1 Motivation

Die vorliegende Ausarbeitung ist das Ergebnis der Projektstudie SCINET [SCI97], die vom Max-Planck-Institut für Plasmaphysik in Garching bei München finanziert und vom Autor am Lehrstuhl für Rechnertechnik und Rechnerorganisation der TU München durchgeführt wurde. Die Aufgabe des Max-Planck-Instituts liegt in der Erforschung der kontrollierten Kernfusion zum Zwecke einer späteren Energieerzeugung. Dies wird u.a. mit Hilfe von plasmaphysikalischen Großexperimenten wie z.B. der ASDEX-Upgrade-Fusionsanordnung bewerkstelligt. Die nächste geplante Anlage, W7-X, soll bereits ein über längere Zeit stabiles Plasma liefern, da sie auf supraleitenden Spulen beruhen wird, die über Wochen verlustfrei Strom tragen. Daraus ergeben sich jedoch ganz neue Anforderungen an die Steuerung, Regelung und vor allem Datenerfassung des Experiments hinsichtlich der aufgenommenen Datenraten- und mengen sowie der Latenzzeit während ihrer Übertragung. Die Studie SCINET soll die Eignung des Scalable Coherent Interface-Standards (SCI) für diese Anwendungszwecke untersuchen. Ihre Ergebnisse können auch für das Design des Datenerfassungssystems des „International Thermonuclear Experimental Reactor (ITER)“ verwendet werden, dem gemeinsamen Fusionsreaktorprojekt der Amerikaner, Europäer, Japaner und Russen, mit dessem Bau in den nächsten Jahren begonnen werden soll.

Datenerfassungssysteme für Fusionsreaktorexperimente haben eine lange Tradition [Preckshot86], [vBeken87], [McHarg87], [Balme88], [Nijman88], [Hertweck88], [Korteetal91], [vHaren93]. Stets wurde in deren Design versucht, die zum Zeitpunkt der Konzeption jeweils neueste Technologie einzusetzen, eingedenk der Tatsache, daß Entwurf, Bau und Betrieb einer Kernfusionsanlage ca. 2 Dekaden Zeit in Anspruch nehmen. Heutzutage stellt das Scalable Coherent Interface eine solche Technologie dar, was die Motivation für die Durchführung der Studie SCINET war.

Bei SCI sind folgende Fragen aus informatiktechnischer und ingenieurmäßiger Sicht für die genannten Echtzeitanwendungen von besonderem Interesse:

- Welchen Durchsatz und welche Latenz bietet SCI?
- Ist SCI für Echtzeitübertragungen geeignet?
- Gibt es Paketverluste bei SCI?
- Bei welcher Datenrate kommt ein SCI-Ring in Sättigung?
- Wie stark verändert sich der Durchsatz eines Empfängers bei einer zweiten, nebenläufigen Kommunikation auf demselben Ring?
- Wie hoch ist Durchsatz und Latenz eines SCI-Schalters mit z.B. 4 Ports?

- Haben SCI-Schalter bzw. Netze, die aus SCI-Schaltern aufgebaut sind, eine garantierte maximale Latenzzeit und einen garantierten minimalen Durchsatz?
- Welche Netztopologie ist für eine Datenerfassung am besten geeignet, bei der Tausende von räumlich entfernten Meßaufnehmern mit Abtastraten zwischen 10^2 und 10^7 Hz anzuschließen sind?
- Ist im Netzwerk Fehlertoleranz realisierbar?

Die Beantwortung dieser Fragen ist untrennbar mit der Lösung einiger Aufgabenstellungen aus den Gebieten der Rechnerarchitektur und des verteilten Rechnens verbunden. Das verteilte Rechnen hat in seinen aktuellen Ausprägungen „Cluster Computing“ (CC), „Networks of Workstations“ (NOWs), „Cluster of Workstations“ (COWs) und „Pile of PCs“ (PoPCs) das Potential, viele Anwendungen der klassischen Parallelrechner erheblich kostengünstiger ausführen zu können. Die Voraussetzung dafür ist, daß die überall verfügbaren und preisgünstigen Arbeitsplatzrechner und PCs effizient über Hochgeschwindigkeitsnetze, deren Leistung weit über der des konventionellen Ethernets liegt, gekoppelt werden können. SCI gilt als eine der Technologien, die eine Rechnerkopplung mit hoher Bandbreite, niedriger Latenz und einfachem Programmiermodell erlauben.

Die Leistungsbewertung des Verbindungsnetzwerks, so wie sie in dieser Studie vorgenommen wird, ist, neben den Programmiermodellen, die für eine transparente Nutzung der gekoppelten Rechner sorgen, ein wichtiger Beitrag zur Weiterentwicklung des verteilten Rechnens.

Ausgehend von den eingangs gestellten Fragen werden im Kapitel 2 "SCI-Netztechnologie" und im Kapitel 3 "Statische/dynamische SCI-Netze" die Grundlagen der SCI-Technologie und die daraus resultierenden Netze dargestellt. Im Kapitel 4 "Anwendungsbeispiel für SCI: Datenerfassungssystem" werden, ausgehend von dem konkreten Anwendungsfall eines SCI-basierten Datenerfassungssystems, SCI-Knoten, Ringe und Schalter hinsichtlich ihres funktionalen und zeitlichen Verhaltens allgemein modelliert. Den Abschluß dieses Kapitels bildet die Modellbildung eines SCI-basierten Datenerfassungssystems. Alle Modellierungen wurden in dem neuentwickelten SCINET-Simulator in Software implementiert.

Im Kapitel 5 "SCINET-Testsystem" werden Aufbau und Meßergebnisse eines SCI-Teststandes beschrieben, der zur Validierung von Modell und Simulator dient. Daran schließt sich das Kapitel 6 "SCINET-Simulator" an, in dem die Konzepte und Leistungsmerkmale des Simulators im Überblick dargestellt sind.

In den folgenden drei Abschnitten von Kapitel 7 "Analyse von SCI-Ringen", Kapitel 8 "Analyse von SCI-Schaltern" und Kapitel 9 "Analyse von SCI-Banyan-Netzen" werden per Simulation SCI-Ringe, -Schalter und -Netze in Bezug auf ihre Leistungsparameter von Durchsatz, Latenz und Paketverluste untersucht. Nach allen Untersuchungen werden jeweils Vorschläge zur Leistungssteigerung gemacht, die hinsichtlich ihrer Wirksamkeit im Allgemei-

nen und ihrer Eignung bei Echtzeitanwendungen im Speziellen bewertet werden.

Im Kapitel „Zusammenfassung und Bewertung“ sind die neuen wissenschaftlichen Erkenntnisse, die im Laufe der Studie gewonnen wurden, zusammengefaßt, und es wird eine Beurteilung über SCI bei Datenerfassungssystemen durchgeführt. Den Abschluß der Ausarbeitung bildet eine ausführliche Literaturliste, die als Grundlage für weitere Literaturrecherchen und Forschungsarbeiten dienen kann.

2 SCI-Netztechnologie

2.1 Einleitung

Bei der Entwicklung moderner Bussysteme wie des FASTBUS [IEEE89] oder des Futurebus+ [IEEE91a] wurde offenbar, daß Hochleistungsbussysteme zunehmend die Grenze des technisch und finanziell Machbaren erreicht haben. Um die deutlich sichtbar gewordenen Kosten- und Leistungsprobleme zu überwinden, wurde von Repräsentanten aus Industrie und Forschungseinrichtungen [Gustavso92] das Scalable Coherent Interface (SCI) entwickelt, das anschließend von ANSI und IEEE standardisiert worden ist [IEEE92].

SCI standardisiert mechanische, elektrische und logische Aspekte der Datenübertragung auf einem Ringmedium und erlaubt die flexible Verknüpfung vieler Ringe zu einem SCI-Netzwerk, das für räumlich verteilte Anwendungen geeignet ist. Im folgenden sollen hauptsächlich die logischen Aspekte der Standardisierung, d.h. die Datenformate und Protokolle von SCI erläutert werden.

Der Kern der logischen Aspekte von SCI stellen spezielle Protokolle dar, die einen gemeinsamen Adreßraum aufbauen und verwalten. Aus der Sicht eines Benutzers von SCI sieht diese Verbindungstechnologie aus wie ein Bus, sogar mit wahlweiser Cache-Koherenz zwischen Prozessoren. In Wahrheit werden jedoch beim Zugriff auf nicht-lokale Adressen Speicherinhalte von SCI-Schnittstellenkarten in Pakete verpackt, formatiert und über eine Kette von Punkt-zu-Punkt-Verbindungen dem Eigner der gewünschten Adressen zugestellt. Dieser transferiert auf die gleiche Weise die gewünschten Daten an den Anforderer zurück, ohne jede Benutzerintervention.

SCI fungiert somit wie ein *virtueller* Bus, ist jedoch nicht an dessen räumliche und elektrische Begrenzungen gebunden. SCI-Protokolle unterstützen alle üblichen Busoperationen; neben Lesen und Schreiben gibt es atomare Befehle zur Synchronisation bei Mehrfachzugriffen auf gemeinsam benutzte Adressen. Desweiteren sind Unterbrechungen (Interrupts) einzelner Netzteilnehmer und netzweite Rundrufe (Broadcasts) möglich.

Optional können von SCI automatisch die Cache-Inhalte aller an das Netz angeschlossenen Prozessoren miteinander abgeglichen werden, so daß Kopien von Variablen, die in anderen Caches gespeichert sind, systemweit auf dem neuesten Stand gehalten werden. Eine verteilte Verzeichnisstruktur, die dezentral verwaltet, wo welche Kopien existieren, erlaubt in gewissen Grenzen sogar die Skalierbarkeit der Systemgröße.

2.2 Warum SCI?

SCI ist durch seine hohe Bandbreite und niedrige Latenz bei der Datenübertragung prädestiniert für alle Anwendungen im Bereich des parallelen und verteilten Hochleistungsrechnens sowie der lokalen Netze. Differenziert man bei diesen beiden Gebieten nach den Kategorien eng und lose gekoppelter Systeme sowie Ein-/Ausgabesysteme, zeigt sich, daß SCI in allen drei Kategorien gegenüber anderen Hochleistungsnetztechniken Vorteile bietet:

- Eng gekoppelte Rechensysteme, zu denen hauptsächlich die symmetrischen Multiprozessoren (SMPs) gehören, erlauben eine kleinere Zahl von Prozessoren (≤32) miteinander zu koppeln. Als Programmiermodell werden meist gemeinsame Variable verwendet, die über einen globalen Speicher adressiert werden. SCI gestattet, mehrere SMP-Rechner zu einem Cluster zu vernetzen ohne daß dabei Bandbreite, Adreßräume oder Cache-Konsistenz verloren gehen. Die Grenzen der Skalierbarkeit werden hinausgeschoben bzw. überwunden. Verzichtet man auf die Cache-Konsistenz, kann der gemeinsame Adreßraum relativ leicht dadurch etabliert werden, daß die SMP-Rechner über ihre peripheren Bussysteme mittels SCI verbunden werden. Modifikationen des Betriebssystem sind nicht nötig.
- Lose gekoppelte Rechensysteme basieren auf lokalen Netzen wie dem Ethernet und verwenden Botschaftenaustausch als Programmiermodell. Ihre Interprozessorkommunikation basiert auf komplexen Protokollen wie TCP/IP, die per Software abgewickelt werden, so daß hohe Latenzen entstehen. Um dennoch effizient zu sein, werden auf lose gekoppelten Systemen grobgranulare, verteilte Anwendungen ausgeführt, deren Komponenten rel. selten und nur mittels langer Datenblöcke kommunizieren. SCI erlaubt mit seinen per Hardware interpretierten, auf Geschwindigkeit optimierten Protokollen feingranulare Anwendungen auszuführen, die kurzen und häufigen Datenaustausch erfordern. Gleichermaßen wird auch der Transfer langer Datenblöcke durch DMA-Einrichtungen effizient unterstützt.
- Ein-/Ausgabesysteme, die in der Regel busbasierend sind, erfordern aufgrund immer schnellerer Prozessortakte adäquate Geschwindigkeitssteigerungen. Hohe Durchsatzraten begrenzen jedoch, wie am Beispiel des PCI-Bus [Kau93] sichtbar geworden ist, die Zahl der Steckplätze auf ein Minimum. SCI erlaubt, mehrere Peripheriebusse entweder im selben Rechnergehäuse oder räumlich verteilt zusammenzuschalten, so daß die Ein-/Ausgabe skalierbar wird und über eine große Zahl von Steckplätzen verfügt.

Generell können mit SCI Multiprozessorsysteme, PCs, Arbeitsplatzrechner, Speicher oder periphere Geräte zu einem heterogenen System verbunden werden. Die zugrunde liegenden SCI-Protokolle sind hersteller- und geräteunabhängig. Auf der Ebene des physikalischen Transports von Daten sind sowohl

Kupfer- als auch Glasfaserkabel möglich. Das letztere erlaubt räumlich weit entfernte Systeme zu betreiben. Im Vergleich zu anderen Technologien zur Hochgeschwindigkeits-Datenübertragung, wie Fiber Channel, FDDI, HIPPI oder ATM, hat SCI einen breiteren Einsatzbereich bei gleichmäßig hohen Leistungsdaten, was in Bild 2.2.1 graphisch dargestellt ist. Die mit SCI erreichba-

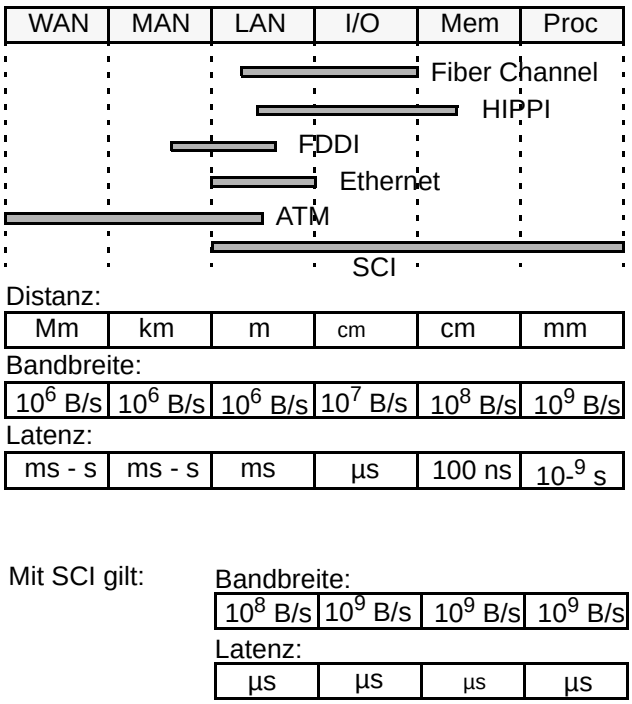

Bild 2.2.1: Vergleich Fiber Channel, HIPPI, FDDI, Ethernet, ATM und SCI.

ren Datenraten sind 500 MB/s auf dem Ring, bei Latenzen im µs-Bereich, die für den End-zu-End-Transfer zwischen den Speichern von Erzeugern und Verbrauchern von Daten gemessen werden.

Gleichwohl weist SCI auch einige systembedingten Schwächen auf, die herauszufinden und evtl. Verbesserungsvorschläge zu unterbreiten Gegenstand des zweiten Teils der vorliegenden Schrift darstellt.

2.3 Eigenschaften von SCI

Die SCI-Netztechnologie weist folgende Schlüsseleigenschaften auf:

- *Punkt-zu-Punkt-Verbindungen anstelle von gemeinsamen Busleitungen*. Die Datentransferrate bei Bussen ist durch die zu treibenden kapazitiven Busla-

sten und die prinzipiell nicht an die wechselnden Lastimpedanzen anpaßbaren Busleitungen limitiert. Unidirektionale SCI-Links hingegen können je ein Sender/Empfängerpaar mit impedanzmäßig idealen Übertragungsleitungen koppeln und vermeiden so Reflektionen der Signale. Die Transferrate wird nur durch die Höhe des zulässigen Ausgangsstroms des Senders begrenzt, der für jede Pegeländerung eine Empfängereingangskapazität umladen muß. Darüberhinaus erlauben multiple Punkt-zu-Punkt-Verbindungen, gleichzeitig mehrere Datentransfers im SCI-System durchzuführen.

- *Hohe Datenrate und niedrige Latenz*. SCI definiert Link-Geschwindigkeiten von bis zu 1 GB/s bei Latenzen im µs-Bereich. Zur weiteren Geschwindigkeitssteigerung sind alle SCI-Transaktionen in eine Request- und eine Response-Phase unterteilt, was es gestattet, den SCI-Ring während der Bearbeitungszeit einer Anforderung zur Übertragung anderer Transaktionen freizugeben (sog. Split Transactions). Schließlich sind bei jedem Anforderer multiple offenstehende Requests erlaubt, die vom Bearbeiter empfangen, gepuffert und pipeline-artig abgearbeitet werden können (Pending Transactions).
- *Garantierte Datenzustellung*. Jede SCI-Transaktion muß per Echopaket quittiert werden und wird bei negativem Echo automatisch vom Sende wiederholt. Zusätzlich kann Bandbreite beim Übertragungsmedium und Pufferplatz beim Empfänger reserviert werden, so daß die Daten des Senders bereits beim ersten Versuch zum Ziel übertragen und dort auch gespeichert werden können.
- *Gemeinsamer Adreßraum und Botschaftenaustausch*. SCI etabliert in jedem SCI-System einen 64 Bit breiten Adreßraum bestehend aus bis zu 64 K Teilnehmern mit jeweils 48 Bit lokalen Adressen. Das Lesen und Schreiben in diesem Adreßraum erfolgt transparent für den Benutzer und wird durch die SCI-Schnittstellen in Protokolle umgesetzt, die per Hardware ausgeführt werden. Zusätzlich können zusammenhängende Speicherbereiche über DMA transferiert werden, um so Botschaftenaustausch effektiv zu unterstützen.
- *Prozessorunabhängigkeit.* SCI-Protokolle sind unabhängig von konkreten Prozessorimplementierungen, so daß der Aufbau heterogener Systeme möglich ist. Individuelle Schnittstellenkarten setzen die Protokolle in Bussignale der jeweiligen SCI-Teilnehmer um.
- *Robustheit.* SCI-Protokolle gelten als robust. Pakete werden beispielsweise nach Ablauf eines „Verfallsdatums“ vom Ring entfernt. Sind trotzdem Fehler aufgetreten, liefern die Schnittstellenkarten über ihre Kommando- und Statusregister, die gemäß der IEEE CSR-Norm [IEEE91b] aufgebaut sind, detaillierte Fehlerinformationen.
- *Topologieunabhängigkeit*. SCI auferlegt nur geringe Restriktionen, wie die Netzteilnehmer verschaltet werden sollen. Insbesondere sind SCI-Schalter explizit im Standard vorgesehen, so daß skalierbare Systeme aufgebaut werden können. Werden Schalter, die für dynamische Netze gedacht sind, nicht verwendet, können auch alle statischen Topologien, die auf Ringen basieren,

wie Torus oder Hypercube, direkt mit Hilfe der Schnittstellenkarten realisiert werden.

- *Echtzeitfähigkeit*. Über spezielle Datenpakete, die bei Bussen einem Unterbrechungssignal entsprechen würden, können Teilnehmer individuell im Netz getriggert werden. Ebenso ist es möglich, alle Teilnehmer gleichzeitig über systemweit relevante Ereignisse mittels Broadcast zu unterrichten. Ferner gibt es Bandbreiteallozierungsmechanismen, die einen garantierten Zugriff auf das Übertragungsmedium sicherstellen und Pufferallozierungsmechanismen, die im Receive-Puffer des Empfängers Speicherplatz reservieren. Bei den Bandbreiteallozierungsmechanismen kann man zwischen einer prioritätsgesteuerten und einer fairen Zuteilung wählen. Im ersten Fall erhält das Paket mit der höchsten Priorität als erstes Zugang zum Übertragungsmedium, im zweiten Fall wird die Bandbreite gleichmäßig zwischen allen Teilnehmern eines Ringes aufgeteilt. Schließlich existiert eine globale Uhr mit 64 Bit Auflösung, die systemweit abgeglichen werden kann.

Kleine SCI-Netze bestehen aus einem einzigen Ring, der mindestens 2 und höchstens 64 K Teilnehmer haben kann. Große Netze enthalten eine beliebige Anzahl von Ringen, die über Schalter gekoppelt werden, allerdings darf die Gesamtzahl der Teilnehmer im System die Obergrenze von 64 K nicht übersteigen. Jeder Ring besteht aus einzelnen Segmenten, die jeweils eine Punkt-zu-Punkt-Verbindung zwischen einem Sender/Empfängerpaar etablieren. Auf jedem Segment kann simultan zu anderen Segmenten je ein Paket übertragen werden (Slotted Ring Protocol). Jeder Netzteilnehmer ist zugleich Sender und Empfänger, so daß auf jeder SCI-Schnittstellenkarte eine Punkt-zu-Punkt-Verbindung beginnt und eine endet. In Bild 2.3.1 ist die Beispielkonfiguration eines einzelnen SCI-Ringes gezeigt, die eine heterogenes System aus fünf Teilnehmern darstellt.

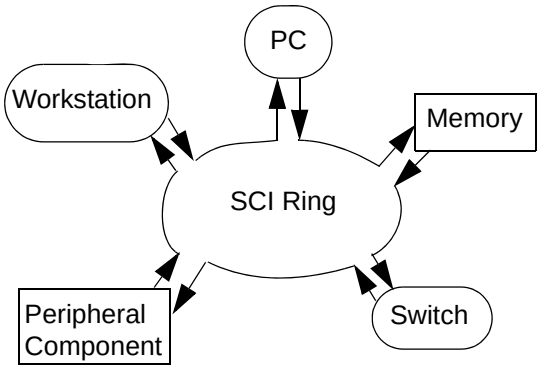

Bild 2.3.1: Beispielkonfiguration eines SCI-Ringes.

Größere Systeme verwenden zwei- oder Vier-Port Schalter [Dolphin94b], die als Brücke zwischen Ringen, als Router in einem statischen Netz oder als reiner Schalter in einem dynamischen Netz eingesetzt werden können. Die jeweilige

Funktion der Bausteine wird durch entsprechende Adreß- und Routing-Mechanismen im Schalter realisiert. In Bild 2.3.2 sind Beispiele für alle drei Möglichkeiten gezeigt, so wie sie in einem Doppelringsystem, einem Torus und einem mehrstufigen Netz auftreten.

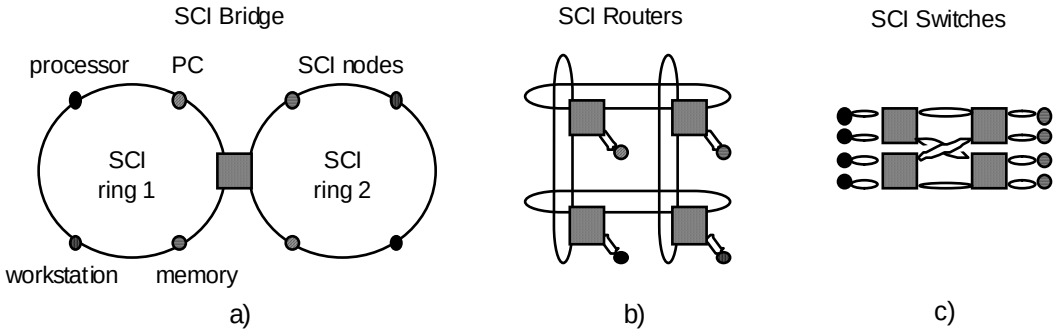

Bild 2.3.2: SCI-Schalter als Brücke (a), Router (b) und Netzschalter (c).

Innerhalb jedes SCI-Knotens wird die Verbindung zum Ring über eine Schnittstelle hergestellt, deren genereller Aufbau von IEEE genormt wurde [IEEE92]. Die wesentlichen Komponenten jeder Schnittstelle sind ein Adreßdekoder, ein Sende- und Empfangspuffer, ein Bypass-Fifo und ein Ausgangsmultiplexer. Ankommende SCI-Pakete werden hinsichtlich ihrer Zieladresse mit der Adresse des SCI-Knotens verglichen und bei Übereinstimmung dem Ring entnommen und im Empfangspuffer zwischengespeichert, bis der der Schnittstelle nachgeschaltete Teilnehmer sie weiterverarbeitet. Bei Nichtübereinstimmung der Adressen wird das Paket über den Bypass-Fifo und den Ausgangsmultiplexer auf dem nächsten Ringsegment wieder ausgegeben. Der Sendepuffer dient dazu, Pakete vom Teilnehmer zwischenzuspeichern, bis der Ausgangsmultiplexer frei ist. Der Multiplexer „mischt“ Pakete vom Bypass-Fifo und vom Sendepuffer zu einem gemeinsamen Datenstrom und gibt diesen auf dem Ring aus. Bild 2.3.3 zeigt den schematischen Aufbau einer SCI-Schnittstelle. Bemerkenswert ist, daß die beiden Pakettypen, die bei SCI unterschieden werden (Request und Response), getrennte Speicherbereiche im Sende- und Empfangspuffer haben. Dies ist notwendig, um eine gegenseitige Verklemmung (Deadlock) mehrerer Teilnehmer zu vermeiden, die wechselseitig auf freie Puffer warten.

SCI basiert darauf, daß Pakete auf einem unidirektionalen Ring kreisen. An jedem SCI-Knoten, der passiert wird, erfolgt eine Regeneration der elektrischen Signale, aus denen das Paket besteht. Basiert das Übertragungsmedium auf Kupferkabeln, werden vom Paket 16 Bit Daten parallel übertragen. Zur Steuerung des Empfängers dient ein Clock- und ein Flag-Signal, das Paketbeginn und -ende anzeigt, so daß insgesamt 18 Signalleitungen zwischen Sender und Empfänger verlaufen. Die Leitungen haben entweder ECL- oder LVDS-Pegel. Bei Glasfasern wird in der Regel bitseriell übertragen, jedoch ist auch eine 10-Bit breite Übertragung auf Glasfaserflachbandkabeln möglich.

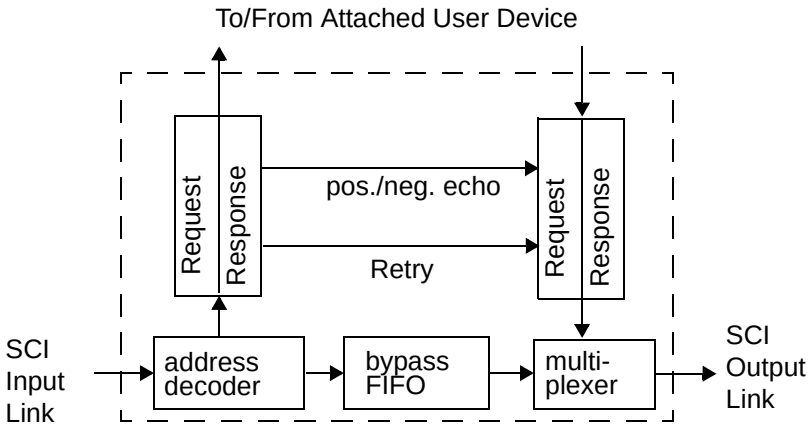

Bild 2.3.3: Schematischer Aufbau einer SCI-Schnittstelle.

2.4 SCI-Operationen und Datenformate

SCI-Operationen werden allgemein als Transaktionen bezeichnet. Jede Transaktionen wie z.B. Lesen oder Schreiben einer Speicherzelle, besteht aus zwei Phasen (Request und Response), die jede für sich positiv quittiert werden müssen, so daß insgesamt ein Vierphasenprotokoll entsteht. Die Quittungen werden in Form von (kurzen) Echopaketen verpackt und wandern einmal in Umlaufrichtung des Ringes, bis sie zum Sender des Request- oder Response-Pakets zurück gelangen. In Bild 2.4.1 ist der Ablauf einer Transaktion graphisch dargestellt. Wichtig ist, daß zwischen Request- und Response-Phase ein belie-

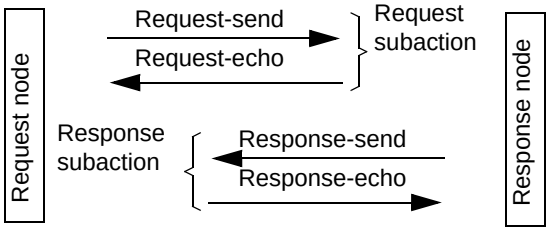

Bild 2.4.1: Standardisierte SCI-Transaktion.

big langer Zeitraum liegen kann, währenddessen andere Transaktionen vom SCI-Medium übertragen werden können. Diese „Split Transaction“-Betriebsweise ist bereits von Hochleistungsbussystemen wie dem Futurebus+ bekannt. Insgesamt können von einem SCI-Knoten bis zu 64 Anforderungen (Requests) ausgeschickt werden, bevor die erste Antwort (Response) am Knoten eingetrof-

fen sein muß. Durch Split Transaction wird die Transferkapazität von Ring und Schnittstelle besser ausgenutzt, da überlappend zur Bearbeitung einer Anforderung beim Empfänger, der Sender neue Pakete erzeugen kann (Pipeline-Betrieb). Stillstandszeiten werden so reduziert.

Ist der Empfangspuffer eines Knotens voll, kann eine an ihn gerichtete Anforderung oder Antwort nicht mehr zwischengespeichert werden und der Knoten gibt dem Sender ein negatives Echo zurück. Daraufhin wird automatisch das abgelehnte Paket vom Sender erneut auf den Ring geschickt (Retry), solange bis ein positives Echo erhalten wird. Eine größere Zahl von Retry-Paketen kann dazu erforderlich sein.

In Bild 2.4.2 ist das Format von Request-, Response- und Echopaketen dargestellt für den einfacheren Fall, daß Cache-Konsistenz nicht erforderlich ist.

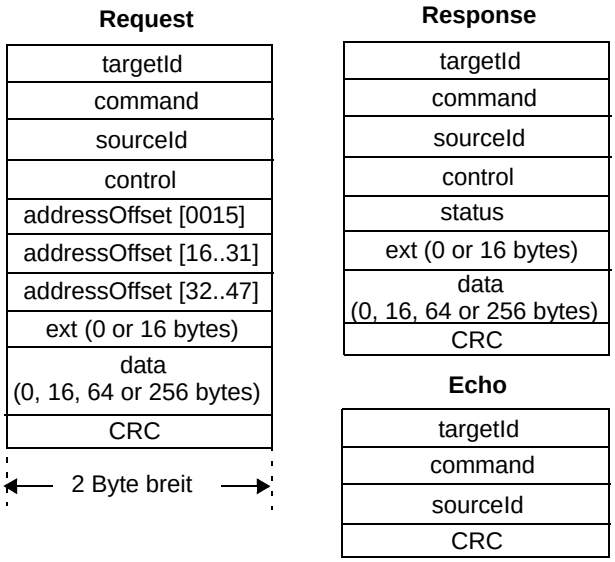

Bild 2.4.2: Format von Request-, Response- und Echopaketen.

Das Anforderungspaket enthält als erstes Datum die 16-Bit-Knotenadresse, so daß ein Adreßdekoder, der das Paket entgegennimmt, während des Empfangs nachfolgender 2-Byte-Worte desselben Pakets bereits dessen Adresse dekodieren kann. Das an die Adresse anschließende Kommandofeld beinhaltet beim Request-Paket eines von insgesamt 48 SCI-Kommandos sowie verschiedene Steuerbits. Nach dem Kommandofeld folgt im Paket die Herkunftsadresse, die notwendig ist, damit das zugehörige Echopaket zum Sender zurückfindet. Das vierte Wort im Paket ist ein 16 Bit-Kontrollfeld mit verschiedenen Verwaltungsangaben, die im nächsten Abschnitt erläutert werden. Im Anschluß an das Kontrollfeld werden im SCI-Paket 3 Worte mit Adreßversätzen (Offsets) übertragen, die einen 64-Bit-Adreßraum ermöglichen. Schließlich kann optional ein Erweiterungswort (Ext) gesendet werden sowie die eigentlichen Daten des Request-Pakets. Das Paketende wird von einer CRC-Prüfsumme gebildet.

Das Anwortpaket ist ähnlich wie das Anforderungspaket aufgebaut, jedoch unterscheiden sich beide hinsichtlich der Anzahl ihrer Datenworte. Ein Lesebefehl beispielsweise enthält im Anforderungspaket keine Daten, während die zugehörige Antwort bis zu 128 Datenworte enthalten kann. Weiterhin gibt es beim Antwortpaket keine Adreßversätze (Offsets), dafür jedoch ein Statuswort, das Auskunft über den Erfolg der Transaktion gibt.

Die Echopakete sind mit insgesamt 8 Byte relativ kurz. Sie enthalten in ihrem Kommandofeld zusätzliche Statusinformationen, die ein Paketsender zur Abwicklung der SCI-Protokolle benötigt. Im folgenden werden die einzelnen Bitfelder der Pakete näher erläutert.

2.4.1 Das Kommandowort eines SCI-Request-Pakets

Das Kommandowort des Anforderungspakets besteht aus Bitfeldern, die die folgende Bedeutung haben (in der Reihenfolge ihrer Bitwertigkeit aufgelistet, MSB zuerst):

- *mpr* (Maximum Ringlet Priority): Bei der Erzeugung eines Request-Pakets wird dieses 2-Bit-Feld vom Sender auf den Wert 0 gesetzt. Anschließend wird es von anderen Knoten modifiziert, um die höchste im Ring vorkommende Priorität anzuzeigen.
- *spr* (Send Priority): Damit kann vom Sender eine von vier Prioritäten für das Paket ausgewählt werden. Die Sendepriorität wird anhand der eigenen Transaktionspriorität (tpr) festgelegt sowie der anderer Transaktionen, die zur selben Zeit vom Sender abgewickelt werden.
- *phase*: Dieses 2-Bit-Feld wird von der Pufferallozierungs-Hardware verwendet, um auch in dem Fall die Paketzustellung zu gewährleisten, wenn ein negatives Echo empfangen wurde. In diesem Fall wird automatisch von der SCI-Schnittstelle eine Wiederholung der Paketsendung durchgeführt, die als Retry bezeichnet wird. Das Phase-Feld hat sowohl für normale Request und Response-Pakete als auch für Retry-Pakete eine auswertbare Bedeutung, wobei ein nicht-Retry-Paket zwei und ein Retry-Paket vier mögliche Zustände (Phases) aufweisen kann. Der jeweilige Zustand wird vom Phase-Feld des zuvor empfangenen negativen Echopakets bestimmt. Das Phase-Feld eines nachfolgenden Retry-Pakets wird von der Pufferallozierungs-Hardware des Empfängers ausgewertet, und je nach Empfängerzustand wird das Retry-Paket akzeptiert oder nicht, so daß u.U. mehrere Retry-Paketwiederholungen durchgeführt werden müssen. Die vier Zustände des Phase-Felds sind:

  · *NOTRY*: Ein normales Request und Response-Paket wurde abgeschickt mit der Hoffnung, daß ein freier Pufferplatz beim Empfänger vorhanden ist. Dieser Zustand kann auch von einem Retry-Paket angenommen werden, allerdings nur, wenn zuvor ein negatives Echo mit Phase-Feld = BUSY_N empfangen wurde.

· *DOTRY*: Ein normales Request und Response-Paket wurde abgeschickt mit der Bitte um Empfangspufferreservierung für ein nachfolgendes Retry-Paket, falls der Empfänger im Augenblick keinen Speicherplatz haben sollte. Dieser Zustand kann auch vom ersten Retry-Paket angenommen werden, wenn zuvor negatives Echo mit Phase-Feld = BUSY_D empfangen wurde. Für ein evtl. notwendig werdendes zweites Retry-Paket wird mit DOTRY ein Empfangspufferplatz reserviert. DOTRY ist der Zustand, der sich für das erste Retry-Paket ergibt, wenn das vorangegangene normale Request oder Response-Paket mit phaseFeld = NOTRY abgeschickt worden ist und anschließend mit einem BUSY_D-Echo geantwortet wurde.

· *Retry_A*: Es handelt sich um eine Paketwiederholung, nachdem ein negatives Echo-Paket mit BUSY_A-Status empfangen wurde und

· *Retry_B:* Paketwiederholung nach Empfang von einem negativen Echo-Paket mit BUSY_B-Status.

Weitere Bitfelder im Kommandowort sind:

- *old*: Mit diesem Bit können Echopakete identifiziert werden, die bereits einmal komplett im Ring gekreist sind. Sie werden von einem speziellen Ringüberwachungsknoten, dem „Scrubber“ entfernt.
- *ech:* (Echo). Dieses Bit ist bei Anforderungspaketen auf 0 gesetzt, um anzuzeigen, daß es sich nicht um ein Echo handelt.
- *eh:* (Extended Header). Das eh-Bit im Kommandofeld zeigt an, ob eine 16-Byte-Erweiterung nachfolgt oder nicht.
- *cmd:* (command). Im 7-Bit-Kommandofeld sind ca. 48 SCI-Kommandos kodiert, die in Tabelle 2.4.1 dargestellt sind, sowie bei einigen Kommandos zusätzliche Unterbefehle, die als Subactions bezeichnet werden.

2.4.2 Das Kommandowort eines Response-Pakets

Die Antwort zu einem Anforderungspaket wird als Response-Paket bezeichnet und vom Empfänger der Anforderung abgeschickt, nachdem das Request-Paket oder eines seiner nachfolgenden Retry-Pakete akzeptiert, d.h. im Empfangspuffer eingespeichert, und vom SCI-Knoten die Antwort formuliert worden worden ist. Das Kommandowort der Response ist vom Format her bis auf das 7-Bit-Kommandofeld identisch mit dem Kommandowort der Anforderung. Beim Response-Paket hingegen kodiert das Kommandofeld zusätzlich die Länge der Antwort, wobei 0/16/64 oder 256-Byte an Länge möglich sind.

2.4.3 Das Kommandowort eines Echo-Pakets

Die Kommandoworte der Echos (Request Echo und Response Echo) sind ähnlich zum Kommandowort des Anforderungspakets. Bei ihnen ist jedoch das

Name	Beschreibung
readsb	read selected byte
WRITESB	write selected byte
nread256/64	non-coherent read
nwrite16/64/256	non-coherent write
mread00/64	coherent read
mwrite16/64	coherent write
cread00	cache control
cread64	cache-to-cache-read
cwrite64	cache-to-cache-write
smovesb	start broadcast selected-byte move

Name	Beschreibung
rmovesb	resume broadcast selected-byte move
dmovesb	directed selected-byte move
smove00/16/64/256	start broadcast 00/16/64/256 -byte move
rmove00/16/64/256	resume broadcast 00/16/64/256 -byte move
dmove00/16/64/256	directed 00/16/64/256-byte move
event00	clockStrobe signal
event16/64/256	≤16/64/256 events
xread64/256	status and 64/256-byte return

Tabelle 2.4.1: Zusammenfassung der SCI-Kommandos.

ech-Bit gesetzt, um anzuzeigen, daß es sich um ein Echo handelt. Weiterhin ist anstelle des 7-Bit-Kommandofeldes die Kopie einer 6-Bit-Transaktionsnummer untergebracht, die bereits zuvor in den Kontrollfeldern von Request und Response-Paket übertragen worden ist. Im verbleibenden freien Bit (res) des 7-Bit-Feldes wird unterschieden, ob es sich um das Echo für Request- oder Response handelt.

Alle anderen Bits in den Echopaketen sind formal analog zu korrespondierenden Bitfeldern des Request-Kommandowortes, jedoch sind ihre Inhalte verschieden von denen des Requests. Beispielsweise enthält das mpr-Feld des Echos (mpr=Maximum Ringlet Priority), die Priorität, die die das gesendete Paket zu dem Zeitpunkt hatte, als das Echo für das empfangene Paket erzeugt wurde. Ähnlich informiert das Phase-Feld des Echos über den Zustand des Empfängers, nachdem dort ein Request-Paket eingetroffen ist.

Das Phase-Feld eines Echos hat zwei verschiedene Bedeutungen, je nachdem, ob es sich um eine positive oder eine negative Quittung (Busy/Non-Busy Echo) handelt. Im ersten Fall (Non-Busy Echo) gibt es zwei verschiedene Zustände:

- *Done*: Das gesendete Paket wurde in den Receive-Puffer des Empfängers eingespeichert. Der Sender braucht keine Paketwiederholung durchzuführen

und kann seine lokale Kopie des Pakets löschen.

- *None*: Das gesendete Paket hatte eine SCI-Zieladresse, die es im Ring nicht gibt, so daß kein Empfänger angesprochen wurde. In diesem Fall wird das Phase-Feld des Echos nicht von einem regulären Ringknoten erzeugt, sondern von dem „Scrubber“ des Rings.

Im zweiten Fall, d.h. bei negativem Echo (Busy Echo), steuert das Phase-Feld des Echos die Phase des nachfolgenden Retry-Pakets. In den vier möglichen Zuständen, die das Phase-Feld annehmen kann, fließt die momentane Situation des Empfängers hinsichtlich seines Pufferfüllgrades ein. Der genaue Mechanismus des Wechselspiels zwischen den Phase-Feldern von negativem Echo und nachfolgendem Retry-Paket ist im Kapitel 2.5.3 "Pufferallozierung" beschrieben. Die vier Zustände des Echo-Phase-Feldes sind:

- *BUSY_N*: Dieser Zustand bedeutet, daß das nachfolgende Retry-Paket in seinem Phase-Feld den Zustand NOTRY aufweisen soll. NOTRY heißt, daß für das Retry-Paket kein Puffer reserviert wird.
- *BUSY_D*: Das nachfolgende Retry-Paket soll den Zustand DOTRY haben. Wenn das Retry-Paket ebenfalls abgelehnt werden sollte, wird vom Empfänger Pufferplatz für die Einspeicherung eines erneuten Retry-Pakets reserviert.
- *BUSY_A*: Es wurde Platz im Receive-Puffer für ein Retry-Paket mit Zustand Retry_A reserviert.
- *BUSY_B*: Es wurde Platz im Receive-Puffer für ein Retry-Paket mit Zustand Retry_B reserviert.

2.4.4 Das Kontrollfeld eines SCI-Pakets

Die Kontrollfelder von Request- und Response-Paket enthalten verschiedene Verwaltungsangaben im jeweils gleichen Format und mit gleicher Bedeutung. Diese sind im einzelnen (in der Reihenfolge ihrer Bitwertigkeit aufgelistet):

- *trace*: Dieses Bit dient zur Fehlersuche und zur Protokollierung. Ein Paket mit gesetztem Trace-Bit kann dazu verwendet werden, in einem Knoten eine Zusatz-Hardware anzustoßen, den Paketkopf zusammen mit einem Zeitstempel in ein Logbuch einzutragen. Ein nachfolgende Analyse des Logbuchs gibt Auskunft darüber, wann welches Paket den Knoten passierte.
- *timeOfDeath*: Das Paketverfallsdatum wird in Gleitkommadarstellung mit 2 Bit Mantisse und 5-Bit-Exponent angegeben. Eine Mantisse mit dem Wert 0 heißt, daß das Paket nicht verfällt.
- *tpr*: Die Priorität der Transaktion kann in vier verschiedenen Abstufungen spezifiziert werden. Sie wird vom Sender des Request-Pakets festgelegt und von den Bandbreiteallozierungsprotokollen verwendet. Darüberhinaus wird die Transaktionspriorität vom Sender des Request-Pakets benötigt, um die eigentliche Sendepriorität spr festzulegen, die im Kommandowort des Pakets steht.

- *transactionid*: Anhand der Transaktionsnummer, die man sich an die Herkunftsadresse angehängt denken muß, kann man sehen, um die wievielte offenstehende Transaktion es sich bei dem betreffenden Knoten handelt. Maximal sind 63 offenstehende Transaktionen zulässig, da 6 Bit zur Zählung zur Verfügung stehen.

2.4.5 Das Statuswort eines SCI-Pakets

Bei Response-Paketen wird ein Statuswort mit übertragen. Es besteht aus den Feldern sStat, res, vStat und cStat. Diese Felder bedeuten im einzelnen:

- *sStat* (Summary Status): Die Statuszusammenfassung enthält in einem 4-Bit-Feld Informationen darüber, ob die Transaktion erfolgreich beendet wurde. Dabei wird unterschieden, ob das Response-Paket von einem SCI-Schalter oder von einem Nicht-Schalter-Knoten abgeschickt wurde. SCI-Schalter haben eine besondere Bedeutung, da sie sind nicht die eigentlichen Endabnehmer einer Anforderung, sondern nur Zwischenstationen sind, weshalb man sie auch als „Agenten“ bezeichnet. Entsprechend ist auch ihre Statuszusammenfassung im Kontrollfeld leicht unterschiedlich. Für deren genaue Beschreibung wird auf den IEEE-Standard [IEEE92] verwiesen.
- *res* (Reserved): Dieses Bit ist momentan noch nicht verwendet, sondern steht für zukünftige Erweiterungen zur Verfügung.
- *vStat* (Vendor Status): Das 3-Bit-vStat-Feld steht dem Hersteller eines SCI-Knotens für eigene Statusinformationen offen, die nicht unter den Rahmen der IEEE-Norm fallen.
- *cStat* (Coherence Status): Das 8-Bit-cStat-Feld wird nur benötigt, wenn Cache-Kohärenz verlangt ist. In diesem Fall werden dem Statuswort des Response-Pakets noch zwei weitere Worte angehängt, die einen Vorwärtszeiger und einen Rückwärtszeiger in einer verketteten Liste darstellen. Die Liste dient zur Verwaltung der Information, welche Kopien von Variablen in welchen Cache-Speichern existieren.

2.4.6 Die übrigen Felder eines SCI-Pakets

- *addressOffset*: Bei den Anforderungspaketen ergeben die Adreßversätze zusammen 48 Adreßbit, die der Adressierung innerhalb eines Knotens dienen. Davon sind einige Adreßkombinationen aufgrund der sog. Command/Status-Architektur des IEEE bereits belegt [IEEE91b].
- *Ext* (Extension): Das Erweiterungsfeld des Anforderungspakets wird von den Cache-Kohärenzprotokollen benötigt und ist nur bei Cache-Kohärenz vorhanden.
- *Data*. Das Datenfeld des Request- und Response-Pakets kann 0, 16, 64 oder

256 Byte aufnehmen. Bei Anforderungspaketen ist es im Lesefall und bei einigen anderen Kommandos leer (Responseless Transactions).

- *CRC*. Am Ende aller Pakettypen folgt die Prüfsumme, die anhand eines Generatorpolynoms Einzelfehler im Paket korrigiert und Doppelfehler erkennt. Die Prüfsumme wird schritthaltend mit dem Empfang eines Pakets berechnet und sollte fertiggestellt sein, sobald die gesendeten 16 CRC-Bits eingelesen werden. Danach kann die vom Sender mitgelieferte mit der vom Empfänger berechneten Prüfsumme verglichen werden.

2.4.7 Idle-Symbole

Idle-Symbole füllen auf dem Ring die Zeit zwischen zwei Paketübertragungen. Sie werden erzeugt, sobald ein Request- oder Echopaket dem Ring entnommen wird und stellen einen Vor- und Nachspann zu den Datenpaketen dar. Ihre Funktion dient der besseren elektrischen Abtastbarkeit der Datenleitungen und darüberhinaus der dynamischen Bandbreitevergabe. Sie sind 16 Bit breit und werden auf den 16 parallelen Datenleitungen von SCI in einem Takt übertragen. Acht der 16 Idle-Bits sind Nutzbits, die im Einzelnen die folgende Bedeutung haben:

- *ipr* (Idle Priority): enthält eine von vier möglichen Sendeprioritäten der Request-Paketen. Das 2-Bit-Feld dient dazu, die momentan beste Schätzung der höchsten im Ring vorkommen Sendepriorität (Maximum Ringlet Priority) zu verteilen. Die Höhe der Sendepriorität wird über die tpr-, spr- und mpr-Felder der Request-Pakete ermittelt.
- *ac* (Allocation Count): Dieses Bit ändert jedesmal seinen Wert, wenn alle im Ring befindlichen Knoten Gelegenheit hatten, ein Paket abzusenden. Es stellt somit einen 1-Bit-Zähler dar. Das ac-Bit wird dazu verwendet, bei einem Empfänger diejenigen Pufferreservierungen zu löschen, die für zu erwartende Paketwiederholungen vorgenommen worden sind, die jedoch vom Sender niemals durchgeführt wurden.
- *cc* (Circulation Count): Dieses Bit ändert jedesmal seinen Wert, wenn das Idle-Paket einmal komplett im Ring gekreist ist. Das Bit wird verwendet, um verloren gegangene Echo-Pakete und Go-Bits aufzuspüren.
- *lt* (Low Type): Zeigt an, daß es sich um ein Idle-Paket handelt, das für die Bandbreiteallozierung nach dem low-Pass-Protokoll zuständig ist.
- *lg* (Low Go): Erlaubt einem Knoten, dessen Bandbreite nach dem low-Pass-Protokoll geregelt wird, zu senden.
- *hg* (High Go): Erlaubt einem Knoten, dessen Bandbreite nach dem high-Pass-Protokoll geregelt wird, zu senden. Die Low-Go- bzw. High-Go-Bits werden allgemein auch als Go-Bits bezeichnet.
- *old*: Mit diesem Bit können Idle-Symbole identifiziert werden, die bereits einmal komplett im Ring gekreist sind. Sie werden von einem speziellen

Ringüberwachungsknoten, dem „Scrubber“ entfernt.

Innerhalb der Menge der Idle-Symbole gibt es eine Teilmenge, die als konsumierbare Idle-Symbole bezeichnet wird. Idle-Symbole sind dann konsumierbar, wenn gilt: lt == 1 oder ipr == 0.

2.5 SCI-Protokolle

Die SCI-Protokolle zur Bandbreite- und Pufferallozierung sind neben der schnellen Übertragung der Daten auf der physikalischen Ebene der Schlüsselfaktor zur hohen Leistungsfähigkeit dieser Technologie. Eine Leistungsbewertung von SCI erfordert deshalb die eingehende Kenntnis der Protokolle. Leider gehört deren Darstellung zu den komplexesten Kapiteln im IEEE-Standard, so daß hier eine eigene Präsentation gewählt wird. Sie beruht auf einem Satz von Regeln und auf endlichen Automaten, die es erlauben, die Spezifikation effizient in eine Software-Implementierung umzusetzen. Daneben sei noch auf die Tatsache verwiesen, daß zum IEEE-Standard von SCI eine C-Code-Spezifikation der SCI-Protokolle gehört [IEEE92b].

2.5.1 Bandbreiteallozierung

SCI-Ringe bestehen aus Segmenten, die elektrisch voneinander unabhängig sind, deshalb können auf jedem SCI-Ring simultan soviele Pakete unterwegs sein, wie der Ring Segmente hat. Der Zugang eines Knotens zu seinem Ringsegment wird von Bandbreiteallozierungsprotokollen geregelt. Man kann zwischen zwei alternativen Protokollen wählen, die unterschiedliche Bandbreitevergabestrategien realisieren: zum einen kann die Bandbreite gleichmäßig auf alle Knoten aufgeteilt werden (Pass-Protokoll), zum anderen können einzelne Knoten priorisiert werden (Low/High-Protokoll). Die Knoten mit der höchsten Priorität erhalten dabei und den größten Teil der Übertragungskapazität. Beim Low/High-Protokoll ist jedoch garantiert, daß hochpriore Knoten nicht den Ring gänzlich für sich beanspruchen, vielmehr steht Knoten niedrigerer Priorität ca. 10% der Ringbandbreite fest zur Verfügung, die wie beim Pass-Protokoll gleichmäßig unter diesen Knoten aufgeteilt wird.

Die Bandbreiteallozierung beruht auf einer Erlaubnis zum Senden, die ähnlich wie bei Token-Ring-Systemen reihum weitergereicht wird. Bei SCI ist das Token in Form eines „Go“-Bits realisiert, das in einem Idle-Symbol von Knoten zu Nachbarknoten wandert. Hochpriore Knoten empfangen im Vergleich zu niederprioren Knoten öfter ein Idle-Symbol mit gesetztem Go-Bit und können dadurch entsprechend häufiger senden.

Um die Sendeaktivierung von hochprioren Knoten von der von niederprioren Knoten zu unterscheiden, gibt es Low-Type- und High-Type Idle-Symbole, die

beide ein Go-Bit enthalten (Low-Go bzw. High-Go Bit). Das Go-Bit ist innerhalb seiner Prioritätsklasse für die Vergabe des Bandbreiteanteils zuständig, der der Prioritätsklasse zusteht. Die Mechanismen, die die Sendeaktivierung in den beiden Prioritätsklassen bewirken, werden als Low- bzw. High-Protokoll bezeichnet. Zusammen bilden sie das Low/High-Protokoll, das die effizientere, aber auch komplexere Alternative zum rel. einfachen Pass-Protokoll darstellt.

Ergänzend zur Bandbreitevergabe für die Sender existiert noch eine Puffervergabe bei den Empfängern mit dem Ziel, jedem Sender einen Pufferplatz im Receive-Puffer zu reservieren. Die Puffervergabe wird über ein eigenes Protokoll realisiert, das im Kapitel 2.5.3 "Pufferallozierung" erläutert wird.

Die Einteilung, welcher Knoten zu welcher Prioritätsklasse gehört, wird bei SCI dezentral und voll dynamisch vorgenommen. Die Dezentralisierung bewirkt einerseits, daß kein Engpaß existiert, erhöht aber andererseits den Aufwand bei der Implementierung. Dynamische Prioritätsklassen bedeuten, daß sich die Einteilung laufend ändern kann, was die Komplexität der Implementierung weiter steigert. Grundlage der Einteilung sind die verschiedenen Prioritätsangaben, die in den 2-Bit-Feldern tpr, mpr, spr und ipr aller Request-, Response und Echopakete bzw. Idle-Symbole enthalten sind.

Die 2-Bit-Felder erlauben, bis zu vier verschiedene Prioritäten P1-P4 zu unterscheiden. Das bedeutet jedoch nicht, daß in einem SCI-Ring zu einem gegebenen Zeitpunkt tatsächlich Pakete aller Prioritätsstufen existieren müssen. Vielmehr wird es ein oder mehrere Pakete geben, die momentan die höchste Prioritätsstufe haben, beispielsweise P3. Diese P3-Pakete erhalten dann über das High-Protokoll ca. 90% der Ringbandbreite zugeteilt, während alle anderen Pakete niedrigerer Priorität, die verbleibenden 10% gemäß des Low-Protokolls gleichmäßig unter sich aufteilen müssen.

Knoten, die eine Transaktion ausführen wollen, beginnen damit, dem Anforderungspaket (Request) nach eigenem Ermessen eine Transaktionspriorität (tpr) zu verleihen. Das zur Transaktion gehörende Request-Paket bekommt normalerweise diesen Wert als Sendepriorität (spr) mit auf den Weg. Im Rahmen des Mechanismus der *Prioritätsvererbung* kann jedoch vom Knoten die Sendepriorität temporär hochgesetzt werden, um so indirekt Blockierungen anderer Pakete auf dem Ring aufzulösen. Die höchste im Ring vorhandene Paketpriorität ist im mpr-Feld jedes Paktes gespeichert, während das ipr-Feld im Idle-Symbol die beste Schätzung dieses Wertes darstellt.

2.5.2 Low-Protokoll

Bei den Link-Controllern von Dolphin wird das Low-Protokoll verwendet und deshalb soll es hier näher beschrieben werden. Es ist Teil des Low/High-Protokolls und zum Pass-Protokoll sehr ähnlich. Der Unterschied zwischen beiden ist u.a., daß beim Pass-Protokoll die gesamte Ringbandbreite gleichmäßig aufgeteilt wird, während beim Low-Protokoll nur der Teil fair vergeben wird, den das High-Protokoll übriggelassen hat. Zu beachten ist ferner, daß in den Dolphin-

schen Link-Controllern das High-Protokoll bislang nicht implementiert ist, so daß ein komplettes Low/High-Protokoll nicht gefahren werden kann.

