(19) Deutsches Patent- und Markenamt

(10) **DE 11 2020 002 512 T5** 2022.02.17

(12) # Veröffentlichung

der internationalen Anmeldung mit der
(87) Veröffentlichungs-Nr.: **WO 2020/236301**
in der deutschen Übersetzung (Art. III § 8 Abs. 2 IntPatÜbkG)
(21) Deutsches Aktenzeichen: **11 2020 002 512.8**
(86) PCT-Aktenzeichen: **PCT/US2020/024340**
(86) PCT-Anmeldetag: **23.03.2020**
(87) PCT-Veröffentlichungstag: **26.11.2020**
(43) Veröffentlichungstag der PCT Anmeldung in deutscher Übersetzung: **17.02.2022**

(51) Int Cl.: ***H04L 47/2441*** (2022.01)

(30) Unionspriorität:

62/852,273	**23.05.2019**	**US**
62/852,203	**23.05.2019**	**US**
62/852,289	**23.05.2019**	**US**

(71) Anmelder:
Hewlett Packard Enterprise Development LP, West Houston, TX, US

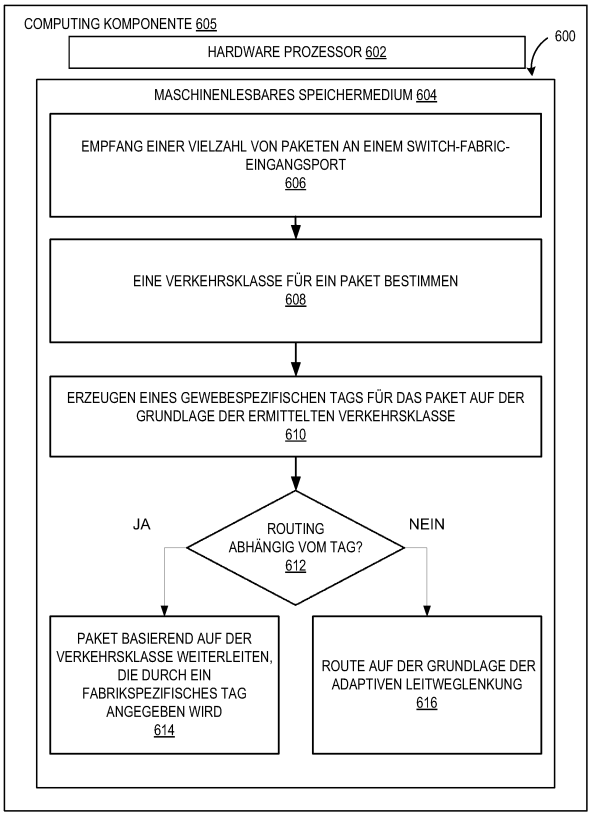
(74) Vertreter:
Prock, Thomas, Dr., EC4A 1BW London, GB

(72) Erfinder:
Ford, Anthony Michael, Fort Collins, CO, US; Beecroft, Jonathan P., Fort Collins, CO, US; Roweth, Duncan, Fort Collins, CO, US; Froese, Edwin L., Fort Collins, CO, US

Die folgenden Angaben sind den vom Anmelder eingereichten Unterlagen entnommen.

(54) Bezeichnung: **SYSTEME UND VERFAHREN ZUR VERKEHRSKLASSENBEZOGENEN LEITWEGLENKUNG**

(57) Zusammenfassung: Es werden Systeme und Verfahren beschrieben, die die Weiterleitung von Daten innerhalb eines Netzwerks nach Verkehrsklassen ermöglichen. Ein Netzwerk-Switch ist in der Lage, Verkehrsdaten auf der Grundlage von High Performance Computing (HPC)-bezogenen Merkmalen zu klassifizieren. Verkehrsklassen werden auf der Grundlage von HPC-Aspekten definiert, wie z. B. Routing, Ordnung, Umleitung, Ruhezustand, HPC-Protokollkonfiguration und Telemetrie. Ein Switch kann Pakete an einem Ingress-Port einer Switch-Fabric empfangen und Verkehrsklassifizierungen für die Pakete festlegen. Die Verkehrsklassifizierung wird aus einer Gruppe von definierten Verkehrsklassen ausgewählt. Dann kann der Switch für das mindestens eine Paket ein Fabric-spezifisches Flag erzeugen, das die ermittelte Verkehrsklassifizierung angibt, wobei das Fabric-spezifische Flag für das Routing von Paketen auf der Grundlage der ihnen zugewiesenen Verkehrsklassifizierung verwendet wird. Beispiele für Verkehrsklassen sind: Klasse für niedrige Latenzzeiten, Klasse für dedizierten Zugang, Klasse für Massendaten, Klasse für beste Bemühungen und Scavenger-Klasse.



Beschreibung

Erklärung der Rechte der Regierung

[0001] Die hier beschriebene(n) Erfindung(en) wurde(n) mit Unterstützung der US-Regierung im Rahmen eines oder mehrerer der unten aufgeführten Verträge gemacht. Die U.S.-Regierung hat bestimmte Rechte an diesen Erfindungen.

Titel des Vertrags	Kunde/Agentur	Vertragsreferenz
FastForward-2	Lawrence Livermore National Security, LLC/Abteilung für Energie	Untervertrag B609229 unter dem Hauptvertrag DE-AC52-07NA27344
BeePresent	Beschaffungsamt Maryland	H98230-15-D-0020; Lieferauftrag 003
SeaBiscuit	Beschaffungsamt Maryland	II98230-14-C-0758
PathForward	Lawrence Livermore National Security, LLC/Abteilung für Energie	Untervertrag B620872 im Rahmen des Hauptvertrags DE-AC52-07NA27344
DesignForward	Die Regenten der Universität von Kalifornien / Energieministerium	Untervertrag 7078453 unter Hauptvertrag DE-AC02-05CII11231
EntwurfVorwärts-2	Die Regenten der Universität von Kalifornien / Energieministerium	Untervertrag 7216357 unter Hauptvertrag DE-AC02-05CII11231

Verwandte Anwendungen

[0002] Gemäß 35 U.S.C. 119 beansprucht dies den Nutzen und das Prioritätsrecht der vorläufigen US-Patentanmeldung Nr. 62/852273, eingereicht am 23. Mai 2019, mit dem Titel „Netzwerk-Switch“, der vorläufigen US-Patentanmeldung Nr. 62/852203, eingereicht am 23. Mai 2019, mit dem Titel „Netzwerk-Schnittstellen--Controller“ und der vorläufigen US-Patentanmeldung Nr. 62/852289, eingereicht am 23. Mai 2019, mit dem Titel „Netzwerk-Computersystem“, deren Offenbarungen durch Bezugnahme hierin enthalten sind.

Beschreibung des Standes der Technik

[0003] Da netzwerkfähige Geräte und Anwendungen immer allgegenwärtiger werden, erfordern verschiedene Arten von Datenverkehr sowie die ständig steigende Netzwerklast immer mehr Leistung von der zugrunde liegenden Netzwerkarchitektur. So können beispielsweise Anwendungen wie High-Performance Computing (HPC), Medien-Streaming und Internet of Things (IOT) verschiedene Arten von Datenverkehr mit unterschiedlichen Merkmalen erzeugen. Infolgedessen stehen Netzwerkarchitekten zusätzlich zu den herkömmlichen Netzwerkleistungskennzahlen wie Bandbreite und Verzögerung weiterhin vor Herausforderungen wie Skalierbarkeit, Vielseitigkeit und Effizienz.

Figurenliste

[0004] Die vorliegende Offenbarung wird in Übereinstimmung mit einer oder mehreren verschiedenen Ausführungsformen unter Bezugnahme auf die folgenden Figuren im Detail beschrieben. Die Figuren dienen lediglich der Veranschaulichung und stellen lediglich typische oder beispielhafte Ausführungsformen dar.

Abb. 1 zeigt ein Beispielnetz, in dem verschiedene Ausführungsformen implementiert werden können.

Abb. 2 zeigt ein Beispiel für einen Switch, der das Routing pro Verkehrsklasse ermöglicht.

Abb. 3A zeigt Kreuzschienen, die in einem Beispiel für einen Kreuzschienenschalter gemäß verschiedenen Ausführungsformen implementiert sind.

Abb. 3B zeigt eine Beispiel-Kachelmatrix, die den Anschlüssen des Beispiel-Kantenvermittlungssystems von **Abb. 2** in Übereinstimmung mit verschiedenen Ausführungsformen entspricht.

Abb. 3C zeigt ein Beispiel für eine Kachel, die die Kachelmatrix von **Abb. 3B** in Übereinstimmung mit verschiedenen Ausführungsformen bildet.

Abb. 4 zeigt ein konzeptionelles Diagramm eines Switches in einem Edge-Switching-System von **Abb. 2**, das eine Verkehrsklassifizierung in Übereinstimmung mit verschiedenen Ausführungsformen durchführt.

Abb. 5A und **Abb. 5B** sind Blockdiagramme einer beispielhaften FRF-Komponente, die an jedem Anschluss des beispielhaften Kantenschaltsystems von **Abb. 2** implementiert ist.

Abb. 6 zeigt ein Flussdiagramm eines beispielhaften Prozesses der Leitweglenkung pro Verkehrsklasse in Übereinstimmung mit verschiedenen Ausführungsformen.

Abb. 7 ist ein Beispiel für eine Computerkomponente, die zur Implementierung verschiedener Merkmale der in der vorliegenden Offenbarung beschriebenen Ausführungsformen verwendet werden kann.

[0005] Die Abbildungen sind nicht erschöpfend und beschränken die vorliegende Offenbarung nicht auf die genaue Form, die offenbart wird.

Detaillierte Beschreibung

[0006] Große Netze bestehen aus vielen einzelnen Switches, die über Datenverbindungen miteinander verbunden sind. Herkömmliche Netze teilen Daten in überschaubare Pakete oder Rahmen auf. Auf diese Weise können sich viele separate und unterschiedliche Kommunikationen die Bandbreite einer einzigen Verbindung teilen. Insbesondere verhindert eine einzige große Datenübertragung für eine Kommunikation nicht, dass viele andere kleine Kommunikationen abgeschlossen werden können. Die große Kommunikation wird in viele einzelne Pakete aufgeteilt und ihre Pakete werden mit den Paketen anderer kleiner und großer Kommunikationen zeitlich gemultiplext. Mit diesem Ansatz kann eine einzige gemeinsam genutzte Netzressource viele gleichzeitige Kommunikationen durchführen und die maximale Latenz kleiner Kommunikationen bei Vorhandensein großer Kommunikationen erheblich reduzieren.

[0007] Die gemeinsame Nutzung von Ressourcen durch viele völlig unterschiedliche Kommunikationen funktioniert jedoch nur dann gut, wenn keine der Kommunikationen die gemeinsamen Ressourcen der anderen Kommunikationen erschöpfen kann. Außerdem ist es wichtig, dass der Zugang zu den gemeinsam genutzten Ressourcen fair bleibt und der Bedeutung der einzelnen Kommunikationen entspricht. Die Verkehrsklassifizierung wird traditionell zur Implementierung von datenübertragungsspezifischen Netzattributen verwendet. Diese traditionellen Klassifizierungen berücksichtigen Fragen des Durchsatzes, der Bandbreitenzuweisung und der Latenzzeit. High-Performance-Computing (HPC) hat jedoch mehrere einzigartige Netzwerkverhalten, Arbeitsabläufe, Funktionen und Eigenschaften, die von diesen traditionellen Klassifizierungen nicht abgedeckt werden.

[0008] Um den spezifischen Merkmalen von HPC-Anwendungen gerecht zu werden, beschreiben die vorgestellten Ausführungsformen eine Vermittlungseinrichtung, die die Bestimmung von Datenfluss- und Netzwerkeigenschaften ermöglicht, wie z. B. Routing- und Re-Routing-Direktiven, Trennung des Datenflusses, geordnete oder ungeordnete Datenlieferung, verlustbehaftete oder verlustfreie Übertragung, Erfassung von Telemetriedaten und Regeln zur Verkehrsgestaltung. Es werden zusätzliche Verkehrsklassen verwendet, die über diejenigen hinausgehen, die typischerweise bei herkömmlichen klassenweisen Verkehrslenkungskonzepten verwendet werden. Diese Verkehrsklassen beruhen auf den oben genannten HPC-bezogenen Festlegungen und werden gleichzeitig über dieselbe physische Netzinfrastruktur betrieben.

[0009] Wie im Einzelnen beschrieben wird, kann die offengelegte Routing-Funktionalität pro Verkehrsklasse im gesamten Netz auf Paketbasis aktiviert, konfiguriert und abgestimmt werden, indem Pakete innerhalb eines Datenflusses markiert werden. Darüber hinaus ermöglicht der paketweise Ansatz eine In-Band-Kontrolle von Datenflüssen, die durch die Datenpakete selbst eingerichtet wird. Diese Art der In-Band-Kontrolle von Daten innerhalb des Netzes auf der Grundlage dieser Art von HPC-bezogenen Merkmalen wurde bisher noch nicht berücksichtigt. Mit anderen Worten, die hier vorgestellten Systeme und Techniken ermöglichen Überlegungen, die über Latenz und Bandbreite hinausgehen, einschließlich des Routing-Verhaltens, der Trennung von Anwendungsdatenströmen und der Staukontrolle auf der Basis der einzelnen Verkehrsklassen. Letztlich führt das Routing pro Verkehrsklasse gemäß den Ausführungsformen zu einer feinkörnigen Steuerung des Datentransits im Netzwerk.

