Protokollanalyse mit Transputernetzen

Torsten Braun
Institut für Telematik, Universität Karlsruhe,
Postfach 6980, Zirkel 2, W7500 Karlsruhe
Telefon: 0721/608-3982, Fax: 0721/60 60 97
E-Mail: braun@telematik.informatik.uni-karlsruhe.de

Kurzfassung

Zur Fehlersuche und Leistungsbewertung in schnellen Kommunikationsnetzwerken sind leistungsfähige Protokollanalysatoren wünschenswert, die mit der Netzwerkbandbreite schritthalten können und so dem Anwender sämtliche interessierende Daten liefern können. Der in diesem Bericht vorgestellte Architekturansatz basiert auf parallelen Verarbeitungseinheiten innerhalb des Protokollanalysators. Nach der Diskussion einiger Parallelisierungskonzepte, wird eine parallele Protokollanalysator-Architektur vorgestellt und am Beispiel eines Analysators für ISDN näher erläutert. Bereits dessen Hardware-Anschaltung an ISDN unterstützt die parallele Verarbeitung der Protokolldaten.

1. Einleitung

Transputer [1] haben in der Kommunikationstechnik mittlerweile einen gewichtigen Stellenwert erreicht. Zahlreiche Kommunikations-Controller mit Transputern für ISDN, Ethernet, Token-Ring oder FDDI [2] sind kommerziell verfügbar oder wurden als Prototypen realisiert. Transputernetze werden speziell in Hochgeschwindigkeitsnetzwerk-Umgebungen zur Realisierung intelligenter Controller auf parallelen Architekturen eingesetzt [3]. Durch die damit mögliche parallele Verarbeitung der Kommunikationsprotokolle können beträchtliche Leistungswerte erzielt werden [4].
Die Entwicklung und der Test von Kommunikationssoftware bzw. Implementierungen von Kommunikationsprotokollen sowie der Betrieb von Netzwerken erfordern Werkzeuge zur Fehlersuche und zu Leistungsuntersuchungen. Eine bedeutende Klasse solcher Werkzeuge sind sogenannte ***Protokollanalysatoren***. Sie werden direkt an ein zu untersuchendes Netz angeschlossen, erfassen elektrische Signale und werten diese aus. Leistungsfähige Protokollanalysatoren zeigen dabei nicht nur die auf dem Netz transportierten Daten an, sondern nehmen zusätzlich noch Analysen der zwischen den Netzknoten bzw. Endsystemen ausgetauschten Protokollinformationen höherer Schichten vor. Die interessierenden Daten werden gefiltert und on-line oder off-line in einer Grafikstation visuell aufbereitet.
Das Filtern und Aufbereiten der Protokollinformation sind sehr zeitaufwendige Vorgänge, die bei sehr schnellen Netzen nicht mehr mit der auf dem Netz verfügbaren Übertragungsrate schritthalten können. In diesem Beitrag wird deshalb die Struktur eines Protokollanalysators erläutert, der durch den Einsatz mehrerer Transputer die für die on-line-Analyse erforderliche Leistungsfähigkeit erreichen kann. Die Grundlage zum Entwurf bildeten Konzepte zur Paralle-

lisierung der Analysefunktionen, die im wesentlichen aus Arbeiten gewonnen wurden, welche die parallele Verarbeitung von Kommunikationsprotokollen in Netzwerkadaptern behandeln.

2. Parallele Implementierung von Kommunikationsprotokollen

Im Bereich der Hochgeschwindigkeitskommunikation hat sich in den letzten Jahren das Gebiet der Parallelverarbeitung von Kommunikationssoftware etabliert. Bild 1 zeigt die parallele Implementierung eines OSI-Transportsubsystems, welches die OSI-Schichten 2-4 bearbeitet.