Die Bandbreitevergabe des gemeinsamen SCI-Rings erfolgt beim pass- wie beim Low-Protokoll so, daß die verfügbare Ringkapazität gleichmäßig auf alle Sender aufgeteilt wird, sofern deren Senderate d.h. deren Bandbreitebedarf gleich groß ist. Haben zwei Sender S_1 und S_2 unterschiedlichen Bandbreitebedarf b_1 bzw. b_2 mit $b_1 > b_2$, dann wird die Ringbandbreite im Verhältnis b_1/b_2 vergeben. Der Mechanismus für das Low-Protokoll basiert auf den folgenden Grundregeln für Paketsender:

- Sobald ein sendewilliger Sender ein Low-Type Idle-Symbol mit gesetztem Low-Go-Bit empfängt, hat er die Erlaubnis zum Senden („low transmission enabled“).
- Zu Beginn der Sendung, die als aktive Übertragungsphase bezeichnet wird, gibt er das empfangene Idle-Symbol wieder aus und legt eine Kopie davon im knoteninternen savedIdle-Speicher ab. Danach wird ein Paket aus seinem Ausgabe-Fifo auf den Ring gegeben.
- Wird während einer Paketsendung genau ein konsumierbares Idle-Symbol empfangen, wird das Low-Go-Bit dieses Symbols im knoteninternen save.lg-Flag, das Teil des idleMerge-Blocks ist, gespeichert. Die anderen Bits dieses Symbols werden ebenfalls im idleMerge-Blocks abgelegt.
- Wird während einer Paketsendung mehr als ein konsumierbares Idle-Symbol empfangen, werden diese Symbole zu einem einzigen Low-Type Idle verschmolzen. Das Resultat der verschmolzenen Low-Go-Bits wird im save.lg-Flag gespeichert. Das save.lg-Flag wird gesetzt, sobald ein Idle-Symbol mit gesetztem Low-Go-Bit konsumiert wird. Der Vorgang wird auch als Idle-Vernichtung bezeichnet und dient zur Verringerung der Latenz auf dem Knoten.
- Während einer Paketsendung empfangene, „nicht-konsumierbare“ Idle-Symbole werden unter Mißachtung der Tatsache, daß sie nicht konsumierbar sind, zusammen mit den konsumierbaren Idle-Symbole verschmolzen, allerdings wird dabei jedesmal ein „Schuldenzähler“ erhöht. Der Sinn dieser Maßnahme dient ebenfalls der Latenzverringerung.
- Während einer Paketsendung empfangene, normale Paketsymbole werden im Bypass-Fifo gespeichert, bis der SCI-Link-Ausgang frei ist. Der Bypass-Fifo ist mindestens so groß, wie es dauert, das Sendepaket auszugeben. Aus Latenzgründen ist er jedoch nicht größer als das längste bei SCI vorkommende Paket.
- Unmittelbar nach der Paketsendung wird vom Sender der savedIdle-Speicher ausgegeben. Bei der ausgegebenen Kopie ist ebenso wie im Original das Low-Go-Bit gesetzt.
- Nach der Ausgabe des savedIdle-Speichers, werden die im Bypass-Fifo enthaltenen Pakete gesendet. Dabei wird nach jeder Paketsendung der savedI-

dle-Speichers erneut ausgegeben, so daß die Pakete umrahmt von Idle-Symbolen mit gesetztem Low-Go-Bit sind. Dieser Abschnitt wird als Übertragungserholungsphase (Low Transmission Recovery) bezeichnet und der dabei ablaufende Vorgang als Idle-Einfügung.

- Werden während der Übertragungserholungsphase neue Pakete empfangen, werden sie in den Bypass-Fifo eingespeichert und wieder ausgegeben, sobald sie an der Reihe sind. Empfangene Idle-Symbole werden verschmolzen, und der Schuldenzähler wird gegebenenfalls erhöht.
- Nachdem der Bypass-Fifo leer ist, wird für jedes „nicht-konsumierbare“ Idle-Symbol, das verschmolzen wurde, die angesammelte „Schuld“ abgetragen, indem so oft, wie der Schuldenzähler angibt, empfangene Low-Type Idle-Symbole in High-Type Idles umgewandelt und wieder ausgegeben werden. Dieser Abschnitt heißt Schuldenrückzahlphase. Die Schuldenrückzahlung ist dadurch sichergestellt, daß vom Sender mindestens soviele Idle-Symbole empfangen werden, wie der Paketempfänger erzeugt, sobald er das Paket dem Ring entnimmt. Das Abtragen der Schuld wird vorzeitig beendet, sobald ipr≤spr gilt.
- In der Schuldenrückzahlphase kann der Knoten keine neuen Pakete ausgeben, einlaufende Pakete und Idle-Symbole werden jedoch weitergereicht.

Zusätzlich zu diesen Grundregeln gibt es noch einen Regelsatz für den Fall, daß ein Sender blockiert, d.h. vom Senden abgehalten wird. Eine Blockierung kann entweder dann auftreten, wenn der Sender kein Idle-Symbol mit gesetztem Low-Go-Bit empfängt oder wenn er in der Übertragungserholungsphase ununterbrochen Pakete empfängt, die über den Bypass-Fifo zum Ausgabelink zu transportieren sind. Es gilt:

- Ein blockierter Sender bleibt blockiert, solange sein Bypass-Fifo nicht leer ist.
- Ein blockierter Sender setzt beim Weiterreichen von Low-Type Idle-Symbolen deren Low-Go-Bit zurück, um eine Paketsendung bei im Ring nachfolgenden Knoten zu unterdrücken.

Bei der Ausführung des Low-Protokolls müssen auch die Paketempfänger Regeln beachten. Diese sind:

- Wird vom Empfänger ein Request- oder Response-Paket der Länge N (in Symbolen gerechnet) dem Ring entnommen, werden dafür ein Echo und in der Summe (N-4) Low-Type und High-Type Idle-Symbole auf den Ring gegeben. Das Verhältnis zwischen Low-Type und High-Type Idle-Symbolen bestimmt die Bandbreiteaufteilung zwischen dem Low-Pass und dem High-Pass-Protokoll.
- Wird vom Empfänger ein Echopaket dem Ring entnommen, werden dafür in der Summe 4 Low-Type und High-Type Idle-Symbole auf den Ring gegeben.

2.5.3 Pufferallozierung

Werden einem SCI-Knoten von mehreren Sendern gleichzeitig Pakete geschickt, kann dessen Empfangspuffer schnell voll werden. In diesem Fall muß sichergestellt sein, daß die nachfolgenden Retry-Pakete jedes Senders zu einem späteren Zeitpunkt vom Empfänger akzeptiert werden, und daß kein Knoten sich auf Kosten eines anderen Knotens bevorzugt Zugang zum Empfänger verschafft. Dazu dienen Pufferallozierungsprotokolle, die auf endlichen Automaten mit vier Zuständen beruhen, die in jedem Receive-Puffer vorhanden sind. Die vier Zustände spiegeln den jeweiligen Füllgrad des Puffers wieder. Der Füllgrad fließt in das Phase-Feld des negativen Echopakets ein und wird so indirekt dem Sender übermittelt, damit dieser darauf reagieren kann. Die vier Zustände eines Receive-Puffers sind:

- *SERVE_NA*. Der Puffer akzeptiert neue Pakete (Phase-Feld = NOTRY oder DOTRY) und Retry-Pakete, deren Phase-Feld NOTRY, DOTRY oder Retry_A-Status haben, vorausgesetzt, daß Pufferplatz vorhanden ist. Ist kein Puffer frei, werden NOTRY-Pakete mit BUSY_D und DOTRY und Retry_A-Pakete mit BUSY_A negativ beantwortet, und danach wechselt der Empfangspuffer in den Zustand *SERVE_A*. Werden im Zustand *SERVE_NA* Retry-Pakete mit Retry_B-Status empfangen, ist ein Fehler aufgetreten.
- *SERVE_A*. Der Puffer akzeptiert bei freiem Puffer nur Retry-Pakete mit Phase-Feld = Retry_A. Ist kein Puffer frei, erhalten Retry-Pakete mit Retry_A ein BUSY_A-Echo. NOTRY-Pakete werden unabhängig vom Pufferfüllgrad immer mit BUSY_D und DOTRY- und Retry_B-Pakete mit BUSY_B negativ beantwortet. Der Zustand *SERVE_A* wechselt in den Zustand *SERVE_NB* über, sobald alle Retry-Pakete mit Phase-Feld = Retry_A akzeptiert worden sind.
- *SERVE_NB*. Der Puffer akzeptiert neue Pakete (Phase-Feld = NOTRY oder DOTRY) und Retry-Pakete, deren Phase-Feld NOTRY, DOTRY oder Retry_B-Status haben, vorausgesetzt, daß Pufferplatz vorhanden ist. Ist kein Puffer frei, werden NOTRY-Pakete mit BUSY_D und DOTRY und Retry_B-Pakete mit BUSY_B beantwortet, und danach wechselt der Empfangspuffer in den Zustand *SERVE_B*. Werden im Zustand *SERVE_NB* Retry-Pakete mit Retry_A-Status empfangen, ist ein Fehler aufgetreten.
- *SERVE_B*. Der Puffer akzeptiert bei freiem Puffer nur Retry-Pakete mit Phase-Feld = Retry_B. Ist kein Puffer frei, erhalten Retry-Pakete mit Retry_B ein BUSY_B-Echo. NOTRY-Pakete werden unabhängig vom Pufferfüllgrad immer mit BUSY_D und DOTRY- und Retry_A-Pakete mit BUSY_A negativ beantwortet. Der Zustand *SERVE_B* wechselt in den Ausgangszustand *SERVE_NA* zurück, sobald auch alle Retry-Pakete mit Phase-Feld = Retry_B akzeptiert worden sind.

Der sich aus dem Gesagten ergebende Graph der Zustandsübergänge des endlichen Automaten ist in Bild 2.5.1 gezeigt. Eine zusammenfassende Darstellung

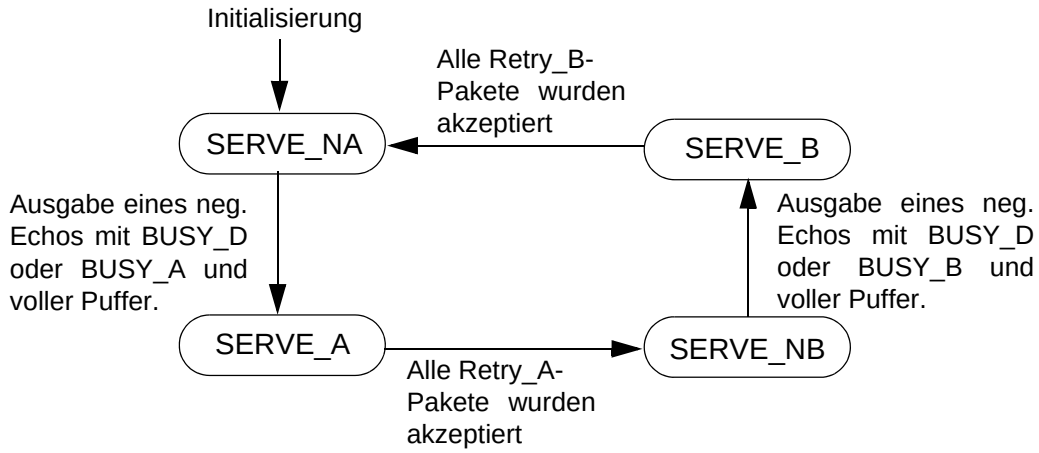

Bild 2.5.1: Zustandsübergänge des endlichen Automaten im SCI-Receive-Puffer.

seiner Ausgaben ist in Tabelle 2.5.1 angegeben. Die Zustände, Übergänge und Ausgaben des Automaten lassen sich auch algorithmisch mit Hilfe des folgenden Pseudokodes beschreiben:

```
IF RequestOrResponseOrRetryArrived = TRUE {Ist ein Paket da?}
 IF state = SERVE_NA {Ist der Zustand = SERVE_NA?}
 IF inFifo = NOT FULL {Eingangspuffer ist nicht voll}
 IF PhaseField = NOTRY
 EchoPhaseField := DONE;
 ELSIF (PhaseField = DOTRY) OR (PhaseField = Retry_A)
 EchoPhaseField := DONE;
 ELSE {PhaseField = Retry_B}
 EchoPhaseField := BUSY_A; SetError;
 state := SERVE_A; {Nachfolgezustand einnehmen}
 END IF; {PhaseField = NOTRY}
 ELSE {inFifo is FULL}
 IF PhaseField = NOTRY
 EchoPhaseField := BUSY_D;
 ELSIF (PhaseField = DOTRY) OR (PhaseField = Retry_A)
 EchoPhaseField := BUSY_A;
 ELSE {PhaseField = Retry_B}
 EchoPhaseField := BUSY_A; SetError;
 END IF; {PhaseField = NOTRY}
 state := SERVE_A; {Nachfolgezustand einnehmen}
 END IF; {Eingangspuffer ist nicht voll}
 ELSIF state = SERVE_A
 IF inFifo = NOT FULL {Eingangspuffer ist nicht voll}
 IF PhaseField = NOTRY
 EchoPhaseField := BUSY_D;
 ELSIF (PhaseField = DOTRY) OR (PhaseField = Retry_A)
 EchoPhaseField := BUSY_B;
 ELSE {PhaseField = Retry_A}
 EchoPhaseField := DONE;
 END IF; {PhaseField = NOTRY}
 ELSE {inFifo is FULL}
```

Freier Empfangspuffer	
normales/ Retry-Paket	Echo-Paket
Zustand SERVE_NA	
NOTRY	DONE
DOTRY, Retry_A	DONE
Retry_B	BUSY_A+err
Zustand SERVE_A	
NOTRY	BUSY_D
DOTRY, Retry_B	BUSY_B
Retry_A	DONE
Zustand SERVE_NB	
NOTRY	DONE
DOTRY, Retry_B	DONE
Retry_A	BUSY_B+err
Zustand SERVE_B	
NOTRY	BUSY_D
DOTRY, Retry_A	BUSY_A
Retry_B	DONE

Voller Empfangspuffer	
normales/ Retry-Paket	Echo-Paket
Zustand SERVE_NA	
NOTRY	BUSY_D
DOTRY, Retry_A	BUSY_A
Retry_B	BUSY_A+err
Zustand SERVE_A	
NOTRY	BUSY_D
DOTRY, Retry_B	BUSY_B
Retry_A	BUSY_A
Zustand SERVE_NB	
NOTRY	BUSY_D
DOTRY, Retry_B	BUSY_B
Retry_A	BUSY_B+err
Zustand SERVE_B	
NOTRY	BUSY_D
DOTRY, Retry_A	BUSY_A
Retry_B	BUSY_B

Tabelle 2.5.1: Ausgaben des endlichen Automaten für freien bzw. vollen Empfangspuffer.

```
IF PhaseField = NOTRY
EchoPhaseField := BUSY_D;
ELSIF (PhaseField = DOTRY) OR (PhaseField = Retry_A)
EchoPhaseField := BUSY_B;
ELSE {PhaseField = Retry_A}
EchoPhaseField := BUSY_A;
END IF; {PhaseField = NOTRY}
END IF; {Eingangspuffer ist nicht voll}
IF AllRetry_AServed = TRUE {Alle Retry_A akzeptiert?}
state := SERVE_NB; {Nachfolgezustand einnehmen}
END IF; {Alle Retry_A akzeptiert}
ELSIF state = SERVE_NB {Ist der Zustand = SERVE_NB?}
IF inFifo = NOT FULL {Eingangspuffer ist nicht voll}
```

```
 IF PhaseField = NOTRY
 EchoPhaseField := DONE;
 ELSIF (PhaseField = DOTRY) OR (PhaseField = Retry_B)
 EchoPhaseField := DONE;
 ELSE {PhaseField = Retry_A}
 EchoPhaseField := BUSY_B; SetError;
 state := SERVE_B; {Nachfolgezustand einnehmen}
 END IF; {PhaseField = NOTRY}
 ELSE {inFifo is FULL}
 IF PhaseField = NOTRY
 EchoPhaseField := BUSY_D;
 ELSIF (PhaseField = DOTRY) OR (PhaseField = Retry_B)
 EchoPhaseField := BUSY_B;
 ELSE {PhaseField = Retry_A}
 EchoPhaseField := BUSY_B; SetError;
 END IF; {PhaseField = NOTRY}
 state := SERVE_B; {Nachfolgezustand einnehmen}
 END IF; {Eingangspuffer ist nicht voll}
 ELSE {state = SERVE_B}
 IF inFifo = NOT FULL {Eingangspuffer ist nicht voll}
 IF PhaseField = NOTRY
 EchoPhaseField := BUSY_D;
 ELSIF (PhaseField = DOTRY) OR (PhaseField = Retry_A)
 EchoPhaseField := BUSY_A;
 ELSE {PhaseField = Retry_B}
 EchoPhaseField := DONE;
 END IF; {PhaseField = NOTRY}
 ELSE {inFifo is FULL}
 IF PhaseField = NOTRY
 EchoPhaseField := BUSY_D;
 ELSIF (PhaseField = DOTRY) OR (PhaseField = Retry_A)
 EchoPhaseField := BUSY_A;
 ELSE {PhaseField = Retry_B}
 EchoPhaseField := BUSY_B;
 END IF; {PhaseField = NOTRY}
 END IF; {Eingangspuffer ist nicht voll}
 IF AllRetry_BServed = TRUE {Alle Retry_B akzeptiert?}
 state := SERVE_NA; {Nachfolgezustand einnehmen}
 END IF; {Alle Retry_B akzeptiert}
 END IF; {Zustand = SERVE_NA}
END IF; {Paket da}
```

Da nur diejenigen Sender, die unmittelbar vorher ein negatives Echo von einem Empfänger mit vollem Receive-Puffer empfangen haben, über dessen Pufferzustand Bescheid wissen, jedoch nicht andere Knoten, kann es geschehen, daß an einen Empfänger normale und Retry Request- und Response-Pakete geschickt werden. Je nach Zustand des endlichen Automaten und je nach Art des empfangenen Pakets, sendet dieser Echopakete mit entsprechenden Phase-Feldern zurück. Ebenso ist es auch möglich, daß vom selben Sender multiple offenstehende Requests an denselben Empfänger abgeschickt werden, bevor das erste (negative) Echo eintrifft, so daß vom Empfänger, auch wenn er in ein und demselben Zustand ist, Echos mit verschiedenen Phase-Feldern erzeugt werden

können. Zur Erläuterung der einzelnen Phase-Felder sind deren Namen und ihre Bedeutung in Tabelle 2.5.2 zusammenfassend dargestellt.

Phase-Feld von Request/Response/Retry-Paket	
Name	Bedeutung
NOTRY	Paket mit Hoffnung auf freien Pufferplatz
DOTRY	Paket mit Empfangspufferreservierung bei negativem Echo
Retry_A	Paketwiederholung nach Empfang von BUSY_A-Echo
Retry_B	Paketwiederholung nach Empfang von BUSY_B-Echo
Phase-Feld von positivem Echo-Paket	
Done	Paket eingespeichert
Phase-Feld von negativem Echo-Paket	
BUSY_N	nachfolgendes Retry-Paket soll NOTRY haben
BUSY_D	nachfolgendes Retry-Paket soll DOTRY haben
BUSY_A	Puffer reserviert für Retry-Paket mit Retry_A
BUSY_B	Puffer reserviert für Retry-Paket mit Retry_B

Tabelle 2.5.2: Zusammenfassung der Phase-Felder.

3 Statische/dynamische SCI-Netze

3.1 Einleitung

SCI eignet sich im Prinzip sowohl für statische als auch dynamische Netztopologien. Aufgrund seiner inhärenten Ringstruktur können bei statischen Netzen jedoch nur solche Topologien direkt aufgebaut werden, die auf Ringen beruhen. Dazu zählen hauptsächlich die k-nären n-Kuben, die auch als n-dimensionale Tori bekannt sind. Der bei weitem größte Teil bekannter statischer Topologien wie Gitter, Bäume, deBruijn- und Star-Graphen kann von SCI nur indirekt realisiert werden, indem man die gerichteten oder ungerichteten Kanten ihrer Graphen durch kleine unidirektionale Ringe, die sog. Ringlets, ersetzt. In Bild 3.1.1 ist ein Beispiel für ein 2-D-Gitter in SCI-Technologie dargestellt. Statische SCI-Netze wurden für den Fall von k-nären n-Kuben von [Bothetal93] eingehend untersucht, für andere statische Netze liegen jedoch kaum Analysen vor.

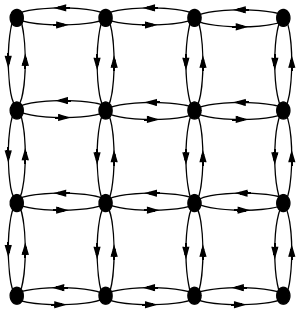

Bild 3.1.1: 2-D-Gitter in SCI-Technologie.

Bei dynamischen Netzen können alle Topologien unmittelbar in SCI umgesetzt werden. Die Ringstruktur wird hier außerhalb der eigentlichen Netztopologie vom Netzausgang zurück zum Eingang geschlossen. Selbstverständlich hat man auch die Möglichkeit Ringlets einzusetzen. In Bild 3.1.2 sind anhand eines Beispiels die beiden Varianten a) und b) eines mehrstufigen Netzes in SCI-Technologie gezeigt. Durch das Netz werden die Prozessoren P0-P7 am Netz-

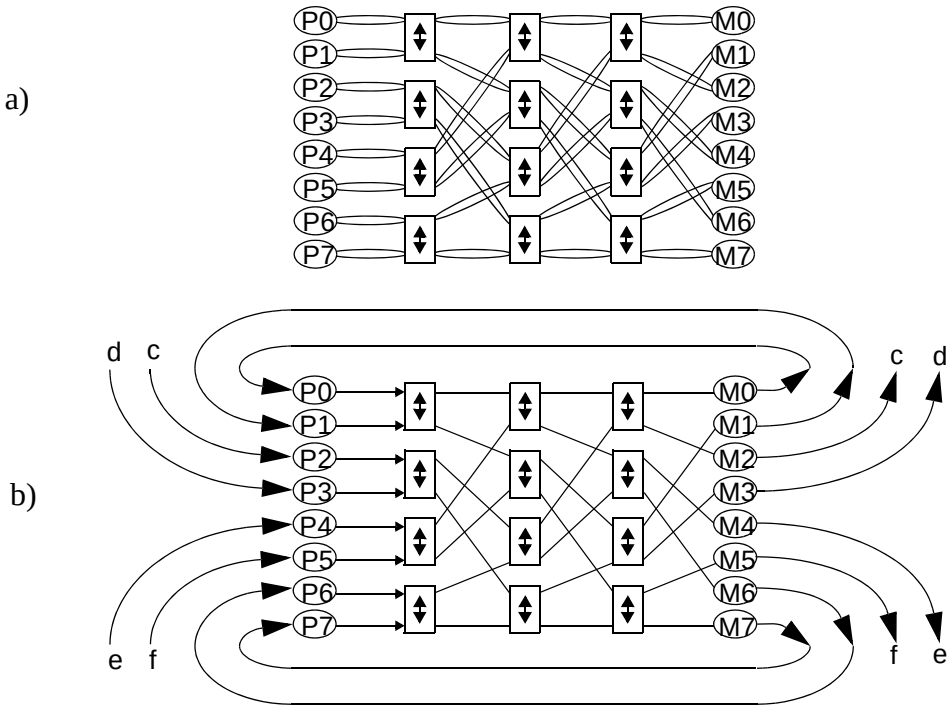

Bild 3.1.2: Dynamisches Netz in SCI-Technologie mit und ohne Ringlets (a bzw. b).

eingang mit Speichermodule M0-M7 am Ausgang verbunden. Bei Verwendung von Ringlets (Bild 3.1.2a) wird das Netz über kleine SCI-Ringe angeschlossen, ansonsten über lange, durchgängige Ringe (Bild 3.1.2b). Im letzteren Fall erfolgt die Rückrichtung entlang korrespondierender Verbindungen c-c bis f-f.

Dynamische SCI-Netze wurden bislang wenig analysiert. Ein Beispiel stellt die Untersuchung der Baseline-Topologie durch [Wu94a][Wu94b] dar.

3.2 Die Deadlock-Problematik

Die Deadlock-Problematik bei Netzen wurde an anderer Stelle bereits ausführlich erörtert. Wichtig hier ist festzuhalten, daß Deadlocks bei statischen Netzen in Abhängigkeit von der gewählten Topologie und des Routing-Verfahrens auftreten können. Dynamische Netz hingegen sind Deadlock-frei, sofern sie in die Kategorie der kreisfreien Graphen fallen. Dies trifft beispielsweise auf alle Banyan-Topologien zu. Allerdings sind SCI-basierte, dynamische Netze nie kreisfrei, da SCI geschlossene Ringe benötigt. Der Unterschied zu den statischen Netzen besteht jedoch darin, daß jeder SCI-Ring in einem mehrstufigen Netz, z.B. in der Art nach Bild 3.1.2a oder b, nur einen einzigen Sender und einen Empfänger enthält. Zwischen Sender und Empfänger sind Schaltknoten, die keine autonomen Datenquellen darstellen und dementsprechend auch nicht von sich aus Pakete erzeugen. Deshalb kann bei dynamischen SCI-Netzen nicht die Situation auftreten, daß zwei oder mehr Sender sich gegenseitig blockieren. Das bedeutet in der Praxis, daß das Deadlock-Problem gelöst ist, sobald man dynamische SCI-Netze verwendet.

3.3 Motivation für SCI-basierte Banyan-Netze

Neben der potentiellen Deadlock-Gefahr, die bei einigen statischen Netzen auftreten können, wenn man „einfache“ Routing-Algorithmen verwendet, haben statische Topologien bei Echtzeitanwendungen im Vergleich zu dynamischen Netzen weitere Nachteile. Da bei statischen Netzen in der Regel mehrere Sender an denselben SCI-Ring angeschlossen sind, kann die maximale Latenz einer Transaktion im Ring nicht vorausgesagt werden, vielmehr hängt sie vom Verkehrsaufkommen am gemeinsam benutzten Medium und der Zahl der Retry-Pakete ab. Aus denselben Gründen ist die minimale Bandbreite, die zwischen zwei Knoten verschiedener Ringe erreicht werden kann, nicht bestimmbar. Im selben Ring kann allerdings durch die SCI-Bandbreiteallozierungsprotokolle ein Minimum garantiert werden.

Bei einem echtzeitfähigen Rechensystem muß eine Obergrenze für die Latenz und eine Untergrenze für die Bandbreite angebbar sein, sonst lassen sich damit keine Steuerungen, Regelungen oder Datenerfassungssysteme realisieren. Wenn die maximale Latenz L_{max} und die minimale Bandbreite b_{min} gege-

ben sind, dann kann man die Reaktionszeit T des Netzes in Abhängigkeit von der zu transferierenden Bytezahl n angeben als:

Gl. 3.3.1:
$$T(n) = L_{max} + \frac{n}{B_{min}}$$

Als Nachteil dynamischer Netze gilt allgemein deren höhere Kosten, weswegen sie für nicht-Echtzeitanwendungen häufig unattraktiv erscheinen. Daß dies bei SCI nicht notwendigerweise richtig ist, zeigt der folgende Vergleich zwischen einem binären Hyperkubus und einem SCI-Banyan-Netz:

Die Kosten bei beiden Netzen werden überwiegend von der Zahl der Netzschnittstellen, d.h. Link-Controller-Bausteinen und weniger von deren Verkabelung bestimmt. Um bei dynamischen SCI-Netzen Kosten zu sparen, kommen nur unidirektionale Banyan-Topologien ähnlich wie in Bild 3.1.2b in Frage, da sie bei gegebener Portzahl P (P>1) pro Schalter die kleinstmögliche Stufenzahl s gemäß $s = \log_P N$ aufweisen, wobei N die Netzgröße ist. Die Kosten K_B für ein Banyan-Netz berechnen sich gemäß Gl. 3.3.2. Ein solches Banyan-Netz

Gl. 3.3.2:
$$K_B(N) = N \cdot \log_P N$$

verbindet N Prozessoren mit genausovielen peripheren Einheiten oder Speichermodule. Wählt man als Vergleich einen binären n-Kubus mit unidirektional betriebenen Verbindungen zwischen den Knoten, so benötigt man an jedem Kreuzungspunkt der Topologie n Netzschnittstellen, um Pakete in alle n Dimensionen schicken zu können. Ein Kubus hat insgesamt $N=2^n$ Kreuzungspunkte, an denen genauso viele Prozessoren, periphere Einheiten oder Speichermodule untergebracht sein können. Seine Kosten K_K lassen sich anhand von Gl. 3.3.3 bestimmen. Daraus sieht man, daß der Hyperkubus nur dann bil-

Gl. 3.3.3:
$$K_K(N) = N \cdot n = N \cdot \log_2 N$$

liger ist, wenn $K_K < K_B$ ist, d.h. wenn P<2 gilt. Dies ist jedoch nach Voraussetzung nicht möglich.

Umgekehrt folgt, daß ein SCI-Banyan billiger als ein binärer Hyperkubus ist, sofern er aus Schaltern mit mehr als zwei Ports (P>2) aufgebaut ist. Bei k-nären n-Kuben erhält man als Bedingung für kostengünstigere SCI-Banyans $P > k$, d.h. die Port-Zahl muß größer als die Kantenlänge des entsprechenden Überwürfels sein.

Beispiel:

Ein 4-D Hyperkubus enthält 16 Kreuzungspunkte und benötigt 64 SCI-Schnitt-

stellen. Der entsprechende SCI-Banyan kommt bei $P = 4$ mit nur 32 Schnittstellen aus. Der Graph seiner Topologie ist in Bild 3.3.1 dargestellt, dabei bezeichnet S einen Paketsender (Prozessor) und D einen Paketempfänger (Speicher, Peripherie etc.).

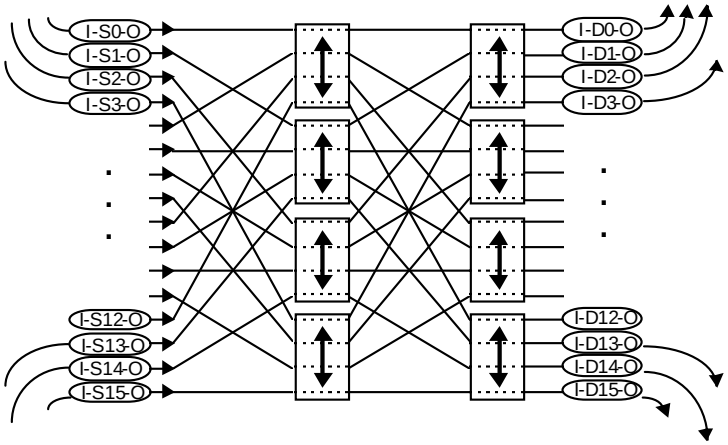

Bild 3.3.1: Graph eines SCI-Banyans mit 16 Ein- und Ausgängen.

Anders fällt der Vergleich mit einem binären Baum aus. Bei n Knotenebenen hat der Baum $N = 2^n$-1 Knoten, von denen ein Knoten die Spitze darstellt, 2^{n-1} Knoten bilden die Blätter, und dazwischen liegen 2^n-1 - 1 - 2^{n-1} übrige Knoten. Die Spitze benötigt zwei Netzschnittstellen, die Blätter je eine und alle Zwischenknoten drei, damit ein zusammenhängender Graph entsteht. Daraus ergeben sich die Kosten K_T für den Binärbaum gemäß Gl. 3.3.4. Der Vergleich zum

Gl. 3.3.4:
$$K_T = 2^{n+1} - 4 = 2N - 2$$

Banyan mit P=4 zeigt, daß für N=3 beim Baum bzw. N=4 beim Banyan beide Topologien 4 Schnittstellen benötigen. Ab N=15 bzw. N=16 sind jedoch beim Baum 28 und beim Banyan 32 Schnittstellen erforderlich. Für alle weiteren N ist der Baum ebenfalls günstiger. Allerdings haben die am weitesten entfernten Knoten eines Baumes höhere Latenz und geringere Bandbreite als beim Banyan.

Ergebnis:

Die vorangegangenen Analysen lassen sich in vier Stichpunkten zusammenfassen:

- Statische SCI-Netze lassen unmittelbar nur Topologien zu, die auf Ringen basieren.

- Dynamische SCI-Banyans haben kein Deadlock-Problem.
- Statische SCI-Netze sind für Echtzeitanwendungen wegen schlechter Vorhersagbarkeit der Obergrenze der Latenz sowie der Untergrenze der Bandbreite nur bedingt geeignet.
- Dynamische SCI-Banyans haben ab Schaltergrößen von mindestens vier Ports geringere Kosten als vergleichbare Torustopologien.

Aus diesen Gründen werden im folgenden die Leistungsdaten von dynamischen SCI-Banyans verschiedener Topologien und Routing-Verfahren untersucht werden.

4 Anwendungsbeispiel für SCI: Datenerfassungssystem

SCI ist vom Preis seiner Komponenten und von seinen Leistungsdaten her an „high-end“-Anwendungen der Informationstechnik ausgerichtet. Dazu zählen Hochgeschwindigkeitsvernetzungen von Prozessoren untereinander, wie z.B. bei dem Parallelrechner der Fa. HP-CONVEX [Convex94b], oder extrem schnelle Kopplungen zwischen peripheren Plattenspeichern und Zentralsystemen wie bei der Fa. SGI-Cray [Scott96] sowie leistungsfähige Cluster-Verbindungen zwischen SMP-Servern wie bei den Firmen Data General, Sequent, Siemens/SNI und Sun [Omang96].

Weniger spektakulär aber deswegen nicht weniger wichtig sind Anwendungen von SCI bei Steuerungen, Regelungen und Datenerfassungen [Richtetal93], bei denen extrem große Datenmengen anfallen, die in Echtzeit transportiert und vorverarbeitet werden müssen. Anwendungen mit diesen Anforderungsprofilen sind in zivilen Bereichen u.a. im Kraftwerksbau sowie in Großexperimenten der Hochenergie- und der Plasmaphysik zu finden. Beispiele solcher Großexperimente, bei denen potentiell SCI eingesetzt werden kann, sind der zukünftige Beschleuniger des CERN für schwere Elementarteilchen (Large Hadron Collider) [Bogaerts92], das Experiment W7-X des Max-Planck-Instituts für Plasmaphysik zur Erforschung der kontrollierten Kernfusion, sowie der von Amerikanern, Europäern, Japanern und Russen geplante experimentelle Fusionsreaktor ITER.

Im folgenden soll eine Leistungsanalyse eines SCI-Netzwerks für ein Datenerfassungssystem durchgeführt werden, um anhand eines konkreten Beispiels

das prinzipielle Vorgehen bei einer Leistungsanalyse zu demonstrieren. Generall erfolgt die Leistungsanalyse eines Netzes in sechs Schritten:

- Zunächst ist die Wahl geeigneter Metriken erforderlich, anhand derer man das Netz beurteilen möchte.
- Danach muß eine Modellierung des Netzwerkes anhand von charakteristischen Parametern durchgeführt werden, um das Problem zu vereinfachen und zu formalisieren.
- Nach der Modellierung erfolgt die Implementierung des Modells, zumeist in Form eines Netzwerksimulators.
- In der Regel schließt sich daran eine Validierung von Modell und Implementierung anhand einfacher Spezialfälle und leicht überschaubarer Beispiele an. Manchmal wird auch ein exemplarischer Testaufbau hergestellt, an dem konkret Messungen durchgeführt werden können.
- Jetzt kann die eigentliche Simulation auf dem Rechner erfolgen, bei der eine Vielzahl von Testläufen durchgeführt wird, um die hinsichtlich der gewählten Metriken optimale Netzvariante zu finden.
- Für die abschließende Leistungsbewertung müssen die Simulationsergebnisse mit der Spezifikation der gewünschten Leistungsanforderungen verglichen werden, um daraus eine Aussage über die Netzgüte treffen zu können.

Die weiteren Kapitel erläutern jeden der sechs Schritte im Detail.

4.1 Anforderungen zukünftiger Fusionsexperimente

Der erste Schritt der Leistungsanalyse eines Netzes als Teil einer bestimmten Anwendung ist die Wahl geeigneter Metriken anhand derer das Netz beurteilt und optimiert werden soll. Diese ergeben sich zumeist aus den charakteristischen Kennzeichen der Anwendung. Bezogen auf das Beispiel einer Datenerfassung in Plasma- und Elementarteilchenphysik ist ein solches Kennzeichen das exponentielle Wachstum der Datenmengen im Laufe der Betriebsdauer der physikalischen Experimentieranordnung. Zur Erläuterung der daraus erforderlichen Skalierbarkeit des Netzes ist in Bild 4.1.1 bei verschiedenen Fusionsexperimenten die Zunahme der Datenmenge pro Meßaufnahmezyklus als Funktion der letzten 3 Dekaden dargestellt. Die Daten wurden anhand der Angaben in [Preckshot86], [vBeken87], [McHarg87], [Balme88], [Nijman88], [Korteetal91], [vHaren93], und [Hertweck88] gewonnen.
Zu Beginn der 80er Jahre lagen die Anforderungen der Datenmengen bei maximal 10 MB pro Meßzyklus, die in ca. 10 Sekunden anfielen [McHarg85]. Anhand von Bild 4.1.1 wird ersichtlich, daß während der Lebensdauer des

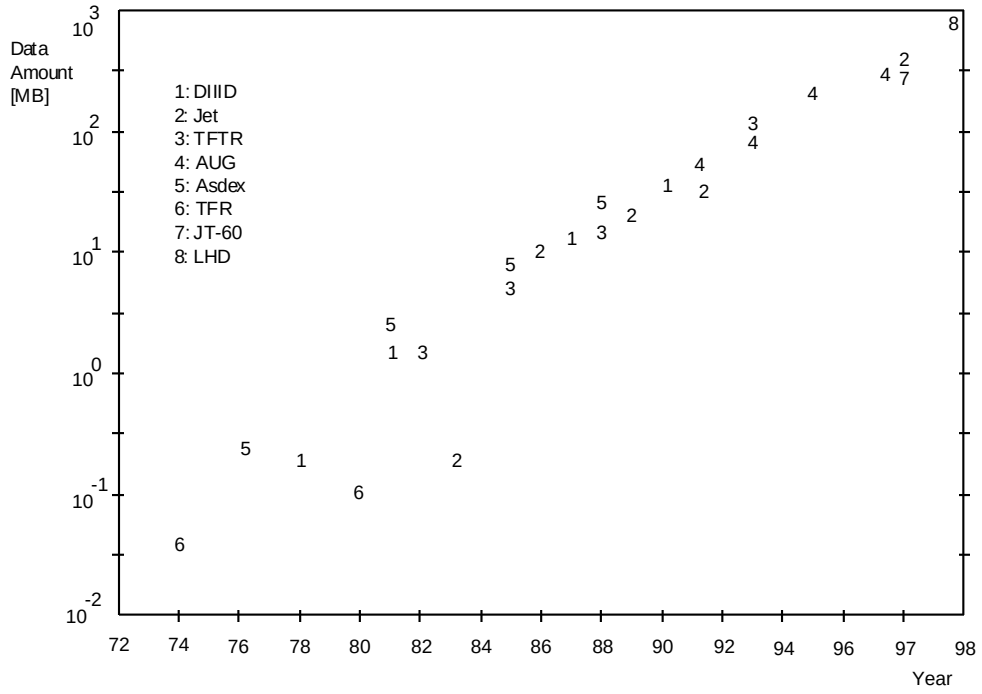

Bild 4.1.1: Exponentielle Zunahme der Datenmenge bei Fusionsexperimenten.

Datenerfassungssystems, die in der Plasmaphysik bei ca. 8 Jahren liegt, die aufzunehmende Datenmenge um ungefähr den Faktor 20 ansteigt, so daß man heute Datenmengen im Bereich einiger hundert MB pro Meßzyklus hat, die in ungefähr derselben Meßzeit anfallen, so daß entsprechend höhere Datenraten erforderlich sind.

Daraus ergibt sich, daß als Metrik für das Netz die Skalierbarkeit ein wichtiges Kriterium ist. Die Skalierbarkeit kann z.B. als größtmöglicher Erweiterungsfaktor bezogen auf die Netzursprungsgröße angegeben werden.

Als zweite Metrik zur Leistungsbeurteilung eines Netzes für ein Datenerfassungssystem ergibt sich aus dem bisher Gesagten die Sensordatenrate, gemessen in MB/s/Datenaufnahmekanal.

Daß hohe Datenraten auch in der Elementarteilchenphysik wichtig sind, wird klar, wenn man sich vergegenwärtigt, daß Kernreaktionen in sehr kurzen Zeitintervallen ablaufen, so daß Sensoren, die diese Reaktionen verfolgen, entsprechend oft abgetastet werden (Mikro- bis Nanosekundenbereich). Ebenso sind viele (i.a. Tausende) von Meßstellen erforderlich.

Bereits 1991 wurde beispielsweise für den damals geplanten *Superconducting Supercollider* in Texas eine summierte Rohdatenrate von 100 TB/s prognostiziert, die nach Datenreduktion auf 10-100 MB/s reduziert werden sollten [Milner91]. Das zukünftige *Large Hadron Collider* (LHC) Beschleunigerexperiment beim CERN soll 10^7-10^8 Meßkanäle aufweisen, die in der ersten Meßstufe 10^{10}-10^{11} Bytes/s liefern, die schrittweise auf 10^8-10^9 bzw. 10^7-10^8 Bytes/s reduziert werden [Mapelli91]. Kleinere Datenerfassungssysteme wie

für die *Continuous Electron Beam Accelerator Facility* (CEBAF) weisen immerhin noch 160 MB/s Rohdatenrate auf [Quarie91].

Daraus kann man schließen, daß die beiden wichtigsten Metriken von Netzen für Datenerfassungssysteme der Hochenergie- und Plasmaphysik die Sensordatenrate und die maximal erreichbare Zahl der Kanäle (Skalierbarkeit) sind. Netze verschiedener Topologien, Betriebsweisen und RoutingVerfahren sollten anhand dieser Metriken verglichen werden. Weitere Maße wie z.B. Zuverlässigkeit, Wartbarkeit, maximal tolerierbare Fehlerraten u.s.w. müssen allerdings im Einzelfall mit berücksichtigt werden.

4.2 Beispiel eines Datenerfassungssystems

Der zweite Schritt der Leistungsanalyse eines Netzes für eine bestimmte Anwendung ist die Modellierung des Netzes anhand von möglichst wenigen Parametern, die das Verhalten des realen Systems ausreichend gut wiedergeben. Um beurteilen zu können, welche Parameter wichtig sind und deshalb in das Netzmodell einfließen sollten, muß man zuerst wissen, wie das übergeordnete System aussieht, in dem das Netz integriert ist. Dazu wird im folgenden für das Beispiel eines Datenerfassungsnetzwerkes der Aufbau einer SCI-basierten Datenaufnahme vorgestellt.

Datenerfassungssysteme der Experimentalphysik müssen im wesentlichen vier Aufgaben leisten:

- Abtastung und Digitalisierung analoger Sensorwerte,
- Übertragung der Rohdaten von den Aufnahmeeinheiten zu den Verarbeitungsrechnern,
- Kurzzeitspeicherung der Daten zur Vorverarbeitung (Kalibrierung, Filterung, etc.) und
- Langzeitarchivierung und Auswertung über FFT, Tomographie, Statistik, und andere Verfahren.

Seit etwa drei Jahrzehnten werden rechnergestützte Datenerfassungssysteme bei Anlagen zur Erforschung der kontrollierten Kernfusion sowie der Hochenergiephysik verwendet. Den betrachteten Datenerfassungssystemen ist gemeinsam, daß von einem räumlich verteilten Rechnersystem Meßwerte von einer Vielzahl von Kanälen ($>10^3$) eingelesen werden. Die Rohdaten wiederum werden von Meßaufnahmeapparaturen bereitgestellt, die CAMAC-, FASTBUS- oder VMEbus-basierend sind. Jedes Rechnersubsystem stellt ein eigenes, eingebettetes Datenerfassungssystem “im Kleinen“ dar, an das einige Dutzend bis Hunderte von Kanälen angeschlossen sind. Dabei sind zwischen den Meßaufnahmeeinheiten und den ihnen zugeordneten Rechnern räumliche Distanzen im Bereich von mehreren Metern bis zu einigen Kilometern zurückzulegen. Zu-

meist werden dazu Glasfaser aufgrund ihrer Zuverlässigkeit und der üblicherweise hohen Datenraten zur Datenübertragung eingesetzt.

Aus diesen Randbedingungen kann man eine Referenzarchitektur ableiten, die im wesentlichen drei Stufen umfaßt:

- das Datenaufnahmesystem,
- das Datenvorverarbeitungssystem und
- das Auswertesystem.

Das Blockschaltbild der Referenzarchitektur ist in Bild 4.2.1 dargestellt. Die

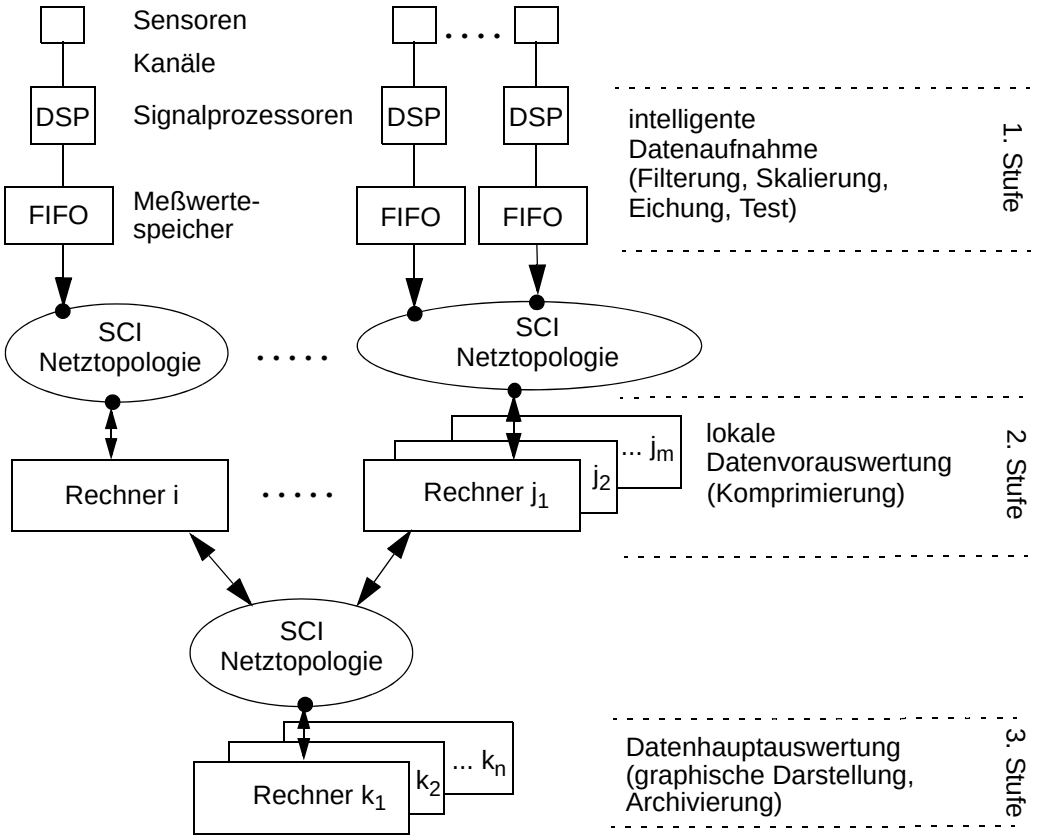

Bild 4.2.1: Blockschaltbild eines SCI-basierten Datenerfassungssystems.

Komponenten der verschiedenen Stufen werden über je ein SCI-Verbindungsnetz miteinander gekoppelt. Typischerweise nimmt die Anzahl der Komponenten von Stufe zu Stufe ab. Die Daten werden in der ersten Stufe mit Hilfe einer Vielzahl von Sensoren aufgenommen und in je einem Meßwertespeicher (organisiert als FIFO) temporär abgelegt. Diese Stufe wird hier nicht näher betrachtet. Sie ist nur insoweit relevant, als die Abtastrate der Sensoren, die Datenbreite der Kanäle und die Kapazität der Meßwertespeicher die Rate bestimmt, mit der die aufgenommenen Daten in die zweite Stufe geschafft werden müssen.

Auf der zweiten Stufe befinden sich die Datenvorverarbeitungsrechner, die Filterung, Normierung und/oder Kompression auf den Daten vornehmen. Interessant ist hier die Art und Weise, wie die Daten über das SCI-Verbindungsnetz von den Meßwertespeichern geholt bzw. geliefert werden: Möglich sind aus der Sicht der Meßwertespeicher eine Pull- oder eine Push-Strategie. Das bedeutet, daß entweder die Daten von den Rechnern gelesen werden (SCI Read Transaction) oder, daß die Sensoren ihre Daten in die zugeordneten Rechner schreiben (SCI Write Transaction), was jedoch aktive Komponenten bei den Meßwertespeichern voraussetzt.

Die Rechner der zweiten Stufe sind über ein weiteres SCI-Verbindungsnetz mit dem oder den Rechnern der dritten Stufe verbunden, welche die vorverarbeiteten Daten auswerten, graphisch darstellen und archivieren.

Die Referenzarchitektur erlaubt eine effiziente Pipeline-Verarbeitung. Die Unabhängigkeit der Aufgaben, welche die Komponenten und Rechner der drei Stufen zu erledigen haben, läßt eine überlappte Abarbeitung im Sinne einer Makro-Pipeline zu, wodurch der Datendurchsatz wesentlich erhöht wird.

Für die Praxis ist jedoch die projektierte Referenzarchitektur zu grob und gewährt zu viele Freiheitsgrade, die für eine nachfolgende Modellierung noch eingegrenzt werden müssen. Die zu bestimmenden Freiheitsgrade sind im einzelnen:

- Wie viele Sensoren/Meßwertespeicher können pro SCI-Ring und pro Rechner zweiter Stufe angeschlossen werden, in Abhängigkeit von der Datenaufnahmerate und der Kapazität der Speicher?
- Wie viele Rechner zweiter Stufe können von einem Rechner dritter Stufe bedient werden?
- Muß es mehr oder kann es weniger Stufen in der Hierarchie geben?
- Müssen Rechner auf einer Stufe miteinander gekoppelt sein? Wenn ja, wie?
- Wie kommen Daten von den Meßwertespeichern zu den Rechnern der zweiten Stufe: Pull oder Push, über Nachrichtentransfers *(*Message Passing*)* oder Speicheroperationen (Shared Memory)?
- Wer übernimmt die Rolle eines Masters beim Transport der Daten von Rechnern zweiter auf die dritte Stufe?
- Wie viele Daten können pro Ring transportiert werden?
- Welche Netzart (statisch/dynamisch) und Netztopologie ist günstig bzw. in jeder Stufe erforderlich?

Durch eine Vereinfachung der Referenzarchitektur und eine weitere Konkretisierung können die Freiheitsgrade reduziert werden. Die Netztopologie beispielsweise wird aufgrund des in Kapitel 3 "Statische/dynamische SCI-Netze" Gesagten als Banyan-Netz gewählt. Das Blockschaltbild der daraus resultierenden vereinfachten Datenaufnahme ist in Bild 4.2.2 gezeigt.

Die noch verbleibenden Fragen bzgl. der Modellierung können konzeptionell nicht geklärt werden, vielmehr müssen sie als zu variierende Parameter in das Netzmodell übernommen werden. Durch eine größere Zahl von Simulations-

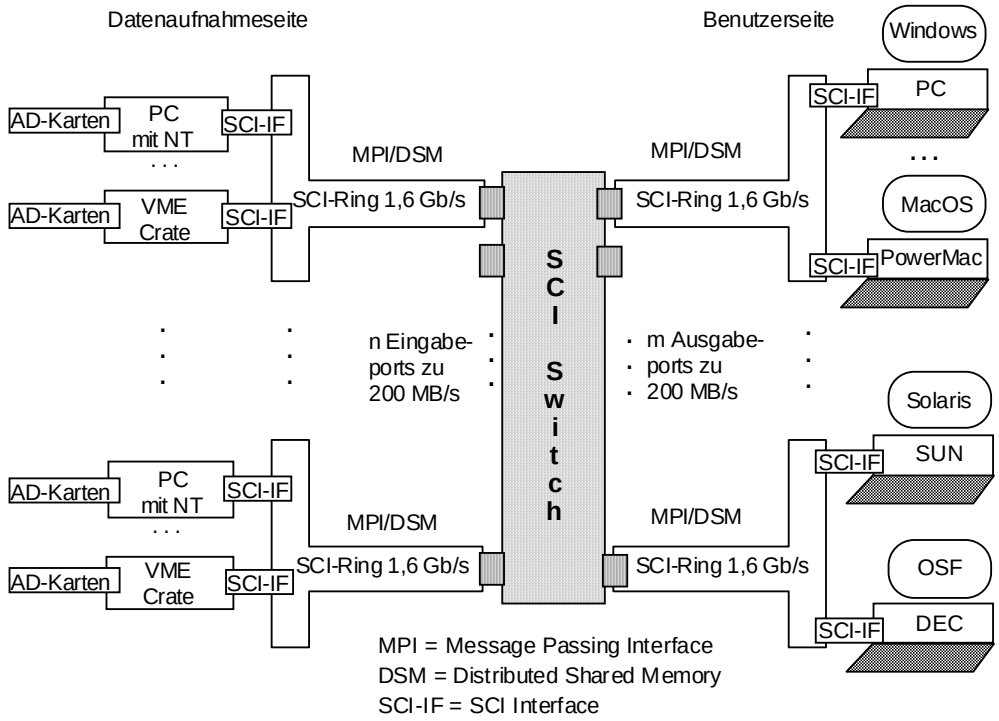

Bild 4.2.2: Blockschaltbild eines vereinfachten Datenerfassungssystems.

läufen kann man im Anschluß anhand der gewählten Metriken bestimmen, welche Netzvarianten am günstigsten sind. Die zu variierenden Simulationsparameter sind:

- der Typus der Netztopologie innerhalb der Kategorie der Banyan-Netze,
- die maximal möglichen Datenraten der Meßaufnehmer,
- die größte Zahl der Sensoren, d.h. die zu simulierende Netzgröße,
- die maximal zulässige Zeit zwischen Datenaufnahme und Verarbeitung (Latenzzeit),
- die maximal zulässige Datenverlustrate für noch störungsfreien Betrieb (Paketverluste) und
- den größten Faktor, um den das Netz erweiterbar ist (Skalierung).

Die Simulationsläufe müssen unter verschiedenen Lastbedingungen durchgeführt werden, um realistische Aussagen treffen zu können. Unter Last versteht man dabei die Art und Größe des von den simulierten Datenquellen erzeugten Verkehrs.

Prinzipiell stehen bei einer Lasterzeugung die beiden Möglichkeiten einer deterministischen bzw. einer stochastischen Datenerzeugung zur Disposition. Da bei Datenerfassungssystemen Sensoren stets aufgrund eines periodisch wiederkehrenden Trigger-Signals abgetastet werden, kommt hier nur deterministische Lasterzeugung in Frage. Darüberhinaus beginnen in der Regel die Trigger-Si-

gnale der Sensoren während eines Meßzyklus synchron zu einem gemeinsamen Datenaufnahmestartsignal. Dadurch sind die individuellen Meßaufnahmetrigger zueinander in einer festen Phasenbeziehung, so daß die Meßwerte bzgl. ihrer Abtastzeitpunkte und Werte miteinander verglichen werden können. Dieses periodische und synchrone Lastverhalten steht im Gegensatz zur Last bei Rechnernetzen oder Parallelrechnern bei denen man von stochastischem Verkehr ausgeht.

4.3 Modellierung des Datenerfassungsnetzwerks

Der dritte Schritt zur Leistungsbewertung eines Netzes, der im weiteren demonstriert werden soll, ist die Netzmodellierung mit Hilfe einer vollständigen Liste aller wichtigen Modellparameter, um so ein ausreichend genaues Abbild der realen Verhältnisse zu erhalten. Aus Komplexitätsgründen wird jedoch in den meisten Fällen darauf verzichtet, ein 100% exakte Repräsentation der Wirklichkeit zu finden, da dazu im Einzelfall beliebig hoher Aufwand notwendig wäre, währenddessen die Genauigkeit der Simulationsergebnisse nur minimal zunehmen würde.

Zur Modellierung eines Datenerfassungssysteme wird bei SCINET die Realität in die drei Kategorien Datenquellen, Netzwerk und Datensenken abstrahiert. Diese sind in der Realität räumlich getrennt voneinander angebracht. Die Entfernung zwischen einer Datenquelle und dem Netz sowie zwischen dem Netz und einer Senke wird bei SCINET durch individuell lange Signallaufzeiten (linkDelays) nachgebildet. Als generelle Voreinstellung für die Signallaufzeiten zwischen Quellen und Netz gelten 100 ns (Default-Wert), was bei einer Geschwindigkeit von 20 cm/ns eines elektrischen Signals in einem Kupferkabel einer Entfernung von 20 m entspricht und einen realistischen Wert für ein Datenerfassungssystem darstellt. Bei der Modellierung von Glasfaserkabeln gilt gegebenenfalls eine andere Signalausbreitungsgeschwindigkeit. Innerhalb des Netzes können zwischen jedem Knotenpaar ebenfalls individuelle Laufzeiten spezifiziert werden. Deren Default-Wert beträgt 1 ns (=20cm).

Weitere Modellierungsparameter sind der Netztypus und die Netzgröße. Innerhalb der Kategorie der Banyan-Netze stehen dabei eine Reihe verschiedener Topologien zur Verfügung. Als Randbedingung ist zu beachten, daß die Netzgröße N in Abhängigkeit des jeweiligen Netztyps als Zweier- oder Viererpotenz gewählt werden muß. Der Maximalwert von N ist dabei 256, d.h., es können Netze bis zu 256 Ein- und Ausgängen simuliert werden.

4.4 Lastprofile der Datenerfassung

Bei SCINET kann die „Last“, die von den Datenquellen in das Netz in Form von Paketen eingespeist wird, in weitem Maß variiert werden: Die Quellen können entweder in deterministischen oder stochastischen Zeitintervallen Pakete erzeugen und diese über das Netz zu zufällig ausgewählten oder fest vorgegeben Zielen senden. Die Art der ausgesandten Pakete ist bei allen Datenquellen gleich und muß vor Beginn der Simulation angegeben werden. Es stehen dabei sämtliche von IEEE definierten SCI-Pakettypen bzw. Transaktionen zur Verfügung.