[0010] Die vorliegende Offenlegung beschreibt Systeme und Verfahren, die für Exascale-Computing geeignet sind, z. B. für die Durchführung datenintensiver Aufgaben wie Simulationen, Datenanalyse und künstliche Intelligenz bei Exascale-Geschwindigkeiten. Insbesondere wird ein HPC-Netzwerk oder eine HPC-Verbindungsstruktur bereitgestellt, die Ethernet-kompatibel sein kann, mit Datenspeichern von Drittanbietern verbunden werden kann und mit einer Switch-Komponente aufgebaut werden kann, die eine extrem hohe Bandbreite aufweist, z. B. in der Größenordnung von 12,8 TB/s/dir pro Switch mit z. B. 64 200-Gbit/s-Ports, die eine große Netzwerkbildung mit sehr geringem Durchmesser (z. B. nur drei Netzwerksprünge) unterstützen.

Darüber hinaus können niedrige Latenzzeiten durch neuartige Staukontrollmechanismen, adaptives Routing und die Verwendung von Verkehrsklassen erreicht werden, die eine flexible Gestaltung von Bandbreite, Priorität und Routing-Politik ermöglichen.

[0011] Was die adaptive Leitweglenkung betrifft, so können die hier beschriebenen Techniken und Systeme eine dynamische Leitweglenkung von Datenströmen erreichen, indem sie die Identifizierung und Verwaltung von Datenstromkanälen nutzen. Beim Routing von Datenpaketen zwischen einem Quellknoten und einem Ziel sind typische Routing-Techniken entweder statisch oder adaptiv (z. B. dynamisch). Bei einem Beispiel für adaptives Routing werden lokale Routing-Entscheidungen dynamisch auf der Grundlage von Lastinformationen und anderen Faktoren getroffen. In aktuellen Systemen kann die adaptive Leitweglenkung dazu führen, dass sich Überlastungen ausbreiten. Dabei können bestimmte Datenströme als Stauverursacher identifiziert werden, während andere Datenströme einfach als Stauopfer identifiziert werden können. Bei den hier beschriebenen adaptiven Leitweglenkungstechniken, die sich mit persistenten Datenströmen befassen, können die Opferströme weiterhin herkömmliche Leitweglenkungsentscheidungen treffen, während die Leitweglenkung von Datenströmen, die Überlastungen verursachen, eingeschränkt wird. Wie bereits angedeutet, wird diese Fähigkeit durch die Identifizierung und Verwaltung von Flusskanälen ermöglicht.

[0012] **Abb. 1** zeigt ein Beispiel für ein Netzwerk 100 mit einer Vielzahl von Switches, das auch als „Switch Fabric“ bezeichnet werden kann. Wie in **Abb. 1** dargestellt, kann das Netzwerk 100 Switches 102, 104, 106, 108 und 110 umfassen. Jeder Switch kann eine eindeutige Adresse oder Kennung (ID) innerhalb der Switch--Fabric 100 haben. Verschiedene Arten von Geräten und Netzwerken können mit einer Switch-Fabric verbunden werden. Beispielsweise kann ein Speicherarray 112 über den Switch 110 mit der Switch Fabric 100 verbunden werden; ein InfiniBand (IB)-basiertes HPC-Netzwerk 114 kann über den Switch 108 mit der Switch Fabric 100 verbunden werden; eine Reihe von Endhosts, wie der Host 116, kann über den Switch 104 mit der Switch Fabric 100 verbunden werden; und ein IP/Ethernet-Netzwerk 118 kann über den Switch 102 mit der Switch Fabric 100 verbunden werden. Beispielsweise kann ein Switch wie Switch 102 802.3-Rahmen (einschließlich der eingekapselten IP-Nutzdaten) über Ethernet-Geräte wie Netzwerkschnittstellenkarten (NICs), Switches, Router oder Gateways empfangen. IPv4- oder IPv6-Pakete, speziell für das Netzwerk 100 formatierte Frames usw. können ebenfalls empfangen und durch die Switch-Fabric 100 zu einem anderen Switch, z. B. Switch 110, transportiert werden. Auf diese Weise ist das Netz 100 in der Lage, mehrere Arten von Datenverkehr gleichzeitig zu verarbeiten. Im Allgemeinen kann ein Switch über Edge-Ports und Fabric-Ports verfügen. Ein Edge-Port kann mit einem Gerät verbunden werden, das sich außerhalb der Fabric befindet. Ein Fabric-Port kann über eine Fabric-Verbindung mit einem anderen Switch innerhalb der Fabric verbunden werden.

[0013] Normalerweise kann der Verkehr über einen Eingangsport eines Edge-Switches in die Switch-Fabric 100 eingespeist werden und die Switch-Fabric 100 über einen Ausgangsport eines anderen (oder desselben) Edge-Switches verlassen. Ein Ingress-Edge-Switch kann injizierte Datenpakete in Flows gruppieren, die durch Flow-IDs identifiziert werden können. Das Konzept eines Datenflusses ist nicht auf ein bestimmtes Protokoll oder eine bestimmte Schicht (wie Schicht-2 oder Schicht-3 im OSI-Referenzmodell) beschränkt. Ein Datenfluss kann z. B. dem Datenverkehr mit einer bestimmten Quell-Ethernet-Adresse, dem Datenverkehr zwischen einer Quell-IP-Adresse und einer Ziel-IP-Adresse, dem Datenverkehr, der einem TCP- oder UDP--Port/IP-5-Tupel entspricht (Quell- und Ziel-IP-Adresse, Quell- und Ziel-TCP- oder -UDP-Portnummer und IP-Protokollnummer), oder dem Datenverkehr, der von einem auf einem Endhost laufenden Prozess oder Thread erzeugt wird, zugeordnet werden. Mit anderen Worten: Ein Fluss kann so konfiguriert werden, dass er Daten zwischen beliebigen physischen oder logischen Einheiten zuordnet. Die Konfiguration dieser Zuordnung kann per Fernzugriff oder lokal am Ingress Edge Switch vorgenommen werden.

[0014] Beim Empfang von injizierten Datenpaketen kann der Ingress-Edge-Switch dem Datenstrom eine Flow-ID zuweisen. Diese Flow-ID kann in einem speziellen Header enthalten sein, den der Ingress Edge Switch zur Verkapselung der injizierten Pakete verwenden kann. Darüber hinaus kann der Ingress-Edge-Switch auch die ursprünglichen Header-Felder eines injizierten Pakets untersuchen, um die entsprechende Adresse des Egress-Edge-Switch zu ermitteln, und diese Adresse als Zieladresse in den Einkapselungs-Header aufnehmen. Beachten Sie, dass die Flow-ID ein lokal signifikanter Wert sein kann, der für eine Verbindung spezifisch ist, und dass dieser Wert nur für einen bestimmten Eingangsport auf einem Switch eindeutig sein kann. Wenn das Paket an den Next-Hop-Switch weitergeleitet wird, tritt das Paket in eine andere Verbindung ein, und die Flow-ID kann entsprechend aktualisiert werden. Da die Pakete eines Flusses mehrere Verbindungen und Switches durchlaufen, können die diesem Fluss entsprechenden Flow-IDs eine eindeutige Kette bilden. Das heißt, dass an jedem Switch, bevor ein Paket den Switch verlässt, die Flow-ID des Pakets auf eine Flow-ID aktualisiert werden kann, die von der ausgehenden Verbindung verwendet wird. Diese Eins--

zu-Eins-Zuordnung zwischen den Fluss-IDs kann am Ingress-Edge-Switch beginnen und am Egress-Edge-Switch enden. Da die Fluss-IDs nur innerhalb einer eingehenden Verbindung eindeutig sein müssen, kann ein Switch eine große Anzahl von Flüssen aufnehmen. Wenn eine Fluss-ID beispielsweise 11 Bits lang ist, kann ein Eingangsanschluss bis zu 2048 Flüsse unterstützen. Darüber hinaus kann das Match-Muster (ein oder mehrere Header-Felder eines Pakets), das zur Zuordnung zu einem Datenfluss verwendet wird, eine größere Anzahl von Bits enthalten. Ein 32-Bit langes Abgleichmuster, das mehrere Felder in einem Paketkopf enthalten kann, kann beispielsweise 2^32 verschiedene Kopffeldmuster abbilden. Wenn eine Fabric über N Ingress-Edge-Ports verfügt, kann eine Gesamtzahl von N*2^32 identifizierbaren Flows unterstützt werden.

[0015] Ein Switch kann jedem Datenfluss eine eigene, dedizierte Eingangswarteschlange zuweisen. Diese Konfiguration ermöglicht es dem Switch, den Grad der Überlastung einzelner Datenströme zu überwachen und zu verwalten und eine Blockierung der Warteschlange zu verhindern, die auftreten könnte, wenn ein gemeinsamer Puffer für mehrere Datenströme verwendet wird. Wenn ein Paket an den Ziel-Egress-Switch geliefert wird, kann der Egress-Switch eine Bestätigung (ACK) in Upstream-Richtung über denselben Datenpfad an den Ingress-Edge-Switch zurücksenden. Da dieses ACK-Paket denselben Datenpfad durchläuft, können die Switches entlang des Pfades die Zustandsinformationen erhalten, die mit der Zustellung des entsprechenden Datenflusses verbunden sind, indem sie die Menge der ausstehenden, unbestätigten Daten überwachen. Diese Zustandsinformationen können dann verwendet werden, um ein flussspezifisches Verkehrsmanagement durchzuführen, um den Zustand des gesamten Netzes und eine faire Behandlung der Flüsse zu gewährleisten. Wie weiter unten näher erläutert, kann die Switch Fabric durch diese Warteschlangenbildung pro Datenfluss in Kombination mit flussspezifischen Zustellungsbestätigungen eine effektive, schnelle und genaue Staukontrolle implementieren. Im Gegenzug kann die Switch Fabric den Datenverkehr mit einer deutlich verbesserten Netzwerkauslastung bereitstellen, ohne dass es zu Überlastungen kommt.

[0016] Flows können dynamisch oder „on the fly“ je nach Bedarf eingerichtet und freigegeben werden. Insbesondere kann ein Fluss von einem Edge-Switch eingerichtet werden (z. B. wird die Zuordnung von Fluss-ID zu Paketkopf hergestellt), wenn ein Datenpaket am Switch ankommt und diesem Paket zuvor keine Fluss-ID zugewiesen wurde. Während dieses Paket das Netz durchläuft, können Flow-IDs an jedem Switch, den das Paket durchläuft, zugewiesen werden, und es kann eine Kette von Flow-IDs vom Eingang bis zum Ausgang gebildet werden. Nachfolgende Pakete, die zum selben Fluss gehören, können auf dem Datenpfad dieselben Fluss-IDs verwenden. Wenn Pakete an den Ziel-Egress-Switch zugestellt und ACK-Pakete von den Switches entlang des Datenpfads empfangen werden, kann jeder Switch seine Zustandsinformationen in Bezug auf die Menge der ausstehenden, nicht quittierten Daten für diesen Fluss aktualisieren. Wenn die Eingangswarteschlange eines Switches für diesen Datenfluss leer ist und es keine weiteren unbestätigten Daten gibt, kann der Switch die Fluss-ID freigeben (d. h. diesen Flusskanal freigeben) und die Fluss-ID für andere Flüsse wiederverwenden. Durch diesen datengesteuerten dynamischen Mechanismus für die Einrichtung und den Abbau von Datenflüssen wird eine zentrale Verwaltung der Datenflüsse überflüssig, und das Netz kann schnell auf Änderungen der Verkehrsmuster reagieren.

[0017] Beachten Sie, dass sich die hier beschriebene Netzwerkarchitektur von Software-definierten Netzwerken (SDN) unterscheidet, die in der Regel das OpenFlow-Protokoll verwenden. In SDN werden Switches von einem zentralen Netzwerk-Controller konfiguriert, und Pakete werden auf der Grundlage eines oder mehrerer Felder in den Headern der Schicht 2 (Datenverbindungsschicht, z. B. Ethernet), Schicht 3 (Netzwerkschicht, z. B. IP) oder Schicht 4 (Transportschicht, z. B. TCP oder UDP) weitergeleitet. Im SDN wird eine solche Header-Feld-Suche an jedem Switch im Netzwerk durchgeführt, und es gibt keine schnelle, auf der Flow-ID basierende Weiterleitung, wie sie in den hier beschriebenen Netzwerken erfolgt. Da die OpenFlow-Header-Feld-Suche mit ternärem inhaltsadressierbarem Speicher (TCAM) durchgeführt wird, können die Kosten für solche Suchvorgänge hoch sein. Da die Konfiguration der Header-Feld-Zuordnung von der zentralen Steuereinheit vorgenommen wird, ist der Auf- und Abbau jeder Zuordnungsbeziehung relativ langsam und kann eine beträchtliche Menge an Steuerverkehr erfordern. Infolgedessen kann die Reaktion eines SDN-Netzwerks auf verschiedene Netzwerksituationen, wie z. B. eine Überlastung, langsam sein. Im Gegensatz dazu können in dem hier beschriebenen Netzwerk die Flows dynamisch auf der Grundlage der Verkehrsnachfrage auf- und abgebaut werden, und die Pakete können mit einer Flow-ID fester Länge weitergeleitet werden. Mit anderen Worten: Flusskanäle können datengesteuert und dezentral verwaltet (d. h. eingerichtet, überwacht und abgebaut) werden, ohne dass ein zentraler Controller eingreifen muss. Darüber hinaus kann die auf der Fluss-ID basierende Weiterleitung die Menge des verwendeten TCAM-Speicherplatzes reduzieren, so dass eine viel größere Anzahl von Flüssen untergebracht werden kann.

