorsystemen. Das Transportsubsystem stellt insgesamt ein Automatensystem dar, das sich aus einer Menge innerhalb einer Instanz kooperierender Automaten zusammensetzt. Automaten einer Instanz tauschen zu Synchronisations- und Kooperationszwecken Nachrichten aus.

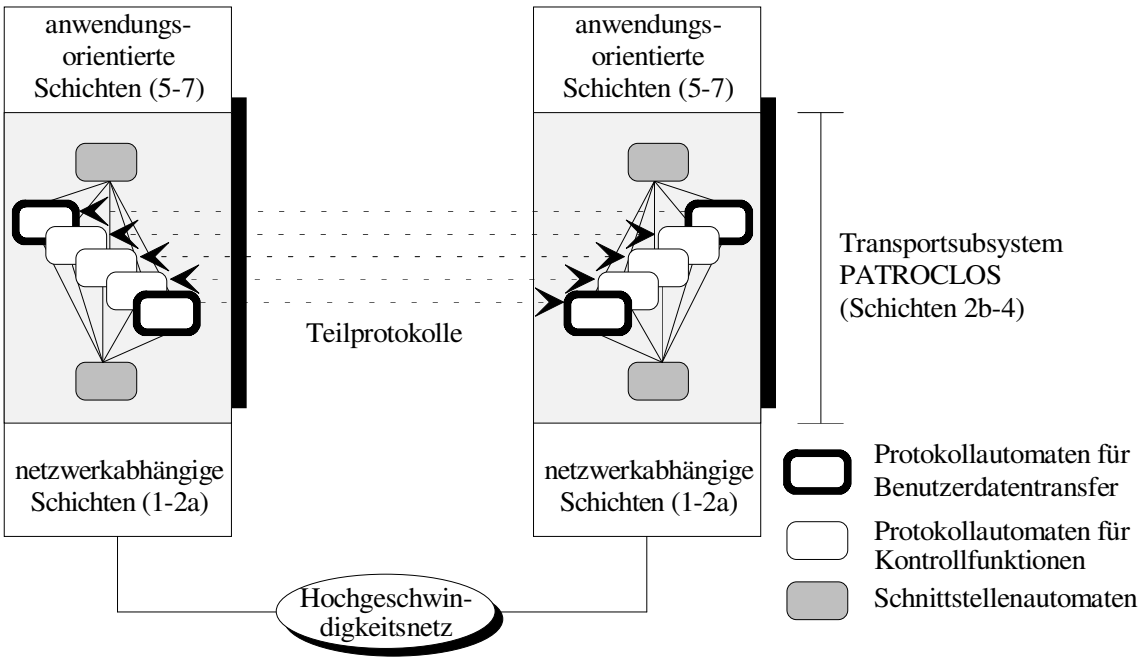

Bild 1: PATROCLOS-Architektur

Das Automatensystem unterscheidet Protokoll- und Schnittstellenautomaten (vgl. Bilder 1 und 2). Schnittstellenautomaten unterstützen Interaktionen mit dem Transportdienstbenutzer bzw. mit dem unterliegenden Netzwerk. Sie sind lediglich zur lokalen Kommunikation innerhalb eines Systems notwendig und haben im Gegensatz zu Protokollautomaten keinen Automaten in der Partnerinstanz, mit dem sie kommunizieren. Im Transportdienstschnittstellen-Automaten ist der erweiterte Transportdienst festgelegt, der sich stark an den im OSI95-Projekt [5] definierten Transportdienst orientiert. Die Art des Transportdienstes beeinflußt nur den Transportdienstschnittstellen-Automaten, für die Protokollautomaten ist die Art des Dienstes transparent. Der Transportdienstschnittstellen-Automat läßt sich aber auch leicht an andere Dienstformen anpassen.