Bei deterministischem Zwischenankunftszeiten (Interarrival Time) der Pakete im Netzwerk gibt es zwei Optionen. Entweder senden alle Quellen zur selben Zeit und mit derselben Rate *R*, die von außen als Simulationsparameter wählbar ist, oder die Quellen haben als Senderate ein ganzzahliges Vielfaches eines Grundtaktes zueinander. Bei der zweiten Option sind die Zeiten, zu denen Pakete erzeugt werden, ebenfalls „in Phase“, nur die Paketfrequenzen sind verschieden. Es gilt, daß durch das ganzzahlige Verhältnis der Paketfrequenzen Phasenbeziehungen und damit auch Synchronizität erhalten bleiben.

Die festen Phasenbeziehungen dienen dazu, ein gemeinsames Startsignal der Datenaufnahmetrigger zu modellieren. Die Synchronizität ist charakteristisch für jede Datenerfassung, denn um die Meßwerte miteinander vergleichen zu können, müssen die Zeitpunkte der Sensorabtastung auf der Zeitskala eines gemeinsamen Grundrasters liegen. Das Grundraster wird bei SCINET vom Grundtakt vorgegeben.

Die verschiedenen Vielfachen eines Grundtaktes spiegeln die unterschiedlichen Abtastraten der Sensoren wieder, die üblicherweise periodisch ausgelesen werden. Bekanntlicherweise muß dabei das Abtasttheorem eingehalten werden.

Der Grundtakt *G* wird von SCINET aus der vorgegeben Datenrate *R* gemäß Gl. 4.4.1 berechnet (*N* ist die Netzgröße). Der kleinste zulässige Grundtakt ist

Gl. 4.4.1:
$$G = \frac{R}{N}$$

1 Byte pro Sekunde. Bei dem größtmöglichen Netz von 256x256 Ein-/Ausgängen muß deshalb *R* mindestens 256 Byte/s betragen. Allgemein gilt $R \geq N$. Die Datenraten r_i der einzelnen Paketgeneratoren *i* (i=1,2,...,N) werden gemäß des Zufallsprinzips einer Menge *M* von Datenraten entnommen, die anhand von Gl. 4.4.2 bestimmt wird.

Gl. 4.4.2:
$$M = \{ r_i \mid r_i = i \cdot G, \text{ für } i = 1, 2, \ldots, N \}$$

Jede Datenquelle erhält genau eines der Elemente von *M* als Datenrate zugeordnet und zwar so, daß alle paarweise verschieden sind, d.h. es gilt:

$r_i \neq r_j$, für $i \neq j$ und $i, j = 1, 2, \ldots, N$. Die Zuordnung von Datenquellen zu Datenraten erfolgt durch SCINET automatisch vor Beginn eines Simulationslaufs und wird mit Hilfe eines gleichverteilten Zufallszahlengenerators durchgeführt, der zusätzlich Gl. 4.4.2 abprüft.

Bei stochastischen Zwischenankunftszeiten der Pakete im Netzwerk wird die Datenrate *R* als Mittelwert der Zwischenankunftszeiten interpretiert, und die Zeitintervalle zwischen zwei Paketen werden während des Simulationslaufs anhand eines exponentiell verteilten Zufallszahlengenerators ausgewählt. In diesem Fall existiert keine feste Phasenbeziehung zwischen den einzelnen Datenquellen.

Die Auswahl der Zieladresse erlaubt ebenfalls verschiedene Varianten und zwar in Abhängigkeit davon, ob man eine feste Paketwiederholrate oder eine stochastische Zwischankunftszeit gewählt hat.

Im deterministischen Fall (feste Paketwiederholrate) gibt es wahlweise entweder genau eine feste Zieladresse, zu der die Pakete aller Generatoren geschickt werden, oder es wird vom Simulator jeder Datenquelle eine andere Datensenke zufällig zugeordnet. Die erste Möglichkeit dient dazu, eine Überlastungsbedingung im Netz zu simulieren (Hot Target), während die zweite Option dem üblichen Betrieb eines Datenerfassungssystems entspricht, bei dem die Sensoren und die Rechner an die Netz-Ports nach einem beliebigen Schema angeschlossen werden können (Distributed Target). Durch die beiden Möglichkeiten werden somit eine N->1 Abbildung (inverser Broadcast), bzw. eine 1:1-Funktion realisiert.

Üblicherweise ist in der Realität ein einmal angeschlossener Sensor über längere Zeit mit demselben Netz-Port verbunden. Das bedeutet für SCINET, daß die im Falle der deterministischen Paketrate zufällig ausgewählten Ziele während der Dauer eines Simulationslaufs konstant bleiben. Die Zuordnung von Datenquellen zu Datensenken erfolgt mit Hilfe eines gleichverteilten Zufallszahlengenerators, der im Falle von Distributed Target abprüft, daß jede Senke genau einmal als Ziel ausgewählt ist. Beim Hot Target-Fall wird ein Ziel vor Beginn der Simulation ausgewürfelt, bei Distributed Target sind es N Ziele.

Im Falle einer stochastischen Zwischenankunftszeit der Pakete kann bei jedem Paketgenerator mit Hilfe eines zusätzlichen Parameters (deterministicTargetFraction) ausgewählt werden, welcher Prozentsatz der Pakete zu einem festen Ziel geschickt werden soll. Der zu 100% verbleibende Prozentsatz der Pakete wird während der Simulation nach dem Zufallsprinzip zu allen anderen Zielen geschickt, die im Netz möglich sind, mit Ausnahme des festen Ziels. Die zur Laufzeit zufällig ausgewählten Ziele sind in ihrer Häufigkeit gleichverteilt. Das feste Ziel wird von SCINET vor Beginn der Simulation nach dem Zufallsprinzip einmal ausgewählt, und bleibt während der Dauer eines Simulationslaufs konstant. Mit Hilfe des deterministicTargetFraction-Parameters kann somit der Grad der Lokalität im Kommunikationsverhalten der Datenquellen berücksichtigt werden. Hohe Lokalität heißt, daß häufig, jedoch nicht immer, dasselbe Ziel ausgewählt wird.

Aufgrund der bei SCINET verwendeten Zufallszahlengeneratoren ist zu be-

achten, daß bei jedem neuen Simulationslauf dieselben Zufallszahlen ausgewürfelt werden, so daß das Verhalten von SCINET reproduzierbar ist. Will man bei einem Simulationslauf andere Zieladressen, muß man von SCINET nacheinander mehrere Netztopologien erzeugen lassen, bevor das erste Netz simuliert wird.

4.5 Modellierung eines SCI-Knotens

In SCINET lassen sich alle SCI-Knoten in die drei Kategorien Datenquellen, Datensenken und Schalter einordnen. Die Modellierung der Knoten aller Kategorien erfolgt über Parameter, die ihre statische und dynamische Eigenschaften wiederspiegeln. Zu den Parametern statischer Eigenschaften zählen beispielsweise Angaben wie Knotenadresse und -typ oder Größe von Sende- und Empfangspuffer, während die dynamischen Parameter das Zeitverhalten anhand von Set-Up-, Verzögerungs- und Durchlaufzeiten spezifizieren.

4.5.1 Statische Knotenparameter

Bei allen Knotenkategorien gibt es eine gemeinsame Grundmenge von Parametern, die fundamentale statische Knoteneigenschaften festlegen. Je nach Kategorie sind jedoch die Werte der Parameter häufig verschieden. Die gemeinsamen Parameter sind (mnemonische Bezeichnung in Klammern):

- die Knotendresse (nodeid),
- der Knotentyp (typeid),
- die Größe des Empfangspuffers (inFifoSize),
- die Größe des Sendepuffers (outFifoSize)
- sowie die Anzahl der SCI-Schnittstellen pro B-Link (SwitchSize).

Knotenadresse

Die Knotenadresse (nodeid) dient zur netzweit eindeutigen Identifikation eines SCI-Knotens. Ihre Aufgabe entspricht der IEEE-SCI-Adresse mit der Einschränkung, daß nicht 64, sondern 16 Adreßbits verwendet werden. Da bei SCINET aus anderen Gründen die Gesamtzahl der zu simulierenden Schalterknoten auf 2048 beschränkt ist, stellt dies jedoch keine Einschränkung dar. Vielmehr stehen auch bei 16 Adreßbit freie Bits zur Kodierung zusätzlicher Informationen zur Verfügung.
Bei SCINET werden Knotenadressen anders als üblich vergeben. Konventio-

nelle Methoden ordnen zur Verfügung stehende Adressen, die innerhalb eines festgelegten Adreßbereichs vergeben werden können, entweder zufällig oder nach dem first-come-first-served-Prinzip den Netzteilnehmern zu. Werden sie der Reihe nach vergeben, spiegeln sie zwar eine Historie wider, haben jedoch sonst keine Semantik. Bei SCINET dagegen kodieren die unteren 11 Adreßbit die Position des Knotens im Netz, und die oberen 5 Adreßbit enthalten zusätzliche Knotenattribute. Die Netzposition wird über eine 8-Bit-Zeilenadresse und eine 3-Bit-Spaltenadresse eindeutig in der x-y-Ebene festgelegt. Die Knotenattribute informieren, ob der Knoten für uni- oder bidirektionale Schalter vorgesehen ist, ob er in einer regulären oder zusätzlichen Netzstufe enthalten ist oder, ob es sich beispielsweise um einen Knoten für Lastausgleich oder für normalen Datentransport handelt. Für den SCINET-Simulator ist die Vereinigung von Knotenposition und Knotenattributen zu einer 16 Bit-"Adresse" wichtig für die effiziente Durchführung seiner adaptiven und deterministischen Routing-Verfahren und hat hohen konzeptionellen Wert.

Knotentypparameter

Der Knotentypparameter (typeid), der nach der Knotenadresse den zweiten allgemeinen Parameter statischer Knoteneigenschaften darstellt, wählt einen bestimmten Knotentypen aus derzeit 20 in SCINET möglichen Typen aus. Jeder Knotentyp ist mit Hilfe eines eigenen MODSIM-Objekts und entsprechenden Vererbungsregeln implementiert. Als Knotentypkodierung werden 32 Bit verwendet, deren Vergabe analog zur Adreßkodierung nach einem eigenen semantischen Schema erfolgt. In diesem Schema ist in verschiedenen Bitfeldern Informationen darüber enthalten, welche Netztopologie mit Hilfe des Knotentyps aufgebaut werden kann, welche Zahlenbasis der Verdrahtung zwischen den Netzstufen benötigt wird, welches Routing-Verfahren im Typ vorhanden ist sowie weitere Informationen. Die einzelnen Bitfelder des Knotentyps stellen ein Klassifizierungsschema für Netze dar, das eine Reihe von Dimension im Designraum der Netze erfaßt. Es hat sich gezeigt, daß durch Variation der Werte eines Bitfeldes rein formal neue Netzstrukturen erzeugt werden können, so daß dem Schema eine gewisse Allgemeinheit zugesprochen werden kann.

Sende- und Empfangspuffer

Die Größe der Sende und Empfangspuffers (in- bzw. outFifoSize) stellt den dritten bzw. vierten allgemeinen Parameter statischer Knoteneigenschaften dar. Ihr Wert ist auf jeweils 4 voreingestellt und somit zur Puffertiefe des Dolphinschen LC-II Link-Controllers [Dolphin97] identisch. Durch Variation der Puffertiefen ergeben sich unterschiedliche Kennwerte des Netzes. Simulationen bei Dolphin und CERN [Wu95a] haben gezeigt, daß bis zu einer Puffertiefe von 4 die größten Leistungszuwächse hinsichtlich des Netzdurchsatzes erzielt werden. Danach flacht die Kurve ab und geht in die Sättigung. Aufgrund der bereits

gemachten Untersuchungen wird dieser Parameter in den SCINET-Simulationen in der Regel nicht variiert.

Port-Zahl

Die Anzahl der SCI-Schnittstellen pro B-Link (SwitchSize) ist der letzte allgemeine Parameter statischer Knoteneigenschaften und enthält die Information, wie viele Ports an dem B-Link des betreffenden Knotens angeschlossen sind. Für Knoten der Kategorie Datenquelle oder Datensenke ist dieser Wert gleich 1, bei Schalterknoten repräsentiert er die Zahl der Ports an dem betreffenden Schalter. Da jeder Netzknoten über diesen Parameter verfügt, ist es möglich, Schalter verschiedener Größe, z.B. solche mit zwei und vier Ports, im Netz unterzubringen.

Die folgenden Modellierungsparameter sind nicht mehr allen drei Knotenkategorien gemeinsam, sondern kategorienspezifisch:

- Bei Schalterknoten gibt es die Knotenadressen (ownNodeid bzw. RemoteNodeids) als Parameter, die die Adressen aller an das B-Link angeschlossenen SCI-Schnittstellen enthalten. Die Sequenz der Adressen spielt bei der Wegewahl eine Rolle und muß bei allen Knoten desselben Schalters gleich sein.
- Bei Datenquellen gibt es drei spezifische Modellierungsparameter:
  - das SCI-Kommando (packettype),
  - die Zieladresse der Transaktion (targetid)
  - und die deterministische bzw. mittlere Senderate (dataRate).
- Bei Datenquellen, die nach zufälligen Zeitintervallen Pakete erzeugen, gibt es einen Parameter (deterministicTargetFraction), der angibt, welcher Prozentsatz der Pakete zur weiter oben angegebenen, deterministischen Zieladresse (targetid) gehen. Die übrigen Prozent der Pakete gehen an zufällig ausgewürfelte Ziele, jedoch nicht an die angegebene Zieladresse.

Das SCI-Kommando-Parameter (packettype) gibt an, welche der von IEEE definierten Transaktionen simuliert werden sollen. Es sind alle Transaktionen zulässig, jedoch werden sie von SCINET auf einer rein formalen Basis durchgeführt. Das bedeutet beispielsweise, daß das Response-Paket, das zu einer nread64-Transaktion gehört, leer ist, d.h. keine Daten enthält, da der an die betreffende SCI-Schnittstelle angeschlossene Rechner nur hinsichtlich seiner Reaktionszeit, aber nicht hinsichtlich seines Hauptspeichers oder anderer Komponenten oder Funktionen simuliert wird.

Der Parameter für die Zieladresse der Transaktion (targetid) enthält eine der Knotenadressen, die im zu simulierenden System vorkommen, An sie wird das Request-Paket der Transaktion geschickt.

Der Parameter für die deterministische bzw. mittlere Senderate (dataRate) gibt an, nach welchem Zeitintervallen exakt oder im Mittel eine neues Request-Paket ausgegeben wird. Bei Datenquellen, die Pakete nach zufälligen Zeitintervallen erzeugen, wird hier der Mittelwert der Zwischenankunftszeit (Interarri-

val Time) der Pakete angegeben. In Tabelle 4.5.1 sind die zur Modellierung einer SCI-Schnittstelle und ihrer nachgeschalteten Recheneinheit notwendigen statischen Parameter, ihre mnemonische Bezeichnung und Voreinstellung zusammengefaßt.

Parameter	Name	Default	Einheit
Knotendresse	nodeid	-	-
Knotentyp	typeid	-	-
Empfangspuffer	inFifoSize	4	Plätze
Sendepuffers	outFifoSize	4	Plätze
Port-Zahl	SwitchSize	-	-
Port-Adressen	RemoteNodeids	-	-
Kommando	packettype	NWRITE64	-
Zieladresse	targetid	-	-
Senderate	dataRate	100	MB/s
festesZiel	det.TargetFraction	100	%

Tabelle 4.5.1: Statische Parameter eines SCI-Knotens.

Die Voreinstellungen (Default-Werte) wurden anhand der Angaben in [Dolphin97] vorgenommen. Dadurch ist SCINET bzgl. seiner Default-Werte auf die Simulation von Netzen ausgerichtet, die auf dem Link-Controller LC-II basieren. Da alle Simulationsparameter über die graphische Benutzerschnittstelle modifiziert werden können, kann bei Bedarf auch eine andere Schnittstellen-Hardware nachgebildet werden.

4.5.2 Dynamische Knotenparameter

Die Spezifikation der dynamischen Parameter eines SCI-Knotens erfolgt anhand der von IEEE standardisierten SCI-Schnittstelle [IEEE92] sowie zusätzlicher Zeitinformationen, die im Standard nicht vorgegeben sind. Die Zeitinformationen reflektieren die jeweilige konkrete Implementierung der Schnittstelle in einer bestimmten Technologie, wie z.B. CMOS, BiCMOS oder GaAs. Deren Werte können als Eingabeparameter vom Benutzer von SCINET frei gewählt werden. Die Parameter geben Auskunft über das Zeitverhalten der Implementierung der Transportebene von SCI. Zur Definition dieser Eingabeparameter dienen im folgenden 12 Koordinatenpunkte (1-12), deren Positionen in der SCI-Schnittstelle gemäß Bild 4.5.1 angegeben sind.
Bei der Modellierung des Zeitverhaltens wurde angenommen, daß die der SCI-Schnittstelle nachfolgende Rechnereinheit (PC, Arbeitsplatzrechner, Periphe-

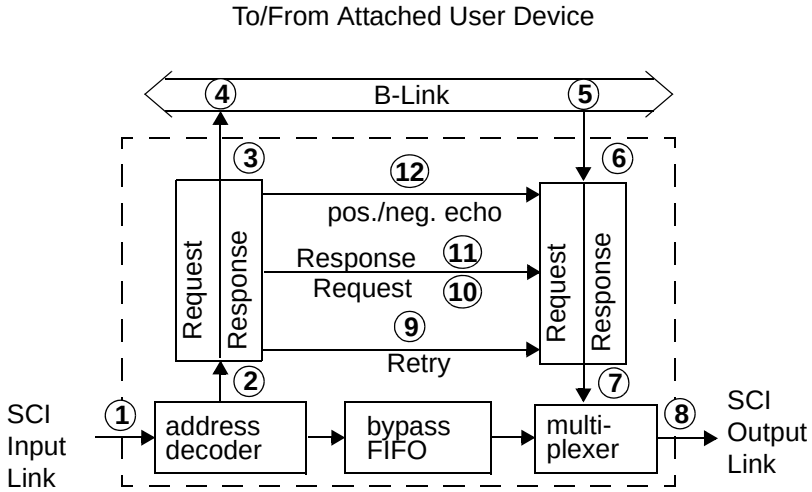

Bild 4.5.1: SCI-Schnittstelle mit den ausgezeichneten Koordinatenpunkten 1-12.

rie, Speicher, etc.), die zusammen mit der Schnittstelle den SCI-Knoten bildet, über einen intermediären Bus angekoppelt ist. Dieser Bus entspricht im Falle von SCINET dem sog. B-Link 91 [Dolphin94a] der Fa. Dolphin. Die Ausrichtung an einem speziellen Bussystem stellt an dieser Stelle keine Einschränkung der Simulationsfähigkeiten von SCINET dar, da über eine geeignete Wahl der Zeitparameter des B-Links auch Busse anderer Hersteller modelliert werden können. Die Wahl fiel deshalb auf den B-Link-Bus, weil Dolphin der zur Zeit einzige Anbieter ist, der SCI-Schnittstellen kommerziell vertreibt. Deren Produkt, das sog. Link Chip LC1 bzw. LC2 [Dolphin95][Dolphin97], verwendet das „Backside-Link“.

Die Modellierung des Zeitverhaltens der SCI-Schnittstelle spiegelt ihre funktionelle Arbeitsweise wieder. Erreicht ein Datenpaket auf einem Ring eine Schnittstelle am Koordinatenpunkt 1, wird als erstes vom Adreßdekoder der Schnittstelle die Paketzieladresse inspiziert und mit der lokalen SCI-Adresse verglichen. Stimmen beide Adressen überein, erscheint das Paket am Eingang des Knotenempfangspuffers (Koordinatenpunkt 2). Die Zeit, die vom Knoteneingang bis zum Empfangspuffereingang verbraucht wird, wird als $T_{1,2}$ (AddressDecoderDelay) bezeichnet. Stimmen die SCI-Adressen nicht überein, wird das Paket über den Bypass-Fifo und den Multiplexer zum Ausgang der Schnittstelle weitergeleitet. Die Durchlaufzeit vom Eingang zum Ausgang heißt $T_{1,8}$ und trägt die mnemonische Bezeichnung bypassDelay. Wichtig ist festzustellen, daß im bypassDelay nicht nur die Durchlaufzeit des Bypass-Fifos selbst sondern auch des Adreßdekoders sowie des Ausgangsmultiplexers subsumiert werden. Dies ist eine Abstraktion im Rahmen der Knotenmodellierung, die die weitere Behandlung des Zeitverhaltens vereinfacht, ohne die Simulationsgenauigkeit zu beeinträchtigen, da die reine Bypass-Fifo-Durchlaufzeit als Einzelgröße nicht interessiert.

Wird das Datenpaket im Empfangspuffer eingespeichert, dauert es eine un-

bestimmte Zeit $T_{2,3}$, bis das Paket von dem B-Link-Anschluß der eigenen Schnittstelle zum B-Link-Anschluß der nachfolgenden Rechnereinheit übertragen und dadurch dem Empfangspuffer entnommen wird. $T_{2,3}$ beinhaltet die Dynamik des zeitlichen Verlaufs des Pufferfüllgrades und kann deshalb nicht über einen festen Wert modelliert werden. Fest ist hingegen die Zeit $T_{3,4}$ (InFIFOOutToBLinkDelay), die angibt, wie lange es dauert, bis das erste Byte des eingespeicherten Pakets auf dem B-Link erscheint, vorausgesetzt, daß erstens das B-Link frei ist und daß zweitens die setup-Zeit des Links gleich null ist. Beide Vorraussetzung sind im Allgemeinfall nicht gegeben, so daß sich in der Regel zum InFIFOOutToBLinkDelay noch beide Zeiten hinzuaddieren. Aus Vereinfachungsgründen wird in SCINET wie in Kapitel 4.7.3 "Modellierung des zeitlichen B-Link-Verhaltens" dargestellt, die B-Link-Arbitrierungszeit und seine setup-Zeit zu einer gemeinsamen „Setup-Zeit“ zusammengefaßt. Befindet sich ein Paket auf dem B-Link, kann es an alle daran angeschlossenen Ports übertragen werden.

Auf der Ausgabeseite der SCI-Schnittstelle wird für die Modellierung die Zeit $T_{5,6}$ (BLinkToOutFIFOInDelay) benötigt, die über die Dauer eines Simulationslaufs konstant ist und die angibt, wie lange es dauert, bis das erste Byte eines Pakets vom B-Link in den Ausgabepuffer eingespeichert ist. Die B-Link Setup-Zeit muß hier nicht berücksichtigt werden, weil sie bereits vom schreibenden B-Link-Anschluß abgewartet wurde, ebenso entfällt die B-Link-Arbitrierungszeit. Analog zum Eingabepuffer verbleibt das Paket auch im Ausgabepuffer eine nicht vorhersagbare Zeit $T_{6,7}$, die in diesem Fall davon abhängt, wann die Schnittstelle Zugang zum SCI-Ring erhält. Der Ringzugang wird von den Bandbreiteallozierungsprotokollen von SCI geregelt, die im Kapitel 2.4 "SCI-Operationen und Datenformate" beschrieben wurden. Als weitere Abstraktion in der Modellierung des Zeitverhaltens des Knotens wird im Simulator die Durchlaufzeit $T_{7,8}$ durch den Ausgangsmultiplexer und die Pufferverweildauer $T_{6,7}$ zu einer einzigen Zeit zusammengefaßt. Dies stellt wiederum keine Einschränkung dar, da $T_{7,8}$ alleine nicht interessiert.

Schließlich muß in einer SCI-Schnittstelle die Abwicklung des Handshakes in Form von positiven oder negativen Echo- und/oder Retry-Paketen bezüglich des zeitlichen Verhaltens modelliert werden. Dazu dienen die Zeitangaben T_9 bis T_{12}. Das RetryDelay T_9 gibt an, wie lange es nach Eintreffen eines negativen Echos am Empfangspuffer einer Schnittstelle dauert, bis von dieser versucht wird, ein Retry-Paket auf dem Ring auszugeben. Erhält die Schnittstelle sofort Zugang zum Ring, erscheint unmittelbar nach der Zeit T_9 das Retry-Paket auf dem Ring. Im Allgemeinfall vergeht jedoch aus Ring-Arbitrierungsgründen mehr als die Zeit T_9 bis zur Aussendung des ersten Retries, und auch alle evtl. nachfolgenden Retry-Pakete desselben Inhalts benötigen in der Regel mehr Zeit als T_9. Der genaue Wert ist jedoch nicht vorhersagbar.

Die Zeit T_{10} (RequestDelay) gibt an, wie lange es nach Eintreffen eines Requests dauert, bis ein dazu gehörendes Response-Paket vom Knoten formuliert

worden ist. Analog dazu kennzeichnet T_{11} (ResponseDelay) die Bearbeitungszeit eines eingetroffenen Response-Paketes.

T_{12}, der letzte Parameter, der zur Modellierung des Zeitverhaltens eines Knotens dient, kann vom Benutzer nicht vorgegeben werden, vielmehr ist er fest auf den Wert 0 gesetzt. D.h., daß nach Ablauf der Adreßdekodierungszeit $T_{1,2}$ keine weitere Zeit vergeht, bis der Knoten versucht, ein positives oder negatives Echo-Paket auf dem Ring auszugeben. Wiederum entscheiden die SCI-Bandbreiteallozierungsprotokolle und die momentane Ringbelegung über den tatsächlichen Ausgabezeitpunkt.

In Tabelle 4.5.2 sind die dynamischen Parameter zusammengefaßt. Wieder-

Parameter	Name	Default	Einheit
$T_{1,2}$	AddressDecoderDelay	20	ns
$T_{1,8}$	bypassDelay	48	ns
$T_{3,4}$	InFIFOOutToBLinkDelay	106	ns
$T_{5,6}$	BLinkToOutFIFOInDelay	82	ns
T_{9}	RetryDelay	const 1	ns
T_{10}	RequestDelay	0 bzw. 40	ns
T_{11}	ResponseDelay	0 bzw. 40	ns

Tabelle 4.5.2: Dynamische Parameter eines SCI-Knotens.

um wurden die Default-Werte anhand der Angaben in [Dolphin97] gewählt.

4.6 Modellierung eines SCI-Ringes

Ein SCI-Ringe wird dadurch modelliert, daß man die Ringgeschwindigkeit (ringSpeed) angibt, die Art und Anzahl der Knoten, die im Ring enthalten sind sowie deren räumliche Entfernung (linkDelay) zueinander. Die Art der Knoten teilt man dazu in die drei Kategorien Datenquellen, Schalteranschlüsse und Datensenken ein. Bei der Spezifikation der Knoten ist die Sequenz in der textuellen Auflistung wichtig. Sie muß Knotenfolge im zu simulierenden Ring entsprechen und spiegelt den Umlaufsinn der Datenpakete wider. Die räumliche Entfernung zwischen je zwei benachbarten Knoten wird anhand der Signallaufzeiten (linkDelay) konfiguriert. Jeder Knoten hat die Laufzeit zu seinem nächsten Nachbarn in fortschreitender Umlaufrichtung des Ringes als Parame-

ter. Der Default-Werte für die Ringgeschwindigkeit ist 500 MB/s, für die Signallaufzeit werden 100 bzw. 1 ns gewählt.

4.7 Modellierung eines SCI-Schalters

SCI-Schalter entstehen dadurch, daß mehrere SCI-Schnittstellen auf der Seite ihres intermediären Busses nicht mit nachfolgenden Recheneinheiten wie PC, Arbeitsplatzrechner oder der Peripherie gekoppelt werden, sondern mit sich selbst. In Bild 4.7.1 ist das symbolische Blockschaltbild einer SCI-Schnittstelle gezeigt sowie deren Kopplung zu einem bidirektionalen 4x4-Schalter.

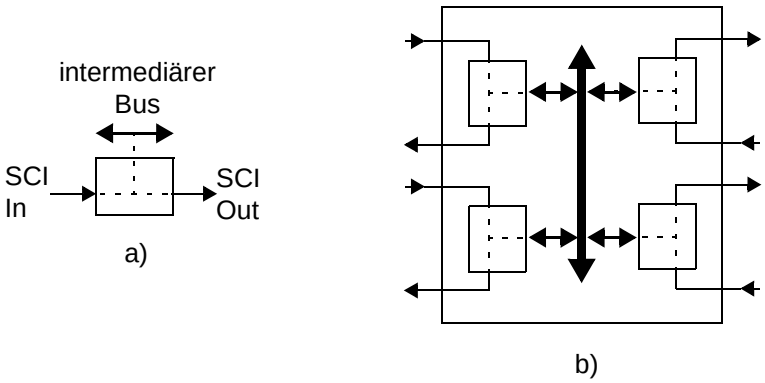

Bild 4.7.1: Symbolische Darstellung einer SCI-Schnittstelle und eines SCI-Schalters.

Als intermediärer Bus wird bei SCINET gemäß den in Kapitel 4.5 "Modellierung eines SCI-Knotens" erörterten Gründen das Dolphinsche B-Link [Dolphin94a] gewählt. Die Konzeption des B-Link-Busses sowie des Dolphinschen Link-Controllers, das über diesen B-Link-Anschluß verfügt, erlaubt, bis zu 14 Link-Controller [Dolphin97] ohne zusätzlichen Hardware-Aufwand miteinander zu verbinden, so daß auf einfache Art Schalter mit bis zu 14 Ports aufgebaut werden können. Aufgrund der Tatsache, daß das B-Link nicht wesentlich schneller als ein einzelnes SCI-Link arbeitet, sind in der Praxis nur Schalter mit einer kleinen Zahl von Ports auf diese Weise realisierbar.

Der LC-I Link-Controller-Baustein hat eine B-Link-Geschwindigkeit von 500 MB/s und eine SCI-Link-Geschwindigkeit von 200 MB/s. Damit sind ohne Bandbreiteverlust nur zwei Link-Controller vom Typ LC1 koppelbar. Im Falle des LC-II-Bausteins sind die Verhältnisse noch ungünstiger, da 500 MB/s SCI-Link-Geschwindigkeit 600 MB/s B-Link-Geschwindigkeit gegenüberstehen. In Kapitel 8.3 "Durchsatzerhöhung im Schalter" wird gezeigt, wie man die sehr niedrige Zahl vernünftig koppelbarer Link-Controller durch andere Maßnahmen erhöht.

4.7.1 Graphische Äquivalenztransformation

Ein bidirektionale Vier-Port-Schalter gemäß Bild 4.7.1 kann über eine graphische Äquivalenztransformation in ein unidirektionales 4-Port-Schaltelement umgewandelt werden (Bild 4.7.2). Allgemein gilt, daß sich jeder bidirektionale SCI-Schalter ohne weitere Modifikation auch als Bauelement für unidirektionale Netze eignet, er muß nur geeignet verdrahtet werden.

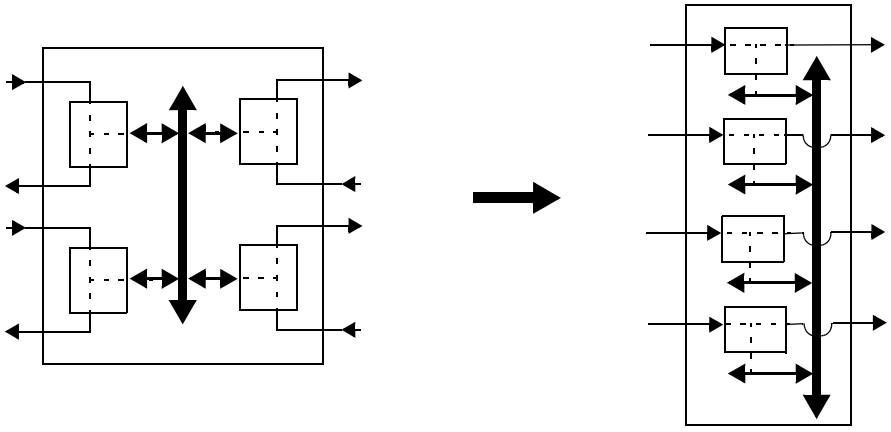

Bild 4.7.2: Graphische Äquivalenztransformation vom bi- zum unidirektionalen Schalter.

Im Mittelpunkt der Modellierung eines SCI-Schalters steht die Beschreibung des B-Links hinsichtlich seines funktionalen und zeitlichen Verhaltens, denn dieses bestimmt wesentlich die Bandbreite und Latenz des Schalters. Zur Beschreibung des funktionalen Verhaltens ist festzustellen, daß das B-Link ein Bus ist, der nur Schreiboperationen durchführen kann. Um ein Lesen zu ermöglichen, wird jede Busoperation, die analog zur SCI-Transaktion B-Link-Transaktion heißt, in eine Request- und eine Response-Phase zerlegt. zuerst gibt der B-Link Anforderer (Requester) seinen Lesewunsch auf dem Bus aus, worauf der Beantworter (Responder) die Antwort auf den Bus zurückschreibt. Zwischen Request- und der Response-Phase kann ein beliebig langer Zeitraum liegen. Für die Beschreibung des zeitlichen Verhaltens muß beachtet werden, daß sich die B-Link-Zeiten zu den im Kapitel 4.5 "Modellierung eines SCI-Knotens" beschriebenen Zeitparametern hinzuaddieren.

Im weiteren Verlauf dieses Kapitels soll zuerst die funktionale und danach die zeitliche Beschreibung des B-Link-Verhaltens sowie deren Modellierungen vorgenommen werden.

Beschreibung des funktionalen B-Link-Verhaltens

Die wichtigste Eigenschaft eines B-Links ist dessen Multimaster-Fähigkeit, die in Form einer dezentralen Zugriffsarbitrierung implementiert ist. Um diese Eigenschaft korrekt zu modellieren, muß man zuerst verstehen, wie sie abläuft.

Multimaster heißt, daß alle an das B-Link angeschlossenen Einheiten für eine begrenzte Zeit Bus-Master sein können, wobei der Buszugang ohne eine übergeordnete Hardware-Instanz geregelt wird. Zum Anmelden eines Buszugriffswunschs verfügt jede der max. 14 an das B-Link anschließbaren Einheiten über eine sog. RequestOut- sowie 14 RequestIn-Leitungen. (In einfacheren Implementierungen eines B-Links für bis zu max. 8 anschließbare Einheiten sind 8 RequestIn-Leitungen ausreichend.) Die Verschaltung zwischen RequestIn- und RequestOut-Leitungen ist dergestalt, daß die RequestOut-Leitung Nr. i mit 14 korrespondierenden RequestIn-Leitungen derselben Nummer i verbunden ist (i=0,1,2...,13), so daß alle Einheiten gleichzeitig erfahren, welche RequestOut-Leitungen im System aktiviert wurden, d.h., welche Einheiten einen Buszugriff angemeldet haben. Der gleiche Informationsstand aller Busteilnehmer ist Voraussetzung für die dezentrale Buszuteilung, die von den Einheiten in Eigenverantwortung durchgeführt wird.

Eine Busarbitrierung beginnt damit, daß die RequestOut-Leitung mindestens einer der B-Link-Teilnehmer aktiv wird, vorausgesetzt, daß zuvor eine sog. Arbitrierungs-Idle-Phase voranging, in der alle RequestOut-Leitung inaktiv waren. Werden nach einer Arbitrierungs-Idle-Phase mehrere RequestOut-Leitungen gleichzeitig aktiv, erhält diejenige Einheit den Bus, die die höchste Priorität hat. Prioritäten sind statisch anhand der 4-Bit Adresse, die die 14 anschließbaren Einheiten unterscheiden, festgelegt. Die Einheit mit der Adresse 0 hat die höchste, diejenige mit der Adresse 13 hat die niedrigste Priorität.

Mit der Zuteilung des Busses an die Einheit, die den Zugriffswunsch mit der höchsten Priorität angemeldet hat, beginnt ein sog. Arbitrierungsintervall, währenddessen alle Einheiten mit Zugriffswünschen je einmal den Bus bekommen. Die Priorität einer Einheit wirkt sich nur darauf aus, ob diese am Anfang oder am Ende eines Arbitrierungsintervalls den Buszugang erhält. Die Länge des Arbitrierungsintervalls entspricht der Zahl der an das B-Link angeschlossenen Einheiten, maximal gibt es 14 verschiedene Zeitscheiben, in denen jeweils diejenige Einheit Busmaster ist, die die momentan die höchste Priorität hat.

Hat eine Einheit ihren Buszugriff beendet, wird die Rolle des Bus-Masters anschließend zu der Einheit mit der nächstniedrigeren Priorität weitergereicht, solange bis alle Zugriffswünsche befriedigt sind. Abgesehen von einer zusätzlichen, rein prioritätsgesteuerten Arbitrierungsweise, die auch wählbar ist, kann deshalb im Normalfall von einer fairen Buszuteilung gesprochen werden, weil keine Einheit während des Arbitrierungsintervalls vom Buszugang ausgeschlossen bleibt. Das beschriebene B-Link-Arbitrierungsschema entspricht deshalb zusammengefaßt formuliert dem bekannten Round Robin Scheduling.

4.7.2 Modellierung des funktionalen B-Link-Verhaltens

Aus der Beschreibung des funktionalen Verhaltens ergibt sich daß eine hinreichend genaue Modellierung des B-Links auf einer Hardware-Ebene aufwendig ist, weil asynchrone Zustandsänderungen (low -> high bzw. high -> low) von

einer größeren Zahl von Signalen festgestellt werden müßten, die nebenläufig zu anderen Prozessen erfolgen. Deswegen wurde beispielsweise im Simulationsprogramm SCILAB von [Bogaerts94b] und [Wu95a] auf das erforderliche Round Robin-Arbitrierungsschema verzichtet und statt dessen das wesentlich einfacher zu implementierende First-Come-First-Served-Scheduling verwendet, was jedoch einen anderen Durchsatz und einer andere Latenzzeit des Schalters zur Folge hat.

Die Frage, die sich hier stellt, ist, wie ein B-Link möglichst genau modelliert werden kann ohne daß in der nachfolgenden Implementierung zuviel CPU-Zeit verbraucht wird. Dazu muß die konkrete Implementierungssprache näher betrachtet werden, die im Falle von SCINET MODSIM ist. Die Feststellung von Pegeländerungen bei Signalen würde in einer anderen Programmiersprache als MODSIM [CACI95] eine Endlosschleife benötigen, in der permanent die Absolutwerte der Signalvariablen abgefragt werden müßten, was einen hohen Rechenzeitbedarf zur Folge hätte. Bei MODSIM und anderen Sprachen zur ereignisabhängigen Simulation gibt es die Möglichkeit, Prozesse von der Liste rechenbereiter Programme zu entfernen und aufgrund von Trigger-Signalen anderer Prozesse der Liste wieder hinzuzufügen. Dazu existiert in MODSIM das sprachliche Konstrukt „TRIGGER“. Mit Hilfe eines TRIGGERs und der korrespondierenden WAIT-FOR-Anweisung kann ein De- bzw. Re-Scheduling einer Prozedur erreicht werden, ohne daß wesentlich CPU-Zeit verbraucht wird.

Trotz dieser sprachlichen Hilfsmittel erfordert die Round Robin-Arbitrierung, die im jedem B-Link durchgeführt wird, eine effiziente Implementierung, da in einem größeren Netz eine Vielzahl von Schaltern mit ebensovielen B-Links, d.h. Schedulern existieren, so daß die benötigte Rechenzeit schnell zu hoch wird. Eine Implementierung der Arbitrierung mit Hilfe von Listen, die beispielsweise die Unterteilung der B-Link-Ports in solche mit und ohne Zugriffswünsche verwalten, ist ineffizient, weil die Verwaltung und Manipulation der Listen viel Zeit kostet.

Für SCINET wurde als Ausweg ein endlicher Automat konzipiert, der das prioritätsgesteuerte Round Robin-Schema „in Hardware“ realisiert. Die Implementierung der B-Link-Arbitrierung wird mit dieser Maßnahme auf die Simulation eines endlichen Automaten reduziert. Jedem B-Link ist ein solcher Automat zugeordnet. Dadurch kann auf Listen ganz verzichtet werden. Die Abstraktion, die bei der Modellierung eines B-Links gemacht wird, ist somit, das dezentrale Buszuteilungsschema zentral mit Hilfe eines endlichen Automaten zu bewerkstelligen, der seine Zustandsübergänge bei Eintreffen asynchroner Trigger durchführt, die ihrerseits die Buszugriffswünsche der an das B-Link angeschlossenen Einheiten darstellen.

Der konzipierte endliche Automat eines B-Link-Busses, der in einem SCI-Schalter zur Kopplung von z.B. vier Link-Controller-Bausteinen verwendet wird, verwaltet im wesentlichen die vier Zustände ServePort1-ServePort4, die kennzeichnen, welcher Link-Controller momentan Busmaster ist. In Tabelle 4.7.1 sind die Zustandsübergänge des endlichen Automaten angegeben. Die Abkürzungen r1-r4 deuten dabei an, daß ein Buszugriffswunsch (Request) von einem der vier angeschlossenen Einheiten vorliegt. Der Anfangszustand des

		Von Zustand			
		ServePort1	ServePort2	ServePort3	ServePort4
Nach Zustand	ServePort1	r2∧r3∧r4	$\overline{r3}\wedge\overline{r4}\wedge r1$	$\overline{r4}\wedge r1$	r1
	ServePort2	r2	$\overline{r1}\wedge\overline{r3}\wedge\overline{r4}$	$\overline{r4}\wedge\overline{r1}\wedge r2$	$\overline{r1}\wedge r2$
	ServePort3	r2∧r3	r3	$\overline{r1}\wedge\overline{r2}\wedge\overline{r4}$	$\overline{r1}\wedge\overline{r2}\wedge r3$
	ServePort4	r2∧r3∧r4	$\overline{r3}\wedge r4$	r4	$\overline{r1}\wedge\overline{r2}\wedge\overline{r3}$

Tabelle 4.7.1: Zustandsübergänge des endlichen Automaten für die B-Link Arbitrierung.

Automaten nach „Einschalten“ ist der Zustand ServePort1.

Ein Beispiel der Arbitrierung eines B-Links mit Hilfe des endlichen Automaten zeigt Tabelle 4.7.2. Darin sind links die Zugriffswünsche (Requests bzw. r) und rechts die erfolgten Buszuteilungen (grants bzw. g) eingetragen, die zu einer bestimmten Simulationszeit erfolgen. Arbitrierungsintervalle sind mit einer doppelten Linie abgegrenzt. Deutlich ist das Round Robin-Verhalten erkennbar, das die Ports nach aufsteigender Nummer sortiert, wenn entsprechende Zugriffswünsche vorliegen. Zu beachten ist noch, daß im betrachtetem Beispiel von keinem Port mehr als ein Zugriffswunsch ausgeht, bevor die korrespondierende Buszuteilung erfolgt. Im Allgemeinfall sind jedoch auch multiple Zugriffswünsche von jedem Port zulässig, die vor einer Buszuteilung vom Automaten zwischengespeichert werden.

Beschreibung des zeitlichen B-Link-Verhaltens

Der B-Link-Bus ist von der Fa. Dolphin auf maximalen Durchsatz ausgelegt und deshalb vollzieht sich seine dezentrale Arbitrierung in einer pipeline-Betriebsweise parallel zur Übertragung von Daten, die jeweils zum vorangegangenen Arbitrierungsintervall gehören. Dadurch sind zwar die Datenleitungen des B-Link-Busses optimal ausgenutzt, da keine Pausen zwischen den Datenübertragungen entstehen, die Beschreibung seines zeitlichen Verhaltens ist jedoch aufwendig. Zur weiteren Komplexitätssteigerung trägt die Tatsache bei, daß es eine Reihe unterschiedlicher Varianten der B-Link-Arbitrierung gibt (Idle-cycle arbitration, release-cycle arbitration etc.) sowie eine spezielle Retry-Möglichkeit eines an ein B-Link angeschlossenen Teilnehmers, sobald dessen Zielteilnehmer temporär nicht datenaufnahmebereit ist.

Einfacher stellen sich die Verhältnisse nach der Arbitrierung dar, wenn die Daten auf den 64-Busleitungen des B-Links übertragen werden. Dazu wird das zu übertragende SCI-Request- oder Response-Paket in einen Datenvor- und Nachspann eingekapselt. Ein NWRITE64-Befehl beispielsweise wird durch die Kapselung von 80 auf 88 Byte und dessen Response-Paket von 16 auf 24 Byte

Port 1	Port 2	Port 3	Port 4	Sim Time
r	r	r	r	0
-	r	-	-	30
r	-	-	-	52
-	-	r	-	60
-	r	-	-	82
-	-	-	r	90
r	-	-	-	91
-	-	r	-	112
-	r	-	-	121
r	-	-	-	130
-	-	-	r	168
-	-	r	-	190
-	-	-	r	233
Σ4r	Σ4r	Σ4r	Σ4r	

a)

Port 1	Port 2	Port 3	Port 4	Sim Time
-	g	-	-	0
-	-	g	-	13
-	-	-	g	26
g	-	-	-	39
-	g	-	-	52
-	-	g	-	65
g	-	-	-	78
-	g	-	-	91
-	-	-	g	104
g	-	-	-	117
-	g	-	-	130
-	-	g	-	143
g	-	-	-	156
-	-	-	g	169
-	-	g	-	190
-	-	-	g	233
Σ4g	Σ4g	Σ4g	Σ4g	

b)

Tabelle 4.7.2: Beispiele für eine B-Link-Arbitrierung durch den endlichen Automaten.

verlängert. Das Format das Vor und Nachspanns und damit die Länge eines beliebigen SCI-Pakets, das über ein B-Link transferiert wird, ist in Bild 4.7.3 dargestellt.
Soll ein Datenpuffer bestehend aus einer größeren Zahl von Bytes (>64) auf einem B-Link übertragen werden, sind dazu mehrere SCI-Pakete notwendig, die nacheinander transferiert werden. Als SCI-Pakettyp kommt entweder das NWRITE64- oder das DMOVE64-Kommando in Frage, das mit bzw. ohne Response ist. Datenpuffer kleinerer Größe (zwischen 16 und 64 Byte) werden bei SCI sinnvollerweise mit Hilfe von 1-4 WRITESB-Pakete übertragen, um dadurch weniger Verwaltungszusatzaufwand zu erhalten.

SCI kennt von der Konzeption her nur Pakete jedoch keine im Speicher zusammenhängenden Pufferbereiche, so daß die Zerlegung der Pufferinformation in Pakete von demjenigen Rechner vorgenommen werden muß, der an die sendende SCI-Schnittstelle angeschlossen ist. Dementsprechend werden in SCI-

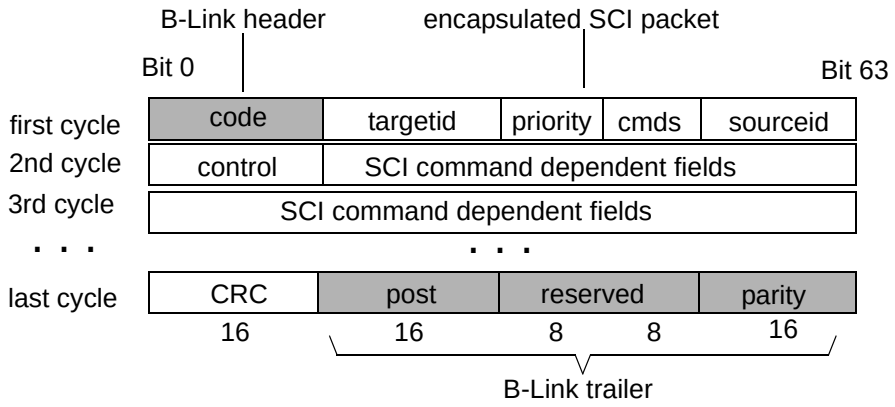

Bild 4.7.3: Kapselung eines SCI-Pakets auf dem B-Link durch die grau hinterlegten Bereiche.

NET Pakete und nicht komplette Speicherbereiche verschickt, allerdings kann der Pakettyp (NWRITE16, 64 etc.) als Parameter vorgegeben werden. Die Zergliederung eines Puffers muß sich nach dem von SCINET gewählten Pakettyp richten. Der Puffer wird in der Regel mittels WRITESB, NWRITE64 oder DMOVE64-Pakete übertragen, die 16 bzw. 64 Byte pro Paket auf dem B-Link transferieren. Daraus erhält man die erforderlich Anzahl N_{Pak} von zu transferierenden Paketen gemäß Gl. 4.7.1, wobei L_{Buf} die Länge des Puffers und $L_{PakNutz}$ die Nutzlast des SCI-Pakets ist. ($L_{PakNutz} \in \{16, 64\}$).

Gl. 4.7.1:
$$N_{Pak} = L_{Buf} \,\mathrm{DIV}\, L_{PakNutz} + r_1, \text{ mit}$$

$$r_1 = \begin{cases} 0 \text{ für } L_{Buf} \,\mathrm{MOD}\, L_{PakNutz} = 0 \\ 1 \text{ für } L_{Buf} \,\mathrm{MOD}\, L_{PakNutz} > 0 \end{cases}$$

Durch die hardware-mäßig notwendige Quantisierung der SCI-Daten in 8-Byte-Portionen, die pro B-Link-Zyklus übertragen werden, ergibt sich für die Zyklenzahl Z, die für die Übertragung eines einzelnen eingekapselten Pakets notwendig ist, ein Wert nach Gl. 4.7.2. Darin ist L_{PakGes} die Gesamtlänge eines SCI-Pakets gemäß SCI-Spezifikation und B_{over} der Verwaltungszusatzaufwand, der für die Einkapselung anfällt.

Gl. 4.7.2:
$$Z = (L_{PakGes} + B_{over}) \,\mathrm{DIV}\, 8 + r_2, \text{ mit}$$

$$r_2 = \begin{cases} 0 \text{ für } (L_{PakGes} + B_{over}) \,\mathrm{MOD}\, 8 = 0 \\ 1 \text{ für } (L_{PakGes} + B_{over}) \,\mathrm{MOD}\, 8 > 0 \end{cases}$$

Bei einem NWRITE64-Paket beispielsweise ist L_{PakGes}= 80, und B_{over} berech-

net sich gemäß Gl. 4.7.3. Mit der B-Link-Übertragungsrate s (BLinkSpeed), die

Gl. 4.7.3: $$B_{over} = const = 8 .$$

in MB/s gemessen wird und in SCINET als Parameter vorgebbar ist, ergibt sich die Zeit T_{TrPak}, die für den reinen Transfer eines Paket ohne B-Link-Arbitrierung erforderlich ist, gemäß Gl. 4.7.4. Die Transferzeit T_{TrBuf} für die Übertra-

Gl. 4.7.4: $$T_{TrPak} = \frac{Z}{s/(8Byte)} .$$

gung des kompletten Puffers berechnet sich ebenfalls ohne B-Link-Arbitrierung anhand von Gl. 4.7.5.

Gl. 4.7.5: $$T_{TrBuf} = \frac{Z \cdot N_{Pak}}{s/(8Byte)} .$$

4.7.3 Modellierung des zeitlichen B-Link-Verhaltens

Aufgrund der vorangegangenen Beschreibung des zeitlichen B-Link-Verhaltens wurde ersichtlich, daß die Timing-Diagramme der B-Link-Arbitrierung für eine 100% exakte Modellierung zu aufwendig sind, insbesondere im Hinblick auf die dabei zu erwartende Genauigkeitserhöhung, so daß unter SCINET die Timing-Diagramme subsummarisch zu einem einfacheren Zeitmodell zusammengefaßt werden, das vom endlichen Automaten des B-Links realisiert wird. In dieses abstrahierte Modell geht die Gesamtlänge des zu übertragenden SCI-Pakets incl. Overhead, die B-Link-Übertragungsrate s sowie eine Setup-Zeit T_{setup} ein.

Die Datenlänge wird durch das gewünschte SCI-Kommando vorgegeben, die Übertragungsgeschwindigkeit und die Setup-Zeit können als Parameter frei gewählt werden. Die Setup-Zeit repräsentiert die Zeitverzögerung zwischen der Absicht eines B-Link-Ports, Daten auf dem B-Link auszugeben, und dem tatsächlichen Beginn des Transfers. D.h., T_{setup} faßt alle bei der B-Link-Arbitrierung erstehenden Zeiten inclusive eines optionalen Retries zu einem einzigen Parameter zusammen. Die Zeit T_{BLink}, die aufgrund des B-Link-Transfers eines Pakets entsteht, ergibt sich somit zu:

Gl. 4.7.6: $$T_{BLink} = Z \cdot T_{setup} + T_{TrPak}.$$

Die gemäß Gl. 4.7.6 bestimmte B-Link-Zeit resultiert in einer netto B-Link-Übertragungsgeschwindigkeit s_{BLink} von:

Gl. 4.7.7:

$$s_{BLink} = \frac{L_{PakNutz}}{T_{BLink}} .$$

Beispiel:

Für s = 600 MB/s und T_{setup} = 6 ns ergibt sich für den Datentransfer eines NWRITE64-Pakets eine Nettodatenrate von s_{BLink}= 301 MB/s, also der halbe maximale Wert. Darin ist jedoch die Zeit für den Transfer des Response-Pakets noch nicht berücksichtigt. Ein kompletter Puffer der Länge L_{Buffer}= 130 Byte, der mittels des DMOVE64-Befehls übertragen wird, erreicht eine Datenrate von s_{BLink}= 204 MB/s.

In Tabelle 4.7.3 sind die zur Modellierung eines B-Links notwendigen Parameter, ihre mnemonische Bezeichnung sowie Voreinstellung zusammengefaßt. Die Voreinstellungen wurden anhand der Angaben in [Dolphin94a] gewählt.

Parameter	Name	Default	Unit
s	BLinkSpeed	600	MB/s
T_{setup}	SetupTime	1 bzw. 6	ns

Tabelle 4.7.3: B-Link Modellierungsparameter.

Die Setup-Zeit ist 6 ns bei SCI-Schaltern und 1 ns bei SCI-Quellen und Zielen.

4.8 Modellierung eines SCI-Netzes

Ein komplettes Netz ist in SCINET durch eine Vielzahl von Parametern spezifiziert. Diese kann man in die Kategorien Systemparameter, Simulationsparameter, Wegewahlparameter, Ring-, Schalter- und Knotenparameter einteilen. Die Ring-, Schalter- und Knotenparameter wurden bereits in den vorangegangen Kapiteln eingehend beschrieben. Unter Systemparameter ist die Art und Größe des zu simulierenden Netzes zu verstehen. Gegenwärtig können einzelne Ringe bestehend aus bis zu 7 Datenquellen und 8 Datensenken simuliert werden sowie 13 verschiedene Typen von Banyan-Netzen, darunter 3 neue Topologien. Als Netzgrößen sind 2-256 Ein- und Ausgänge zulässig, wobei die Netzgröße je nach Zahlenbasis der Verdrahtung zwischen den Schalterstufen dem Raster von Zweier- oder Viererpotenzen folgen muß.

Zu den Simulationsparametern zählen die gewünschte Simulationsdauer

(simtime), die Zeit, nach der das Netz einen stationären Zustand erreicht hat (resetTime), eine optionale „Abklingzeit“ (coolDownTime) am Ende der Simulation sowie ein timeout-Parameter. Bei Erreichen von resetTime werden die Zählvariablen des Simulators zurückgesetzt, und die Statistik für das Netz beginnt. Die coolDownTime gibt an, wie lange nach Ablauf von simtime das Netz noch simuliert wird, ohne daß neue Datenpakete von den Quellen eingespeist werden. Sie ermöglicht, das Netz von Paketen zu entleeren. Die statistische Datenerfassung läuft dabei mit. Der timeout-Parameter überwacht die Maximalzeit, die vergehen darf, bis ein Knoten das sog. Go-Bit erhält, das für die SCI-Bandbreiteallozierungsprotokollen notwendig ist. In Tabelle 4.8.1 ist zusammenfassend die Liste der Simulationsparametern, deren Bezeichnung und Voreinstellungen angegeben.

Parameter	Name	Default	Unit
Simulations-dauer	simtime	80	ms
Abklingzeit	coolDown-Time	20	µs
Zeitgrenze	timeout	10000	ns

Tabelle 4.8.1: Simulationsparameter bei SCINET.

