



(19)
Bundesrepublik Deutschland
Deutsches Patent- und Markenamt

(10) **DE 698 32 884 T2** 2006.08.24

(12) **Übersetzung der europäischen Patentschrift**

(97) **EP 1 005 745 B1**

(51) Int Cl.⁸: **H04L 12/56** (2006.01)

(21) Deutsches Aktenzeichen: **698 32 884.1**

(86) PCT-Aktenzeichen: **PCT/US98/16762**

(96) Europäisches Aktenzeichen: **98 940 865.3**

(87) PCT-Veröffentlichungs-Nr.: **WO 1999/011033**

(86) PCT-Anmeldetag: **20.08.1998**

(87) Veröffentlichungstag
der PCT-Anmeldung: **04.03.1999**

(97) Erstveröffentlichung durch das EPA: **07.06.2000**

(97) Veröffentlichungstag
der Patenterteilung beim EPA: **21.12.2005**

(47) Veröffentlichungstag im Patentblatt: **24.08.2006**

(30) Unionspriorität:

918556	22.08.1997	US
84636	26.05.1998	US

(84) Benannte Vertragsstaaten:

**AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT,
LI, LU, MC, NL, PT, SE**

(73) Patentinhaber:

Avici Systems, Billerica, Mass., US

(72) Erfinder:

**DALLY, J., William, Stanford, US; CARVEY, P.,
Philip, Bedford, US; DENNISON, R., Larry,
Norwood, US; KING, Allen, P., Needham, US**

(74) Vertreter:

**Patentanwälte Westphal Mussgnug & Partner,
78048 Villingen-Schwenningen**

(54) Bezeichnung: **WEGESUCHEINHEIT MIT ZUTEILUNG VON VIRTUELLEN KANÄLEN**

Anmerkung: Innerhalb von neun Monaten nach der Bekanntmachung des Hinweises auf die Erteilung des europäischen Patents kann jedermann beim Europäischen Patentamt gegen das erteilte europäische Patent Einspruch einlegen. Der Einspruch ist schriftlich einzureichen und zu begründen. Er gilt erst als eingelegt, wenn die Einspruchsgebühr entrichtet worden ist (Art. 99 (1) Europäisches Patentübereinkommen).

Die Übersetzung ist gemäß Artikel II § 3 Abs. 1 IntPatÜG 1991 vom Patentinhaber eingereicht worden. Sie wurde vom Deutschen Patent- und Markenamt inhaltlich nicht geprüft.

Beschreibung

Hintergrund der Erfindung

[0001] Datenkommunikation zwischen Computersystemen für Anwendungen wie z. B. Webbrowsing, E-Mail, Dateitransfer und elektronischer Handel wird oft unter Verwendung einer Familie von Protokollen durchgeführt, die als IP (Internet Protocol) oder manchmal TCP/IP bekannt sind. Da Anwendungen immer beliebter werden, die erhebliche Datenkommunikation verwenden, wachsen die Verkehrsanforderungen an das IP-Basisnetzwerk bzw. Backbone IP Netzwerk exponentiell an. Es wird erwartet, dass IP-Router mit mehreren hundert Anschlüssen bzw. Ports, die mit einer Gesamtbandbreite von Terabits pro Sekunde arbeiten, innerhalb der nächsten Jahre benötigt werden, um das Wachstum bei der Backbone-Nachfrage aufrechtzuerhalten.

[0002] Wie in [Fig. 1](#) dargestellt, ist das Internet als eine Hierarchie von Netzwerken aufgebaut. Ein typischer Endverbraucher besitzt eine Arbeitsstation **22**, die mit einem lokalen Netzwerk oder LAN **24** verbunden ist. Um den Benutzern des LANs zu gestatten, auf den Rest des Internets zuzugreifen, ist das LAN über einen Router R mit einem regionalen Netzwerk **26** verbunden, das von einem regionalen Netzbetreiber oder RNP (Regional Network Provider) gewartet und betrieben wird. Die Verbindung wird oft über einen Internet Service Provider oder ISP durchgeführt. Um auf andere Regionen zuzugreifen, verbindet sich das regionale Netzwerk mit dem Backbone Netzwerk **28** an einem Netzzugangsknoten bzw. Network Access Point (NAP). Die NAPs befinden sich üblicherweise lediglich in größeren Städten.

[0003] Das Netzwerk besteht aus Übertragungsabschnitten und Routern. Im Netzwerk Backbone sind die Übertragungsabschnitte üblicherweise faseroptische Kommunikationskanäle, die unter Verwendung des SONET (Synchronous Optical Network) Protokolls betrieben werden. SONET Übertragungsabschnitte arbeiten bei einer Vielzahl von Datenraten, die von OC-3 (155 Mbit/s) bis OC-192 (9,9 Gbit/s) reichen. Diese Übertragungsabschnitte, manchmal Fernleitungsstränge bzw. Trunks genannt, bewegen Daten von einem Punkt zu einem anderen, oft über erhebliche Distanzen.

[0004] Router verbinden eine Gruppe von Übertragungsabschnitten miteinander und führen zwei Funktionen aus: Weiterleiten und Routen. Ein Datenpaket, das an einem Übertragungsabschnitt eines Routers ankommt, wird weitergeleitet, indem man es an einem anderen Übertragungsabschnitt aussendet, je nach seinem letztendlichen Ziel und dem Status der Ausgangsübertragungsabschnitte. Um den Ausgangsübertragungsabschnitt für ein gegebenes Paket zu berechnen, nimmt der Router an einem Rou-

ting-Protokoll teil, wo sämtliche Router im Internet Informationen über die Verbindungsfähigkeit des Netzes austauschen und basierend auf diesen Informationen Routing-Tabellen berechnen.

[0005] Die meisten Internet-Router des Standes der Technik basieren auf einem gemeinsamen Bus ([Fig. 2](#)) oder einem Kreuzschienenschalter bzw. Crossbar Switch ([Fig. 3](#)). Bei dem Bus-basierten Schalter aus [Fig. 2](#) ist z. B. ein gegebener SONET Übertragungsabschnitt **30** mit einem Leitungsschnittstellenmodul **32** verbunden. Dieses Modul extrahiert die Pakete aus dem ankommenden SONET Datenstrom. Für jedes ankommende Paket liest die Leitungsschnittstelle den Paketkopfsatz bzw. Paketheader und bestimmt unter Verwendung dieser Information den Ausgangsanschluss (oder -anschlüsse), zu dem/denen das Paket weitergeleitet werden soll. Um das Paket weiterzuleiten, vermittelt das Leitungsschnittstellenmodul für den gemeinsamen Bus **34** per Arbitration. Wenn der Bus zugeteilt wird, wird das Paket über den Bus zum Ausgangsleitungsschnittstellenmodul übertragen. Das Modul überträgt nachfolgend das Paket an einem abgehenden SONET Übertragungsabschnitt **30** zum nächsten Sprung bzw. Hop auf dem Weg zu seinem Ziel.

[0006] Bus-basierte Router weisen eine begrenzte Bandbreite und Skalierbarkeit auf. Der zentrale Bus wird zu einem Flaschenhals, durch den der gesamte Verkehr fließen muss. Ein sehr schneller Bus arbeitet z. B. mit einem 128-Bit breiten Datenpfad bei 50 MHz, woraus sich eine Gesamtbandbreite von 6,4 GBit/s ergibt, viel zu wenig für die Terabits pro Sekunde, die von einem Backbone Switch benötigt werden. Auch begrenzen die Auffächerungseinschränkungen der Bus-Schnittstellen die Anzahl von Anschlüssen an einen Bus-basierten Switch auf typischerweise nicht mehr als **32**.

[0007] Die Bandbreitenbegrenzung eines Busses kann durch Verwendung eines Crossbar Switch, wie in [Fig. 3](#) dargestellt, überwunden werden. Für N Leitungsschnittstellen **36** enthält der Switch N (N - 1) Kreuzungspunkte, die jeweils mit einem Kreis markiert sind. Jede Leitungsschnittstelle kann eine beliebige der anderen Leitungsschnittstellen als ihren Eingang durch Verbinden der beiden Leitungen auswählen, die sich an dem entsprechenden Kreuzungspunkt **38** treffen. Um ein Paket innerhalb dieser Organisation weiterzuleiten, vermittelt eine Leitungsschnittstelle für die angeforderte Ausgangsleitungsschnittstelle. Wenn die Anfrage gewährt wird, wird der entsprechende Kreuzungspunkt geschlossen und die Daten werden von dem Eingangsmodul zum Ausgangsmodul übertragen. Da die Kreuzschiene gleichzeitig viele Eingänge und viele Ausgänge miteinander verbinden kann, stellt diese Struktur ein Vielfaches der Bandbreite eines Bus-basierten Switches bereit.

[0008] Trotz ihrer erhöhten Bandbreite fehlt es den Kreuzschienenbasierten Routern immer noch an der Skalierbarkeit und der Bandbreite, die für einen IP-Backbone Router erforderlich ist. Das durch die Kreuzschienenverbindung erforderliche Auffächern und Einfächern, wobei jeder Eingang mit jedem Ausgang verbunden wird, begrenzt die Anzahl der Anschlüsse auf typischerweise nicht mehr als **32**. Diese begrenzte Skalierbarkeit resultiert auch in einer begrenzten Bandbreite. Z. B. könnte eine Kreuzschiene des Standes der Technik gleichzeitig **32** 32-Bit-Kanäle bei 200 MHz betreiben, woraus sich eine Spitzenbandbreite von 200 GBit/s ergibt. Dies ist immer noch weniger als die Bandbreite, die von einem Backbone IP-Router verlangt wird.

[0009] Parulkar et al., Computer Communications Review, Band 25, Nr. 4, 1995, S. 49–58 offenbart einen Router, der unter Verwendung eines ATM-Switches und von Hauptrechner-Schnittstellenkomponenten gestaltet ist, der einen Träger bildet, der einen Satz von IP-Verarbeitungselementen (IPPE) verbindet. IPPEs erledigen die IP-Paketverarbeitung und steuern den ATM-Träger direkt. IP-Datagramme werden über eine ATM-virtuelle Schaltung an einen IPPE gesendet, der das Datagramm zu einem ausgewählten Ausgangsanschluss weiterleitet.

[0010] US 5,583,990 offenbart eine mehrdimensionale Verbindungs- und Routereinrichtung für einen Computer mit paralleler Verarbeitung, der Verarbeitungselemente in einer dreidimensionalen Struktur miteinander verbindet. Die Verbindungs- und Routereinrichtung umfasst eine Vielzahl von Verarbeitungselementknoten. Eine Kommunikationsverbindung verbindet mindestens eines der Verarbeitungselemente mit einem Hostsystem. Ein Verbindungsnetz verbindet die Verarbeitungselementknoten in einer X-, Y- und Z-Dimension miteinander. Das Netzwerk umfasst Kommunikationswege, die jedes der Vielzahl von Verarbeitungselementen mit angrenzenden Verarbeitungselementen in den Plus- und Minusrichtungen jeder der X-, Y- und Z-Dimensionen verbinden.

Zusammenfassung der Erfindung

[0011] Demgemäß stellt die vorliegende Erfindung einen Internet-Router zum Koppeln an eine Vielzahl von Internet-Übertragungsabschnitten bereit, wobei der Router Datenpakete von den Internet-Übertragungsabschnitten erhält, Header-Informationen in den Datenpaketen analysiert, um die Datenpakete zu routen, und wobei der Router die Datenpakete auf den Internet-Übertragungsabschnitten weiterleitet, wobei der Router eine Schaltstruktur von Strukturverbindungsabschnitten aufweist, die durch Strukturknoten verbunden sind, wobei die Anzahl der Strukturverbindungsabschnitte zu jedem Strukturknoten kleiner als die Anzahl der vom Internet-Router bedienten Internet-Übertragungsabschnitte ist, wobei die Struk-

turverbindungsabschnitte und -knoten eine Datenkommunikation zwischen den Internet-Übertragungsabschnitten über einen oder mehrere Sprünge bzw. Hops durch die Struktur bereitstellen; dadurch gekennzeichnet, dass die Strukturknoten Struktur-Router sind, die für eine Zwischenspeicher-Arbitrationslogik zwischen konkurrierenden Paketen sorgen.

[0012] In einem weiteren Aspekt stellt die vorliegende Erfindung ein Verfahren zum Routen von Datenpaketen zwischen Internet-Übertragungsabschnitten bereit, bestehend aus: Analysieren von Header-Informationen in den Datenpaketen, um die Datenpakete zu Ausgangs-Internet-Übertragungsabschnitten zu routen; und Durchleiten der Datenpakete durch ein Multihop-Strukturnetzwerk von Strukturknoten hin zu den Ausgangs-Internet-Übertragungsabschnitten; dadurch gekennzeichnet, dass die Strukturknoten Struktur-Router sind, die die Zwischenspeicher unter den konkurrierenden Paketen mittels Arbitrationslogik vermitteln.

[0013] Obwohl sie eine begrenzte Bandbreite und Skalierbarkeit aufweisen, haben Kreuzschienenbasierte Router zwei wünschenswerte Merkmale:

1. Sie sind nicht blockierend (non-blocking). So lange keine zwei Eingänge anfragen, mit dem gleichen Ausgang zu kommunizieren, können alle Eingänge gleichzeitig mit ihren angefragten Ausgängen verbunden werden. Sobald ein Ausgang verstopft wird, beeinträchtigt der Verkehr zu diesem Ausgang nicht den Verkehr, der an andere Ausgänge gerichtet ist.
2. Sie sorgen für starken Gegendruck. Die direkte Verbindung zwischen der Quelle und dem Ziel über die Kreuzschiene umfasst üblicherweise einen Rückkanal, der für eine unmittelbare Flusssteuerung verwendet werden kann. Dieser Gegendruck kann z. B. von einem überlasteten Ziel verwendet werden, um einer Quelle zu signalisieren, keine Daten mehr zu senden.

[0014] Um die Routing-Anforderungen für das Internet-Backbone zu erfüllen, würden wir gerne dies bei den Eigenschaften bewahren, während gleichzeitig eine um Größenordnung höhere Bandbreite und Skalierbarkeit bereitgestellt wird.

[0015] Gemäß eines Aspekts der vorliegenden Erfindung erhält man Vorteile der Kreuzschienenbasierten Internet-Router mit größerer Bandbreite und Skalierbarkeit durch Einrichten des Routers selbst als ein Multihop-Netzwerk.

[0016] Ein die Erfindung verkörpernder Internet-Router empfängt Datenpakete von einer Vielzahl von Internet-Übertragungsabschnitten und analysiert die Header-Informationen in den Datenpaketen, um die Datenpakete zu Ausgangs-Internet-Übertragungsabschnitten zu routen. Der Internet-Router

weist eine Struktur aus Strukturverbindungsabschnitten auf, die durch Struktur-Router verbunden sind, wobei die Anzahl der Strukturverbindungsabschnitte zu jedem Struktur-Router wesentlich geringer ist als die Anzahl der Internet-Verbindungsabschnitte, die von dem Internet-Router bedient werden. Die Strukturverbindungsabschnitte und Struktur-Router sorgen für eine Datenkommunikation zwischen Internet-Verbindungsabschnitten über einen oder mehrere Sprünge bzw. Hops durch die Struktur. In einer Ausführungsform werden z. B. 600 Internet-Verbindungsabschnitte von einem $6 \times 10 \times 10$ dreidimensionalen ringförmigen Struktur bedient.

[0017] Durch Bereitstellen einer Vielzahl von Zwischenspeichern in jedem Struktur-Router können virtuelle Kanäle definiert werden, die sich Strukturausgabeverbindungsabschnitte teilen. Die virtuellen Kanäle und Verbindungsabschnitte bilden ein virtuelles Netzwerk zwischen den Internet-Routereingängen und -ausgängen, wobei eine Verstopfung in einem virtuellen Netzwerk im Wesentlichen nicht blockierend für den Datenfluss durch andere virtuelle Netzwerke wirkt. Eine Leitungsschnittstelle zu jedem Internet-Verbindungsabschnitt analysiert die Header-Informationen in den Datenpaketen, die von dem Internet-Verbindungsabschnitt empfangen werden, um die Ausgangsinternet-Verbindungsabschnitte über ein Internet Routing Protokoll zu identifizieren. Die Leitungsschnittstelle bestimmt des Weiteren über ein Struktur-Routing-Protokoll eine Routingstrecke durch die Struktur zu dem identifizierten Ausgangsinternetverbindungsabschnitt. Die Pakete können in Segmente oder Flits (Flow Control Digits) an der Leitungsschnittstelle unterteilt werden, und die Segmente werden durch die Struktur unter Verwendung des Wormhole Routings weitergeleitet. Die Leitungsschnittstelle kann die Routingstrecke durch die Struktur mittels Einschließen einer Verbindungsabschnittsdefinition jedes nachfolgenden Verbindungsabschnitts in die Routingstrecke in einem Header definieren. Jeder Struktur-Router entlang der Routingstrecke speichert eine zugehörige Verbindungsabschnittsdefinition aus dem Header zum Weiterleiten von nachfolgenden Segmenten des Pakets.

[0018] Vorzugsweise werden zwischen den Sprüngen bzw. Hops auf den Strukturverbindungsabschnitten Flits in den Struktur- Routern an Speicherorte gespeichert, die virtuellen Kanälen zugeordnet sind, welche viel Internet-Verbindungsabschnitten entsprechen. In einer Ausführungsform wird der Satz von Ziel-Internet-Verbindungsabschnitten in disjunkte Untermengen unterteilt, und jeder virtuelle Kanal wird exklusiv einer Untermenge von Ziel-Internet-Verbindungsabschnitten zugeordnet. In bevorzugten Ausführungsformen beträgt die Anzahl an Internet-Verbindungsabschnitten, die von dem Internet-Router bedient werden, mindestens eine Größenordnung mehr als die Anzahl der Struktur-Verbin-

dungsabschnitte zu jedem Struktur-Router, und die Anzahl an virtuellen Kanälen pro Struktur-Router ist wesentlich größer als die Anzahl der Verbindungsabschnitte zu dem Struktur-Router.

[0019] Um virtuelle Kanäle unter den Datenpaketen und Struktur-Verbindungsabschnitte unter den virtuellen Kanälen gemeinsam zu benutzen, wird eine Arbitrationslogik bei jedem Struktur-Router durchgeführt, um ein Paket einem virtuellen Kanal zur Ausgabe aus dem Struktur-Router zuzuordnen und um einen virtuellen Kanal einem Ausgangsstruktur-Verbindungsabschnitt von dem Struktur-Router zuzuordnen. Zur Flusssteuerung wird ein virtueller Kanal beim Empfang einer Anzeige, dass ein Eingangszwischenspeicher am entgegengesetzten Ende des Verbindungsabschnitts verfügbar ist, zur möglichen Zuordnung zu einem Ausgangsstruktur-Verbindungsabschnitt berechtigt.

[0020] Eine weitere Ausführungsform ist auf Internet-Router und andere Netzwerk-Router einschließlich Multicomputern anwendbar, sie betrifft eine ereignisgesteuerte Technik zur Handhabung von virtuellen Kanälen in einem Router, der Datenpakete routet. Der Router umfasst physikalische Eingabekanäle, die Teile der Datenpakete empfangen, physikalische Ausgabekanäle und Datenzwischenspeicher, die mit den physikalischen Eingabe- und Ausgabekanälen gekoppelt sind. Die Datenzwischenspeicher speichern die Teile der Datenpakete. Der Router umfasst des Weiteren Steuerschaltkreise, die mit den physikalischen Eingabe- und Ausgabekanälen und den Datenzwischenspeichern gekoppelt sind. Die Steuerschaltkreise erzeugen Kanaluordnungen als Reaktion auf anstehende Ereignisse, und gibt die Teile der Datenpakete über die physikalischen Ausgabekanäle aus entsprechend der erzeugten Kanaluordnungen. Vorzugsweise ordnen die Steuerschaltkreise virtuelle Kanäle zu den Datenpaketen zu, ordnet den physikalischen Ausgabekanälen als Reaktion auf die anstehenden Ereignisse den virtuellen Kanälen zu. In einer Ausführungsform umfasst der Router des Weiteren eine Leitungsschnittstelle, die mit einem physikalischen Eingabekanal und einem physikalischen Ausgabekanal gekoppelt ist, so dass der Router einen Internet-Schaltstruktur-Router bildet. In einer weiteren Ausführungsform umfasst der Router des Weiteren eine Multicomputer-Schnittstelle, die mit einem physikalischen Eingabekanal und einem physikalischen Ausgabekanal gekoppelt ist, so dass der Router einen Multicomputer-Router für ein Multicomputersystem bildet.

[0021] Gemäß der bevorzugten Ausführungsform umfassen die Steuerschaltkreise Ausgangssteuer-einrichtungen, die den physikalischen Ausgabekanälen entsprechen. Jede Ausgangssteuer-einrichtung weist eine Statustabelle auf, die die Status der virtuellen Ausgabekanäle aufzeichnet und virtuelle Einga-

bekannäle identifiziert, die mit den virtuellen Ausgabekanälen verbunden sind. Die virtuellen Eingabekanäle enthalten die Teile der Datenpakete.