2.2. Protokollautomaten

Das Kommunikationsprotokoll zwischen Transportsubsystem-Instanzen setzt sich aus mehreren, voneinander weitgehend unabhängigen Teilprotokollen zusammen, wobei die Kommunikation zwischen den einzelnen Instanzen durch die direkte Kommunikation zwischen einzelnen, in der Regel gleichartigen Protokollautomaten erfolgt. Die Teilprotokolle verwenden jeweils spezielle Protokolldateneinheiten, um das Multiplexen von Protokolldateneinheiten verschiedener Protokollautomaten zu vermeiden und Koordinationspunkte zwischen Protokollautomaten einer Instanz zu minimieren. Die verwendeten Datenformate der Protokolldateneinheiten orientieren sich an der Zellengröße in zellenbasierten Netzen. Sie erlauben damit eine effiziente Übertragung und vereinfachen ihre Komposition und Analyse. Der vom Austausch der Benutzerdaten entkoppelte Kontrolldatenaustausch erlaubt eine weitgehende Unabhängigkeit von Steuer- und Datentransferfunktionen und damit deren parallele Verarbeitung.

Für jede Steuerfunktion sowie für die Datentransferfunktionen sind separate Protokollautomaten vorhanden. Der Austausch von Quittungen und Verbindungszustandsdaten zwischen Instanzen wird nicht durch gesendete oder empfangene Benutzerdatenpakete veranlaßt, sondern periodisch durchgeführt oder durch explizite Anforderungen bzw. Fehlersituationen initiiert. Da jeweils separate Protokollautomaten für die Sende- und die Empfangsrichtung einer Verbindung existieren, wird die Verarbeitung der beiden Datenflußrichtungen vollständig getrennt und kann damit parallel erfolgen. Die Teilprotokolle der einzelnen Protokollautomaten besitzen soweit erforderlich ihre eigenen Fehlerbehandlungsmechanismen und Zeitgeber, womit die in der Regel aufwendige Zeitgeberverwaltung auf mehrere Protokollautomaten verteilt wird. Zur Unterstützung einer Duplex-Verbindung existieren folgende Protokollautomaten:

- der Verbindungsverwaltungs-Automat zum impliziten oder Handshake-basierten Aufbau, zur Überwachung und zum normalen, geordneten oder Zeitgeber-gesteuerten Abbau von Verbindungen,
- zwei Automaten zum Benutzerdatenaustausch (Segmentieren, Reassemblieren von Transportdienstdateneinheiten und die reihenfolgetreue Auslieferung sind wählbar),
- je ein Automat zur Quittierung und Übertragungswiederholung (Go-Back-N-Strategie oder selektive Übertragungswiederholung),
- zwei Automaten zur Flußkontrolle (Fenstermechanismus, Übertragungsratenkontrolle, Stop-And-Go-Verfahren oder Kombinationen) sowie
- zwei Automaten zur Staukontrolle (verwendete Algorithmen: adaptive Zugangskontrolle [6] und Paketpaarverfahren [7]).