Die Wegewahlparameter geben bei Netzen, die redundante Wege oder multiple B-Links enthalten, Auskunft, ob adaptive Strategien bzgl. der Paketannahme, der Schalterausgangsauswahl oder der B-Link-Selektion durchgeführt werden sollen. Bei Netzen mit vertikalem Lastausgleich, die über redundante vertikale Ringe verfügen, kann zusätzlich noch die Richtung (+y/-y) ausgewählt werden, über die der Lastausgleich vorgenommen wird. In Tabelle 4.8.2 sind zusam-

adapt. B-Link-Auswahl (BLinkSelection)	adapt. Paketannahme (TestAdaptivelyForBLinkExit)
roundRobin	WatchNothing
WatchInFifosMustTake	-
WatchOutFifosMustTake	-
WatchInAndOutFifosMustTake	-
WatchInFifosNeedntTake	WatchInFifos
WatchOutFifosNeedntTake	WatchOutFifos
WatchInAndOutFifosNeedntTake	WatchInAndOutFifos

Tabelle 4.8.2: Möglichkeiten für die adaptive B-Link-Auswahl bzw. Paketannahme.

menfassend die Auswahlmöglichkeiten für die adaptive B-Link-Selektion bzw. Paketannahme angegeben, die alternativ zueinander gewählt werden müssen. Bei Netzen mit multiplen B-Links existieren die in der Tabelle angegebenen B-Link-Auswahlstrategien, während Netze mit einfachen B-Link-Schaltern und Pfadkompensation die in der Spalte für adaptive Paketannahme aufgelisteten Möglichkeiten haben. In der Tabelle 4.8.3 sind die Auswahlmöglichkeiten bei adaptiver Schalterausgangsauswahl und für Vertikalringselektion angegeben.

adapt. Schalterausgangsauswahl (GetBridgeAdaptiveRouting)	adapt. Vertikalringselektion
NoAdaptiveRemotePort	ShortestPath
ComplementedLSB	CheckAvailability
BypassFifo	-

Tabelle 4.8.3: Optionen bei adaptiver Schalterausgangsauswahl und Vertikalringselektion

Zu beachten dabei ist, daß bei Netzen mit Redundanz aber ohne Pfadkompensation nicht alle Strategien der Schalterausgangsauswahl möglich sind, adaptive Vertikalringselektion geht nur bei Netzen mit redundanten vertikalen Ringen. Zu beachten ist ferner, daß die Strategien der B-Link-Selektion bzw. Paketannahme mit denen der Schalterausgangsauswahl kombinierbar sind, so daß daraus zusätzliche Möglichkeiten entstehen.

4.9 Modellierung eines SCI-Pakets

Die von IEEE definierten Felder in den SCI-Paketformaten sind im wesentlichen auch im SCINET-Simulator vorhanden. Darüberhinaus existieren jedoch eine Reihe weiterer Buchungsgrößen, die aus implementierungstechnischen Gründen erforderlich sind. Die wichtigsten davon sind die erweiterten Adreßfelder. So wird in SCINET bei jedem Paket grundsätzlich zwischen zwei verschiedenen Adreßtypen unterschieden. Zum einen gibt es die sog. Transaktionsadressen, die systemweit eindeutig den Sender und Empfänger der zu einem Paket gehörenden Transaktion beschreiben. Zum anderen gibt es die Paketadressen, die innerhalb eines Ringes angeben, von welchem Knoten das Paket ausgesandt worden ist (Paketherkunft) und wohin es im selben Ring geschickt werden soll (Paketziel).

Die Paketzieladresse bei SCINET kann man als das Analogon einer physikalischen Ethernet-Adresse ansehen, während die Transaktionszieladresse vergleichbar mit der weltweit eindeutigen, logischen Internet-Adresse vergleich-

bar ist. Leider ist in der IEEE SCI-Norm nur eine Kategorie von Adreßfeldern im Datenformat der SCI-Pakete vorgesehen, so daß Adreßauflösungsprotokolle zur Umsetzung von logischen in physikalische Adressen ähnlich der Art von TCP/IP nicht durchgeführt werden können. Darüberhinaus existieren in den gegenwärtigen Implementierungen von SCI-Schaltern, und dementsprechend auch bei deren Simulation, keine Tabellen oder sonstige Speicher, die Informationen darüber enthalten, welcher physikalische „Gateway“ im Ring für welche logische Zieladresse zuständig ist. Bei Systemen, die aus mehr als einem Ring bestehen, können Pakete jedoch nur dann richtig geleitet werden können, werden, wenn diese Informationen vorhanden sind.

Dieses Problem wurde bei SCINET so gelöst, daß zum einen eine Differenzierung in Paket- und Transaktionsadressen vorhanden ist, und daß zum anderen jeder simulierte Schalterausgang Informationen bereitstellt, welcher SCI-Anschluß als Eingangsknoten für den Nachfolgeschalter im Ring fungiert, d.h. jeder Knoten kennt die Adressen seiner Nachbarn. Verläßt ein Paket eine Datenquelle, werden sowohl die Transaktions- als auch die Paketadressen initialisiert, verläßt es hingegen einen Schalterausgang, werden nur die Paketadressen neu gesetzt. Ein Schalterausgang bestimmt für jedes Paket, das den Ausgang verlassen will, den Nachfolgeknoten im Ring, indem anhand der Knotenliste des Rings, auf die der Knoten zugreifen kann, reihum Knoten für Knoten inspiziert wird, ob der betrachtete Knoten gemäß des gewählten Routing-Schemas, das adaptiv oder deterministisch sein kann, zur Transaktionszieladresse hinführt. Die SCI-Adresse des ersten Knotens, der diese Bedingung erfüllt, dient dann als Paketzieladresse.

Modellierung der Paketprioritäten

Bei SCI-Ringen werden Daten in unterschiedlichen Pakettypen, wie Request-, Response-, Retry- und Echo übertragen. Darüberhinaus muß jeder Knoten über seinen SCI-Link-Ausgang die Pakete seines Bypass-Fifos sowie die empfangenen Idle-Symbole weiterreichen. Der IEEE-Standard für SCI stellt es absichtlich frei, mit welcher Priorität die verschiedenen Typen vom Knoten abzusenden sind. Bei SCINET wurde folgende Priorität festgelegt: 1. idle, 2. forward, 3. Retry-Response, 4. Retry-Request, 5. Response und 6. Request. D.h., daß eintreffende Idle-Symbole sofort weitergereicht werden, während Request-Pakete, die im Ausgabepuffer des Knotens gespeichert sind, am längsten warten müssen bis sie die gemeinsame Ressource des SCI-Link-Ausgangs zugeteilt bekommen. Alternativ wäre es auch möglich, den Ausgabekanal nicht anhand von Prioritäten, sondern reihum zuzuteilen (Round Robin Scheduling).

Modellierung des Retry-Verkehrs

Erhält ein Sender eine negatives Echo für ein Request-Paket, muß er eine Paketwiederholung durchführen. Leider ist im IEEE-Standard für SCI nichts darüber ausgesagt, nach welcher Zeit nach Eintreffen des negativen Echos die

Paketwiederholung durchgeführt werden soll. Bei SCINET wurde dieser offene Punkt als von außen einstellbarer Parameter implementiert, um einerseits größtmögliche Flexibilität zu erreichen und andererseits die damit verbundenen Fragestellungen bzgl. des Durchsatzes beim Empfänger zum Gegenstand simulativer Untersuchungen machen zu können. Insgesamt können bei SCINET drei verschiedenen Strategien für die Zeit bis zum Aussenden eines Retry-Pakets gewählt werden, bei denen ausgehend von einem Anfangswert entweder eine konstante- oder eine linear- bzw. exponentiell ansteigende Wartezeit vergeht.

5 SCINET-Testsystem

5.1 Einleitung

Der vierte Schritt zur Leistungsbewertung von SCI ist die Validierung von Modell und Implementierung. Hier soll die Validierung u.a. anhand eines Testaufbaus exemplarisch durchgeführt werden. Der Teststand erlaubt, konkrete Messungen hinsichtlich Latenz und Durchsatz durchzuführen, um dadurch Modell und Simulation zu unterstützen oder zu verwerfen.

5.2 Teststandaufbau

Zur Validierung sind zwei unterschiedliche Teststände zur SCI-basierten Datenübertragung aufgebaut worden, zum einen, um die erzielten Meßergebnisse vergleichen zu können, zum anderen, um eine höhere Aussagekraft der Resultate zu erzielen. Jeder Teststand besteht aus zwei PCs, die über SCI-Schnittstellenkarten und einen SCI-Ring verbunden sind. Als Schnittstellenkarten wurden kommerzielle Produkte der Fa. Dolphin verwendet [Dolphin96a], der Ring besteht aus speziellen SCI-Kupferkabeln.

Beim ersten Teststand wurden PentiumPro-PCs mit 200 MHz Taktrate eingesetzt, die auf dem Intel 440FX PCI-Bus-Chipsatz und einer GA-686DX Grundplatine basieren, zusammen mit Linux 2.0.30 als Betriebssystem. Die kommerziell erworbenen Schnittstellenkarten wurden mit selbstgeschriebenen Gerätetreibern und Meßprogrammen betrieben, um Bandbreite und Latenz der SCI-Datenübertragungen zu bestimmen.

Der zweite Teststand besteht aus zwei 100 MHz Pentium PCs mit dem üblichen FX-PCI-Bus-Chipsatz und Windows NT als Betriebssystem. Hier wurden Gerätetreiber und Testprogramme von Dolphin übernommen. Für den ersten

Teststand wurden NWRITE64 und NREAD64-Transaktionen für die Messungen verwendet, bei beide 64 Byte pro Paket verschicken, während beim zweiten Teststand DMOVE64-Transaktionen herangezogen wurden, die zwar die gleiche Paketlänge aufweisen jedoch keine Response benötigen.

Eine Schnittstellenkarte beruht intern auf einem sog. LC1-Link-Controller-Baustein [Dolphin95] mit 200 MB/s Datenrate auf dem Link, der die physikalischen Schichten der SCI-Transportprotokolle realisiert, sowie einer PCI-Busbrücke, die das PCI-Busprotokoll [Kau93] des PCs umsetzt in die B-Link-Spezifikation [Dolphin94a]. Der Aufbau des Link-Controllers entspricht der durch IEEE genormten Vorgabe. Anstelle des „angeschlossenen Benutzerendgeräts“ (attached user device) ist die PCI-Busbrücke mit dem PC getreten. Das BlockDiagramm der Schnittstellenkarte ist in Bild 5.2.1 gezeigt. Die Brücke

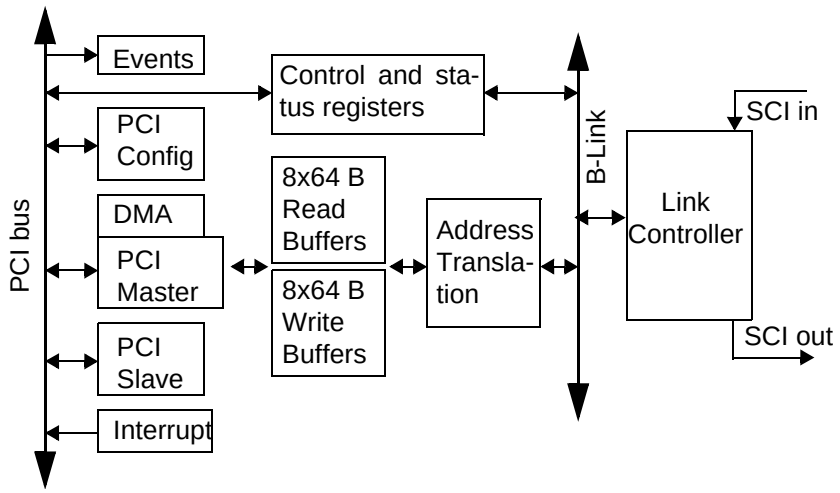

Bild 5.2.1: Blockdiagramm der Schnittstellenkarte im PC.

besteht aus den PCI-Bus-Master/Slave-Funktionsblöcken, einer DMA-Einheit und Pufferspeichern der Tiefe 8 für Lese- und Schreiboperationen. Weitere wichtige Komponenten sind ein Adreßumsetzungs-Cache, der die Abbildung von PCI-Busadressen in SCI-Ringadressen vornimmt, Konfigurations-, Steuer- und Statusregister sowie eine Interrupt-Steuerung.

Die Brücke ist in der Lage, alle 8 Speicherplätze ihrer Schreib- bzw. Lesepufferspeicher bis zu 8 verschiedenen, externen PC-Bus-Mastern zuzuordnen, die in Form von Prozessen oder Threads auf dem PC existieren, um dadurch 8 voneinander unabhängige Transfers simultan zu unterstützen. Alternativ können auch alle 8 Speicherplätze zusammengeschaltet werden, um dadurch für einen einzigen Master den achtfachen Durchsatz zu erzielen. Die Voraussetzung für die Kopplung der Pufferspeicher ist, daß auf dem PCI-Bus genügend schnell ausreichend viele Daten mit aufeinanderfolgenden Adressen angelegt werden, bei 8-fach Kopplung also 512 Byte in 128 Buszyklen, und daß die Steuer- und Statusregister entsprechend konfiguriert worden sind. Die Pufferplatzkopplung wird auch als Stream-Combining bezeichnet.

5.3 Meßergebnisse

Auf dem PentiumPro-basierten Teststand wurden die in Bild 5.3.1 dargestellten Durchsätze bei Speicher-zu-Speicher-Transfer erzielt. Da es bei SCI nicht er-

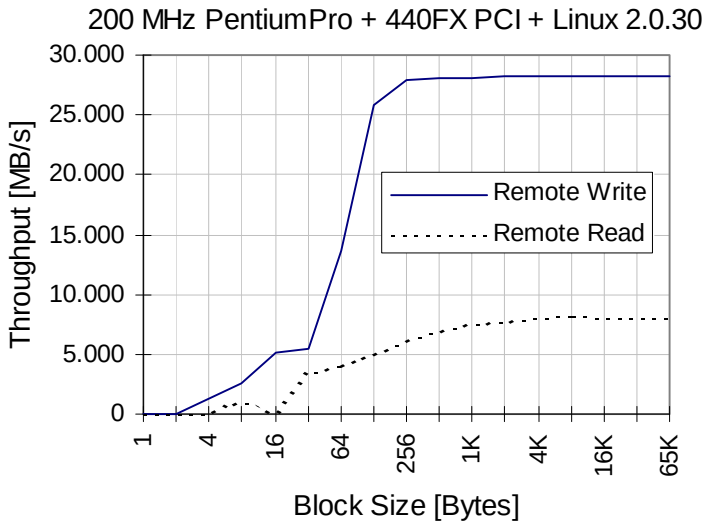

Bild 5.3.1: Durchsätze für den Speicher-zu-Speicher-Transfer als Funktion der Blöckgröße.

forderlich ist, Daten zwischen Systempuffern auszutauschen, die anschließend in den Benutzeradreßraum kopiert werden, stellen die Ergebnisse zugleich die für den Benutzer erzielbaren Netto-Datenraten dar. Wie man sieht, erreicht der Durchssatz bereits für kurze Blocklängen von 256 Byte sein Maximum von ca. 28 MB/s bei einer Remote Write-Transaktion und bleibt stabil auf diesem Niveau. Bei nur 64 Byte Blocklänge wird die halbe maximale Geschwindigkeit erreicht.

Das entfernte Lesen (Remote Read) ist um den Faktor 3.5 langsamer als das entfernte Schreiben, was darauf zurückzuführen ist, daß für den Schreibanforderer (Write Requester) die Transaktion bereits dann abgeschlossen ist, wenn seine Daten in den Pufferspeicher der Schnittstellenkarte eingespeichert sind, während bei einer Lese-Transaktion gewartet werden muß, bis die Daten tatsächlich gelesen wurden. Sobald die Geschwindigkeit eine Rolle spielt, sollte deshalb bei Datenerfassungsanwendungen, auf entferntes Lesen verzichtet werden, vielmehr sollten die Sensoren auf ein Trigger-Signal hin autonom ihre Daten in den Zielrechnern ablegen (Push-Strategie). Aus diesem Grunde wird im weiteren nur von den Bandbreitemeßergebnissen für entferntes Schreiben berichtet.

In Bild 5.3.2 ist der Zeitverbrauch, der zum Transfer eines Blockes notwendig ist, zur Ergänzung der Bandbreitemessungen als Funktion der Blocklänge dargestellt. Die Latenzzeiten sind bei entferntem Lesen aufgrund der geringen

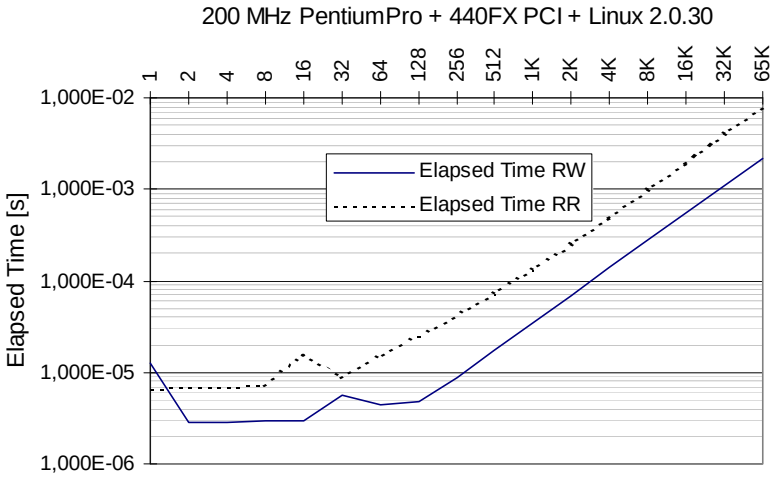

Bild 5.3.2: Gemessene Latenzzeiten in Abhängigkeit von der Blocklänge.

Bandbreite entsprechend größer. Wie man in Bild 5.3.2 sieht, ist bei SCI der Verwaltungsaufwand, der notwendig ist, um einen Pakettransfer anzustoßen (Setup Time), rel. gering: ca. 10 µs sowohl für entferntes Lesen wie für Schreiben sind ausreichend. Ein auf SCI-basierende Steuerung oder Regelung würde deshalb eine sehr kurze Reaktionszeit bieten.

Der Durchsatz des Pentium-Teststands ist zum Vergleich in Bild 5.3.3 gezeigt. Trotz der hier verwendeten DMOVE64-Transaktion, die keine Response erfordert, werden nur 12 MB/s, also weniger als die Hälfte, erreicht.

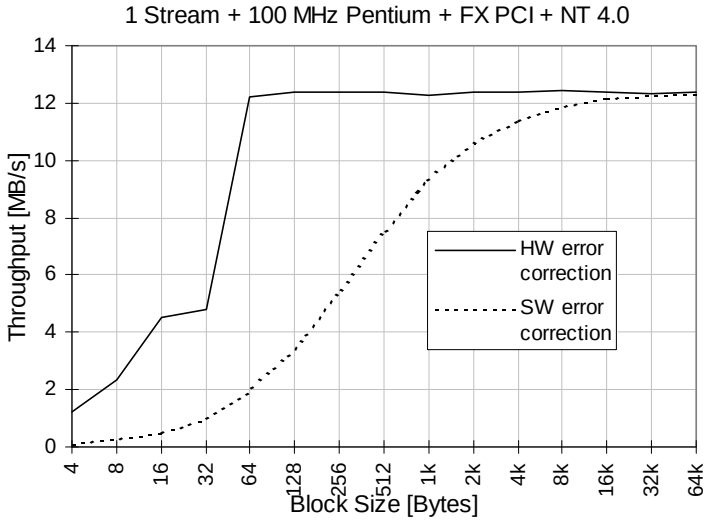

Bild 5.3.3: Durchsatz des Pentium-Teststands bei DMOVE64 und einem Stream.

In Bild 5.3.3 ist gestrichelt dargestellt, wie die Datenrate absinkt, wenn man zusätzlich zur Hardware-Fehlererkennung und Korrektur, die bereits die Schnittstellenkarte bzw. der Link-Controller auf Paketbasis bietet, eine Software-

Fehlererkennung und Korrektur auf Blockbasis vom PC durchführen läßt. In beiden Fällen ist die erzielbare Endgeschwindigkeit dieselbe, jedoch wird sie bei der Software-Lösung erst bei einer wesentlich größeren Paketlänge von >64 KB erreicht. Ohne Software-Blockprüfung und Korrektur wird im Gegensatz zum PentiumPro-Teststand die maximale Transferrate bereits bei 64 Byte erzielt. Die Setup-Zeit ist mit 2-3 µs ebenfalls geringer, was auf einen effizienteren Gerätetreiber schließen läßt (Bild 5.3.4). Bei zusätzlicher Software-

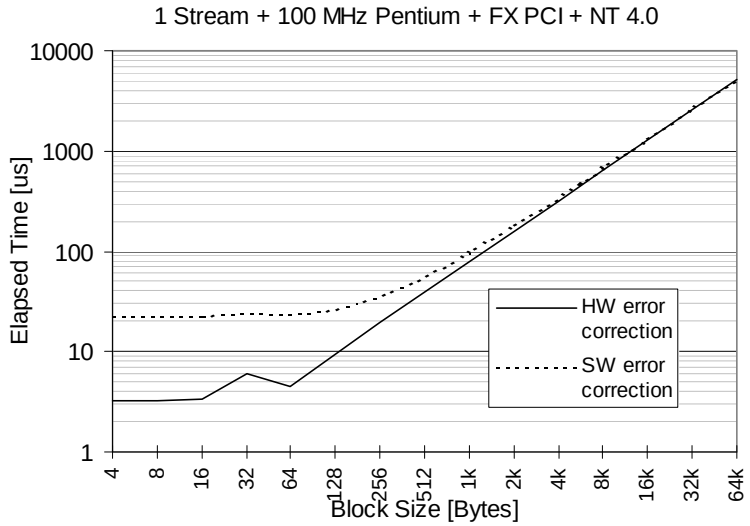

Bild 5.3.4: Gemessene Latenzen bei einem Stream auf dem Pentium-Testsystem.

Blockprüfung steigt sie auf 11-12 µs an. Für große Blocklängen konvergieren die Latenzen von beiden Varianten gegen denselben Wert und befinden sich damit in Übereinstimmung mit den gemessenen Bandbreiten.

Der beim Pentium-Teststand verwendete Gerätetreiber erlaubt wahlweise, ein, zwei, vier oder 8 Pufferspeicherplätze zu koppeln, um so den Durchsatz bei entsprechend größerer Blocklänge zu steigern. Beim Linux-Treiber sind dagegen einem Stream vier Speicherplätze fest zugeordnet. Wie man anhand von Bild 5.3.5 und Bild 5.3.6 sieht, ist der Anstieg der maximalen Bandbreite nahezu linear zur Zahl der gekoppelten Speicherplätze, wobei sich die erforderliche Blocklänge ebenfalls verdoppelt. Die größte erzielbare Datenrate liegt bei 45 MB/s. Bei 8 gekoppelten Puffern zeigt sich ein Einbruch im Durchsatz, der verhindert, daß eine höhere Datenrate als 45 MB/s erreicht wird. Bei der für diese Pufferzahl erforderlichen Blockgröße von 64*8 = 512 Byte fällt der Durchsatz auf 34 MB/s zurück. Dafür verantwortlich ist vermutlich die rel. langsame CPU des PCs, unter der Annahme, daß sie nicht schnell genug Daten auf dem PCI-Bus zur Verfügung stellen kann (Bild 5.3.7). Zur Ergänzung der Bandbreitemessungen sind in Bild 5.3.8-Bild 5.3.10 die Zeitverbräuche (Latenzen) für 2,4 und 8 Streams gezeigt. Wie erwartet, ist die Latenz für den Fall von 4 Streams für alle Blocklängen am kleinsten. Die Setup-Zeiten hingegen sind unabhängig von der Zahl der verwendeten Streams.

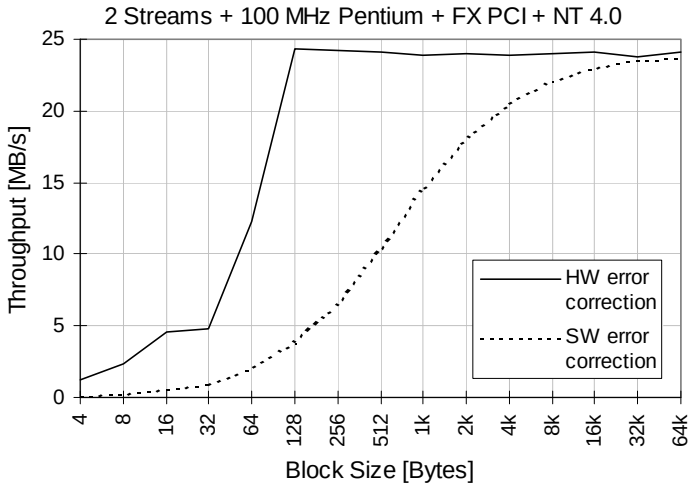

Bild 5.3.5: Durchsatz bei 2 Streams und der DMOVE64-Transaktion.

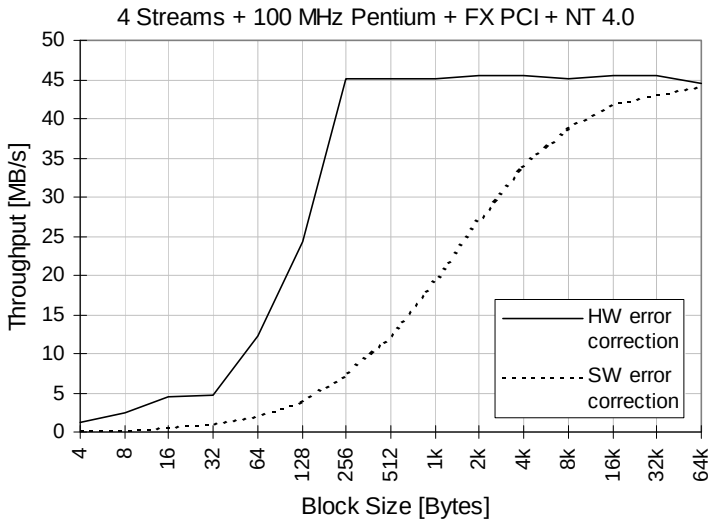

Bild 5.3.6: Durchsatz bei 4 Streams und der DMOVE64-Transaktion.

5.4 Bewertung der Meßergebnisse

Die durchgeführten Messungen konnten verschiedene Antworten auf die gestellten Fragen liefern. Beispielsweise wurde die Frage, ob in einem Datenerfassungsystem die aufgenommenen Sensorwerte von den Rechnern aus den Sensormeßaufnahmespeichern gelesen werden sollen (Pull-Strategie) oder ob besser die Sensoren die Werte aktiv an die Rechner abliefern (Push-Strategie), eindeutig zugunsten der Push-Strategie beantwortet. Nur beim Schreiben auf einen entfernten Speicher - und nicht beim Lesen - lassen sich über SCI-Schnitt-

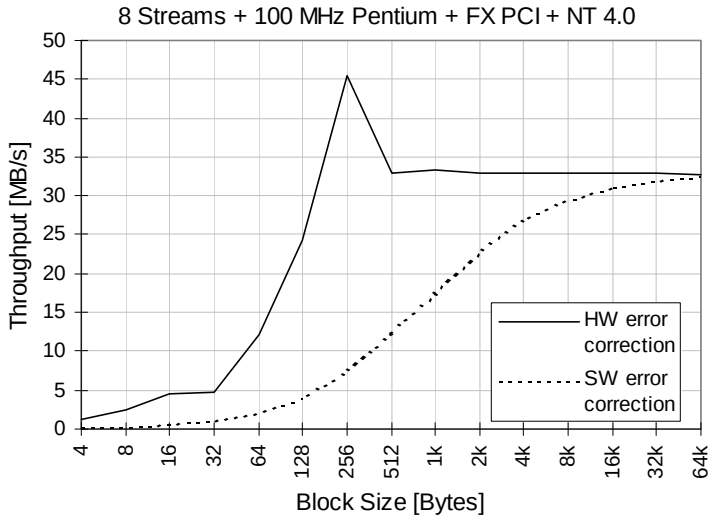

Bild 5.3.7: Durchsatz bei 8 Streams.

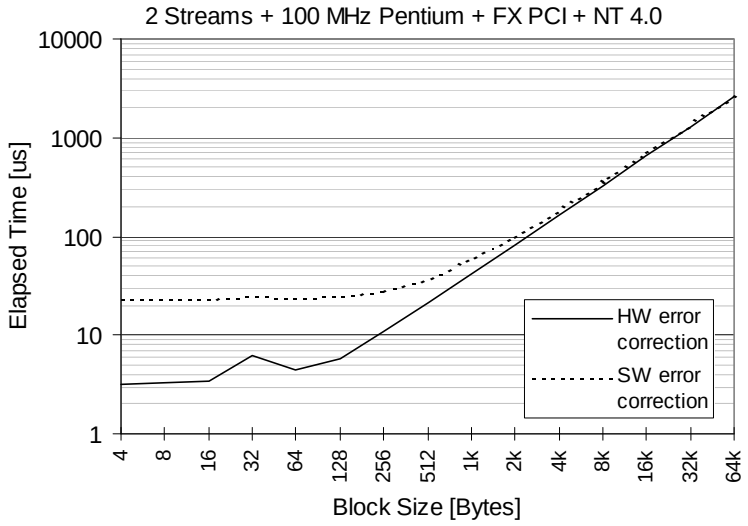

Bild 5.3.8: Latenzen bei 2 Streams und DMOVE64.

stellen hohe Durchsätze erzielen, so daß die Pull-Strategie aus diesem Grunde ausscheidet. Der Grund für die Geschwindigkeitsunterschiede liegt darin, daß Schreiboperation für den Schreibenden bereits dann beendet sind, sobald er seine Werte in den Sendepuffer der Schnittstellenkarte eingespeichert hat. Unmittelbar danach kann der Recher bereits ein neues Datum schreiben. Bei entferntem Lesen hingegen muß der lesende Recher warten, bis das gewünschte Datum bei seiner Schnittstellenkarte eingetroffen ist, bevor er einen Nachfolgewert lesen kann. Um sicherzustellen, daß alle Sensoren zur selben Zeit ihre jeweiligen Meßgrößen abtasten und speichern, muß ein zentraler Trigger die Abtastzeitpunkte bestimmen. Nach dem Empfang eines Triggersignals sollten die Sensoren selbstständig ihre Meßwerte in die Arbeitsspeicher der Meßauf-

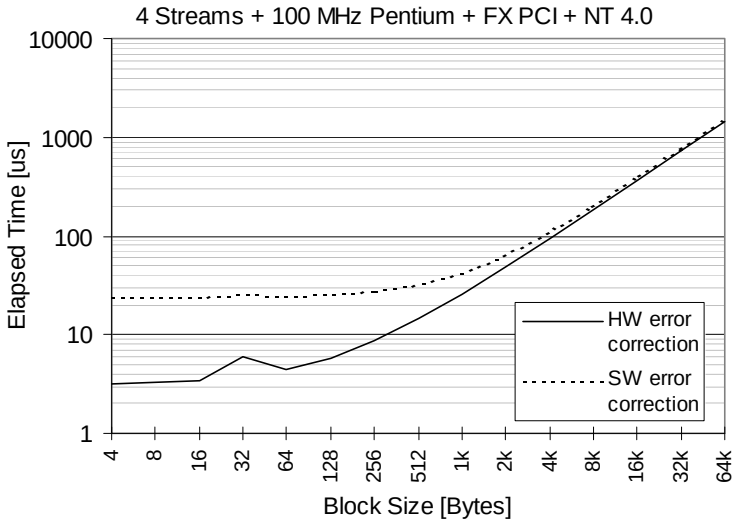

Bild 5.3.9: Latenzen bei 4 Streams und DMOVE64.

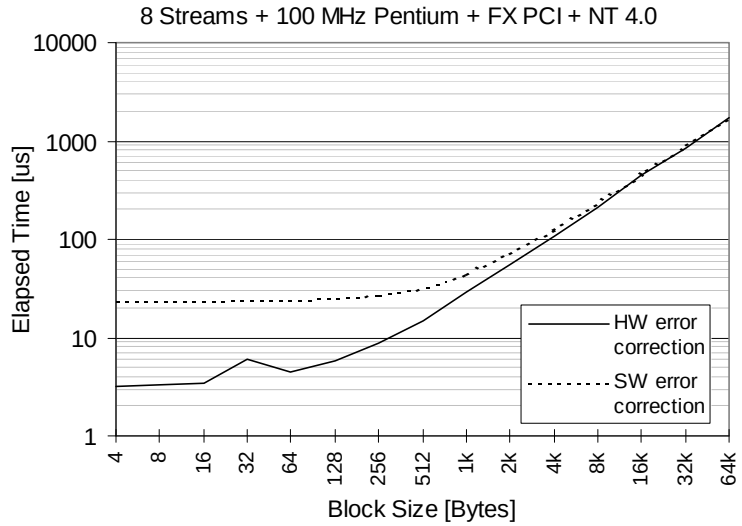

Bild 5.3.10: Latenzen bei 8 Streams und DMOVE64.

nahmerechner schreiben, von wo sie weiterverarbeitet werden können.

Die am Teststand durchgeführten Messungen haben weiterhin gezeigt, daß multiple Streams den Durchsatz proportional erhöhen. Das Optimum liegt bei vier gekoppelten Pufferspeichern. Dabei werden für Blocklängen ab 256 Byte ca. 45 MB/s Nettodatenrate erzielt. Die gemessenen Setup-Zeiten für die Initiierung eines Transfers sind mit 2 bis 10 µs sehr niedrig.

Insgesamt haben die gewonnenen Resultate die prinzipielle Eignung von SCI für Datenerfassungssysteme der Hochenergie- und Plasmaphysik hinsichtlich der Metriken Bandbreite und Latenz gezeigt. Die Frage der Skalierbarkeit hingegen kann nicht mit Hilfe der Teststände untersucht werden, vielmehr bleibt dies der Simulation vorbehalten.

6 SCINET-Simulator

6.1 Einleitung

Der fünfte Schritt bei der Leistungsanalyse von SCI-Netzen ist die Durchführung der Simulation. Um zu wissen, welche Ausgabegrößen beim SCINET-Simulator welche Bedeutung haben und um den prinzipiellen Aufbau des Simulators zu verstehen, wird im weiteren ein kurzer Überblick über dessen Aufbau und die von ihm berechneten Größen gegeben.

6.2 Kurzbeschreibung des Simulators

Der SCINET Simulator dient dazu, das funktionale und zeitliche Verhalten eines einzelnen SCI-Rings oder Schalters sowie von ganzen SCI-Systemen bestehend aus einer Vielzahl von über Schalter verbundenden Ringen präzise vorauszusagen. Zur Durchführung eines Simulationslaufs sind eine größere Zahl von Timing- und anderen Eingabeparametern notwendig, die in Kapitel 4.5 "Modellierung eines SCI-Knotens" bis Kapitel 4.8 "Modellierung eines SCI-Netzes" erläutert worden sind, sowie die Größe und Art der zu simulierenden SCI-Struktur. Als Ergebnis erhält man detailliert Auskunft über das Verhalten aller Komponenten des Systems. In der Regel werden die Ergebnisse nach erfolgter Simulation der besseren Übersichtlichkeit wegen in graphischer Form als x-y-Diagramme dargestellt. Eine graphische Benutzeroberfläche dient der erleichterten Definition der Eingabeparameter und Netzstruktur sowie dem Filtern interessierender Ausgabewerte. Die auf Pull-Down-Menüs basierende Benutzeroberfläche ist integraler Bestandteil des SCINET Simulators, während zur graphischen Aufbereitung der Daten kommerzielle Tabellenkalkulationsprogramme wie z.B. Excel oder auch Plot-Programme wie GNUplot genutzt werden können.

Der schematische Ablauf einer Simulation und seine programmiertechnische Implementierung in Form von Programmodulen ist in Bild 6.2.1 dargestellt. Nach der interaktiven Definition des Netzes und seiner funktionalen und temporalen Parameter wird vom Modul TOPOENGINE eine Netzliste erstellt, die in einer eigens für diesen Zweck definierten Macrosprache verfaßt ist. Der Vorteil dieser „hochsprachlichen“ Netzliste ist ihre leichte Lesbarkeit, die dem Benutzer eine nachträgliche, manuelle Editiermöglichkeit der Liste erlaubt, um so spezielle Simulationswünsche zu realisieren. Die mnemonische Liste wird anschließend von einem Parser in eine Form umgewandelt, die vom Simulator direkt verarbeitet werden kann. Die eigentliche Simulation benötigt je nach Netzgröße

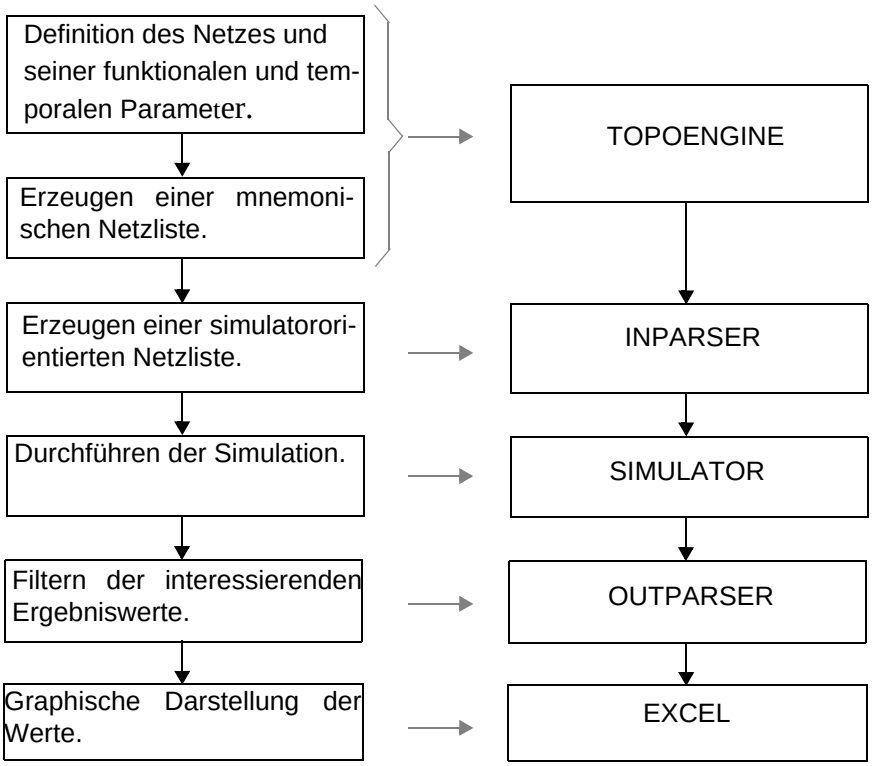

Bild 6.2.1: Schematischer Ablauf einer Simulation und seine Programmimplementierung.

und Datenrate der Eingabequellen bis zu einer Stunde Rechenzeit auf einem Pentium II-PC von 266 MHz Taktrate und 128 MB Hauptspeicher.

Der Simulator ist in der objektorientierten Simulationssprache MODSIM II geschrieben und umfaßt ca. 400 Seiten Quellausdrucke. Während des Simulationslaufs werden eine Vielzahl statistischer Größen, die in den einzelnen SCI-Komponenten anfallen, in eine Ausgabedatei protokolliert, so daß zur Evaluierung der Ergebnisse eine Vorauswahl interessierender Größen vorgenommen werden muß. Dies erfolgt in einem Ausgabe-Parser, der ebenfalls über Pull-Down Menüs vom Benutzer bedient werden kann. Dessen Ausgabedatei stellt einen Auszug der protokollierten Werte dar, und zwar in einer Form, wie sie unmittelbar vom Tabellenkalkulationsprogramm Excel eingelesen werden kann.

Bei der Netzdefinition in TOPOENGINE sind verschiedene Banyan-Netze implementiert, die entweder reine Einpfadnetze darstellen oder über zusätzliche Stufen oder Ausgänge verfügen, um dadurch Redundanz zu erzielen. Zur besseren Handhabbarkeit sind diese Netze nach dem untenstehenden Klassifizierungsschema benannt. Die Netze, für die eine Netzliste mit Hilfe von TOPOENGINE erzeugt werden kann, sind:

- *bO2n:* Banyan in *O*mega-Topologie mit *b*idirektionalen Ringen zwischen den Netzstufen, die zur Permutationsbasis *2* verdrahtet sind, mit *n*icht-adaptiver Wegewahl.
- *bF2n:* Banyan in *F*lip-Topologie mit *b*idirektionalen Ringen zwischen den

Netzstufen, die zur Permutationsbasis *2* verdrahtet sind, mit *n*icht-adaptiver Wegewahl.

- *bG2n:* Banyan in *G*eneralized Cube-Topologie mit *b*idirektionalen Ringen zwischen den Netzstufen, die zur Permutationsbasis *2* verdrahtet sind, mit *n*icht-adaptiver Wegewahl.
- *bI2n:* Banyan in *I*ndirect Binary n-Cube-Topologie mit *b*idirektionalen Ringen zwischen den Netzstufen, die zur Permutationsbasis *2* verdrahtet sind, mit *n*icht-adaptiver Wegewahl.
- *uO2n:* Banyan in *O*mega-Topologie mit *u*nidirektionalen Ringen zwischen den Netzstufen, die zur Permutationsbasis *2* verdrahtet sind, mit *n*icht-adaptiver Wegewahl.
- *uF2n:* Banyan in *F*lip-Topologie mit *u*nidirektionalen Ringen zwischen den Netzstufen, die zur Permutationsbasis *2* verdrahtet sind, mit *n*icht-adaptiver Wegewahl.
- *uG2n:* Banyan in *G*eneralized Cube-Topologie mit *u*nidirektionalen Ringen zwischen den Netzstufen, die zur Permutationsbasis *2* verdrahtet sind, mit *n*icht-adaptiver Wegewahl.
- *uI2n:* Banyan in *I*ndirect Binary n-Cube-Topologie mit *u*nidirektionalen Ringen zwischen den Netzstufen, die zur Permutationsbasis *2* verdrahtet sind, mit *n*icht-adaptiver Wegewahl.
- *uO4n:* Banyan in *O*mega-Topologie mit *u*nidirektionalen Ringen zwischen den Netzstufen, die zur Permutationsbasis *4* verdrahtet sind, mit *n*icht-adaptiver Wegewahl.
- *uG4n:* Banyan in *G*eneralized Cube-Topologie mit *u*nidirektionalen Ringen zwischen den Netzstufen, die zur Permutationsbasis *4* verdrahtet sind, mit *n*icht-adaptiver Wegewahl.
- *uO2va:* Erweiterter Banyan in *O*mega-Topologie mit *u*nidirektionalen Ringen zwischen den Netzstufen, die zur Permutationsbasis *2* verdrahtet sind, mit adaptiver Wegewahl durch *v*ertikalen Lastausgleich in *a*llen Stufen.
- *uO2vl:* Erweiterter Banyan in *O*mega-Topologie mit *u*nidirektionalen Ringen zwischen den Netzstufen, die zur Permutationsbasis *2* verdrahtet sind, mit adaptiver Wegewahl durch *v*ertikalen Lastausgleich in der *l*etzten Stufe.
- *uO2p:* Erweiterter Banyan in *O*mega-Topologie mit *u*nidirektionalen Ringen zwischen den Netzstufen, die zur Permutationsbasis *2* verdrahtet sind, mit adaptiver Wegewahl durch *p*arallele Leitungen.
- *uO4e:* Erweiterter Banyan in *O*mega-Topologie mit *u*nidirektionalen Ringen zwischen den Netzstufen, die zur Permutationsbasis *4* verdrahtet sind, mit adaptiver Wegewahl durch vertikalen Lastausgleich in einer *e*xtra Stufe.

Die genaue Bedeutung der Netze, ihr Aufbau und ihre Vorteile werden ab Kapitel 9.2.1 "Omega-Netz" beschrieben. Hier soll nur auf die Möglichkeit deren Erzeugung hingewiesen werden.

Zum Schluß dieses Kapitels sollen noch einige Beispielabbildungen der gra-

phischen Benutzeroberfäche des Simulators gezeigt werden. In Bild 6.2.2 ist das Fenster des Hauptmenüs dargestellt, vom dem aus die beschriebenen Netze in verschiedener Größe und mit verschiedener Zahl interner B-Links erzeugt werden können. Im darauffolgenden Bild 6.2.3 ist die Visualisierung eines Generalized Cube-Netzes der Größe 16x16. gezeigt.

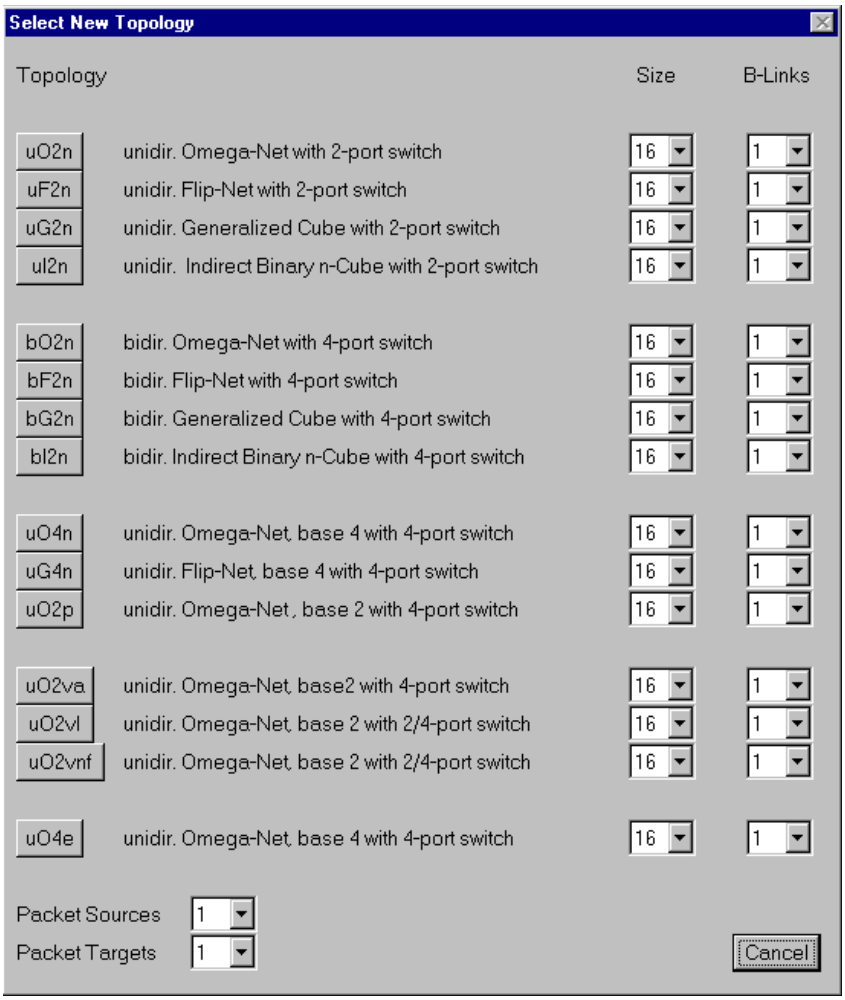

Bild 6.2.2: Hauptmenü des Simulators.

6.2.1 Ausgabewerte der Simulation

Für den Benutzer besonders wichtig ist die Frage, welche Größen er nach der Simulation zur Auswertung erhalten kann, d.h., welche Werte in der SCINET-Ausgabedatei protokolliert sind. Die Ausgabedatei erreicht bei einer maximalen Netzgröße von 256 Ein-/ und Ausgängen eine Größe von ca. 43 MB und darin ist über eine Vielzahl von Größen wie beispielsweise die Anzahl der Pakete in allen Knoten, die Paketlängen und Bandbreiten sowie über die Latenzzeiten Buch geführt. Zusätzlich sind bereits statistische Vorauswertungen

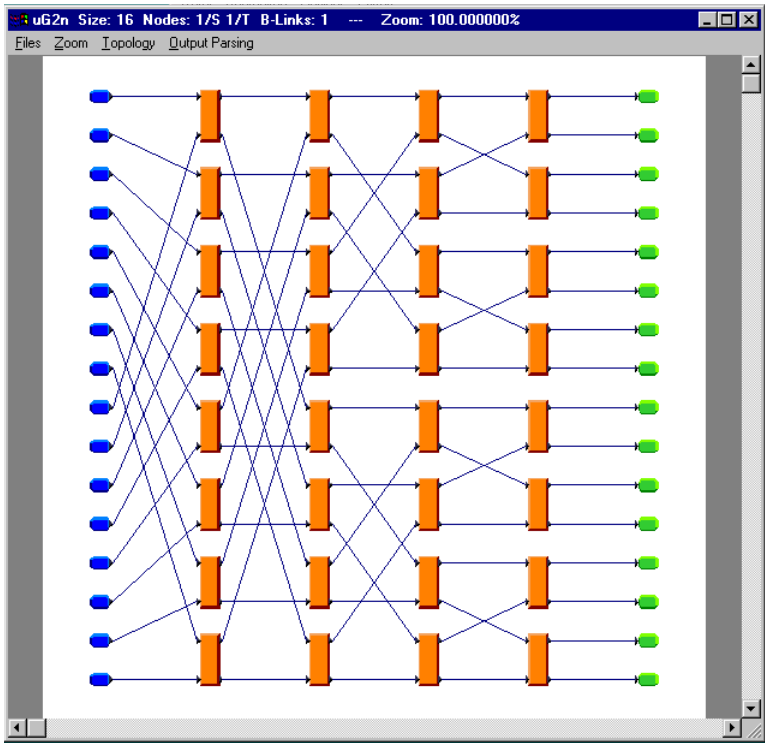

Bild 6.2.3: Visualisierung eines Generalized Cube-Netzes der Größe 16x16.

einiger Größen in Form von summierten Einzelwerten, Maxima und Minima, Mittelwerten, Varianzen und Standardabweichungen enthalten.

Die Ausgabedatei ist in einzelne Tabellen gegliedert, die auch bei unterschiedlichen Zeileninhalten denselben Aufbau in Form von vier Spalten haben. Die Spalten entsprechen den Kategorien von Daten, die jeweils bei den SCI-Ein/- und Ausgabekanälen sowie den B-Link-Ein/- und Ausgängen anfallen. Diese Kategorien werden in der Ausgabedatei des besseren Umgangs wegen mit *SSent*, *SRec*, *BSent* und *BRec* abgekürzt, was eine mnemonische Kurzform für Senden und Empfangen auf einem SCI- bzw. B-Link ist. Am Ende der Ausgabedatei sind die Einzelwerte aller SCI-Knoten eines bestimmten Typus nach Zeilen und Spalten getrennt aufsummiert und als *GSent*, *GRec*, *GBSent* und *GBRec* bezeichnet, was an „Gruppenwerte“ erinnern soll. Das bedeutet, daß die Knoten eines Typs zu einer Gruppe zusammengefaßt und korrespondierende Ausgabewerte der einzelnen Zeilen addiert werden. Beispielsweise entspricht die aufsummierte Zahl der Pakete, die von jeder Datenquelle auf ihrem SCI-Link ausgegeben wird, der Gesamtzahl aller in das Netz eingespeister Pakete. (N.B.: Der Typ eines Knotens spezifiziert, um welche Paketquelle, um welches Paketziel oder um welchen Schalterknoten es sich handelt.)

Die Zeilen der Ausgabedatei des Simulators sind in drei große Abschnitte unterteilt, die Informationen zu jedem einzelnen SCI-Knoten, zu den aufsummierten Gruppenwerten und zu den Latenzzeiten, die die Pakete im Netz erfahren haben, enthalten. Die Abschnitte heißen entsprechend *Node Information*, *Group Information* und *Latency Information*. Innerhalb des Abschnitts für die

Knoteninformationen gibt es für jede der vier Spaltenkategorien und für jeden SCI-Knoten den gleichen Satz von Ausgabewerten. Diese Größen werden entweder von oder zu einem Knoten auf einem SCI- oder B-Link gesendet und lauten im einzelnen:

- *NorReqPac* (= No Retry Request Packets in [Packets]): Zahl der Request-Pakete ohne Paketwiederholungen.
- *NorResPac* (= No Retry Response Packets in [Packets]): Zahl der Response-Pakete ohne Paketwiederholungen.
- *RetReqPac* (= Retry Request Packets in [Packets]): Zahl der Paketwiederholungen für Requests.
- *RetResPac* (= Retry Response Packets in [Packets]): Zahl der Paketwiederholungen für Responses.
- *PosEchReqPac* (= Positive Echos for Request Packets in [Packets]): Zahl der positiven Echos für Requests.
- *NegEchReqPac* (= Negative Echos for Request Packets in [Packets]): Zahl der negativen Echos für Requests.
- *PosEchResPac* (= Positive Echos for Response Packets in [Packets]): Zahl der positiven Echos für Responses.
- *NegEchResPac* (= Negative Echos for Response Packets in [Packets]): Zahl der negativen Echos für Responses.
- *TotEchRaw* (= Total Echo Raw in [Bytes]): Gesamtzahl der Bruttopaketlängen aller Echos inclusive der Paketköpfe und Prüfsummen sowie der Idle-Bytes, die die Pakete einrahmen. Bei SCINET werden vor und nach jedem Paket je zwei Idle-Bytes gesendet.
- *TotReqRaw* (= Total Requests Raw in [Bytes]): Gesamtzahl der Bruttopaketlängen aller Request-Pakete.
- *TotResRaw* (= Total Responses Raw in [Bytes]): Gesamtzahl der Bruttopaketlängen aller Response-Pakete.
- *TotRetRaw* (= Total Retries Raw in [Bytes]): Gesamtzahl der Bruttopaketlängen aller Retry-Pakete.
- *TotRin/BLiRaw* (= Total Ring or B-Link Raw in [Bytes]): Gesamtzahl der Bruttopaketlängen von allen Pakettypen. Dazu zählen Request-, Response-, Echo- und Retry-Pakete, incl. deren Paketköpfen, Prüfsummen und gegebenenfalls Idle-Bytes. Zu beachten ist, daß Pakete auf SCI- bzw. B-Links unterschiedliche Längen haben und daß auf den B-Links keine Echos oder Idle-Bytes übertragen werden.
- *NorReqPay* (= No-Retry Request Payload in [Bytes]): Gesamtzahl der Nettolängen aller Request-Pakete ohne Retries, d.h. die reine Zahl von Nutzdaten ohne Formatierung oder Paketwiederholungen. Paketwiederholungen dürfen nicht zu den Nutzdaten gezählt werden, da jedes erzeugte Paket nur einmal beim Empfänger Daten abliefert.

- *NorResPay* (= No-Retry Response Payload in [Bytes]): Gesamtzahl der Nettolängen aller Response-Pakete ohne Paketwiederholungen.
- *TotEchRawBdW* (= Total Echos Raw Bandwidth in [MB/s]): Bandbreite der Gesamtzahl der Bruttopaketlängen aller Echos.
- *TotReqRawBdW* (= Total Requests Raw Bandwidth in [MB/s]): Bandbreite der Gesamtzahl der Bruttolängen aller Request-Pakete.
- *TotResRawBdW* (= Total Responses Raw Bandwidth in [MB/s]): Bandbreite der Gesamtzahl der Bruttolängen aller Response-Pakete.
- *TotRetRawBdW* (= Total Retries Raw Bandwidth in [MB/s]): Bandbreite der Gesamtzahl der Bruttolängen aller Retry-Pakete.
- *TotRin/BLiBdW* (= Total Ring or B-Link Raw Bandwidth in [MB/s]): Bandbreite der Gesamtzahl der Bruttolängen von allen Pakettypen.
- *NorReqPayBdW* (= No-Retry Request Payload Bandwidth): Bandbreite der reinen Nutzdaten in den Request-Paketen ohne Paketwiederholungen.
- *NorResPayBdW* (= No-Retry Response Payload Bandwidth): Bandbreite der reinen Nutzdaten in den Response-Paketen ohne Paketwiederholungen.

Innerhalb des Abschnitts für die Gruppeninformationen (*Group Informations*) gibt es für jede der vier Spaltenkategorien und für jeden Knotentyp den gleichen Satz von Ausgabewerten, die die Summen der zuvor beschriebenen Knotenausgabewerte enthalten. Ihre Bezeichnung ist bis auf ein vorangestelltes „Grp“ mit deren Bezeichnung identisch. So kennzeichnet beispielsweise *GrpNorReqPac* die Summe aller Request-Pakete ohne Paketwiederholungen eines bestimmten Kontentyps. (Zur Unterscheidung der verschiedenen Knoten und ihrer Typen werden von TOPOENGINE automatisch sog. nodeid und typeid-Nummern vergeben.) Zusätzlich zu den geschilderten protokollierten Daten in den Knoten- und Gruppeninformationen werden weitere Informationen berechnet, die spezifisch bei Datenquellen sowie bei Schalterknoten anfallen.

Bei den Datenquellen wird Buch über die Zahl der bei der Einspeisung in den Ring verloren gegangenen sowie der nicht absendbaren Pakete geführt. Bei den Schaltern wird die Zahl der Adreßersetzungen protokolliert, die notwendig geworden sind, weil adaptives Routing durchgeführt wurde oder weil multiple B-Links vorhanden sind.

6.2.2 Paketverluste

Bei SCI können Pakete aus zwei Gründen verloren gehen. Der erste Grund liegt dann vor, wenn von einer Datenquelle mit zu hoher Rate versucht wird, Pakete in einen Ring einzuspeisen, und wenn gleichzeitig der Sendepuffer der SCI-Schnittstelle voll ist. Die Ursache für den Paketverlust dafür liegt darin, daß der Zugang zum Ring nicht zu jedem beliebigen Zeitpunkt möglich ist, sondern über spezielle Bandbreiteallozierungsprotokolle geregelt wird, wodurch für je-

den Knoten implizit ein Maximalwert der Sendebandbreite existiert. Der zweite Grund für Paketverluste ist gegeben, wenn Pakete nicht abgesendet werden konnten, auch wenn der Zugang zum Ring möglich gewesen wäre, weil der Sender bereits eine Maximalzahl offenstehender Transaktionen erreicht hat. Gemäß der IEEE-Spezifikation liegt diese Maximalzahl bei 64 Anforderungen, für die höchstens die Anwort ausstehen darf.