[0022] Jede Ausgangssteuereinrichtung weist des Weiteren einen Arbitrator auf, der geeignet ist, Ankunftsereignisse aus mehreren Ankunfts Warteschlangen auszuwählen, sowie eine Zustandstabellelogik, die auf diese Zustandstabelle der Ausgangssteuereinrichtung zugreift, um als Reaktion auf die ausgewählten Ankunftsereignisse virtuelle Ausgabekanäle zuzuordnen. Jede Zustandstabelle umfasst Zustandsvektoren, die den virtuellen Ausgabekanälen entsprechen.

[0023] Jeder Zustandsvektor umfasst eine Beleg-Anzeige, die anzeigt, ob der den Zustandsvektor entsprechende virtuelle Ausgabekanal zu einem Datenpaket zugeordnet ist. Zusätzlich weist jeder Zustandsvektor ein Wartefeld auf, das anzeigt, welche der physikalischen Eingabekanäle zumindest Teile von Datenpaketen empfangen haben, die auf die Zuordnung zu dem diesem Zustandsvektor entsprechenden virtuellen Ausgabekanal warten. Jedes Wartefeld zeigt darüber hinaus eine Reihenfolge an, in welcher die physikalischen Eingabekanäle die Datenpaketteile empfangen haben. Jeder Zustandsvektor umfasst des Weiteren ein Vorhanden-Feld, das eine Anzahl von Teilen eines Datenpakets anzeigt, die zum Übertragen durch den virtuellen Ausgabekanal dieses Zustandsvektors hin zu einem stromab gelegenen Router vorhanden sind. Weiterhin umfasst jeder Zustandsvektor ein Guthaben- bzw. Credit-Feld, das eine Menge an verfügbarem Zwischenspeicherplatz in einem stromab gelegenen Router anzeigt, der mit dem diesem Zustandsvektor entsprechenden virtuellen Ausgabekanal gekoppelt ist.

[0024] Jede Ausgangssteuereinrichtung umfasst des Weiteren eine Transportschaltung, die Transportanforderungen aneinanderreicht, wenn auf die Zustandstabelle dieser Ausgangssteuereinrichtung als Reaktion auf die aneinander gereihten Ereignisse zugegriffen wird, und die Datenpakete durch den physikalischen Ausgabekanal dieser Ausgangssteuereinrichtung gemäß der aneinander gereihten Transportanfragen weiterleitet. Die Teile der Datenpakete sind Flits der Datenpakete. Jede Transportschaltung überträgt ein Flit als Reaktion auf eine wartende Transportanfrage.

[0025] Jede Ausgangssteuereinrichtung empfängt Guthaben- bzw. Credit-Ereignisse von einem stromab gelegenen Router und reiht eine Transportanfrage in die Warteschlange, als Reaktion auf die empfangenen Guthaben- bzw. Credit-Ereignisse, um einen Teil des Datenpaketes über den entsprechenden physikalischen Ausgabekanal zu übertragen. In einer Ausführungsform umfassen die in Warteschlange gereihten Ereignisse Endpunkt-Guthaben- bzw. Credit-Er-

eignisse, und die Ausgangssteuereinrichtungen geben virtuelle Kanäle nur als Reaktion auf die Endpunkt-Guthaben- bzw. Credit-Ereignisse frei.

[0026] Die Steuerschaltkreise können von mehreren virtuellen Kanälen gemeinsam benutzt werden und aktiviert werden, um einen bestimmten virtuellen Kanal als Reaktion auf ein Ereignis zu behandeln.

[0027] Vorzugsweise sind die Steuerschaltkreise geeignet, virtuelle Kanaluordnungen zu erzeugen, die virtuelle Kanäle zu den Datenpaketen zuordnen, und physikalische Kanaluordnungen zu erzeugen, die die physikalischen Ausgabekanäle zu den virtuellen Kanälen zuordnen. Jede der Zuordnungen als Reaktion auf in Warteschlange aneinander gereihten Ankunfts- und Guthaben- bzw. Credit-Ereignisse erzeugt werden. Die Teile der Datenpakete werden von den Datenzwischenspeichern zu den physikalischen Ausgabekanälen gemäß der erzeugten virtuellen und physikalischen Kanaluordnungen weitergeleitet.

[0028] In der hauptsächlichen Implementierung ist jeder Zielknoten einem virtuellen Netzwerk zugeordnet, das in jedem Router einen eindeutigen virtuellen Kanal aufweist, um einen kontinuierlichen Fluss von Datenpaketen trotz der Verstopfung zu einigen Zielen zu gewährleisten. Ein beliebiges Datenpaket, das für einen bestimmten Ausgang bestimmt ist, wird in einem Datenzwischenspeicher gespeichert, der einem eindeutigen virtuellen Kanal für dieses Ziel entspricht, und so blockiert eine Stauung, die ein Datenpaket blockiert, nicht notwendigerweise ein weiteres.

[0029] Dieser Ansatz ist hinsichtlich des Zwischenspeicherplatzes teuer und hat eine begrenzte Skalierbarkeit. Gemäß eines weiteren Aspekts der vorliegenden Erfindung, der auf beliebige Router ohne die Einschränkung auf Struktur-Router angewendet werden kann, werden virtuelle Netzwerke erstellt mit sich überlappenden Zwischenspeicherzuordnungen durch das gemeinsame Verwenden von virtuellen Kanälen. Um jedoch zu verhindern, dass eine Überlastung bzw. Stauung in einem virtuellen Netzwerk Übertragungen in einem anderen virtuellen Netzwerk blockiert, teilen sich virtuelle Netze weniger als die Gesamtheit der virtuellen Kanäle.

[0030] Gemäß dieser Ausführungsform weist ein Router zum Routen von Datenpaketen physikalische Eingabekanäle, physikalische Ausgabekanäle und Zwischenspeicher auf, die mit den physikalischen Eingabe- und Ausgabekanälen gekoppelt sind, zum Speichern von mindestens Teilen der Datenpakete. Steuerschaltkreise, die mit den physikalischen Eingabe- und Ausgabekanälen und den Datenzwischenspeichern gekoppelt sind, erzeugen Zuordnungen der Datenpakete zu virtuellen Kanälen, die auf physikalischen Kanälen gebündelt werden (Multiplex). Ein erstes Datenpaket hat Zugang zu einem ersten Satz

von virtuellen Kanälen der Größe C_1 , und ein zweites Datenpaket hat Zugang zu einem zweiten Satz von virtuellen Kanälen der Größe C_2 . Es gibt eine Schnittmenge der ersten und zweiten Sätze der Größe S , wobei $0 < S < C_1$ und $S < C_2$ gilt. Im Ergebnis können Datenpakete den Zugang zu mehreren virtuellen Kanälen gemeinsam verwenden, zu denen andere Datenpakete Zugang haben, wobei ein erstes Datenpaket einige, jedoch nicht sämtliche virtuelle Kanäle mit einem zweiten Datenpaket gemeinsam benutzt.

[0031] Vorzugsweise werden im Wesentlichen alle Datenpakete über eine Vielzahl von virtuellen Netzen geroutet, wobei jedes virtuelle Netzwerk i eine Untermenge der Größe C_i der virtuellen Kanäle verwendet. Die Untermengen an virtuellen Kanälen von mehreren virtuellen Netzen überlappen sich, wobei sie nicht mehr als $S < C_i$ virtuelle Kanäle gemeinsam benutzen.

[0032] In einem bevorzugten Router wird das erste Paket auf einem ersten virtuellen Netzwerk und das zweite Paket auf einem zweiten virtuellen Netzwerk geroutet. Alle auf dem ersten Netzwerk geroutete Pakete benutzen den ersten Satz an virtuellen Kanälen gemeinsam und alle Pakete, die auf dem zweiten virtuellen Netzwerk geroutet werden, benutzen den zweiten Satz von virtuellen Kanälen gemeinsam.

[0033] Vorteilhafterweise überträgt jedes virtuelle Netzwerk Pakete, die für einen bestimmten Satz von Zielknoten bestimmt sind, wie z. B. ein Paar von in einem Netzwerk einander gegenüberliegenden Knoten. Mindestens einer der virtuellen Kanäle in jedem Satz von virtuellen Kanälen kann durch eine Zielknotenadresse bestimmt werden, insbesondere durch eine dimensionale Koordinate des Zielknotens.

[0034] Um eine interdimensionelle Blockierung zu verhindern, können bestimmte Drehungen (Turns) in jedem virtuellen Netzwerk gesteuert werden. In einer Ausführungsform werden zwei virtuelle Netze jedem Satz von Zielknoten zugeordnet und unterschiedliche Drehungen werden in jedem der beiden virtuellen Netze gesperrt. In einer anderen Ausführungsform wird ein einziges virtuelles Netzwerk jedem Satz von Zielknoten zugeordnet. Mehrere Drehungen können in diesem einzelnen virtuellen Netzwerk gesperrt werden.

[0035] In einer bevorzugten Ausführungsform umfassen die Steuerschaltkreise Ausgangssteuereinrichtungen, die den physikalischen Ausgabekanälen entsprechen. Jede Ausgangssteuereinrichtung weist eine Zustandstabelle auf, die die Zustände der virtuellen Ausgabekanäle aufzeichnet und virtuelle Eingabekanäle identifiziert, die die Teile der Datenpakete enthalten. Eine Zustandstabilenlogik greift auf die Zustandstabelle zu, um virtuelle Ausgabekanäle zuzuordnen. Die Zustandstabelle kann einen Virtualka-

nal-Allokationsvektor für jedes virtuelle Netzwerk und einen Belegt-Vektor umfassen, der virtuelle Kanäle anzeigt, die in dem physikalischen Ausgabekanal belegt sind. Ein virtueller Ausgabekanal wird aus einer Kombination des Virtualkanal-Allokationsvektors und dem Belegt-Vektor ausgewählt.

[0036] Eine Anwendung des Routers ist als ein Router innerhalb eines Multicomputernetzes, und eine weitere Anwendung ist als ein Struktur-Router innerhalb eines Internet-Paketrouters.

Kurze Beschreibung der Figuren

[0037] Das Vorstehende und andere Ziele, Merkmale und Vorteile der Erfindung werden anhand der nachfolgenden genaueren Beschreibung von bevorzugten Ausführungsformen der Erfindung deutlich, wie in den beigefügten Zeichnungen veranschaulicht, in denen gleiche Bezugszeichen sich auf dieselben Teile über die unterschiedlichen Ansichten hinweg beziehen. Die Figuren sind nicht notwendigerweise maßstabsgetreu, die Betonung liegt stattdessen auf der Veranschaulichung der Prinzipien der Erfindung.

[0038] [Fig. 1](#) stellt eine Internetkonfiguration von Routern dar, auf die die vorliegende Erfindung angewendet werden kann.

[0039] [Fig. 2](#) ist ein Bus-basierter Internet-Router des Standes der Technik.

[0040] [Fig. 3](#) ist ein Crossbar Switch Internet-Router des Standes der Technik.

[0041] [Fig. 4](#) veranschaulicht ein zweidimensionales ringförmiges Array, das früher in direkten Multiprozessor-Netzwerken verwendet wurde.

[0042] [Fig. 5](#) veranschaulicht ein indirektes Netzwerk.

[0043] [Fig. 6](#) stellt eine Baumausnutzung eines Netzes dar.

[0044] [Fig. 7](#) stellt eine dreidimensionale Struktur dar, die die vorliegende Erfindung verkörpert.

[0045] [Fig. 8](#) veranschaulicht das Leitungsschnittstellenmodul eines Knotens im Array aus [Fig. 7](#).

[0046] [Fig. 9](#) veranschaulicht einen Struktur-Router, der in der Ausführungsform der [Fig. 7](#) und [Fig. 8](#) verwendet wird.

[0047] [Fig. 10A](#) und [Fig. 10B](#) veranschaulichen Zwischenspeicher, Register und Steuer-Vektoren, die im Router aus [Fig. 9](#) verwendet werden.

[0048] [Fig. 11A](#) und [Fig. 11B](#) stellen eine alternati-

ve Allokationssteuerlogik dar, die in den Eingabe- bzw. Ausgangssteuerinrichtungen des Routers aus [Fig. 9](#) vorgesehen sind.

[0049] [Fig. 12](#) stellt eine Zustandstabelle von virtuellen Kanälen dar, die im Router aus [Fig. 9](#) verwendet wird.

[0050] [Fig. 13](#) veranschaulicht eine Schleife, die verwendet wird, um ein Dispersionsrouting zu zeigen.

[0051] [Fig. 14](#) stellt Drehungen (Turns) dar, die erforderlich sind, um ein Ziel von jedem Quadrant um das Ziel herum zu erreichen.

[0052] [Fig. 15](#) veranschaulicht einen Virtuale Kanal-Auswahlschaltkreis, der die Erfindung verkörpert.

Detaillierte Beschreibung der Erfindung

[0053] Bei der Implementierung eines Internet-Routers nimmt die vorliegende Erfindung Anleihen aus der Multiprozessor-Technologie und ändert diese Technologie ab, um die einzigartigen Eigenschaften und Anforderungen von Internet-Routern zu erfüllen. Insbesondere ist jeder Internet-Router selbst entweder als ein direktes oder indirektes Netzwerk konfiguriert.

[0054] Multicomputer und Multiprozessoren verwenden seit vielen Jahren direkte und indirekte Verbindungsnetzwerke, um Adressen und Daten für Speicherzugriffe zwischen Prozessoren und Speicherbänken zu senden oder um Nachrichten zwischen Prozessoren zu senden. Die frühen Multicomputer waren aufgebaut unter der Verwendung der Bus- und Kreuzschienen-Verbindungen, die in den [Fig. 2](#) und [Fig. 3](#) dargestellt sind. Um jedoch diesen Maschinen zu ermöglichen, auf größere Anzahlen an Prozessoren zu skalieren, wechselten sie zur Verwendung von direkten und indirekten Verbindungsnetzwerken.

[0055] Ein direktes Netzwerk, wie in [Fig. 4](#) veranschaulicht, setzt sich aus einem Satz von Verarbeitungsknoten **40** zusammen, wobei jeder einen Router R zusammen mit einem Prozessor P und einem gewissen Speicher M umfasst. Diese Multicomputer-Router sollten nicht mit den oben beschriebenen IP-Routern verwechselt werden. Sie führen lediglich Weiterleitungsfunktionen durch und nur in der sehr eingeschränkten Umgebung eines Multicomputer-Verbindungsnetzwerks. Jeder Multicomputer-Router weist eine gewisse Anzahl, vier in dem Beispiel, von Verbindungen zu anderen Routern im Netzwerk auf. Ein Verarbeitungsknoten kann eine Nachricht senden oder einen Speicherzugriff auf irgendeinen beliebig anderen Knoten in dem System durchführen. Er ist nicht darauf beschränkt, nur mit

den unmittelbar angrenzenden Knoten zu kommunizieren. Nachrichten zu den Knoten, die weiter weg liegen, werden durch die Router entlang der Strecke zwischen den Quell- und Zielknoten weitergeleitet.

[0056] Das in [Fig. 4](#) dargestellte Netzwerk wird als direkt bezeichnet, da die Kanäle direkt zwischen den Verarbeitungsknoten des Systems verlaufen. Im Gegensatz dazu zeigt [Fig. 5](#) ein indirektes Netzwerk, bei dem die Verbindungen zwischen den Prozessknoten **42** indirekt hergestellt sind, über einen Satz von Nur-Router-Schaltknoten **44**. Direkte Netzwerke werden im Allgemeinen für große Maschinen aufgrund der Skalierbarkeit bevorzugt. Während ein indirektes Netzwerk üblicherweise für eine feste Anzahl von Knoten gebaut ist, wächst ein direktes Netzwerk mit den Knoten. Sobald mehr Knoten hinzugefügt werden, wird auch mehr Netzwerk hinzugefügt, da ein kleiner Teil des Netzwerks, ein Router, innerhalb jedes Knotens umfasst ist.

[0057] Multicomputer-Netzwerke sind im Detail beschrieben in Dally, W. J. "Network and Processor Architectures for Message-Driven Computing", VLSI and Parallel Computation, herausgegeben von Suaya und Birtwistle, Morgan Kaufmann Publishers, Inc., 1990, S. 140–218. Es sollte hervorgehoben werden, dass Multicomputer-Netzwerke lokal auf einen einzelnen Schaltschrank oder einen einzelnen Raum begrenzt sind, im Gegensatz zum Internet-Backbone-Netz, das sich über einen Kontinent hinweg erstreckt.

[0058] Direkte und indirekte Multicomputer-Netzwerke sind skalierbar. Für die meisten allgemeinen Topologien ist die Einfächerung (Fan-in) und Auffächerung (Fan-out) jedes Knotens konstant, unabhängig von der Größe der Maschine. Auch die Verkehrslast auf jedem Verbindungsabschnitt ist entweder konstant oder eine sehr langsam wachsende Funktion der Maschinengröße. Aufgrund dieser Skalierbarkeit wurden diese Netzwerke erfolgreich verwendet, um Parallelcomputer mit tausenden von Verarbeitungsknoten aufzubauen.

[0059] Obwohl Multicomputer-Netzwerke skalierbar sind, geben sie unglücklicherweise die beiden Eigenschaften von Crossbar-Networks auf, die für das IP-Switching entscheidend sind: die Eigenschaft des Nicht-Blockierens und den festen Gegendruck. Die meisten ökonomischen direkten und indirekten Netzwerke sind blockierend. Da die Verbindungsabschnitte unter mehreren Quelle-Ziel-Paaren gemeinsam verwendet werden, kann eine belegte Verbindung zwischen einem Knotenpaar die Errichtung einer neuen Verbindung zwischen einem komplett anderen Knotenpaar blockieren. Da Pakete in Multicomputernetzwerken über mehrfache Verbindungsabschnitte mit erheblichen Warteschlangen (queuing) bei jedem Verbindungsabschnitt weitergeleitet werden, ist

der Gegendruck, sofern vorhanden, von einem überlasteten Zielknoten zu einem übertragenden Quellknoten spät und weich, sofern überhaupt vorhanden.

[0060] Die Blockiereigenschaft dieser Switches und die weiche Eigenschaft dieses Gegendrucks ist kein Problem für einen Multicomputer, da der Multicomputer-Verkehr selbstdrosselnd (selfthrottling) ist. Nachdem ein Prozessor eine geringe Anzahl von Nachrichten oder Speicheranfragen (typischerweise 1–8) gesendet hat, kann er keine weiteren Nachrichten mehr aussenden, bis er eine oder mehrere Antworten erhält. Damit wird, wenn das Netzwerk sich aufgrund des Blockierens oder einer Stauung verlangsamt, der dem Netzwerk angebotene Verkehr automatisch reduziert, da die Prozessoren immer noch Antworten erwarten. Ein IP-Switch ist im Gegensatz dazu nicht selbstdrosselnd. Wenn einige Kanäle im Netzwerk blockiert oder überlastet werden, wird der angebotene Verkehr nicht reduziert. Pakete kommen ungeachtet des Zustands des Netzwerks weiterhin über die Eingangsverbindungsabschnitte zum Switch an. Aufgrund dessen wird ein IP-Switch oder -Router, der aus einem unmodifizierten Multicomputer-Netzwerk aufgebaut ist, wahrscheinlich baumgesättigt, und wird vielen Knoten, die nicht an der ursprünglichen Blockierung beteiligt sind, den Dienst verweigern. Darüber hinaus existieren in IP-Routern oft Übergangszustände, wobei aufgrund eines Fehlers bei der Berechnung von Routing-Tabellen ein einzelner Ausgabeknoten für eine anhaltende Zeitdauer überlastet werden kann. Bei einem Crossbar-Router führt dies nicht zu Problemen, da andere Knoten nicht betroffen sind. Bei einem Multicomputer-Netzwerk erzeugt dies jedoch eine Baumsättigung.

[0061] Man betrachte die Situation, die in [Fig. 6](#) dargestellt ist. Ein einzelner Knoten in einem zweidimensionalen Maschennetzwerk (Knoten 3,3), gekennzeichnet mit a, wird mit ankommenden Nachrichten überlastet. Da er nicht in der Lage ist, Nachrichten aus den Kanälen mit der Geschwindigkeit zu akzeptieren, in der sie ankommen, werden alle vier Eingabekanäle zu dem Knoten (b, a), (c, a), (d, a), (e, a) aufgestaut und sind blockiert. Verkehr, der an den Knoten b–e ankommt, der über diese blockierten Verbindungsabschnitte hinweg weitergeleitet werden muss, kann sich nicht mehr fortbewegen und wird sich entlang der Ränder zu den Knoten b–e aufstauen. Z. B. staut sich der Verkehr zu Knoten b entlang (f, b), (g, b) und (h, b) auf. Wenn die Blockade weiterhin besteht, werden die Kanäle zu f–h und verwandte Knoten ebenfalls blockiert usw. Wenn die Überlastung bei Knoten a fortbesteht, werden schließlich die meisten Kanäle im Netzwerk blockiert, da ein Sättigungsbaum sich vom Knoten a nach außen ausdehnt.

[0062] Das Hauptproblem mit einer Baumsättigung ist, dass sie Verkehr beeinflusst, der nicht für Knoten

a bestimmt ist. Ein Paket von (1, 4) zu (5, 3) kann z. B. entlang einer Strecke (gestrichelte Linie) geroutet werden, die z. B. (f, b) und (b, a) umfasst. Da diese Verbindungsabschnitte blockiert sind, wird Verkehr vom Knoten (1, 4) zum Knoten (5, 3) blockiert, obwohl keiner dieser Knoten überlastet ist.