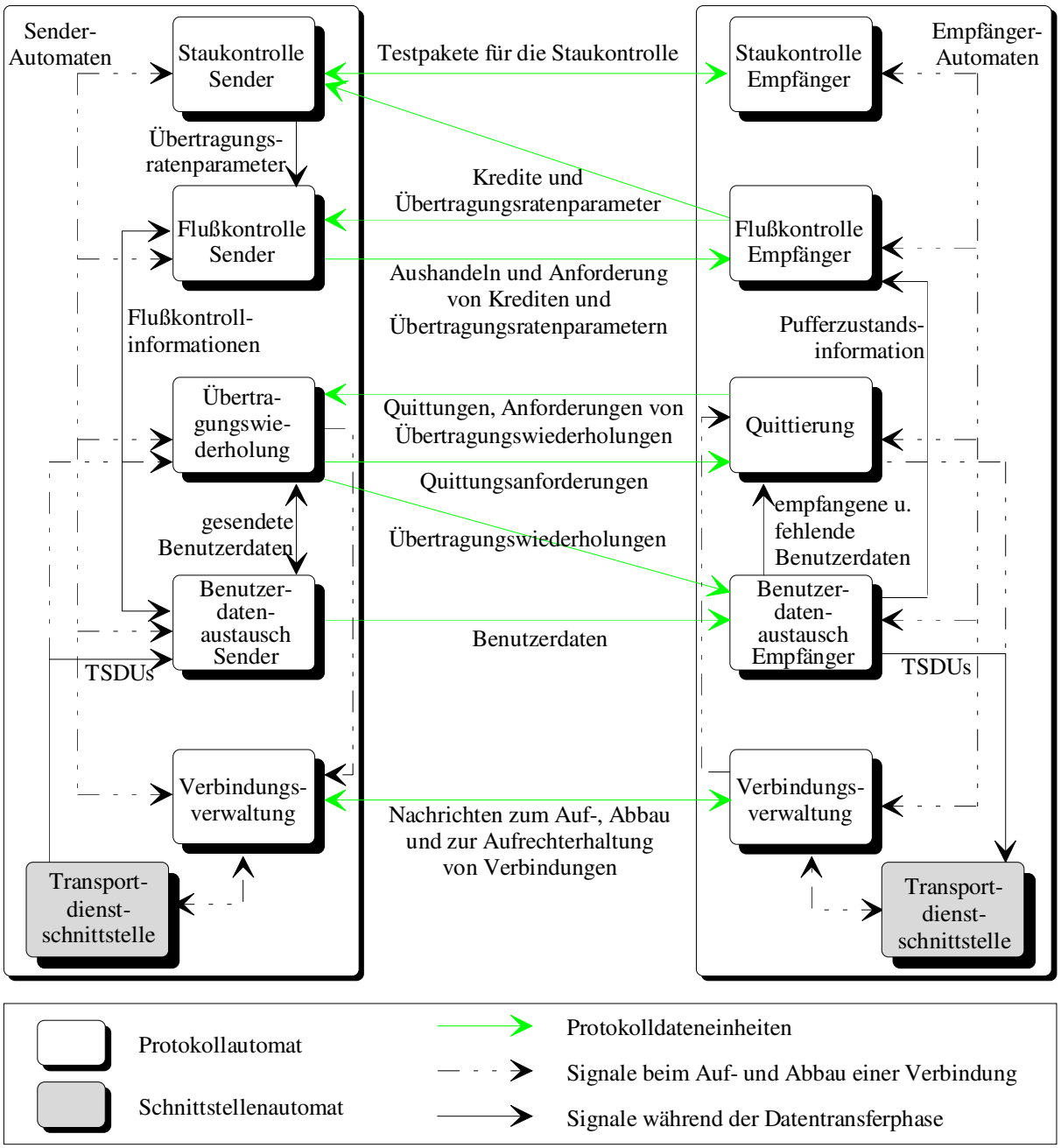

Bild 2: PATROCLOS-Protokollautomaten

Innerhalb einer Protokollinstanz kooperieren die einzelnen Automaten durch den Austausch von Signalen (s. Bild 2). Diese interne Kommunikation wurde auf ein Minimum reduziert, um die Abhängigkeiten zwischen den Automaten zu reduzieren und damit einen höheren Parallelitätsgrad zu erreichen. Beispielsweise verläuft der Signalaustausch zwischen den Automaten während der Datentransferphase in der Regel unidirektional. Signale werden außerdem nicht abhängig von eintreffenden oder zu sendenden Benutzerdatenpaketen, sondern periodisch generiert. Beim Senden von Benutzerdaten muß beispielsweise der Benutzerdatenaustausch-Sender-Automat dem Übertragungswiederholungsautomat periodisch die im letzten Zeitintervall übertragenen Benutzerdaten mitteilen. Außerdem erhält der Benutzerdatenaustausch-Sen-

der-Automat ebenfalls periodisch vom Flußkontrollautomaten die maximal zum Senden erlaubte Sequenznummer. Auf der Empfangsseite teilt der Benutzerdatenaustausch-Empfänger-Automat in gewissen Zeitabständen dem Quittierungsautomaten die angekommenen Benutzerdaten mit und schickt dem Flußkontrollautomaten Informationen über den Zustand die zur Speicherung der Benutzerdaten verfügbaren Puffer.

Ein wesentlicher Vorteil der funktionsbasierten Dekomposition in Protokollautomaten besteht auch in der sich aus der Modularität der Protokollarchitektur ergebenden leichten Konfigurierbarkeit und Anpaßbarkeit des Protokolls an Eigenschaften der unterliegenden Netzwerke sowie an unterschiedliche, sich eventuell dynamisch ändernde Anwendungsanforderungen. Insbesondere zur Unterstützung der wachsenden Menge verschiedenartiger Anwendungsszenarien und zur effizienten Arbeitsweise über unterschiedlichen Netzwerken ist diese Eigenschaft von eminenter Bedeutung [8]. Dadurch, daß die einzelnen Protokollautomaten nur dedizierte, streng voneinander abgegrenzte Protokollfunktionen bearbeiten, wird die Selektion und die Parametrisierung einzelner Protokollfunktionen wesentlich erleichtert. Es zeigte sich, daß die Auswahl von Protokollmechanismen und zugehöriger Parameter nur wenige, meist sogar nur einen einzigen Protokollautomaten beeinflußt.