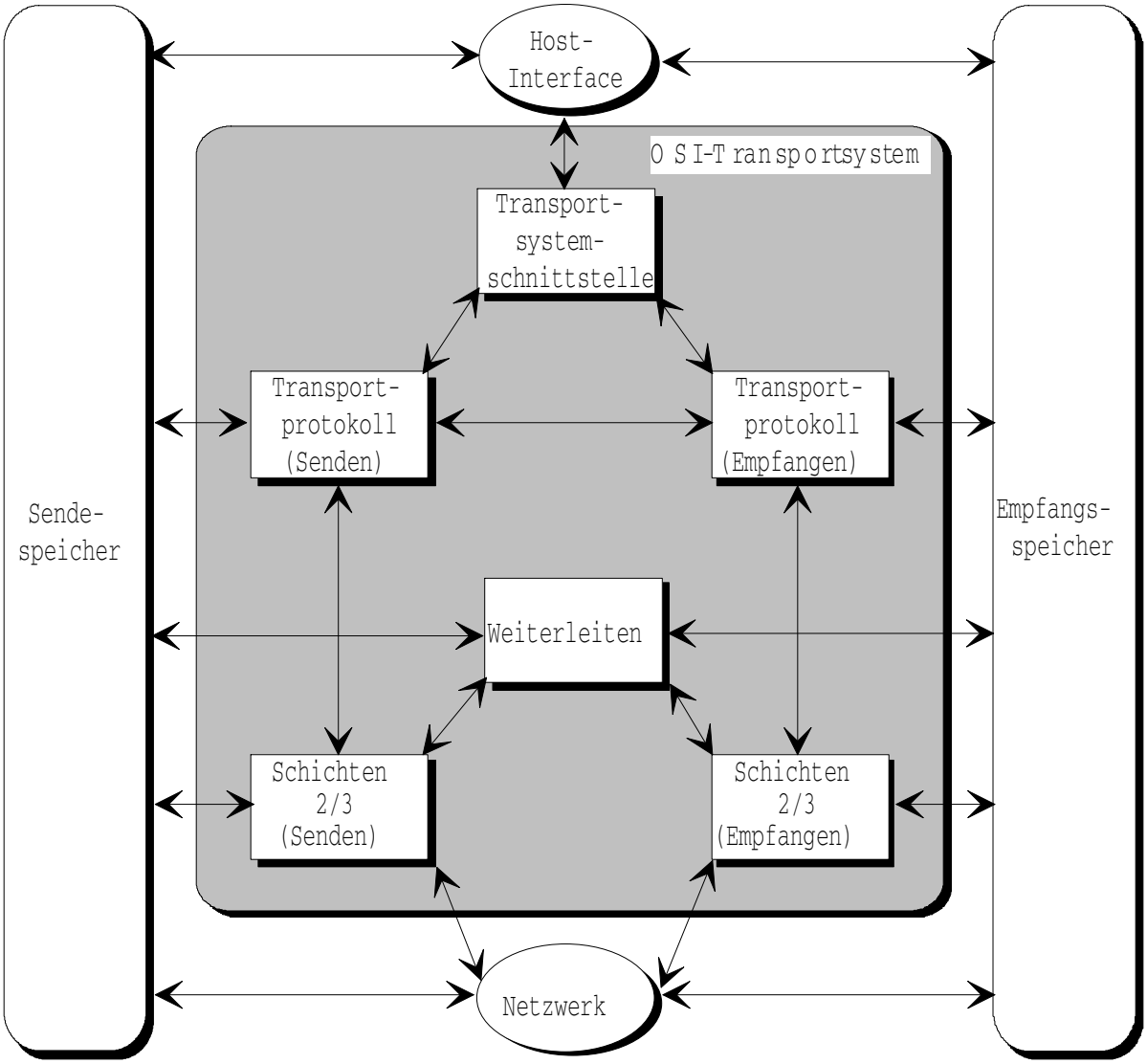

Abb. 1: Parallele Implementierung eines OSI-Transportsystems

Als wesentliches Element bei der Verarbeitung von Kommunikationsprotokollen dient die sogenannte Protokolldateneinheit (protocol data unit, PDU) oder auch einfach Paket genannt, die zum Austausch von Nachrichten und Daten zwischen kommunizierenden Protokollinstanzen dient. Die Verarbeitung, d.h. das Erzeugen und Analysieren, von Paketen bildet den Hauptbestandteil bei der Protokollverarbeitung. Die dabei verbreiteten und für die parallele Architektur von Protokollanalysatoren verwendbaren Konzepte (vgl. auch [5]) werden in der folgenden

Übersicht vorgestellt. Um eine hohe Leistungsfähigkeit zu erzielen, können die einzelnen Konzepte integrierend angewendet werden.
Die Verarbeitung von Paketen kann grundsätzlich zeitlich oder räumlich parallel erfolgen [6]. Bei der ***zeitlichen Parallelität*** werden die Pakete in einer Pipeline verarbeitet, dabei kann jede Protokollschicht ein Element einer solchen Pipeline bilden. Bei einer guten Auslastung befinden sich mehrere Pakete gleichzeitig in einer Pipeline. Da die Leistungsfähigkeit von der langsamsten Pipelinestufe abhängt, müssen die einzelnen Pipelinestufen möglichst gleichmäßig ausgelastet werden.
Diese gleichmäßige Auslastung kann durch den Einsatz ***räumlicher Parallelität*** unterstützt werden. Einzelne Pipelineelemente oder ganze Pipelines können dazu mehrfach angeordnet werden, um mehrere Pakete gleichzeitig bearbeiten zu können. Die zu bearbeitenden Pakete werden nach bestimmten Strategien auf die einzelnen Pipelineelemente verteilt.