[0063] Der Router der vorliegenden Erfindung überwindet die Bandbreiten- und Skalierbarkeitseinschränkungen der Bus- und Crossbar-basierten Router des Standes der Technik durch Verwendung eines Multihop-Verbindungsnetzwerks, insbesondere eines dreidimensionalen ring- bzw. torusförmigen Netzwerks als Router. Mit dieser Anordnung enthält jeder Router in dem weit reichenden Backbone-Netzwerk in Wirklichkeit ein kleines sich im Schrank befindendes (in-cabinet) Netzwerk. Um Missverständnisse zu vermeiden, bezeichnen wir das kleine Netzwerk innerhalb jedes Routers als die Schaltstruktur (switching fabric) und die Router und Verbindungsabschnitte innerhalb dieses Netzes als die Struktur-Router (fabric router) und die Struktur-Verbindungsabschnitte (fabric links).

[0064] Anders als Multicomputer-Netzwerke ist das Schaltstrukturnetzwerk nicht blockierend und stellt einen festen Gegendruck bereit. Die kreuzschienenartigen Attribute werden durch Bereitstellen eines separaten virtuellen Netzwerks für jeden Zielknoten im Netzwerk erreicht.

[0065] Typische über das Internet weitergeleitete Pakete liegen im Bereich von 50 B bis 1,5 kB. Zum Transfer durch das Struktur-Netzwerk des Internet-Routers der vorliegenden Erfindung werden die Pakete in Segmente oder Flits von jeweils 36 B unterteilt. Zumindest der in dem ersten Flit eines Pakets enthaltene Header wird zur Steuerung des Datentransfers durch die Struktur des Routers modifiziert. Beim bevorzugten Router werden die Daten durch die Struktur gemäß eines Wormhole-Routingprotokolls übertragen.

[0066] Jedes virtuelle Netzwerk weist einen Satz von Zwischenspeichern auf. Einer oder mehrere Zwischenspeicher für jedes virtuelle Netzwerk werden bei jedem Knoten in der Struktur zur Verfügung gestellt. Jeder Zwischenspeicher ist so dimensioniert, dass er mindestens ein Flow-Control-Digit oder Flit einer Nachricht enthält. Die virtuellen Netzwerke verwenden alle den einzigen Satz an physikalischen Kanälen zwischen den Knoten des echten Strukturnetzwerks gemeinsam. Eine gerechte Arbitrationsstrategie wird verwendet, um die Verwendung der physikalischen Kanäle über die konkurrierenden virtuellen Netze hinweg zu bündeln. Jedes virtuelle Netzwerk weist einen unterschiedlichen Satz an Zwischenspeichern auf, die zum Halten der Flits seiner Nachrichten verfügbar sind.

[0067] Für jedes Paar von virtuellen Netzwerken A und B enthält der Satz an Zwischenspeichern, der A zugeordnet ist, mindestens einen Zwischenspeicher, der nicht B zugeordnet ist. Damit ist A in der Lage, wenn Netzwerk B blockiert ist, voranzukommen, indem es Nachrichten unter Verwendung dieses Zwischenspeichers weiterleitet, der nicht gemeinsam mit B benutzt wird, obwohl er mit einem anderen virtuellen Netzwerk gemeinsam benutzt werden kann.

[0068] Ein einfaches Verfahren zum Aufbau von virtuellen Netzwerken ist es, einen getrennten Flit-Zwischenspeicher, einen virtuellen Kanal auf jeden Knoten für jedes virtuelle Netzwerk und damit für jedes Ziel bereitzustellen. Z. B. würde in einer Maschine mit $N = 512$ Knoten und damit 512 Zielen jeder Knoten 512 distinkte Flit-Zwischenspeicher umfassen. Der Zwischenspeicher i in jedem Knoten wird nur verwendet, um Flits von Nachrichten zu halten, die für den Knoten i bestimmt sind. Diese Zuordnung erfüllt die obigen Einschränkungen deutlich, da jedes virtuelle Netzwerk einem eindeutigen Satz von Zwischenspeichern auf jedem Knoten zugeordnet ist, wobei keine Zwischenspeicher von den virtuellen Netzwerken gemeinsam benutzt werden. Wenn ein einzelnes virtuelles Netzwerk verstopft wird, sind lediglich dessen Zwischenspeicher betroffen und der Verkehr fährt auf den anderen virtuellen Netzwerken ohne Einschränkung fort. Eine Alternative ist der weiter unten diskutierte dispersive Ansatz.

[0069] Der bevorzugte Router ist ein dreidimensionales ringförmiges Netzwerk von Knoten wie in [Fig. 7](#) dargestellt. Jeder Knoten N weist ein Leitungsschnittstellenmodul auf, das eingehende und ausgehende SONET Internetverbindungsabschnitte verbindet. Jeder dieser Leitungsschnittstellenknoten weist einen Schaltstruktur-Router auf, der Struktur-Verbindungsabschnitte zu seinen sechs benachbarten Knoten im Ring umfasst. IP-Pakete, die über einen SONET Verbindungsabschnitt, z. B. am Knoten A, ankommen, werden untersucht, um den SONET Verbindungsabschnitt zu bestimmen, auf dem sie den Internet-Router z. B. Knoten B, verlassen sollen, und werden anschließend von A zu B über die 3-D-Ringschaltstruktur weitergeleitet.

[0070] Der Aufbau jedes Knotens oder Leitungsschnittstellenmoduls ist in [Fig. 8](#) dargestellt. Pakete kommen über den ankommenden SONET Verbindungsabschnitt **46** an und die Leitungsschnittstellenschaltung **48** wandelt die optische Einspeisung in elektrische Signale um und extrahiert die Pakete und deren Header aus dem eingehenden Strom. Ankommende Pakete werden anschließend zur Weiterleitungseinrichtung **50** übergeben und in dem Paketzweischenspeicher **52** gespeichert. Die Weiterleitungseinrichtung verwendet den Header jedes Pakets, um den erforderlichen Ausgangsverbindungsabschnitt für das Paket zu ermitteln. Bei der herkömmlichen IP-Router-

Verarbeitungsweise wird diese Ermittlung mittels Durchlaufen eines Baumes durchgeführt, der von den Header-Feldern indiziert ist. Die Blätter des Baumes enthalten den nötigen Ausgangsverbindungsabschnitt wie in einem herkömmlichen IP-Router und umfassen zusätzlich die Strecke durch die Schaltstruktur zum Ausgangsverbindungsabschnitt. Schließlich werden das Paket zusammen mit seinem Ziel und seiner Strecke an den Struktur-Router **54** des Knotens zum Weiterleiten durch die Struktur zum Ausgabeknoten übergeben. Von dem Struktur-Router **54** des Ausgabeknotens wird das Paket über den Paketzweischenspeicher **52** dieses Knotens und über die Leitungsschnittstellenschaltung **48** zu dem Ausgangsverbindungsabschnitt **56** geliefert. Die Pakete in dem Internet-Router werden von dem Leitungsschnittstellenmodul, das dem Eingangsanschluss zugeordnet ist, zu dem Leitungsschnittstellenmodul, das dem Ausgangsanschluss zugeordnet ist, unter Verwendung von Quellen-Routing (Source-Routing) weitergeleitet. Beim Source-Routing wird die Strecke von Verbindungsabschnitten durch dazwischen liegende Struktur-Router durch ein Nachschauen in einer Tabelle im Eingangsmodul bestimmt. Dieses Nachschauen wird von der Weiterleitungseinrichtung durchgeführt, bevor das Paket dem Struktur-Router überreicht wird. Alternative Wege ermöglichen Fehlertoleranz und Lastausgleich.

[0071] Die Source-Route ist ein Vektor mit zehn Elementen, wobei jedes Element ein 3-Bit Hop-Feld ist. Jedes Hop-Feld codiert den Ausgangsverbindungsabschnitt, der vom Paket zu nehmen ist, für einen Schritt seiner Strecke, einen der sechs Verbindungsabschnitte zwischen den Knoten oder des siebenten Verbindungsabschnitt zu dem Paketzweischenspeicher des vorliegenden Knotens. Die achte Codierung wird nicht verwendet. Dieser Vektor mit zehn Elementen kann verwendet werden, um alle Strecken von bis zu zehn Hops zu codieren, was ausreichend ist, um zwischen einem beliebigen Paar von Knoten in einem $6 \times 10 \times 10$ -Ring zu routen. Es ist anzumerken, dass sämtliche zehn Elemente nicht für kürzere Strecken verwendet werden müssen. Das letzte verwendete Element wählt den Verbindungsabschnitt zu dem Paketzweischenspeicher **52** aus oder kann für eine Strecke mit zehn Sprüngen bzw. Hops eingeschlossen werden.

[0072] Sobald das Paket an jedem Strukturknoten entlang der Strecke ankommt, wird der lokale Weiterleitungsvektoreintrag für dieses Paket gleich dem Element ganz links der Source-Route gesetzt. Die Source-Route wird anschließend drei Bits nach links geschoben, um dieses Element zu verwerfen und das nächste Element der Strecke dem nächsten Router darzulegen. Während dieses Schiebens wird der 3-Bit-Code, der dem Paketzweischenspeicher des vorliegenden Knotens entspricht, von rechts hinein geschoben. Nachfolgende Flits in diesem Paket fol-

gen dem Routing, das für dieses Paket in dem Router gespeichert ist.

[0073] Der kundige Fachmann wird verstehen, dass es viele mögliche Codierungen der Struktur-Route gibt. In einer alternativen Ausführungsform kann die Tatsache ausgenutzt werden, dass Pakete dazu neigen, sich in einer bevorzugten Richtung in jeder Dimension fortzubewegen, um eine kompaktere Codierung der Struktur-Route zu ergeben. In dieser Ausführungsform wird die Strecke als eine 3-Bit bevorzugte Richtung gefolgt von einer Vielzahl von 2-Bit Hop-Feldern codiert. Das 3-Bit Feld codiert die bevorzugte Richtung (entweder positiv oder negativ) für jede Dimension des Netzwerks (x, y und z). Für jeden Schritt oder Hop der Strecke wählt ein 2-Bit Feld die Dimension bzw. Maßangabe aus, für welche der nächste Hop (Sprung) genommen werden muss ($0 = x$, $1 = y$ oder $2 = z$). Die Richtung dieses Hops wird durch das Feld der bevorzugten Richtung bestimmt. Die vierte Codierung des 2-Bit Hop-Feldes (3) wird als Escape-Sequenz verwendet. Wenn ein Hop-Feld eine Escape-Sequenz enthält, wird das nächste Hop-Feld verwendet, um die Strecke zu bestimmen. Wenn dieses zweite Hop-Feld eine Dimensionsangabe (0–2) enthält, ist der Hop in der bestimmten Dimension in der der bevorzugten Richtung entgegen gesetzten Richtung durchzuführen und die bevorzugte Richtung wird umgedreht. Wenn das zweite Hop-Feld eine zweite Escape-Sequenz enthält, wird das Paket zum Ausgangsanschluss des Struktur-Routers weitergeleitet. Mit dieser Codierung wird, sobald Pakete an einem Strukturknoten ankommen, der lokale Weiterleitungsvektoreintrag für dieses Paket aus dem Feld für die bevorzugte Richtung und dem Hop-Feld ganz links berechnet. Die Hop-Felder werden anschließend zwei Bits nach links geschoben, um dieses Feld zu verwerfen und um das nächste Feld dem nächsten Router zu präsentieren. Während dieses Schiebens wird die 2-Bit Escape-Sequenz in das Hop-Feld ganz rechts geschoben. Für Pakete, die sich hauptsächlich in der bevorzugten Richtung fortbewegen, ergibt diese Codierung eine kompaktere Struktur-Route, da lediglich zwei Bits, lieber als zwei, benötigt werden, um jeden Hop (Sprung) der Strecke zu codieren.

[0074] Ein Struktur-Router, der verwendet wird, um ein Paket über die Schaltstruktur von dem Modul, das zu seinem Eingangsverbindungsabschnitt gehört, zu dem Modul weiterzuleiten, das zu seinem Ausgangsverbindungsabschnitt gehört, ist in [Fig. 9](#) dargestellt. Der Router weist sieben Eingangsverbindungsabschnitte **58** und sieben Ausgangsverbindungsabschnitte **60** auf. Sechs der Verbindungsabschnitte verbinden angrenzende Knoten in dem 3-D Ringnetzwerk aus [Fig. 7](#). Der siebte Eingangsverbindungsabschnitt akzeptiert Pakete von der Weiterleitungseinrichtung **50** und der siebte Ausgangsverbindungsabschnitt sendet Pakete zu dem Paketaus-

gangszwischenspeicher **52** in dem Leitungsschnittstellenmodul dieses Routers. Jeder Eingangsverbindungsabschnitt **58** gehört zu einem Eingangszwischenspeicher **62** und jeder Ausgangsverbindungsabschnitt **60** gehört zu einem Ausgaberegister **64**. Die Eingangszwischenspeicher und Ausgaberegister sind miteinander verbunden über einen 7×7 Crossbar-Switch **66**.

[0075] Der Fachmann wird verstehen, dass die vorliegende Erfindung in Strukturnetzwerken mit unterschiedlichen Topologien und unterschiedlichen Anzahlen von Dimensionen eingesetzt werden kann. Auch kann mehr als ein Verbindungsabschnitt zu und von der Leitungsschnittstelle vorgesehen sein. In einer alternativen Ausführungsform sind zwei Ausgangsverbindungsabschnitt von der Struktur zu der Leitungsschnittstelle vorgesehen, wodurch die Gesamtanzahl an Ausgangsverbindungsabschnitten und damit an Ausgaberegistern auf acht gebracht wird. In diesem Fall sind die Eingangszwischenspeicher und Ausgaberegister mittels eines 7×8 Crossbar-Switches verbunden. Der zweite Ausgangsverbindungsabschnitt sorgt für eine zusätzliche Bandbreite, um Pakete von dem Strukturnetzwerk abzuleiten, wenn ein einzelner Knoten gleichzeitig Verkehr aus vielen Richtungen erhält.

[0076] Ein virtuelles Netzwerk ist für jedes Paar von Ausgabeknoten vorgesehen. Jeder der sieben Eingangszwischenspeicher **62** enthält einen Zwischenspeicher, z. B. von einem Flit, für jedes virtuelle Netzwerk in der Maschine. In einer Ausführungsform stellt eine $6 \times 10 \times 10$ Ringstruktur 600 Knoten bereit. Ein einzelnes virtuelles Netzwerk wird einem Paar von maximal entfernten Ausgabeknoten in dem Netzwerk zugeordnet, da gewährleistet wird, dass minimale Strecken zwischen diesen beiden Knoten keine Verbindungsabschnitte gemeinsam benutzen und damit gewährleistet ist, dass sie sich nicht gegenseitig beeinflussen. Des Weiteren sind zwei virtuelle Netzwerke für jedes Knotenpaar vorgesehen, um zwei Prioritäten beim Bedienen von unterschiedlichen Verkehrsklassen zu ermöglichen. Damit existieren in dem Router 600 virtuelle Netzwerke: zwei virtuelle Netzwerke für jedes der 300 Knotenpaare. Jeder Eingangszwischenspeicher **62** enthält Platz für 600 36-Byte Flits (insgesamt 21,600 Bytes).

[0077] Als Verbesserung weist jeder Eingangszwischenspeicher Speicher für zwei Flits für jeden virtuellen Kanal auf. Die Größe eines Flits bestimmt das maximale Lastverhältnis eines einzelnen virtuellen Kanals und den Fragmentierungsverlust, der mit dem Aufrunden der Pakete auf eine ganze Anzahl an Flits verbunden ist. Die maximale Bandbreite auf einem einzelnen Strukturverbindungsabschnitt, der von einem einzelnen virtuellen Kanal verwendet werden kann, kann nicht mehr als die Flit-Größe multipliziert mit der Anzahl der Flits pro virtuellem Kanal Zwi-

schenspeicher geteilt durch die Zeit sein, die ein Header-Flit benötigt, um sich durch einen Router auszubreiten. Wenn ein Flit z. B. 36 Bytes hat, wenn es ein einzelnes Flit pro Zwischenspeicher gibt, und ein Header-Flit **10** 10 ns-Takte benötigt, um sich durch einen Router auszubreiten, dann beträgt die maximale Bandbreite pro virtuellem Kanal 360 MB pro Sekunde. Wenn die Verbindungsabschnittsbandbreite 1200 MB pro Sekunde beträgt, kann ein einzelner virtueller Kanal höchstens 30% der Verbindungsabschnittsbandbreite verwenden. Wenn die Flit-Zwischenspeicherkapazität mindestens so groß ist wie die Verbindungsabschnittsbandbreite geteilt durch die Router-Latenzzeit (120 Bytes in diesem Fall), kann ein einzelner virtueller Kanal die gesamte Verbindungsabschnittskapazität verwenden.

[0078] Man möchte die Flit-Größe so groß wie möglich machen, um sowohl die Verbindungsabschnittsbandbreite zu maximieren, die ein einzelner virtueller Kanal nutzen kann, als auch um den Overhead der Flit-Verarbeitung für größere Nutzlasten zu amortisieren. Andererseits reduziert ein großes Flit die Effektivität durch das Verursachen einer internen Fragmentierung, wenn kleine Pakete auf ein Vielfaches der Flit-Größe aufgerundet werden müssen. Wenn z. B. die Flit-Größe 64 Bytes beträgt, muss ein 65 Byte Paket auf 128 Bytes aufgerundet werden, was zu fast 50% Fragmentierungs-Overhead führt.

[0079] Ein Verfahren zum Nutzen der Vorteile einer großen Flit-Größe ohne das Auftreten eines Fragmentierungs-Overheads ist es, angrenzende Flits in Paare zu gruppieren, die so gehandhabt werden können, als wären sie einzelne Flits von doppelter Größe. Für alle Flits außer dem letzten einer Nachricht mit ungerader Länge wird die gesamte Flit-Verarbeitung einmal für jedes Flit-Paar durchgeführt, was den Flit-Verarbeitungs-Overhead halbiert. Das letzte ungerade Flit wird als solches verarbeitet. Diese ungeraden Einzel-Flits sind jedoch selten und so wird ihr erhöhter Verarbeitungs-Overhead herausgemittelt. In der Tat ist die Flit-Paarung äquivalent mit zwei Größen von Flits – normale Größe und doppelte Größe. Das Ergebnis ist, dass lange Nachrichten den niedrigen Verarbeitungs-Overhead von Flits mit doppelter Größe sehen und kurze Nachrichten den niedrigen Fragmentierungs-Overhead von Flits mit normaler Größe sehen. In der bevorzugten Ausführungsform haben Flits eine Länge von 36 Bytes und werden in Paare von 22 Bytes Gesamtlänge gruppiert.

[0080] Wenn ein virtueller Kanal eines Struktur-Routers, der für einen Ausgangsknoten bestimmt ist, frei ist, wenn das Header-Flit eines Pakets für diesen virtuellen Kanal ankommt, wird der Kanal diesem Paket für die Dauer des Pakets zugeordnet, d. h. bis der Wurm vorüber ist. Mehrere Pakete jedoch können an einem Router für denselben virtuellen Kanal durch mehrere Eingänge empfangen werden. Wenn

ein virtueller Kanal bereits zugeordnet ist, muss das neue Header-Flit in seinem Flit-Zwischenspeicher warten. Wenn der Kanal nicht zugeordnet ist, aber zwei Header-Flits für diesen Kanal zusammen ankommen, muss eine gerechte Arbitration stattfinden. Weil begrenzter Zwischenspeicherplatz jedem virtuellen Kanal zugeordnet ist, wird eine Blockierung an einem Ausgabeknoten von der Struktur durch den Gegendruck zu der Eingangsleitungsschnittstelle für jedes Paket auf diesem virtuellen Netzwerk gesehen. Die Eingangsleitungsschnittstelle kann dann entsprechende Maßnahmen ergreifen, um die nachfolgenden Pakete umzuleiten. Mit der Zuordnung von unterschiedlichen Zielen zu unterschiedlichen virtuellen Netzwerken wird eine gegenseitige Beeinflussung zwischen den Zielen vermieden. Der Verkehr wird isoliert.

[0081] Wenn ein Flit einmal zu einem virtuellen Ausgabekanal zugeordnet ist, hat es nicht mehr die Möglichkeit für die Übertragung über einen Verbindungsabschnitt, bis ein Signal von dem stromab gelegenen Knoten empfangen wird, dass ein Eingangszwischenspeicher an diesem Knoten für den virtuellen Kanal verfügbar ist.

[0082] Ein einfaches Flusssteuerungsverfahren ist in den [Fig. 9](#), [Fig. 10A](#) und [Fig. 10B](#) dargestellt. In jedem Zyklus wird eine Anzahl M der aktiven Flits in jedem Eingangszwischenspeicher von einem gerechten Arbitrationsprozess **68** ausgewählt, um in Konkurrenz zum Zugang zu ihren angeforderten Ausgangsverbindungsabschnitt zu treten. Die ausgewählten Flits leiten ihre Ausgangsverbindungsabschnittsanfragen an einen zweiten Arbitrator **70** weiter, der mit dem angefragten Ausgangsverbindungsabschnitt verbunden ist. Dieser Arbitrator wählt höchstens ein Flit für die Weiterleitung zu jedem Ausgangsverbindungsabschnitt aus. Die erfolgreichen Flits werden anschließend über den Crossbar-Switch zum Ausgaberegister weitergeleitet und anschließend über den Ausgangsverbindungsabschnitt zu dem nächsten Router in der Schaltstruktur übertragen. Bis sie in diesem zweistufigen Arbitrationsprozess ausgewählt werden, verbleiben die Flits in dem Eingabezwischenspeicher, wobei stromauf Gegendruck anliegt.