2.3. Vergleich mit anderen Ansätzen

Neben XTP gibt es eine Reihe weiterer Transportprotokolle, die parallele Implementierungen besser als konventionelle Protokolle unterstützen sollen. SNR [9] ist ein Transportprotokoll zur Unterstützung von Hochgeschwindigkeitskommunikation in Datagramm- und verbindungsorientierten Netzwerken und wurde an den AT&T Bell Laboratories im wesentlichen für eine parallele Implementierung auf einer Multiprozessorarchitektur entworfen. Ziel der Entwickler war außerdem eine Reduzierung der Protokollkomplexität. Das Protokoll verwendet hauptsächlich einige der sich für Hochgeschwindigkeitsnetze etablierenden Protokollmechanismen, hat insgesamt aber nur eine relativ geringe Funktionalität. Neben der Trennung von Kontroll- und Benutzerdaten bestand die wesentliche Idee darin, während der Datenübermittlung periodisch und unabhängig von sonstigen Vorkommnissen Zustandsinformationen in speziellen Kontrollpaketen zur Flußkontrolle und zur Fehlerbehandlung zwischen den Partnerinstanzen auszutauschen. Die Granularität der Aufteilung der Funktionen in parallele Prozesse ist relativ grob, so daß insgesamt nur ein geringer Parallelitätsgrad erreicht wird. Die Protokollarchitektur erlaubt nur zwei vollständig voneinander unabhängige Teilsysteme zum Senden und Empfangen von Daten. Die Übertragungswiederholung ist nicht wie in PATROCLOS von der

regulären Übertragung von Benutzerdaten getrennt. Die parallelen Prozesse sind außerdem durch gemeinsame Verbindungszustandsdaten enger miteinander gekoppelt.

Das MultiStream Protocol (MSP) [10] ist ebenfalls in mehrere unterschiedliche sogenannte Protokollmaschinen unterteilt. Das Protokoll basiert auf den in der HOPS-Architektur (HOPS: horizontally oriented protocol structure) [11] beschriebenen Konzepten und ist als eine Menge von parallel ausführbaren Protokollfunktionen spezifiziert. Die angebotenen Protokollfunktionen können vom Benutzer ein- oder ausgeschaltet werden. Die Protokollfunktionen beschränken sich im wesentlichen auf die Quittierung, die Übertragungswiederholung und die Auslieferungsstrategie an den Benutzer. Allerdings hat der Benutzer lediglich die Wahl zwischen acht sogenannten Strömen, die jeweils eine bestimmte Kombination der Protokollfunktionen festlegen. Die einzelnen Protokollfunktionen sind aber nicht wie in PATROCLOS unabhängig voneinander wählbar. Für die Übertragungswiederholung ist ebenfalls eine eigene Protokollmaschine vorgesehen. Benutzerdaten- und Kontrollfunktionen sind jedoch enger miteinander gekoppelt. Beispielsweise werden Quittungen in Benutzerdatenpaketen übertragen (piggy-backing). Eine Signalisierung zwischen den Protokollmaschinen wird teilweise dadurch vermieden, daß verschiedene Berechnungen in mehreren Protokollmaschinen redundant ausgeführt werden.

TP++ [12] ist ein Ansatz für ein Hochgeschwindigkeits-Transportprotokoll, der bei Bellcore verfolgt wird. Spezielle Datenformate sollen die voneinander unabhängige Verarbeitung von Daten und Kontrollinformationen ermöglichen. Desweiteren wird beim Entwurf darauf geachtet, daß die meisten Protokollfunktionen eintreffende Pakete nicht in einer bestimmten Reihenfolge, sondern unabhängig voneinander bearbeiten können.