[0018] Nehmen wir an, dass die Speichermatrix 112 Daten über TCP/IP an den Host 116 senden soll (siehe das Beispiel in **Abb. 1**). Während des Betriebs kann die Speichermatrix 112 das erste Paket mit der IP-Ad-

resse des Hosts 116 als Zieladresse und einem vorbestimmten TCP-Port, der im TCP-Header angegeben ist, senden. Wenn dieses Paket die Vermittlungsstelle 110 erreicht, kann der Paketprozessor am Eingangsport der Vermittlungsstelle 110 ein TCP/IP-5-Tupel dieses Pakets identifizieren. Der Paketprozessor der Vermittlungsstelle 110 kann auch feststellen, dass dieses 5-Tupel derzeit keiner Fluss-ID zugeordnet ist, und kann diesem 5-Tupel eine neue Fluss-ID zuweisen. Darüber hinaus kann die Vermittlungsstelle 110 den Ausgangs-Switch, d. h. den Switch 104, für dieses Paket auf der Grundlage der IP-Adresse des Ziels (d. h. des Hosts 116) bestimmen (vorausgesetzt, die Vermittlungsstelle 110 weiß, dass der Host 116 mit dem Switch 104 verbunden ist). Anschließend kann der Switch 110 das empfangene Paket mit einem Fabric-Header einkapseln, der die neu zugewiesene Flow-ID und die Fabric-Adresse des Switches 104 angibt. Switch 110 kann dann die Weiterleitung des eingekapselten Pakets an Switch 104 auf der Grundlage einer Fabric-Weiterleitungstabelle planen, die von allen Switches in Fabric 100 unter Verwendung eines Routing-Algorithmus wie Link State oder Distance Vector berechnet werden kann.

[0019] Beachten Sie, dass die oben beschriebenen Vorgänge im Wesentlichen mit Leitungsgeschwindigkeit und mit geringer Pufferung und Verzögerung durchgeführt werden können, wenn das erste Paket empfangen wird. Nachdem das erste Paket verarbeitet und für die Übertragung eingeplant wurde, können nachfolgende Pakete desselben Datenflusses von der Vermittlungsstelle 110 noch schneller verarbeitet werden, da dieselbe Datenfluss-ID verwendet wird. Darüber hinaus können die Flusskanäle so gestaltet werden, dass die Zuweisung, der Abgleich und die Freigabe von Flusskanälen im Wesentlichen die gleichen Kosten verursachen. So können beispielsweise eine bedingte Zuweisung eines Flusskanals auf der Grundlage einer Nachschlageübereinstimmung und eine separate, unabhängige Freigabe eines anderen Flusskanals fast in jedem Taktzyklus gleichzeitig durchgeführt werden. Das bedeutet, dass die Erzeugung und Kontrolle der Flusskanäle fast keinen zusätzlichen Overhead zur regulären Weiterleitung von Paketen bedeutet. Der Staukontrollmechanismus hingegen kann die Leistung einiger Anwendungen um mehr als drei Größenordnungen verbessern.

[0020] An jeder Vermittlungsstelle entlang des Datenpfads (einschließlich der Vermittlungsstellen 110, 106 und 104) kann ein dedizierter Eingangspuffer für diesen Datenfluss bereitgestellt werden, und die Menge der übertragenen, aber nicht quittierten Daten kann verfolgt werden. Wenn das erste Paket den Switch 104 erreicht, kann der Switch 104 feststellen, dass die Fabric-Zieladresse im Fabric-Header des Pakets mit seiner eigenen Adresse übereinstimmt. Daraufhin kann der Switch 104 das Paket aus dem Fabric-Header entkapseln und das entkapselte Paket an den Host 116 weiterleiten. Außerdem kann der Switch 104 ein ACK-Paket erzeugen und dieses ACK-Paket an den Switch 110 zurücksenden. Da dieses ACK-Paket denselben Datenpfad durchläuft, können die Switches 106 und 110 jeweils ihre eigenen Statusinformationen für die unbestätigten Daten für diesen Fluss aktualisieren.

[0021] Im Allgemeinen kann eine Überlastung des Netzes dazu führen, dass sich die Netzpuffer füllen. Wenn ein Netzpuffer voll ist, sollte der Verkehr, der den Puffer passieren will, idealerweise verlangsamt oder gestoppt werden. Andernfalls könnte der Puffer überlaufen, und die Pakete könnten verworfen werden. In herkömmlichen Netzen erfolgt die Staukontrolle in der Regel von Ende zu Ende am Rand. Es wird davon ausgegangen, dass der Kern des Netzes nur als „dumme Röhre“ fungiert, deren Hauptzweck die Weiterleitung des Datenverkehrs ist. Ein solches Netzdesign leidet oft unter einer langsamen Reaktion auf Überlastungen, da Überlastungsinformationen oft nicht schnell an die Edge-Geräte gesendet werden können und die daraus resultierenden Maßnahmen der Edge-Geräte die Überlastung nicht immer wirksam beseitigen können. Diese langsame Reaktion schränkt wiederum die Auslastung des Netzes ein, denn um das Netz staufrei zu halten, muss der Netzbetreiber häufig die Gesamtmenge des in das Netz eingespeisten Verkehrs begrenzen. Außerdem ist eine Ende-zu-Ende-Überlastungskontrolle in der Regel nur dann wirksam, wenn das Netz nicht bereits überlastet ist. Sobald das Netz stark überlastet ist, würde eine Ende-zu-EndeÜberlastungssteuerung nicht mehr funktionieren, da die Überlastungsmeldungen selbst überlastet sein können (es sei denn, für das Senden von Überlastungssteuerungsmeldungen wird ein separates Netz der Steuerungsebene verwendet, das sich vom Netz der Datenebene unterscheidet).

[0022] Im Gegensatz dazu können die Flusskanäle verhindern, dass eine solche Überlastung innerhalb der Switch-Fabric entsteht. Der Flow-Channel-Mechanismus kann erkennen, wenn ein Fluss einen gewissen Grad an Überlastung erfährt, und als Reaktion darauf neue Pakete desselben Flusses verlangsamen oder daran hindern, in die Fabric zu gelangen. Im Gegenzug können diese neuen Pakete in einer Flow-Channel--Warteschlange am Edge-Port zwischengespeichert werden und werden erst dann in die Fabric gelassen, wenn Pakete für denselben Flow die Fabric am Edge-Zielport verlassen. Durch diesen Prozess kann der Gesamtpufferbedarf dieses Flusses innerhalb der Fabric auf eine Menge begrenzt werden, die nicht dazu führt, dass die Fabric-Puffer zu voll werden.

[0023] Mit Flow Channels verfügen die Switches über einigermaßen genaue Statusinformationen über die Menge der ausstehenden Daten im Transit innerhalb der Fabric. Diese Zustandsinformationen können für alle Ströme an einem Ingress-Edge-Port aggregiert werden. Dies bedeutet, dass die Gesamtmenge der von einem Ingress-Edge-Port eingespeisten Daten bekannt ist. Folglich kann der Flow-Channel-Mechanismus eine Grenze für die Gesamtdatenmenge in der Fabric festlegen. Wenn alle Edge-Ports diese Begrenzung anwenden, kann die Gesamtmenge der Paketdaten in der gesamten Fabric gut kontrolliert werden, was wiederum verhindern kann, dass die gesamte Fabric gesättigt wird. Die Flusskanäle können auch den Fortschritt eines einzelnen überlasteten Flusses innerhalb der Fabric verlangsamen, ohne andere Flüsse zu verlangsamen. Mit dieser Funktion können Pakete von einem Stau-Hotspot ferngehalten werden, während gleichzeitig verhindert wird, dass die Puffer voll werden, und freier Pufferplatz für nicht zusammenhängenden Verkehr gewährleistet wird.

[0024] Der Crossbar-Switch 210 kann einen oder mehrere Crossbar-Switch-Chips enthalten, die so konfiguriert werden können, dass sie Datenpakete und Steuerpakete (wie ACK-Pakete) zwischen den Kommunikationsports weiterleiten. Der EFCT-Logikblock 212 kann von einem Edge Link empfangene Pakete verarbeiten und die empfangenen Pakete auf der Grundlage eines oder mehrerer Header-Felder in den Paketen den jeweiligen Flüssen zuordnen. Darüber hinaus kann der EFCT-Logikblock 212 FGFC-Ethernet-Frames zusammenstellen, die an einen Endhost übermittelt werden können, um die von einzelnen Prozessen oder Threads eingespeiste Datenmenge zu steuern. Der IFCT-Logikblock 214 kann den IFCT enthalten und verschiedene Flusssteuerungsmethoden als Reaktion auf Steuerpakete durchführen, wie z. B. ACKs zur Endpunkt-Stau-Benachrichtigung und auf Fabric-Link-Kredit basierende Flusssteuerungs-ACKs. Der OFCT-Logikblock 216 kann eine Speichereinheit enthalten, die die OFCT speichert und mit dem IFCT-Logikblock eines anderen Switches kommuniziert, um die Fluss-ID eines Pakets zu aktualisieren, wenn das Paket an einen Next-Hop-Switch weitergeleitet wird.

[0025] In einer Ausführungsform ist der Switch 202 ein anwendungsspezifischer integrierter Schaltkreis (ASIC), der 64 Netzwerkanschlüsse bereitstellen kann, die entweder mit 100 Gbit/s oder 200 Gbit/s für einen Gesamtdurchsatz von 12,8 Tbps arbeiten können. Jeder Netzwerk-Edge-Port kann IEEE 802.3-Ethernet und auf Optimized-IP basierende Protokolle sowie Portals unterstützen, ein erweitertes Frame-Format, das höhere Raten kleiner Nachrichten unterstützt. Ethernet-Frames können auf der Grundlage ihrer L2-Adresse überbrückt oder auf der Grundlage ihrer L3-Adresse (1Pv4//1Pv6) weitergeleitet werden. Optimized-IP-Frames können nur einen L3 (IPv4/IPv6)-Header haben und werden geroutet. Spezielle NIC-Unterstützung kann für das erweiterte Frame-Format von Portals verwendet werden und kann direkt auf das Fabric-Format des Netzwerks 100 abgebildet werden, z. B. ein Fabric-Format, das bestimmte Steuer- und Statusfelder zur Unterstützung einer Multi-Chip-Fabric bereitstellt, wenn Switches/Switch-Chips wie die Switches 102, 104, 106, 108 und 110 angeschlossen sind und miteinander kommunizieren. Wie bereits angedeutet, kann ein auf Flusskanälen basierender Staukontrollmechanismus von solchen Switches verwendet werden und auch hohe Übertragungsraten für kleine Pakete (z. B. mehr als 1,2 Milliarden Pakete pro Sekunde pro Port) erreichen, um den Anforderungen von HPC-Anwendungen gerecht zu werden.

[0026] Der Switch 202 kann systemweite Quality of Service (QoS)-Klassen bereitstellen und steuern, wie die Netzwerkbandbreite verschiedenen Verkehrsklassen und verschiedenen Anwendungsklassen zugewiesen wird, wobei eine einzelne privilegierte Anwendung auf mehr als eine Verkehrsklasse zugreifen kann. Bei Konflikten um die Netzwerkbandbreite wählen die Arbitratoren die weiterzuleitenden Pakete auf der Grundlage ihrer Verkehrsklasse und der für diese Klasse verfügbaren Credits aus. Netzwerk 100 kann minimale und maximale Bandbreiten für jede Verkehrsklasse unterstützen. Wenn eine Klasse ihre Mindestbandbreite nicht ausschöpft, können andere Klassen die ungenutzte Bandbreite nutzen, aber keine Klasse kann mehr als die ihr zugewiesene Höchstbandbreite erhalten. Die Möglichkeit, die Bandbreite zu verwalten, bietet die Möglichkeit, Netzwerkressourcen sowie CPUs und Speicherbandbreite einer bestimmten Anwendung zuzuweisen.

[0027] Zusätzlich zur Unterstützung von QoS-Klassen führt der Switch 202 eine auf dem Flusskanal basierende Staukontrolle durch und kann die Anzahl der Netzwerk-Sprünge, z. B. in einem Netzwerk mit einer Libellen-Topologie, von fünf auf drei Netzwerk-Sprünge reduzieren. Das Design des Switches 202, das weiter unten ausführlicher beschrieben wird, kann die Netzwerkkosten und den Stromverbrauch reduzieren und die Verwendung innovativer adaptiver Routing-Algorithmen erleichtern, die die Anwendungsleistung verbessern. Eine durch eine Vielzahl von Switches, wie z. B. eine Vielzahl von Switches 202, geschaffene Struktur kann auch beim Aufbau von Fat-Tree-Netzwerken verwendet werden, z. B. beim Aufbau eines Speichersubsystems zur Integration mit Netzwerken und Software von Drittanbietern. Darüber hinaus ermöglicht die Verwendung von Switch 202 eine feinkörnige adaptive Leitweglenkung unter Beibehaltung einer geordneten Paketzustellung. In einigen Ausführungsformen kann der Switch 202 so konfiguriert werden, dass er den Header

eines Pakets von einem Eingangsport an einen Ausgangsport sendet, bevor die vollständige Datennutzlast eintrifft, so dass die Lastmetrik des Ausgangsports künftige Lasten widerspiegeln kann, wodurch die von Switch 202 getroffenen adaptiven Routing-Entscheidungen verbessert werden.

[0028] Der Kreuzschienenschalter 210 kann aus separaten, verteilten Kreuzschienen bestehen, die Daten/-Datenelemente zwischen Eingangs- und Ausgangsports weiterleiten. In einigen Ausführungsformen und wie in **Abb. 3A** dargestellt, gibt es fünf verteilte Kreuzschienen, einschließlich einer Anforderungs-Kreuzschiene 210a, einer Erteilungs-Kreuzschiene 210b, einer Kredit-Kreuzschiene 210c, einer Ack-Kreuzschiene 210de und einer Daten-Kreuzschiene 210d zwischen Eingangsanschluss 220b und Ausgangsanschluss 220c.