[0083] Der Struktur-Router an jedem Leitungsschnittstellenmodul verwendet eine Guthaben- bzw. Credit-basierte Flusssteuerung, um den Fluss von Flits durch das Strukturnetzwerk zu regeln. Mit jedem Satz von Eingabezwischenspeichern **62** sind 2 V-Bit Vektoren verbunden; ein Präsenzvektor P und ein aktivierter Vektor E . V ist, wie in [Fig. 10A](#) dargestellt, die Anzahl der virtuellen Netzwerke und damit die Anzahl der Einträge in dem Zwischenspeicher. Ein Bit des Präsenzvektors $P[v, i]$ wird gesetzt, wenn der Eingangszwischenspeicher i ein Flit aus dem virtuellen Netzwerk v enthält. Das Bit $E[v, i]$ wird gesetzt,

wenn dieses Flit aktiviert wird, um den nächsten Hop bzw. Sprung der Strecke zu seinem Zielverbindungsabschnitt zu nehmen.

[0084] Wie in [Fig. 10B](#) dargestellt, ist zu jedem Ausgaberegister zugehörig ein V-Bit Guthaben- bzw. Credit-Vektor C , der das Gegenstück (Komplement) des Präsenzvektors an dem gegenüberliegenden Ende des Strukturverbindungsabschnitts an dem empfangenden Knoten spiegelt. D. h. $C[v, j]$ wird an einen gegebenen Ausgang j gesetzt, wenn $P[v, i]$ an dem Eingangsanschluss auf der gegenüberliegenden Seite des Verbindungsabschnitts frei ist. Wenn $C[v, j]$ gesetzt ist, dann weist das Ausgaberegister ein Guthaben bzw. Credit für den leeren Zwischenspeicher am gegenüberliegenden Ende des Verbindungsabschnitts auf.

[0085] Flits in einem Eingangszwischenspeicher werden aktiviert, um ihren nächsten Hop bzw. Sprung durchzuführen, wenn ihr angefragter Ausgangsverbindungsabschnitt ein Guthaben bzw. Credit für ihr virtuelles Netzwerk aufweist. Man nehme z. B. an, dass das Paket im virtuellen Netzwerk v des Eingangszwischenspeichers i den ausgewählten Ausgangsverbindungsabschnitt j für den nächsten Hop bzw. Sprung seiner Strecke aufweist. Wir bezeichnen dies als $F[v, i] = j$, wobei F der Weiterleitungsvektor ist. Das Flit in diesem Eingangszwischenspeicher wird aktiviert, um seinen nächsten Hop durchzuführen, wenn zwei Bedingungen erfüllt sind. Zuerst muss $P[v, i] = 1$ vorliegen und zweitens muss ein Guthaben bzw. Credit für Zwischenspeicherplatz am nächsten Hop $C[v, j] = 1$ vorliegen.

[0086] Der Eingabezwischenspeicherplatz wird zu jedem virtuellen Netzwerk separat allokiert, wohingegen die Ausgaberegister und die zugehörigen physikalischen Kanäle von den virtuellen Netzwerken gemeinsam benutzt werden. Das Guthaben- bzw. Credit-basierte Flusssteuerungsverfahren gewährleistet, dass ein virtuelles Netzwerk, das blockiert oder verstopft ist, nicht unbegrenzt die physikalischen Kanäle zuschnürt, da nur aktive Flits bei der Arbitration nach Ausgangsverbindungsabschnitten miteinander in Konkurrenz stehen. Des Weiteren wird, da lediglich ein oder zwei Flits pro virtuellem Netzwerk in jedem Eingangszwischenspeicher gespeichert werden, ein starrer Gegendruck von einem beliebigen blockierten Ausgabeknoten auf die Weiterleitungsrichtung des Eingabeknoten ausgeübt.

Allokation

[0087] Die Arbitration und Flusssteuerung können als ein Allokationsproblem angesehen werden, dass das Zuordnen von virtuellen Kanäle zu Paketen, die von verschiedenen Eingabeknoten ankommen und für gemeinsame Ausgabeknoten bestimmt sind, und das Zuordnen von physikalischer Kanalbandbreite zu

Flits umfasst, die für denselben nächsten Knoten in dem Strukturweg bestimmt sind.

[0088] In einer mehrstufigen Schaltstruktur bewegen sich Pakete, die aus einem oder mehreren Flits zusammengesetzt sind, von ihrer Quelle zu ihrem Ziel über einen oder mehrere Struktur-Router vorwärts. Bei jedem Hop bzw. Sprung, kommt das Header-Flit einer Nachricht an einem Knoten auf einem virtuellen Eingabekanal an. Es kann sich nicht weiter vorwärts bewegen, bis ihm ein virtueller Ausgabekanal zugeordnet wird. In der Schaltstruktur der bevorzugten Ausführungsform kann jedes Paket auf lediglich einem virtuellen Kanal geroutet werden. Wenn der virtuelle Kanal freigegeben ist wenn das Paket ankommt, wird er dem ankommenden Paket zugeordnet. Wenn jedoch der virtuelle Kanal belegt ist, wenn das Paket ankommt, muss das Paket warten, bis der virtuelle Ausgabekanal frei wird. Wenn ein oder mehrere Pakete auf einen virtuellen Kanal warten, wenn er freigegeben wird, wird eine Arbitration durchgeführt und der Kanal wird einem der wartenden Pakete zugeordnet.

[0089] Sobald ein Paket erfolgreich den virtuellen Kanal an sich gebracht hat, muss es in Konkurrenz um physikalische Kanalbandbreite treten, um seine Flits zum nächsten Knoten auf seiner Strecke voranzubringen. Ein Paket kann nur dann um Bandbreite konkurrieren, wenn zwei Bedingungen gelten. Erstens muss mindestens ein Flit des Pakets in dem Knoten vorhanden sein. Zweitens muss mindestens ein Flit an Zwischenspeicherplatz an dem nächsten Knoten verfügbar sein. Wenn diese zwei Bedingungen nicht erfüllt sind, gibt es entweder kein Flit zum Weiterleiten oder keinen Platz, an dem das Flit am nächsten Hop abgelegt wird. Wenn beide Bedingungen für ein gegebenes Paket erfüllt sind, dann wird dieses Paket aktiviert, um ein Flit zu übertragen. Bevor jedoch ein Flit gesendet werden kann, muss das Paket zwei Arbitrationen erfolgreich überstehen. Unter allen aktivierten Paketen muss, damit ein Flit des Pakets zum nächsten Knoten auf der Strecke vorwärts kommt, einem Paket sowohl ein Ausgabeanschluss vom Eingabe-Flit-Zwischenspeicher und der physikalische Ausgabekanal zugesprochen werden.

[0090] Für kleine Anzahlen von virtuellen Kanälen kann das Allokationsproblem parallel für die grundlegenden Fälle der [Fig. 9](#), [Fig. 10A](#) und [Fig. 10B](#) unter Verwendung von kombinatorischer Logik gelöst werden.

[0091] Man betrachtet zunächst das Problem der virtuellen Kanalallokation. Ein Zustandsbit H ist mit jedem der V virtuellen Eingabekanal an jedem der K Eingangssteuereinrichtungen verbunden. Dieses Bit wird gesetzt, wenn der virtuelle Eingabekanal ein Header-Flit enthält, dem noch kein virtueller Ausgabekanal zugeordnet wurde. Das Bit-Array $H[1:V, 1:K]$

bestimmt die Nachfrage nach virtuellen Kanälen. Ein Zustandsbit B ist mit jedem der V virtuellen Ausgabekanal in jedem der K Ausgangssteuereinrichtungen verbunden. Dieses Bit wird gesetzt, wenn der virtuelle Ausgabekanal belegt ist. Das Bit-Array $B[1:V, 1:K]$ bestimmt den Allokationsstatus der virtuellen Kanäle.

[0092] Um einen virtuellen Kanal v in der Ausgangssteuereinrichtung k zu allokierten, muss zuerst eine Arbitration unter den virtuellen Kanälen v in jedem der k Eingangssteuereinrichtungen durchgeführt werden, wobei die Eingangsteuereinrichtung i lediglich in Konkurrenz tritt, wenn (1) $H[v, i]$ gesetzt ist und (2) und das Ziel des Kanals $F[v, i] = k$ ist. Dem Eingang, der die Arbitration gewinnt, wird der virtuelle Kanal nur gewährt, wenn $B[v, k] = 0$ ist.

[0093] Die Situation ist für die Allokation von physikalischer Kanalbandbreite zu Flits ähnlich. Der Zwischenspeicherstatus jedes virtuellen Eingabekanal wird durch ein Präsenzbit P angezeigt, das gesetzt ist, wenn sich ein oder mehrere Flits in dem vorliegenden Knoten befinden. Jeder virtuelle Ausgabekanal schaut voraus und enthält ein Guthaben- bzw. Credit-Bit C, das gesetzt ist, wenn ein oder mehrere leere Zwischenspeicher für den nächsten Knoten verfügbar sind. Angenommen wir wählen aus, die Allokation seriell durchzuführen (was suboptimal ist); zuerst die Arbitration nach einem Ausgangsport der Eingangsteuereinrichtung und anschließend die Arbitration nach einem Ausgabekanal. Angenommen jeder Eingangszwischenspeicher weist M Ausgabeanschlüsse auf. Dann bestimmen wird für den Eingangszwischenspeicher i zuerst, welche virtuellen Kanäle aktiv sind. Ein aktivierter Vektor $E[v, i]$ wird als $E[v, i] = \neg H[v, i] \wedge P[v, i] \wedge C[v, i]$ berechnet, wobei die logische Negation bezeichnet, \wedge eine logische UND-Operation bezeichnet und j das Ziel des Pakets auf dem virtuellen Kanal v der Eingangsteuereinrichtung i ist. Somit wird einem Paket ermöglicht, ein Flit weiterzuleiten, wenn es nicht auf einen virtuellen Kanal wartet, wenn mindestens ein Flit in seinem Zwischenspeicher vorhanden ist und wenn es mindestens ein Flit an verfügbarem Speicher am nächsten Hop gibt. Als nächstes wird eine Arbitration für alle der aktivierten Kanäle in dem Eingangszwischenspeicher für die M Ausgabeanschlüsse des Eingangszwischenspeichers durchgeführt. Dies erfordert einen V-Eingabe-, N-Ausgabe-Arbitrer. Schließlich führen die Gewinner jeder lokalen Arbitration eine Arbitration für die virtuellen Ausgabekanal durch, dies erfordert K-, MK-Eingabearbitrer.

[0094] Mit großen Anzahlen an virtuellen Kanälen erfordert eine kombinatorische Realisierung der Allokationslogik eine unerschwingliche Anzahl von Gattern. Die bevorzugte Schaltstruktur weist $V = 600$ virtuelle Kanäle und $K = 7$ Anschlüsse auf. Um dieses Allokationsverfahren mit kombinatorischer Logik zu implementieren sind somit 4200 Elemente an Vektoren

H und B, 4200 3:8 Decoder, um die Arbitrationen zu bewerten, und 4700 7-Eingabearbitrer erforderlich, um die Gewinner auszuwählen. Zwischen den Flip-Flops, um den Zustand zu halten, den Decodern und den Arbitern sind etwa **50** 2-Eingangs-Gatter für jeden der 4200 virtuellen Kanäle erforderlich, mit einer Gesamtanzahl von über 200 000 Logik-Gattern, einer unerschwinglichen Anzahl.

[0095] Für den bevorzugten Router weisen die P- und C-Arrays ebenfalls jeweils 4200 Elemente auf. Zwischen den C-Multiplexern und den Arbitern erfordert jedes Element etwa 40 Gatter. Damit erfordert die Bandbreitenallokation zusätzlich 160 000 Logik-Gatter.

[0096] Während sie für Router mit kleinen Anzahlen an virtuellen Kanälen mit $V \leq 8$ ziemlich vernünftig ist, ist die kombinatorische Allokation offensichtlich für den Router mit $V = 600$ nicht machbar.

Ereignis-gesteuerte Allokation

[0097] Die Logik, die erforderlich ist, um die Allokation durchzuführen, kann in erheblichem Maße reduziert werden durch die Beobachtung, dass für große Anzahlen von virtuellen Kanälen der Zustand der meisten virtuellen Kanäle von einem Zyklus zum nächsten unverändert ist. Während eines gegebenen Flit-Intervalls kann höchstens an einem virtuellen Kanal einer gegebenen Eingangsteuereinrichtung ein Flit ankommen, und an höchstens M virtuellen Kanälen können Flits abgehen. Die verbleibenden $V-M-1$ virtuellen Kanäle bleiben unverändert.

[0098] Die Eigenschaft der geringen Änderungen auf den Zustand des virtuellen Kanals kann zum Vorteil ausgenutzt werden über die Verwendung einer Ereignis-gesteuerten Allokationslogik. Mit diesem Ansatz wird eine einzelne Kopie (oder eine kleine Anzahl von Kopien) der Statusaktualisierung des virtuellen Kanals und die Allokationslogik über eine große Anzahl von virtuellen Kanälen gebündelt. Nur aktive virtuelle Kanäle, wie sie durch das Vorkommen von Ereignissen identifiziert werden, werden in ihrem Zustand untersucht und aktualisiert und nehmen an der Arbitration teil.

[0099] Zwei Arten von Ereignissen, Ankunftsereignisse und Guthaben- bzw. Credit-Ereignisse, aktivieren die Zustandsaktualisierungslogik des virtuellen Kanals. Eine dritte Art von Ereignis, ein Transportereignis, bestimmt, welche virtuellen Kanäle an der Arbitration nach physikalischer Kanalbandbreite teilnehmen. Jedes Mal, wenn ein Flit an einem Knoten ankommt, wird ein Ankunftsereignis in eine Warteschlange gestellt, um den Zustand des mit diesem Flit verbundenen virtuellen Kanals zu überprüfen. Eine ähnliche Überprüfung wird als Reaktion auf ein Guthaben- bzw. Credit-Ereignis durchgeführt, das je-

des Mal in eine Warteschlange gestellt wird, wenn der Zustand eines stromab gelegenen Zwischenspeichers eines virtuellen Kanals geändert wird. Das Untersuchen des Zustands eines virtuellen Kanals kann zur Allokation des Kanals zu einem Paket und/oder zum Festlegen eines Flits zum Transport zu dem stromab gelegenen Knoten führen. Im letzteren Fall wird ein Transportereignis erzeugt und in die Warteschlange gestellt. Nur virtuelle Kanäle mit bevorstehenden Transportereignissen nehmen an der Arbitration bezüglich der Ausgangsports der Eingangszwischenspeicher und der physikalischen Ausgabekanäle teil. Sobald ein Flit bei beiden Arbitrationen erfolgreich ist und tatsächlich transportiert wird, wird das entsprechende Transportereignis aus der Warteschlange genommen.

[0100] Die Logik, um Ereignis-gesteuerte Kanalallokation zu implementieren, ist in den [Fig. 11A](#) und [Fig. 11B](#) dargestellt. [Fig. 11A](#) zeigt eine von sieben Eingangssteuereinrichtungen, während [Fig. 11B](#) eine von sieben Ausgangssteuereinrichtungen zeigt. Jede Eingangssteuereinrichtung ist mit jeder Ausgangssteuereinrichtung an den drei gezeigten Punkten verbunden. Jede Eingangssteuereinrichtung umfasst einen Zieltabelle **72**, eine Ankunfts warteschlange **74**, eine Guthabenwarteschlange **76** und einen Flitzwischenspeicher **62**. Eine Virtualkanalzustandstabelle **80** und eine Transportwarteschlange **82** sind in jeder Ausgangssteuereinrichtung inkludiert. Die Figuren zeigen eine Ereignis-gesteuerte Anordnung, wobei der Virtualkanalzustand mit jeder Ausgangssteuereinrichtung verbunden ist. Es ist ebenfalls möglich, den Zustand mit den Eingangssteuereinrichtungen zu verbinden. Das Anordnen der Zustandstabelle in der Ausgangssteuereinrichtung hat den Vorteil, dass die Allokation der virtuellen Kanäle (die an der Ausgangssteuereinrichtung durchgeführt werden muss) und die Bandbreitenallokation (die an beiden Enden durchgeführt werden kann) unter Verwendung desselben Mechanismus durchgeführt werden kann.

[0101] Die Zieltabellen, Flit-Zwischenspeicher und Zustandstabellen der virtuellen Kanäle haben Einträge für jeden virtuellen Kanal, während die drei Warteschlangen lediglich eine kleine Anzahl an Einträgen erfordern. Für jeden virtuellen Kanal zeichnet die Zieltabelle den Ausgabeanschluss auf, der von dem aktuellen Paket auf diesem Eingabekanal angefordert wird, falls vorhanden (d. h. F_a), der Flit-Zwischenspeicher **62** sorgt für die Speicherung ein oder mehrerer Flits des Paketes und der Zustand des virtuellen Ausgabekanal wird in der Zustandstabelle aufgezeichnet. Die Ankunfts-, Guthaben- und Transportwarteschlangen enthalten Einträge für jedes Ereignis, das aufgetreten ist, jedoch noch nicht verarbeitet wurde.

[0102] Auf der Eingangsseite dienen die Ankunfts-

warteschlange mit zwei Anschlüssen, die Guthabenwarteschlange und der Flit-Zwischenspeicher auch als Synchronisationspunkt wie durch die gestrichelte Linie in [Fig. 11A](#) dargestellt. Der linke Anschluss dieser drei Strukturen und die gesamte Logik der gestrichelten Linie (einschließlich der Zieltabelle) arbeitet in der Taktdomain des Eingabekanal. Der rechte Anschluss dieser drei Strukturen und sämtliche Logik rechts der gestrichelten Linie einschließlich [Fig. 11B](#) arbeitet in der internen Taktdomain des Routers.

[0103] In einer alternativen Ausführungsform werden ankommende Flits auf die lokale Taktdomain synchronisiert, bevor sie auf die Ankunfts warteschlange oder die Zieltabelle zugreifen.

[0104] Mit der in den [Fig. 11A](#) und [Fig. 11B](#) gezeigten Anordnung wird eine Allokation eines Flit-Zyklus eines virtuellen Kanals oder eines physikalischen Kanals über eine Abfolge von drei Ereignissen Ankunft, Transport und Guthaben bzw. Credit durchgeführt. Ein ankommendes Flit verlangt nach Arbitration für den Zugriff auf die Zustandstabelle für einen virtuellen Ausgabekanal. Wenn gewährt, wird die Tabelle aktualisiert, um das ankommende Flit zu berücksichtigen und, wenn der Kanal zu dessen Eingangssteuereinrichtung allokiert ist und ein Guthaben bzw. Credit verfügbar ist, wird eine Transportanfrage in die Warteschlange gestellt, um das Flit zu weiterzubewegen. Die Transportanfrage arbitriert für den Zugang zu dem Eingabeflit-Zwischenspeicher. Wenn der Zugriff gewährt wird, wird das Flit aus dem Zwischenspeicher entfernt und zum nächsten Knoten weitergeleitet. Immer dann wenn ein Flit aus dem Flit-Zwischenspeicher entfernt wird, wird ein Guthaben bzw. Credit in die Warteschlange gestellt, um zum vorherigen Knoten übertragen zu werden. Wenn die Guthaben bzw. Credits an einem Knoten ankommen, aktualisieren sie die Zustandstabelle der virtuellen Kanäle, und aktivieren irgendwelche Flits, die auf null Guthaben bzw. Credit warten. Schließlich aktualisiert die Ankunft eines End-Flits an einem Knoten den Zustand des virtuellen Kanals, um den Kanal frei zu geben.

[0105] Jedes Mal, wenn ein Flit an einer Eingangssteuereinrichtung ankommt, werden die Inhalte des Flits in dem Flit-Zwischenspeicher **62** gespeichert. Gleichzeitig wird auf die Zieltabelle **72** zugegriffen und ein Ankunftsereignis, das mit der angefragten Ausgabeportnummer markiert ist, wird bei **74** in die Schlange gestellt. Die Zieltabelle wird durch das Header-Flit jedes Pakets aktualisiert, um den Ausgabeanschluss des Pakets aufzuzeichnen und anschließend von den verbleibenden Flits des Pakets befragt, um die gespeicherte Anschlussnummer zu ermitteln. Ein Ankunftsereignis umfasst eine Kennung des virtuellen Kanals (10 Bits) ein Header-Bit und eine Kennung des Ausgabeports (3 Bits). Die Ankunftsereignisse an den Köpfen von jeder der K Ankunfts wartesch-

schlangen der Eingangssteuereinrichtung (zusammen mit den Kennungen des Eingabeanschlusses (3 Bit)) werden zu den Arbitern **84** an jede Ausgangssteuer-einrichtung verteilt. An jeder Ausgangssteuer-einrichtung streben die Ankunftsereignisse, die diesen Ausgabeanschluss anfragen, nach Arbitration für den Zugang zu der Zustandstabelle **80**. In jedem Zyklus werden die erfolgreichen Ankunftsereignisse aus der Warteschlange genommen und verarbeitet. Die nicht erfolgreichen Ereignisse bleiben in der Warteschlange und konkurrieren wiederum miteinander um den Zugriff auf die Zustandstabelle bei dem nachfolgenden Zyklus.