Die beiden beschriebenen Fälle von nicht erhaltenem Ringzugang sowie zu vielen offenstehenden Paketen stellen knotenkritische Ereignisse dar, da sie Auskunft darüber geben, ob sich ein Knoten im Grenzbereich seiner Leistungsfähigkeit befindet. In dem Abschnitt der Knoteninformationen sind diese Daten unter folgenden Namen protokolliert:

- *packetLossCounter* (= Packet Losses in [Packets]): Zahl der Pakete, die bei der Einspeisung in den Ring am Knoten verloren gingen, weil die Echtzeitbedingung „Senderate ≤ zugeteilte Bandbreite“ nicht eingehalten wurde.
- *transactionsNotIssued* (= Transactions not Issued in [Packets]): Zahl der Pakete, die vom Knoten nicht abgeschickt werden konnten, weil die Maximalzahl offenstehender Transaktionen erreicht war. Auch diese Pakete sind verloren.
- *PacLosBdW (net)* (= Packet Losses Bandwidth in MB/s]): Nettobandbreite, die über die zugeteilte Bandbreite hinausging und deshalb nicht übertragen werden konnte. Diese Größe drückt den Nettobandbreiteverlust aus, der durch Nichteinhaltung der Echtzeitbedingung entstanden ist.
- *TraNIsBdW (net)* (= Transactions not Issued Bandwidth in [MB/s]): Nettobandbreite, die über die Bandbreite der Maximalzahl offenstehender Transaktionen hinausging und deshalb nicht übertragen werden konnte.

Die korrespondierenden Größen aus dem Abschnitt der Gruppeninformationen lauten *GrpPacLosPGe*, *GrpTraNIsPGe*, *GrpPacLosPGeBdW (net)* und *GrpTraNIsPGeBdW (net)*.

6.2.3 Adreßersetzungen

Bei SCI-Schalterknoten gibt es drei Möglichkeiten, warum Ersetzungen von Paketadressen auftreten. Sie sind zum einen erforderlich, wenn an einem Schaltereingang eine Paketannahme adaptiv durchgeführt wurde, oder wenn zum anderen ein Schalterausgang adaptiv ausgewählt worden ist, oder wenn zum dritten eines von mehreren B-Links adaptiv selektiert wurde. Die damit zusammenhängende Vorgänge werden bei SCINET auch als adaptive Paketannahme, B-Link-Auswahl und Schalterausgangsauswahl bezeichnet. Alle drei Möglichkeiten sind unabhängig voneinander, so u.U. daß für ein Paket an einem SCI-Knoten alle drei Mechanismen aktiv sein können.

In der Ausgabedatei des Simulators gibt es beim Abschnitt der Knoteninformationen für jeden Schalterknoten ein ausführliche Analyse, wie oft eine

Adreßkorrektur durchgeführt wurde und von welcher Art sie war. Dabei wird erfaßt, wie viele Male der betrachtete Knoten selbst eine Korrektur durchgeführt hat, wie oft von einem anderen Knoten eine Korrektur zu der Adresse des betrachteten Knotens hin, und wie oft eine Korrektur von seiner Adresse weg durchgeführt worden ist. Daraus ergeben sich drei Kategorien, die in ebenso vielen Spalten in der Ausgabetabelle *Address Corrections* in SCINET repräsentiert sind. Die Spalten tragen die Bezeichnung *performedByThisNode*, *OffThisNode*, und *ToThisNode*. Für jede dieser Spalten existiert derselbe Satz von Korrekturinformationen, deren Bezeichner im Einzelnen lauten:

- *TargetNotEqualCurrentNodeInMultiBLinks:* Dieser Zähler gibt die Anzahl der Request-, Response- und Retry-Pakete an, die bei Schaltern mit multiplen B-Links von einem anderen Eingangsknoten akzeptiert worden sind als demjenigen, an den sie vom vorangegangenen Schalterausgang im selben Ring geschickt worden sind. D.h, es wird gezählt, wie oft ein Multi-B-Link-Eingangsknoten aufgrund von adaptiver B-Link-Auswahl Pakete annimmt, die nicht an ihn adressiert waren.
- *ForSourceAddressOfBusyEchoInMultiBLinks:* Dies ist die Zahl der Korrekturen für Paket-Herkunftsadressen von negativen Echopaketen, die von Schaltern mit multiplen B-Links ausgeschickt worden sind, bei denen zuvor das entsprechende Request-, Response- oder Retry-Paket von einem anderen Eingangsknoten akzeptiert worden ist als ursprünglich vorgesehen. D.h., es wird gezählt, wie oft aufgrund von adaptiver B-Link-Auswahl die Herkunftsadresse eines negativen Echopakets geändert wurde, bevor es abgeschickt worden ist.
- *ForSourceAddressOfNonBusyEchoInMultiBLinks:* Zahl der Korrekturen für Paket-Herkunftsadressen von positiven Echopaketen, die von Schaltern mit multiplen B-Links ausgeschickt worden sind, bei denen zuvor das entsprechende Request-, Response- oder Retry-Paket von einem anderen Eingangsknoten akzeptiert worden ist als ursprünglich vorgesehen. (Entspricht in der Semantik dem vorangegangenen Zähler, zählt jedoch nur positive Echos.)
- *ForRequestOrResponseInMultiBLinks:* Zahl der Korrekturen für Request-, Response- und Retry-Pakete, die bei Schaltern mit multiplen B-Links von keinem Eingangsknoten akzeptiert wurden. D.h., hier wird gezählt, wie oft Pakete aufgrund von adaptiver Paketannahme an dem Schalter vorbeigegangen sind, für den sie adressiert worden waren.
- *ForRequestOrResponseInSingleBLinks:* Zahl der Korrekturen für Request-, Response- und Retry-Pakete, die bei Schaltern mit einem B-Link von keinem Eingangsknoten akzeptiert wurden. (Entspricht in der Bedeutung dem vorangegangenen Zähler, gilt jedoch für für normale Schalter ohne multiple B-Links.)
- *ForRemotePortInMultiBLinks*: Zahl der Korrekturen für Pakete, die bei Schaltern mit multiplen B-Links aufgrund einer adaptiven Wahl des Schalterausgangs notwendig wurden. Dieser Wert gibt an, wie oft aufgrund von ad-

aptiver Schalterausgangsauswahl ein anderer als der reguläre Ausgang eines Multi-B-Link-Schalters ausgewählt worden ist.

- *ForRemotePortInSingleBLinks:* Zahl der Korrekturen für Pakete, die bei Schaltern mit einem B-Link aufgrund einer adaptiven Wahl des Schalterausgangs notwendig wurden. (Entspricht dem vorigen Zähler, gilt jedoch für normale Schalter ohne multiple B-Links.)

Im Abschnitt der Gruppeninformationen der Ausgabedatei werden in der Tabelle *GrpAddressCorrections* für die Kategorie *performedByThisNode* die Zeilenwerte der einzelnen Knoten aufsummiert. Zusätzlich stehen als statistische Informationen die Mittelwerte über alle Knoten sowie der jeweils größte Wert zur Verfügung. Die Bezeichnung der Summen entspricht den vorangegangenen Bezeichnern, jedoch ist zu deren Unterscheidung die Silbe „For“ weggelassen.

6.2.4 Paketlatenzzeiten

Der letzte Abschnitt in der Ausgabedatei des Simulators enthält Informationen zu den Latenzzeiten, die die Pakete im Netz erfahren, und wird demzufolge als *Latency Informations* bezeichnet. Darin ist für die einzelnen Paketquellen, die jeweils in deterministisch und stochastisch unterteilt sind, aufgelistet, wie viele Pakete ausgegeben wurden und wie lange der Mittelwert, das Maximum, das Minimum, die Standardabweichung und die Varianz der mit dem Paket verbundenen Transaktion gedauert hat. Daran kann man Asymmetrien der einzelnen Quellen erkennen. Den Abschluß schließlich bildet die Tabelle der aufsummierten Paketzahlen und der Minimal-, Mittel- und Maximalwerte der Latenzen. Diese werden mit *SumP/RGeDoTDurCnt*, *MinP/RGeDoTDurMin*, *MeaP/RGeDoTDurMea* und *MaxP/RGeDoTDurMax* bezeichnet.

6.3 Validierung des Simulators

Zur Validierung von Modell und Simulator wäre es am besten, die durchführten Simulationen einer SCI-Datenübertragung mit den an den Testständen erzielten Messungen zu vergleichen. Leider ist dies nicht möglich, weil die Leistungsfähigkeit einer Übertragungsstrecke wesentlich von den an die Schnittstellenkarten angeschlossenen Rechnern abhängt. Deren Zeitverhalten kann jedoch aus Komplexitätsgründen vom Simulator nicht abgedeckt werden. Eine einfache „Black Box“-Approximation der Rechner, die nur ihre Zeitverzögerung berücksichtigt, wurde als unzureichend erachtet. Die Verifikation des Simulators muß deshalb anhand anderer Methoden wie Plausibilitäts- und Konsistenzsprüfungen sowie mit Hilfe von begleitenden Überschlagsrechnungen durchgeführt werden.

Voraussetzung für eine korrekte Simulation ist in der Regel, daß sich alle Paketpuffer in den Knoten in einem stationären Zustand befinden, bevor mit der Erfassung statistischer Größen begonnen wird. Dies ist nur bei einem genügend langen „Vorlauf“ der Simulation sichergestellt.

Beispielsweise ist es für die Simulation der Paketverluste eines Sendeknotens erforderlich, daß sich der Paketaufnahmepuffer des betreffenden Senders in einem stationären, d.h „eingeschwungenen“ Zustand befindet. Im Fall der Paketverluste bedeutet das, daß der Puffer voll sein muß. Implementierungsseitig erfolgt die Bewertung des Füllgrades der Puffer nach dem Rücksetzen der Zählvariablen des Simulators, was default-mäßig nach 200 µs der Fall ist (sog. statistics reset time). Die zur Verfügung stehenden 200 µs Zeit bis zum Zählerrücksetzen ist ca. 340 mal länger als es dauert, die existierenden 4 Paketplätze des Sendepuffers der Schnittstelle über ihr B-Link zu füllen (4*84 Bytes * 1,66 ns/Byte = 557 ns), so daß der eingeschwungene Zustand tatsächlich erreicht werden kann.

Aus den genannten Überlegungen hinsichtlich des stationären Zustandes läßt sich ein Test ableiten, den jeder Simulator bestehen muß. Der Test besteht darin, ein und dieselbe Simulation für verschieden lange Simulationszeiten, die alle oberhalb der Einschwingzeit liegen, auszuführen. Dabei müssen die berechneten Ergebnisse identisch sein. In Tabelle 6.3.1 sind die Ergebnisse dieses Tests angewandt auf SCINET aufgelistet. Es wurde dazu der Durchsatz (NorReqPayBdW) an einem Empfängerknoten in einem Omega-Netz der Größe 8x8 berechnet. Man sieht, daß sich die Ergebnisse praktisch nicht ändern. Die Größe

SimTime in µs	NorReqPayBdW in MB/s
0,800	61,1
1,200	60,8
1,600	61
2,000	60,8
2,400	60,9
2,800	61
3,200	60,9

Tabelle 6.3.1: Simulationsergebnisse bei verschieden langen Simulationszeiten.

der Simulationsdauer hat also, wie es sein muß, ab dem eingeschwungenen Zustand keinen Einfluß auf die Resultate. In Ausnahmefällen kann zur Erfassung nicht-stationärer (transienter) Vorgänge darauf verzichtet werden, eine Simulation so lange laufen zu lassen, bis sich nichts mehr ändert. Allerdings muß man in diesen Fällen besondere Vorsicht walten lassen, um gesicherte Simulationsergebnisse zu erhalten. Zur weiteren Validierung von Modell und Simulator wird im nächsten Kapitel der einfache Spezialfall eines einzelnen SCI-Ringes

bestehend aus einem Sender und einem Empfänger simulativ und analytisch untersucht, und beide Ergebnisse werden miteinander verglichen.

7 Analyse von SCI-Ringen

7.1 Einleitung

Am Anfang der Leistungsanalyse eines SCI-Systems steht die Frage, welche Maßzahlen (Bandbreite, Latenz, etc.) jeder einzelne SCI-Ring besitzt. Im zweiten Schritt der Leistungsanalyse werden die SCI-Schalter untersucht, die die Ringe miteinander verbinden. Die Leistung des ganzen Systems ergibt sich im dritten Schritt aus der Kombination der jeweiligen Maßzahlen.

Zu den wichtigsten Metriken eines Ringes zählen der Durchsatz, die Latenzzeit und die Zahl der Paketverluste, die bei der Datenübertragung auftreten. Die ersten beiden Maße geben Auskunft über die Geschwindigkeit der Strecke und die letzte Maßzahl über deren Qualität. Im Falle des SCI-Protokolls, das eine automatischen Wiederholung (Retry) von Datenpaketen beinhaltet, die vom Empfänger negativ quittiert wurden, ist als weitere Metrik die Größe des Retry-Verkehrs hinzuzurechnen. Der Retry-Verkehr kostet Bandbreite, die von der Ringbandbreite abzuziehen ist, wenn man den Nettodurchsatz ermitteln will.

Die erwähnten Maßzahlen hängen von einer Reihe von Parametern ab, von denen der wichtigste der angebotene Verkehr ist, der in den betrachteten Ring eingespeist werden soll (offered input traffic). Andere Parameter wie Ringgeschwindigkeit, Puffergrößen und Leitungslängen definieren die Randbedingungen, unter denen das SCI-System arbeitet. Daraus ergibt sich, daß die Ergebnisse der Leistungsanalyse Diagramme darstellen, in denen die Nettowerte von Durchsatz, Retry-Verkehr, Latenz und Paketverluste als Funktion des angebotenen Verkehrs dargestellt sind. Die daraus erhaltenen Abhängigkeiten geben Auskunft über das Input/Output-Verhalten des betrachteten Rings, Schalters oder Netzes, dem jede Systemanalyse zugrunde liegt.

Paketverluste können bei SCI wegen der durch Protokolle und CRC gesicherten Datenübertragung nur dann auftreten, wenn ein Sender für eine gewisse Zeit schneller Pakete erzeugt als der Ring in der Lage ist aufzunehmen. Gemäß der SCI-Spezifikation sind pro Sender bis zu 64 offenstehende Transaktionen möglich, bevor das erste aufgrund des Handshakes erforderliche Echopaket beim Sender eintreffen muß. Die Netzwerkschnittstelle des betreffenden Senders hat einen durch die SCI-Bandbreiteallozierungsprotokolle geregelten Anteil an der gesamten Ringbandbreite und kann deshalb unter ungünstigen Umständen ihre gepufferten Pakete nicht mit derselben Rate auf den Ring geben, mit der sie erzeugt werden. Bei Verletzung dieser Echtzeitbedingung laufen zuerst die Puffer in der Netzwerkschnittstelle des Senders voll und danach gehen zu viel erzeugte Pakete verloren.

7.2 Elementarer SCI-Ring

Ein elementarer SCI-Ring besteht im einfachsten Fall aus einem einzigen Sender-/Empfängerpaar, dessen Sende- und Empfangsgeschwindigkeit aufeinander abgestimmt ist. In der Praxis treten jedoch die Fälle, bei denen der Sender entweder schneller oder langsamer Pakete erzeugt als sie der Empfänger verbrauchen kann, weitaus häufiger auf, als das aufeinander abgestimmte Sender-/Empfängerpaar. Um alle drei Lastfälle untersuchen und miteinander vergleichen zu können, ist es günstig, in einem Diagramm die Sendedatenrate von Null bis zum Maximalwert zu variieren, um so den gesamten Wertebereich des angebotenen Verkehrs abzudecken. In einem elementaren Ring entspricht der Maximalwert des Verkehrs der Ringbandbreite. Im Falle einer Simulation von Dolphin LC-II Link-Controller-Bausteinen sind dies 500 MB/s.

7.2.1 Durchsatz im Ring

Für eine Leistungsbeurteilung des elementaren Rings sind in Bild 7.2.1, neben

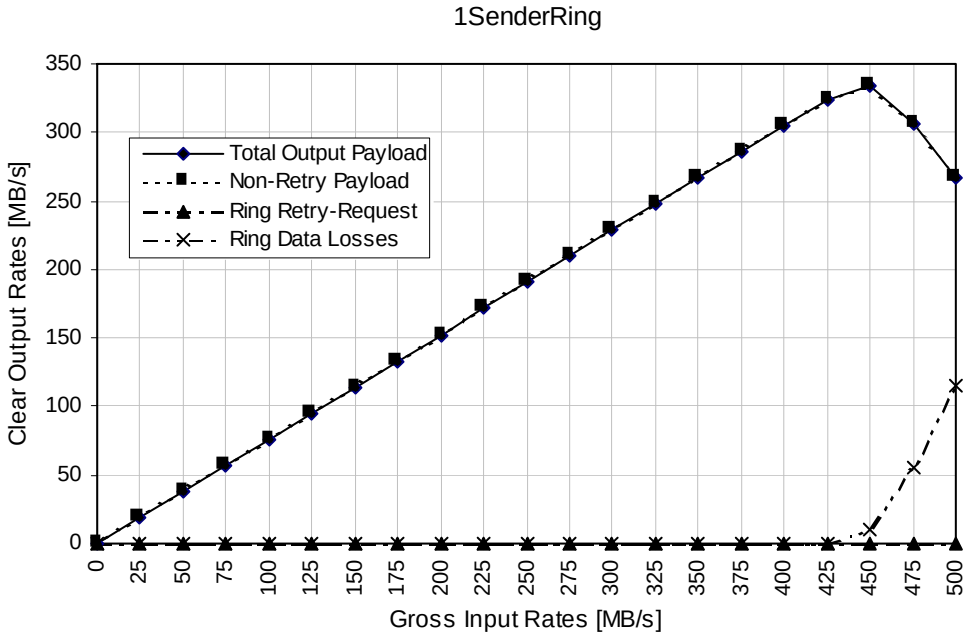

Bild 7.2.1: Durchsatz, Paketwiederholungen und -verluste im elementaren Ring.

dem gesamten Paketdurchsatz beim Empfänger (Total Output Payload) und dem Anteil, der ohne Paketwiederholungen aufgebracht wird (Non-Retry Payload), die Bandbreite der Paketwiederholungen auf dem Ring (Ring Retry-Requests) und die Paketverluste beim Sender (Ring Data Losses) in Abhängigkeit von dem angebotenen Verkehr dargestellt. Die Randbedingungen des Rings, unter denen die Simulation durchgeführt wurde, entsprechen den Default-Wer-

ten, wie sie in Kapitel 4.5 "Modellierung eines SCI-Knotens" bis Kapitel 4.8 "Modellierung eines SCI-Netzes" angegeben worden sind. Als SCI-Pakettyp wurde der NWRITE64-Befehl verwendet, der aus Request- und Response-Phase besteht, und bei dem 64 Byte Nutzdaten vom Sender zum Empfänger übermittelt werden. Als Sender wird ein deterministischer Paketgenerator verwendet, der in periodischen Intervallen Pakete ausgibt. Der angebotene Verkehr läßt sich somit präzise einstellen und entspricht der Datenrate des Paketgenerators. Die deterministische Art der Lasterzeugung entspricht der Datenerzeugung, wie sie auch bei einem Datenerfassungssystem erfolgt.

Alle dargestellten Ausgabegrößen sind Nettowerte, d.h. ohne Verwaltungszusätze (Overhead) wie Adressen oder Prüfsummen, so daß die Nutzdatenraten aus dem Diagramm direkt abgelesen werden können. Als Eingabegröße wird jedoch der gesamte eingespeiste Verkehr, d.h. der Bruttowert incl. Adressen, Prüfsummen und Idle-Symbolen verwendet, da dieser die tatsächliche Ringbelastung darstellt.

Aus dem Diagramm in Bild 7.2.1 wird ersichtlich, daß der erreichte Empfängerdurchsatz bis zu einer bestimmten Größe dem angebotenen Verkehr exakt entspricht. Der eingespeiste Brutto-Verkehr (Gross Input Rate) und die Nutzdaten des Empfängers stehen in einem streng linearen Zusammenhang. Die gesamten Nutzdaten beim Empfänger (Total Output Payload) und der Anteil der Nutzdaten, der nicht von Paketwiederholungen herrührt, bilden dieselbe Kurve. Die Steigung der Geraden spiegelt das Verhältnis der Pakete ohne und mit Verwaltungszusatzaufwand wieder. Beispielsweise erhält man bei 400 MB/s Eingangsrate einen Nettodurchsatz von 304,7 MB/s. Der Nettodurchsatz entspricht damit auf 3 geltenden Ziffern dem erwarteten Sollwert, der sich aus dem Verhältnis von Netto- zu Bruttopaketlänge und der eingespeisten Datenrate berechnet, d.h., es gilt: (64/84)*400 ≈ 304,7. Die vom Simulator berechneten Durchsatzwerte sind also sehr präzise, da der Empfänger genau die Datenmenge empfängt, die abgeschickt wurde.

Oberhalb einer Eingangsdatenrate von 425 MB/s, d.h. kurz vor Erreichen des maximalen Durchsatzes, flacht die Kurve leicht ab. Laut Bild 7.2.1 beträgt das Maximum des Durchsatzes eines elementaren SCI-Rings auf Basis des Dolphinschen LC-II Bausteins 333,3 MB/s. Dieser Wert wird bei 450 MB/s Eingangsrate erreicht. Der berechnete Sollwert an dieser Stelle beträgt (64/84)*450 MB/s = 342,9 MB/s, liegt also höher als das, was der Simulator ausgibt. Die Erklärung für die Abweichung liegt darin, daß es ab 450 MB/s zu Paketverlusten kommt. Diese betragen nach den Berechnungen des Simulators 9,5 MB/s, analytisch sind es (342,9-333,3) MB/s = 9,6 MB/s. Beide Werte stimmen sehr gut überein und erklären das Abflachen des Durchsatzes zwischen 425 und 450 MB/s.

Bei Raten höher als 450 MB/s geht der Durchsatz sogar wieder zurück, und die Paketverluste steigen im gleichen Maße an. Hier zeigt sich ein nichtlinearer Effekt der Sättigung, der durch den Ring selber verursacht wird. Bei 500 MB/s Eingangsrate hat man noch 266,3 MB/s Durchsatz bei 114,7 MB/s an Paketverlusten (analytisch gilt: 266,3+114,7 = (64/84)*500). Bei diesen Datenraten ist der SCI-Ring alleine bereits ein begrenzendes Element.

Die Tatsache, daß bei 450 MB/s Paketverluste in Höhe von 9.5 MB/s existieren deuten darauf hin, daß ein auf dem Diagramm nicht sichtbarer Maximalwert etwas unterhalb von 450 MB/s existieren muß, bei dem noch alle erzeugten Pakete vom Ring übernommen werden. Dort wird zugleich der Maximalwert des Durchsatzes erreicht. Die nachstehende Überlegung erlaubt, exakt zu berechnen, bei welcher Eingangsrate dieser Punkt sich befindet.

Allgemein gilt, daß Paketverluste ein Indiz dafür sind, daß das Ringsegment zwischen Sender und Empfänger überlastet ist. Für die Berechnung der im Überlastungsfall vom Ring transportierten Daten muß berücksichtigt werden, daß zusätzlich zu den 84 Byte pro Request-Paket noch 12 Echo-Byte übertragen werden müssen, während gleichzeitig auf der Rückrichtung des Rings 16 Byte pro Response-Paket und 12 Byte pro Echo transferiert werden. Da der elementare SCI-Ring aus 2 Segmenten besteht, können die Transfers in Hin- und Rückrichtung jedoch unabhängig voneinander ablaufen, ohne sich gegenseitig der Bandbreite zu berauben.

Die Summe der Längen von Request- und Echopaket ergibt 96 Bytes. Daraus resultiert, daß in einen elementaren SCI-Ring mit einer Bandbreite von 500 MB/s pro Segment maximal Requests mit einer Bruttodatenrate von (84/96)*500 MB/s = 437,5 MB/s eingespeist werden können, ohne daß Paketverluste auftreten. An diesem Punkt sind (64/84)*437,5 MB/s = 333,3 MB/s Nettodurchsatz zu erwarten. Bei höheren Raten als 437,5 MB/s gehen Daten verloren, weil der Sender stationär schneller Pakete produziert als sie seine SCI-Schnittstelle über den Ring abtransportieren kann.

Versucht man beispielsweise 450 MB/s Eingangsdatenrate in den Ring einzuspeisen, sind dies laut Rechnung 12,5 MB/s zuviel, was netto in Paketverlusten von (64/84)*12,5 MB/s = 9,5 MB/s resultiert. Subtrahiert man die Paketverluste vom berechneten Sollwert, erhält man denselben Wert wie bei 437,5 MB/s Eingangsrate, nämlich 333,3 MB/s. Beide Werte wurden vom Simulator ebenfalls geliefert, wodurch Rechnung und Simulation sich gegenseitig bestätigen. Für alle Eingangsraten, die oberhalb von 437,5 liegen, sollte man unter Vernachlässigung der nichtlinearen Ringsättigung ebenfalls 333,3 MB/s Nettodurchsatz beim Empfänger erhalten, da jeder zusätzlich eingespeiste Verkehr in gleichem Maße in Paketverluste umgesetzt wird und so den Durchsatz nicht verändern kann. Es müßte sich also in diesem Bereich im Diagramm das für Sättigung typische Hochplateau ergeben.

Wie sich der Durchsatz in Wirklichkeit verhält, zeigt die Ausschnittsvergrößerung der Simulation in Bild 7.2.2. Tatsächlich steigt der Durchsatz bis zum berechneten Wert von 437,5 MB/s an und erreicht dort sein Maximum, und im Bereich um 450 MB/s ergibt sich das vorausgesagte Sättigungshochplateau, das ab 452,5 MB/s aufgrund der Ringsättigung wieder abfällt. Zusätzlich zeigt die Simulation einen nicht vorausgesagten Effekt zwischen 437,5 MB/s und 447,5 MB/s, der bewirkt, daß es in diesem Bereich zu einem lokalen Abfall des Durchsatzes kommt.

Insgesamt stimmen Rechnung und Simulation sehr gut überein, wodurch die Präzision von SCINET bestätigt wird. Der Vorteil der Simulation ist, selbst bei dem sehr einfachen System eines elementaren SCI-Rings, daß der Simulator

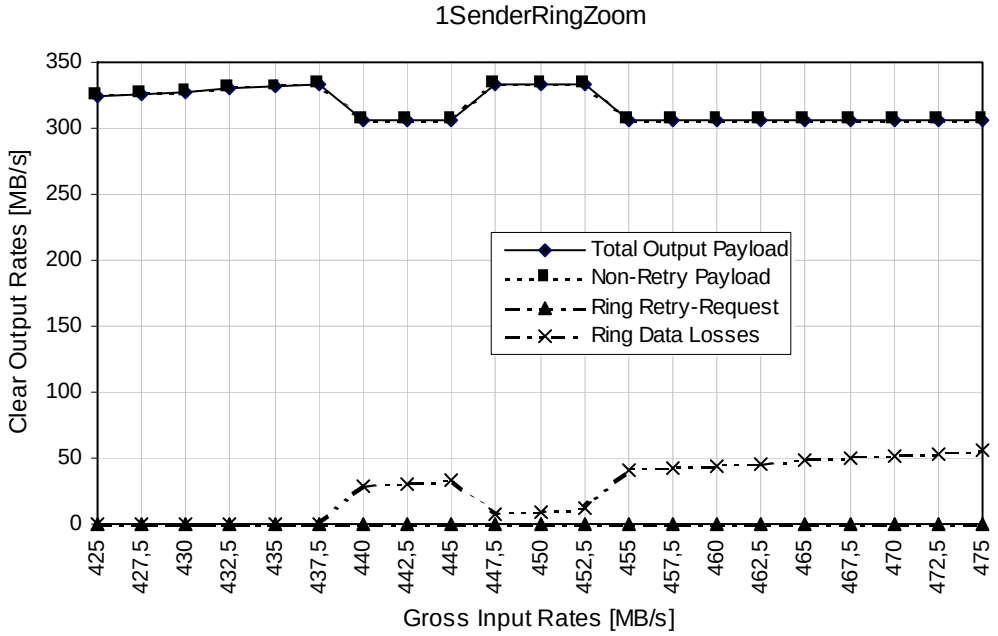

Bild 7.2.2: Ausschnittsvergrößerung der Simulation von Bild 7.2.1.

noch zusätzliche Details und Effekte enthüllen kann, die analytisch nicht sichtbar geworden wären.

Die Simulationen zeigen auch, daß Paketwiederholungen (Ring Retry-Requests) nicht auftreten. Offenbar ist der Empfänger unter den gewählten Simulationsparametern stets schneller als der Sender. Daß dies so sein muß, zeigt folgende Rechnung:

Die Geschwindigkeit, mit der ein SCI-Knoten Pakete aufnehmen kann, wird im wesentlichen bestimmt von seiner Adreßdekodierungszeit (AddressDecoderDelay), von der Zeit, die für den Pakettransfer vom Eingabepufferausgang bis zum B-Link vergeht (InFIFOOutToBLinkDelay), und der Anforderungsbearbeitungszeit (RequestDelay). Im Falle des Dolphin LC-II Link-Controller-Bausteins sind dies (20 + 106 + 40) ns = 166 ns. Das heißt, daß maximal alle 166 ns ein neues Paket von 84 Byte Bruttolänge akzeptiert werden kann, was einer Datenrate von 506 M/s entspräche. Dies ist mehr als der Ring zu transportieren vermag, so daß das limitierende Glied in der Übertragungskette nicht der Empfängerknoten ist. Deshalb treten Paketwiederholungen nicht auf.

Zusammenfassend kann gesagt werden, daß die simulativ und analytisch ermittelnden Werte ein hohes Maß an Übereinstimmung zeigen. Als Ergebnis konnte festgestellt werden, daß es im Bandbreitediagramm des elementaren SCI-Rings die drei Phasen „linearer Anstieg“, „Sättigung“ und „nichtlinearer Abfall“ gibt, wobei der lineare Anstieg den bei weitem größten Teil darstellt. Der maximale Durchsatz von 333,3 MB/s wird am Sättigungspunkt bei 437,5 MB/s Eingangsdatenrate erzielt.

7.2.2 Latenzzeit im Ring

Das zweite Maß, das über die Geschwindigkeit einer Datenübertragung Auskunft gibt, ist die Latenzzeit. Bei SCINET ist dies die Zeit, die vom Absenden des 1. Bytes eines Pakets bis zum Empfang seines letzten Bytes benötigt wird. Die Latenzzeit unterscheidet sich also von der Setup-Zeit, die angibt, wie lange es vom Absenden des 1. Bytes eines Pakets bis zum Empfang dieses Bytes dauert. In die Latenzzeit geht die Leitungslänge des Rings in Form des „link delays“ ein, das die Signallaufzeit zwischen benachbarten Knoten berücksichtigt.

In Bild 7.2.3 ist der Maximal-, Mittel- und Minimalwert der Latenzzeit in Ab-

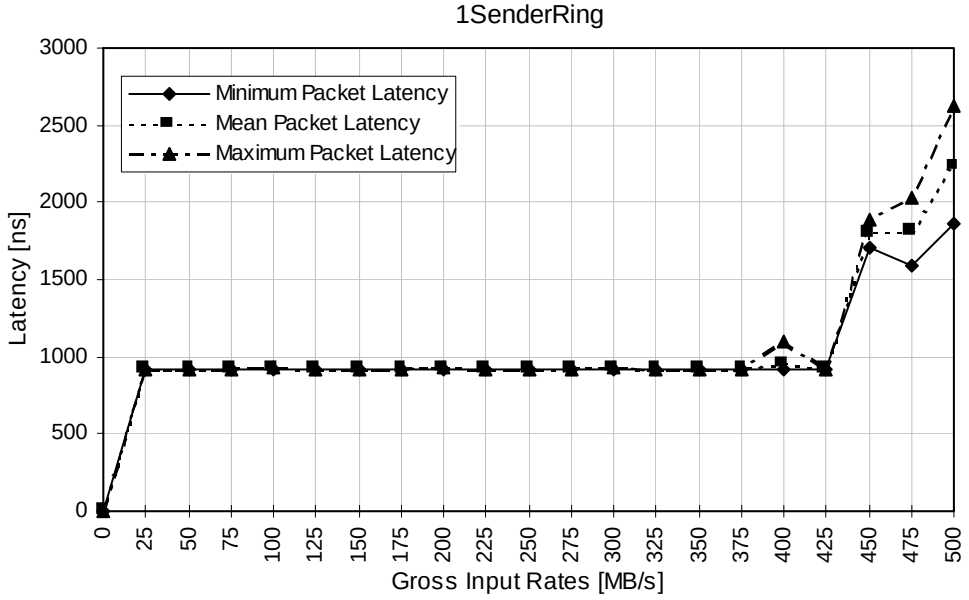

Bild 7.2.3: Latenzzeiten beim elementaren SCI-Ring in ns (1 ns = 10^{-3} μs).

hängigkeit von dem eingespeisten Verkehr aufgetragen. Aus diesem Diagramm wird ersichtlich, daß bis zu einer Eingangsdatenrate von 375 MB/s (näherungsweise bis 425 MB/s) alle drei Kurven identisch sind und einen konstanten Wert von 915 ns aufweisen. In der Zeit von 915 ns sind die Setup-Zeiten der Schnittstellen, die Pakettransferzeiten und link delays für Request-, Response und Echopakete mit eingeschlossen. Dies ist insofern bemerkenswert niedrig, weil bei den Simulationen davon ausgegangen wird, daß Sender und Empfänger in einer Entfernung von 20 m voneinander aufgestellt sind, was in einer durch das Kabel bedingten Latenz von 100 ns pro Paket resultiert. (In einem Kupferkabel beträgt die Ausbreitungsgeschwindigkeit elektrischer Signale ca. 20cm/ns). Es wurde eine Entfernung von 20 m zwischen Sender und Empfänger gewählt, um die Verhältnisse bei einem Datenerfassungssystem nachzubilden.

Das heißt, daß bei SCI die Zeit für eine komplette NWRITE64-Transaktion bestehend aus 64 Byte Nutzdaten ca. 515 ns beträgt, da 4 Kabellaufzeiten für Request-, Echo-, Response- und nochmaliges Echopaket abgerechnet werden müssen. Umgerechnet entspricht dies einer Nettodatenrate von ca. 124 MB/s

für einen Transfer, bei dem nicht lange Blöcke, sondern nur ein einziges Paket übertragen wird.

Interessant ist ferner, daß die Latenz bei Eingangsraten bis ca. 425 MB/s (exakt sind es 437,5 MB/s) trotz zunehmenden Verkehrsangebots konstant bleibt, was daher kommt, daß der Ring exklusiv einem einzigen Sender-/Empfängerpaar zur Verfügung steht.

Ab 425 MB/s (bzw. 437,5 MB/s) Eingangsrate springt die Latenzzeit und beträgt bei 450 MB/s 1890 ns und bei 500 MB/s 2629 ns (jeweils Maximum). Während zwischen 425 MB/s und 450 MB/s die Maximal- und Minimalwertwerte noch rel. dicht beieinander liegen, fächern sie ab 450 MB/s stark aus. Zur Erinnerung: Beim Durchsatzdiagramm waren 425 MB/s der Wert, bis zu dem der Ring streng lineares Verhalten zeigte, ab 425 MB/s ging der Ring in die Sättigung, und von 450 MB/s an sank der Durchsatz wieder ab. Die Eingangsraten, bei denen sich das Verhalten der Latenz jeweils ändert, korrelieren also gut mit dem Durchsatzverhalten. Da bei SCINET Durchsatz und Latenz unabhängig voneinander berechnet werden, ist dies ein weiteres Indiz für die Verläßlichkeit des Simulators.

Zusammenfassend kann man sagen, daß es im Latenzdiagramm des elementaren SCI-Rings die drei Phasen „konstante Latenz“, „ansteigende Latenz“ und „ausfächernde Latenz“ gibt, die mit den Phasen des Durchsatzes korrelieren. Bis zum Sättigungspunkt hat man eine konstante und damit verhersagbare Latenz von 915 ns (incl. der durch die Leitungslängen bedingten Verzögerungen).

Ergebnis:

Elementare SCI-Ringe bestehend aus einem Sender und einem gleich schnellen oder schnelleren Empfänger haben einen sehr hohen Nettodurchsatz von bis zu 333 MB/s bei 437,5 MB/s Bruttoeingangsdatenrate und eine sehr geringe und vor allem deterministische Latenz von 915 ns, sofern man sie unterhalb des Sättigungspunktes von 437,5 MB/s betreibt. Oberhalb der Sättigung steigt die Latenzzeit an und wird indeterministisch, jedoch läßt sich das Intervall in Form eines Maximal- und Minimalwertes angeben, in dem sie sich bewegt.

7.2.3 Relevanz für Datenerfassungssysteme

Vorhersagbare Transferraten und Latenzzeiten sind für Datenerfassungssysteme oder für Echtzeitsteuerungen und Regelungen von großer Bedeutung. Nach den bisherigen Ermittlungen kann ein SCI-Ring unter bestimmten Bedingungen diese Forderungen erfüllen. Die Bedingungen, bei denen der Durchsatz eines Ringes sich streng linear verhält (=deterministische Datentransferrate) und die Latenz einen konstanten Wert aufweist (=deterministische Latenzzeit), sind:

- Pro SCI-Ring darf es nur einen Sender geben,
- Jeder Ring muß unterhalb seiner Sättigungsgrenze betrieben werden, so daß

Paketwiederholungen nicht auftreten.

In der Praxis ist die Einhaltung dieser Bedingung nicht zu gewährleisten, so daß im weiteren untersucht wird, wie sich SCI bei multiplen Sendern, bei Überlastung und bei Paketwiederholungen verhält.

7.3 Leistungsanalyse bei Retry-Verkehr

Bei SCI ist die Datenzustellung vom Sender zum Empfänger garantiert. Pakete, die transient nicht akzeptiert werden, werden solange wiederholt, bis sie angenommen worden sind. Die negative Seite der Paketwiederholungen ist, daß der Ring belastet wird, wodurch die für „normale“ Pakete zur Verfügung stehende Bandbreite sinkt, während gleichzeitig die Latenz ansteigt. Für die Leistungsanalyse eines elementaren SCI-Rings ist es wichtig, zu wissen, welche Werte Latenz und Nutzbandbreite bei Paketwiederholungen annehmen.

In diesem Zusammenhang muß festgelegt werden, in welchem zeitlichen Abstand die Wiederholungen für ein nicht-akzeptiertes Datenpaket durchgeführt werden sollen. Seitens der SCI-Spezifikation von IEEE ist hierüber nichts ausgesagt, so daß bei SCINET die Zeit zwischen zwei Wiederholungen desselben Pakets ein frei wählbarer Parameter darstellt, der als „RetryDelay“ bezeichnet wird.

In Bild 7.3.1 ist der Paketdurchsatz beim Empfänger (Total Output Payload), der Anteil, der ohne Paketwiederholungen aufgebracht wird (Non-Retry Payload), die Bandbreite der Paketwiederholungen auf dem Ring (Ring Retry-Requests) und die Bandbreite der Paketverluste beim Sender (Ring Data Losses) in Abhängigkeit von dem angebotenen Verkehr dargestellt. Beim Sender wurde als RetryDelay 1 ns gewählt, was heißt, daß gleichsam sofort nach Eintreffen eines negativen Echos die zugehörige Paketwiederholung durchgeführt wird. Der Empfänger hat eine gegenüber Bild 7.2.1 erhöhte, willkürlich gewählte Anforderungsbearbeitungszeit (RequestDelay) von 400 ns, so daß Paketwiederholungen provoziert werden. Das Ziel dabei ist, den Durchsatz beim Empfänger zu untersuchen.

Das Diagramm von Bild 7.3.1 läßt sich in zwei Phasen unterteilen: bei Eingangsdatenraten unterhalb von 150 MB/s verhält sich der Paketdurchsatz streng linear, darüber ist der Empfänger gesättigt, wie man am waagrechten Verlauf des Durchsatzes erkennen kann. Am Knickpunkt von 150 MB/s werden 114,3 MB/s Gesamtdurchsatz erzielt, was sehr gut dem berechneten Sollwert von (64/84)*150 MB/s entspricht. Der bei 500 MB/s erreichte Nettodurchsatz beträgt 116,8 MB/s und liegt nur unwesentlich über dem bei 150 MB/s Eingangsrate erzielten Wert. Folgende überschlagsmäßige Rechnung zeigt, daß der Empfänger tatsächlich bei ca. 150 MB/s in die Sättigung kommen muß:

Die Summation der Zeiten für Adreßdekodierung (AddressDecoderDelay), interner Transferzeit bis zum B-Link (InFIFOOutToBLinkDelay) und Bearbei-

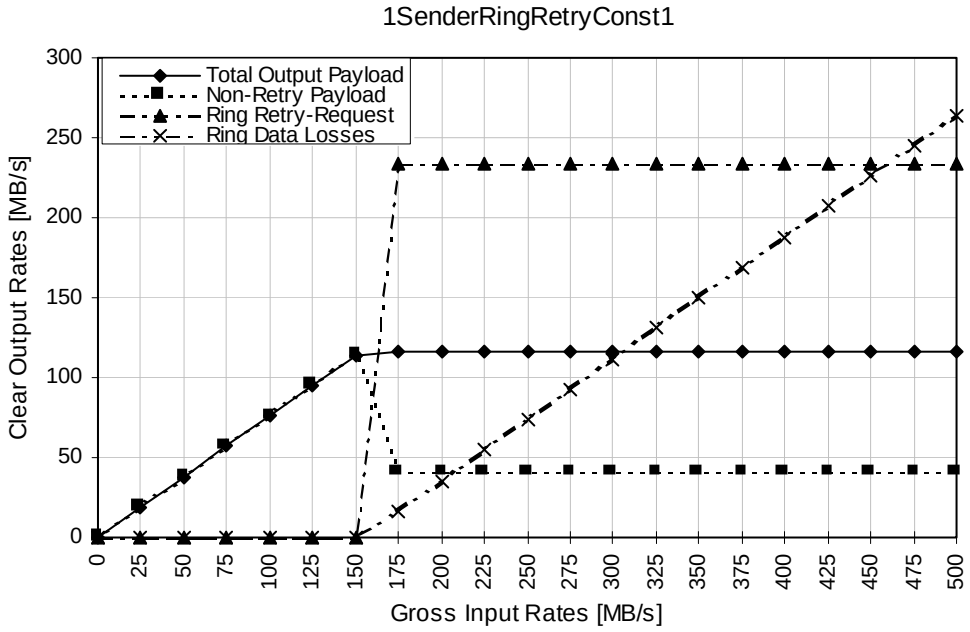

Bild 7.3.1: Durchsatz auf dem elementaren SCI-Ring bei Retry-Verkehr.

tungszeit am Empfängerknoten (RequestDelay), ergibt einen Wert von 526 ns. Das heißt, daß der Empfänger höchstens alle 526 ns ein neues Paket von 84 Byte Bruttolänge akzeptieren kann, was einer Sendedatenrate von 159,7 MB/s entspricht. Oberhalb dieses Wertes ist der Empfänger gesättigt.

Aufgrund der Sättigung des Empfängers steigen ab dem Knickpunkt die Paketverluste in genau dem Maße an, wie sich die Eingangsrate erhöht. Bei 500 MB/s Eingangsverkehr beispielsweise ergeben sich 264,3 MB/s Paketverluste, die sich zusammen mit den akzeptierten 116,8 MB/s zu 381,1 MB/s addieren, dem Wert, der ohne Retry-Verkehr und ohne Ringsättigung erreicht werden würde (analytisch: (64/84)*500≈381).

Interessant ist, daß ab dem Knickpunkt von 150 MB/s der Anteil der Daten, die vom Empfänger beim ersten Mal akzeptiert wurden (non-Retry payload) schlagartig auf 41,5 MB/s abfällt, während der Retry-Verkehr auf dem Ring genauso sprunghaft auf 233,7 MB/s ansteigt, was eine erhebliche Ringbelastung darstellt. Die Summe aus Retry-Requests und nicht-Retry-Requests ergibt 275,2 MB/s und erreicht noch nicht das Bandbreitelimit. Warum sich diese beiden Kurven am Sättigungspunkt sprungartig verändern, wird in Kapitel 7.4 "Reduzierung des Retry-Verkehrs" erklärt.

Ähnlich wie der Durchsatz verhält sich auch Latenz, sobald der Knickpunkt der Sättigung überschritten wird. Wie Bild 7.3.2 zeigt, springt der Maximalwert der Latenz bei 150 MB/s von 1275 ns auf bis zu 8678 ns, was das 1,4 bzw. 3,3-fache gegenüber den Zahlen in Bild 7.2.3 ist. Unterhalb der Sättigung ist die Latenz um genau den Betrag größer, um den die Anforderungsbearbeitungszeit beim Empfänger (=RequestDelay) erhöht wurde, nämlich um 915 ns + 360 ns = 1275 ns. In diesem Bereich bleibt die Latenz deterministisch. Sobald jedoch

Retry-Verkehr auftritt, ist nur noch vorhersagbar, daß sie zwischen 4508 ns (Minimalwert) und 8678 ns (Maximalwert) liegen wird. Immerhin sind die Maximal- und Minimalwerte nicht von der Eingangsdatenrate abhängig.

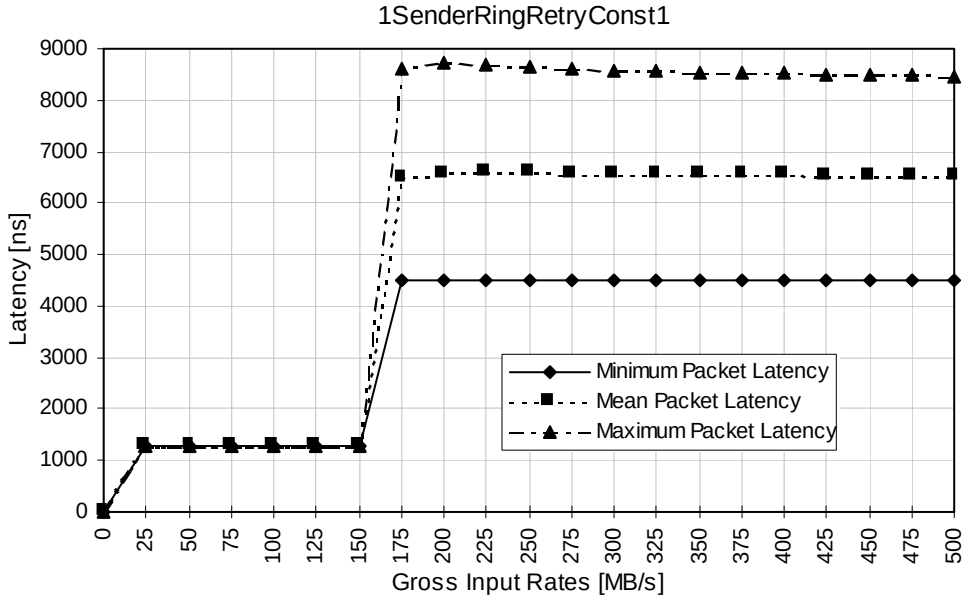

Bild 7.3.2: Latenz im elementaren SCI-Ring bei Retry-Verkehr.

Zusammenfassend kann man sagen, daß sich das Verhalten eines elementaren SCI-Rings oberhalb der Datenrate, bei der der Sender schneller sendet als der Empfänger Daten aufnehmen kann, stark verändert. Am Knickpunkt der Empfängersättigung geht der Durchsatz in eine waagrechte Gerade über, während die Paketverluste im gleichen Maße wie die Senderate ansteigen. Zugleich werden mehr als die Hälfte der Ringbandbreite (233,7 MB/s) für Paketwiederholungen verbraucht. Der Maximalwert der Latenz steigt bei einem langsamen Empfänger oberhalb des Sättigungspunkts um den Faktor 3,3 an und wird indeterministisch.

7.4 Reduzierung des Retry-Verkehrs

Die Leistungsanalyse des elementaren SCI-Rings bei Retry-Verkehr hat ergeben, daß netto 233,7 MB/s der Ringbandbreite für Paketwiederholungen verbraucht werden, was einen erheblichen Teil der Ringbandbreite darstellt. Der Empfänger akzeptiert davon nur (116,8-41,5) MB/s = 75,3 MB/s, so daß im Schnitt 233,7/75,3 = 3,1 gleiche Retry-Pakete auf dem Ring kreisen müssen, bevor eines davon angenommen wird. Zur Erläuterung der Rechnung: 116,8

MB/s waren der maximale Durchsatz beim Empfänger und 41,5 MB/s sind der Anteil an Bandbreite, der vom Empfänger beim ersten Versuch, d.h. ohne Paketwiederholungen (Non-Retry Payload) akzeptiert worden ist.

Interessant ist, daß sich gemäß des vorigen Kapitels die Latenz auf das 3,3-fache bei Retry-Verkehr erhöht, was ebenfalls darauf hindeutet, daß ca. 3 mal mehr Pakete im Ring unterwegs sind. Als Optimierung des SCI-Rings wird man deshalb versuchen, den Retry-Faktor von 3 auf 2 oder 1 zu reduzieren. Die Idee dazu ist, zwischen zwei nachfolgenden Wiederholungen desselben Pakets eine Verzögerungszeit einzufügen, so daß die Retry-Rate für das betreffende Paket absinkt. Zu beachten ist, daß ein solches Retry-Delay nicht in den Link-Controller-Bausteinen der Fa. Dolphin implementiert ist, es widerspricht jedoch nicht dem IEEE-SCI-Standard, der keine Festlegung in dieser Hinsicht trifft. In Bild 7.4.1 ist das Ergebnis der Bemühungen dargestellt.

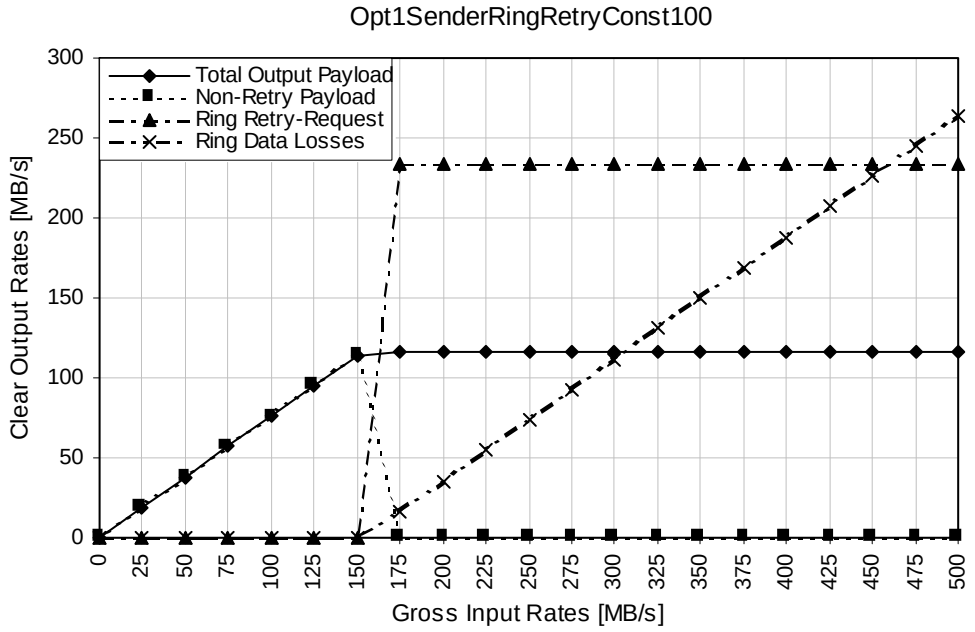

Bild 7.4.1: Leistungsanalyse des SCI-Rings bei 100 ns Retry-Verzögerungszeit.

Überraschenderweise hat sich gegenüber den Verhältnissen ohne Retry-Verzögerung fast nichts verändert. Lediglich der Anteil der Durchsatzes beim Empfänger, der nicht aus Paketwiederholungen bezogen wurde (non-Retry-payload), hat sich auf Null reduziert. Der Retry-Verkehr auf dem Ring (ring Retry-Request) hat jedoch nicht abgenommen. Auch eine Erhöhung des Verzögerungszeit auf 1000 ns bringt nicht das gewünschte Resultat. Der Retry-Verkehr bleibt unverändert. Zusätzlich nehmen die Latenzzeiten um so mehr zu, je größer die Retry-Verzögerungszeit wird (Bild 7.4.2). Bei 100 ns Verzögerung steigt die Latenz bei Paketwiederholungen um 692 ns auf 9370 ns an. Bei 1000 ns Retry-Verzögerung erhält man sogar 14185 ns Latenz. Ohne Paketwiederholungen sind die Latenzen, so wie es sein muß, für alle drei Retry-Verzögerun-

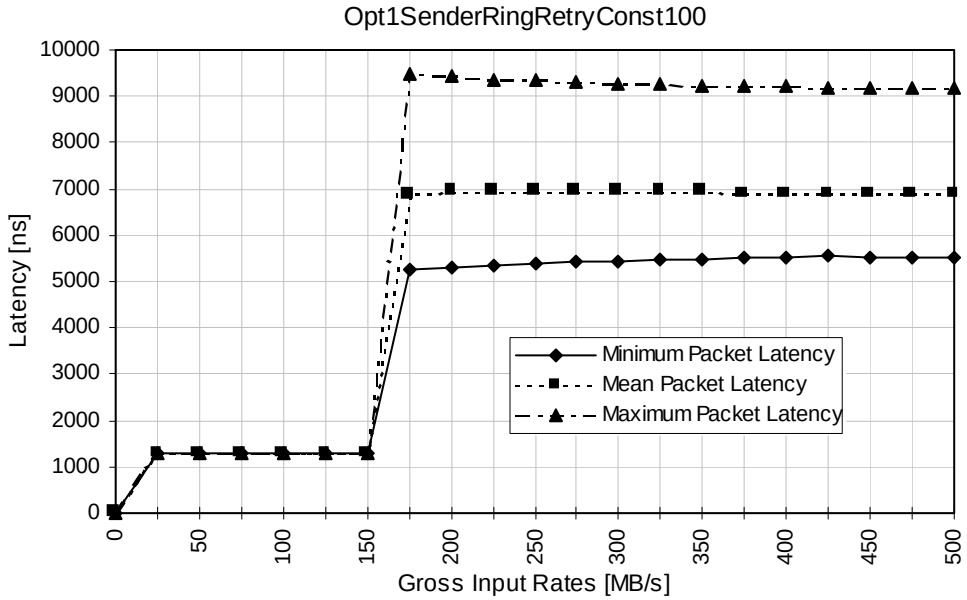

Bild 7.4.2: Latenz des SCI-Rings bei einer Retry-Verzögerungszeit von 100 ns.

gen gleich. Daß dieses unerwartete Ergebnis trotzdem korrekt ist, zeigt folgende Überlegung:

Wird von einem Sender stationär auch nur 1 Paket pro Sekunde mehr in einen SCI-Ring eingespeist, als der Empfänger aufnehmen kann, akkumulieren sich im Laufe der Zeit die überschüssigen Pakete in immer größerer Zahl, da auf dem Ring aufgrund seiner gesicherten Übertragung kein Paket verloren gehen kann. Beispielsweise resultiert eine stationäre Senderate von 110 MB/s und eine feste Empfangsgeschwindigkeit von 100 MB/s in einer „Überschußproduktion von 10 MB/s. Die Paketakkumulation bewirkt, daß nach 40 s insgesamt 40*10 MB/s + 100 MB/s = 500 MB/s an Daten auf dem Ring kreisen, so daß dessen Bandbreitelimit erreicht ist. Berücksichtigt man zusätzlich die Echopakete, wird das Limit schon nach kürzerer Zeit erreicht. Es kann also die Überschußproduktion von 10 MB/s nach höchstens 40 s Laufzeit nicht mehr in den Ring eingespeist werden und muß vollständig in Paketverluste umgesetzt werden. Daraufhin stellt sich ein eingeschwungener Zustand ein, bei dem permanent 500 MB/s an Daten auf dem Ring kreisen, wovon stationär 100 MB/s dem Ring entnommen werden, während gleichzeitig neue 100 MB/s in den Ring eingespeist werden.

Nach den bisherigen Überlegungen würde es im Vergleich zur Simulationsdauer, die default-mäßig 800 µs beträgt, zu lange dauern, bis der beschriebene stationäre Zustand erreicht wird. Dabei wurde jedoch nicht berücksichtigt, daß jedes überschüssige Paket durch dessen Paketwiederholungen zusätzlichen Verkehr auf dem Ring verursacht. Bei einer Verzögerungszeit von 100 ns beispielsweise würde bereits eine einmalige Überschußproduktion von einem einzigen Paket in einer brutto-Retry-Datenrate von 84 B/(168+100) ns = 313 MB/s resultieren. (168 ns ist die Zeit, die ein SCI-Ausgang benötigt, um 84 Byte

auszusenden.) Bei dieser Retry-Datenrate wäre das Bandbreitelimit des Rings und damit der eingeschwungene Zustand bereits nach weniger als 2 s erreicht. Bei mehr als einem Paket Überschußproduktion geht es entsprechend schneller.