[0029] Die Anforderungs-Kreuzschiene 210a wird verwendet, um Anforderungen von einem Eingang an eine gezielte Ausgangswarteschlange zu senden. Grant crossbar 210b wird verwendet, um einen Grant zurück an den Eingang zu senden, um eine Anforderung zu erfüllen. Insbesondere gibt die Grant-Kreuzschiene 210b einen Zeiger zurück, der anzeigt, wo sich ein Paket in einem Eingangspuffer befindet. Es sollte beachtet werden, dass ein Grant zurückgegeben wird, wenn in der Ausgabe Platz für das entsprechende Paket vorhanden ist. Die Grant-Kreuzschiene 201b kann optional auch eine Gutschrift für angeforderten Platz in der Ausgabe zurückgeben. Es ist zu beachten, dass die Gewährung zurückgegeben wird, wenn es einen Landeplatz für ein Paket am Ausgang gibt, z. B. einen Ausgangsanschluss 220c, so dass Pakete nicht blockiert werden können (obwohl sie vorübergehend um Ressourcen konkurrieren können).

[0030] Es sollte klar sein, dass in Übereinstimmung mit verschiedenen Ausführungsformen ein Kreditprotokoll verwendet werden kann, um zu garantieren, dass es einen Landeplatz für eine Anforderung am Ausgang gibt. Dementsprechend kann ein Guthaben-Querbalken 210c verwendet werden, um Guthaben für angeforderten Platz in der Ausgabe zurückzugeben.

[0031] Eine Datenkreuzschiene 210d wird verwendet, um gewährte Pakete von einem Eingangspuffer zu einem gezielten Ausgangspuffer zu übertragen. Eine Ack-Kreuzschiene 210e dient der Weiterleitung von Ack-Paketen von den Ausgangsanschlüssen 220c zu den Eingangsanschlüssen 220b. Die Acks werden entsprechend einem in einer Ausgangsflusskanaltabelle gespeicherten Zustand gesteuert.

[0032] Es sollte klar sein, dass die Daten-Kreuzschiene 210d Multitakt-Pakete mit Kopfzeilen und Daten bewegt, während die anderen vier Kreuzschienen (Anforderungs-Kreuzschiene 210a, Erteilungs-Kreuzschiene 210b, Kredit-Kreuzschiene 210c und Ack-Kreuzschiene 210e) nur Kopfzeilen von Paketen mit einem Takt bewegen. Alle fünf Kreuzschienen verwenden dieselbe Architektur mit Zeilen- und Spaltenbussen innerhalb einer 8 x 4-Matrix von 32 Dual-Port-Kacheln (wie unten beschrieben).

[0033] Zurück zu **Abb. 2**: Der Schalter 202 kann eine Vielzahl von Sende-/Empfangsports haben, z. B. Port 220. Die Vielzahl der Anschlüsse kann in einer Kachelmatrix strukturiert sein. **Abb. 3B** zeigt ein Beispiel für eine solche Kachelmatrix 300. In einer Ausführungsform umfasst die Kachelmatrix 300 32 Kacheln, von denen jede zwei Ports umfasst, die zur Implementierung der Crossbar-Umschaltung zwischen den Ports und zur Bereitstellung der folgenden Elemente verwendet werden: eine Serializer/De-Serializer (SERDES)--Schnittstelle zwischen dem Kern des Schalters 202 und externen seriellen Hochgeschwindigkeitssignalen zum Treiben der Signale vom Schalter 202; eine Media Access Control (MAC)-Unterschichtschnittstelle zur Physical Coding Sublayer (PCS); eine PCS-Schnittstelle zwischen dem SERDES und der Ethernet-MAC--Funktion; eine Link Level Retry (LLR)-Funktion, die auf einer paketweisen Basis arbeitet und geordnete Sätze verwendet, um Initialisierungssequenzen, Acks und Nacks zu liefern; und einen Ingress Transforms--Block zum Konvertieren zwischen verschiedenen Frame-Fabric-Formaten. Jede Kachel enthält einen Kreuzschienenschalter wie den Kreuzschienenschalter 210 für jede der Kreuzschienen (210a-201e). Wie im Folgenden näher erläutert wird, kann das Routing in der Switch-Fabric durch eine Fabric-Routing-Funktion (FRF) gesteuert werden, die im Switch 202 implementiert ist, wobei eine separate Instanz der FRF-Komponente (**Abb. 4A**, **Abb. 4B)** in der Eingangslogik für jeden Port des Switch 202 implementiert sein kann. Wie oben erwähnt, enthält jeder Port eine Instanz der FRF-Komponente, aber zur Vereinfachung der Referenz bzw. zur Veranschaulichung werden nur zwei FRF-Instanzen (400a, 400b) als Beispiel angeführt.

[0034] Jeder Kreuzschienenschalter 210 hat sechzehn Eingangsanschlüsse 220b, einen für jeden Anschluss in seiner Reihe, und acht Ausgangsanschlüsse 220c, einen für jeden Anschluss in seiner Spalte. Zeilenbusse können von jeder Quelle in einer Zeile zu allen acht Kreuzschienen in dieser Zeile geführt werden (one-to-all). Die Arbitrierung kann an der Kreuzschiene von den sechzehn Zeilenbussen in dieser Zeile zu den acht Spaltenbussen in einer bestimmten Spalte durchgeführt werden. An jeder 16 x 8-Kreuzschiene kann für jeden der Zeilenbusse eine Pufferung vorgesehen werden, um Pakete während der Zeiten aufzufan-

gen, in denen ein Spaltenbus umkämpft ist. In einigen Ausführungsformen wird ein Nicht-Jumbo-Paket von einem Zeilenbus ferngehalten, es sei denn, es ist Platz für das gesamte Paket im Eingangspuffer der Zielkreuzschiene. Aufgrund von Flächenbeschränkungen dürfen Jumbo-Pakete auch dann übertragen werden, wenn nicht genügend Platz vorhanden ist (der Crossbar-Eingangspuffer ist nur so groß, dass ein Nicht-Jumbo-Paket darin versenkt werden kann), wobei der Zeilenbus so lange blockiert wird, bis das Paket die Arbitrierung gewinnt und Platz frei wird, wenn es auf einen Spaltenbus übertragen wird.

[0035] Spaltenbusse werden von einer bestimmten Kreuzschiene zu jedem Zielanschluss innerhalb einer Spalte geführt (all-to-all). Jedes Ziel kann eine weitere Arbitrationsebene zwischen den Spaltenbussen der vier Zeilen haben. Bei sechzehn Zeilenbussen, die acht Kreuzschienen ansteuern und jeweils acht Spaltenbusse versorgen, ergibt sich ein 4-facher Geschwindigkeitszuwachs zwischen Zeilen und Spalten. Jede Reihe hat identische Verbindungen, wobei die Verbindungen von einem zu allen Reihenbussen für eine einzelne Reihe in Reihenbussen dargestellt sind. Jede Kachel hat je nach Crossbar eine Verzögerung von einem (request, grant, credit) oder zwei (data, ack) Takten pro Kachel. Dies ergibt eine maximale Verzögerung von sieben oder vierzehn Takten, um zwischen der Spalte ganz links und der Spalte ganz rechts zu gelangen. Kreditrückgaben, die über die Kredit-Kreuzschiene 210c geleitet werden, können eine Verzögerung von einem Takt pro Kachel aufweisen und daher maximal sieben Takte bis zur vollständigen Übertragung benötigen.

[0036] Es sollte beachtet werden, dass jede Spalte identische Verbindungen mit den All-to-All-Spaltenbus--Verbindungen für eine einzelne Spalte haben kann, und dass es eine Verzögerung von zwei Takten pro Kachel geben kann, was zu einer Verzögerung von sechs Takten führt, um von der obersten Zeile zur untersten Zeile zu gelangen. Es sollte auch klar sein, dass sowohl Zeilen- als auch Spaltenbusse das oben erwähnte kreditbasierte Protokoll verwenden, um zu bestimmen, wann sie in der Lage sind zu senden. Im Falle von Zeilenbussen verwaltet der Quellanschluss die Anzahl der Guthaben für die Eingangspuffer der Kreuzschienen innerhalb dieser Zeile. Bei der Daten-Kreuzschiene ist darauf zu achten, wann ein Paket auf einen Zeilenbus gehen darf. Wenn Zuwendungen, die auf einen bestimmten Crossbar-Eingangspuffer abzielen, alle über eine einzige Warteschlange laufen, muss vor Beginn der Paketübertragung Platz für das Paket am Anfang der Warteschlange geschaffen werden. Wenn die Zuwendungen auf mehrere Warteschlangen verteilt sind, wird eine Paketübertragung erst dann gestartet, wenn im Puffer Platz für ein ganzes Paket maximaler Größe vorhanden ist, um zu verhindern, dass kleine Pakete große Pakete verdrängen. Auf diese Weise wird eine einmal begonnene Paketübertragung auf einem Zeilenbus erst dann beendet, wenn das gesamte Paket übertragen wurde. Dementsprechend sind die Eingangspuffer der Kreuzschienen so konfiguriert, dass sie groß genug sind, um die maximale Paketgröße plus zusätzlichen Platz für den ungünstigsten Fall eines Roundtrips (Paketversand bis Kreditrückgabe) zu bewältigen. Dies ist bei Jumbo-Paketen nicht der Fall. Um Pufferfläche zu sparen, sind die Crossbar-Eingangspuffer nur tief genug, um eine MTU ohne Jumbo-Größe (1500 Byte) zu verarbeiten, wobei ein Jumbo-Paket einen Zeilenbus blockieren kann, während es darauf wartet, Zugriff auf den gewünschten Spaltenbus zu erhalten.

[0037] Bei Spaltenbussen verwaltet jede Kreuzschiene die Anzahl der Guthaben für die Eingangspuffer an jedem Zielanschluss in dieser Spalte. Im Gegensatz zu Zeilenbussen ist es nicht erforderlich, dass für ein Paket maximaler Größe Guthaben verfügbar sind, bevor die Übertragung dieses Pakets auf einem Spaltenbus beginnt. Einzelne Wörter des Pakets werden übertragen, wenn Guthaben verfügbar wird. Daher muss der Eingangspuffer am Zielort für jeden Spaltenbus nur so groß sein, dass er im schlimmsten Fall den Hin- und Rückweg (Paket zu Guthaben) abdeckt.

[0038] **Abb. 3C** zeigt detaillierter eine Beispielimplementierung von zwei Anschlüssen, z. B. den Anschlüssen 0 und 1, die von Kachel 1 verwaltet werden, zusammen mit einem Kreuzschienenschalter 210, der einen Satz von Zeilenbussen und Spaltenkanälen mit Kreuzschienen pro Kachel umfasst. Auf diese Weise hat jeder Anschluss seinen eigenen Reihenbus, der über seine Reihe kommuniziert, und jede Kachel hat die bereits erwähnte 16 x 8-Kreuzschiene, die für Eckumkehrungen verwendet wird, und einen Satz von acht Spaltenkanälen, die die acht Anschlüsse in dieser Spalte versorgen. Mit anderen Worten: Jeder Kreuzschienenschalter 210 hat sechzehn Zeilenbus-Eingangspuffer und acht mögliche Ziele. Damit die Daten z. B. vom Eingangsanschluss 17 zum Ausgangsanschluss 52 gelangen, werden sie vom Eingangsanschluss 17 entlang eines Zeilenbusses geleitet, durchlaufen eine lokale Kreuzschiene, die eine 16-zu-8-Arbitrierung darstellt, und gelangen dann über einen Spaltenkanal zum Ausgangsanschluss 52. Bezogen auf die gesamte Leitweglenkung durch alle verteilten Kreuzschienen ist die interne Bandbreite viermal größer als die externe Bandbreite, was dazu führt, dass bei der Leitweglenkung nahezu jeder beliebigen Permutation des Datenverkehrs durch den Switch 202 mit dem Eingang Schritt gehalten werden kann.

[0039] Zwischen den sechzehn Quellen für jedes Ziel kann ein faires Round-Robin-Verfahren angewendet werden. Sobald eine Quelle die Arbitrierung für die Datenkreuzschiene 210d gewonnen hat, behält sie die Kontrolle über den Zielspaltenbus, bis das gesamte Paket gesendet wurde. Jeder Ausgang gewährt eine begrenzte Menge an Paketnutzlast, so dass zu erwarten ist, dass der Wettbewerb um einen bestimmten Spaltenbus bei größeren Paketen recht begrenzt sein sollte. Aus diesem Grund wird erwartet, dass eine Round-Robin-Arbitrierung selbst bei möglicherweise großen Unterschieden in der Paketgröße zwischen den Anforderern ausreichend ist.

[0040] Die Teile des Switches 202, die mit den Ausgangsfunktionen verbunden sind, arbeiten im Allgemeinen mit Frames im Switch-Fabric-Format und haben einen Fabric-Header, auch z. B. für einen Frame, der an einem Ethernet-Port innerhalb eines einzelnen Switches 202 ankommt und sich darauf stützt.