3. Implementierung

Um die Leistungsfähigkeit der entwickelten Protokollarchitektur zu belegen, wurde das Transportsubsystem auf einer hybriden Multiprozessorarchitektur (Bild 3) implementiert, die sich aus Universalprozessoren (Transputer) für Funktionen auf Protokolldateneinheiten- und auf Transportdienstdateneinheiten-Ebene sowie aus dedizierten Hardwareeinheiten für Funktionen auf Zellenebene und zur Unterstützung der Parallelität zusammensetzt. Mit dem hybriden Ansatz soll eine hohe Leistungsfähigkeit bei gleichzeitiger Flexibilität garantiert werden.

Die in Tabellenform und in der Spezifikationssprache PROMELA [13] erstellte, formale Spezifikation des parallelen Transportsubsystemprotokolls wurde direkt in eine prototypische Implementierung umgesetzt, indem die Automaten auf Transputerprozesse abgebildet wurden. Die Prozesse sind auf separaten Prozessoren bzw. Prozessormodulen ablaufbar und über Ka-

näle miteinander verbunden, welche auf Software- oder Interprozessorkanäle abgebildet werden. Ein Prozessormodul setzt sich dabei aus mindestens einem Prozessor, einer Hardwareeinheit zum Reassemblieren empfangener Zellen und einem gemeinsamen Speicher mit entsprechender Zugriffsteuerung auf den mit den Prozessoren gemeinsam benutzten Bus zusammen. Die gemeinsamen Speicher stehen den Prozessoren für zu sendende oder empfangene Benutzerdaten zur Verfügung, insbesondere um Kopiervorgänge zwischen den Prozessoren zu vermeiden.