[0106] Wie in [Fig. 12](#) dargestellt hält für jeden virtuellen Ausgabekanal v beim Ausgang k die Zustandstabelle der virtuellen Kanäle **80** einen Zustandsvektor $S[v, k]$ mit:

1. Der Allokationsstatus des Kanals B , ungenutzt (0), belegt (1) oder Ende bevorstehend (2).
2. Die Eingangssteuereinrichtung, die diesem Kanal zugeordnet ist (wenn B gesetzt ist), I (3 Bits).
3. Einen Bit-Vektor von Eingangssteuereinrichtungen, die auf diesen Kanal warten, W (7 Bits).
4. Die Anzahl der Credits (leere Zwischenspeicher an dem nächsten Knoten), C (1 Bit).
5. Die Anzahl der in diesem Knoten vorhandenen Flits, P (1 Bit).

[0107] Die ersten drei von diesen (B , I , W) sind mit der Allokation von virtuellen Ausgabekanälen zu virtuellen Eingabekanälen verbunden, während die letzten beiden (C , P) mit der Allokation von physikalischer Kanalbandbreite zu Flits verbunden sind. Die Anzahl von Flits in jedem Element des Zustandsvektors kann soweit erforderlich variiert werden. Wenn z. B. mehr Flit-Zwischenspeicher an jedem Knoten verfügbar sind, dann würden mehr Bits zu dem C - und P -Feld allokiert. Ziel dieses Zustands hier entspricht direkt den Zustandsbits in dem Ansatz der kombinatorischen Logik. Die B -, C -, und P -Bits sind identisch. Die W -Bits entsprechen den H -Bits, eingeschränkt auf den angefragten Ausgabekanal.

[0108] Die Anzahl an Bits in dem Wartevektor V kann erhöht werden, um für eine verbesserte Fairness bei der Arbitration zu sorgen. Mit lediglich einem einzelnen Bit kann eine Zufalls oder Round-Robin-Arbitration durchgeführt werden. Wenn 3-Bits für jeden Eintrag gespeichert werden, kann eine Warteschlangen-Arbitration mit den virtuellen Eingabekanälen durchgeführt werden, die in der Reihenfolge bedient werden, in der ihre Anfragen ankommen. Jeder virtuelle Kanal "zieht" eigentlich "eine Nummer", wenn er an der Zustandstabelle ankommt, und diese Nummer wird in seinem Eintrag des W -Vektors gespeichert. Wenn der Kanal frei wird, wird die "nächste" Nummer bedient.

[0109] Wenn ein zu einem virtuellen Kanal v gehöriges

Ankunftsereignis von der Eingangssteuereinrichtung I an der Zustandstabelle für den Ausgang k ankommt, liest es $S[v, k]$ und führt eine der nachfolgenden Aktionen aus je nach Art des Ereignisses (Header vs. Körper) und des Zustands des Kanals.

1. Wenn das Flit ein Kopfsatz ist, der Kanal unbelegt ist, $B = 0$, und stromab gelegene Guthaben bzw. Credits vorhanden sind, $C \neq 0$, (a) wird der Kanal durch das Setzen von $B = 1$, $I = i$, dem Eingang zugeordnet, (b) durch das Dekrementieren von C ein stromab gelegener Zwischenspeicher allokiert und (c) eine Transportanfrage für (v, i, k) bei **82** in die Warteschlange gestellt.

2. Wenn das Flit ein Kopf (Satz) ist, der Kanal unbelegt ist, aber keine stromab gelegenen Guthaben bzw. Credits vorhanden sind, wird der Kanal dem Eingang zugeordnet und der Präsenzzähler P wird erhöht. Es wird kein stromab gelegener Zwischenspeicher allokiert und keine Transportanfrage in die Warteschlange gestellt.

3. Wenn das Flit ein Kopfsatz ist und der Kanal belegt ist, $B = 1$, wird die Anfrage nach dem virtuellen Kanal durch Setzen des i -ten Bits des Wartevektors W in die Warteschlange gestellt.

4. Wenn das Flit ein Haupt-Flit (Body-Flit) ist und stromab gelegene Guthaben bzw. Credits vorhanden sind, wird ein stromab gelegener Zwischenspeicher allokiert und eine neue Transportanfrage in die Warteschlange gestellt.

5. Wenn das Flit ein Haupt-Flit ist und keine stromab gelegenen Guthaben bzw. Credits vorhanden sind, wird der Präsenzzähler erhöht.

6. Wenn das Flit ein Endsatz (Tail) und $W = 0$ ist, keine wartenden Kopfsätze vorhanden sind, dann wird, wenn ein Guthaben bzw. Credit vorhanden ist, das End-Flit zum Transport in die Warteschlange gestellt und der Kanal als unbelegt markiert, $B = 0$. Ansonsten wird, wenn kein Guthaben bzw. Credit vorhanden ist, der Kanal als für den Endsatz bevorstehend (Tail Pending) markiert, $B = 2$, und somit wird die Ankunft eines Guthabens dem Endsatz übertragen und dem Kanal freigegeben.

7. Wenn das Flit ein Endsatz (Tail) ist, ein Guthaben bzw. Credit verfügbar ist ($C \neq 0$) und wartende Pakete vorhanden sind ($W \neq 0$) wird das End-Flit (Tail Flit) für den Transport in die Warteschlange gestellt wie in den obigen Fällen 1 und 4. Eine Arbitration wird durchgeführt, um einen der wartenden Eingänge j auszuwählen. Der Kanal wird diesem Eingang zugeordnet ($B = 1$, $I = j$), und, wenn zusätzliches Guthaben bzw. Credit vorhanden ist wird dieses neue Header-Flit für den Transport in die Warteschlange gestellt; ansonsten wird es als vorhanden gekennzeichnet.

8. Wenn das Flit ein End-Flit (Tail) ist, und ein Guthaben bzw. Credit nicht verfügbar ist, ($C = 0$) wird der Präsenzzähler erhöht und der Status des Kanals als "End-Flit (Tail) bevorstehend" gekennzeichnet ($B = 2$).

[0110] Wenn nur ein einzelner Flit-Zwischenspeicher pro virtuellem Kanal vorhanden ist, besteht, wenn ein Haupt-Flit ankommt, kein Bedarf, den Allokationsstatus des virtuellen Kanals (B , I und W) zu überprüfen, da das Flit nur ankommen könnte, wenn der Kanal bereits zu seinem Paket allokiert wäre ($B = 1$, $I = i$). Wenn mehr als ein Flit-Zwischenspeicher pro virtuellem Kanal vorhanden ist, muss der virtuelle Kanal jeder Haupt-Flit-Ankunft überprüft werden. Flits, die für Kanäle ankommen, die auf einen virtuellen Ausgabekanal warten, werden Ereignisse erzeugen, die ignoriert werden müssen. Auch muss die Anzahl der Flits, die in einem wartenden virtuellen Kanal zwischengespeichert sind, an die Zustandstabelle **80** kommuniziert werden, wenn der Ausgabekanal zu dem wartenden Kanal allokiert wird. Dies kann z. B. durch Aktualisieren des Flit-Zählers in der Zustandstabelle von dem Zähler in dem Flit-Zwischenspeicher erreicht werden, immer dann, wenn ein Header-Flit transportiert wird. Es ist anzumerken, dass wir in dem obigen Fall 1 sowohl den virtuellen Kanal als auch die Kanalbandbreite für das Header-Flit in einer einzigen Operation in der Zustandstabelle allokierten. End-Flits (Tail Flits) haben ein Paar von Aktionen zur Folge: das End-Flit wird zuerst als ein Haupt-Flit (Body Flit) verarbeitet, um die Bandbreite zu allokierten, um das End-Flit vorwärts zu bewegen, anschließend wird das End-Flit als ein End-Flit verarbeitet, um den Kanal freizugeben und möglicherweise ein bevorstehendes Header-Flit zu bewegen. Sofern nicht die Transportwarteschlange gleichzeitig zwei Eingänge akzeptieren kann, muss dies sequentiell erfolgen, da eine Ankunft eines End-Flits zwei Flits für den Transport in die Warteschlange stellen kann: das End-Flit selbst und das Header-Flit eines wartenden Pakets.

[0111] Jeder Eintrag in der Transportwarteschlange (v , i , k) ist eine Anfrage, die Inhalte des Flit-Zwischenspeichers v in der Eingangssteuereinrichtung i zum Ausgang k vorwärts zu bewegen. Bevor die Anfrage beschieden werden kann steht sie zuerst zur Arbitration bei **86** für den Zugang zum Flit-Zwischenspeicher i an. Bei jedem Zyklus werden die Transportanfragen am Kopf der Warteschlangen in jeder der K Ausgangssteuereinrichtungen ihren angefragten Eingabezweischenspeichern vorgelegt, wo sie durch Arbitration für den Zugang zu den M Anschlüssen vermittelt werden. Die siegreichen Transportanfragen werden aus der Schlange genommen und ihre Flits werden an den entsprechenden Ausgabemultiplexer **88** weitergeleitet. Die anderen Anfragen verbleiben in den Transportwarteschlangen. Es besteht kein Bedarf, hier eine Arbitration für einen Strukturverbindungsabschnitt durchzuführen, da der zu jedem der ausgehenden Strukturverbindungsabschnitte gehörenden Ausgangssteuereinrichtung höchstens eine Anfrage pro Zyklus macht. Jedes Mal, wenn eine Transportanfrage erfolgreich ein Flit an einen Ausgang weiterleitet, wird ein Guthaben (Credit) erzeugt,

um den in dem Eingabe-Flit-Zwischenspeicher frei gewordenen Platz wiederzuspiegeln. Dieses Guthaben (Credit) wird in einer Guthabenwarteschlange **76** aneinander gereiht für die Übertragung zu der Ausgangssteuereinrichtung des vorherigen Knotens. Wenn ein Guthaben (Credit) für den virtuellen Kanal v an der Ausgangssteuereinrichtung k eines Knotens ankommt, liest sie den Zustandsvektor $S[v,k]$ um zu überprüfen, ob irgendwelche Flits auf Guthaben (Credits) warten. Sie fährt wie folgt fort je nach Zustand des Präsenzzählers.

1. Wenn keine Flits warten, $P = 0$, wird der Guthabenzähler erhöht, $C = C + 1$.
2. Wenn Flits warten, $P \neq 0$, wird die Anzahl der wartenden Flits vermindert, $P = P - 1$, und eine Transportanfrage für das erste wartende Flit wird in die Warteschlange gestellt.
3. Wenn ein End-Flit (Tail Flit) ansteht ($B = 2$), wird eine Transportanfrage für das End-Flit in die Warteschlange gestellt. Wenn keine Header-Flits auf den Kanal warten ($W = 0$), wird der Kanal auf unbesetzt gesetzt ($B = 0$). Ansonsten wird, wenn Header-Flits warten ($W \neq 0$), eine Arbitration durchgeführt, um einen wartenden Kanal auszuwählen, z. B. von der Eingangssteuereinrichtung j wird der Kanal zu diesem Kanal allokiert ($B = 1$, $I = j$), und das Header-Flit wird als vorhanden markiert ($P = 1$), so dass das nächste ankommende Guthaben (Credit) dazu führt, dass das Header-Flit übertragen wird.

[0112] In der oben beschriebenen Ereignis-gesteuerten Ausführungsform verarbeitet die Ausgangssteuereinrichtung Haupt-Flits (Body Flits) und End-Flits (Tail Flits) auf unterschiedliche Weise. Insbesondere verarbeitet die Ausgangssteuereinrichtung Haupt-Flits (Body Flits) entsprechend der Techniken 4 und 5 und sie verarbeitet End-Flits (Tail Flits) entsprechend der Techniken 6, 7 und 8, die oben beschrieben wurden.

[0113] Wie in der Technik 7 beschrieben kann ein Header-Flit eines Datenpakets direkt auf ein End-Flit eines vorhergehenden Datenpaketes folgen. Z. B. kann ein Datenpaket einen virtuellen Kanal besetzen, während ein oder mehrere Datenpakete (d. h. ein oder mehrere Header-Flits) auf diesen virtuellen Kanal warten. Wenn ein Ankunftsereignis für ein End-Flit des belegenden Datenpakets die Ausgangssteuereinrichtung erreicht, stellt die Ausgangssteuereinrichtung das End-Flit zur Übertragung zum nächsten stromab gelegenen Struktur-Router in die Warteschlange und teilt den virtuellen Kanal einem der wartenden Datenpakete zu (d. h. einem der wartenden Header-Flits). Demgemäß gesteht die Ausgangssteuereinrichtung dem virtuellen Kanal einem neuen Datenpaket zu, sobald der Struktur-Router das End-Flit zur Übertragung in die Warteschlange stellt.

[0114] In einer alternativen Ereignis-gesteuerten

Ausführungsform verarbeitet die Ausgangssteuereinrichtung Haupt-Flits und End-Flits in ähnlicher Weise. Insbesondere verarbeitet die Ausgangssteuereinrichtung sowohl Haupt- als auch End-Flits gemäß der Techniken 4 und 5 wie oben beschrieben. Wenn ein Ankunftsereignis für ein End-Flit die Ausgangssteuereinrichtung erreicht und wenn ein Guthaben bzw. Credit verfügbar ist, stellt so die Ausgangssteuereinrichtung das End-Flit zur Übertragung in die Warteschlange, ohne den virtuellen Kanal freizugeben oder den virtuellen Kanal einem wartenden Datenpaket zuzuteilen. Wenn ein Struktur-Router, der sich stromab von dem vorliegenden Struktur-Router befindet, das End-Flit empfängt, verarbeitet und weiterleitet, erzeugt der stromab gelegene Struktur-Router ein spezielles End-Flit-Guthaben bzw. Credit anstatt des normalen Guthabens. Der stromab gelegene Router sendet dieses End-Flit-Guthaben stromauf zu dem vorliegenden Struktur-Router. Wenn die Ausgangssteuereinrichtung des vorliegenden Struktur-Routers das End-Flit-Guthaben empfängt, erhöht die Ausgangssteuereinrichtung den Guthabenzähler des virtuellen Kanals in einer Art und Weise, die ähnlich ist zu der für normale Guthaben, und gibt den virtuellen Kanal frei. An diesem Punkt führt, wenn Datenpakete vorhanden sind, die auf den virtuellen Kanal warten, die Ausgangssteuereinrichtung einen Arbitrationsprozess durch, um den virtuellen Kanal einem der wartenden Datenpakete zuzuordnen.

[0115] Der Struktur-Router gemäß der alternativen Ereignisgesteuerten Ausführungsform weist eine geringere Leistung auf als der Struktur-Router der Ereignis-gesteuerten Ausführungsform, die Haupt- und End-Flits unterschiedlich verarbeitet. Insbesondere wird, nachdem der Struktur-Router der alternativen Ausführungsform eine Transportanfrage zur Übertragung eines End-Flits zu einem stromab gelegenen Router in die Warteschlange gestellt hat, der dem Datenpaket dieses End-Flits zugeordnete virtuelle Kanal unbewegt. Der virtuelle Kanal ist nicht für die Verwendung durch ein weiteres Datenpaket verfügbar, bis der Struktur-Router ein End-Flit-Guthaben von dem stromab gelegenen Struktur-Router empfängt.

[0116] Die alternative Ereignis-gesteuerte Ausführungsform hat jedoch eine erheblich einfachere Logik zum Ergebnis, und zwar aus mehreren Gründen. Erstens vereinfacht sie die Handhabung von Ereignissen durch das Reduzieren der Verarbeitungskomplexität eines Ankunftsereignisses eines End-Flits. Die Arbeit wird stattdessen zwischen der Ankunft des End-Flits und den End-Flit-Guthaben- bzw. Credit-Ereignissen aufgeteilt. Des Weiteren vereinfacht sie die Logik, indem sie gewährleistet, dass nur ein einzelnes Paket sich in einem Flit-Zwischenspeicher eines gegebenen virtuellen Kanals zu einem beliebigen Zeitpunkt befindet. Dies wird dadurch gewährleistet, dass einem neuen Paket kein virtueller Kanal gewährt wird, bis das End-Flit des vorhergehenden Pakets den

Flit-Zwischenspeicher freigegeben hat – wie vom End-Flit-Guthaben signalisiert. Im Gegenteil, bei der Ereignis-gesteuerten Ausführungsform, die Haupt-Flits und End-Flits auf unterschiedliche Weise verarbeitet, kann ein Header-Flit eines nächsten Pakets direkt im Anschluss an das End-Flit eines vorliegenden Paketes folgen und zwei oder mehrere Pakete können zur selben Zeit in einem Flit-Zwischenspeicher eines einzelnen virtuellen Kanals in die Warteschlange gestellt werden.

[0117] Jedes hierin beschriebene Ereignis-gesteuerte Allokationsverfahren reduziert die Größe und die Komplexität der Logik, die zur Allokation erforderlich ist, auf zwei Wege. Zunächst kann die Zustandsinformation für die virtuellen Kanäle in einem RAM-Array mit über 10-facher Dichte des Flip-Flop-Speichers, der durch den kombinatorischen Logikansatz erforderlich ist, gespeichert werden. Zum Zweiten wird die Selektions- und Arbitrationslogik um einen Faktor V reduziert. Die Arbitration für den Zugang zu den Zwischenspeichern der virtuellen Kanäle wird nur bei den Kanälen durchgeführt, bei denen Änderungen aufgetreten sind (Ankunft eines Flits oder eines Guthabens bzw. Credits), und nicht auf allen V Kanälen.

[0118] Nur der Flit-Zwischenspeicher, die Zustandstabelle und die Zieltabelle in den [Fig. 11A](#) und [Fig. 11B](#) müssen V Einträge aufweisen. Eine bescheidene Anzahl von Einträgen in den Anfrage-, Transport- und Guthabenwarteschlangen wird ausreichen, um die Geschwindigkeitsungleichgewichte zwischen den verschiedenen Komponenten des Systems auszugleichen. Wenn sich eine Warteschlange füllt, kann der Betrieb der die Warteschlange füllenden Einheit einfach ausgesetzt werden, bis ein Eintrag aus der Warteschlange entfernt wird. Ein Stillstand kann mittels Durchbrechen des Zyklus zwischen den Ereigniswarteschlangen vermieden werden. Z. B. ist durch Fallenlassen von Transportereignissen, wenn die Transportwarteschlange sich füllt, die Zustandstabelle in der Lage, weiter Guthaben- und Ankunftsereignisse aufzubrechen. Verlorene Ereignisse können durch periodisches Scannen der Zustandstabelle wieder hergestellt werden. Alternativ kann eine der N Warteschlangen, z. B. die Transportwarteschlange, groß genug gemacht werden, um alle möglichen gleichzeitigen Ereignisse zu verarbeiten, üblicher V mal N (wobei N die Anzahl der Flits in dem Eingangszwischenspeicher jedes Kanals ist).

Dispersion

[0119] Während das Zuordnen eines separaten virtuellen Kanals zu jedem virtuellen Netzwerk wie oben beschrieben eine einfache Lösung darstellt, ist sie kostspielig und weist eine begrenzte Skalierbarkeit auf. Die Anzahl der Zwischenspeicher, die in jedem Verbindungsnetzwerk-Router erforderlich sind, steigt linear mit der Anzahl von Knoten in dem System an.

Bei 512 virtuellen Netzwerken treibt die Anzahl der erforderlichen Flit-Zwischenspeicher an die physikalischen Grenzen dessen, was wirtschaftlich bei den integrierten Schaltkreisen konstruiert werden kann, die die Schallstruktur des Routers bilden. Auch verbleibt mit jedem virtuellen Kanal, der ausschließlich einem virtuellen Netzwerk gewidmet ist, wenn ein virtuelles Netzwerk unbenutzt ist, der virtuelle Kanal ungenutzt und es resultiert eine Unterauslastung von virtuellen Kanälen über das Netzwerk.

[0120] Des Weiteren wäre es wünschenswert, die Anzahl an Flit-Zwischenspeichern für jeden virtuellen Kanal bei jedem Knoten zu erhöhen, um die Geschwindigkeit des Netzwerks zu erhöhen. Bei dem oben beschriebenen Design werden zwei Flit-Zwischenspeicher für jeden virtuellen Kanal in jedem Knoten bereitgestellt, aber die Rückmeldungen verzögern die Übertragung von Flits zwischen den Flit-Zwischenspeichern. Durch Erhöhen der Anzahl von Flit-Zwischenspeichern pro virtuellem Kanal pro Knoten können die Rückmeldungen parallel mit den Übertragungen durchgeführt werden und die Geschwindigkeit des Systems kann erhöht werden.