[0041] Die Ausgangskontrolle der Alterswarteschlange ist für die Annahme von Anforderungen von allen Eingangsanschlüssen, z. B. den Eingangsanschlüssen 220b, über die Anforderungsquerleiste 210a, die Pufferung der Anforderungen, die Unterscheidung zwischen ihnen nach Verkehrsklassen unter Verwendung eines Traffic Shapers und die Weiterleitung der Anforderungen an die OFCT 216 zur Gewährung über die Gewährungsquerleiste 210b verantwortlich. Die Pufferung der Alterswarteschlange wird so verwaltet, dass jeder Eingang genügend Platz für den Datenfluss hat, während einem Eingang mit mehreren Datenflüssen, die auf einen bestimmten Ausgang abzielen, mehr Platz zur Verfügung steht. Insbesondere wird der Platz in der Alterswarteschlange durch die Ausgabesteuerung verwaltet. Die Alterswarteschlange/Ausgabesteuerung kann auch für die Verwaltung des Zugriffs auf die Verbindung zuständig sein, und zwar entweder mit Hilfe einer kreditbasierten Flusskontrolle für einen angeschlossenen Eingangspuffer oder einer pausenbasierten Flusskontrolle für Nicht-Fabric-Verbindungen. Wenn ein Paket von der Warteschlange für das Alter freigegeben wird, wird es auf die Verbindung übertragen. Darüber hinaus verfügt die Alterungswarteschlange über einen Pfad, der es Paketen, die an einem bestimmten Anschluss, z. B. einem der Eingangsanschlüsse 220b, initiiert wurden (z. B. Wartungs- oder Verkleinerungspakete), ermöglicht, sich um Ressourcen an dem betreffenden Anschluss zu bemühen.

[0042] Die Anfragen kommen über einen Spaltenbus von jeder Zeile der Matrix 30 in den Ausgangssteuerblock. Jeder Spaltenbus speist ein unabhängiges FIFO (z. B. ein First-in-First-out-Schieberegister oder einen Puffer), wobei der Platz im FIFO über Kredite verwaltet wird. Die FIFOs können so bemessen sein (24 tief), dass sie einen Umlauf plus zusätzlichen Platz abdecken, damit Anforderungen aus den Querschienen 210a-210e herausgeschoben werden können und ein Blockieren des Kopfes der Zeile verhindert wird. Vor dem Schreiben in ein FIFO kann eine Anforderung auf einen gültigen Fehlerkorrekturcode (ECC) geprüft werden. Weist die ECC-Prüfung entweder einen Multi-Bit-Fehler (MBE) oder einen Single-Bit-Fehler (SBE) im Zielfeld auf (d. h. die Anforderung wurde an den falschen Anschluss weitergeleitet), wird die Anforderung als ungültig betrachtet und mit einer Fehlermarkierung verworfen.

[0043] Die LRU-Arbitrierung (Least Recently Used) kann zwischen den Spaltenbus-FIFOs durchgeführt werden, um zu entscheiden, welcher FIFO an die Alterswarteschlangenverwaltung weitergeleitet wird. Wenn Anforderungen aus jedem FIFO entfernt werden, werden Guthaben an die entsprechende Kreuzschiene zurückgegeben. Die Zeile, mit der ein eingehender Spaltenbus korrespondiert, kann sowohl davon abhängen, wo in der Matrix sich die Kachel befindet, als auch davon, in welcher Hälfte der Kachel sich der Block befindet.

[0044] Der Ausgabepuffer (OBUF) stellt Anforderungen an den Ausgabesteuerungsblock, um Reduzierungs- und Wartungspakete über eine Verbindung zu senden. Diesen Anforderungen kann die höchste Priorität eingeräumt werden. Ein FIFO mit 8 Speicherplätzen kann verwendet werden, um diese Reduktions-/Wartungspaketanforderungen zu puffern, während sie auf Ressourcen warten. Reduktionspakete brauchen keine Flusskanäle zu verwenden, und Wartungspakete können Loopback verwenden, um einen Fluss zu erzeugen, so dass eine Überprüfung der Verfügbarkeit von Flusskanälen oder ein Durchlauf durch den OFCT zur Erzeugung eines Grant nicht erforderlich ist. Reduzierungs- und Wartungspakete müssen auch keinen Platz im Ausgangspuffer verwenden, so dass keine Überprüfung des Platzes erforderlich ist. Stattdessen kann eine Prüfung auf die Eingangsbutter des Linkpartners durchgeführt werden. Wenn dies zulässig ist, kann eine Shaping Queue (SQ) oder ein virtueller Kanal (VC) gewährt werden, wodurch die Gewährung von Zuschüssen aus dem Age-Queue-Pfad während dieses Zyklus blockiert wird.

[0045] Die Größe des nächsten zu verarbeitenden Auftrags aus dem Ausgabepuffer wird mit max_frame_size abgeglichen. Übersteigt sie diese Einstellung, wird der Auftrag nicht verarbeitet und ein Fehlerflag gesetzt. Dies hat zur Folge, dass der Ausgabepuffer-Anforderungspfad blockiert wird, bis ein Warm--

Reset durchgeführt wird. Das Fehlerflag bleibt gesetzt, bis der Reset erfolgt ist. Die Bedingung kann auch aufgehoben werden, indem die Einstellung von max_frame_size auf einen Wert erhöht wird, der über der Größe der blockierten Ausgabepufferanforderung liegt. Die für den Vergleich verwendete Größe kann die in der Ausgabepufferanforderung angegebene Größe sein (die eine auf der Leitung verwendete 4-Byte-Rahmenprüfsumme (FCS) enthalten kann).

[0046] Jedem Eingang kann ein fester Platz in der Warteschlange für das Alter zugewiesen werden. Dieser Platz in der Warteschlange ist groß genug, um einen Platz für jede SQ/VC zu reservieren, mit genügend zusätzlichem Platz, um einen Request/Credit Roundtrip abzudecken. Es obliegt der Eingabe, den ihr zugewiesenen Platz für ihre SQs/VCs zu verwalten. Diese Zuweisung (fixed_al/oc) ist über ein Steuer- und Statusregister (CSR) in jeder Eingangswarteschlange (INQ) programmierbar und kann z. B. im Bereich von 64-96 Speicherplätzen liegen. Der verbleibende Platz in der Warteschlange (8K- 64*fixed_al/oc) kann ein gemeinsam genutzter Platz sein, der für alle Eingänge zur Verfügung steht. Der gemeinsam genutzte Speicherplatz kann von der Ausgabe verwaltet werden, indem eingehende Anfragen bei ihrem Eintreffen vom statischen in den gemeinsam genutzten Speicherplatz verschoben werden, sofern im gemeinsam genutzten Speicherplatz Platz vorhanden ist, vorbehaltlich der Grenzen pro Eingabe. Wenn eine Anforderung in den gemeinsam genutzten Bereich verschoben wird, wird ein Guthaben zurückgegeben, z. B. sofort über die Guthaben--Kreuzschiene 210c, wobei die Anforderung in der Warteschlange als im gemeinsam genutzten Bereich befindlich markiert wird.

[0047] Wird eine Anfrage bewilligt, so wird der gemeinsam genutzte Speicherplatz gutgeschrieben, wenn er als gemeinsam genutzter Speicherplatz markiert ist. Wenn sie nicht als gemeinsam genutzter Speicherplatz gekennzeichnet ist, wird die Anfrage als Nutzung des statischen Speicherplatzes betrachtet und ein Guthaben wird mit der Gewährung an die Eingabe zurückgegeben.

[0048] Aufgrund von Konflikten in der Credit-Crossbar 210c ist es möglich, dass Credits nicht in jeder Taktperiode gesendet werden. Dementsprechend bietet ein FIFO einen Puffer für diese vorübergehenden Unterbrechungen. Der Platz in diesem FIFO wird benötigt, bevor eine Anforderung von der Anforderungskreuzschiene übernommen wird. Ein FIFO mit einer Tiefe von 32 Speicherplätzen kann verwendet werden, um die Wahrscheinlichkeit eines Rückstaus in die Anforderungskreuzschiene 210a zu begrenzen. Der gemeinsam genutzte Speicherplatz kann Grenzen dafür haben, wie viel Platz eine Eingabe (von einem Eingangsanschluss 220b) einnehmen kann. Diese Grenzen können als Prozentsatz des verfügbaren Platzes festgelegt werden. Ist der Grenzwert beispielsweise auf 50 % festgelegt, hat ein aktiver Eingangsanschluss Zugriff auf 50 % des Pufferspeichers, bei zwei aktiven Eingangsanschlüssen erhält jeder 37,5 % ((space_used_by_l + space_left*.5)/2 = (50 %+50 %*.5)/2), bei drei aktiven Eingangsanschlüssen erhält jeder 29,2 % ((space_used_by_2 + space_left*.5)/3 = (75 %+25 %*.5)/3) usw. Darüber hinaus kann der gesamte von den aktiven Eingangsanschlüssen belegte Platz auf die angegebene Gesamtzahl (50%, 75%, 87,5%) begrenzt werden. Somit kann der jedem Eingangsanschluss 220b zugewiesene Platz dynamisch variieren, je nachdem, wie viele Eingangsanschlüsse gerade aktiv sind. Die Hinzufügung eines aktiven Eingangsanschlusses führt dazu, dass andere aktive Eingangsanschlüsse ihren Platz aufgeben, der dann von dem neuen Eingang eingenommen wird.

[0049] Da die Teilung in Hardware nicht einfach zu bewerkstelligen ist, kann die oben erwähnte Funktion zur Verwaltung der Guthaben in der Warteschlange als Nachschlagetabelle mit 64 Einträgen implementiert werden. Die Anzahl der derzeit aktiven Eingänge in den Alterswarteschlangen indiziert die Nachschlagetabelle. Die Werte in der Nachschlagetabelle spiegeln die Obergrenze für die Anzahl der gemeinsam genutzten Speicherplätze wider, die ein Eingang einnehmen kann, sowie den Gesamtspeicherplatz, den er insgesamt verbrauchen kann. Es liegt also an der Software, die Werte in der Nachschlagetabelle zu programmieren, je nachdem, wie viel gemeinsam genutzter Speicherplatz insgesamt vorhanden ist und welchen Anteil jeder Eingangsanschluss einnehmen darf. Je mehr Eingangsanschlüsse 220b aktiv werden, desto weniger Platz wird jedem Eingangsanschluss 220b zugestanden, und der insgesamt verfügbare Platz nimmt zu. Eingehende Anfragen von den Eingangsanschlüssen 220b, die diesen Grenzwert überschreiten oder insgesamt den Grenzwert für den Gesamtspeicherplatz überschreiten, dürfen keinen weiteren gemeinsamen Speicherplatz beanspruchen. Um die Anzahl der aktiven Eingangsanschlüsse 220b in den Alterswarteschlangen zu verfolgen, wird ein Satz von 64 Zählern (einer für jeden Eingangsanschluss) verwendet. Diese Zähler werden hochgezählt, wenn eine Anforderung in die Warteschlangen gestellt wird, und sie werden heruntergezählt, wenn sie herausgenommen werden (d. h. wenn sie gewährt werden). Die Anzahl der Zähler, die nicht Null sind, wird als Index in der Lookup-Tabelle verwendet. Zur Verwaltung des gemeinsam genutzten Speicherplatzes kann ein zusätzlicher Satz von 64 Zählern verwendet werden, um die aktuelle Nutzung des gemeinsam genutzten Speicherplatzes durch jeden Eingang zu verfolgen. Es kann auch ein einziger Zähler verwen-

det werden, um die Gesamtnutzung des gemeinsam genutzten Speicherplatzes zu verfolgen. Diese Zähler werden mit den aktuellen Quoten verglichen, um festzustellen, ob eine Anfrage den gemeinsam genutzten Speicherplatz nutzen darf oder nicht. Die Zähler können z. B. 13 Bit breit sein, um den maximalen Wert eines Objekts, der etwas weniger als 8K betragen kann, ausreichend zu erfassen.

[0050] Alters-Warteschlangen können einen einzigen Speicher-RAM 321 mit 8K Speicherplätzen verwenden. Diese Speicherplätze können dynamisch 32 separaten Warteschlangen zugewiesen werden (eine für jede SQ/VC), wobei jede aus einer verknüpften Liste von Speicherplätzen innerhalb des RAM 321 besteht. Dadurch kann jede SQ/VC je nach Bedarf mehr Platz beanspruchen.

[0051] Eine Alterswarteschlange kann mit einem vorderen Zeiger erstellt werden, der auf den Anfang der Warteschlange zeigt, und einem nächsten Zeiger für jede Position, der auf das nächste Element in der Warteschlange zeigt. Die letzte Position in der Warteschlange kann durch einen „Back Pointer“ angezeigt werden. Die Elemente werden vom Anfang der Warteschlange genommen und am Ende der Warteschlange eingefügt. Zusätzlich zu den oben genannten Datenstrukturen verfügt jede Warteschlange über einen FIFO mit Einträgen an ihrem Kopf. Diese FIFOs können sicherstellen, dass eine Warteschlange in jedem Takt eine Anforderung mit einer Lesezugriffszeit von mehreren Takten aus dem Anforderungs-RAM erhalten kann. Wenn eine neue Anforderung eintrifft und der Kopf-FIFO für diese Warteschlange nicht voll ist, umgeht sie den Anforderungs-RAM und kann direkt in den Kopf-FIFO geschrieben werden. Sobald die Anforderungen für eine bestimmte Warteschlange in den Anforderungs-RAM geschrieben werden, werden auch die nachfolgenden Anforderungen in den Anforderungs-RAM geschrieben, um die Reihenfolge aufrechtzuerhalten. Der Bypass-Pfad kann wieder verwendet werden, wenn sich keine weiteren Anforderungen für diese Alterswarteschlange im Anforderungs-RAM befinden und Platz im Kopf-FIFO vorhanden ist. Wenn eine Anforderung aus einem Kopf-FIFO gelesen wird und sich entsprechende Anforderungen in der Warteschlange im Anforderungs-RAM befinden, wird eine De-Queue eingeleitet. Es kann jeweils ein Head-FIFO 328 gelesen werden, so dass in jeder Taktperiode ein einzelner Dequeue-Vorgang ausgelöst werden kann. Es kann eine Logik eingebaut werden, um die verschiedenen Wettlaufbedingungen zwischen einer laufenden oder bevorstehenden Enqueue-Operation und einem gelesenen Head-FIFO zu behandeln.