- Pakete werden je nach Auslastung oder in einem Round-Robin-Verfahren den freien Pipelineelementen zugeordnet.
- Sende- und Empfangsfunktionen können getrennt werden, d.h. ein Pipelineelement bearbeitet zu sendende Pakete, das andere Pipelineelement bearbeitet empfangene Pakete.
- Separate Pipelineelemente für jedes angeschlossene Teilnetz erlauben die unabhängige Verarbeitung der Pakete der einzelnen Teilnetze.
- Beim Zuordnen je nach Verbindung werden für einzelne Verbindungen oder für Gruppen von Verbindungen (z.B. auf Transportschicht oder für jeden ISDN-B-Kanal) Pipelineelemente zur Verfügung gestellt.
- Das Trennen von Daten- und Kontrollpaketen ist speziell in fortgeschritteneren Protokollarchitekturen möglich, die zwischen reinen Benutzerdatenpaketen und speziellen Paketen für Steuerzwecke unterscheiden. Dieses Konzept wird beispielsweise in einer parallelen XTP-Implementierung auf Transputernetzen angewendet [7].

Einen noch feineren Parallelitätsgrad erhält man durch das gleichzeitige parallele Bearbeiten eines Pakets durch verschiedene Verarbeitungseinheiten, z.B. durch die parallele Verarbeitung unterschiedlicher Paketteile. So bietet es sich beispielsweise an, die meist rechenintensive Prüfsummenberechnung durch spezielle Verarbeitungseinheiten vorzunehmen.

3. Protokollanalysator-Architektur

Die im vorigen Abschnitt beschriebenen Konzepte bilden die Grundlage für eine parallele Protokollanalysator-Architektur. Bild 2 stellt eine mögliche Architektur dar. Der Protokollanalysator ist in diesem Fall über Netzwerk/Link-Interfaces an zwei verschiedene Netzwerke angeschlossen. Die Interfaces haben die Aufgabe, die Signale des jeweils angeschlossenen Netzwerks aufzunehmen und die erfaßten Datenbits über eine Link-Schnittstelle der auf Transputernetzen laufenden Protokollanalysator-Software zur Verfügung zu stellen. Diese Netzwerk/Link-Interfaces werden ausschließlich durch dedizierte Hardware realisiert.
Hinter dem Netzwerk-Link-Interface werden die erfaßten Daten durch die Protokollanalysator-Software weiterverarbeitet. Für jedes angeschlossene Netz ist dabei eine Pipeline vorgesehen. Jede Pipelinestufe wird als Transputer-Task unter Verwendung von INMOS-ANSI-C [8] im-

plementiert. Abhängig vom Zeitbedarf der einzelnen Pipelinestufen und der gewünschten Leistungsfähigkeit einer Pipeline werden die Tasks auf eine geeignete Transputernetz-Konfiguration abgebildet. Mit dieser Pipelinearchitektur können die Datenströme verschiedener Netze parallel zueinander analysiert werden. Außerdem besteht die Möglichkeit, innerhalb einer Pipeline die Protokollanalyse für verschiedene Protokollschichten parallel auszuführen. Besonders rechenintensive Aufgaben wie z.B. die Überprüfung von Prüfsummen können weiter auf spezielle Transputer ausgelagert werden, d.h. durch räumliche Parallelität kann die Leistungsfähigkeit einer Pipelinestufe erhöht werden.