In dem Diagramm nach Bild 7.4.1 werden ab dem Knickpunkt von 150 MB/s Eingangsdatenrate sehr viel mehr Pakete als nur eines zuviel produziert, von denen jedes versucht, für sich 313 MB/s für Paketwiederholungen zu beanspruchen. Damit ist klar, daß das Bandbreitelimit des Rings bereits nach wenigen hundert ns erreicht wird, so daß der eingeschwungene Zustand sich in der zur Verfügung stehenden Simulationszeit einstellen kann.

Das bedeutet, daß im Simulationsdiagramm von Bild 7.3.1 die Kurve für den Retry-Verkehr (Ring Retry-Request) ab 150 MB/s tatsächlich auf den von den SCI-Bandbreiteallozierungsprotokollen zugewiesenen Wert springen muß. Den Protokollen, die die zur Verfügung stehende Ringbandbreite gleichmäßig aufteilen, ist es zu verdanken, daß der Retry-Verkehr nicht die gesamte Bandbreite okkupiert.

Um die geschilderte Überlegung zu verifizieren, wurde ein weiterer Simulationslauf unternommen, bei dem der Durchsatz, der Retry-Verkehr auf dem Ring und die Paketverluste in Abhängigkeit von der Retry-Verzögerungszeit untersucht werden. Der Sender produziert Daten mit einer Rate von 200 MB/s, erzeugt also 50 MB/s zuviel. Nach der Überlegung sollten die obigen Maße von der Verzögerungszeit unabhängig sein, solange wie die Simulationsdauer ausreicht, um den eingeschwungenen Zustand zu erreichen. Das Simulationsresultat nach Bild 7.4.3 bestätigt voll diese Theorie. Alle Werte sind konstant.

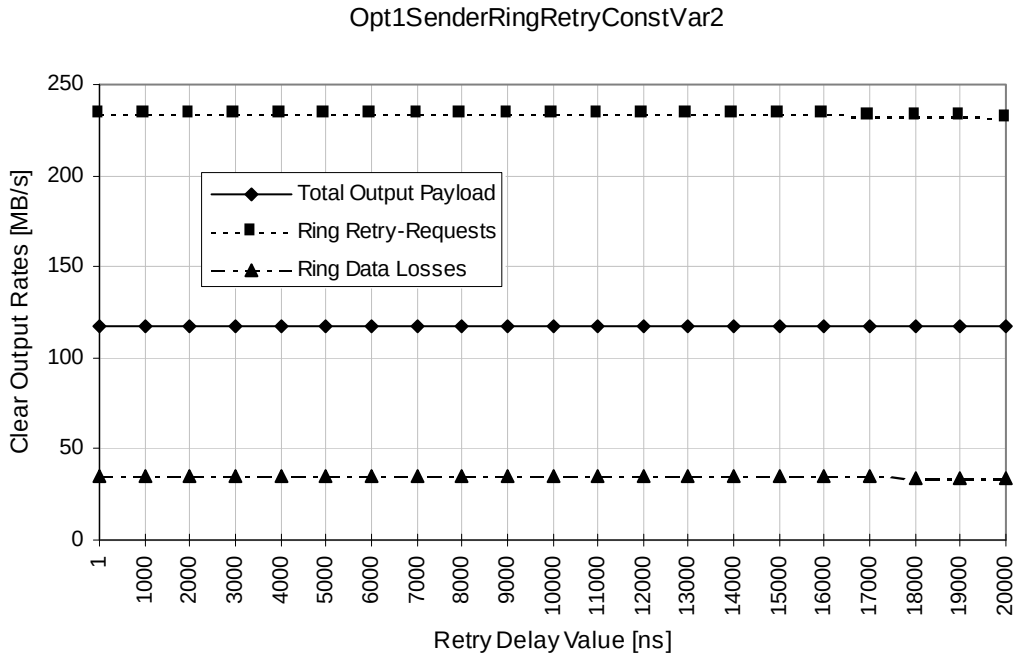

Bild 7.4.3: Leistungsdaten des SCI-Rings als Funktion von der Retry-Verzögerungszeit.

Das bedeutet, daß ein durch stationäre Überschußproduktion verursachter Retry-Verkehr nicht durch Einführung einer Retry-Verzögerungszeit reduziert werden kann. Eine dauerhafte Nichtanpassung von Sende- und Empfangsrate führt unweigerlich zum Erreichen eines Endzustandes, bei dem der Retry-Verkehr den durch die Bandbreiteallozierungsprotokolle vorgegebenen Maximalwert annimmt. Die Frage ist nur, ob ein Retry-Verkehr, der durch einen transient zu schnellen Sender bzw. zu langsamen Empfänger verursacht wird, durch eine Retry-Verzögerungszeit vermindert wird.

7.4.1 Reduzierung transienten Retry-Verkehrs

Die Simulation technischer Systeme basiert auf zwei Grundvoraussetzungen. Zum einen wird angenommen, daß das System einen eingeschwungen Zustand aufweist, zum anderen geht man davon aus, daß die Simulationsdauer groß genug gewählt wurde, so daß das System den stationären Zustand eingenommen hat, bevor die Aktivierung der Zählvariablen, d.h. die statistische Datenerfassung beginnt. Beide Voraussetzungen bewirken, daß die Simulationsergebnisse ab dem Erreichen des stationären Zustands nicht mehr von der Simulationsdauer abhängen, d.h., jede weitere Erhöhung der Simulationsdauer muß zu denselben Resultaten führen. Das bedeutet umgekehrt, daß eine Erfassung transienter Vorgänge nicht möglich ist. Die Untersuchung der Abhängigkeit vorübergehenden Retry-Verkehrs von der Retry-Verzögerungszeit stellt deshalb ein implementierungstechnisches Problem dar. Bei der Simulation muß der Wert der Retry-Verzögerungszeit variiert werden. Eine zu große Erhöhung der Verzögerungszeit bewirkt jedoch, daß man den eingeschwungenen Zustand verläßt. Die dann erhaltenen Ergebnisse sind nur noch bedingt aussagekräftig.

Das Problem läßt sich näherungsweise lösen, indem die Retry-Verzögerungszeit auf die Simulationsdauer normiert wird. Ein Zeitverhältnis nahe 1 heißt dann, daß die Einschwingzeit des Systems sicher nicht erreicht werden kann, weil die die Einschwingzeit bestimmende Retry-Verzögerungszeit so groß wie die gesamte Simulationsdauer ist. Ein Zeitverhältnis nahe 0 bedeutet, daß das System den eingeschwungenen Zustand erreicht hat. Zwischen 0 und 1 gibt es einen Wert, bei dem das zu simulierende technische System einen Phasenübergang vom eingeschwungenen zum nicht eingeschwungenen Zustand durchführt. Der Phasenübergang wird anhand des Verlaufs einer charakteristischen Größe des Systems deutlich: beim Übergang verändert sich die charakteristischen Größe von einer konstanten Geraden zu einer von der Simulationsdauer abhängigen Funktion. Als Indikator für den Phasenübergang ist z.B. der Durchsatz beim Empfängers geeignet. Eine Erhöhung der Retry-Verzögerungszeit kann solange vorgenommen werden kann, wie sich der Indikator nicht verändert.

Zweifellos ist die Wahl eines geeigneten Indikators kritisch, wie man im Falle des elementaren SCI-Rings an zwei anderen Größen sieht, die sich nicht dazu eignen würden. Diese Größen sind diejenigen Anteile am Durchsatz des Emp-

fängers, die ohne bzw. mit Paketwiederholungen erzielt werden (Retry Payload bzw. Non-Retry Payload). In Bild 7.4.4 sind diese Größen in Abhängigkeit von der Retry-Verzögerungszeit aufgetragen. Man sieht, daß beide sehr stark von

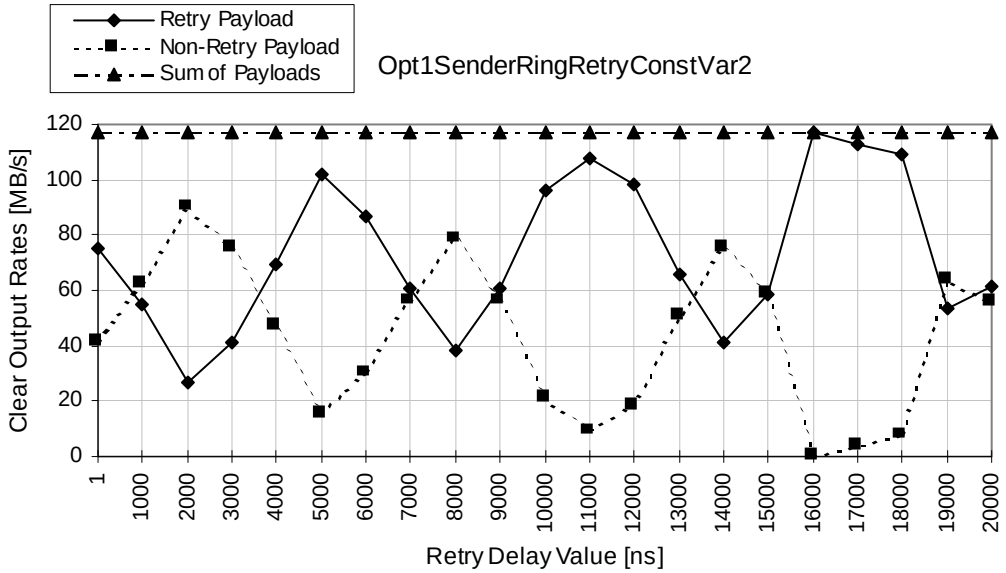

Bild 7.4.4: Andere Größen in Abhängigkeit von der Retry-Verzögerungszeit.

der jeweiligen Verzögerungszeit abhängen. Beachtlicherweise addieren sie sich trotz ihres oszillierenden Verlaufs zu einem konstanten Wert (Sum of Payloads). Dieser Wert entspricht genau dem in Bild 7.4.3 berechneten Empfängerdurchsatz (Total Output Payload). Der Empfängerdurchsatz hängt also nicht von der Verzögerungszeit ab und ist deshalb als Indikator geeignet. Warum die Anteile des Durchsatzes mit entgegengesetzter Phase oszillieren, ist unbekannt.

In Bild 7.4.5 sind der Nettodurchsatz beim Empfänger, die Paketverluste und der Retry-Verkehr in Abhängigkeit von der normierten Retry-Verzögerungszeit aufgetragen. Als Bezugsgröße für die Verzögerungszeit wurden 600 µs gewählt, das ist die Standardsimulationsdauer abzüglich der default-mäßigen Einschwingzeit. Der Sender hat eine konstante Datenrate von 200 MB/s, der Empfänger weist eine Request-Bearbeitungszeit von 400 ns auf.

Aus dem Diagramm ist ersichtlich, daß der Durchsatz bis zu einem Zeitverhältnis von ca. 0,25 konstant auf 116 MB/s bleibt, so daß bei der gegebenen Simulationsdauer Retry-Verzögerungszeiten bis ca. 150 µs als noch für die Simulation verträglich angesehen werden können. Ergebnisse, die man für Verzögerungszeiten größer als 0,25 Einheiten normierter Verzögerungszeit erhält, sind hingegen zu verwerfen. Bis zu dieser Grenze können mindestens vier Wiederholungen eines Pakets durchgeführt werden, was über den berechneten 3,3 Paketwiederholungen bei Retry-Verkehr liegt und dadurch die gemachten Überlegungen bestätigt.

In der Darstellung nach Bild 7.4.6 ist eine Ausschnittsvergrößerung des relevanten Bereichs von 0 bis 0,25 Einheiten normierter Verzögerunzeit dargestellt.

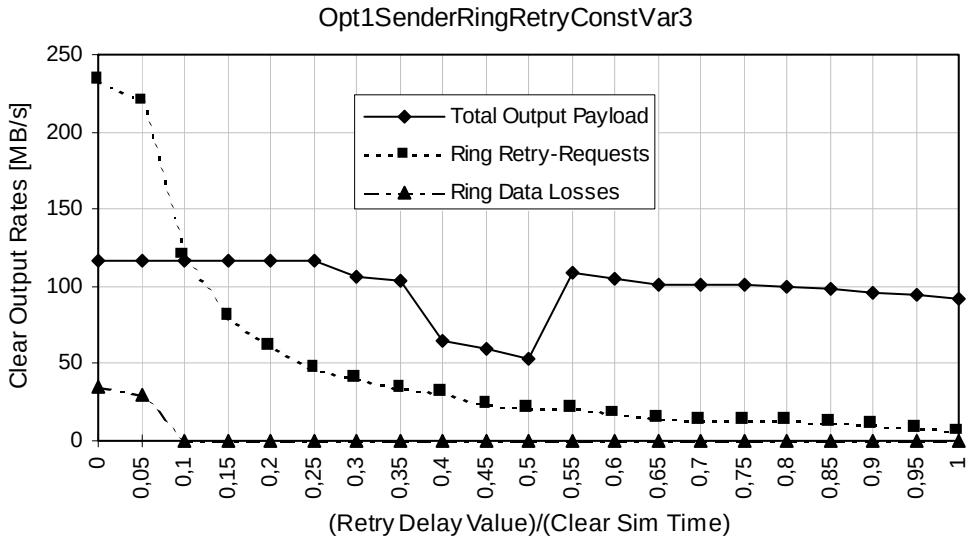

Bild 7.4.5: Leistungsanalyse bei Phasenübergang.

Deutlich ist der Rückgang des Retry-Verkehrs bei zunehmender Retry-Verzögerung sichtbar. Bedingt durch weniger Ringbelastung sinken auch die Paketverluste und erreichen bei 0,1 Einheiten Verzögerungszeit den Wert 0.

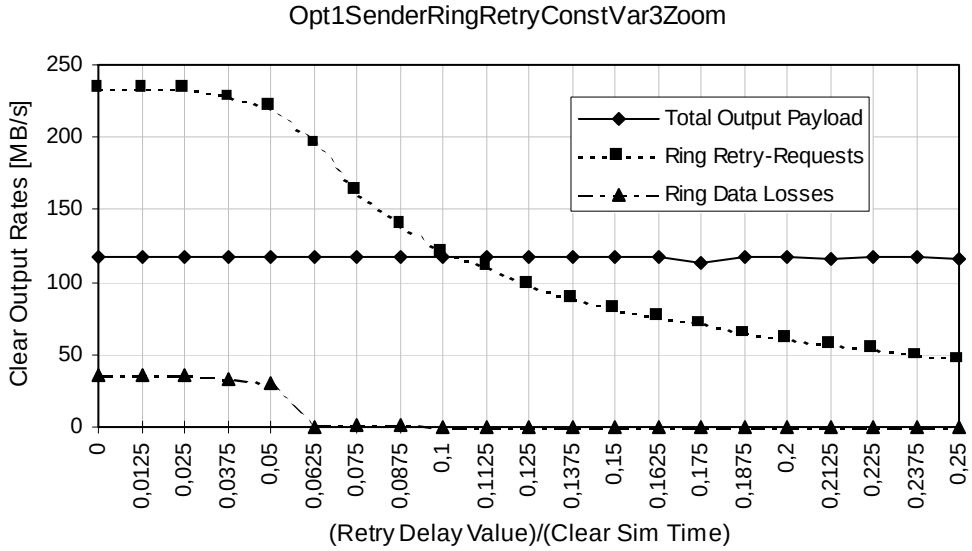

Bild 7.4.6: Durchsatz zwischen 0 und 0,25 Einheiten normierter Retry-Verzögerungszeit.

Zusammenfassend kann man sagen, daß die Reduzierung transienten Retry-Verkehrs, der durch eine vorübergehende Fehlanpassung von Sende- und Empfangsrate entsteht, durch eine Retry-Verzögerungszeit reduziert werden kann. Die Reduktion ist um so stärker, je größer die gewählte Verzögerungszeit ist.

Ein Blick auf die dabei entstehenden Latenzen zeigt, daß die Reduktion des

Retry-Verkehrs mit einer dramatischen Zunahme der Latenzzeit erkauft wird.

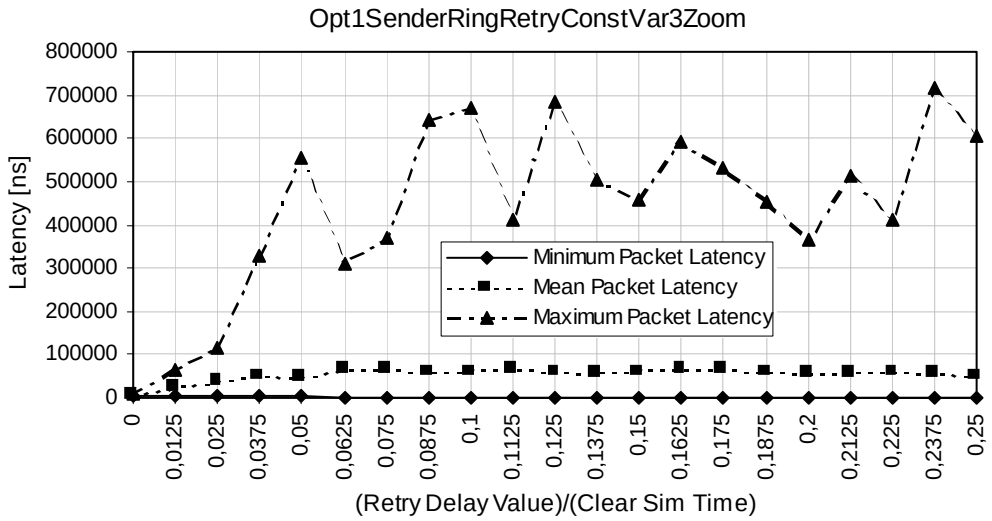

Bild 7.4.7: Latenz zwischen 0 und 0,25 Einheiten normierter Retry-Verzögerungszeit.

In Bild 7.4.7 ist Latenz im Bereich des relevanten Ausschnitts von 0 bis 0,25 Einheiten normierter Verzögerungszeit dargestellt. Aus den ursprünglichen maximal 8678 ns Latenz bei einem Retry Delay von 1 ns werden nun bis zu 718 µs (bei 0,2375 Einheiten normierter Verzögerungszeit). Dies entspricht einer Zunahme um 2 Zehnerpotenzen und ist für viele Anwendungen, wie z.B. der Echtzeitsteuerung eines Fusionsreaktorexperiments ein unakzeptabel hoher Wert.

7.4.2 Reduzierung der Latenz bei Retry-Verkehr

Die Frage ist, ob eine andere Strategie der Erhöhung der Retry-Verzögerungszeit zur Reduktion des Retry-Verkehrs bei einer moderaten Zunahme der Latenz führt. Man befindet sich dabei in einer Dilemma-Situation: einerseits sind ohne eine Begrenzung des Retry-Verkehrs erhebliche Leistungseinbußen zu erwarten, andererseits darf die Latenz auf nicht auf zu hohe Werte steigen.

Zur Untersuchung der Latenzzeitfrage sind in SCINET neben einer konstanten Retry-Verzögerungszeit zwei andere, adaptive Strategien vorgesehen, bei denen die Verzögerungszeit davon abhängt, die wievielte Wiederholung eines bestimmten Pakets durchgeführt wird. Die adaptiven Strategien berücksichtigen auf Paketbasis, wie oft ein bestimmtes Paket bereits abgewiesen worden ist. Beide Methoden unterschieden sich in der Geschwindigkeit mit der die Verzögerungszeit anwächst. Wählbar ist eine linear oder eine exponentiell ansteigende Verzögerung. Die adaptiv-lineare Strategie beruht darauf, daß die Zeit zwischen zwei nachfolgenden Paketwiederholungen von einem wählbaren Anfangswert startet, der mit jeder weiteren Wiederholung mit einem um eins

inkrementierten Faktor multipliziert wird, so daß sich die doppelte, dreifache, vierfache usw. Zeit ergibt. Bei der adaptiv-exponentiellen Strategie startet man ebenfalls von einem Anfangswert, der wird jedoch mit jeder weiteren Wiederholung jeweils verdoppelt.

In Bild 7.4.8 ist das Verhalten der Latenz bei den drei verschiedenen Strategien in Abhängigkeit von dem gewählten Anfangswert dargestellt. Dabei fallen

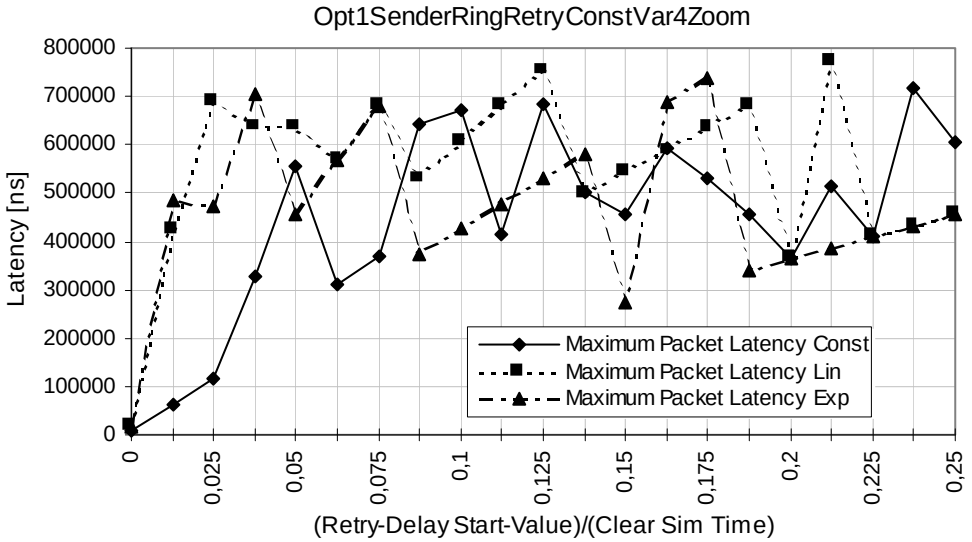

Bild 7.4.8: Anstieg der Latenz bei verschiedenen Startwerten und Strategien.

als erstes die hohen Fluktuationen auf, die bei Werten oberhalb von ca. 0,0375 Einheiten normierter Verzögerungszeit auftreten. Dies führt noch einmal das indeterministische Verhalten der Latenz deutlich vor Augen. Unterhalb von 0,05 schneidet die konstante Strategie bzgl. der Latenz am besten ab, darüber besteht nur wenig Unterschied zwischen den einzelnen Strategien. Daraus könnte man schließen, daß eine konstante Verzögerungszeit die beste Methode darstellt. Daß dem nicht so ist, wird klar, wenn man sich das „Dämpfungsverhalten“ der konstanten Strategie beim Retry-Verkehr anschaut (Bild 7.4.9). Hier schneidet die exponentielle Strategie am besten und die konstante Strategie am schlechtesten ab. Das folgende Zahlenbeispiel zeigt, daß aufgrund dieser Tatsache die exponentielle Strategie nicht nur hinsichtlich der Retry-Dämpfung, sondern auch bzgl. der Latenz die insgesamt beste Methode ist.

Bei der exponentiellen Strategie sinkt der Retry-Verkehr von beispielsweise 234 MB/s auf 103 MB/s bei 0,05 Einheiten normierter Verzögerungszeit ab (Bild 7.4.9). Dabei hat man eine Latenz von 456 µs. Derselbe Retry-Wert wird von der konstanten Strategie erst bei 0,125 Einheiten erreicht, was in der höheren Latenz von 683 µs resultiert. Das bedeutet, daß ein vorgegebenen Faktor der Reduktion des Retry-Verkehrs von der exponentiellen Strategie bereits für rel. kleine Anfangswerte der Verzögerungszeit erreicht wird. Da bei kleinen Anfangswerten auch die Latenz am kleinsten ist, schneidet die exponentielle Strategie insgesamt am besten ab.

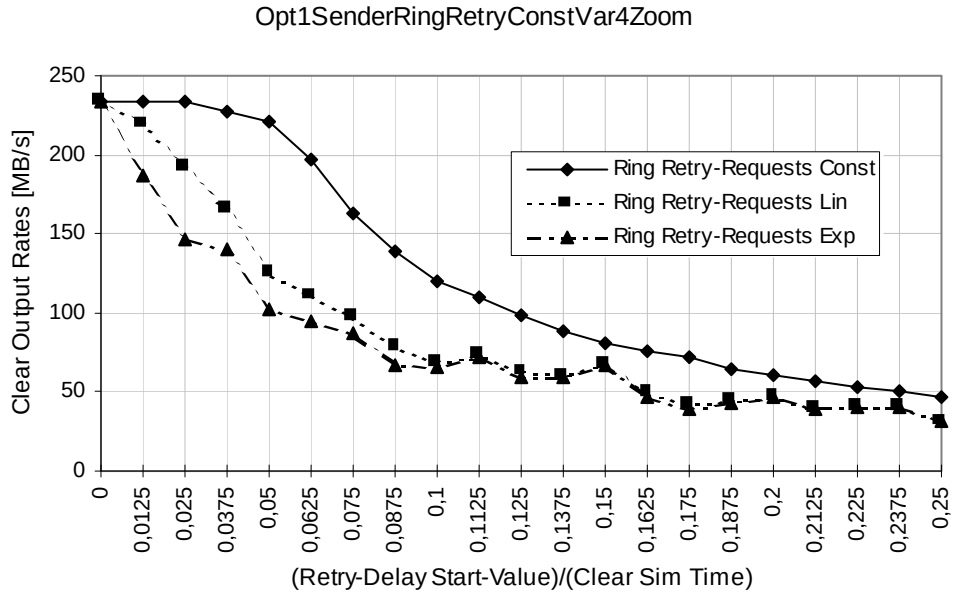

Bild 7.4.9: Reduktion des Retry-Verkehr bei verschiedenen Strategien.

Ergebnis:

Zusammenfassend kann gesagt werden, daß bei einem elementaren SCI-Ring durch Paketwiederholungen ein Großteil der Ringbandbreite (233,7 MB/s) verbraucht wird. Besteht ein anhaltendes Ungleichgewicht zwischen der Rate mit der Paket erzeugt werden im Vergleich zur Verbrauchsrate in der Art, daß zuviele Pakete produziert werden, dann kann der Überschuß nicht durch Einführung einer Verzögerungszeit zwischen zwei Paketwiederholungen vermindert werden. Bei einer transienten Überschußproduktion hingegen bewirkt eine Retry-Verzögerung eine erhebliche Reduktion der vom Retry-Verkehr okkupierten Bandbreite (von 234 MB/s auf 31,7 MB/s bei 0,25 Einheiten normierter Verzögerungszeit). Die Verbesserung wird durch einen starken Anstieg der Latenz erkauft. Der Anstieg in der Latenz fällt bei einer konstanten Verzögerungszeit am kleinsten aus (von ursprünglich 8678 ns auf 718 µs bei 0,2375 Einheiten normierter Verzögerungszeit). Hingegen ist zur Reduktion des Retry-Verkehrs eine adaptive Strategie am besten geeignet, bei der die Zeit zwischen zwei Wiederholungen desselben Pakets exponentiell mit der Zahl der abgelehnten Pakete zunimmt. Hier reichen schon kleine Anfangswerte der Verzögerung aus, um den Retry-Verkehr wirkungsvoll zu reduzieren. Das bedeutet, daß ein vorgegebener Faktor der Reduktion des Retry-Verkehrs von der exponentiellen Strategie bereits für rel. kleine Anfangswerte der Verzögerungszeit erreicht wird. Aufgrund dieser niedrigeren Anfangswerte hat die exponentielle Strategie insgesamt auch eine kleinere Latenz als Methoden mit konstanter oder linear ansteigender Verzögerungszeit. Sie ist deshalb zu bevorzugen.

7.5 Leistungsanalyse bei multiplen Sendern

In der Praxis wird man oft Systeme haben, die nicht aus elementaren SCI-Ringen aufgebaut sind. Beispielsweise bietet es sich bei Datenerfassungen an, mehrere Sensoren wegen der hohen Übertragungsrate von SCI in einem Ring zusammenzuschalten und die zur Verfügung stehende Ringbandbreite aufzuteilen. Die nachfolgende Leistungsanalyse untersucht dabei, wie stark sich die einzelnen Sensoren, d.h. Sender gegenseitig beeinflussen.

Es wird der einfachste Fall von zwei Sendern S1, S2 und zwei Empfängern E1, E2 in einem Ringe betrachtet, bei dem stationär S1 mit E1 und S2 mit E2 kommuniziert. Weiterhin soll S1 exemplarisch eine konstant niedrige Datenrate von 10 MB/s aufweisen, während die Rate von S2 von Null bis zum Ringlimit von 500 MB/s variiert wird. Bild 7.5.1 zeigt die Konfiguration von S1 und E1 bzw. S2 und E2, die untersucht wird (S1 ist der Sender mit der konstanten Datenrate).

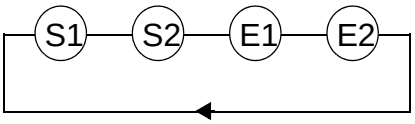

Bild 7.5.1: Beispiel für multiple Sender (S1 und S2) in einem SCI-Ring.

Die Frage ist, welchen Durchsatz der zum zweiten Sender gehörende Empfänger (E2) erzielen kann, während gleichzeitig eine andere Kommunikation (S1->E1) auf dem Ring stattfindet. Die Geschwindigkeit, mit denen beide Empfänger Pakete aufnehmen können, wird zunächst so hoch gewählt, daß kein Retry-Verkehr auftreten kann. Die Konfiguration zeigt dann das in Bild 7.5.2 dargestellte Verhalten. Der Durchsatz von E2 steigt linear bis auf den Spitzenwert von 267 MB/s an, der bei 350 MB/s Eingangsdatenrate erreicht wird, und fällt dann auf einen konstanten Sättigungswert von 229 MB/s ab. Ab 350 MB/s Eingangsrate treten Paketverluste auf. Zum Vergleich: beim elementaren SCI-Ring wurde bei 450 MB/s ein Durchsatz von 333 MB/s erreicht. 267 MB/s bedeuten eine Abnahme um 66 MB/s bzw. um 20%.

Das bedeutet, daß durch die Anwesenheit des ersten Senders (S1) der Durchsatz der nebenläufigen Kommunikation (S2->E2) signifikant reduziert wird, obwohl S1 mit 10 MB/s nur wenig Bandbreite benötigt. Der Grund dafür ist, neben erhöhtem Aufwand bei den Bandbreiteallozierungsprotokollen, daß sich die Umlaufzeit im Ring durch die Bypass-Fifos von S2 und E2 und deren Leitungslängen um 2*(48+2)ns = 100 ns erhöht hat. Das ist eine Wert, der bereits in derselben Größenordnung liegt wie Zeit, die nötig ist, ein Paket über eine SCI-Link auszusenden (168 ns).

Die Latenzen verhalten sich bis zur Erreichen der Sättigung mit 2147 ns im wesentlichen deterministisch. Die Latenz ist gegenüber der Einzelkommunika-

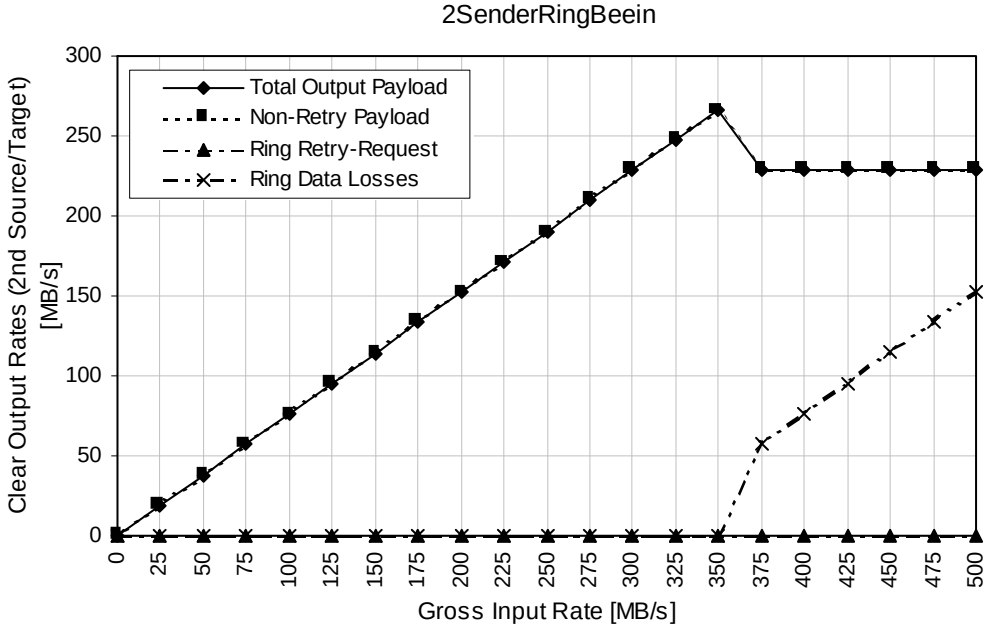

Bild 7.5.2: Durchsatz bei zwei Sendern ohne Retry-Verkehr.

tion auf einem elementaren SCI-Ring um den Faktor 2,3 erhöht, was einen überproportionalen Zuwachs darstellt. Ab der Sättigung fächert die Latenzzeit ebenso wie beim elementaren SCI-Ring auf und kann zwischen 1597 (Minimalwert) und 6103 ns (Maximalwert) schwanken, was wiederum dem 2,3-fachen des elementaren Rings entspricht.

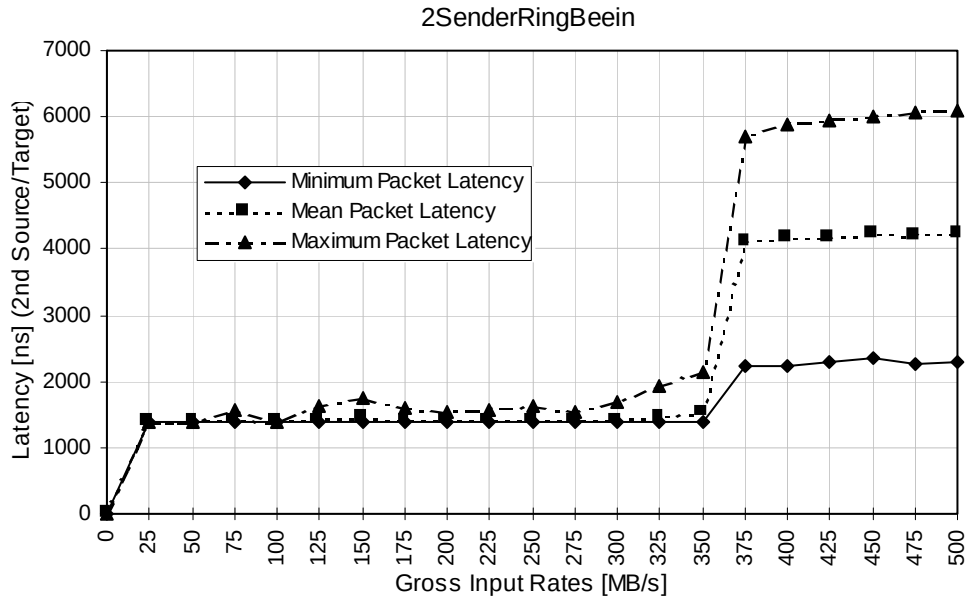

Bild 7.5.3: Latenz bei zwei Sendern ohne Retry-Verkehr.

Ergebnis:

Ein SCI-Ring bestehend aus zwei Sendern und zwei Empfängern weist pro Sender/Empfängerpaar einen überproportional niedrigeren Durchsatz sowie eine überproportional höhere Latenzzeit im Vergleich mit den Einzelkommunikationen der entsprechenden elementaren SCI-Ringe auf.

7.5.1 Verhalten bei Retry-Verkehr

Im zweiten Teil der Analyse der wechselseitigen Beeinflussung zweier gleichzeitiger Kommunikationen auf einem SCI-Ring wird das Verhalten bei Retry-Verkehr untersucht. Die Retry-Pakete werden dadurch provoziert, daß der erste Empfänger (E1) mit der sehr hohen Anforderungsbearbeitungszeit (RequestDelay) von 40000 ns beaufschlagt wird. Praktisch alle von S1 abgeschickten Pakete erfahren so vielfache Wiederholungen.

Erwartungsgemäß sinkt der Durchsatz der zweiten nebenläufigen Kommunikation (S2->E2) ab, wie Bild 7.5.4 zeigt, und der Retry-Verkehr der ersten Kommunikation okkupiert den größten Teil der Ringbandbreite. Überraschend

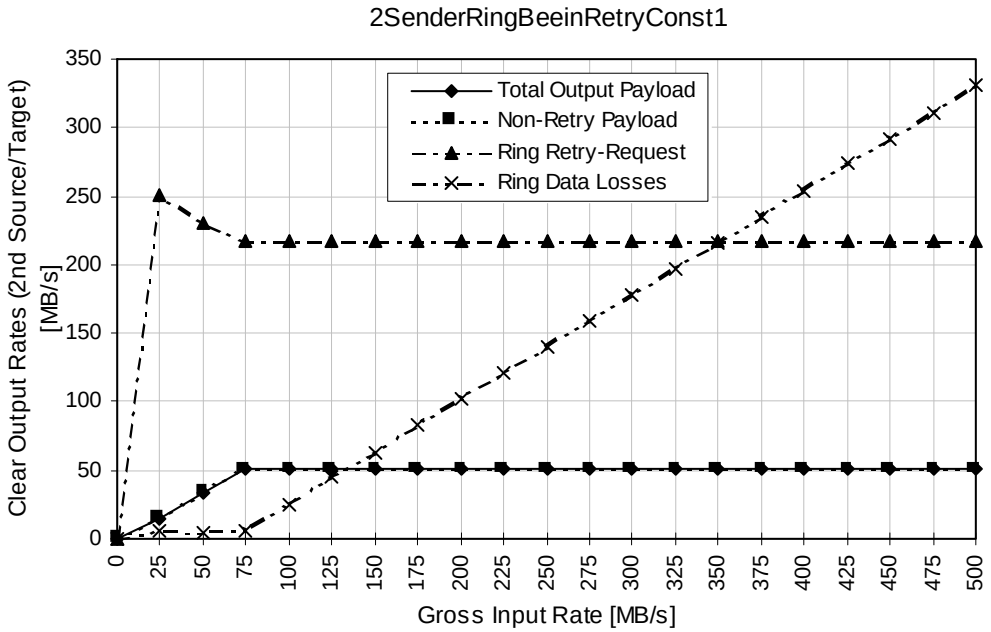

Bild 7.5.4: Leistungsabfall am zweiten Empfänger bei zwei Sendern mit Retry-Verkehr.

ist jedoch, daß der Nutzverkehr auf den sehr niedrigen Wert von 51 MB/s abfällt, die bei 75 MB/s S2-Eingangsdatenrate erreicht werden (von zuvor 267 MB/s bei 350 MB/s Datenrate), was einen Rückgang auf 19% bedeutet.

Ähnlich ungünstig verhält sich die Latenz der Kommunikation von S2->E2 unter Beeinflussung durch Retry-Verkehr von S1->E1. Von einem lokalen

Spitzenwert von 42 µs bei 50 MB/s Eingangsdatenrate abgesehen, hat man oberhalb des Sättigungspunkts als Maximalwert 8979 ns. Dies entspricht einer Zunahme gegenüber dem Ein-Sender-Ring um eine Größenordnung. Erfreulicherweise treten bei hohen Datenraten keine Fluktuationen in der Latenz auf. Stark nachteilig ist hingegen, daß jetzt die Latenz auch unterhalb des Sättigungspunkt von 75 MB/s indeterministisch ist.

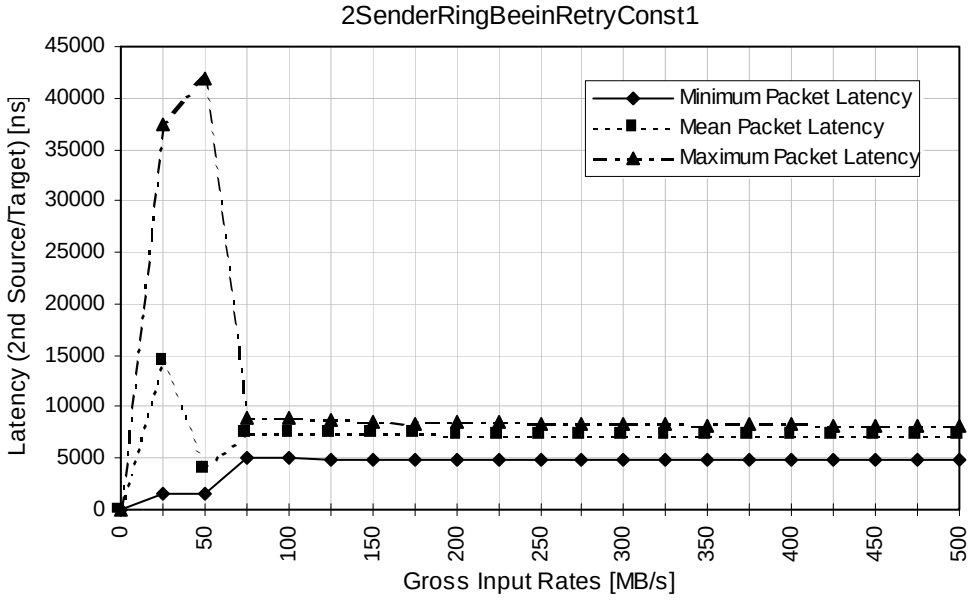

Bild 7.5.5: Latenzanstieg am zweiten Empfänger bei zwei Sendern mit Retry-Verkehr.

Ergebnis:

Sind in einem SCI-Ring zwei Sender und zwei Empfänger angeschlossen, reduziert ein erster langsamer Empfänger erheblich den Durchsatz der zweiten Kommunikation, obwohl beide nichts miteinander zu tun haben (Durchsatzabfall auf 19%). Gleichzeitig steigt die Latenz der zweiten Kommunikation um 1 Größenordnung an und wird bei allen Eingangsdatenraten ihres Sender indeterministisch. Der Grund dafür liegt im Retry-Verkehr zum ersten, langsamen Empfänger, der den Ring belastet.

Aufgrund des Leistungsabfalls in der Bandbreite und der indeterministischen Latenz im Zwei-Sender-Ring, wird klar, daß man bei Echtzeitanwendungen, die über SCI realisiert sind, vorsichtig hinsichtlich der Überlastung eines Empfängers sein muß, denn es ist schwierig, garantierte Zeiten einzuhalten, wenn die Leistungsdaten einer Kommunikation von der Paketannahmegeschwindigkeit anderer, unbeteiligter Empfänger abhängen, also indeterministisch sind.

Es verbleibt zu klären, ob zumindest der Bandbreiteabfall bei transientem Retry-Verkehr durch Einfügung einer Retry-Verzögerungszeit abgemildert oder evtl. aufgehoben werden kann. Stationärer Retry-Verkehr kann, wie bereits im

vorigen Kapitel erläutert, aus prinzipiellen Gründen nicht von einer Retry-Verzögerungszeit beeinflußt werden.

7.6 Durchsatzerhöhung im Zwei-Sender-Ring

Bereits die Analyse des elementaren SCI-Rings hat ergeben, daß transienter Retry-Verkehr durch die Einführung einer Retry-Verzögerungszeit reduziert werden kann. Die besten Ergebnisse aller drei Strategien zeigte dabei die exponentiell-adaptive Methode. Jetzt soll untersucht werden, wie sich die Strategien von konstanter bzw. linear oder exponentiell ansteigender Verzögerungszeit auf den Durchsatz im Zwei-Sender-Ring auswirken, sobald vorübergehender Retry-Verkehr auftritt.

Da im folgenden transiente Vorgänge simuliert werden sollen, muß die Verzögerungszeit auf die Simulationsdauer normiert werden und darf ein bestimmtes Maß nicht überschreiten (maximal 0,25 Einheiten normierter Verzögerungszeit). Die übrigen Randbedingungen für die Simulationen sind: S1 sendet mit 10 MB/s auf den sehr langsamen Empfänger E1, der ein Request Delay von 40 µs hat, um dadurch Retry-Verkehr zu provozieren. Die zweite, nebenläufige Kommunikation findet zwischen S2, der mit 400 MB/s sendet, und E2 statt, der mit 40 ns Request Delay in jedem Fall schneller als sein Sender ist und keiner Retry-Pakete bedarf.
In Bild 7.6.1 ist das Ergebnis der Untersuchungen dargestellt. Hieraus wird er-

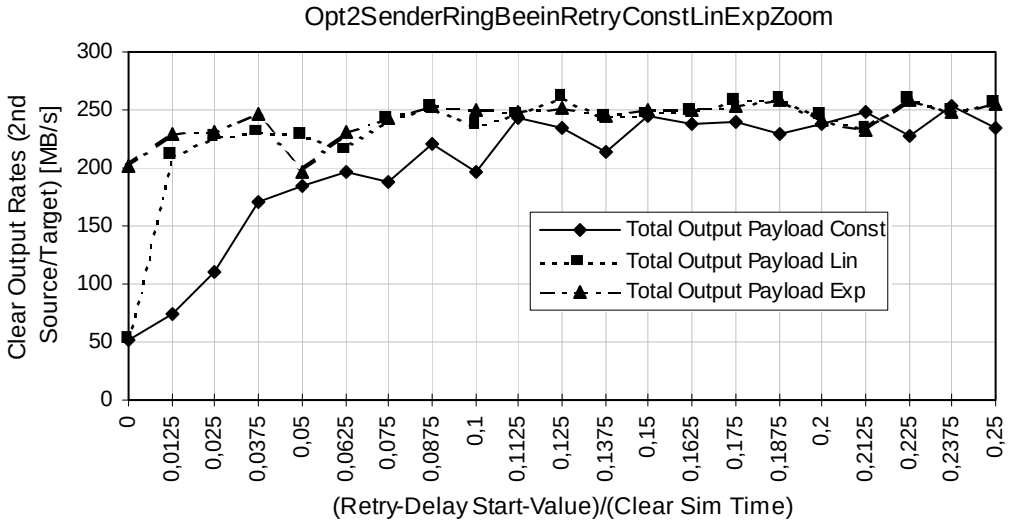

Bild 7.6.1: Durchsatz bei verschiedenen Anfangswerten der Verzögerungszeit.

sichtlich, daß ebenso wie beim Ein-Sender-Ring die exponentielle Strategie am

besten abschneidet. Bereits für sehr kleine Anfangswerte der Verzögerungszeit wird ein großer Durchsatz von 201 MB/s erzielt. Beispielsweise reichen 100 ns Anfangsverzögerung aus, um den Retry-Verkehr wirkungsvoll zu reduzieren. Der kleine Anfangswert wirkt sich entsprechend günstig auf die Latenz aus. Bei 100 ns ergeben sich 16083 ns maximale Latenz, was nur das Doppelte dessen ohne Retry-Dämpfung darstellt. Der Vergleich mit der Strategie der konstanten Verzögerungszeit zeigt, wie wirkungsvoll die exponentielle Methode ist: für ungefähr den gleichen Durchsatz (196 MB/s) braucht man bei konstanter Verzögerungszeit bereits 0,1 Einheiten normierter Verzögerungszeit, was in 155 µs Latenz resultiert, also dem Vielfachen der exponentiellen Strategie.

In Bild 7.6.2 ist gezeigt, wie sich der Anfangswert der Retry-Verzögerungszeit auf die Latenz der Datenübertragung auswirkt. Im wesentlichen ist bei allen

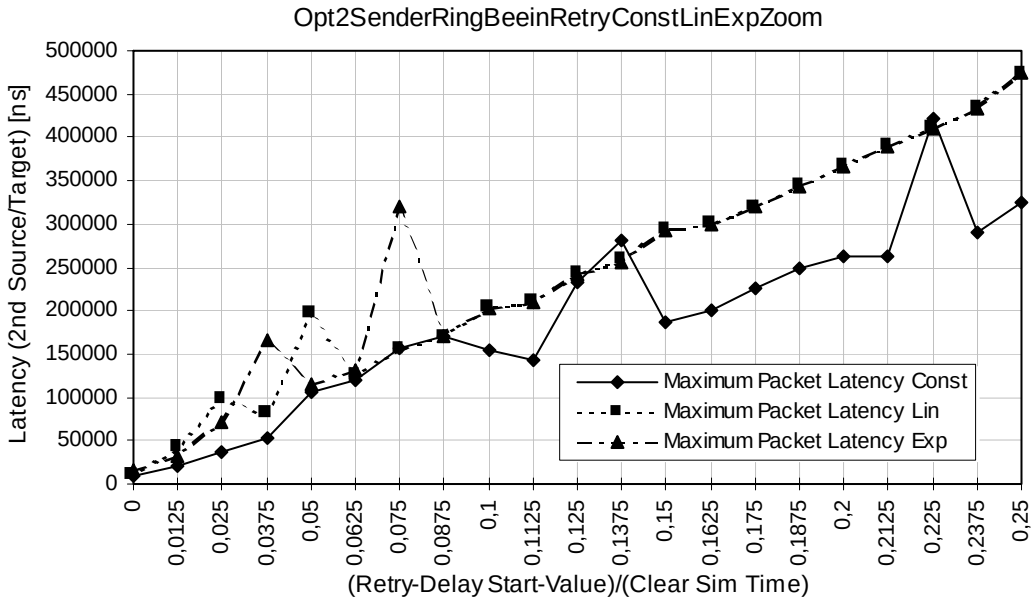

Bild 7.6.2: Latenz bei verschiedenen Anfangswerten der Verzögerungszeit.

drei Strategien ein ungefähr proportionaler Zusammenhang erkennbar, wobei die konstante Methode am besten abschneidet. Ab 0,0875 normierte Zeiteinheiten fallen die Latenzen von linearer und exponentieller Strategie zusammen. Sobald der Anfangswert einen nenneswerten Bruchteil von z.B. 0,05 der Gesamtsimulationsdauer erreicht, ist auch bei konstanter Strategie eine sehr hohe Latenz von ca. 100 µs zu verzeichnen. Um die Latenz auf beispielsweise 40 µs zu begrenzen, sollte ein Anfangswert kleiner als 0,0125 Einheiten normierter Verzögerungszeit gewählt werden.

In Bild 7.6.3 ist die komplette Leistungsanalyse des Zwei-Sender-Rings bei 100 ns Anfangsverzögerung und exponentieller Strategie gezeigt. Der Durchsatz des zweiten Empfängers erreicht 202 MB/s bei 500 MB/s Eingangsdatenrate des zweiten Sender. Der Durchsatz hat sich von ursprünglich 19% des Wertes beim Zwei-Sender-Ring ohne Retry-Verkehr auf jetzt 76% erhöht. Der

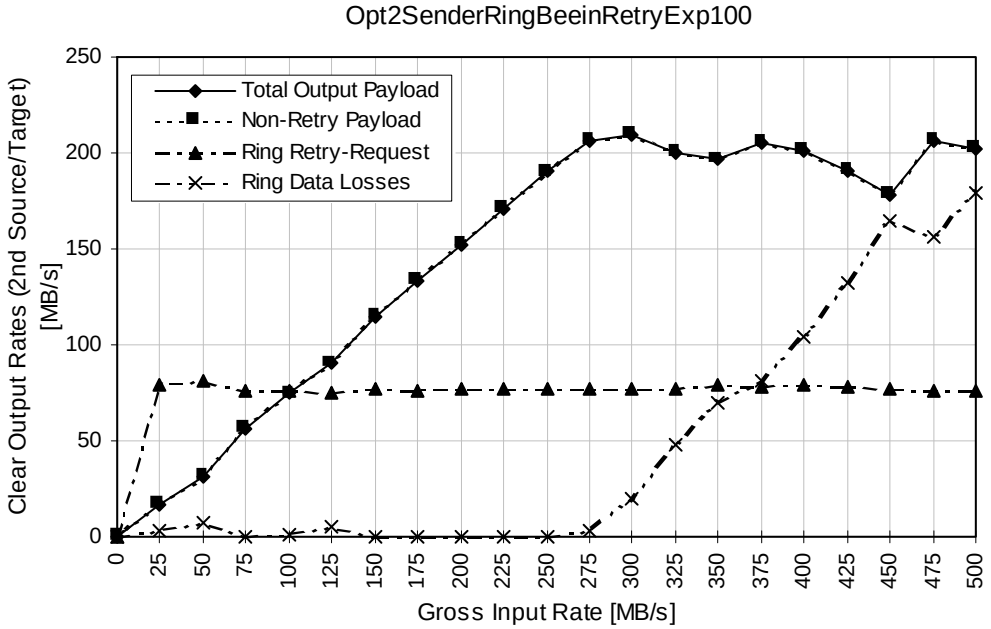

Bild 7.6.3: Durchsatz im Zwei-Sender-Ring bei exponentieller Strategie und 100 ns Anfangswert der Retry-Verzögerung.

Retry-Verkehr sinkt von 217 MB/s auf 76 MB/s, was einer Abnahme um den Faktor 2,9 entspricht. Die Paketverluste sinken auf 179 MB/s, und der Sättigungspunkt verschiebt sich von 75 MB/s auf 275 MB/s.

Die Latenz ist für keine Eingangsrate deterministisch, vielmehr schwankt sie um eine Zehnerpotenzen zwischen 1405 ns Minimalwert und 32 µs Maximalwert. Insgesamt kann man sagen, daß die exponentielle Strategie bzgl. der gewünschten Durchsatzerhöhung erfolgreich ist, insbesondere auch im Hinblick auf einen moderaten Anstieg der Latenz.

Ergebnis:

Im Zwei-Sender-Ring kann der Durchsatz einer Kommunikation trotz Belastung des Rings aufgrund von Retry-Paketen einer anderen, nebenläufigen Kommunikation auf 76% des Wertes ohne transienten Retry-Verkehr stabilisiert werden. Die Voraussetzung dafür ist, daß eine exponentiell ansteigende Verzögerungszeit zwischen den Wiederholungen eines Pakets eingeführt wird. Diese Strategie schneidet sowohl bzgl. der Retry-Dämpfung als auch im Hinblick auf eine Latenzerhöhung am besten im Vergleich zu einer konstanten oder linear ansteigenden Verzögerungszeit ab. Ein sehr kleiner Anfangswert der Verzögerungszeit (z.B. 100 ns) reicht bei der exponentiellen Strategie aus, um den Retry-Verkehr wirkungsvoll zu reduzieren. Die Latenz ist dabei indeterministisch und schwankt zwischen 2 und 32 µs (bei 100 ns Anfangswert).

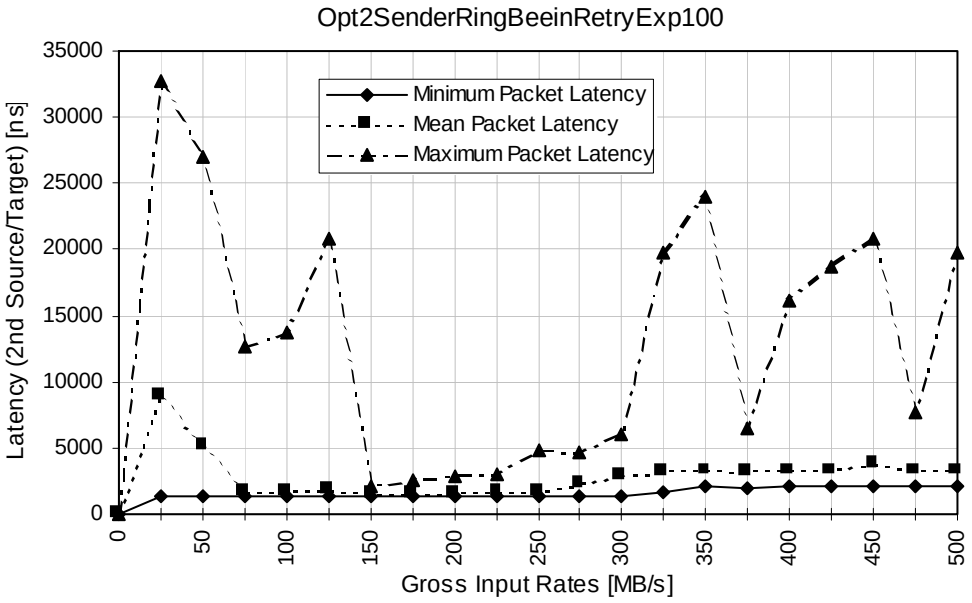

Bild 7.6.4: Latenz im Zwei-Sender-Ring bei exponentieller Strategie und 100 ns Anfangswert der Retry-Verzögerung.

8 Analyse von SCI-Schaltern

8.1 Einleitung

In einem SCI-Ring kann man Knoten unterschiedlichster Art, wie Rechner, Prozessoren, Speicher oder Peripherie zusammenschalten. Größere SCI-Systeme ab ca. 4-8 Knoten lassen sich jedoch aus Bandbreitegründen und wegen der Gefahr zunehmender Paketwiederholungen nicht mehr über einen einzelnen SCI-Ring koppeln, statt dessen müssen mehrere Ringe verwendet werden, die mit Hilfe von speziellen Schaltern (Switches), Brücken (Bridges) oder Einwählern (Routern) verbunden sind. Durch die Kopplung der Ringe entstehen Topologien, die entweder in der Kategorie der statischen oder der dynamischen Netze eingeteilt werden können.