[0121] Um die Anzahl an Zwischenspeichern zu reduzieren, und damit die Kosten der Schaltstruktur, und um für eine größere Skalierbarkeit, Geschwindigkeit und Auslastung zu sorgen, können virtuelle Netzwerke mit überlappenden Zwischenspeicherzuordnungen durch gemeinsames Verwenden von virtuellen Kanälen konstruiert werden. Auf diese Weise kann die Anzahl der virtuellen Kanäle, die erforderlich ist, um alle virtuellen Netzwerke zu bedienen, erheblich reduziert werden. Andererseits ist es für das System, um sich dem Verhalten eines Crossbar-Switches anzunähern, wichtig, dass der Verlust eines gemeinsam verwendeten virtuellen Kanals aufgrund der Überlastung in einem virtuellen Netzwerk die Übertragungen eines anderen virtuellen Netzwerks nicht blockiert. Demgemäß muss jedes virtuelle Netzwerk Zugriff auf mehrere virtuelle Kanäle an jedem Knoten haben und es ist wichtig, dass für zwei beliebige virtuelle Netzwerke x und y das virtuelle Netzwerk x Zugriff auf einen virtuellen Kanal hat, auf den das virtuelle Netzwerk y keinen Zugriff aufweist.

[0122] Das System von gemeinsam benutzten virtuellen Kanälen kann unter Verwendung von Dispersions- bzw. Verteilungscodes implementiert werden, die die Zuordnung von virtuellen Kanälen über die virtuellen Netzwerke verteilen. Man betrachtet z. B. ein Netzwerk mit M Knoten (und damit M virtuellen Netzwerken), die in jedem Knoten N virtuelle Kanäle gemeinsam benutzen. Jedem virtuellen Netzwerk j wird ein Verteilungscode zugeordnet, ein N -Bit Bitvektor, der bestimmt, welchen der N virtuellen Kanäle über das physikalische Netzwerk hinweg es benutzen darf. D. h. der Vektor enthält eine 1 an jeder Bitposition, die einem erlaubten virtuellen Kanal entspricht

und Nullen an allen anderen Positionen. Die Verteilungscodes müssen zugeordnet sein, so dass für jedes Paar an virtuellen Netzwerken x und y der x entsprechende Bitvektor eine 1 an mindestens einer Position enthält, an dem der y entsprechende Bitvektor eine Null enthält.

[0123] Wenn in Paket an einem Knoten in einem Netzwerk ankommt, das Dispersions- bzw. Verteilungs-Routing anwendet, wird der virtuelle Kanal, der für den nächsten Sprung bzw. Hop auf der Route des Pakets verwendet werden soll, durch Überschneiden zweier Datensätze bestimmt, die als Bit-Vektoren dargestellt sind. Ein Dispersionscodevektor A beschreibt den Satz von virtuellen Kanälen, den das Paket benutzen darf (d. h. er dient als ein Kanal-Allokationsvektor), und ein Belegt-Vektor B beschreibt den Satz an verfügbaren Kanälen.

[0124] Wie in [Fig. 15](#) dargestellt ist jedes virtuelle Netzwerk mit einem N -Bit Dispersionscodevektor A verbunden. Hier ist N die Anzahl an virtuellen Kanälen die auf jedem physikalischen Kanal in dem Netzwerk gebündelt sind. Der zu einem bestimmten virtuellen Netzwerk x gehörige Dispersionscodevektor A weist C_x Bits auf, die gesetzt sind, um die Untermenge an virtuellen Kanälen anzuzeigen, auf denen x geroutet werden darf. Die Dispersionscodevektoren für zwei virtuelle Netzwerke x und y haben höchstens S Bits gemeinsam, wobei $S < C_x$ und C_y ist.

[0125] Die Dispersionscodevektoren für sämtliche virtuelle Netzwerke können in einer Tabelle gespeichert werden, wie in [Fig. 15](#) dargestellt. Wenn es jedem virtuellen Netzwerk erlaubt ist, auf einem relativ großen Anteil der virtuellen Kanäle geroutet zu werden, wird die Tabelle am effizientesten direkt als Bildvektoren codiert wie in der Fig. dargestellt. Andererseits werden, wenn jedes virtuelle Netzwerk auf einen kleinen Anteil der Gesamtkanäle beschränkt ist, die Vektoren effektiver in komprimierter Form als eine Liste von Kanalindizes gespeichert. Z. B. kann der Vektor 0000100001001000 als die Liste von Indizes 3, 6, 11 dargestellt werden. Hier entsprechen 3, 6 und 11 den Positionen der 1 Bits in dem Binärvektor, wobei das Bit ganz rechts die Position Null darstellt. Die Darstellung des Vektors als Indizes erfordert 12 Bits für 34-Bit Indizes im Vergleich zu 16 Bits für den vollen Vektor. Alternativ kann auf die Tabelle komplett verzichtet werden und die Kanalallokationsvektoren können von dem virtuellen Netzwerkindex (in diesem Fall die Adresse des Zielknotens) unter Verwendung von kombinatorischer Logik abgeleitet werden. Z. B. können, wenn dem virtuellen Netzwerk ein Kanal für seine x -Koordinate, einer für seine y -Koordinate und einer für seine z -Koordinate zugeordnet ist, wie nachfolgend beschrieben, die Koordinaten der Zieladresse direkt decodiert werden, um den Dispersionscodevektor zu erzeugen. Zu jedem beliebigen Zeitpunkt wird der Zustand der virtuellen Kanäle, die zu einem

physikalischen Kanal gehören, in einem Satz von Vektoren aufgezeichnet. Ein Bit des Belegt-Vektors B wird gesetzt wenn der entsprechende virtuelle Kanal augenblicklich zugeordnet ist, um ein Paket zu tragen und, da er belegt ist, nicht verfügbar ist, um ein neues Paket zu verarbeiten.

[0126] Wenn ein Paket, das sich auf einem virtuellen Netzwerk x fortbewegt, an einem Knoten ankommt, wird eine Routing-Entscheidung getroffen, um einen Ausgangsport p auszuwählen. Innerhalb der VC (Virtual Channel) Zustandstabellenlogik **80** ([Fig. 11B](#)), die diesem Ausgangsport entspricht, wird das Komplement des Belegt-Vektors des Ausgangsports B_p anschließend bei 104 mit dem Dispersionscodevektor A_x für das virtuelle Netzwerk x UND-verknüpft, um einen Vektor von möglichen virtuellen Kanälen $V = B_p \wedge A_x$ zu erzeugen. Wenn V der Nullvektor ist (alles Nullen), ist kein Kanal verfügbar und die Anfrage wird in die Warteschlange gestellt, um dann wieder versucht zu werden, wenn ein virtueller Kanal für Anschluss p verfügbar wird. Wenn V nicht Null ist, wählt ein Arbitr 106 eines der nicht aus Nullen bestehenden Bits von V aus, der entsprechende Kanal wird dem Paket zugeteilt und das entsprechende Bit von B_p wird gesetzt, um den Kanal als belegt zu kennzeichnen. Sobald der virtuelle Kanal zugeteilt ist, findet der Transport von Bits wie oben beschrieben statt.

[0127] Über die Vermeidung von der Ausbreitung von Blockierungen, die den Datentransfer verlangsamt, auf andere virtuelle Netzwerke über die gemeinsam benutzten virtuellen Kanäle hinaus, muss beim Zuordnen von Dispersionscodes Acht gegeben werden, um kanalabhängige Stockungen zu vermeiden, die die Übertragungen zum Erliegen bringen können. Die Zuordnung von Dispersionscodes, die garantiert blockierungsfrei ist, für 1-D und 2-D Netzwerke und anschließend für ein 3-D ringförmiges Netzwerk folgen.

[0128] Man betrachte ein 1-D bidirektionales Ringnetzwerk wie in [Fig. 13](#) dargestellt. In jeder Richtung um die Schleife herum trägt die Spannweite eines virtuellen Netzwerks (VN) der Satz an physikalischen Kanälen, die von dem virtuellen Netzwerk verwendet werden. Bei einem minimalen Routing würde eine Nachricht, die z. B. am Knoten 2 ihren Ursprung hat, nicht der Strecke 2-1-6-5-4 folgen, da eine minimale Route 2-3-4 existiert, somit bedeckt die Spannweite jedes VN die Hälfte der Kanäle in dem Kreis. In [Fig. 13](#) besteht z. B. die Spannweite des VN, das dem schattiert dargestellten Knoten entspringt, in der Richtung des Urzeigersinns aus drei fett dargestellten Kanälen. Seine Spannweite in der anderen Richtung besteht aus den drei dünn gezeichneten Kanälen, die in Gegenrichtung laufen.

[0129] In Netzwerken mit einer Wurzel (Radix) k von

5 oder mehr ([Fig. 13](#) weist eine Radix von 6 auf) und einer unbeschränkten Zuordnung eines einzelnen virtuellen Kanals zu jeder Nachricht kann ein abhängiger Kreis von drei VNs mit überlappenden Spannweiten auf dem Ring zu einer Blockierung führen. Man betrachtet z. B. drei Nachrichten, die die entsprechenden Routen 1-2-3-4, 3-4-5-6 und 5-6-1-2 abdecken, wobei alle gleichzeitig angestoßen werden. Das Wormhole der ersten Nachricht würde am Knoten 1 beginnen und sich über Knoten 2 erstrecken, würde jedoch am Eingangszwischenspeicher von Knoten 3 zurückgehalten werden, da die Nachricht bei Knoten 3 sich bereits den gemeinsam benutzten virtuellen Kanal gesichert hätte. Die resultierende Verzögerung der ersten Nachricht bis zur Vervollständigung der zweiten Nachricht, die an Knoten 3 beginnt, wäre an sich akzeptabel. Die bei Knoten 3 beginnende Nachricht würde jedoch den Knoten 5 nicht passieren bis zur Vervollständigung der dritten Nachricht, und die dritte Nachricht wäre durch die Anfangsnachricht blockiert. Damit ist das weitere Vorschreiten jeder Nachricht durch die Vervollständigung einer anderen Nachricht in dem Kreislauf verhindert, die selbst durch die erste Nachricht verhindert ist. Das bedeutet, dass eine Blockierung (Deadlock) vorliegt.

[0130] Mit dem Dispersions-Routen, wo jedes Ziel ein virtuelles Netzwerk (VN) definiert, das C virtuelle Kanäle (VC) mit einer maximalen Überlappung von S virtuellen Kanälen ($0 < S < C$) zwischen einem beliebigen Paar von VNs verwenden kann, sind drei F VNs (wobei $F = \text{floor}(C/S)$; floor: nächstniedrige ganze Zahl) erforderlich, um eine blockierte Konfiguration zu erzeugen, da ein Paket auf F getrennten blockierten VNs blockieren muss, um ein Deadlock zu erzeugen.

[0131] Ausreichende Bedingung, um ein Blockieren (Deadlock) in einer einzelnen Dimension zu vermeiden, ist, dass jedes VN mindestens einen virtuellen Kanal (VC) aufweist, den es nur mit VNs teilt, die sich entweder vollständig oder überhaupt nicht überlappen. Dieses Ergebnis kann durch Zuordnen der Schleife eines VCs erreicht werden, das nur von dem VN verwendet wird, das an dieser Koordinate entspringt, was eine Zuordnung von sechs virtuellen Kanälen zu dem Netzwerk aus [Fig. 13](#) erfordert. Mit minimalem Routen jedoch können z. B. die Spiegelknoten 1 und 4, die in dem Netzwerk einander gegenüberliegen, dieselben virtuellen Kanäle gemeinsam benutzen, da die minimalen Routen zu diesen beiden Knoten niemals physikalische Kanäle gemeinsam verwenden würden. Über einen beliebigen Verbindungsabschnitt innerhalb der Schleife würde eine Nachricht, die für einen Knoten bestimmt ist, sich in einer entgegengesetzten Richtung bewegen, wie eine Nachricht, die für den Spiegelknoten bestimmt ist. Demgemäß erfordert die Schleife aus [Fig. 13](#) ein Minimum von drei virtuellen Kanälen, die jeweils von

nur den gespiegelten virtuellen Netzwerken gemeinsam benutzt werden, um eine Blockierung (Deadlock) zu vermeiden. Zusätzliche virtuelle Kanäle können anschließend der Schleife zugeordnet werden für eine gemeinsame oder nicht gemeinsame Verwendung. Mit diesem Ansatz ist jedes VN immer in der Lage, durch seine nicht gemeinsam benutzten VC (innerhalb einer Dimension) voranzuschreiten.

[0132] Es ist möglich, eine Blockierung (Deadlock) mit einer weniger restriktiven Zuordnung von VCs zu VNs zu vermeiden, da es nur notwendig ist, die Blockierung (Deadlock) an einem Punkt in dem Kreis aufzubrechen.

[0133] In einem mehrdimensionalen Netzwerk ist eine Blockierung (Deadlock) sogar möglich, wenn alle Dimensionen individuell frei von Blockierungen sind. Man betrachte den zweidimensionalen Fall, der leicht auf drei Dimensionen erweiterbar ist. Es kann sich eine Blockierung (Deadlock) bilden, wenn ein Paket, das eine NW-Drehung macht, ein Paket blockiert, das eine WS-Drehung macht, das wiederum ein Paket blockiert, das eine SE-Drehung macht, das wiederum ein Paket blockiert, das eine EN-Drehung durchführt, das wiederum das ursprüngliche Paket blockiert. Dies bildet einen Kreis (NW, WS, SE, EN). Wie von C. J. Glass und L. M. Ni, "The Turn Model for Adaptive Routing", Proceedings of the 19th International Symposium on Computer Architecture, Mai 1992, Seiten 278 bis 287, gezeigt, kann ein Kreis durchbrochen werden, wodurch eine Blockierung (Deadlock) vermieden wird, durch Eliminieren einer beliebigen Drehung in dem Kreis. Obwohl eine vollständige Eliminierung z. B. des Nordrandes des Kreises den Kreis durchbrechen würde, ist es ausreichend, dass lediglich die Drehung EN eliminiert wird. Eine Drehung WN oder geradeaus NN kann zugelassen werden, ohne eine Blockierung (Deadlock) zu riskieren.

[0134] Man nehme ein zweidimensionales Array an, in dem jede der X- und Y-Dimensionen frei von Blockierungen (Deadlocks) gemacht wird durch Zuordnen von virtuellen Kanälen wie oben erläutert. Z. B. würden in einem 8×8 Array $4 + 4 = 8$ virtuelle Kanäle den entsprechenden virtuellen Netzwerken zugeordnet. Die Zuordnung würde definiert durch die Koordinate der Zieladresse (mod $k/2$), wobei k die Anzahl der Knoten in der entsprechenden Dimension ist. Der (mod $k/2$)-Faktor berücksichtigt die Spiegelknoten, die VCs gemeinsam benutzen. Wenn minimales Routen verwendet wird, ist jedes VN selbst frei von Blockierungen (Deadlocks), da in jedem Quadrant um die Zielknoten herum nur zwei Richtungen und deshalb nur zwei von acht möglichen Drehungen verwendet werden. Dies ist in [Fig. 14](#) veranschaulicht. Im Bereich NE des Zielknotens bewegen sich die Pakete nur nach S und W und damit sind nur SW- und WS-Drehungen erlaubt. Dies ist eine Drehung von

dem Kreis im Urzeigersinn und eine Drehung von dem Kreis gegen den Urzeigersinn. Wenn die VNs jedoch VCs gemeinsam benutzen, kann eine Blockierung (Deadlock) auftreten, da die Drehungen, die von einem VN fehlen, in anderen VNs vorhanden sein können, die denselben VC gemeinsam benutzen.

[0135] Wie oben erläutert können die Kreise, die in einer Blockierung (Deadlock) resultieren durch nicht erlauben einer Drehung in dem Kreis eliminiert werden. Um sowohl Kreise im Uhrzeigersinn als auch gegen den Uhrzeigersinn zu durchbrechen muss eine Drehung in jeder Richtung eliminiert werden. Weiterhin muss sichergestellt werden, dass eine in einem virtuellen Kanal eliminierte Drehung nicht durch dieselbe Drehung in einem anderen virtuellen Kanal ersetzt wird, um den Kreis zu vervollständigen. Da das Nichterlauben von beiden Drehungen eines Quadranten in einem VN verhindern würde, dass das Ziel von diesem Quadranten erreicht wird, müssen die nicht erlaubten Drehungen aus unterschiedlichen Quadranten sein. Obwohl eingeschränkt kann das Ziel von einem beliebigen Quadranten erreicht werden, wenn lediglich eine einzelne Drehung dieses Quadranten nicht erlaubt wird.

[0136] Demgemäß ist ein ausreichendes Verfahren zum Verhindern einer interdimensionalen Blockierung (Deadlock), (1) die Dimensionen einzeln blockierungsfrei zu machen und (2) zu verlangen, dass (a) jedes VN eine der vier Drehungen der CW- und CCW-Richtungen in unterschiedlichen Quadranten nicht erlaubt, und (b) dass jedes VN mindestens einen VC aufweist, der nur mit dem VNs gemeinsam genutzt wird, das dieselbe Drehung nicht erlaubt. Dies ist ziemlich restriktiv, da es zwei der vier Quadranten um den Zielknoten herum zwingt, in Dimensionsreihenfolge zu routen. Wenn z. B. die NE-Drehung eliminiert würde, müssten Nachrichten zum Ziel aus dem SW zuerst zu E und anschließend zu N geroutet werden, da lediglich die EN-Drehung verfügbar bleiben würde.

[0137] Eine Strategie, die ein flexibleres Routing zulässt, jedoch in Bezug auf VCs kostspieliger ist, ist es, zwei VNs mit jedem Zielknoten zu verknüpfen, wobei jedes sämtliche Drehungen für einen einzelnen Quadranten nicht erlaubt, wobei die Quadranten unterschiedlich sind für die beiden VNs. Z. B. würde ein VN für alle Quadranten außer dem NW-Quadranten die SE- und ES-Drehungen nicht erlauben und eines für alle Quadranten außer den SE-Quadranten würde die NW- und WN-Drehungen nicht erlauben. VNs aus jeder Klasse können anschließend VCs gemeinsam benutzen ohne Einschränkung so lange sie ohne Blockierung (Deadlock) in jeder Dimension unabhängig voneinander bleiben. Zum Ausgleich wurden in dem Beispiel diagonale Quadranten ausgewählt, es können jedoch beliebige zwei Quadranten für die beiden VNs nicht zugelassen werden.

[0138] Ein durchführbares Verfahren zum Zuordnen von VCs in zwei Dimensionen ist wie folgt:

1. Jedem Ziel werden zwei virtuelle Netzwerke zugeordnet, eines, das SE- und ES-Drehungen nicht erlaubt, und eines, das NW- und WN-Drehungen nicht erlaubt.
2. Jedem VN wird ein VC zugeordnet, der mit der x-Koordinate des Ziels des VNs verbunden ist ($\text{mod } k_x/2$), wobei k_x die Anzahl von Knoten in der x-Dimension ist. Die Zuordnung dieses VC gewährleistet ein nicht Überlappen und damit eine Freiheit von Blockierungen in einer einzelnen Dimension in der x-Dimension.
3. Jedem VN wird ein VC zugeordnet, das mit der y-Koordinate des Ziels verbunden ist ($\text{mod } k_y/2$). Dies gewährleistet die Freiheit von Blockierungen (Deadlocks) einer einzelnen Dimension in der y-Dimension.
4. Beliebige zusätzliche VC-Paare werden willkürlich zugeordnet abhängig von der Einschränkung, dass nicht mehr als S VCs von beliebigen zwei Zielen gemeinsam benutzt werden.
5. Die Routing-Tabellen werden so aufgebaut, dass Knoten in dem NW-Quadranten eines Ziels auf das VN beschränkt sind, das NW/WN nicht erlaubt und so dass Knoten in dem SE-Quadranten auf das andere VN beschränkt sind. Knoten in den NE- und SW-Quadranten können irgendein VN benutzen.

[0139] Als ein Beispiel für ein 2-D Netzwerk von 64-Knoten (8×8) erfordert diese Zuordnung ein Minimum von 8 VC-Paaren (16 VCs insgesamt).

[0140] Die Anzahl der VCs, die verfügbar ist, hängt sowohl vom Zwischenspeicherplatz, der verfügbar ist, als auch von der Anzahl der Flit-Zwischenspeicher pro VC in jedem Knoten ab. Das Erhöhen der Geschwindigkeit des Systems mit erhöhten Flit-Zwischenspeichern pro Knoten vermindert die verfügbaren virtuellen Kanäle. Über das Minimum an VCs hinaus, die erforderlich sind, wie in den Schritten 2 und 3 oben identifiziert, können beliebige verfügbare virtuelle Kanäle bestimmten VNs zugeordnet werden, wodurch S und die Blockierwirkung in einem VN auf ein anderes VN begrenzt wird, oder sie können gemeinsam verwendet werden, um die Auslastung von VCs über das Netzwerk hinweg zu verbessern. Man nehme z. B. an, dass zwei VCs jedem VN in den Schritten 2 und 3 oben zugeordnet sind, und 20 Kanäle für die Zuordnung nach diesen Schritten verfügbar bleiben. Alle 20 können von allen VNs für $C = 22$, $S = 21$ gemeinsam benutzt werden, oder jeder der 20 kann für die ausschließliche Verwendung eines einzelnen VN für $C = 3$ und $S = 1$ zugeordnet werden.

[0141] Die Zunahme des gemeinsamen Benutzens erhöht den Wert S. Das Verhältnis C/S ist eine Angabe dafür, wie viele VNs selbst blockiert werden können aufgrund Überlastung, ohne andere VNs zu be-

einflussen. Je größer das Verhältnis, desto geringer die Wirkung der Überlastung in einem VN auf andere VNs.