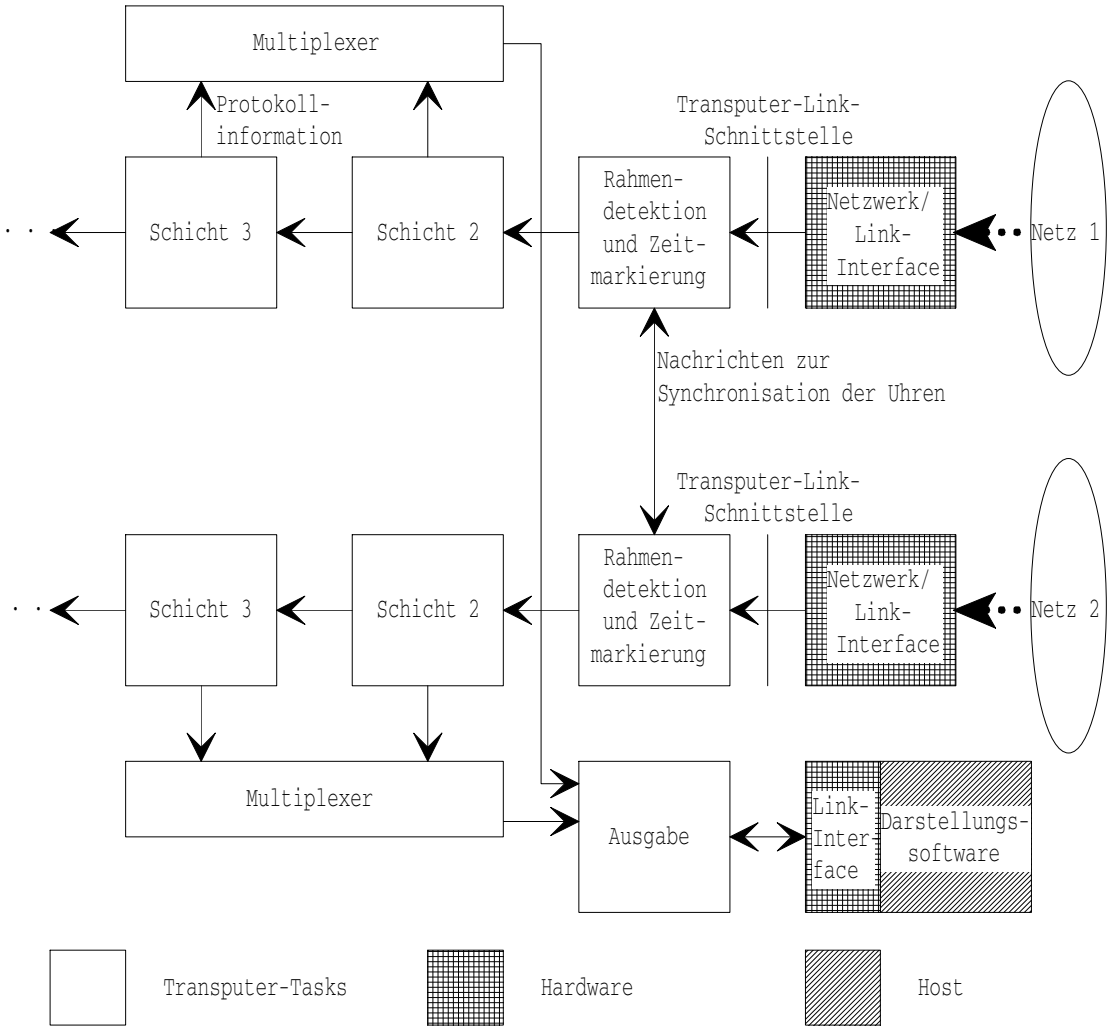

Abb. 2: Architektur des Protokollanalysators

Die erste Pipelinestufe ist zur Erkennung von Übertragungseinheiten (Rahmen) auf dem Netzwerk zuständig. Sie versieht unter Verwendung des Transputer-internen Zeitgebers den detektierten Rahmen mit einem Zeitstempel, der den Detektionszeitpunkt beschreibt. In dieser Stufe können bereits die für eine Auswertung höherer Schichten irrelevanten Teile eines Rahmens gelöscht werden. Oft interessieren z.B. nur die in den einzelnen Paketköpfen enthaltenen Protokollinformationen. Diese haben im Normalfall eine maximale Länge, so daß über diese Länge hinausgehende Pakete abgeschnitten werden können. Der Kommunikationsaufwand läßt sich durch diese Verkürzung der zwischen den Pipelinestufen weiterzuleitenden Nachrichten we-

sentlich reduzieren. Das Verfahren kann jedoch z.B. nicht angewendet werden, wenn Prüfsummen über vollständige Datenpakete überprüft werden sollen.
Die Rahmendetektion gibt anschließend den mit dem Zeitstempel versehenen Rahmen an eine Schicht-2-Einheit weiter. Hier wird das verwendete Protokoll festgestellt und die in der Protokolldateneinheit kodierte Protokollinformation in einer Form aufbereitet, die eine einfache Darstellung erlaubt. Die Schicht-2-Protokolldaten werden vom Rahmen abgestreift, welcher an die nächste Einheit weitergegeben wird.
Die Schicht-3-Einheit verfährt ebenso wie die vorhergehende Einheit, d.h., sie extrahiert die relevanten Protokolldaten und leitet den Rahmen bei Bedarf weiter. Wie in der Abbildung angedeutet, können Einheiten für weitere Schichten folgen.
Die Ergebnisse der einzelnen Einheiten werden über eine oder mehrere Multiplexer-Stufen an die Ausgabeeinheit gesendet, welche schließlich in Zusammenarbeit mit der Darstellungssoftware auf dem Host für die vom Benutzer gewünschte Darstellung der Daten sorgt.
Neben der skalierbaren Leistungsfähigkeit der Analysator-Architektur bestehen wesentliche Vorteile in der Flexibilität und in der Modularität. Durch die Verwendung von Transputer-Link-Schnittstellen an der Schnittstelle zwischen der spezifischen Hardware und den Transputern, sind lediglich die Hardwarekomponenten zum Anschluß an das zu untersuchende Netzwerk netzwerkspezifisch. Es muß lediglich ein den Leistungsanforderungen angemessenes Transputernetz konfiguriert und an die spezifische Hardware angeschlossen werden, bevor die entsprechenden Tasks auf die einzelnen Transputer geladen werden.