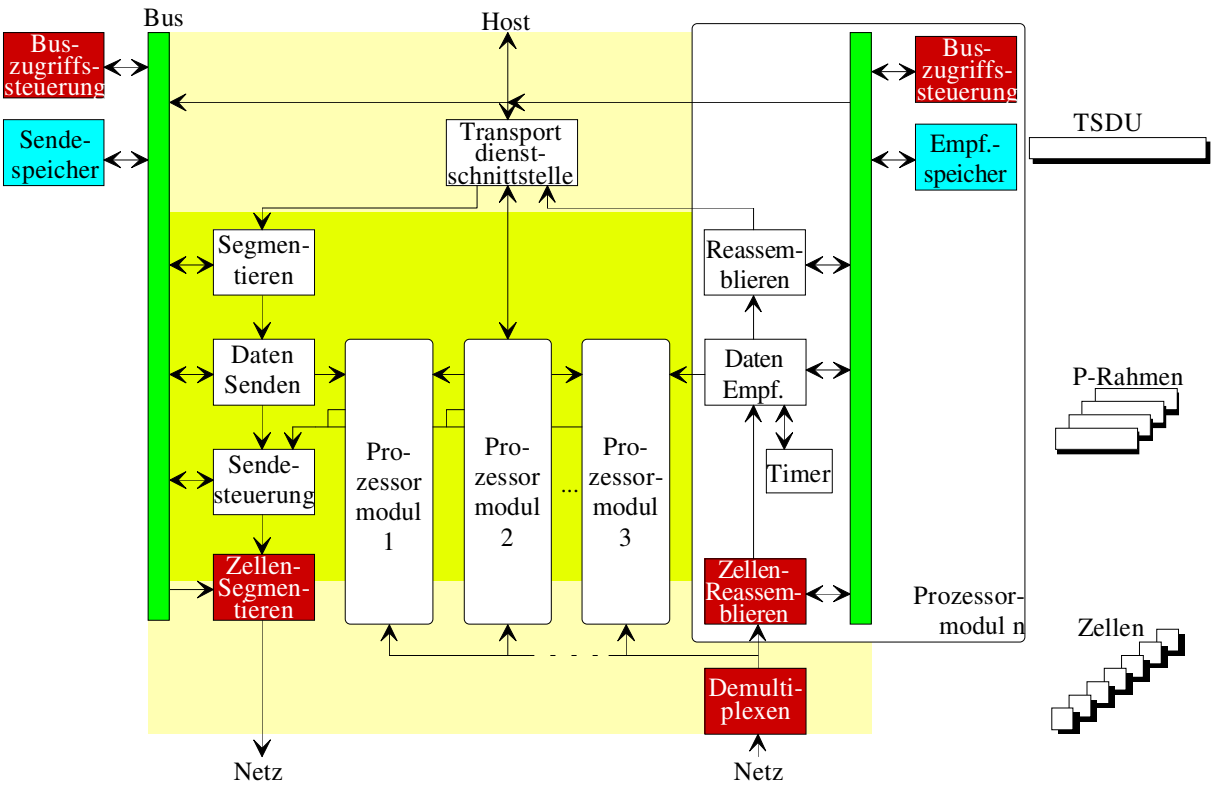

Bild 3: Hybride Implementierungsarchitektur

Ein in Hardware implementierter Demultiplexer [14] verteilt ankommende Zellen an die für deren weitere Verarbeitung zuständigen Prozessormodule, so daß diese Verarbeitung im folgenden parallel erfolgen kann. Das Segmentieren von Protokolldateneinheiten in Zellen und die entsprechenden Reassemblierfunktionen zum Anschluß des Transportsubsystems an zellenbasierte Netze wurden ebenfalls in Hardware implementiert. Weitere Spezialhardwareeinheiten für besonders zeitkritische Funktionen wie z.B. zur Speicherverwaltung können zukünftig Softwarerealisierungen ersetzen.

4. Leistungsbewertung

4.1. Software

Mit Hilfe des hybriden Monitorsystems NETMON (Network Monitor) [15] und des Auswertewerkzeugs NETINT (NETMON Interpretation Tool) wurden an der Software der parallelen Implementierung Leistungsmessungen durchgeführt. Zu Leistungsmessungen mit NETMON wird der Quellcode der Software mit Ereignissen versehen, die während der Laufzeit von der NETMON-Hardware erfaßt und von der NETMON-Software weiterverarbeitet wird.

Bei den Leistungsmessungen kristallisierten sich die Prozesse zum Senden und Empfangen von Benutzerdaten mit 126 µs bzw. 159 µs als Engpässe heraus. Die Durchsatzergebnisse hängen jedoch nicht nur von der Zeit zur Verarbeitung eines Benutzerdatenpakets, sondern auch von der Häufigkeit des Austauschs von Kontrollinformationen mit anderen Prozessen ab. Beispielsweise muß der Daten-Sende-Prozeß in gewissen Abständen dem Übertragungswiederholungs-Prozeß Informationen über die gesendeten, d.h. vom Partner zu quittierenden Pakete zukommen lassen. Der Übertragungswiederholungs-Prozeß benötigt diese Informationen bevor er die entsprechende Quittung vom Empfänger erhält, d.h. der Austausch der Informationen sollte mindestens einmal pro Paketumlaufzeit erfolgen. Analog muß der Daten-Empfangs-Prozeß dem Quittierungs-Prozeß periodisch die empfangenen Daten mitteilen, damit dieser Quittungen erzeugen kann.

Beim Senden wird in der Regel ein Durchsatz von über 6000 Paketen/s (maximal 8000 Pakete/s) erzielt. Beim Empfangen werden üblicherweise mehr als 5000 Pakete/s erreicht, der Durchsatz ist hier auf 6300 Pakete/s begrenzt. Bei Duplex-Verkehr begrenzt der Tansportdienstschnittstellen-Prozeß den erreichbaren Durchsatz auf 5500 Transportdienstdateneinheiten pro Sekunde.

Im Vergleich zu anderen parallelen Protokollimplementierungen auf Transputern stellen die erzielten Werte ein beachtliches Ergebnis dar. Auch der mit der parallelen Architektur erzielbare Leistungsgewinn gegenüber einer sequentiellen Verarbeitung liegt vergleichsweise hoch und bewegt sich üblicherweise zwischen 2,5 und 3,5 beim Senden sowie zwischen 2 und 2,5 beim Empfangen. Diese Werte übertreffen ebenfalls die Ergebnisse anderer paralleler Implementierungen [2, 16].