Kommerziell sind bislang nur SCI-Switches erhältlich [Dolphin94b], die allerdings bei geeigneten Adressierungsverfahren auch als Brücken und Einwähler verwendet werden können. Intern bestehen diese Schalter aus 4 LC-I oder LC-II Link-Controller-Bausteinen, die über ihre B-Link-Busse gekoppelt sind. In Bild 8.1.1 sind zwei funktional gleichwertige Blockdiagrammdarstellungen eines SCI-Schalters gezeigt. In der ersten Darstellung sind die Link-Ein-/ und Ausgänge auf derselben Schalterseite gezeichnet, während sie in der zweiten Darstellung auf verschiedenen Seiten zu sehen sind. Das erste Blockdiagramm

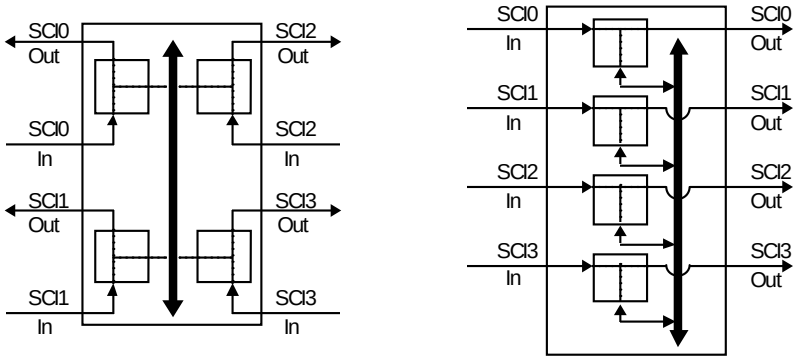

Bild 8.1.1: Zwei gleichwertige Blockdiagrammdarstellungen eines SCI-Schalters.

eignet sich gut für Fälle, bei denen der Schalter über kleine SCI-Ringe, sog. Ringlets angeschlossen ist, während das zweite besser für Repräsentationen von Schaltern geeignet ist, die über lange, durchgängige Ringe mit ihren Sendern und Empfängern verbunden sind.

Die Kerndaten eines LC-II-basierten Schalters sind 500 MB/s externe Datenrate pro Schalteranschluß (Port) und 600 MB/s interne B-Link-Geschwindigkeit. Die Bandbreiten gelten für Lesen und für Schreiben, wobei beachtet werden muß, daß Ports im Voll-Duplex-Modus arbeiten können, während das B-Link aufgrund seines Bus-Charakters nur zu Halb-Duplex fähig ist. Zwischen allen vier Ports eines Schalters können Daten bidirektional ausgetauscht werden, jedoch darf die Gesamtmenge der gleichzeitig gelesenen und geschriebenen Daten die B-Link-Bandbreite nicht überschreiten. So kann ein beliebiger Schalteranschluß mit 500 MB/s von einem anderen Port lesen oder schreiben, vorausgesetzt, daß die anderen Anschlüsse währenddessen zusammen nicht mehr als 100 MB/s an Verkehr erzeugen. Real liegen die Werte niedriger, weil beispielsweise ein NWRITE64-Request-Paket mit 64-Byte Nutzdaten als 88 Byte-Paket verpackt auf dem B-Link transportiert werden muß. Wenn je zwei Ports simultan im Vollduplex-Modus Daten austauschen, bleiben pro Port und Richtung weniger als 150 MB/s an Bandbreite übrig.

8.2 Vier-Port-Schalter mit Ringlet-Anschluß

Die konventionelle Einsatzweise eines Vier-Port-Schalters ist in Bild 8.2.1 dargestellt [Kristians94][Wu94b]: zwei SCI-Knoten P0 und P1, die im Einzelfall Prozessoren, Rechner oder Sensoren in einem Datenerfassungssystem sein können, erzeugen Anforderungspakete, die über kleine SCI-Ringe, sog. Ringlets (R0-R3) in den Schalter eingespeist und von dort von zwei anderen Knoten (M0 und M1), die beispielsweise Speicher oder andere Rechner sein können, aufgenommen werden. Im folgenden werden die Leistungsdaten der Konstellation

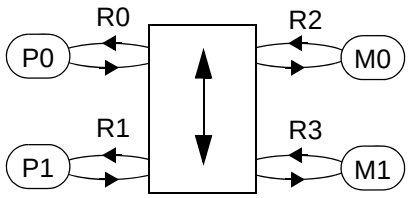

Bild 8.2.1: Konventionelle Einsatzweise eines Vier-Port-Schalters.

nach Bild 8.2.1 untersucht. Dazu werden die Sender P0 und P1 mit ansteigenden Eingangsraten von 0 bis 500 MB/s beaufschlagt und die über alle Sender, Empfänger und Ringe summierten Größen von Durchsatz, Paketverluste, Retry-Verkehr und Latenz analysiert. Aufgrund des B-Link-Engpasses ist ab einer bestimmten Eingangsrate eine Sättigung des Durchsatzes zu erwarten, mit den bekannten Konsequenzen für den Retry-Verkehr und die Paketverluste.

Die Simulationen, die in Bild 8.2.2 dargestellt sind, bestätigen diese Überlegungen. Nach einem streng linearen Anstieg, geht der Durchsatz in eine waagrechte Gerade über, die die Sättigung des Schalters wiederspiegelt. Im Leistungsanalysediagramm sieht man, daß der Sättigungspunkt bei ca. 250 MB/s summierter Eingangsdatenrate liegt. Von da an steigen die Paketverluste linear an, während der summierte Retry-Verkehr auf den ihm von den Bandbreiteallozierungsprotokollen vorgegeben Maximalwert springt.

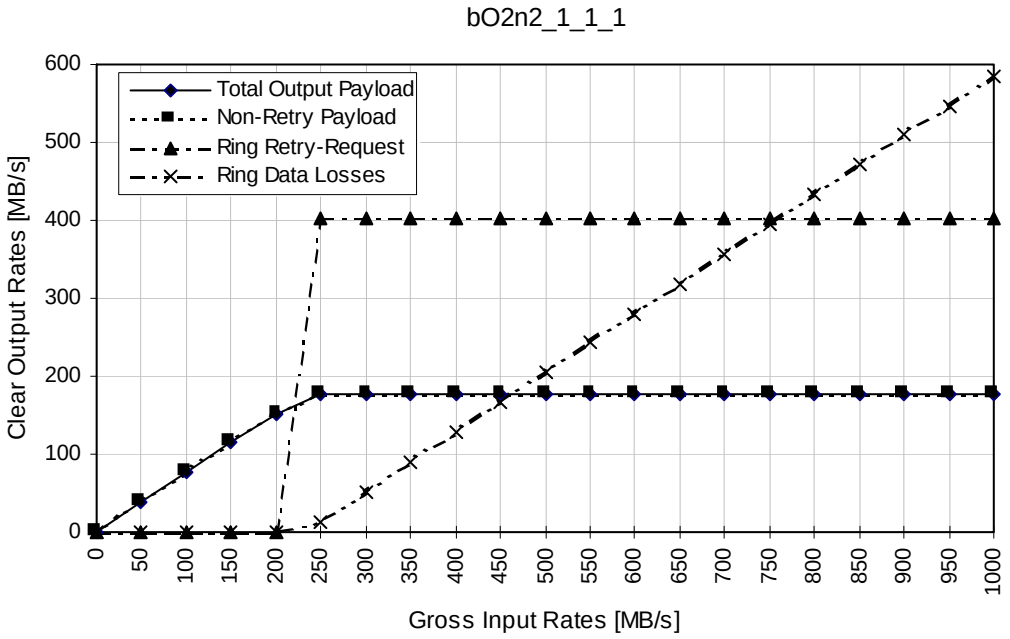

Bild 8.2.2: Leistungsanalyse der Ringlet-Schalterkopplung nach Bild 8.2.1.

Der größte Wert des Nettodurchsatzes von M0 und M1 zusammengenommen beträgt 176 MB/s. Zusammen mit den Paketverlusten von 585 MB/s erhält man

daraus 761 MB/s, was sich unter Einbeziehung des Verpackungs-Overheads von 84/64 in 999 MB/s empfangenen Bruttoverkehrs umrechnen läßt. Der vom Simulator berechnete Wert des Durchsatzes ist präzise, denn von den Sendern wurden brutto 1000 MB/s erzeugt.

Aufgrund dieser Ergebnisse kann man sagen, daß der Durchsatz des kommerziell erhältlichen, LC2-basierten, SCI-Schalters mit nur 176 MB/s enttäuschend niedrig ist. Darüber hilft auch nicht der Umstand hinweg, daß Sender und Empfänger 20 m entfernt aufgestellt sind, wodurch der Durchsatz wegen des Vier-Phasen-Protokolls von SCI reduziert wird. (Die räumliche Entfernung soll ungefähr die Verhältnisse bei einem Datenerfassungssystem wiederspiegeln.)

Der Grund für den geringen Durchsatz beim Vier-Port-Ringlet-Schalter liegt nicht allein in der reinen B-Link-Bandbreite. Schließlich wird die schalterinterne Transferkapazität bei 176 MB/s Durchsatz nicht ausgeschöpft, selbst dann nicht, wenn man die Datenmenge der Response-Pakete und den erhöhten Verpackungsaufwand beim B-Link-Transfer berücksichtigt. Ein Blick auf den Retry-Verkehr gibt einen Hinweis auf den zweiten Grund für die geringe Leistung des Schalters: jedes Sender-Ringlet hat zusätzlich zu den reinen Nutzdaten 202 MB/s an Paketwiederholungen zu übertragen (404 MB/s in der Summe). Der Retry-Verkehr entsteht in der Zeit, die ein Empfangsknoten braucht, bis er den Zugriff auf das B-Link bekommt. Die Zugriffszeit wiederum wird bestimmt von der Zahl der Ports pro Schalter, von der B-Link-Arbitrierungs- und Setup-Zeit sowie von der Zeit pro Datentransfer. Diese vier Faktoren zusammen verursachen den Leistungsverlust.

Die Endpunkt-zu-Endpunkt-Latenzen sind, wie Bild 8.2.3 zeigt, bis zur Sättigungsgrenze deterministisch und liegen mit 2344 ns rel. niedrig, insbesondere wenn man bedenkt, daß darin zwei Ringlatenzen enthalten sind. Das bedeutet, daß die reine B-Link-Transferzeit im Schalter mit (2344-2*915) ns = 514 ns zu Buche schlägt. (Die Latenz eines nicht-gesättigten, elementaren SCI-Rings beträgt 915 ns). Ab ca. 200 MB/s Eingangsdatenrate werden die Latenzen ähnlich wie beim elementaren SCI-Ring mit Retry-Verkehr indeterministisch und schwanken zwischen 7362 (Minimum) und 12797 ns (Maximum). Im Vergleich zu diesem haben sich die Zeiten um ca. 50% erhöht.

Ergebnis:

Ein kommerzieller Vier-Port-SCI-Schalter hat in Ringlet-Konfiguration mit 176 MB/s einen rel. niedrigen Durchsatz, der nur ca. 1/4 des maximal möglichen Durchsatzes beträgt. Eine Kaskadierung mehrerer Schalter ist wegen des geringen Durchsatzes mit hohen Paketverlusten verbunden. Die Sättigung des Durchsatzes wird bei 250 MB/s Bruttoeingangsrate erreicht. Der Schalter weist bis etwa 200 MB/s Eingangsdatenrate eine deterministische Latenz von 1960 ns auf, darüber schwankt sie zwischen 7362 (Minimum) und 12797 ns (Maximum).

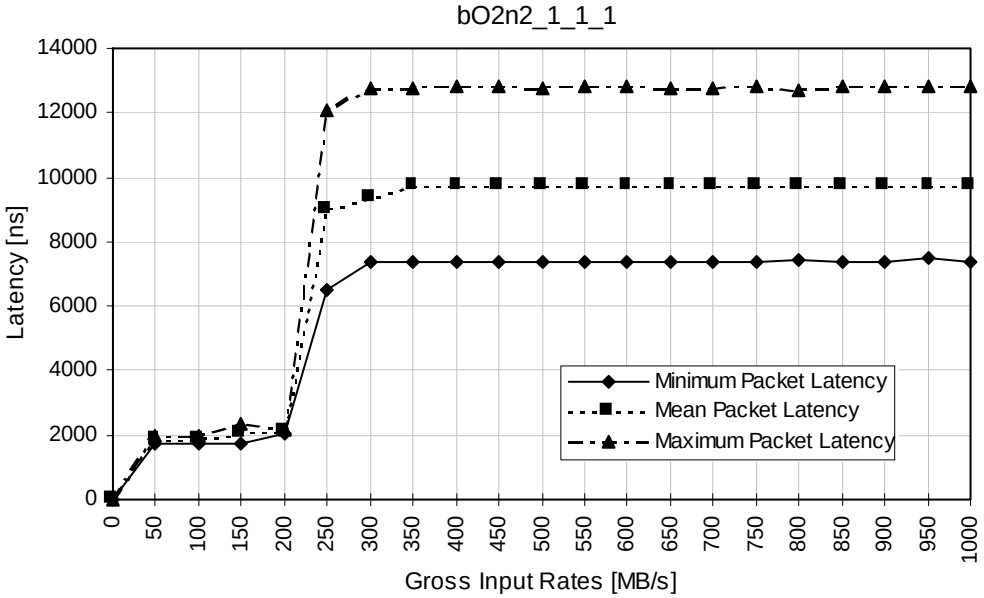

Bild 8.2.3: Latenz der Ringlet-Schalterkopplung nach Bild 8.2.1.

8.3 Durchsatzerhöhung im Schalter

Aufgrund der wenig beeindruckenden Resultate bzgl. des Durchsatzes eines Dolphinschen SCI-Schalters erhebt sich die Frage, wie sich dessen Leistung steigern läßt. Dazu wurden zwei unterschiedliche Konzepte entwickelt, die jedes für sich, zu einer beträchtlichen Durchsatzerhöhung führen. Das erste Konzept basiert darauf, Pakete möglichst lange auf dem Ring laufen zu lassen, auf dem sie erzeugt wurden, was eine Abkehr davon bedeutet, einen Schalter über Ringlets anzuschließen. Das zweite Konzept verwendet schalterintern multiple B-Links, um den Engpaß beim Datentransfer aufzuheben.

8.3.1 Durchgängige Ringe statt Ringlets

Es sei der Durchsatz T eines Schalters definiert als die Summe aller Durchsätze der einzelnen Ports. Dann gilt für T eines über Ringlets gekoppelten Schalters: $T = \min\{2t, B_l\} = B_l$. Darin ist t der Durchsatz an einem Port und B_l die B-Link-Bandbreite. In Bild 8.3.1 ist eine alternative Kopplung dargestellt, die funktional äquivalent zu der von Bild 8.2.1 ist, jedoch keine Ringlets sondern lange, durchgängige Ringe aufweist. Verwendet man die alternative Kopplung, ist ein Durchsatz von T'>T möglich, vorausgesetzt, daß ein gewisse Zahl n von der Gesamtzahl N der Pakete auf demselben Ring bleiben kann, auf dem sie erzeugt wurden. Der Grund für die Leistungssteigerung liegt darin, daß n Pakete den

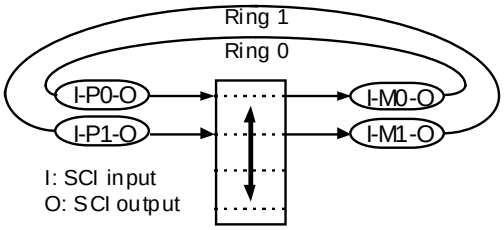

Bild 8.3.1: Alternative zur Ringlet-Schalterkopplung von Bild 8.2.1.

Schalter über einen Bypass-Fifo betreten und verlassen können, so daß der B-Link-Flaschenhals umgangen ist. Die Voraussetzung für $0<<n\leq N$ ist, daß in den Kommunikationsmustern von P0 und P1 Datenlokalität existiert. Im Fall eines Vier-Port-Schalters bedeutet dies, daß Daten häufiger zwischen P0 und M0 bzw. P1 und M1 ausgetauscht werden müssen als zwischen P0 und M1 bzw. P1 und M0. Ist der Verkehr nicht gleichverteilt, sondern hat eine Präferenz, ist es durch Umordnen immer möglich, die Position von P0 und P1 so zu wählen, daß $n>0$ gilt.

In Bild 8.3.2 ist das Ergebnis der Konfiguration von für den Grenzfall n=N zu sehen, d.h., wenn alle Pakete auf dem Ring bleiben können, wo sie erzeugt wurden. Erwartungsgemäß liegt der Durchsatz beträchtlich höher, da die zuvor limitierende B-Link-Arbitrierungszeit keine Rolle mehr spielt. Der Durchsatz

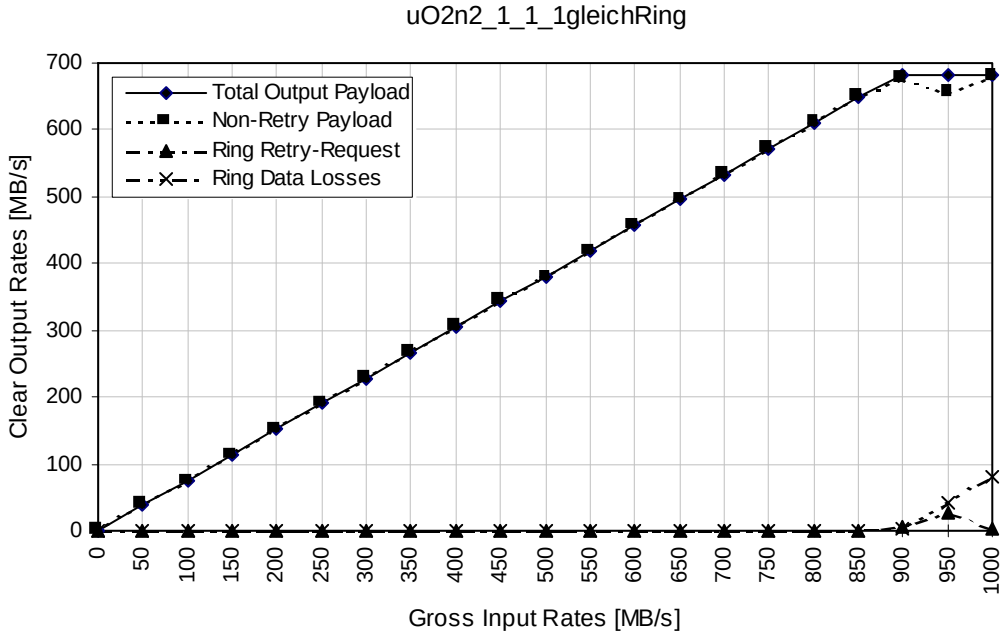

Bild 8.3.2: Leistungsanalyse der alternativen Kopplung für n=N und zwei Sender/Empfänger.

des Schalters entspricht jetzt genau der Summe der Durchsätze zweier elementarer SCI-Ringe. Beispielsweise hat man bei 850 MB/s summierter Eingangs-

rate einen Nettodurchsatz von 647,7 MB/s. Da die Summe zweier einzelner, elementarer SCI-Ringe 647,8 MB/s an Durchsatz ergibt, kann man sagen, daß sich der Schalter bei der gewählten Konstellation ähnlich wie zwei voneinander separierte Ringe verhält.

Allerdings ergeben sich ab dem Sättigungspunkt gewisse Unterschiede. Sie sind hauptsächlich durch eine andere Paketumlaufzeit verursacht, die wegen der zusätzlichen Bypass-Fifo-Durchlaufzeit um 48 ns höher liegt als beim elementaren SCI-Ring. Es kommt beim Schalter zu leichtem Retry-Verkehr (bis zu 26 MB/s bei 950 MB/s Eingangsrate). Außerdem existiert zwischen 450 und 500 MB/s ein ausgeprägtes Sättigungshochplateau, dessen Wert mit 682 MB/s über dem maximalen Durchsatz von 666 MB/s zweier einzelner elementarer SCI-Ringe liegt. Die Paketverluste betragen 80 MB/s bei 1000 MB/s Eingangsrate, statt 230 MB/s bei zwei elementaren SCI-Ringen.

Insgesamt bewirkt die Konfiguration von Bild 8.3.1 für n=N eine Durchsatzerhöhung um den Faktor 3,9 gegenüber Ringlet-Anschlußweise, was einen erheblichen Gewinn darstellt.

Die Latenz ist bis zum Sättigungspunkt ähnlich wie beim elementaren SCI-Ring. Allerdings muß man zusätzlich berücksichtigen, daß sich die erhöhte Umlaufzeit bei jeder Transaktion vier Mal bemerkbar macht (Request, Echo, Response, Echo), so daß wir einen um 48 ns*4 = 192 ns erhöhten Wert erhalten; insgesamt also 915 ns + 192 ns = 1107 ns. Der Simulator berechnet 1127 ns, was eine gute Übereinstimmung darstellt.

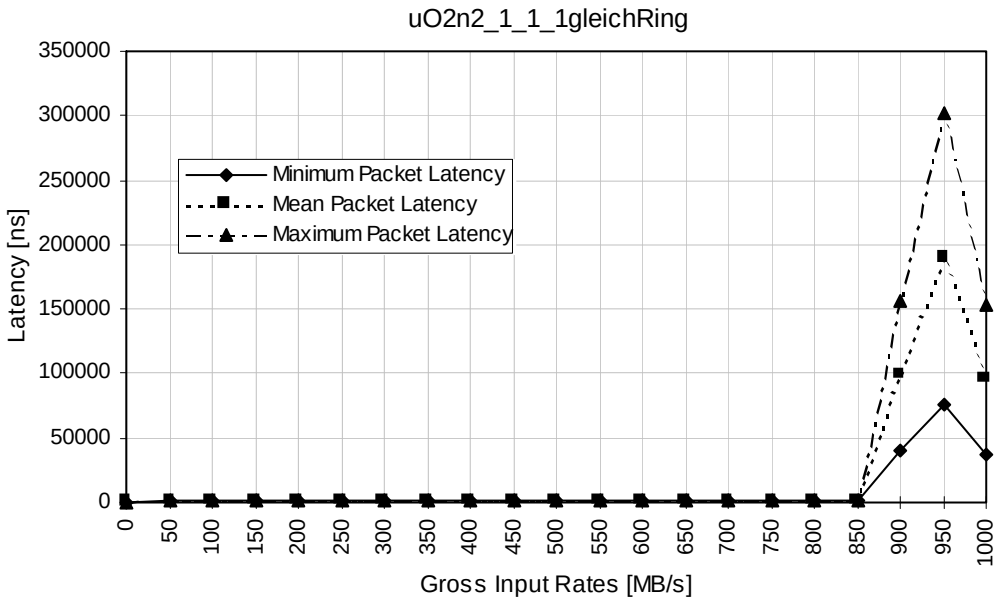

Bild 8.3.3: Latenz der alternativen Kopplung für n=N und zwei Sender/Empfänger.

Ab dem Sättigungspunkt ergeben sich bedingt durch den auftretenden Retry-Verkehr ein erheblicher Anstieg der Latenz. Sie springt auf den Spitzenwert von 301481 ns bei 975 MB/s Eingangsrate, was für einen Schalter sehr viel ist.

Offenbar sollte der Sättigungspunkt nicht überschritten werden, wenn man die Konfiguration in Echtzeitanwendungen einsetzen will.

Zusammenfassend kann man sagen, daß die alternative Konfiguration im günstigsten Fall eine um 48% niedrigere Latenz im Vergleich zur Ringlet-Anschlußweise hat (2344 ns zu 1127 ns), sofern der Schalter nicht überlastet ist.

In der Praxis liegt in der Regel weniger als 100% Datenlokalität vor, es gilt also n<N. Um dieser Tatsache Rechnung zu tragen, wird im folgenden der Extremfall n=0 analysiert. Reale Anwendungen haben je nach Lokalitätsgrad einen Durchsatz, der zwischen beiden Extremen n=0 bzw. n=N liegt. Wie nicht anders zu erwarten, erhält man für n=0 ein ähnliches Verhalten wie bei der konventionellen Ringlet-Anschlußweise, da jetzt sämtliche Pakete über das schalterinterne B-Link gehen müssen. Gemäß Simulation ergibt sich ein Durchsatz

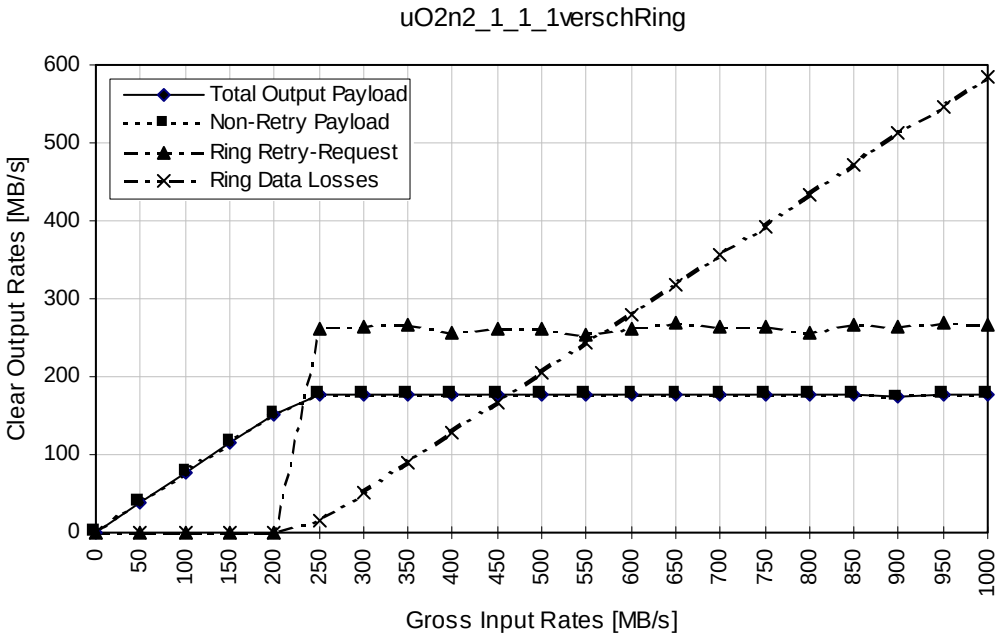

Bild 8.3.4: Leistungsanalyse der alternativen Kopplung für n=0 und zwei Sender/Empfänger.

von 177 MB/s bei 1000 MB/s Eingangsdatenrate, was identisch zu den 176 MB/s des Ringlet-Schalters ist. Der Retry-Verkehr ist mit 265 MB/s zwar geringer als bei der Ringlet-Anschlußweise reicht aber offenbar immer noch aus, den Schalterdurchsatz zu begrenzen.

Bis zum Sättigungspunkt von ca. 200 MB/s ist die Latenz im Vergleich zum Ringlet-Fall leicht erhöht (2770 ns zu 2344 ns). Leider springt sie beim Überschreiten der Sättigung wie schon für n=N auf erheblich größere Werte und erreicht bis zu 48662 ns.

Ergebnis:

Zusammenfassend kann man sagen, daß bei zwei Sendern und Empfängern die

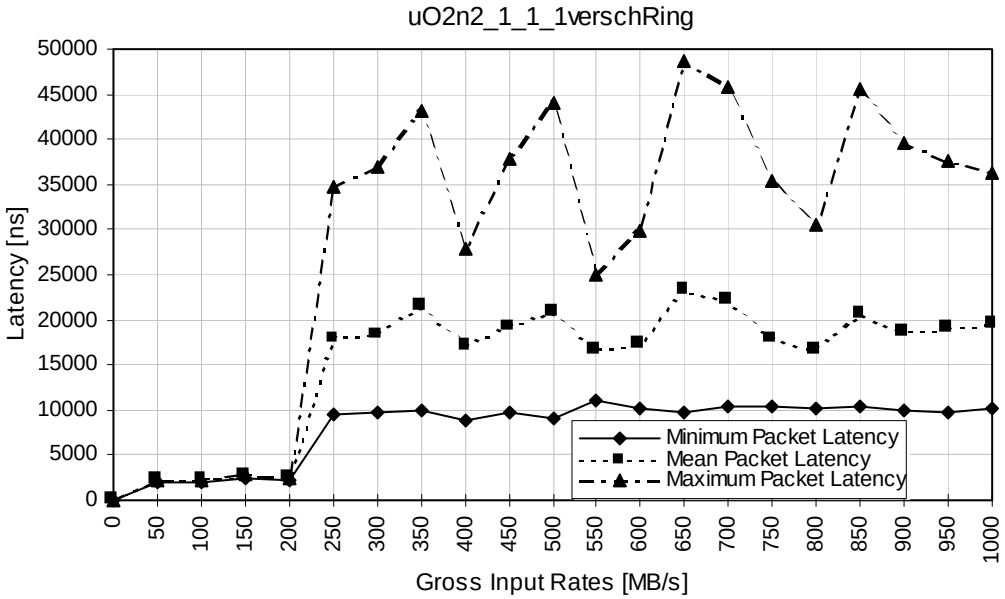

Bild 8.3.5: Latenz der alternativen Kopplung für n=0 und zwei Sender/Empfänger.

alternative Konfiguration im ungünstigsten Fall ungefähr gleich wie die Ringlet-Konstellation ist, sofern man unterhalb der Sättigungsgrenze des Schalters bleibt. Aufgrund des Durchssatzanstiegs um den Faktor 3,9, den man im günstigsten Fall erhält, bedeutet dies, daß die alternative Konfiguration dem über Ringlets angeschlossenen Schalter überlegen ist und deshalb bevorzugt werden sollte. Im nächsten Kapitel wird erläutert, daß die alternative Konfiguration noch einen weiteren Vorteil aufweist.

Vier Sender am Schalter

Bei der Leistungsanalyse der alternativen Konfiguration sind bislang des besseren Vergleichs wegen nur zwei der 4 Ports mit Sendern bzw. Empfängern verbunden gewesen. Um die Anschlußweise der durchgängigen Ringe voll auszunutzen, können jedoch, ohne Mehrkosten beim Schalter, bis zu vier Sender und deren Empfänger angeschlossen sein (Bild 8.3.6a). Wenn n=N gilt, sollte gemäß der Anschauung darunter weder der Durchsatz am einzelnen Empfänger leiden, noch die Latenz ansteigen. Vielmehr sollte sich der Durchsatz verdoppeln. Die Simulation zeigt, daß die Verhältnisse tatsächlich wie vorausgesagt sind: der Gesamtdurchsatz verdoppelt sich auf 1365 MB/s bei gleichbleibender Latenz (1127 ns). Entsprechend des erhöhten Durchsatzes bei verdoppelter Quellenzahl verschiebt sich der Sättigungspunkt von 900 MB/s auf 1800 MB/s Eingangsdatenrate, wobei die Paketverluste selbstverständlich auch verdoppelt sind. Das Diagramm der Latenzen ist bei vier Sendern identisch zu dem von zwei Sendern (Bild 8.3.5) und wird deshalb nicht extra gezeigt.

Speziell im Vergleich zur Zwei-Sender-Ringlet-Lösung schneidet bei n=N die Vier-Sender-Nicht-Ringlet-Lösung, die auf langen, durchgängigen Ringen

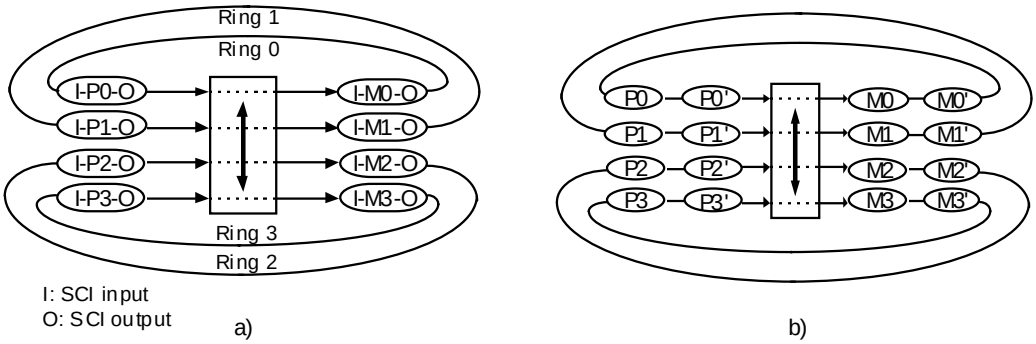

Bild 8.3.6: Erweiterte Nutzung der alternativen Kopplung mit 4 bzw. 8 Sender/Empfänger.

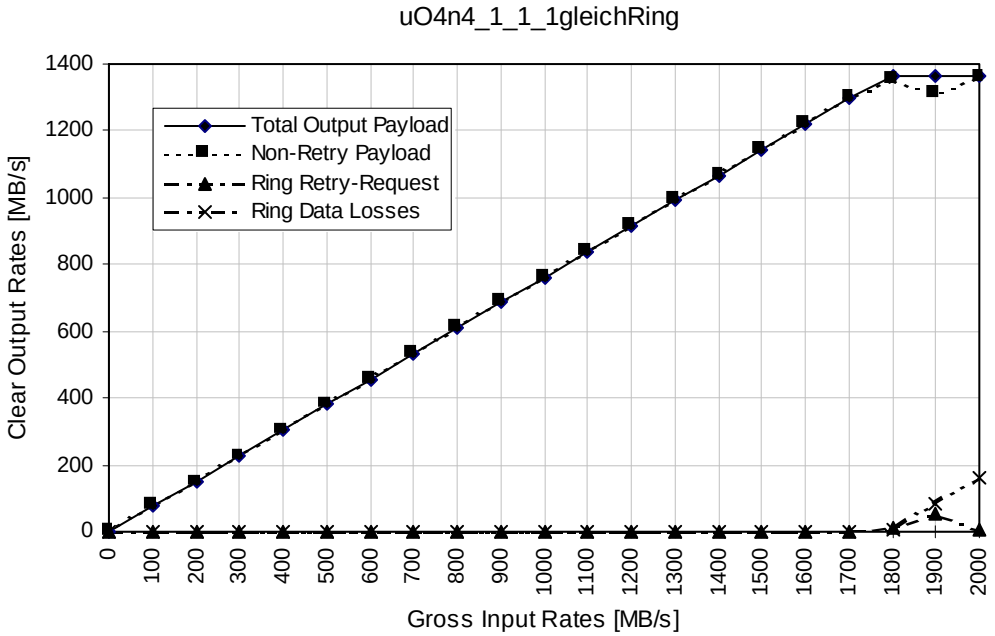

Bild 8.3.7: Leistungsanalyse der alternativen Kopplung für n=N und 4 Sender/Empfänger.

basiert, besonders gut ab. Der Maximalwert des Durchsatzes von 1365 MB/s bei 2000 MB/s Eingangsdatenrate erreicht das 7,8-fache des Ringlet-Durchsatzes. Er übersteigt damit auch mehr als das Doppelte einer B-Link-Bandbreite. Die Paketverluste betragen dabei 160 MB/s, was nur 27% der Verluste der Zwei-Sender-Ringlet-Lösung sind, und der Retry-Verkehr aller Ringe hat eine Größe von lediglich 8,9 MB/s. Der Sättigungspunkt liegt um den Faktor 7,2 höher als bei der Zwei-Sender-Ringlet-Lösung.

Für n=0 sollte sich anhand der Anschauung beim Übergang von zwei auf vier Sender der Durchsatz am einzelnen Empfänger halbieren, bei insgesamt gleichbleibendem Schalterdurchsatz. Die Anschauung sagt weiterhin eine Erhöhung der Latenz voraus. Diese Vorstellung wird von den Simulationen bestätigt: der Nettogesamtdurchsatz erreicht mit 177 MB/s denselben Wert wie wir ihn zuvor

beim 2-Senderschalter sowohl in Ringlet- wie auch in Nicht-Ringlet-Konfiguration hatten. Entsprechend des gleichbleibenden Durchsatzes verändert sich der Sättigungspunkt nicht.

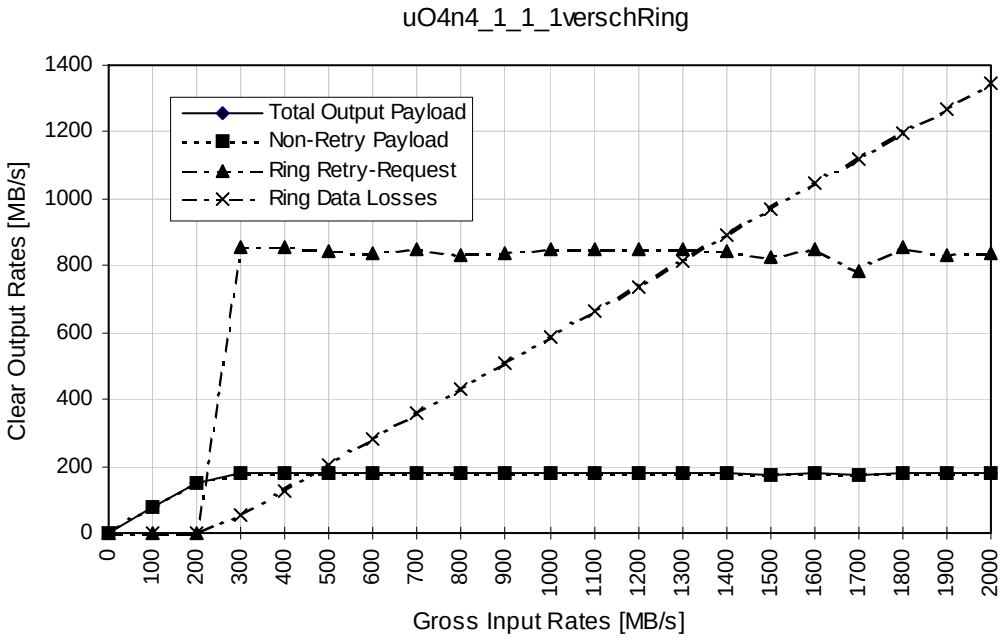

Bild 8.3.8: Leistungsanalyse der alternativen Kopplung für n=0 und 4 Sender/Empfänger.

Die Latenz hat sich bei n=0 unterhalb Sättigungspunkts, von 2490 ns auf 3680 ns leicht erhöht. Leider steigt sie ab dem Sättigungspunkt von ursprünglich 12797 ns bei der Zwei-Sender-Ringlet-Konfiguration bzw. 48663 ns bei der Zwei-Sender-Nicht-Ringlet-Konfiguration auf jetzt 110094 ns an. Dies entspricht einer Zunahme um eine Größenordnung. Allerdings sind dafür auch doppelt so viele Sender am selben Schalter angeschlossen, und bei gegebener Datenlokalität kann eine beträchtliche Durchsatzerhöhung erzielt werden.

Im nächsten Kapitel wird gezeigt, wie der Schalterdurchsatz erhöht werden kann, wenn keine Datenlokalität vorliegt. Ein solcher Fall liegt z.B. bei einer parallelen Matrixtransposition vor, bei der die Elemente zeilen- oder spaltenweise auf Prozessoren verteilt sind.

Ergebnis:

Das bedeutet, daß eine Konfiguration aus vier Sendern, die über durchgängige Ringe angeschlossen sind, bzgl. des Durchsatzes auch im ungünstigsten Fall nicht schlechter als die Ringlet-Konfiguration mit zwei Sendern abschneidet, jedoch doppelt so viele Sender/Empfänger verbinden kann. Im günstigsten Fall sind das 7,8-fache an Durchsatz zu erwarten, bei nur leicht erhöhter Latenz. Voraussetzung ist in allen Fällen, daß der Schalter nicht überlastet wird. Bei Überlastung steigt der Maximalwert der Latenz um eine Größenordnung auf

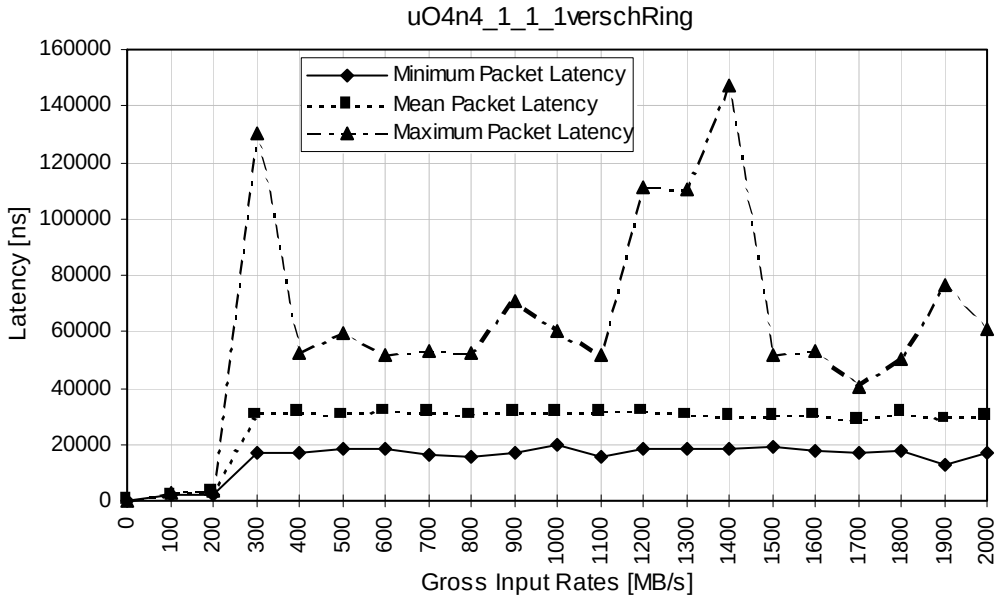

Bild 8.3.9: Latenz der alternativen Kopplung für n=0 und 4 Sender/Empfänger.

110 µs an. Insgesamt ist die Schalteranschlußweise über durchgängige Ringe und mit vier aktiven Ports allen Zwei-Port-Konfigurationen vorzuziehen.

Zu guter Letzt soll noch auf eine architektonische Variante der alternativen Kopplung hingewiesen werden, die eine weitere Verdopplung der Sender- und Empfängerzahl bewirkt, bei allerdings halbierter Bandbreite pro Sender/Empfängerpaar (Bild 8.3.6b). Da nicht für alle Anwendungen die volle Bandbreite eines SCI-Ringes erforderlich ist, erlaubt diese Variante durch geeignete Wahl der Sender pro Ring, jedem Sender die Bandbreite zuzuteilen, die er braucht. Eine solche Flexibilisierung kann z.B. sehr gut in Datenerfassungssystemen eingesetzt werden, da dort die Abtastraten der Sensoren stark streuen.

8.3.2 Multiple B-Links

Das zweite Verfahren zur Erhöhung des rel. geringen Durchsatzes kommerzieller SCI-Schalter besteht darin, statt eines einzigen schalterinternen Kommunikationspfades multiple B-Links einzusetzen und diese parallel zu schalten. Die Parallelschaltung von B-Links wird dadurch bewirkt, daß man zwei oder mehrere Link-Controller-Bausteine in Serie schaltet, so wie dies in Bild 8.3.10 für den Fall von vier LC-Bausteinen gezeigt ist. Jeder Schalteranschluß nach außen besteht dabei intern aus einer Kaskade von vier Controller-Bausteinen, und ein Paket, das von einem Link-Controller nicht akzeptiert wird, hat eine neue Chance bei dessen Nachfolger in der Kaskade. Die Bandbreite des Schalters kann sich dadurch theoretisch auf bis zu 2,4 GB/s erhöhen. Die folgende Analyse wird zeigen, ob dieser Wert tatsächlich erreicht wird.

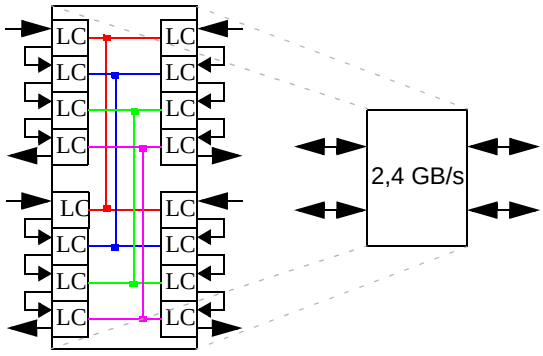

Bild 8.3.10: Multiple B-Links im Schalter durch Kaskadierung von Link-Controllern.

Die zunächst einfach erscheinende Serienschaltung von LC-Bausteinen birgt einige technische Komplikationen. Zum einen wird die Durchlaufzeit des Bypass-Fifos auf 48 ns*4 = 192 ns erhöht, zum anderen bedarf es einer Zuordnungsmethode (Scheduling) von eintreffenden Paketen zu B-Links, um die Ressourcen der schalterinternen Kommunikationspfade gleichmäßig auszulasten. Zum dritten ergeben sich für Retry- und Echopakete spezielle Adressierungsprobleme, die aus den SCI-Protokollmechanismen resultieren.

Die Vervierfachung der Durchlaufzeit kann nicht vermieden werden, sie ist jedoch für die meisten Anwendungen tolerabel und wird deshalb im folgenden nicht weiter untersucht. Wichtig ist der schalterinterne Lastausgleich zwischen parallelen B-Links. Die dazu entwickelten Scheduling-Methoden werden im Kapitel 8.3.3 "Scheduling multipler B-Links" beschrieben. Die Art des Adressierungsproblems und seine Lösung wird im Kapitel 8.3.4 "Das Adressierungsproblem bei der B-Link-Auswahl" dargestellt. Schließlich wird der durch Parallelschalten von B-Links erzielte Leistungszuwachs im Kapitel 8.3.5 "Leistung bei multiplen B-Links" erörtert.

8.3.3 Scheduling multipler B-Links

Der Zweck des B-Link-Schedulings ist es, ein an einem Port eintreffendes Paket, das zu einem anderen Port transferiert werden soll, ein B-Link unter der Randbedingung zuzuordnen, daß alle B-Links im Schalter gleichmäßig ausgelastet sind. Mit der Zuordnung eines B-Links zu einem Paket ist zugleich auch festgelegt, von welchem Link-Controller-Baustein der Kaskade das Paket akzeptiert werden muß.

Eine einfache Methode zur B-Link bzw. LC-Bausteinauswahl ist es, eine Untermenge der im Paket transportierten Adreß- oder Datenbits zu verwenden. Beispielsweise könnten die untersten zwei LSB der Paketzieladresse dazu dienen, einen von vier LC-Bausteinen auszuwählen. Andere Möglichkeiten sind die Herkunftsadresse oder die Transaktionsnummer, von denen 2 Bits für diesen Zweck ausreichen. Der Nachteil dieser Methode ist, daß ein Lastausgleich nicht garantiert werden kann. Werden nacheinander Pakete mit derselben Ziel-

oder Herkunftsadresse empfangen, wird stets derselbe LC-Baustein ausgewählt.

Aus diesem Grunde wurde bei SCINET eine andere Scheduling-Methode herangezogen. Mit Hilfe von dezentralen Zählern, die in den LC-Bausteinen unterzubringen sind, wird eine Reih-um-Paketannahme (Round Robin Scheduling) implementiert, die für eine gerechte Aufteilung der B-Link-Ressourcen sorgt.

Die dezentrale Paketzuordnung funktioniert so, daß immer dann der Paketzähler in einem LC-Baustein dekrementiert wird, wenn ein Paket bei ihm eingetroffen ist. Hat der Zähler den Stand 0 erreicht, wird das Paket akzeptiert, sofern es der Pufferplatz erlaubt. Ist der Puffer voll, muß der selektierte LC-Baustein ein negatives Echo aussenden. In jedem Fall wird anschließend der Zähler abhängig von seiner Position in der Kaskade mit einem neuen Anfangswert geladen. Beispielsweise erhält der oberste Zähler in einer Reihe von vier LC-Bausteinen den Anfangswert 3, der unterste den Wert 0. Das bedeutet, daß jedes Paket, das beim letzten LC-Baustein ankommt, von diesem zu akzeptieren ist. Zwischenkoten an der Position i (beginnend von unten mit Position 0) werden mit dem Wert i geladen. Das hat zur Folge, daß der oberste LC-Baustein, dessen Zähler mit 3 initialisiert worden ist, jedes vierte Paket annimmt. Von den verbleibenden 3 Paketen akzeptiert sein Nachbar in der Kaskade jedes 3. Paket; dessen Nachbar nimmt von den verbleibenden 2 Paketen jedes zweite u.s.w. Dadurch erhält sukzessive von unten nach oben jeder Baustein von je vier Paketen eines, wodurch insgesamt ein dezentral implementiertes Round Robin-Scheduling entsteht.

Ein zentralisiertes Scheduling wurde deshalb nicht gewählt, weil dadurch separate Hardware oder ein „Master“-LC-Baustein erforderlich wäre. Andere dezentrale Scheduling-Methoden, wie beispielsweise über ein Enable-Signal oder über ein Token-Passing-Mechanismus können bei der Link-Controller-Kaskade nicht verwendet werden, wie folgende Überlegung zeigt:

Angenommen, zwei Pakete treffen unmittelbar hintereinander an einer Kaskade ein, und der letzte LC-Baustein der Kaskade wäre an der Reihe, das erste der beiden Pakete zu akzeptieren. Dann müßte er den Nachfolger in der Kaskade, also den ersten LC-Baustein freischalten, damit dieser das zweite Paket akzeptiert. Zu diesem Zeitpunkt hat das zweite Paket jedoch bereits die Position des ersten LC-Bausteins passiert, kann also nicht mehr von diesem angenommen werden und die Paketakzeptanz schlägt fehl.

Zusätzlich zum deterministischen Round Robin-Scheduling der Schalter-B-Links wurden in SCINET noch zwei andere, adaptive Methoden der Link-Zuordnung verwirklicht, die den momentanen Pufferfüllgrad der Link-Controller-Bausteine Rechnung tragen. Das Ziel der adaptiven Paketzuordnung ist es, einem dynamischen Lastausgleich und damit in eine weitere Durchsatzsteigerung herbeizuführen. Bei der einen adaptiven Strategie nimmt der erste LC-Baustein in der Kaskadenreihe, der über einen nicht-vollen Paketempfangspuffer verfügt, das Paket an. Um zu gewährleisten, daß wenigstens einer der Bausteine in der Kaskade für ein eintreffendes Paket zuständig ist, gilt zusätzlich die Regel, daß der letzte Baustein in der Reihe für alle Pakete zuständig ist, die seine Po-

sition erreichen. Hat dieser einen nicht-vollen Empfangspuffer, kann er das Paket einspeichern, im anderen Fall ist er für die Aussendung eines negativen Echos verantwortlich.

Die zweite in SCINET implementierte, adaptive Strategie der B-Link-Auswahl ist eine abgeschwächte Variante der ersten. Hier gilt die Regel nicht, daß der letzte Baustein der Kaskade für alle dort befindlichen Pakete zuständig ist. Vielmehr kann dieser bei vollem Empfangspuffer das Paket passieren lassen, in der Annahme, daß der Pfad eines derart fehlgeleiteten Pakets auf seinem späteren Weg durch das Netz wieder richtig gestellt wird. Die zweite Methode funktioniert selbstverständlich nur bei Netzen mit Pfadkompensation.

8.3.4 Das Adressierungsproblem bei der B-Link-Auswahl

Multi-B-Link-Schalter bestehen aus einer Serienschaltung von SCI-Schnittstellenbausteinen, die an jedem Schalterein- und -ausgang angebracht und als Knotenkaskade verschaltet sind. Funktionell sind alle Knoten einer Kaskade identisch, denn ihre B-Links verlaufen zueinander parallel, geschwindigkeitsmäßig ergibt sich jedoch durch die Parallelschaltung eine Erhöhung des Schalterdurchsatzes. Die Zuordnung von einlaufenden Paketen zu einer bestimmten Schnittstelle innerhalb einer Kaskade, d.h. die Entscheidung, welcher Knoten welches Paket annimmt, kann nach den bereits beschriebenen Strategien erfolgen. Diese basieren auf der Verfügbarkeit der Paketempfangsspuffer oder auf dem einfacheren Round Robin-Scheduling.

Da im SCI-Standard eine Adressierung einer ganzen Gruppe von SCI-Knoten nicht vorgesehen ist, ist es von jedem Absender eines Datenpakets erforderlich, einen bestimmten Knoten innerhalb einer Kaskade als Paketziel auszuwählen. Unabhängig von der gewählten Strategie der Paketannahme resultiert daraus das Problem, daß beispielsweise das Paket an den ersten Knoten einer Kaskade adressiert worden ist, während es potentiell von allen Knoten akzeptiert werden könnte. D.h., die Knoten einer Kaskade sollten auch mit den Adressen ihrer Knotennachbarn aktiviert werden können.

Das Problem kann durch eine veränderte Adreßdekodierung in den einzelnen Kaskaden gelöst werden, die darauf beruht, nur die höherwertigen Bits der Paketadresse mit der eigenen SCI-Adresse zu vergleichen, so daß knotenseitig eine Gruppenadressierung möglich ist. Dies setzt eine Modifikation der z.Z. erhältlichen Implementierungen von SCI-Schnittstellenbausteinen voraus, die in Kapitel 9.3.2 "Implementierung in Silizium" näher beschrieben wird.

Aus der Serienschaltung von SCI-Knoten ergibt sich noch ein zweites Problem, dessen Lösung eine bestimmte Art der SCI-Protokollimplementierung erfordert. Dieses Problem ist graphisch in Bild 8.3.11 dargestellt: Ein Request-Paket wird von S nach A geschickt, aber von B akzeptiert, weil es das B-Link-Scheduling so will. Wenn nun von B die Herkunftsadresse für das nachfolgende Echopaket, das B senden muß, aus dem Feld der Zieladresse des akzeptierten Request-Pakets bestimmt werden würde, würde ein potentielles Retry-Paket

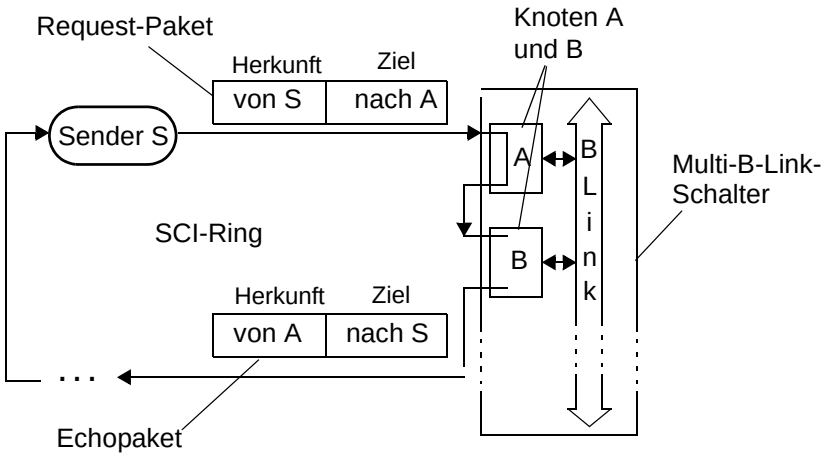

Bild 8.3.11: Adressierungsproblem in einer Kaskade von Link-Controllern.

nach A geschickt werden, und der SCI-Pufferallozierungsmechanismus wäre gestört. Denn die Zieladresse des von B akzeptierten Pakets lautet auf den Knoten A. Um dies zu vermeiden, muß das vom IEEE-Standard vorgeschriebene SCI-Handshake-Protokoll folgendermaßen implementiert werden:

Der nach der jeweiligen Lastverteilungsstrategie für ein ankommendes Paket zuständige Knoten ist verpflichtet, ein positives oder negatives Echo an den Paketsender zurückzuschicken. Im Falle eines negativen Echos muß sichergestellt sein, daß das nachfolgende Retry-Paket zu dem Knoten zurückfindet, der für das Paket zuständig ist. Dabei hat der Sender des Retry-Pakets das Problem, daß er nicht weiß, welcher Knoten das ist. Deshalb muß das abgeschickte negative Echo des Paketannehmers als Herkunftsadresse seine eigene Adresse und nicht die ursprüngliche Paketzieladresse enthalten. Das erfordert u.U. eine Adreßkorrektur, die in Bild 8.3.12 dargestellt ist. Damit auf der Gegenseite der Sender S

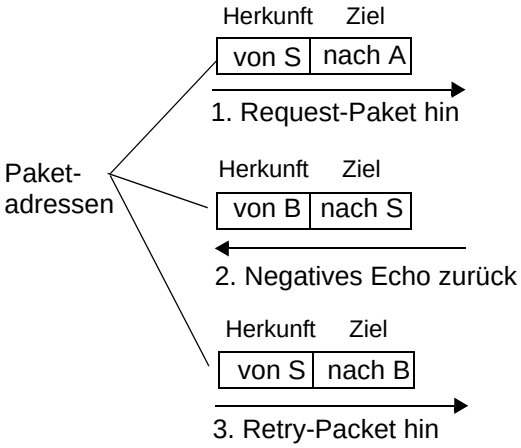

Bild 8.3.12: Lösung des Adressierungsproblems bei negativen Echos.

das Echopaket als zum vorangegangenen Request-Paket gehörend identifizieren kann, darf dazu bei S nicht die Herkunftsadresse des Echos verwendet werden - diese hat sich ja geändert - sondern vielmehr dessen transactionid. Nach der Identifikation des negativen Echos muß S die Herkunftsadresse des Echos als Zieladresse für das Retry-Paket einsetzen.