[0142] Um diesen Ansatz auf drei Dimensionen auszuweiten müssen wir zusätzliche Drehungen ausschließen, um 3-D interdimensionale Kreise zu vermeiden. Wir können dies jedoch mit nur zwei VNs pro Ziel wie oben erreichen. Ein VN schließt die Drehungen aus, die mit dem NWU (Nord, West, oben (ab)) Oktanten verbunden sind (SE, ES, SD, DS, ED, DE), während das andere die Drehungen ausschließt, die mit dem SED-Oktanten (Süd, Ost (east), unten (down)) verbunden sind.

[0143] Ein beispielhaftes 1024-Knoten-Netzwerk, das als $8 \times 8 \times 16$ organisiert ist, benötigt ein Minimum von $4 + 4 + 8 = 16$ VC-Paaren (32 VCs), um ein VC-Paar jedem symmetrisch gespiegelten Paar von Ebenen in dem Netzwerk zuzuordnen.

[0144] Wenn ein einzelnes Ziel eine übermäßige Verkehrsmenge empfängt, laufen alle mit seinen beiden VNs verbundenen VCs in die Sättigung und stauen sich zur Quelle zurück. In erster Näherung ist es, als wenn diese VCs aus dem Netzwerk entfernt worden wären, obwohl in Wirklichkeit die Sättigung Knoten weiter entfernt von dem Ziel weniger wahrscheinlich beeinflusst. Mit der oben vorgeschlagenen Kanalzuordnung, wo jeder Zielknoten zwei VNs mit jeweils drei VCs (einer pro Dimension) aufweist, hat ein VN zu einem Knoten, der exakt eine Koordinate mit dem gesättigten Ziel gemeinsam benutzt, vier verbleibende VCs, auf denen geroutet werden kann. Ein VN zu einem Knoten, Deflexionsrouting.

[0145] Deflexionsrouting (Ablenkungsrouting) ist ein weiteres Verfahren, um Verkehr, der für unterschiedliche Strukturausgänge bestimmt ist, im Wesentlichen nicht blockierend zu machen. Beim Deflexionsrouting wird allen Paketen erlaubt, virtuelle Kanäle ohne Einschränkung gemeinsam zu benutzen. Wenn jedoch ein Paket blockiert wird, es, eher als auf den angefragten virtuellen Kanal zu warten, dass dieser verfügbar wird, fehlgeleitet oder "umgeleitet" zu dem Paketspeicher der Leitungsschnittstelle des vorliegenden Struktur-Routers. Zu einem späteren Zeitpunkt wird es wieder in die Struktur eingeführt. Da ein Paket, das für den Strukturausgang A bestimmt ist, niemals blockieren darf, kann es ein Paket, das für einen Strukturausgang B bestimmt ist, nicht unbegrenzt verzögern.

[0146] Deflexionsrouting hat mehrere Eigenschaften, die es weniger erstrebenswert machen, um eine Isolation zwischen Paketen zu erzielen, die für unterschiedliche Ausgänge bestimmt sind. Erstens bietet das Deflexionsrouting keinen Gegendruck. Wenn ein Ausgang verstopft wird, werden die Pakete, die für diesen Ausgang bestimmt sind, einfach umgeleitet

und die Struktureingänge, die Pakete an den verstopften Ausgang senden, verbleiben in Unkenntnis. Zweitens gibt es, obwohl keine Blockierung vorhanden ist, eine signifikante Beeinträchtigung unter den Paketen, die für unterschiedliche Ausgänge bestimmt sind. Wenn ein Ausgang A verstopft ist, werden die A angrenzenden Verbindungsabschnitte stark ausgelastet und ein für den Ausgang B bestimmtes Paket, das sich über einen dieser Verbindungsabschnitte fortbewegt, hat eine sehr hohe Wahrscheinlichkeit, umgeleitet zu werden. Drittens erhöht die Verwendung des Deflexionsroutings erheblich die Bandbreitenanforderungen des Paketspeichers, da dieser Speicher ausreichend Bandbreite aufweisen muss, um umgeleitete Pakete und ihre Wiedereinleitung zusätzlich zu ihrem normalen Eingang und Ausgang zu verarbeiten. Schließlich ist das Deflexionsrouting durch die endliche Größe des Paketspeichers auf jeder Leitungsschnittstelle begrenzt. Bei sehr hoher Stauung, was bei IP-Routern häufig auftritt, kann der Paketspeicher vollständig mit umgeleiteten Paketen gefüllt sein. Wenn dies auftritt, müssen Pakete fallengelassen werden, um eine Störung und möglicherweise eine Blockierung (Deadlock) zu vermeiden.

[0147] Während diese Erfindung insbesondere unter Bezugnahme auf ihre bevorzugten Ausführungsformen gezeigt und beschrieben wurde, wird der Fachmann verstehen, dass verschiedene Änderungen in der Form und in Details darin durchgeführt werden können, ohne vom Wesen und dem Umfang der Erfindung abzuweichen, wie sie durch die beigefügten Ansprüche definiert ist. Der Fachmann wird erkennen oder in der Lage sein, unter Verwendung von nicht mehr als routinemäßigem Experimentieren viele äquivalente für die bestimmten Ausführungsformen der hierin speziell beschriebenen Erfindung festzustellen. Solche Äquivalente sind mit Absicht vom Umfang der Ansprüche umfasst.

[0148] Z. B. ist die in Verbindung mit den [Fig. 11A](#), [Fig. 11B](#) und [Fig. 12](#) beschriebene Ereignis-gesteuerte Allokationslogik für die Verwendung in einem Internet-Schaltstruktur-Router wie dem in [Fig. 8](#) dargestellten geeignet. Es versteht sich, dass die Ereignis-gesteuerte Allokationslogik ebenfalls für die Verwendung in einem Multicomputer-Router geeignet ist. Z. B. bildet unter Bezugnahme auf [Fig. 8](#), unter Verwendung einer Multicomputer-Schnittstelle als Leitungsschnittstellenschaltung **48** in Kombination mit der Ereignis-gesteuerten Allokationslogik einen Multicomputer-Router für ein Multicomputer-System wie das in [Fig. 4](#) dargestellte.

[0149] Weiterhin versteht es sich, dass die Ereignis-gesteuerte Allokationslogik zur Zuordnung von physikalischen Eingabekanälen zu physikalischen Ausgabekanälen direkt geeignet ist. Vorzugsweise wird eine einfache Kopie der Allokationslogik verwendet.

Die Logik wird durch das Auftreten eines Ereignisses aktiviert.

[0150] Darüber hinaus versteht es sich, dass Teile des Zustandsvektors für die Zustandstabelle **80** der virtuellen Kanäle (siehe [Fig. 12](#)) beschrieben worden sind, dass sie einzelne Bits zum Anzeigen von besonderen Informationen umfassen wie z. B. die Belegt- oder Warte-Information. Anstatt solcher Bits können andere Strukturen verwendet werden wie z. B. skalare Zustandfelder, die die Information codieren.

[0151] In Verbindung mit der in den [Fig. 11A](#), [Fig. 11B](#) und [Fig. 12](#) beschriebenen Ereignis-gesteuerten Allokationslogik versteht es sich, dass jeder physikalische Eingabekanal von einer Vielzahl von virtuellen Eingabekanälen gemeinsam benutzt wird, und dass jeder physikalische Ausgabekanal von vielfachen virtuellen Ausgabekanälen gemeinsam verwendet wird. Die Allokationslogik ist geeignet zum Bereitstellen eines einzelnen virtuellen Kanals für jeden physikalischen Kanal. In solch einem Fall wird jeder physikalische Eingabekanal lediglich von einem virtuellen Eingabekanal verwendet, und jeder physikalische Ausgabekanal wird lediglich von einem virtuellen Ausgabekanal verwendet. Als solche erzeugt die Zustandstabellelogik im Wesentlichen Zuordnungen, die physikalische Eingabekanäle mit physikalischen Ausgabekanälen in Verbindung bringen.

Patentansprüche

1. Internet-Router zum Koppeln an eine Vielzahl von Internet-Übertragungsabschnitten (**46**, **56**), wobei der Router Datenpakete von den Internet-Übertragungsabschnitten erhält, Header-Informationen in den Datenpaketen analysiert, um die Datenpakete zu routen, und wobei der Router die Datenpakete auf den Internet-Übertragungsabschnitten weiterleitet, wobei der Router eine Schaltstruktur ([Fig. 7](#)) von Strukturverbindungsabschnitten aufweist, die durch Strukturknoten (N) verbunden sind, wobei die Anzahl der Strukturverbindungsabschnitte zu jedem Strukturknoten kleiner als die Anzahl der vom Internet-Router bedienten Internet-Übertragungsabschnitte ist, wobei die Strukturverbindungsabschnitte und -knoten eine Datenkommunikation zwischen den Internet-Übertragungsabschnitten über einen oder mehrere Sprünge durch die Struktur bereitstellen; **dadurch gekennzeichnet**, dass die Strukturknoten Struktur-Router (**54**) sind, die für eine Zwischenspeicher-Arbitrationslogik zwischen konkurrierenden Paketen sorgen.

2. Internet-Router nach Anspruch 1, wobei eine Leitungsschnittstelle zu jedem Internet-Übertragungsabschnitt die Header-Informationen in den vom Internet-Übertragungsabschnitt empfangenen Da-

tenpaketen analysiert, um einen Ausgangs-Internet-Übertragungsabschnitt über ein Internet-Routing-Protokoll zu identifizieren, und um über ein Struktur-Routing-Protokoll eine Routing-Strecke durch die Struktur zum identifizierten Ausgangs-Internet-Übertragungsabschnitt zu bestimmen.

3. Internet-Router nach Anspruch 2, wobei die Leitungsschnittstelle die Routing-Strecke durch die Struktur durch Einfügen einer Verbindungsstrecken- definition jeder nachfolgenden Verbindungsstrecke in der Routing-Strecke in einen Header definiert, wobei jeder Struktur-Router entlang der Routing-Strecke eine zugehörige Verbindungsstrecken- definition aus dem Header zum Weiterleiten von nachfolgenden Paketsegmenten speichert.

4. Internet-Router nach Anspruch 1, wobei zwischen Sprüngen auf den Strukturverbindungsabschnitten Paketsegmente in den Struktur-Routern an Speicherorten (62) gespeichert werden, die virtuellen Kanälen zugeordnet sind, die den Ziel-Internet-Übertragungsabschnitten entsprechen.

5. Internet-Router nach Anspruch 1, wobei die Anzahl der durch den Internet-Router bedienten Internet-Übertragungsabschnitten mindestens eine Größenordnung größer ist als die Anzahl der Strukturverbindungsabschnitte zu jedem Struktur-Router, und wobei die Anzahl der virtuellen Kanäle pro Struktur-Router wesentlich größer als die Anzahl der Verbindungsabschnitte zum Struktur-Router ist.

6. Internet-Router nach Anspruch 1, wobei jeder Struktur-Router eine Vielzahl von Zwischenspeichern (62) zum Definieren von virtuellen Kanälen aufweist, die sich die Strukturverbindungsabschnitte teilen, wobei die virtuellen Kanäle und Verbindungsabschnitte virtuelle Netze zwischen den Internet-Routeingängen und -ausgängen bilden, in denen eine Überlastung in einem virtuellen Netzwerk im Wesentlichen die Pakete nicht blockiert, die durch andere virtuelle Netze fließen.

7. Internet-Router nach Anspruch 1, wobei eine Arbitrationslogik in jedem Struktur-Router durchgeführt wird, um ein Paket einem virtuellen Kanal zur Ausgabe aus dem Struktur-Router zuzuordnen und um einen virtuellen Kanal einem Ausgangs-Strukturverbindungsabschnitt vom Struktur-Router zuzuordnen.

8. Internet-Router nach Anspruch 1, wobei jeder Struktur-Router einen Kreuzschienenwähler (crossbar switch, 66) aufweist.

9. Internet-Router nach Anspruch 1, wobei jeder Struktur-Router Eingabezwischenspeicher (62) zum Empfangen von Datenpaketen aus entsprechenden Struktur-Routern und einen Internet-Übertragungs-

abschnitt aufweist.

10. Internet-Router nach Anspruch 1, wobei jeder Struktur-Router eine Ausgabe-Steuereinrichtung (Fig. 11B) zum Weiterleiten von Daten entlang eines Ausgangs-Strukturverbindungsabschnitts bei Empfang eines Hinweises aufweist, dass ein Eingabezwischenspeicher am gegenüberliegenden Ende des Strukturverbindungsabschnitts verfügbar ist.

11. Internet-Router nach Anspruch 1, wobei die Struktur ein direktes Netzwerk ist.

12. Internet-Router nach Anspruch 1, wobei die Struktur ein dreidimensionales Ringnetzwerk (Fig. 7) ist.

13. Verfahren zum Routen von Datenpaketen zwischen Internet-Übertragungsabschnitten bestehend aus:

Analysieren von Header-Informationen in den Datenpaketen, um die Datenpakete zu Ausgangs-Internet-Übertragungsabschnitten zu routen; und Durchleiten der Datenpakete durch ein Multi-hop-Strukturnetzwerk von Strukturknoten hin zu den Ausgangs-Internet-Übertragungsabschnitten; dadurch gekennzeichnet, dass die Strukturknoten Struktur-Router sind, die die Zwischenspeicher unter den konkurrierenden Paketen mittels Arbitrationslogik vermitteln.

14. Verfahren nach Anspruch 13, das weiterhin die Analyse der Header-Informationen in den Datenpaketen an einer Leitungsschnittstelle aufweist, um einen Ausgangs-Internet-Übertragungsabschnitt über ein Internet-Routing-Protokoll zu identifizieren und über ein Struktur-Routing-Protokoll eine Routing-Strecke durch das Strukturnetzwerk zum identifizierten Ausgangs-Internet-Übertragungsabschnitt zu bestimmen.

15. Verfahren nach Anspruch 14, wobei die Verbindungsschnittstelle die Routing-Strecke durch die Struktur durch Einfügen einer Verbindungsstrecken- definition jeder nachfolgenden Verbindungsstrecke in der Routing-Strecke in einen Header definiert, wobei jeder Struktur-Router entlang der Routing-Strecken eine Verbindungsabschnittsdefinition aus dem Header zum Weiterleiten von nachfolgenden Paketsegmenten speichert.

16. Verfahren nach Anspruch 13, wobei zwischen den Sprüngen auf den Strukturverbindungsabschnitten Segmente in Struktur-Routern an Speicherorten gespeichert werden, die virtuellen Kanälen zugeordnet sind, die den Ziel-Internet-Übertragungsabschnitten entsprechen.

17. Verfahren nach Anspruch 13, wobei jeder Struktur-Router eine Vielzahl von Zwischenspeichern

zum Definieren von virtuellen Kanälen aufweist, die sich Strukturverbindungsabschnitte teilen, wobei die virtuellen Kanäle und -verbindungsabschnitte virtuelle Netzwerke zwischen den Internet-Routereingängen und -ausgängen bilden, in denen die Überlastungen einem virtuellen Netzwerk im Wesentlichen dem Nicht-Blockieren von Paketen entspricht, die durch andere virtuelle Netzwerke fließen.

18. Verfahren nach Anspruch 13, wobei eine Arbitrationslogik bei jedem Struktur-Router durchgeführt wird, um ein Paket einem virtuellen Kanal zur Ausgabe aus der Struktur zuzuordnen und einen virtuellen Kanal einem Ausgangs-Strukturverbindungsabschnitt vom Struktur-Router zuzuordnen.

19. Verfahren nach Anspruch 13, wobei die Datenpakete zwischen den Strukturverbindungsabschnitten über einen Kreuzschienenwähler geroutet werden.

20. Verfahren nach Anspruch 13, wobei die Datenpakete zwischen den Internet-Übertragungsabschnitten über ein dreidimensionales ringförmiges direktes Netzwerk geroutet werden.