4. Realisierung eines ISDN-Protokollanalysators

Der Analysator untersucht sämtliche auf dem D-Kanal übertragenen Nachrichten, d.h. die von den Endgeräten über den Netzabschluß an die Vermittlungsstelle gesendeten Nachrichten sowie die Nachrichten, welche von der Vermittlungsstelle an die Endgeräte geschickt werden. Diese beiden Datenströme werden getrennt voneinander, d.h. parallel, untersucht.
Die Trennung der Datenströme erfolgt bereits durch das in Abs. 4.1. näher beschriebene ISDN/Transputer-Link-Interface, welches dem Analysator über zwei Link-Schnittstellen die Datenbits zur Verfügung stellt. Bild 2 zeigt die Architektur einer bereits realisierten Konfiguration des Analysators für ISDN-D-Kanal-Protokolle an einem ISDN-Basisanschluß. Die Netze 1 und 2 stehen dabei für die Datenströme von den Endgeräten zum Netzabschluß bzw. vom Netzabschluß zu den Endgeräten.
Das Interface liefert über die Link-Schnittstelle byteweise die auf dem ISDN-D-Kanal erkannten Datenbits. Funktionen wie die Rahmendetektion und das bei ISDN verwendete Bit-Stuffing-Verfahren werden in der Rahmendetektionseinheit in Software durchgeführt. Die Einheiten der Schichten 2 und 3 extrahieren die jeweiligen Protokolldateneinheiten und die interessierenden Protokollinformationen des D-Kanal-Protokolls.

4.1. ISDN/Transputer-Link-Interface

In Abbildung 3 ist das ISDN/Transputer-Link-Interface [9] dargestellt, welches die auf dem D-Kanal detektierten Bits byteweise über einen Transputer-Link zur Verfügung stellt. Die Bits werden auf dem Baustein PEB 2080 [10] abgegriffen, der auf einer gewöhnlichen ISDN-Controller-Karte für Personal Computer direkt an der S_0-Schnittstelle plaziert ist. An der sogenannten IOM®-Schnittstelle (Signale: Serial Data Out (SDO), Serial Data In (SDI), Frame Sync (FSC), Serial Data Clock (DCLK)) liegen die vom Endgerät empfangenen und gesendeten B- und D-Kanal-Bits an. Von dieser Schnittstelle werden die vom Netzabschluß an die Endgeräte gesendeten D-Kanal-Bits aufgenommen. Da der PEB 2080 an einem speziellen Ausgang (X2) zusätzlich die Echo-Kanal-Bits anlegt, können auch die von allen an den Basisanschluß angeschlossenen Endgeräten in Richtung Netzabschluß gesendeten D-Kanal-Bits erkannt werden.