Bei einer häufigen Durchführung von Kontrollfunktionen geht der Durchsatz sowohl bei der parallelen als auch bei der sequentiellen Implementierung zurück. Allerdings macht sich dieser Effekt bei der sequentiellen Implementierung wesentlich stärker bemerkbar. Da bei der

parallelen Implementierung die Kontrollfunktionen parallel ausgeführt werden können, wird die Verarbeitung von Benutzerdaten und damit der erzielbare Durchsatz in einem wesentlich geringeren Maß beeinträchtigt. Die unterschiedliche Auswirkung der Kontrollfunktionen ist damit für die variierenden Speed-up-Werte verantwortlich. Gleichzeitig unterstreichen die Ergebnisse den Nutzen des Ansatzes, die Benutzerdaten und die Kontrollfunktionen zu trennen und parallel auszuführen.

4.2. Hardware

Obwohl sich die Hardwarekomponenten nur aus einfachen PLDs zusammensetzen, erreichen sie ebenfalls gute Leistungswerte. Die Komponenten zum Segmentieren und Reassemblieren können über 470000 Zellen/s verarbeiten, während der Demultiplexer mehr als 700000 Zellen/s and die angeschlossenen Prozessoren verteilen kann. Die hohe Leistung des Demultiplexers kommt dadurch zustande, daß sämtliche Kontrolloperationen parallel zum Speichern und Weiterleiten der eintreffenden Zellendaten vorgenommen werden können.

5. Ausblick

Mit PATROCLOS wurde ein paralleles Transportsubsystem entwickelt, das sich insbesondere für den Einsatz über zellenbasierten Netzen und die Implementierung auf parallelen Architekturen eignet. Es zeigte sich, daß das bereits beim Protokollentwurf berücksichtigte Parallelitätskonzept wesentliche Leistungsvorteile bei einer parallelen Implementierung mit sich bringt. Zur weiteren Erhöhung der Leistungsfähigkeit der in Software implementierten Teile können schnellere Prozessoren wie digitale Signalprozessoren oder Transputer der neuen Generation eingesetzt werden, welche eine bis zu zehnfache Leistungssteigerung versprechen.

Neben der hybriden Multiprozessorarchitektur wurde für das Transportsubsystem eine VLSI-Implementierungsarchitektur [17] entworfen, um noch höhere Leistungen zu erzielen und damit dem steigenden Bedarf an hochleistungsfähigen Kommunikationssystemen gerecht zu werden. Für die VLSI-Implementierungsarchitektur existieren bereits Beschreibungen der Protokollverarbeitungs- und Speicherverwaltungskomponenten in der Hardwarebeschreibungssprache VHDL. Die PATROCLOS-Protokollarchitektur besitzt dadurch, daß keine von mehreren Funktionseinheiten, d.h. Automaten, gemeinsam genutzten Strukturen wie gemeinsame Verbindungszustandsdaten, Zeitgeber, Protokolldateneinheiten usw. existieren, bei einer Realisierung wesentliche Vorteile gegenüber konventionellen Protokollen [18]. Durch den modularen Aufbau kann auf verschiedene bei der VLSI-Implementierungsarchitektur vorgesehene Komponenten verzichtet werden.

Um die vielfältigen vom Transportsystem angebotenen Protokollfunktionen für verschiedenartige Anwendungen und Netzwerke auszunutzen, ist eine Integration von PATROCLOS mit Werkzeugen des funktionsbasierten Kommunikationssubsystems F-CSS [19] in Bearbeitung. Beispielsweise wird der F-CSS-Protokollkonfigurator dazu verwendet, um aus einer Beschreibung der geforderten Dienstgüte die dazu notwendigen Protokollfunktionen und -parameter für die PATROCLOS-Automaten zu ermitteln.

Literatur

[1] Strayer, W.T.; Dempsey, B.J.; Weaver, A.C.: *XTP: The Xpress Transfer Protocol*, Addison-Wesley Publishing Company, 1992

[2] Braun, T.; Zitterbart, M.: *A Parallel Implementation of XTP on Transputers*, 16th Annual IEEE Conference on Local Computer Networks, 14.-17. Oktober 1991, Minneapolis, Minnesota, S. 172-179

[3] Braun, T.; Zitterbart, M.: *Parallel Transport System Design*, in: Danthine, A.; Spaniol, O. (Hrsg.): High Performance Networking, IV, IFIP, North-Holland, 1993, S. 397-412