Es folgen 11 Blatt Zeichnungen

Anhängende Zeichnungen

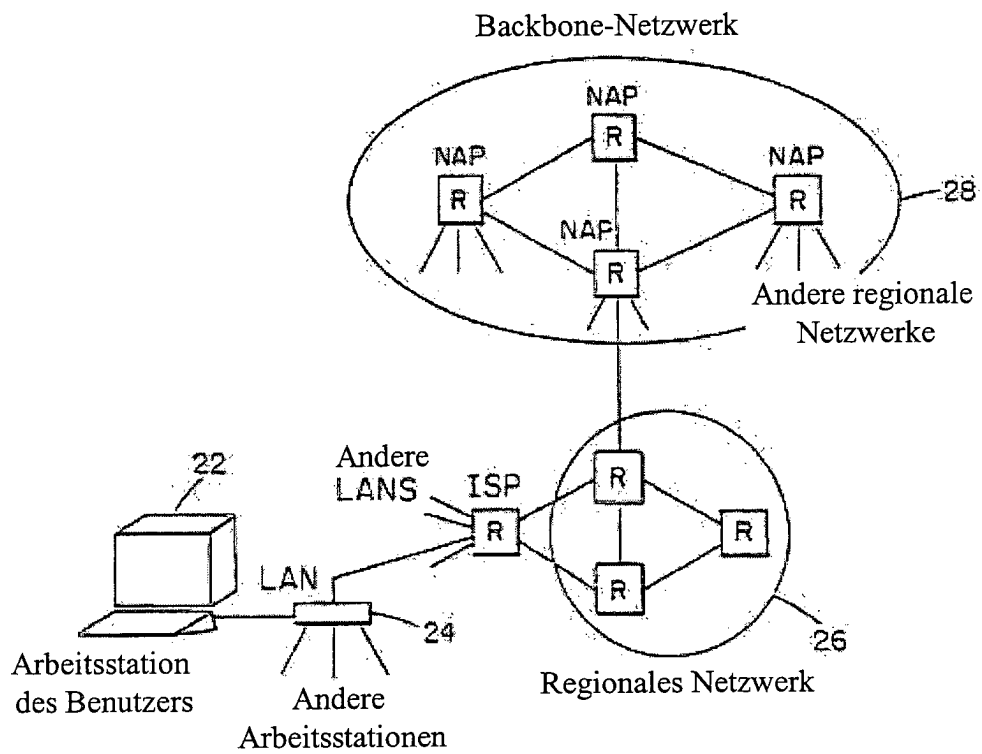


FIG. 1

STAND DER TECHNIK

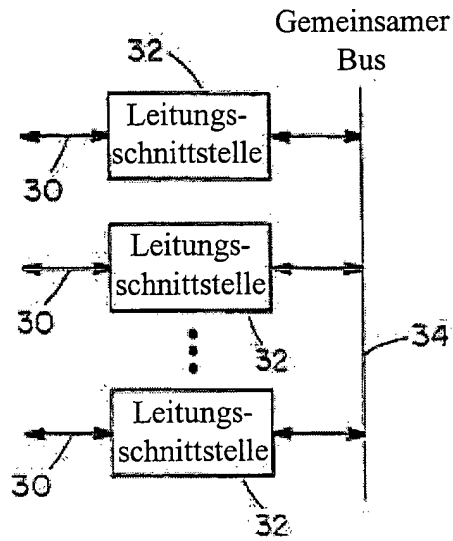


FIG. 2
STAND DER TECHNIK

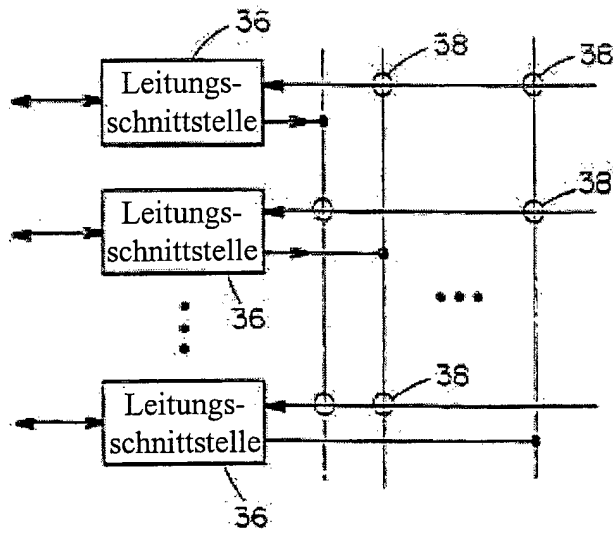


FIG. 3
STAND DER TECHNIK

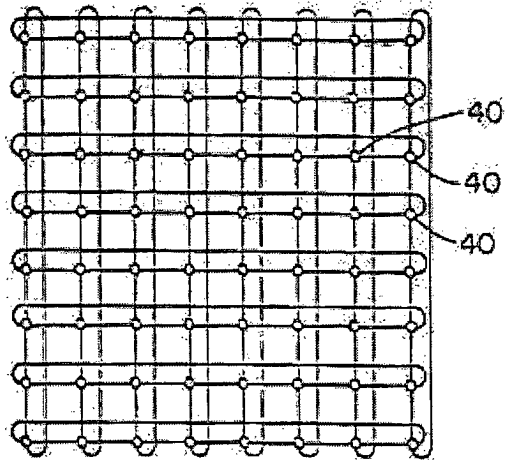


FIG. 4

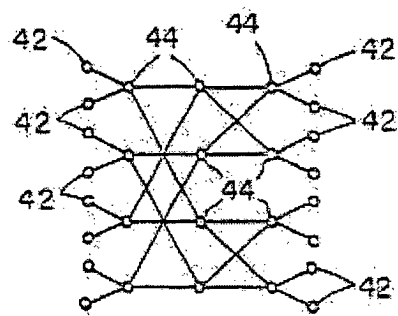


FIG. 5

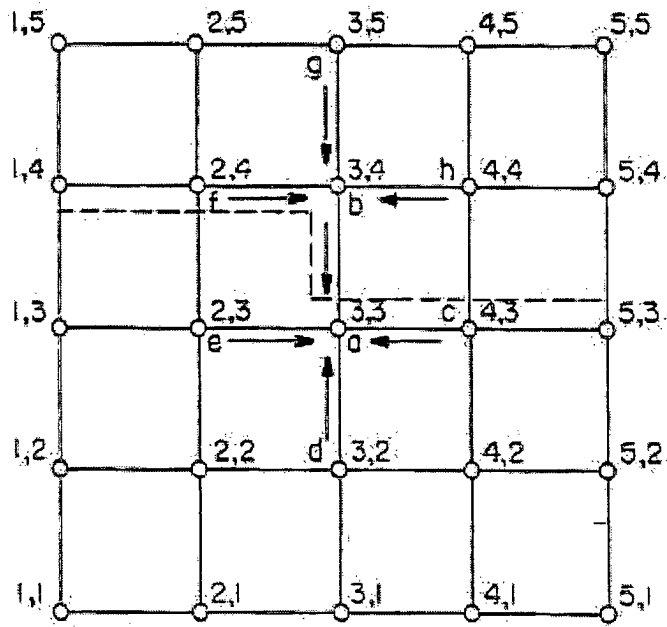


FIG. 6

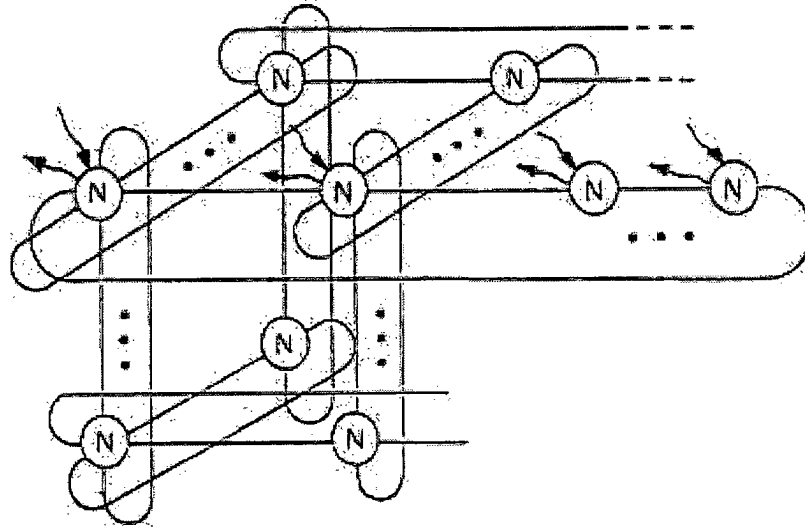


FIG. 7

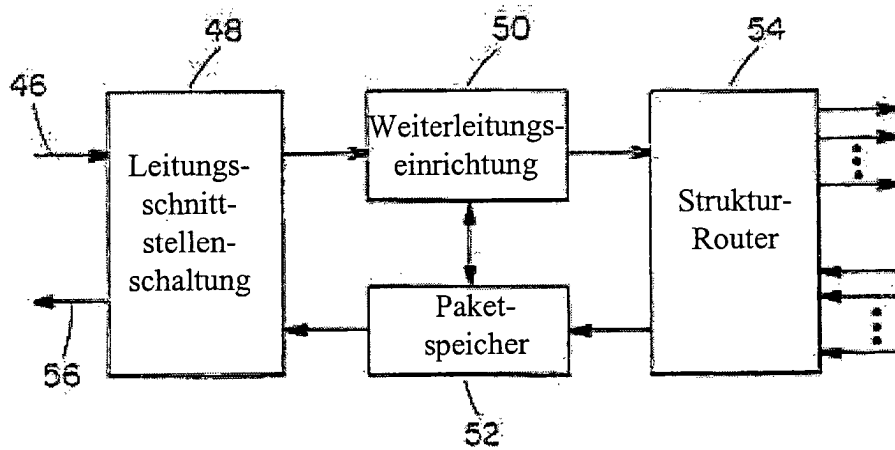


FIG. 8

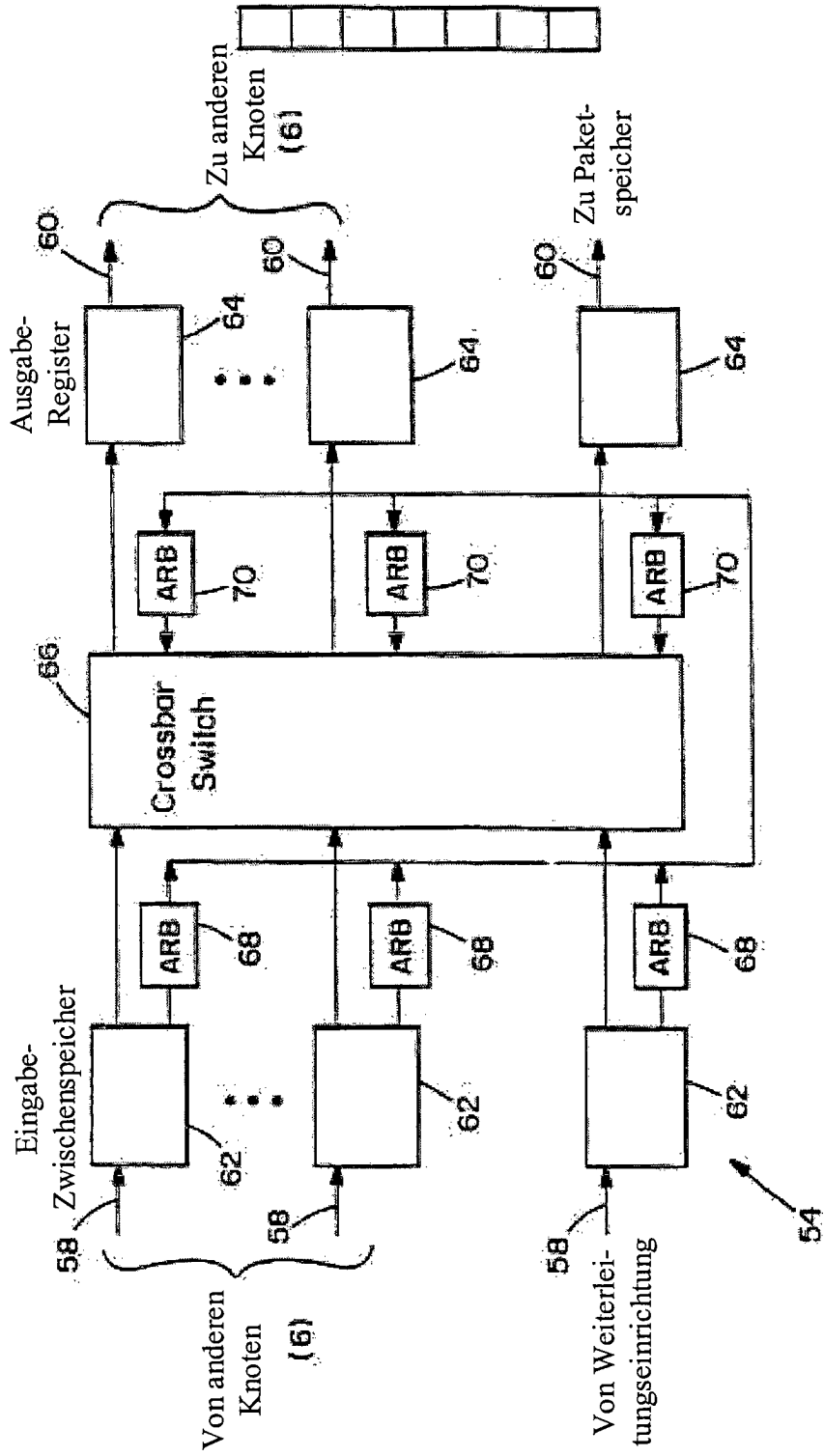


FIG. 9

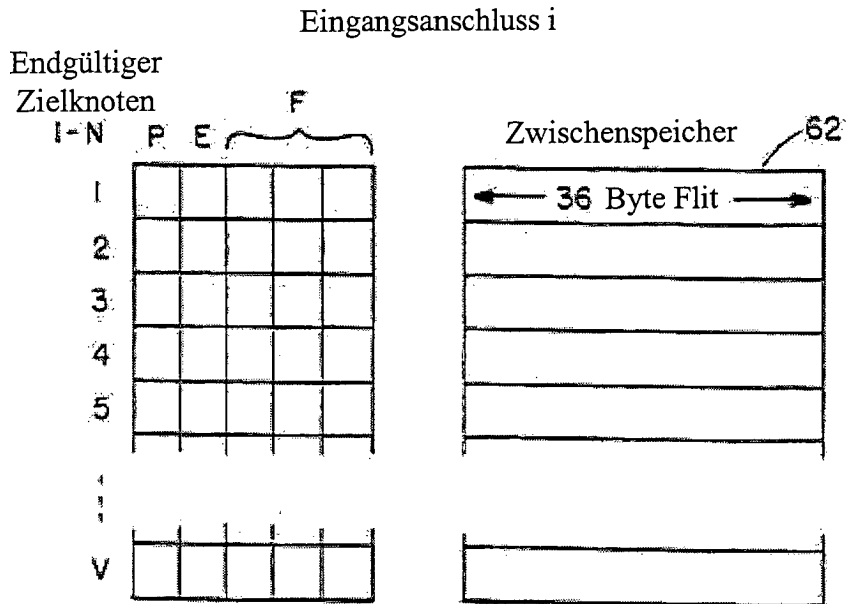


FIG. 10A

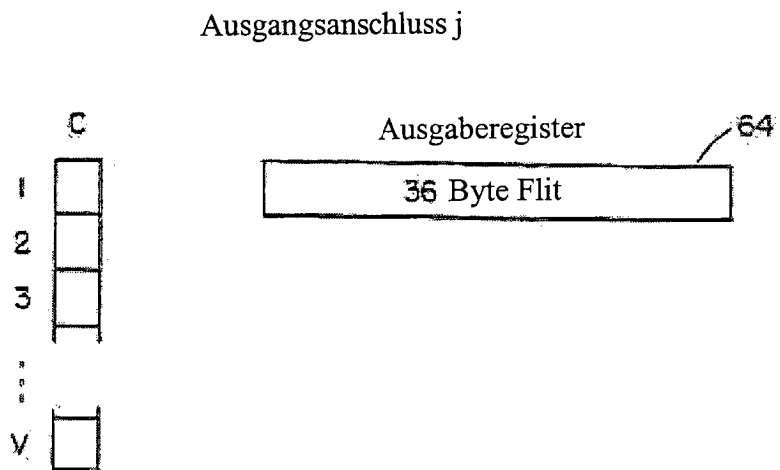


FIG. 10B

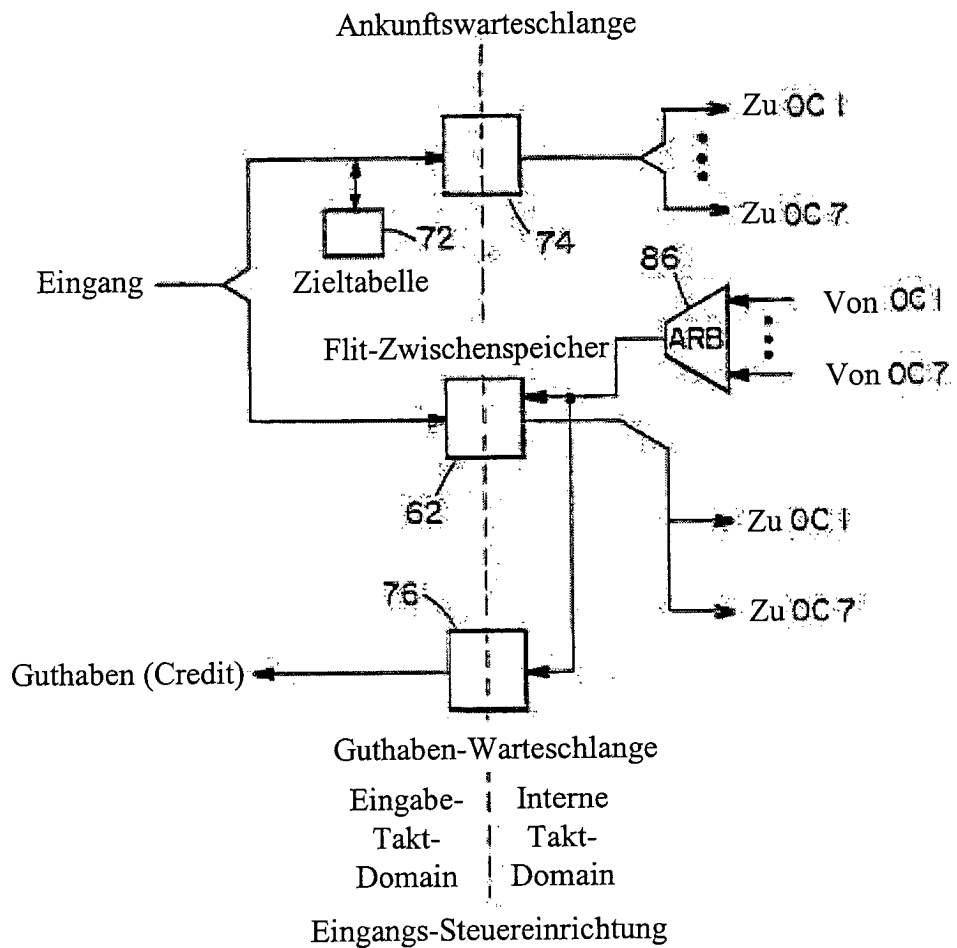


FIG. IIA

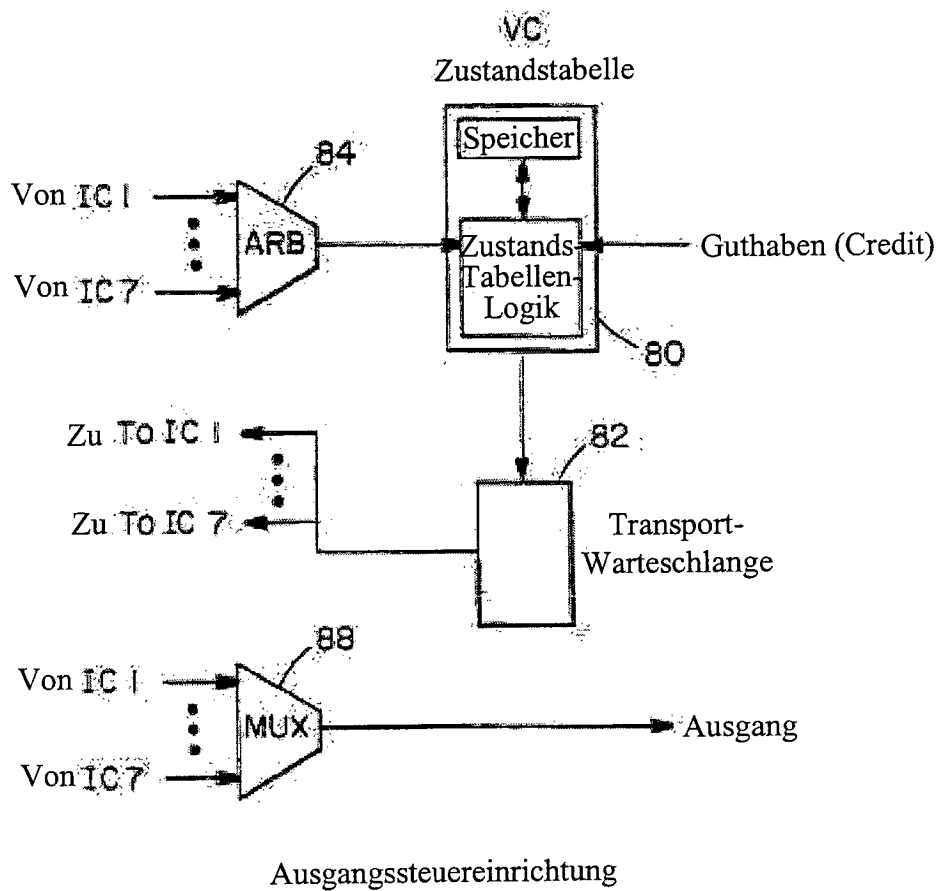



FIG. 11B

Für Ausgabe K	Allokations- Status (B = 0,1,2)	Eingangs- Steuerein- richtung (I = 1 to 7)	Wartende Eingangs- Steuereinrichtung (W)		Gutahben (Credit) (C)	Vorhandene Flits (P)
						
1						
2						
3						
⋮						
⋮						
⋮						
V						

Zustands-
vektor
s[v,K]

FIG. 12



FIG. 13

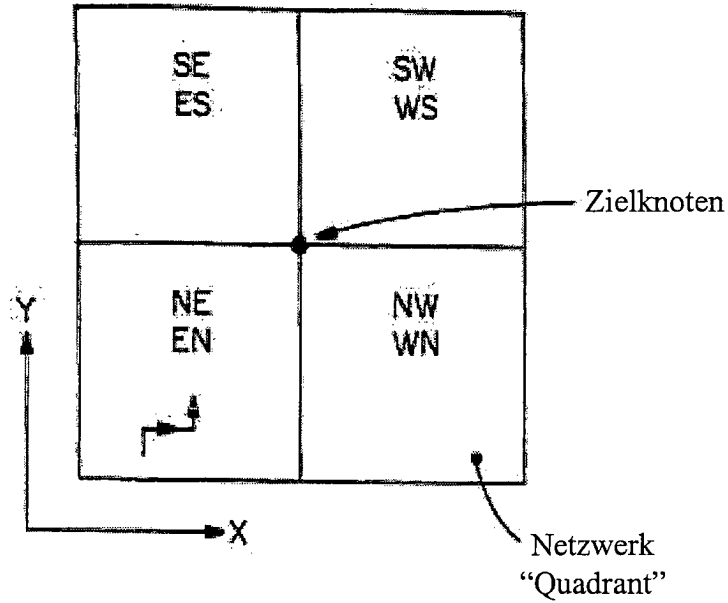


FIG. 14

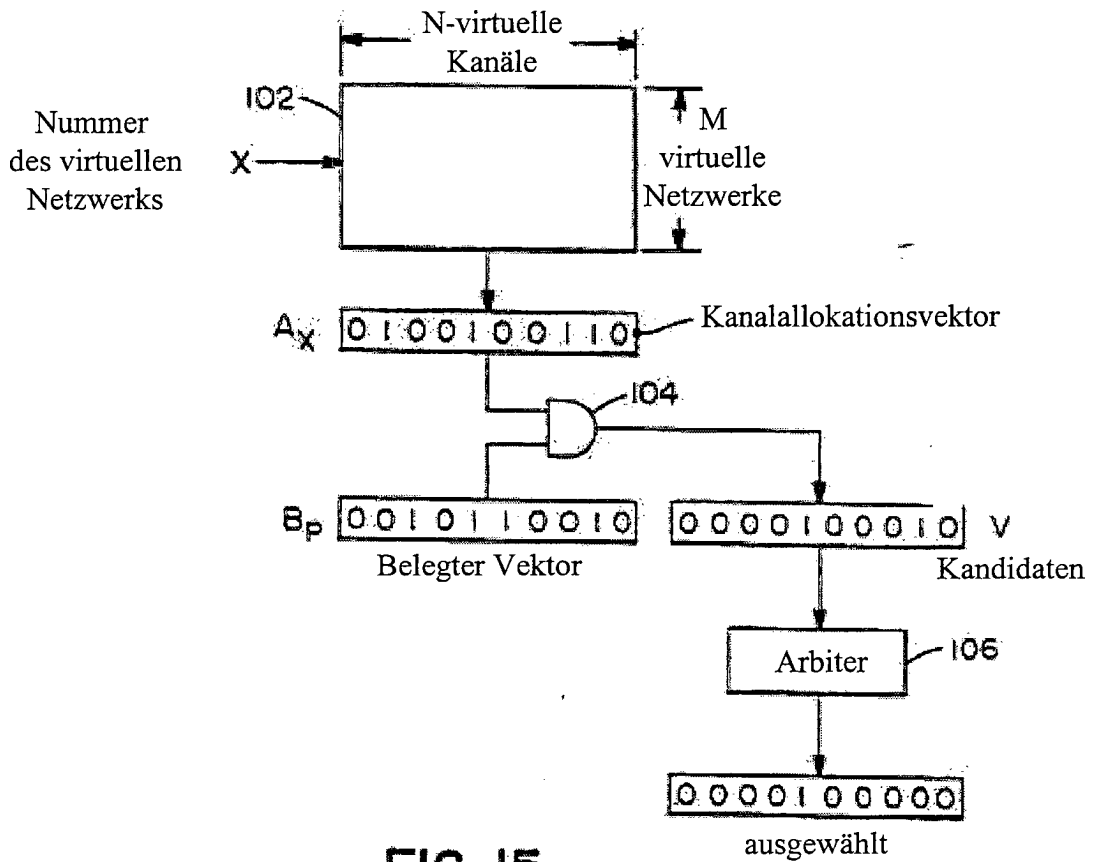


FIG. 15