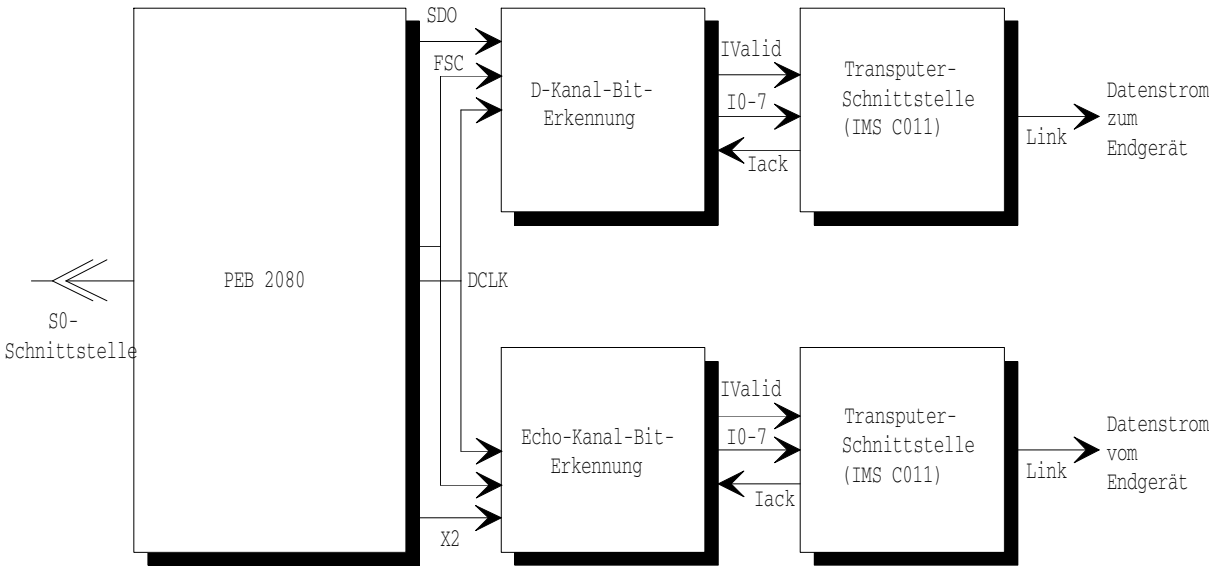

Abb. 3: ISDN/Transputer-Link-Interface

Dadurch, daß am PEB 2080 die Echo-Kanal-Bits taktgleich zu den D-Kanal-Bits anliegen, ist es möglich, für die Echo-Kanal-Bit-Erkennung exakt dieselbe Schaltung wie zur D-Kanal-Bit-Erkennung zu verwenden. In der D- und der Echo-Kanal-Bit-Erkennung erfolgt schließlich die jeweilige Extrahierung der relevanten Bits und die byteweise Weitergabe an den Link-Schnittstellen-Baustein C011. Die beiden realisierten Bit-Erkennungseinheiten unterstützen die Erkennung von Schicht-2-Rahmen, indem sie auf die HDLC-Flags, welche einen Schicht-2-Rahmen umschließen, triggern. Sobald die Bitfolge "01111110" erkannt wird, wird die Bitfolge als ein Byte über die Transputer-Link-Schnittstelle ausgegeben. Folgen mehr als sechs "1"-Bits hintereinander, so ist der D-Kanal im Ruhezustand und das Interface bricht die Ausgabe von Bytes an den angeschlossenen Transputer ab. Aufgrund des auf der Schicht-2 des D-Kanal-Protokolls verwendeten Bit-Stuffing-Verfahrens kann es vorkommen, daß ein Rahmen aus einer nicht durch 8 teilbaren Anzahl übertragener Bits besteht. Einzelne Bits eines HDLC-Flags werden in diesem Fall in zwei unterschiedlichen Bytes an den Transputer übertragen. Die

darauf laufende Software muß diesen Sachverhalt berücksichtigen und diese Bits sowie die aufgrund des Bit-Stuffings eingefügten Bits löschen.

4.2. Funktionalität der Software

4.2.1. Rahmenerkennung und Zeitmarkierung

Vom ISDN/Transputer-Link-Interface erhält die für die Rahmenerkennung und Zeitmarkierung zuständige Task einen Bytestrom. Das Auftreten eines HDLC-Flags zeigt dabei den Anfang bzw. das Ende eines Rahmens an. Die wesentliche Aufgabe dieser Task besteht in der Wiederherstellung des Rahmens, wie er vor dem beim Sender durchgeführten Bit-Stuffing vorlag, d.h. vor dem Einfügen der "0"-Bits nach fünf aufeinanderfolgenden "1"-Bits im Schicht-2-Rahmen. Dabei muß beachtet werden, daß sich eine Folge von fünf aufeinanderfolgenden "1"-Bits über zwei verschiedene Bytes erstrecken kann. Die Aufgabe, die zusätzlichen "0"-Bits wieder zu eliminieren, kann grundsätzlich auch im ISDN/Transputer-Link-Interface in Hardware implementiert werden.
Eine weitere Aufgabe dieser Task besteht im Hinzufügen eines Zeitstempels zum detektierten Rahmen. Dieser Zeitstempel wird aus dem Wert des Transputer-internen Zeitgebers gewonnen. Falls der Analysator mehrere Transputer zur Zeitmarkierung verschiedener Datenströme benutzt, müssen diese synchronisiert werden. Die Synchronisation verschiedener Zeitgeber innerhalb eines Transputernetzes wird im folgenden Abschnitt beschrieben.