[0052] Der oben erwähnte ECC-Schutz, der im Age-Queue-RAM verwendet wird, kann auf die FIFOs 328 ausgedehnt werden, um die Datenweg-Flops zu schützen. Die resultierende Struktur kann 8K Flops umfassen (32 Warteschlangen x 5 tief x SQ-Bits breit). Bei der Generierung des ECC kann die Nummer der Warteschlange des Alters in die Berechnung einbezogen (aber nicht gespeichert) werden, um die Verwaltung der freien Liste zusätzlich zu überprüfen. Bei der Überprüfung der ECC kann die Anforderung als fehlerhaft angesehen werden, wenn in den Bits der Warteschlangennummer ein MBE oder ein SBE vorhanden ist.

[0053] Ein freier Listen-RAM kann ein einfaches FIFO sein, das mit Zeigern auf alle 8K-Einträge initialisiert wird, wenn ein Reset durchgeführt wird. Es kann eine Zählung vorgenommen werden, um zu verfolgen, wie viele Einträge in der freien Liste gültig sind. Wenn Einträge entnommen werden, werden sie von der Vorderseite des FIFO gepoppt und verwendet. Wenn Einträge zurückgegeben werden, werden sie an die Rückseite des FIFO geschoben. Eine bestimmte Anzahl von Einträgen (z. B. drei Einträge) am Kopf der freien Liste kann in Flops gehalten werden, damit sie für einen schnellen Zugriff verfügbar sind. Wie bei den Kopf-FIFOs für die Alterswarteschlangen wird ECC durch die Flops getragen, um Schutz zu bieten. Die resultierende Struktur kann minimale Flops haben (57 = 3 tief x 19 Bit breit).

[0054] Um die volle Leistung für kleine Pakete zu erreichen, unterstützen Age-Warteschlangen sowohl ein Enqueue als auch ein Dequeue pro Taktperiode. Die Operationen in den Datenstrukturen für eine Enqueue--Operation werden im Folgenden erläutert und können sich unterscheiden, je nachdem, ob die zu schreibende Warteschlange leer ist oder nicht.

[0055] In einigen Fällen ist ein gleichzeitiger Enqueue- und Dequeue-Vorgang in einer bestimmten Warteschlange leicht zu handhaben, da sie separate Felder verwenden und aktualisieren. Einige spezielle Szenarien können auftreten, z. B. wenn eine Dequeue-Operation die Alterswarteschlange leert. Um dieses Szenario zu behandeln, wird logischerweise zuerst eine Dequeue-Operation durchgeführt, gefolgt von einer Enqueue-Operation. Dementsprechend wird ein Leer-Kennzeichen als gesetzt angesehen, wenn die Warteschlange durch die Dequeue-Operation geleert wird, und dann durch die Enqueue-Operation gelöscht.

[0056] Die oben erwähnte Arbitrierung kann zwischen Anforderungen durchgeführt werden, die vorbehaltlich der Verwaltung des Eingangspuffers, des Ausgangspuffers und der Flusskanalquoten gewährt werden dürfen. Die Arbitrierung kann auch angehalten werden, wenn keine Guthaben für den OFCT-Eingangs-FIFO vor-

handen sind. In einigen Ausführungsformen kann die Arbitrierung auf zwei Ebenen durchgeführt werden. Erstens kann die Traffic Shaping Arbitration zur Arbitrierung zwischen den SQs verwendet werden. Eine Deficit--Round-Robin-Arbitration kann verwendet werden, um zwischen VCs innerhalb einer bestimmten SQzu arbitrieren. Die Traffic Shaping Arbitration kann eine Reihe von Token Buckets verwenden, um die Bandbreite jeder SQ wie folgt zu steuern: acht Leaf Buckets, eines für jede SQ, vier Branch Buckets und ein Head Bucket.

[0057] Die Schlichtung kann in drei Gruppen unterteilt werden, wobei die erste Gruppe die höchste Priorität hat, gefolgt von einer zweiten Gruppe, die wiederum von einer dritten Gruppe gefolgt wird. Für die erste und zweite Gruppe kann die Schlichtung unter den in Frage kommenden SQs auf die gleiche Weise erfolgen. Zwischen den SQs für jede der acht Prioritätsstufen kann eine x8-Rundlauf-Schlichtung durchgeführt werden (acht parallele Rundlauf-Schlichtungen). Zwischen den Prioritätsstufen kann eine feste Arbitrierung durchgeführt werden. Die Arbitrage der Gruppe 3 hat beispielsweise keine Prioritäten und ist daher einfach eine einzige x8-Rundlauf-Arbitrage.

[0058] Bei der Schlichtung in der ersten Gruppe ergibt sich die Priorität für jeden aus der Einstellung in den Blattbereichen. Für die Arbitrierung in der zweiten Gruppe ergibt sich die Priorität aus der Einstellung in den Zweigen der Blattbereiche. In allen Fällen sind die Bereiche, die als für diese Gruppe in Frage kommend geprüft werden, auch die Bereiche, aus denen Paketgrößen-Token bezogen werden, wenn die Anfrage die Schlichtung gewinnt.

[0059] Was die Auswahl der Warteschlange betrifft, so können Pakete klassifiziert werden, um die SQ auszuwählen, an die ihre Anfrage weitergeleitet wird. Dadurch kann der mit einer Anwendung verbundene Verkehr anders gestaltet werden als der Verkehr, der von einer anderen Anwendung oder einer anderen Verkehrsklasse stammt. Dies kann an den Edge-Ports, die mit einer Netzwerkkarte verbunden sind, von Nutzen sein, da die Anwendungen so konfiguriert sind, dass sie einen Teil der Ressourcen des Knotens nutzen, und ihnen dementsprechend ein Teil der Netzwerkbandbreite zugewiesen wird. Gemäß einer Ausführungsform erfolgt diese Klassifizierung durch Einteilung der Pakete in eine Verkehrsklassenkennung (FTAG), z. B. einen 4-Bit-Code, der Teil des Fabric-Frame-Headers ist, und eine VLAN-ID (VNI), wenn das Paket in die Fabric eintritt. Die FTAG und VNI können dann verwendet werden, wenn das Paket die Fabric verlässt, um die Shaping-Warteschlange auszuwählen.

[0060] Ein Konfigurationsregister kann verwendet werden, um FTAGs auf SQs abzubilden. Diese Konfiguration stimmt mit der entsprechenden Konfiguration in der Eingangswarteschlange überein. Wenn der Ausgangspuffer Guthaben von Verbindungspartnern anfordert oder zurückgibt, wandelt er eine bestimmte FTAG in eine SQ um. Für die Paketinjektion befindet sich die FTAG in R_TF_OBUF_CFG_PFG_TX_CTRL. Bei der Testerzeugung ist die FTAG im Teststeuerungsregister zu finden. Wenn die Reduktionsmaschine (RED) eine Kreditrückgabe anfordert, befindet sich die FTAG im ret_cdtJtag. Wenn ein Reduktionsrahmen aus dem Ausgabestrom entfernt wird und Guthaben des Verbindungspartners zurückgegeben werden müssen, ist die FTAG im Rahmenkopf zu finden.

[0061] Was die hier erörterten SQs betrifft, so kann jede Alterswarteschlange 32 SQs haben, die durch {SO, VC} adressiert werden. Die 3-Bit-SQ kann als Shaping-Funktion betrachtet werden, und der VC wählt eine von vier Warteschlangen innerhalb dieser Shaping-Funktion aus. Bei Ethernet-Egress-Ports (Edge-Ports) wird der VC zur Vermeidung von Deadlocks nicht benötigt. Dementsprechend können alle 32 SQs verfügbar sein. In einem solchen Szenario kann die SQ 330 ausgewählt werden, indem die SQ-Basis aus R_TF _OBUF _CFG_FTAG_SQ_MAP zu den unteren Bits der VNI addiert wird. Die 5-Bit-Summe legt die {SQ,VC} fest, die an die Alterswarteschlange zu senden ist. Es ist zu beachten, dass bei der Injektion von Frames an einem Egress-Port keine VNI verfügbar ist und daher direkt eine SQ-Basis verwendet werden kann. Bei Fabric Links wird die SQ aus den oberen drei Bits der SQ-Basis entnommen. Die VC kann aus dem Frame-Header entnommen werden, wenn Guthaben für Reduktionsframes zurückgegeben werden, oder aus der entsprechenden Steuer-CSR (R_TF_OBUF_CFG_TEST_CTRL oder R_TF _OBUF_CFG_PFG_TX_CTRL), wenn Frames injiziert werden.

[0062] Die Verwaltung des Eingangspuffers eines Verbindungspartners kann von der Art des Geräts abhängen, an das die Verbindung angeschlossen ist. Geräte wie der Switch 202 können eine kreditbasierte Flusskontrolle verwenden, bei der jeder Kredit eine Speicherzelle im Eingangspuffer darstellt. Andere Geräte können eine Standard-Ethernet-Pause oder eine auf Prioritätspausen basierende Flusssteuerung verwenden. Anfragen, die als lokal beendet gekennzeichnet sind (lac term set), müssen die Flusskontrolle des Eingangspuffers des Verbindungspartners nicht berücksichtigen und brauchen keine zugehörigen Zähler zu aktualisie-

ren. Der Speicherplatz des Verbindungspartners muss nicht berücksichtigt werden, wenn sich die Verbindung im Entleerungszustand befindet.

[0063] Für die kreditbasierte Flusskontrolle kann der Linkpartner-Eingangspuffer in acht Pufferklassen unterteilt werden. Jede SQ kann einer dieser 8 Pufferklassen zugewiesen werden. Für jede der Pufferklassen werden Guthaben verwaltet, wobei jedes Guthaben 32 Byte Speicherplatz im Link-Partner-Eingangspuffer darstellt. Damit die auf Guthaben basierende Flusskontrolle mit verschiedenen Geräten (Switch, Enhanced NIC) funktionieren kann, von denen jedes unterschiedliche Zellengrößen haben kann, ist die Zellengröße ein programmierbarer Wert in Einheiten von 32 Byte.

[0064] Es kann zwei Sätze von VCs geben, wobei jeder SQ einem Satz zugewiesen ist. Für jeden VC kann eine maximale Rahmengröße an Speicherplatz reserviert werden, und jeder VC-Satz kann eine andere maximale Rahmengröße haben. Der verbleibende Teil des Linkpartner-Eingangspuffers ist gemeinsam genutzter dynamischer Speicherplatz, der von jeder SQ/VC genutzt werden kann, vorbehaltlich der Grenzen pro VC und Pufferklasse.

[0065] Die Größe, die mit der Anforderung kommt, stellt die Größe des Pakets auf der Leitung dar, die einen 4-Byte-FCS enthält. Diese wird beim Link-Partner in eine interne 2-Byte-FCS umgewandelt, bevor das Paket in den Eingangspuffer des Link-Partners geschrieben wird, so dass die Gutschrift diesen Unterschied berücksichtigen muss, der ein Faktor an der Grenze der Zellengröße sein kann. Bei einer Zelle mit einer Größe von 96 Byte wird beispielsweise für eine Zelle mit einer Größe von 97 oder 98 Byte eine einzelne Zelle benötigt. Um zu wissen, wann dies der Fall ist, enthält die Anfrage einen Korrekturterm, der wie folgt berechnet wird: req.len_correct= (byte_len % 16) == 1 oder 2.

[0066] Eine weitere Validierung dieses Begriffs ist erforderlich, um ihn in eine beliebige Zellgrößengrenze umzuwandeln. Er ist gültig, wenn die Länge die Zellgröße knapp überschreitet. Damit kann der validierte Term fen_correct wie folgt bestimmt werden: len_correct = (((16-Byte-Größe)% (2*32-Byte-Zellengröße))== 1) & req. len_correct

[0067] Ein Beispiel dafür, wie diese Werte für einige Zellen- und Paketgrößen funktionieren, ist in der folgenden Tabelle dargestellt:

[0068] Länge Korrekte Berechnung

Größe (Bytes)	Req len_correct	Größe (16B Einheiten)	Zellengröße (32B Einheiten)	len_corect Modulo-Ergebnis	len_corect	Kredit Genommer
64	0	4	2	0	0	2
65	1	5	2	1	1	2
66	1	5	2	1	1	2
67	0	5	2	1	0	3
96	0	6	3	0	0	3
97	1	7	3	1	1	3
9B	1	7	3	1	1	3
99	0	7	3	1	0	4
128	0	8	4	0	0	4
129	1	9	4	1	I	4
130	1	9	4	1	1	4
131	0	9	4	1	0	5

[0069] Die mit der Anfrage gelieferte Größe verwendet 8-Byte-Einheiten, und die Zellengröße des Link-Partner-Eingabepuffers ist ein Vielfaches von 32 Byte (32*y, wobei y = Zellengröße aus dem CSR). Zunächst wird die 8-Byte-Größe in eine 16-Byte-Größe umgewandelt (ROUNDUP((8-Byte-Größe)/2)). Auch die Zellengröße

wird in 16-Byte-Einheiten umgerechnet (2*y). Mathematisch lässt sich die Anzahl der Zellen, die eine Anfrage benötigt, wie folgt berechnen: ROUNDDN(((I6-Byte-Größe)+ 2*y - 1 - len_correct)/(2*y)) = Anzahl der Zellen

[0070] Eine Teilung ist zwar in der Hardware möglich, kann aber aus zeitlichen Gründen nicht im kritischen Pfad der Arbitrierung durchgeführt werden. Stattdessen wird eine alternative Kreditverwaltung verwendet. Das heißt, die Guthaben werden in Einheiten von 32 Bytes gehalten. Wenn eine Anfrage die Schlichtung gewinnt, wird die Anzahl der in Anspruch genommenen Guthaben um den maximalen Fehlerterm (2*y-1) nach folgender Berechnung angepasst ROUNDDN(((I6-Byte-Größe)+ 2*y - 1)/2) = maximal benötigte 32-Byte-Credits. Da diese Berechnung das für das Paket benötigte Guthaben überschätzt, muss im nächsten Takt eine Modulo-Operation (X = (16-Byte-Größe) MOD 2*y, y = 32-Byte-Zellengröße aus dem CSR) durchgeführt werden, um den tatsächlichen Restwert zu ermitteln. Dieser Wert wird zusammen mit dem len_correct term zur Anpassung des Kreditzählers verwendet. Die Formel, mit der der Anpassungswert (adf_val) für X erstellt wird, lautet: Wenn (X == 0) adj_val = y - 1 sonst wenn (X == 1 und fen _correct) adj_val = y sonst adj_val = ROUNDDN((X-1)/2)

[0071] Die nachstehende Tabelle zeigt ein Beispiel für ein Anforderungsguthaben für 96-Byte-Zellen, das die über mehrere Paketlängen hinweg verwendeten Werte für die 96-Byte-Zellen des Schaltereingangspuffers (y = 3) angibt.