Im Falle eines positiven Echos gibt es für die Herkunftsadresse des Echopakets zwei gleichermaßen funktionierende Varianten. Entweder wird von B als Herkunftsadresse die Zieladresse des Requests eingesetzt, also A, oder er setzt seine eigene ein (=B). Im ersten Fall würde für den Sender S suggeriert, daß A das Paket wie geplant genommen hat, im zweiten Fall würde S das Echopaket anhand seiner transactionid identifizieren. Beides ist möglich, bei SCINET wurde die erste Variante implementiert.

Zusammenfassend kann man sagen, daß SCI-Schalter aus kaskadierten Schnittstellenbausteinen aufgebaut werden können, ohne daß die existierenden SCI-Protokolle verletzt oder modifiziert werden müssen, sofern die beschriebenen Adreßersetzungsmechanismen in zukünftige Generationen von Link-Controller-Bausteinen integriert werden. Die Kaskadierung dient zur Durchsatzerhöhung des Schalters.

8.3.5 Leistung bei multiplen B-Links

In den nächsten Kapiteln sollen zunächst die Ergebnisse der Leistungsanalyse für SCI-Schalter präsentiert werden, die über multiple B-Links verfügen und mit Hilfe von Ringlets angeschlossen sind. Danach wird erläutert, wie sich die Kombination beider Optimierungsmöglichkeiten, durchgängigen Ringe und multiple B-Links, hinsichtlich des Schaltersdurchsatzes und der Latenz auswirken. Den Analysen liegt, sofern nicht anders vermerkt, eine Vergabe der B-Links nach dem Round Robin-Verfahren zugrunde.

Schalter in Ringlet-Konfiguration mit zwei B-Links

In Bild 8.3.13 sind der Nettodurchsatz, der Retry-Verkehr und die Paketverluste für einen mit zwei B-Links ausgestatteten Schalter in Ringlet-Konfiguration dargestellt. Man sieht, daß sich im Gegensatz zum Mono-B-Link-Schalter der größte Durchsatz bei 1000 MB/s Eingangsrate von 176 MB/s auf 353 MB/s exakt verdoppelt hat, während die Paketverluste um 177 MB/s von 585 MB/s auf 408 MB/s zurückgegangen sind. Des weiteren hat sich der Retry-Verkehr von 404 MB/s auf 224 MB/s nahezu halbiert. Schließlich wurde der Sättigungspunkt von 250 MB/s Eingangsrate auf ca. 500 MB/s hinausgeschoben.

Die Latenz verhält sich bis ca. 450 MB/s deterministisch und weist mit 2895 ns gegenüber dem Mono-B-Link-Schalter einen nur leicht erhöhten Wert auf (zuvor waren es 2344 ns). Ab dem Sättigungspunkt wird sie nicht-deterministisch und erreicht Spitzenwerte von 18393 ns (zuvor 12797 ns) bei insgesamt

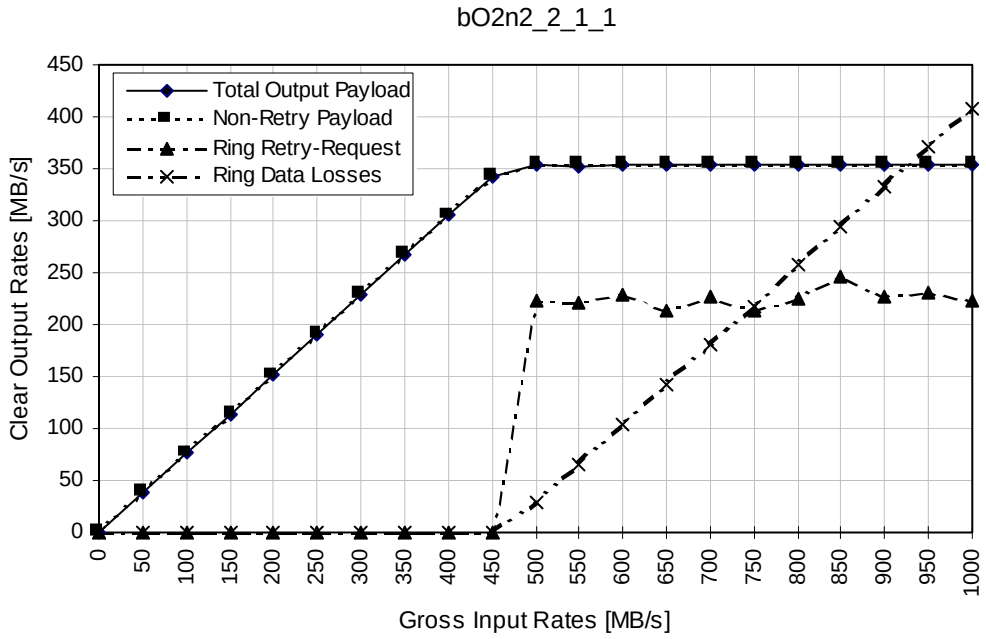

Bild 8.3.13: Durchsatz der Ringlet-Konfiguration bei zwei B-Links pro Schalter.

stärkeren Fluktuationen.

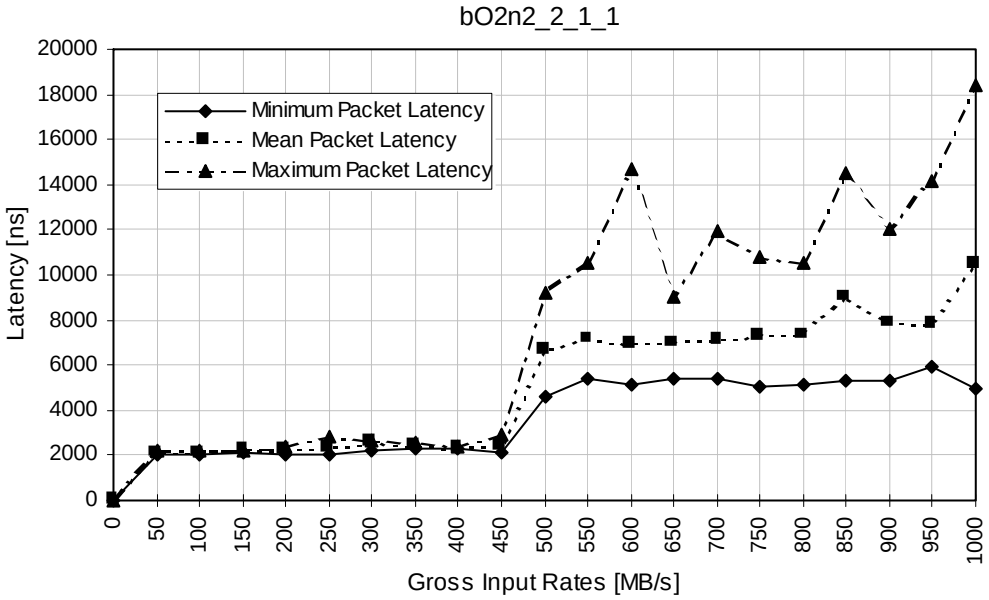

Bild 8.3.14: Latenz der Ringlet-Konfiguration bei zwei B-Links pro Schalter.

Das bedeutet, daß der Einsatz von zwei B-Links bei Schaltern in Ringlet-Konfiguration eine Verdopplung des Durchsatzes bringt, ohne die Latenz wesentlich zu verschlechtern.

Schalter in Ringlet-Konfiguration mit drei B-Links

Bei der Parallelschaltung von drei B-Links in einem über Ringlets mit Sendern und Empfängern gekoppelten Schalter wird mit 529 MB/s exakt der dreifache Durchsatz gegenüber einem Mono-B-Link-Schalter erreicht. Der Sättigungspunkt ist mit 700 MB/s Eingangsrate um nahezu den Faktor drei hinausgeschoben, während der Retry-Verkehr nur noch 41 MB/s beträgt, was einer Abnahme um den Faktor 9,8 entspricht. Der Zunahme des Nettodurchsatzes um 353 MB/s steht eine Abnahme der Paketverluste um denselben Betrag gegenüber, so wie es auch zu erwarten ist. Dem rel. geringen Retry-Verkehr nach zu schließen, sind bei drei B-Links die summierte Port-Bandbreite und die interne Transferkapazität (incl. der B-Link Setup-Zeiten) hinsichtlich deren Geschwindigkeiten nahezu ausbalanciert. Dabei ist die zu vermerkende Randbedingung, daß NWRITE64-Transaktionen bestehend aus Request- und Response- Paketen transferiert werden.

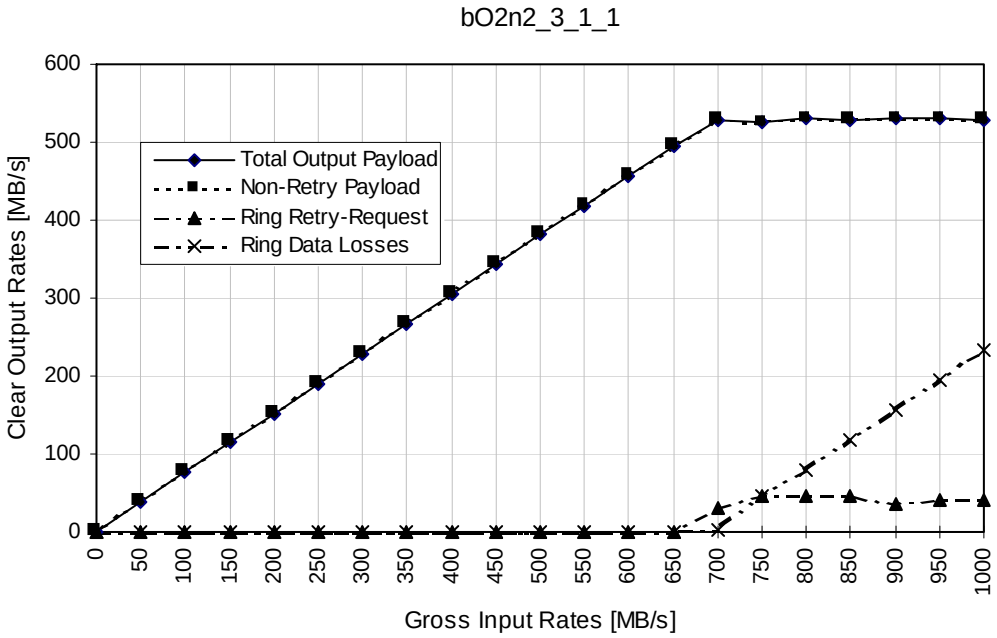

Bild 8.3.15: Durchsatz der Ringlet-Konfiguration bei drei B-Links pro Schalter.

Die Latenz ist bis ca. 600 MB/s Eingangsrate deterministisch und mit maximal 3257 ns gegenüber der Mono-B-Link-Lösung sowie des 2-B-Link-Schalters leicht erhöht. Die beiden letzteren erreichen jedoch erheblich früher die Sättigung. Ab dem Sättigungspunkt werden beim Drei-B-Link-Schalter bis zu 19836 ns für den Transfer eines Pakets benötigt, was etwas über der Latenzzeit des 2-B-Link-Schalters liegt. Die Fluktuationen haben insgesamt abgenommen. Das heißt, daß bei Ringlet-Schaltern drei B-Links eine Verdreifachung des Durchsatzes leisten. Die Latenz ist bis zu 600 MB/s Eingangsdatenrate deterministisch und gegenüber dem Mono-B-Link-Schalter leicht erhöht.

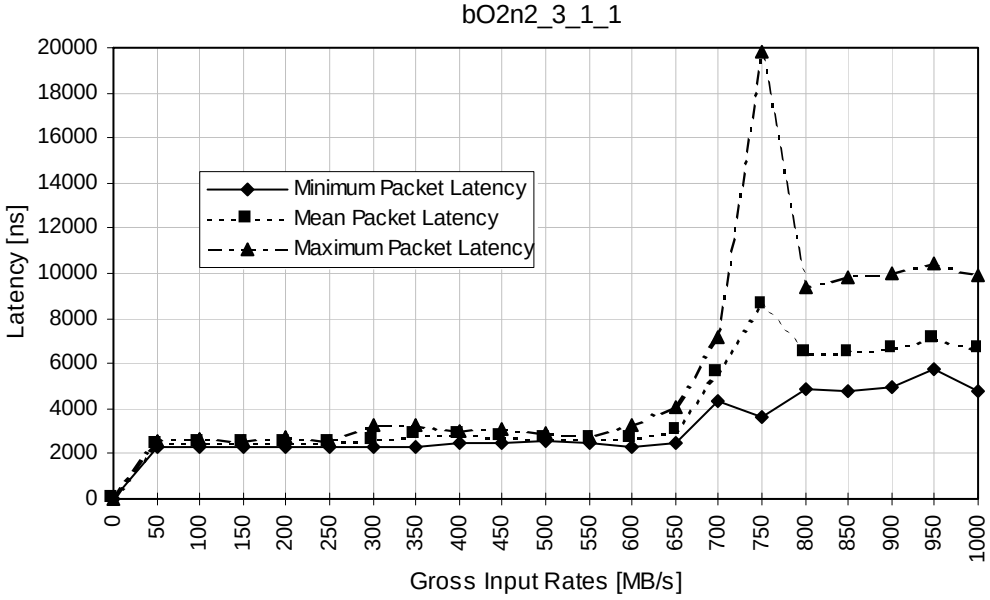

Bild 8.3.16: Latenz der Ringlet-Konfiguration bei drei B-Links pro Schalter.

Schalter in Ringlet-Konfiguration mit vier B-Links

Erwartungsgemäß bringt die Parallelschaltung von vier B-Links keine wesentliche Durchsatzerhöhung gegenüber einem 3-B-Link-Schalter, da die summierte Port-Bandbreite bereits bei drei B-Links gegenüber der internen Transferkapazität nahezu ausbalanciert ist. Der Durchsatz steigt um lediglich 8 MB/s auf 537 MB/s an (bei 950 MB/s Eingangsrate). Der Retry-Verkehr sinkt hingegen nochmals von zuvor 41 MB/s beim Drei-B-Link-Schalter auf jetzt 9,7 MB/s (bei 1000 MB/s Eingangsrate).

Die Latenz ist bis 600 MB/s Eingangsrate deterministisch und hat sich mit 4232 ns gegenüber einem Mono-B-Link-Schalter in etwa verdoppelt, was auf den Zusatzaufwand aufgrund der vermehrten B-Link-Scheduling-Aufgaben und die erhöhten Bypass-Fifo-Durchlaufzeiten zurückzuführen ist.

Ergebnis:

Insgesamt kann man sagen, daß sich der Einbau von bis zu drei B-Links in einen LC2-basierten Vier-Port-SCI-Schalter, der über Ringlets angeschlossen ist, lohnt. Der Durchsatz verhält sich proportional zur Zahl der parallelgeschalteten B-Links. Bei drei B-Links werden 529 MB/s Durchsatz erreicht. Die Latenz ist gegenüber dem Mono-B-Link-Schalter unterhalb des Sättigungspunkts von 600 MB/s Eingangsrate trotz des zusätzlich erforderlichen B-Link-Schedulings und der erhöhten Ringumlaufzeiten deterministisch und mit 3257 ns nur unwesentlich erhöht. Oberhalb der Sättigung wird sie indeterministisch und schwankt zwischen 5725 ns und 19836 ns.

8.3.6 Schalter mit durchgängigen Ringen und multiplen B-Links

In diesem Abschnitt wird die Kombination von durchgehenden SCI-Ringen (=nicht-Ringlet-Konfiguration) mit multiplen B-Links untersucht. Die Frage ist, ob sich die positiven Wirkungen beider Maßnahmen durch deren Kombination addieren lassen. Dazu wird zwischen den beiden Extremfällen n=N (alle Pakete bleiben auf dem Ring, auf dem sie erzeugt wurden) und n=0 (alle Pakete müssen ihren Ring beim Schalter verlassen) unterschieden.

Zwei Sender, durchgehende Ringe und zwei B-Links

Für den Fall n=N sagt die Anschauung, daß es keinen Unterschied in der Leistung der Schalter geben sollte, egal wieviele B-Links intern parallel verlaufen, da kein Pakete über ein B-Link transferiert werden muß. Für n=0 hingegen müßte das Verhalten analog zum Ringlet-Fall sein, d.h. n-fache B-Links (n=1,2,3,4) sollten in n-fachem Durchsatz resultieren.

Die Simulationen bestätigen diese Anschauung: sind zwei Sender und zwei Empfänger an einen Vier-Port-Schalter angeschlossen und gilt n=N, hat der Durchsatz die gleichen Werte (682 MB/s) wie bei einem einzigen B-Link, auch die Latenz verändert sich nicht.

Entgegen der Voraussage bewirkt bei n=0 die Parallelschaltung von zwei B-Links keine Verdopplung des Durchsatzes. Vielmehr steigt der Durchsatz nur um 55% von 177 MB/s auf 275 MB/s an. Der Retry-Verkehr geht hingegen überproportional von 265 MB/s auf 48 MB/s zurück. Die Latenz erhöht sich beim Übergang auf mehrfache B-Links bezogen auf den Ausgangswert von 48663 ns nur leicht auf 76119 ns (jeweils Maximalwert). Dies war bereits bei den über Ringlets gekoppelten Schaltern so. Die Ergebnisse für n=0 sind in Bild 8.3.17 und Bild 8.3.18 gezeigt.

Zusammenfassend kann man sagen, daß sich zwei parallele B-Links in einem Schalter, bei dem zwei Sender und zwei Empfänger über durchgängige Ringe angeschlossen sind, nicht lohnen. Je nach Lokalität der Kommunikation, liegt der erreichbare Durchsatz zwischen 275 MB/s bei 0% Lokalität und 682 MB/s bei 100% Lokalität, was einer Zunahme um 55% bzw. 0% gegenüber dem Mono-B-Link-Schalter entspricht. In der Praxis ist die 2-Sender Konfiguration allerdings auch nicht von Interesse. Bei einem Vier-Port-Schalter will man in der Regel nicht nur zwei Ports nutzen.

Vier Sender, durchgehende Ringe und multiple B-Links

Bei voller Bestückung mit vier Sendern und vier Empfängern, ist der Fall n=N identisch zu der Konfiguration mit 2 Sendern und Empfängern, d.h., ein weiteres B-Link bringt keine Durchsatzerhöhung. Der Durchsatz stagniert bei 1365 MB/s, was allerdings ein sehr hoher Wert ist. Für n=0 resultieren zwei B-Links

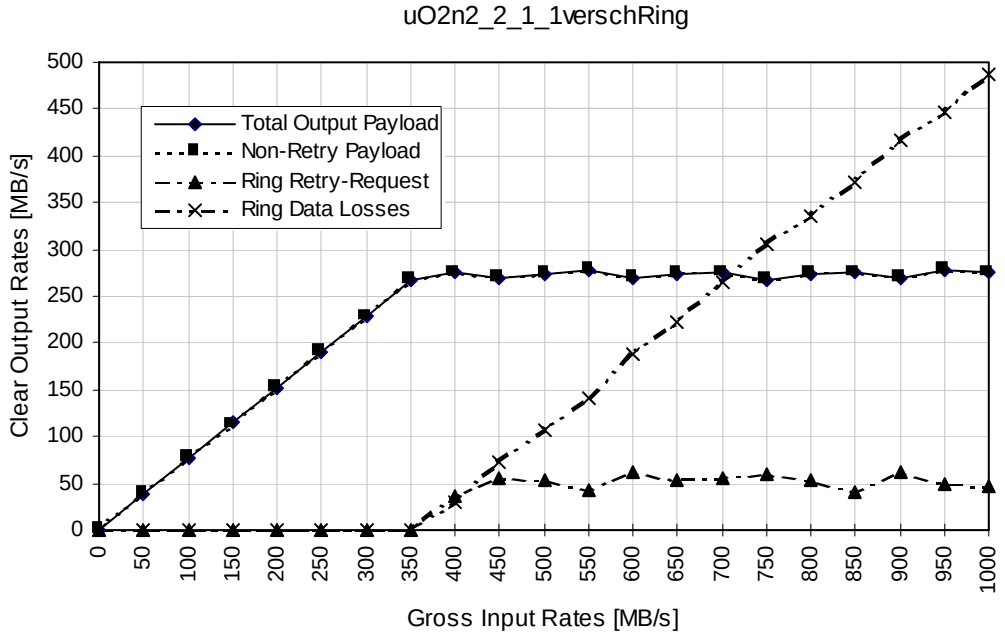

Bild 8.3.17: Durchsatz des Nicht-Ringlet-Schalters bei 2 Sendern, 2 B-Links und n=0.

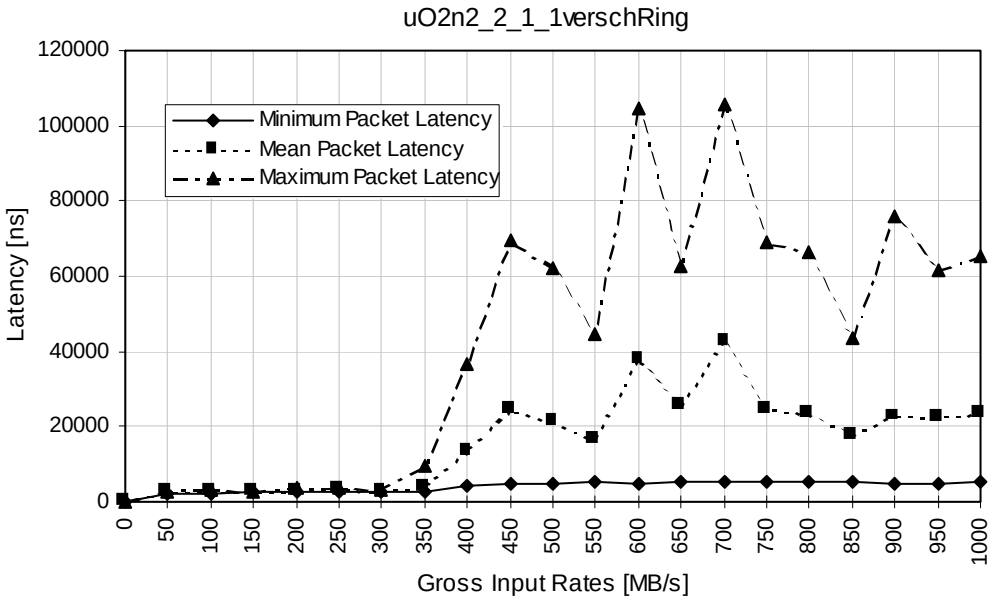

Bild 8.3.18: Latenz des Nicht-Ringlet-Schalters bei 2 Sendern, 2 B-Links und n=0.

in 347 MB/s Durchsatz bei den Empfängern, was ungefähr einer Verdopplung gegenüber einem Mono-B-Link-Schalter gleichkommt. Der Retry-Verkehr hat sich von 839 MB/s auf 399 MB/s praktisch halbiert, und der Sättigungspunkt verschiebt sich auf ca. 500 MB/s. Dieser Sachverhalt ist in Bild 8.3.19 graphisch dargestellt.

Vor der Sättigung hat man eine deterministische Latenz von maximal 3906

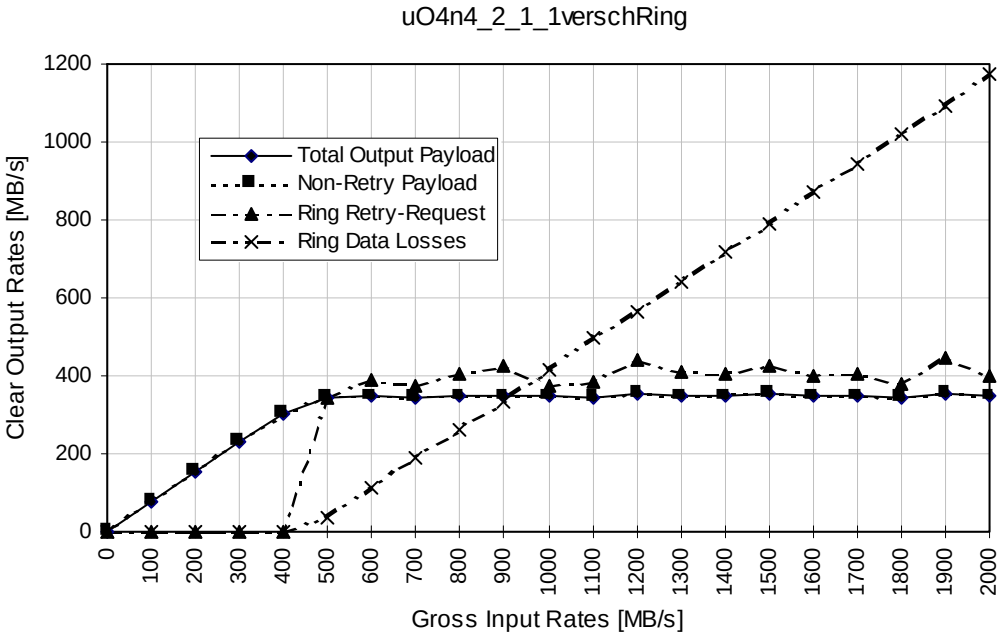

Bild 8.3.19: Durchsatz des Nicht-Ringlet-Schalters bei 4 Sendern, 2 B-Links und n=0.

ns für ein NWRITE64-Paket, was ungefähr der gleiche Wert wie bei einem Mono-B-Link-Schalter ist. Erfreulicherweise steigt die Latenz nach Erreichen des Sättigungspunkts beim Übergang von einem auf zwei B-Links nicht an, sondern geht um 48% von 130316 ns auf 68623 ns zurück, jeweils Maximalwerte (Bild 8.3.20).

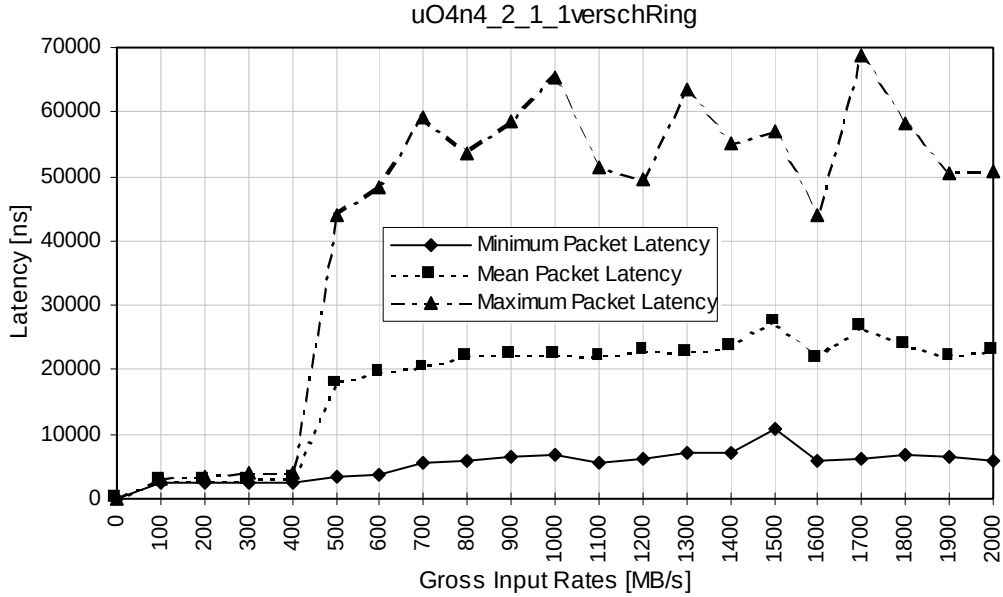

Bild 8.3.20: Latenz des Nicht-Ringlet-Schalters bei 4 Sendern, 2 B-Links und n=0.

Zusammenfassend kann man sagen, daß an einem Vier-Port-SCI-Schalter mit voller Sender/Empfänger-Bestückung und durchgehenden Ringen ein zweites B-Link den Durchsatz auf 347 MB/s verdoppelt, sofern der Verkehr keine Datenlokalität aufweist. Bei 100% Datenlokalität wird der doppelte Durchsatz auch ohne zweites B-Links erreicht. Die Latenz ist unterhalb des Sättigungspunkts von 200 MB/s mit 3906 ns gegenüber dem Mono-B-Link-Schalter mit durchgängigen Ringen leicht erhöht, jedoch oberhalb der Sättigung um 48% reduziert.

Durchgehende Ringe und drei bzw. vier B-Links

Erwartungsgemäß ist für n=N der Durchsatz auch bei drei oder vier B-Links mit 1333 MB/s nahezu unverändert auf hohem Niveau. Die Latenzen sind bis hin zum Sättigungspunkt nur leicht erhöht (1779 ns bei vier B-Links). Überraschenderweise sind jedoch oberhalb der Sättigung die extremen Zunahmen in den Latenzen verschwunden. Als Maximalwert ergeben sich jetzt nur noch 2678 ns bei drei B-Links bzw. 5067 ns bei vier B-Links. Das deutlich bessere Latenzverhalten bei n=N ist in Bild 8.3.21 für vier B-Links gezeigt.

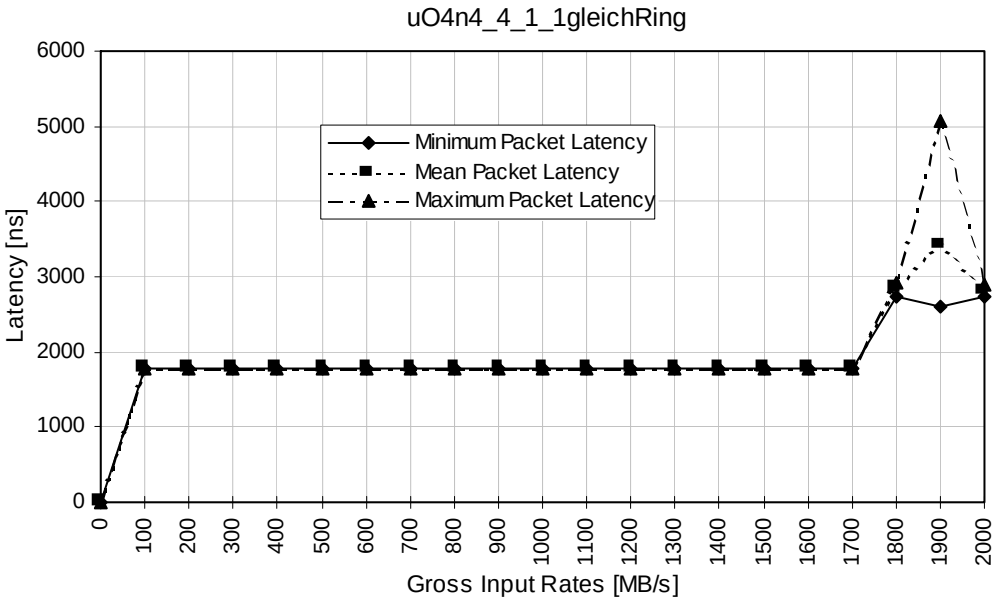

Bild 8.3.21: Latenz des Nicht-Ringlet-Schalters bei 4 Sendern, 4 B-Links und n=N.

Für n=0, d.h. bei 0% Datenlokalität, erhält man bei drei B-Links eine Zunahme des Durchsatzes um den Faktor 2,95 auf 523 MB/s (bei 2000 MB/s Eingangsdatenrate). Der Sättigungspunkt liegt jetzt bei ca. 700 MB/s, und der Retry-Verkehr hat sich gegenüber einem einzelnen B-Link um den Faktor 8,7 auf 96 MB/s reduziert (Bild 8.3.22). Aus dem geringen Retry-Verkehr, der bei drei B-Links noch auftritt, kann man schließen, daß die Balance zwischen den summierten

Port-Bandbreiten und der internen Transferkapazität erreicht ist (unter der Randbedingung, daß NWRITE64-Pakete transferiert werden).

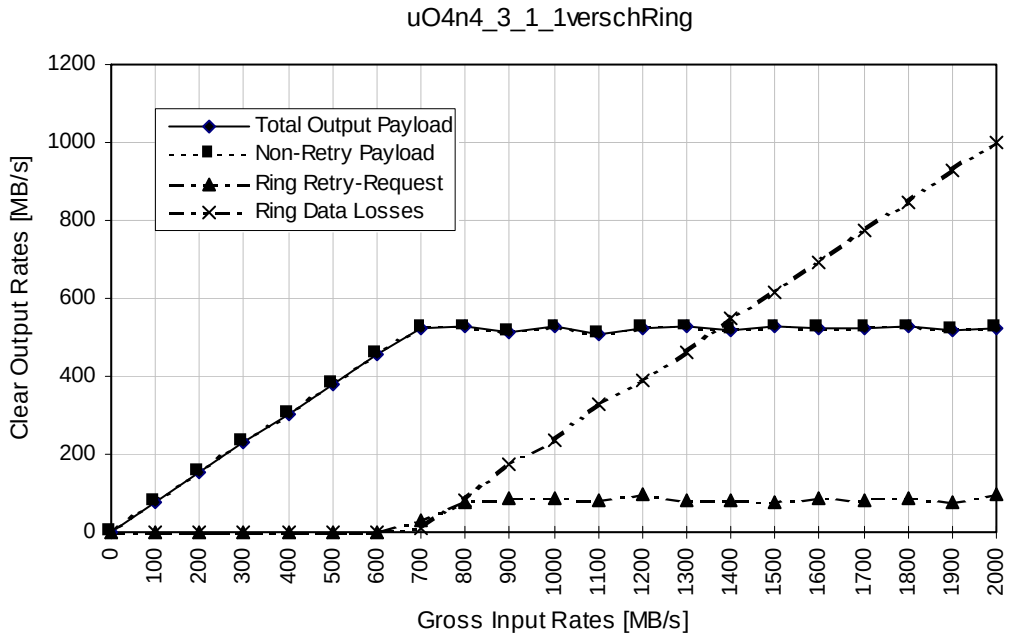

Bild 8.3.22: Durchsatz des Nicht-Ringlet-Schalters bei 4 Sendern, 3 B-Links und n=0.

Die Latenz ist oberhalb des Sättigungspunkts bei 600 MB/s von ursprünglich 110093 ns beim Mono-B-Link-Schalter auf 66659 ns gefallen, was einem Rückgang auf 60% entspricht. Bis zur Sättigung hat man eine deterministische Latenz von ca. 6442 ns. Dies ist höher als zuvor, jedoch haben die Fluktuationen abgenommen. Das Diagramm der Latenzen ist in Bild 8.3.23 gezeigt.
Der Einbau von vier B-Links bringt bei n=0 mit 557 MB/s Nettodurchsatz bei 2000 MB/s Eingangsrate gegenüber der Lösung mit drei B-Links kaum mehr einen Zuwachs, während die Kosten um 1/3 Drittel steigen. Die Latenz bleibt bis zum Sättigungspunkt auf demselben Niveau wie bei drei B-Links, oberhalb der Sättigung rechtfertigt die Abnahme um 3108 ns auf 38994 ns kein viertes B-Link.

Ergebnis:

Zusammenfassend kann man sagen, daß bei Schaltern mit durchgängigen Ringen für den Fall von 100% Datenlokalität Durchsatz und Latenz nicht von der Zahl der B-Links abhängen, vielmehr verbleiben sie auf sehr gutem Niveau (1333 MB/s Durchsatz bzw. 1779 ns Latenz bei vier B-Links). Bei 0% Datenlokalität. hingegen nimmt der Durchsatz proportional zur Zahl der eingesetzten B-Links zu, während die Latenz ab dem Sättigungspunkt in gleichem Maße abnimmt (557 MB/s Durchsatz bzw. maximal 38994 ns Latenz bei vier B-Links). Unterhalb der Sättigung steigt die Latenz mit zunehmender B-Link-Zahl leicht

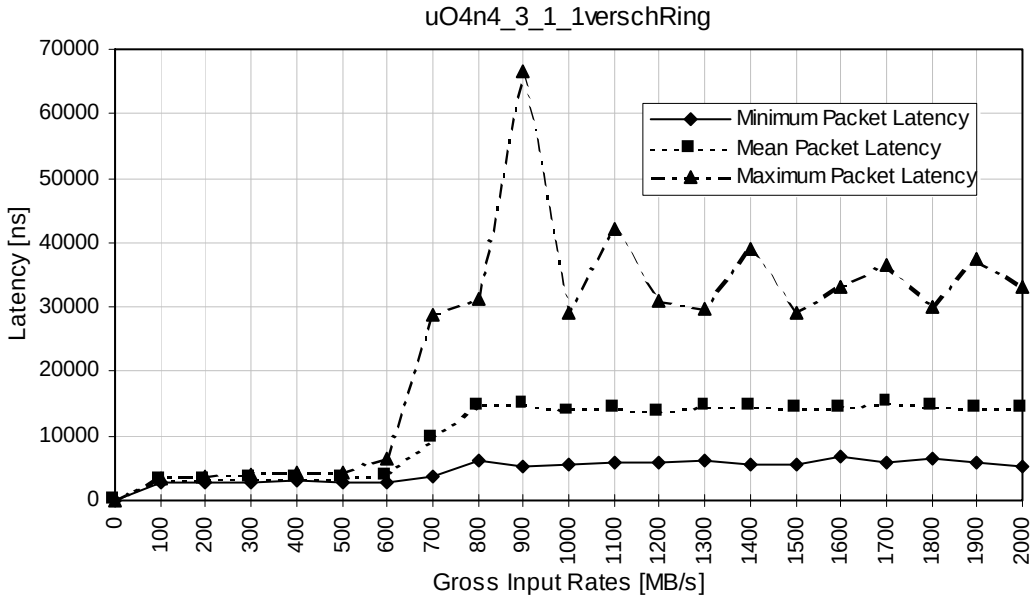

Bild 8.3.23: Latenz des Nicht-Ringlet-Schalters bei 4 Sendern, 3 B-Links und n=0.

an (6930 ns bei vier B-Links). Dafür verantwortlich sind zum einen die akkumulierten Bypass-Fifo-Durchlaufzeiten und zum anderen der erhöhte Scheduling-Aufwand bei mehrfachen B-Links. Das beste Preis/Leistungsverhältnis liegt genau wie bei den Ringlet-gekoppelten Schaltern bei drei B-Links. Der Unterschied zwischen 0 und 100% Datenlokalität beträgt bei den Durchsätzen 1:2,6 und bei der Latenz 1:3,9, oberhalb der Sättigung 1:7,7 (jeweils bei vier B-Links). Ohne multiple B-Links sind die Unterschiede in den Leistungsdaten zwischen 0 und 100% Datenlokalität größer. Deshalb ist die Kombination von durchgängigen Ringen und multiplen B-Links zur Steigerung der Schalterleistung zu empfehlen.

9 Analyse von SCI-Banyan-Netzen

9.1 Einleitung

In diesem Kapitel werden die klassischen logN-Netze, die im ersten Teil der Ausarbeitung beschrieben worden sind, hinsichtlich ihres Durchsatzes, ihrer Paketverluste und Latenzzeiten bei zugrundeliegender SCI-Technologie analysiert. Zum besseren Vergleich werden zunächst diejenigen Netze beurteilt, die auf Schaltern mit Ringlet-Anschlüssen beruhen, da dies die konventioneller Art der Anwendung von SCI bei Banyans darstellt. Im Anschluß daran wird gezeigt, welche Verbesserungen möglich sind. Dazu werden Netze eingeführt, die

auf Schaltern mit durchgängigen Ringen und multiplen B-Links sowie auf Topologien mit einer höheren Permutationsbasis als zwei beruhen.

Aufgrund des möglicher Einsatzes der Netze bei Echtzeitanwendungen wird Augenmerk darauf gelegt, ab welcher Eingangsrate Paketverluste erstmalig auftreten. Der Punkt der ersten Verluste spielt eine größere Rolle als der früher betrachtete Sättigungspunkt, bei dem der maximale Netzdurchsatz erreicht wird, da bei Echtzeitanwendungen Daten mit einer vorgegebenen Ankunftsrate empfangen und weitergeleitet werden müssen. Zusätzlich muß eine bestimmte Fehlerquote garantiert werden. Die Fehlerquote wiederum entspricht bei SCI den Paketverlusten, da die sichere Übertragung einmal in das Netz eingespeister Daten aufgrund der CRC-Prüfsummen und Protokolle gewährleistet ist. Paketverluste treten bereits vor der Einspeisung in das Netz an der Schnittstelle zu SCI auf, sobald die Datenrate zu hoch wird.

Die Untersuchungen beruhen auf Simulationen, die, sofern nicht anders vermerkt, bei Netzen der Größe 16x16 durchgeführt wurden, und bei denen Sender, die z.B. Meßaufnahmesensoren sein können, Daten mit deterministischer Rate in das Netz einspeisen. Die Zuordnung von Sendern zu Empfängern auf der anderen Seite des Netzes erfolgt dabei zufällig und wird vor Beginn einer Simulation mit Hilfe eines Zufallszahlengenerators vorgenommen. Das heißt, daß während der Simulation jedem Sender genau ein Empfänger fest zugeordnet ist. Dies wird auch als „Distributed Target“-Methode bezeichnet. Des weiteren hat jeder Sender dieselbe Datenrate.

Die mit den Banyan-Topologien durchgeführten Simulationen verwenden die Timing-Daten von LC2-basierten Link-Controller-Bausteinen mit 500 MB/s Link- und 600 MB/s B-Link-Geschwindigkeit. Aus diesen Bausteinen werden Vier-Port-Schalter aufgebaut, die je nach Netztopologie miteinander verbunden sind.

9.2 Banyan-Netze mit Ringlet-Schaltern

9.2.1 Omega-Netz

In Bild 9.2.1 ist die Topologie eines Omega-Netzes in Ringlet-Struktur exemplarisch dargestellt. Die Leistungsanalyse dieses Netzes bestehend aus Mono-B-Link-Schaltern ist in Bild 9.2.2 zu sehen. Hier sind in einem Diagramm Durchsatz, Paketverluste und der Maximalwert der Latenz aufgezeichnet. Die Simulation zeigt, daß das Netz bzgl. des Durchsatzes dasselbe Sättigungsverhalten aufweist, wie die Schalter, aus denen es besteht: nach einer linear ansteigenden Geraden geht der Durchsatz in die Waagrechte über. Der Punkt, bei dem die ersten Paketverluste in Höhe von 52 MB/s auftreten, liegt bei 1200 MB/s akkumulierter Eingangsdatenrate. Die Sättigung des Netzes wird bei 2000 MB/s erreicht. Dabei erhält man einen Durchsatz von 997 MB/s bei 524 MB/s an

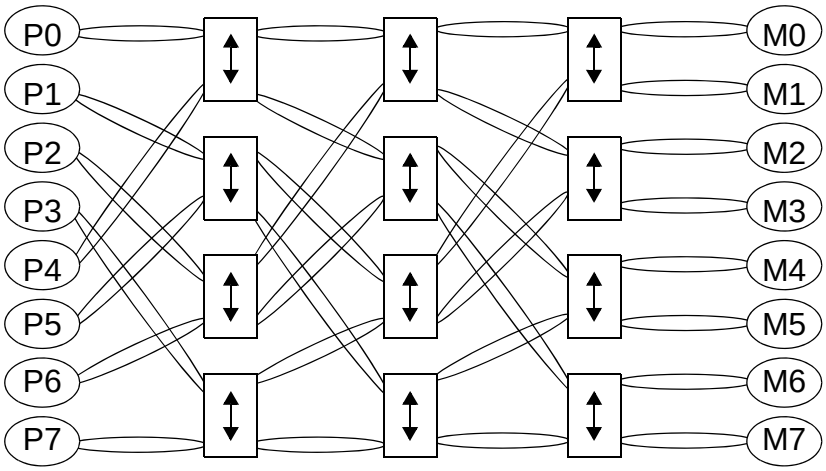

Bild 9.2.1: Omega-Netz der Größe 8x8 in Ringlet-Struktur.

Paketverlusten. Der Maximalwert der Latenz beträgt am Sättigungspunkt 359 µs und erhöht sich bei 8000 MB/s Eingangsrate auf 391 µs. Dazwischen verläuft der Maximalwert ungefähr konstant. Die im Bild nicht dargestellten Mittelwerte der Latenz liegen mit 47 µs am Sättigungspunkt und 53 µs bei 8000 MB/s um fast eine Zehnerpotenz darunter und verlaufen dazwischen ebenfalls nahezu gleichförmig. Aufgrund der Konsistenzprüfung, die sich in Erfüllung der Gleichung (64/84)*2000=997+524 wiederspiegelt, kann man schließen, daß die Simulation verläßlich ist.

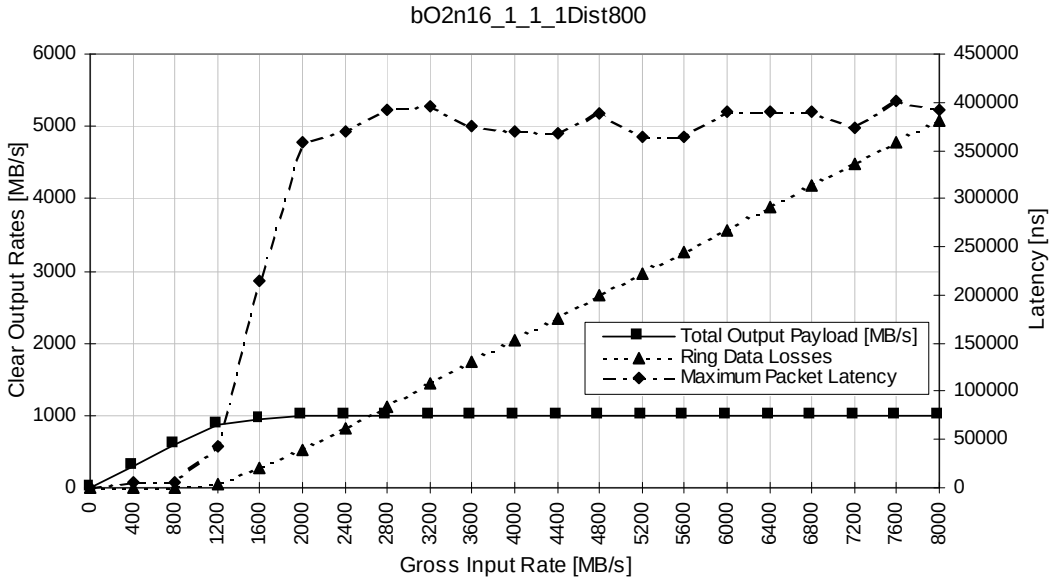

Bild 9.2.2: Leistungsanalyse des Ringlet-Omega-Netzes bei Mono-B-Link-Schaltern.

Insgesamt hat das Omega-Netz in Ringlet-Anschlußweise im Vergleich zum

Verkehr, der an seinen Eingängen angelegt werden kann, einen eher schlechten Durchsatz. Von netto maximal 6095 MB/s angebotenem Verkehr werden nur ca. 1/6 zu den Ausgängen übertragen, der Rest geht bei der Einspeisung ins Netz verloren. Die Latenzen liegen mit 391 µs rel. hoch, obwohl das Netz aus nur 4 Stufen aufgebaut ist.

9.2.2 Flip-Netz

Der Aufbau eines Ringlet-basierten Flip-Netzes ist in Bild 9.2.3 exemplarisch für den Fall von 16 Ein- und Ausgängen dargestellt. Wie bereits im ersten Teil der Ausarbeitung erläutert, handelt es sich topologisch um ein gespiegeltes Omega-Netz, das funktional zu diesem identisch ist. Entsprechend sollten sich auch die Durchsätze gleich verhalten. Die Leistungsanalyse zeigt, daß der

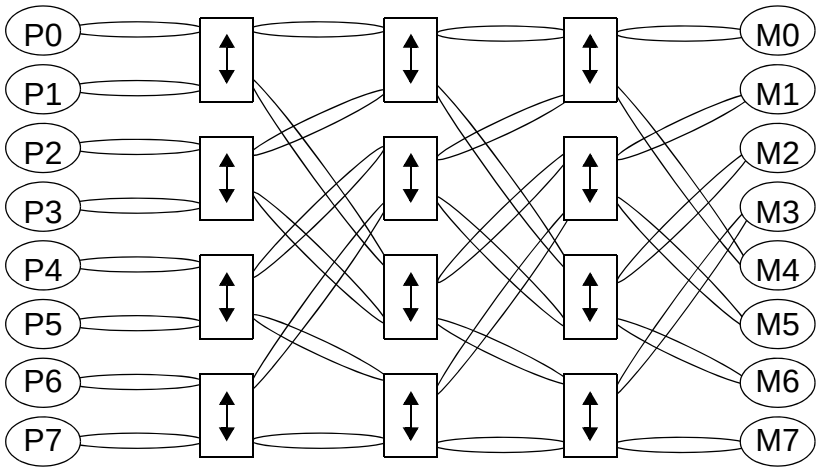

Bild 9.2.3: Flip-Netz der Größe 8x8 in Ringlet-Struktur.

Durchsatz des Flip-Netzes leicht unter dem des Omega-Netzes liegt. Beispielsweise werden bei 1200 MB/s Eingangsrate nur 810 statt 862 MB/s an Nettodurchsatz erzielt, so daß die Paketverluste entsprechend um 52 MB/s auf 104 MB/s ansteigen. Bei 8000 MB/s Eingangsrate hat man 924 MB/s an Durchsatz, also um etwa 8% weniger als beim Omega-Netz. Demgegenüber ist die maximale Latenz mit 357 µs und 383 µs bei 1200 bzw. 8000 MB/s Datenraten um geringfügig erniedrigt.

Zusammenfassend kann also gesagt werden, daß das Flip-Netz gegenüber dem Omega-Netz hinsichtlich des Durchsatzes etwas schlechter und bzgl. der Latenz ungefähr gleich abschneidet. Das Omega-Netz ist somit gegenüber dem Flip-Netz zu bevorzugen.

Generalized Cube- und Indirect Binary n-Cube-Netz

Das Generalized Cube- und das Indirect Binary-n-Cube-Netz sind topologisch und funktional identisch zum Omega- bzw. Flip-Netz. Simulationen zeigen, daß auch deren Durchsätze, Paketverluste und Latenzen 1:1 identisch zu ihren Pendants sind. Diese Netze brauchen deshalb im weiteren nicht mehr betrachtet zu werden.

9.3 Durchsatzerhöhung im Netz

Eine Durchsatzerhöhung im Netz kann analog wie bei einem einzelnen Schalter dadurch bewirkt werden, daß man entweder von der Ringlet-Konfiguration zu durchgängigen Ringen übergeht oder daß man in jedem Schalter multiple B-Links vorsieht. Zunächst soll der Zuwachs im Durchsatz aufgrund von parallelen B-Links untersucht werden.

9.3.1 Ringlet-Netze mit multiplen B-Links

Stellvertretend für die Kategorie der klassischen log-N-Netze wird im folgenden eine Leistungsanalyse der Omega-Topologie durchgeführt. Dazu wird ein Ringlet-gekoppeltes Omega-Netz untersucht, bei dem als schalterinterne B-Link-Arbitrierung das Round Robin-Zuteilungsverfahren Verwendung findet. Das Ergebnis der Simulationen ist in Bild 9.3.1 dargestellt. Daraus geht hervor, daß der Punkt der ersten Paketverluste in Höhe von 87 MB/s sich auf 2000 MB/s akkumulierter Eingangsdatenrate hinausgeschoben hat. Er liegt damit um 2/3 höher als beim Netz aus Mono-B-Link-Schaltern. Die Parallelschaltung zweier B-Links ist also erfolgreich. Allerdings gehorcht jetzt die Kurve des Durchsatzes nicht mehr dem einfachen Zwei-Geraden-Modell bestehend aus einem linearen Anstieg und einer Waagrechten, vielmehr ähnelt der Verlauf mehr einer e-Funktion. Aus diesem Grunde kann ein Sättigungspunkt nicht mehr eindeutig ermittelt werden. Der höchste Durchsatz wird mit 1821 MB/s bei 6000 MB/s Eingangsdatenrate erreicht, was um 80% höher als beim Netz aus Mono-B-Link-Schaltern liegt. Bei dieser Rate hat man 2748 MB/s Paketverluste. Daraus folgt für die Konsistenzprüfung, daß (64/84)*6000 = 1821+2748 gelten muß. (Was auch zutrifft).

Die Latenz beträgt bei 2000 MB/s maximal 85 µs und erhöht sich bei 8000 MB/s akkumulierter Eingangsdatenrate auf 335 µs. Insgesamt weist die Latenzkurve einen anderen Verlauf als beim Mono-B-Link-Netz auf und zeigt niedrigere Werte.

Von besonderem Interesse ist, daß wie beim Mono-B-Link-Netz die Latenzkurve früher als der Durchsatz ihren Knickpunkt erreicht. Daraus kann man

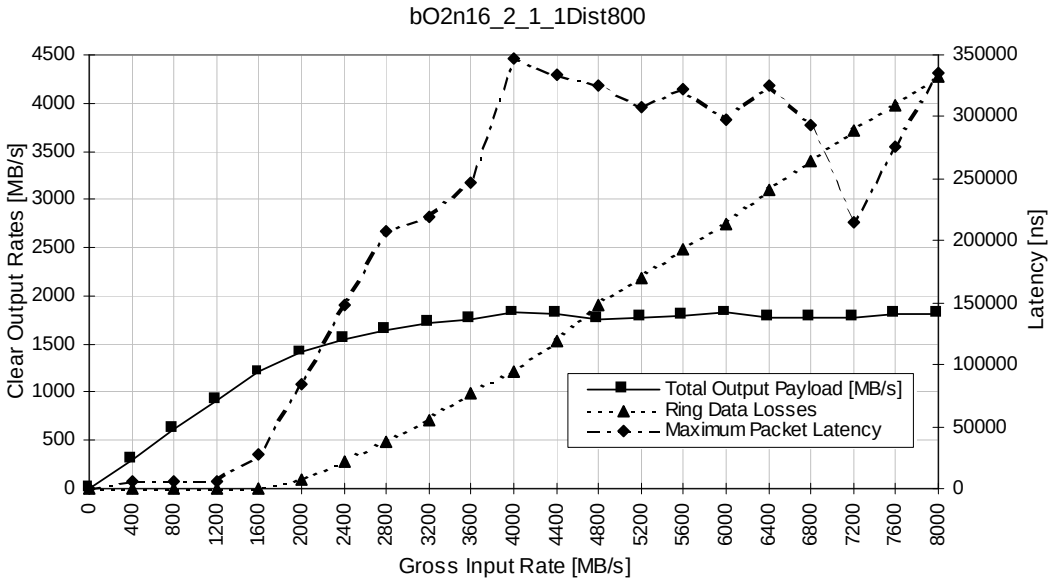

Bild 9.3.1: Leistungsanalyse des Ringlet-Omega-Netzes bei Dual-B-Link-Schaltern.

schließen, daß der nichtlineare Anstieg in der Latenz quasi die bevorstehende Sättigung des Durchsatzes ankündet. Dies läßt sich damit erklären, daß zuerst die Pakete bei ihrem Weg durch das Netz immer länger benötigen, bevor schließlich der maximal mögliche Paketstrom erreicht ist.

Insgesamt läßt sich sagen, das beim Omega-Netz aus Dual-B-Link-Schaltern der Punkt der ersten Paketverluste um 2/3 höher als beim Netz aus Mono-B-Link-Schaltern liegt. Der Durchsatz hat sich um 80% erhöht, während die Latenz um 14% zurückging.

9.3.2 Implementierung in Silizium

Die Implementierung der beschriebenen Zusatzfunktionen zur Leistungssteigerung von SCI-Ringen, Schaltern und Netzen erfordert bei den bestehenden kommerziellen Produkten wie dem Dolphinschen Link-Controller einige zusätzliche Registerbits und Zusatzlogik, die eine geringe Komplexitätserhöhung im Silizium-Layout des Chips bedeuten. Die Registerbits werden zusammen mit den bereits vorhandenen Konfigurationsbits des Link-Controller-Bausteins vom Master des jeweiligen SCI-Rings vor Beginn des Betriebs in einer Initialisierungsphase gesetzt. In Einzelnen ist je ein Registerbit notwendig für:

- Das Einschalten eines Retry-Delays, um den Retry-Verkehr des betreffenden Knotens zu reduzieren. Die Retry-Verzögerung sollte dabei adaptiv mit exponentiell ansteigender Zeit sein.
- Das Ausschalten der Dekodierung der niederwertigsten 2 SCI-Adreßbits, so daß sich bei einer gegebenen Adresse eine Gruppe von bis zu vier Knoten gleichzeitig angesprochen fühlt. Die Adressierung von Knotengruppen dient

der Kaskadierung von Link Chips zu einem Multi-B-Link-Schalter. Mit der Aktivierung dieses Bits wird die auch adaptive B-Link-Auswahl aktiviert.

- Das Einschalten der adaptiven Paketannahme (=MustTake-Option) für Netze mit Pfadkompensation.
- Die Zusatzinformation „IamTheLast“, die für die MustTake-Option erforderlich ist.
- Das Einschalten der adaptiven Schalterausgangsauswahl (für Netze mit Pfadkompensation).

Schließlich ist für das Round Robin Scheduling multipler B-Links in jedem Link-Controller-Baustein ein ladbarer 2-Bit-Zähler erforderlich, der bei Eintreffen eines Pakets dekrementiert wird und der sich nach einem Nulldurchgang selbstständig auf einen Wert zwischen 0 und 3 setzt. Der Wert der Zählerinitialisierung ist in zwei weiteren Registerbits pro LC-Baustein abzulegen, ein drittes Registerbit dient zur Aktivierung des Zählers.