4.2.2. Synchronisation von Transputer-Zeitgebern

Falls der Analysator für die parallele Analyse mehrere Netze ausgelegt ist oder verschiedene Datenströme wie beim vorgestellten ISDN-Analysator unabhängig voneinander untersucht werden, müssen die verschiedenen Rahmendetektionseinheiten ihre lokalen Zeitgeber untereinander synchronisieren, um eine gleichlaufende Zeit in allen Pipelines zu erreichen.
Es wird dabei eine modifizierte Form des in [11] beschriebenen Algorithmus zur Synchronisation von Transputer-Zeitgebern verwendet. Der Algorithmus erfordert eine Ringstruktur des Transputernetzes. Der Ablauf geschieht in drei Phasen: der Zähl-, der Simulations- und der Synchronisationsphase.

- In einer ***Zählphase*** werden die Transputer gezählt und jedem Transputer ein Ringindex zwischen 1 und n zugewiesen.
- In der ***Simulations***- und der ***Synchronisationsphase*** werden jeweils identische Operationen ausgeführt. Ein ausgezeichneter Transputer (Master) sendet über den Ring an die nachfolgenden Transputer (Slaves) eine Synchronisationsnachricht, welche die aktuelle Zeit des Masters (Referenzzeit) und für alle Slaves die individuelle erwartete Verzögerung für das Senden der Synchronisationsnachricht vom Master zum jeweiligen Slave enthält. Die Zeit, auf die ein Slave-Zeitgeber gesetzt werden muß, berechnet sich aus der Summe von Referenzzeit und der erwarteten Verzögerung zwischen Master und dem

jeweiligen Slave. Die Synchronisationsnachricht wird von jedem Slave empfangen und an den jeweiligen Nachfolger weitergesendet bis schließlich der Master-Transputer die Nachricht wieder zurückerhält.

Die jeweilige erwartete Verzögerung der Synchronisationsnachricht zwischen Master und Slave berechnet der Master aus der benötigten Umlaufzeit der Synchronisationsnachricht während der Simulationsphase, indem die gesamte Umlaufzeit der Synchronisationsnachricht durch die Anzahl der Transputer dividiert und mit dem jeweiligen Slave-Index multipliziert wird. Die einzelnen Verzögerungszeiten werden dann an den Slaves durch ihren Ringindex bekannten Positionen in der Synchronisationsnachricht für die Synchronisationsphase eingetragen.

In Bild 4 ist der Algorithmus anhand eines Beispiels erklärt. Es werden 8 Transputer synchronisiert. Die Umlaufzeit während der Simulationsphase beträgt 800 µs, d.h. das Weitergeben der Synchronisationsnachricht zwischen zwei Transputern beträgt 100 µs. Die Synchronisationsnachricht enthält die Referenzzeit (1000) und die jeweiligen Verzögerungen zwischen Master und Slave i (i = 1…7). Die einzelnen Zeitgeber werden dann entsprechend diesen Werten gesetzt. Z.B. setzt der Slave 5 seinen Zeitgeber auf 1000 µs + 500 µs = 1500 µs.