[0072] Beispiel für eine Kreditanfrage für 96-Byte-Zellen

Paketgröße (Bytes)	Paketgröße (16 Byte Einheiten)	Kredit genom-men	Modulo Ergebnis	len_correct	adj_val	Korrigierter Kre-dit genommen
48	3	4	3	0	1	3
64	4	4	4	0	1	3
80	5	5	5	0	2	3
96	6	5	0	0	2	3
97	7	6	1	1	3	3
98	7	6	1	1	3	3
99	7	6	1	0	0	6
128	8	6	2	0	0	6

[0073] Wenn eine Anfrage gefiltert wird, bevor sie an den Eingangspuffer des Linkpartners weitergeleitet wird, gibt die Logik des Ausgangspuffers die SQ und VC zurück, so dass sie verwendet werden können, um die Guthaben an die entsprechenden Guthabenzähler zurückzugeben. Es ist keine Größe erforderlich, da die Paketgröße immer die gleiche ist, nämlich die Länge eines Reduktionsrahmens (69 Byte oder 16 Byte Größe= 5).

[0074] Die lokale (Master-)Seite der Verbindung führt eine Zählung der Anzahl der von jedem VC über beide Sätze gesendeten Pakete (insgesamt 8), eine Zählung der an jeden VC gesendeten Paketmenge (in 3 2-Byte-Mengen) (4) und eine Zählung der für jede Pufferklasse gesendeten Paketmenge (in 32-Byte-Mengen) (8). Der Link-Partner (Slave) der Verbindung behält den gleichen Satz von Zählungen bei, wobei diese regelmäßig über die Verbindung gesendet werden. Der Unterschied zwischen den Zählungen auf der Master- und der Slave-Seite besteht in einer Zählung der Anzahl der Pakete im Eingangspuffer des Link-Partners von jedem VC über beide Sätze hinweg sowie in einer Zählung des derzeit von jedem VC und jeder Pufferklasse belegten Platzes (in 32-Byte-Mengen). Außerdem wird die Gesamtmenge des von allen Paketen belegten Platzes gezählt. Eine Zusammenfassung der Zähler lautet wie folgt: master_vcx_cnt[4]/slave_vcx_cnt[4] - Master- und Slave-Zähler für die Anzahl der Pakete, die an jeden VC im Satz X gesendet wurden;
master_vcy_cnt[4]/slave_vcy_cnt[4] - Master- und Slave-Zähler für die Anzahl der Pakete, die an jeden VC im Satz Y gesendet wurden; master_bc_cnt[8]/slave_bc_cnt[8] - Master- und Slave-Zählungen des von jeder Pufferklasse belegten Speicherplatzes in Einheiten von 32 Byte; master_vc_cnt[4]/slave_vc_cnt[4] - Master- und Slave-Zählungen des von jedem VC belegten Speicherplatzes in Einheiten von 32 Byte; master- tot-cnt/slave- tot - cnt - Master- und Slave-Zählungen des insgesamt belegten Speicherplatzes in Einheiten von 32 Byte.

[0075] Alle Zähler werden bei einem Warm-Reset auf Null gesetzt. Sie werden auch auf Null gesetzt, wenn sich die Verbindung im Entleerungszustand befindet oder wenn das CSR-Bit DBG_RESET zum Löschen ihres Zustands gesetzt ist. Der Ausgangspufferfilter lenkt ein Reduktionspaket auf einen anderen Weg als den zum Eingangspuffer des Link-Partners. In diesem Fall kann ein Signal zusammen mit dem SQ und VC des Pakets zurückgegeben werden. Auch hier ist die Länge nicht erforderlich, da die Größe dieser Pakete festgelegt ist. Diese Information wird verwendet, um die entsprechenden Master Credit Counts anzupassen.

[0076] Eine Anfrage darf an der Arbitrierung teilnehmen, wenn entweder ihre VC-Anzahl 0 ist (was anzeigt, dass ihr ein statisch zugewiesener Slot zur Verfügung steht) oder im dynamischen Bereich Platz für einen Rahmen maximaler Größe ist (vorbehaltlich der angestrebten Pufferklasse und VC-Grenzen). Es kann einen einzigen programmierbaren Wert für die maximale Rahmengröße geben, der für alle VCs und SQs verwendet wird. Die Anforderungsvalidierung für den Eingangspufferspeicher kann mit Hilfe der kreditbasierten Flusskontrolle erfolgen.

[0077] Die kreditbasierte Flusskontrolle kann zur Aufteilung eines dynamischen Raums auf zwei voneinander unabhängige Arten verwendet werden: erstens auf der Grundlage eines Limits, wie viel dynamischen Raum jeder der vier VCs einnehmen kann, und zweitens auf der Grundlage eines Limits, wie viel dynamischen Raum jede der acht Pufferklassen einnehmen kann. In beiden Fällen werden die Grenzen als Prozentsatz des verfügbaren Platzes festgelegt. Für ein bestimmtes Paket sollte sowohl in seinem Ziel-VC als auch in seiner Pufferklasse Platz zur Verfügung stehen. Wenn z. B. für jeden Bereich ein Limit von 50 % festgelegt ist, hat ein aktiver Bereich Zugriff auf 50 % des Pufferbereichs, bei zwei aktiven Bereichen erhält jeder Bereich 37,5 % ((50+50*.5)/2), bei drei aktiven Bereichen erhält jeder Bereich 29,2 % ((75+25*.5)/3) und so weiter. Außerdem kann der Gesamtplatz, der von den aktiven Plätzen belegt wird, auf die angegebene Gesamtzahl (50%, 75%, 87,5%) begrenzt werden. Dementsprechend variiert der jedem Platz zugewiesene Platz dynamisch je nachdem, wie viele Plätze gerade aktiv sind. Wenn ein zusätzlicher Platz aktiv wird, müssen die anderen aktiven Plätze einen Teil ihres Platzes abgeben, der dann von dem neuen Platz eingenommen wird.

[0078] Wie die oben beschriebene Teilungsfunktion ist auch diese Funktion als Nachschlagetabelle implementiert. Für den VC-Speicherplatz gibt es in diesem Beispiel 16 Einträge, wobei jeder Eintrag den für jeden VC verfügbaren Speicherplatz sowie den insgesamt für alle VCs verfügbaren Speicherplatz angibt. Für die Pufferklassen kann es 256 Einträge geben, wobei jeder Eintrag den für jede Pufferklasse verfügbaren Platz sowie den für alle Pufferklassen insgesamt verfügbaren Platz angibt. Der Platz wird jeweils in 2048-Byte-Einheiten angegeben. Die Tiefe jeder Tabelle reicht aus, um alle Kombinationen aktiver Mitglieder (VCs oder Pufferklassen) abzudecken, wobei jede eine unabhängige Einstellung für ihre Prozentsätze haben kann. Damit ist es Sache der Software, die Werte in der Tabelle zu programmieren, je nachdem, wie viel dynamischer Speicherplatz insgesamt zur Verfügung steht und welchen Prozentsatz jedes Mitglied über alle möglichen Kombinationen hinweg einnehmen darf. Je mehr davon aktiv werden, desto weniger Platz wird ihnen zugestanden und desto mehr Platz ist insgesamt verfügbar. Anfragen nach Speicherplatz, die über diesem Grenzwert oder insgesamt über dem Gesamtgrenzwert liegen, dürfen keinen weiteren dynamischen Speicherplatz beanspruchen.

[0079] Eine VC- oder Pufferklasse gilt als aktiv, wenn sie entweder eine Anforderung in einer Alterungswarteschlange hat oder wenn sie ausstehende Gutschriften für Linkpartner-Eingangspufferplätze hat. Nehmen wir als Beispiel an, es gibt nur 4 Pufferplätze (Tabelle mit 16 Einträgen) mit den Prozentsätzen SPACE0 (50%), SPACE1(40%), SPACE2(30%), SPACE3(10%) und einem dynamischen Gesamtspeicherplatz von 16KB. Daraus ergeben sich die in der nachstehenden Beispieltabelle für den Pufferspeicher angegebenen Werte in Mengen von 16 Bytes.

[0080] Beispiel für Pufferraum

Index	RAUM3	RAUM2	RAUM1	RAUM0	Gesamt
0	N/A	N/A	N/A	N/A	N/A
1	N/A	N/A	N/A	512	512
2	N/A	N/A	410	N/A	410
3	N/A	N/A	319	398	717
4	N/A	307	N/A	N/A	307

Index	RAUM3	RAUM2	RAUM1	RAUM0	Gesamt
5	N/A	250	N/A	416	666
6	N/A	255	339	N/A	594
7	N/A	202	270	337	809
8	102	N/A	N/A	N/A	102
9	94	N/A	N/A	469	563
10	94	N/A	377	N/A	471
11	75	N/A	299	374	748
12	95	284	N/A	N/A	379
13	78	234	N/A	389	701
14	80	239	319	N/A	638
15	79	236	315	394	1024

[0081] Als Beispiel werden die Werte in der Zeile für Index 7 wie folgt berechnet: Gesamt%= 0.5 + (1-0.5) *0.4 + (1-0.5-(1-0.5)*0.4)*0.3 = 0.79; SPACEO = (0.5/(0.5+0.4+0.3))*0.79*1024 = 337; SPACE) = (0.4/(0.5 +0.4+0.3))*0.79*1024 = 270; SPACE2 = (0.3/(0.5+0.4+0.3))*0.79*1024 = 202; Gesamt= 337 + 270 + 202 = 809

[0082] Wie oben erwähnt und unter Bezugnahme auf **Abb. 2**, können Switches, wie z. B. der Switch 202, verwendet werden, um eine Switch-Fabric zu erstellen, wobei die Switch-Ports 220 so konfiguriert werden können, dass sie entweder als Edge-Ports oder als Fabric-Ports arbeiten. Wie bereits erwähnt, kann der Switch 202 verschiedene Netzwerktopologien unterstützen, einschließlich, aber nicht beschränkt auf, z. B. Libellen- und Fat-Tree-Topologien. Netzwerke können aus einem oder mehreren Slices bestehen, die jeweils die gleiche Gesamttopologie aufweisen, obwohl sich die Slices in Bezug auf ihre Zusammensetzung unterscheiden können. Die Knoten sind mit einem oder mehreren Ports in jeder Slices verbunden. Wenn ein Netz mehrere Slices hat und ein Knoten mit mehr als einem Slice verbunden ist, wird angenommen, dass der Knoten in jedem Slice an der gleichen Stelle angeschlossen ist.

[0083] Das Routing in der Switch-Fabric kann durch eine in Switch 202 implementierte Fabric-Routing-Funktion (FRF) gesteuert werden. Ein Beispiel für eine FRF-Komponente 500 ist in den **Abb. 5A** und **Abb. 5B** dargestellt. Es versteht sich, dass eine separate Instanz der FRF-Komponente 500 innerhalb der Eingangslogik für jeden Port des Switches 202 implementiert werden kann. Die von der FR-Komponente 500 getroffenen Routing-Entscheidungen können auf diejenigen Frames angewendet werden, die nicht bereits Teil eines etablierten Flusses sind. Es ist zu beachten, dass die FRF-Komponente 500 nicht unbedingt weiß, ob ein bestimmter Rahmen mit einem Fluss verbunden ist oder nicht, sondern vielmehr eine unabhängige Weiterleitungsentscheidung für jeden an einem Eingangsport präsentierten Rahmen trifft. Die FRF-Komponente 500 kann Filter, Tabellen, Schaltkreise und/oder Logik, wie z. B. Auswahlschaltkreise/Logik, umfassen, um die Weiterleitung von Daten durch eine Switch-Fabric, wie hier beschrieben, zu bewirken. Wie dargestellt, umfasst die FRF-Komponente 500 mindestens: eine minimale Ports-Auswahlkomponente 502 (die eine minimale Tabellenkomponente 502A umfasst), verschiedene Ports-Filter (Filter für zugelassene Ports, Filter für betriebsbereite Ports, Filter für belegte Ports); eine Komponente zur Unterscheidung von bevorzugten Ports 502B; Pseudo-Zufallsauswahlkomponenten/-logik 502C; Ausnahmetabellen 504 (einschließlich einer Ausnahmelistentabelle 504A); eine Komponente für betriebsbereite Ports 506, die eine globale Fehlertabelle 506A umfasst; und eine Routing-Algorithmus-Tabelle 508. Wie in **Abb. 5B** dargestellt, kann die FRF-Komponente 500 ferner Folgendes umfassen: eine Komponente zur Auswahl nicht minimaler Ports (510), die eine lokale Komponente zur Auswahl nicht minimaler Ports (510A) und eine globale Komponente zur Auswahl nicht minimaler Ports (510B) umfasst; und eine Ausgangslogikkomponente (512) (die Teil des Ausgangssteuerblocks eines Schalters ist), die eine adaptive Auswahlkomponente oder -logik (512A) umfasst, die wiederum eine Vorspannungskomponente (514) mit einer Vorspanntabelle (514A) enthält. Die FRF-Komponente 500 enthält weitere Komponenten, die hier beschrieben werden.