[4] Braun, T.: *Ein paralleles Transportsubsystem für zellenbasierte Hochgeschwindigkeitsnetze*, Dissertation, Universität Karlsruhe, Fortschritt-Berichte VDI, Reihe 10: Informatik/Kommunikationstechnik, Nr. 264, VDI-Verlag, Düsseldorf, 1993

[5] Danthine, A.; Baguette, Y.; Leduc, G.; Leonard, L.: *The OSI 95 Connection-mode Transport Service: The Enhanced QoS*, in: Danthine, A.; Spaniol, O. (Hrsg.): High Performance Networking, IV, IFIP, North-Holland, 1993, S. 397-412

[6] Haas, Z.: *Adaptive Admission Congestion Control*, Computer Communication Review, Vol. 21, No. 5, Oktober 1991, S. 58-76

[7] Keshav, S.; Agrawala, A.; Singh, S.: *Design and Analysis of a Flow Algorithm for a Network of Rate Allocating Servers*, in: Johnson, M.J. (Hrsg.): Protocols for High-Speed Networks, II, North-Holland, 1991, S. 55-72

[8] Zitterbart, M.; Stiller, B.; Tantawy, A.: *A Model for Flexible High-Performance Communication Subsystems*, Journal of Selected Areas in Communication, Vol. 11, No. 4, Mai 1993, S. 507-518

[9] Sabnani, K.; Netravali, A.; Roome, R.: *Design and Implementation of a High Speed Transport Protocol*, IEEE Transactions on Communications, Vol. 38, No. 11, November 1990, S. 2010-2024

[10] LaPorta, T. F.; Schwartz, M.: *The MultiStream Protocol: A Highly Flexible High-Speed Transport Protocol,* Journal of Selected Areas in Communication, Vol. 11, No. 4, Mai 1993, S. 519-530

[11] Haas, Z.: *A Protocol Structure for High-Speed Communication over Broadband-ISDN*, IEEE Network, Vol. 5, No. 1, Januar 1991, S. 64-70

[12] Feldmeier, D.C.: *An Overview of the TP++ Transport Protocol Project*, in: Tantawy A. (Hrsg.): High Performance Networks: Frontiers and Experience, Kluwer Academic Publishers, 1994

[13] Holzmann, G.J.: *Design and Validation of Protocols: A Tutorial,* Computer Networks and ISDN Systems, Vol. 25, 1993, S. 981-1017

[14] Braun, T.: *Ein Demultiplexer zur Unterstützung von Parallelität in einem ATM-Transportsystem*, 2. ITG/GI-Workshop Verbindungsnetzwerke für Parallelrechner und Breitband-Übermittlungssysteme, Stuttgart, 4.-5. Oktober 1993

[15] Endriss, O.; Steinbrunn, M.; Zitterbart, M.: *NETMON-II a Monitoring Tool for Distributed and Multiprocessor Systems*, 4th International Conference on Data Communications and their Performance, Barcelona, 20.-22. Juni 1990.

[16] Rütsche, E.; Kaiserswerth, M.: *TCP/IP on the Parallel Protocol Engine*, in: Danthine, A.; Spaniol, O. (Hrsg.): High Performance Networking, IV, IFIP, North-Holland, 1993, S. 119-134

[17] Schiller, J.; Braun, T.: *VLSI-Implementation Architecture for Parallel Transport Protocols*, IEEE Workshop on VLSI in Communications, Stanford Sierra Camp, Lake Tahoe, Kalifornien, 15.-17. September 1993

[18] Schiller, J.: *VLSI-Implementierungsarchitektur für parallele Transportprotokolle*, GI/ITG-Fachgruppe "Kommunikation und Verteilte Systeme": 1. Arbeitstreffen "Architektur und Implementierung von Hochleistungs-Kommunikationssystemen", Karlsruhe, 17./18. Januar 1994

[19] Stiller, B.: *FuKSS: Ein funktionsbasiertes Kommunikationssubsystem zur flexiblen Konfiguration von Kommunikationsprotokollen*, GI/ITG-Fachgruppe "Kommunikation und Verteilte Systeme": 1. Arbeitstreffen "Architektur und Implementierung von Hochleistungs-Kommunikationssystemen", Karlsruhe, 17./18. Januar 1994