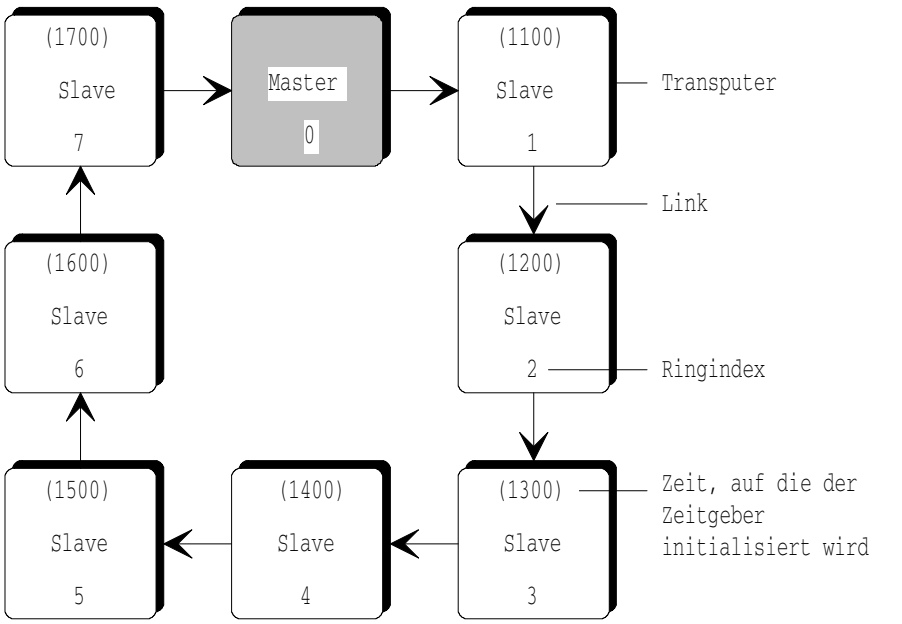

Abb. 4: Algorithmus zur Synchronisation von Transputer-Zeitgebern

Die Synchronisationsphase muß periodisch wiederholt werden, sofern die für die Zeitmarkierung verwendeten Transputer nicht durch einen gemeinsamen Taktgeber versorgt werden. Die Zeitspanne zwischen zwei Synchronisationsphasen hängt vom Drift der Transputer-Zeitgeber ab. Ansonsten reicht eine einmalige Initialisierungsprozedur aus.

4.2.3. D-Kanal-Protokoll der Schicht 2

Die für die Analyse der Schicht-2-Pakete verantwortliche Software bestimmt die Schicht-2-Adresse, die sich aus dem Service Access Point Identifier (SAPI) und dem Terminal Endpoint Identifier (TEI) zusammensetzt. Außerdem wertet diese Task das Steuerfeld aus, welches den Rahmentyp, das sog. P/F-Bit und gegebenenfalls die Sende- und die Empfangsfolgenummer enthält. Schließlich kann noch die Rahmenprüfsumme angegeben und bei Bedarf überprüft werden.

4.2.4. D-Kanal-Protokoll der Schicht 3

Die Schicht-3-Analyse-Software bestimmt den Protokolldiskriminator, die Referenznummer, den Nachrichtentyp des Schicht-3-Protokolldateneinheit sowie die Kennung, die Länge und den Inhalt der einzelnen Nachrichtenelemente. Ein Beispiel für ein Nachrichtenelement ist die Rufnummer in einer SETUP-Nachricht, die beim Aufbau eines B-Kanals vom Initiator an die Vermittlungsstelle gesendet wird.

4.2.5. Host-Software

An den Protokollanalysator wird über eine Host/Link-Schnittstelle ein Host-Rechner, z.B. ein Personal Computer oder eine Workstation, angeschlossen. Auf diesem Host wird ein Server-Programm gestartet, welches nach dem Konfigurieren und Laden der Protokollanalysator-Software auf das Transputernetz die Ergebnisse der Analysator-Software entgegennimmt, und dem Benutzer darstellt. Der Benutzer hat die Möglichkeit, den Analysator zu steuern, z.B. um die Auswahl der darzustellenden Informationen vorzunehmen.

Referenzen

[1] INMOS Ltd.
The Transputer Databook
2nd edition 1989, Redwood Burn Ltd, Trowbridge.

[2] Ulrich, R.; Dietsch H.
A Transputer-Based Communication Controller for FDDI Stations
Proceedings of the EFOC/LAN 90, München, Juni, 1990

[3] Kaiserswerth, M.
A Parallel Implementation of the ISO 8802-2.2 Protocol
Proceedings of the IEEE Conference on Communications Software: Communications for Distributed Applications & Systems, TriComm '91, Chapel Hill, NC, U.S.A., 18.-19. April 1991, pp. 75-78

[4] M. Zitterbart, T. Braun
Parallele Protokollimplementierungen für die Hochgeschwindigkeits-Kommunikation auf Transputer-Netzen
Informatik Forschung und Entwicklung, Vol. 7, Januar, 1992, S. 30-40

[5] Zitterbart, M.
Funktionsbezogene Parallelität in transportorientierten Komunikationsprotokollen
VDI Verlag, Reihe 10: Informatik/Kommunikationstechnik, Nr. 183

[6] Hwang, K.; Briggs, F.A.
Computer Architecture and Parallel Processing
McGraw-Hill, 1984

[7] Braun, T.; Stiller, B.; Zitterbart, M.
XTP and VMTP on Multiprocessor Architectures
Proceedings of the International Workshop on Advanced Communications and Applications for High-Speed Networks, München, 16.-19. März 1992

[8] INMOS Ltd.
ANSI C Toolset Language Refernce
INMOS Ltd., Bristol, U.K., 1990

[9] Skrodzki, S.
Anschaltung eines ISDN-Protokollanalysators an einen ISDN-Controller
Studienarbeit am Institut für Telematik, Universität Karlsruhe, Juli 1991

[10] Siemens AG
ICs for Communication, Data Book 1989/90

[11] U. de Carlini, U. Villano
A Simple Algorithm for Clock Synchronization in Transputer Networks
Software Practice and Experience, Vol. 18, 4/1988