[0084] Insbesondere bestimmt die FRF-Komponente 500 mit dem Diskriminator 502B für bevorzugte Ports einen bevorzugten Port, um jeden am Eingangsport präsentierten Frame weiterzuleiten, und zwar auf der Grundlage der Ziel-Fabric-Adresse (DFA) eines empfangenen Frames, des aktuellen Routing-Zustands des

Frames (wo sich der Frame auf seinem Pfad befindet und welchen Pfad bzw. welche Pfade er genommen hat, um seinen aktuellen Routing-Zustand zu erreichen), des Routing-Algorithmus und der Konfiguration der Switch-Fabric und der mit dem Ausgangsport (dem oben erwähnten bevorzugten Port, an den der Frame weitergeleitet werden soll) verbundenen Lastmetriken unter Verwendung von Filtern für belegte Ports.

[0085] Die FRF-Komponente 500 kann eine Routing-Algorithmus-Tabelle 508 enthalten, die als softwarekonfigurierbare Tabelle ausgeführt sein kann, die auf der Grundlage des aktuellen Routing-Zustands des Rahmens gültige Auswahlmöglichkeiten bestimmt. Gültige Entscheidungen sind beispielsweise, ob ein lokaler minimaler, globaler minimaler, lokaler nicht-minimaler oder globaler nicht-minimaler Pfad für den nächsten Hop des Frames gewählt werden darf. Der Routing-Status enthält Informationen wie den VC, auf dem der Frame empfangen wurde, und ob er sich in der Quell-, der Ziel- oder einer Zwischengruppe befindet. Die Routing-Algorithmus-Tabelle 508 bestimmt zusammen mit der adaptiven Auswahlfunktion oder -logik 512A (wie unten beschrieben) auch den VC, der für den nächsten Sprung des Rahmens verwendet wird.

[0086] Als Beispiel wird das Frame-Routing mit Unicast-DFAs beschrieben. Es ist jedoch zu beachten, dass die DFA der Routing-Anforderung entweder im Unicast- oder im Multicast-Format vorliegen kann. Das Unicast-Format kann ein 9-Bit-Global-ID-Feld (global_id), ein 5-Bit-Switch-ID-Feld (switch_id) und ein 6-Bit-Endpunkt-ID-Feld (endpoint_id) enthalten. Die globale ID kann eine Gruppe innerhalb des Netzes eindeutig identifizieren. Sie identifiziert insbesondere die letzte Gruppe, an die der Rahmen zugestellt werden muss. Die Switch-ID identifiziert eindeutig einen Switch innerhalb der durch die globale ID identifizierten Gruppe. Das Feld Endpunkt-ID identifiziert zusammen mit der globalen ID und der Switch-ID den Endpunkt, der mit dem Rand der Netzwerkstruktur verbunden ist und an den der Rahmen zugestellt werden soll. Dieses Feld wird einem Port oder einer Reihe von Ports auf dem Switch zugeordnet, der durch die globale ID und die Switch-ID identifiziert wird.

[0087] Das Multicast-Format enthält ein 13-Bit-Multicast-ID-Feld (multicast_id). Dieses Feld wird von der FRF-Komponente 500 auf eine Reihe von Ports am aktuellen Switch abgebildet, an die der Rahmen weitergeleitet werden soll.

[0088] Aus diesen Informationen ermittelt die FRF-Komponente 500 einen aktualisierten Routing-Status für den Frame, der dann in den Frame übertragen wird. Um beispielsweise das Routing in einer Libellen-Topologie zu realisieren, kann der aktuelle Status eines Rahmens aus dem VC des Rahmens (wie oben beschrieben) abgeleitet werden. Basierend auf algorithmischen Switch Fabric Routing-Regeln, die für die Switch Fabric spezifiziert sind (deren Auswahl weiter unten beschrieben wird), bestimmt die FRF-Komponente 500 einen bestimmten VC, der für den nächsten Hop des Frames verwendet wird, um Deadlocks zu vermeiden. Je nachdem, wo sich der Frame auf seinem Weg befindet, z. B. ob er sich in seiner Quellgruppe, in einer Zwischengruppe oder in seiner Zielgruppe befindet, können zusätzliche Routing-Statusinformationen bereitgestellt werden. Es sei darauf hingewiesen, dass die FRF-Komponente 500 eine Port-Filterung durchführt (die weiter unten ausführlicher beschrieben wird), indem sie Filter für zugelassene Ports, Filter für in Betrieb befindliche Ports, Filter für belegte Ports usw. verwendet, um festzustellen, ob ein bevorzugter Port, an den ein Frame weitergeleitet werden soll, derzeit fehlerhaft, besetzt, nicht vorhanden usw. ist.

[0089] Die FRF-Komponente 500 erhält Lastmessungen für ihren Ausgangsanschluss vom Switch 202. Lastinformationen für den Eingangsanschluss der FRF-Komponente 500 werden auch von einem benachbarten Switch empfangen. In einigen Ausführungsformen tauscht die FRF-Komponente 500 ihre Eingangs- und Ausgangsanschluss-Lastinformationen mit allen anderen FRF-Komponenteninstanzen innerhalb eines Switches, z. B. Switch 202, und mit benachbarten Switches in der Switch-Fabric aus. Auf diese Weise kennt jede FRF-Komponenteninstanz jedes Switches in der Switch-Fabric die zusammengefassten Lastinformationen für alle benachbarten Switches.

[0090] Es ist zu beachten, dass die FRF-Komponente 500 das Multicasting von Rahmen unterstützen kann. Wenn ein Multicast-DFA empfangen wird, bestimmt die FRF-Komponente 500 eine Reihe von Anschlüssen, an die der mit dem Multicast-DFA verbundene Rahmen weitergeleitet werden soll. Der Satz von Ports kann durch Zugriff auf eine Nachschlagetabelle bestimmt werden, die softwarekonfigurierte Multicast-Fabric-Adressen auf Ausgangsports abbildet. Dadurch werden Probleme im Zusammenhang mit doppelten Multicast--Rahmenkopien vermieden.

Routing pro Verkehrsklasse

[0091] Traditionell werden Verkehrsklassen (Traffic Classes, TCs) und Dienstgüte (Quality of Service, QoS) eingesetzt, um vorhersehbare Laufzeiten zu garantieren und die Leistung von Anwendungen zu verbessern, indem auf der Grundlage der Zuweisung von Netzressourcen und Eigenschaften wie Verkehrspriorität, Bandbreitenanteil oder maximale Latenzzeit Garantien für das Dienstniveau gegeben werden. In der Regel beziehen sich diese Mechanismen auf Latenz- und Bandbreiteneigenschaften. Wie bereits angedeutet, wirken sich die hier offengelegten Routing-Techniken pro Verkehrsklasse auf Eigenschaften außerhalb von Latenz und Bandbreite aus, einschließlich des Routing-Verhaltens, der Trennung von Anwendungsdatenströmen und der Staukontrolle auf einer Basis pro Verkehrsklasse. Dementsprechend ermöglichen die vorgestellten Techniken eine unabhängige, feinkörnige Steuerung des Datentransits im Netzwerk, die in der Lage ist, die Anforderungen von HPC-Anwendungen, -Diensten und -Workflows zusätzlich zu den herkömmlichen Ethernet- und TCP-Netzwerkmerkmalen direkt zu unterstützen.

[0092] So sind TCs derzeit im Ethernet-Bereich etabliert, aber viele der Klassifizierungen und Verhaltensweisen sind am besten für Internet- oder Rechenzentrumsanwendungen und Verkehrsmuster geeignet. Diese Klassifizierungen werden größtenteils durch die Beeinflussung von Paket-Warteschlangen, Puffern und der Arbitrierung über sie erreicht, um zu bestimmen, wie Datenpakete vom Eingang zum Ausgang weitergeleitet werden.

[0093] Die vorgestellten Techniken unterstützen bereits etablierte Klassifizierungen und konzentrieren sich dabei speziell auf HPC-Anwendungen, HPC-bezogene Merkmale und HPC-bezogene Anforderungen für bestimmte Netzwerkverhaltensweisen und Arbeitsabläufe. Zur besseren Unterstützung von HPC-Anwendungen und -Diensten wurden Mechanismen in die Switches (siehe **Abb. 1**) integriert, um spezifische Funktionen bereitzustellen, die vielen HPC-Workflows zugrunde liegen. Diese Funktionen werden im gesamten Netzwerk aktiviert, konfiguriert und abgestimmt, wobei zusätzliche Verkehrsklassifizierungen verwendet werden, die über die derzeit etablierten hinausgehen, um den Datenfluss und die Netzwerkeigenschaften zu bestimmen, z. B. geordnete oder ungeordnete Datenübertragung, verlustbehaftete oder verlustfreie Übertragung, Routing-Richtlinien und Regeln für die Verkehrsgestaltung. Mehrere unabhängige TCs können gleichzeitig über dieselbe physische Netzinfrastruktur betrieben werden, so dass Routing- und Datenflusseigenschaften pro TC mit spezifischen Anwendungen und Diensten verknüpft werden können.

[0094] Neben netzwerkbasierten Datenflussklassifizierungen werden auch HPCanwendungsspezifische Mappings von diesem neuen Modell unterstützt. Die Klassifizierung der prozessübergreifenden Kommunikation innerhalb einer HPC-Anwendung kann auf verschiedene Weise erfolgen, unter anderem:

- Automatische, auf der Anwendungsphase basierende Klassifizierung von Dateneingang und -ausgang sowie von Anwendungssynchronisierungsbarrieren und -reduzierungen. Für diese Form der Klassifizierung sind keine Änderungen am Softwarecode erforderlich.
- Durch die Unterstützung von HPC-Kommunikationsbibliotheken (in MPI oder LibFabrics) werden Bibliotheksaufrufe entsprechend den spezifischen oder allgemein etablierten Aufrufkonventionen innerhalb einer Anwendung klassifiziert. Diese Form der Klassifizierung erfordert die Verknüpfung einer Anwendung mit einer Bibliothek, die diese neuen HPC-Klassifizierungen unterstützt.
- Durch explizite Richtlinien im Anwendungscode selbst, die eine feinkörnige Kontrolle der Klassifizierung ermöglichen. Diese Klassifizierung erfordert eine Änderung der Anwendung selbst, bietet aber die höchste Kontrollstufe.

[0095] Gemäß den Ausführungsformen wird der Verkehrsfluss einer Anwendung identifiziert und bestimmten TCs zugeordnet, indem Pakete innerhalb eines Flusses beim Austritt aus der Netzschnittstelle eines Knotens in das Netz markiert werden. Ein Feld im Paketkopf enthält einen Code Point, eine Bitmap, die ein oder mehrere bevorzugte Netzwerkverhaltensweisen angibt. Bei der Klassifizierung durch das Netz wird der Verkehr in mehrere Klassen eingeteilt, die jeweils auf unterschiedliche Weise behandelt werden und unterschiedlichen Routing-, Vorrang-, Formungs- und Planungsregeln unterliegen.

[0096] Network implementiert die Klassifizierung, Formung und Kontrolle des Netzwerkverkehrs auf ähnliche Weise wie die Modelle Differentiated Services und Precedence, bei denen Pakete, die zu den Datenströmen einer Anwendung gehören, durch das Setzen von Bits in reservierten Feldern des Paketkopfes gekennzeichnet werden, die Differentiated Service Code Point (DSCP) bzw. Precedence Code Point (PCP) genannt werden. Diese Code-Point-Bits werden bei der Klassifizierung durch das Netz verwendet, um aus einer Reihe von netzweiten, vordefinierten Verkehrsklassifizierungen zu wählen, wobei die Code-Points den Netzaktionen

zugeordnet werden, die ein bestimmtes Verhalten implementieren. Mit diesem Modell werden sowohl etablierte Ethernet- und TCP-TCs als auch HPC-spezifische Klassifizierungen unterstützt.

[0097] In herkömmlichen Systemen rufen Standard-DSCPs die entsprechenden Standardverhaltensweisen auf, wobei die Zuordnung im gesamten Netzwerk einheitlich erfolgt, vom Netzwerkeingang bis zum Netzwerkausgang. Diese allgemeinen Zuordnungen definieren das spezielle Verhalten, das für eine HPC-Anwendung typisch ist, nur unzureichend. Daher verwenden die vorgestellten Techniken HPC-bezogene TCs, um Verkehrsströme zu identifizieren, die mit transaktionalen Vorgängen mit geringer Latenz, wie Synchronisierungssperren oder -reduzierungen, mit Verkehr, der Teil einer Massendatenübertragung war, oder mit Verkehr, der mit einer bestimmten Anwendung oder einem bestimmten Benutzer verbunden ist, der einen bevorzugten Netzwerkzugang benötigt, zusammenhängen.

[0098] Zusätzlich zu diesen Klassifizierungen des HPC-Anwendungsverkehrs können netzwerkfähige Fabrics viele der in den Switch ASIC integrierten Funktionen zur Leistungsdifferenzierung festlegen, konfigurieren und steuern. Adaptive Routing-Bias, In-Order/Out-of-Order-Zustellung und verlustbehaftete oder verlustfreie Zustellungsmarkierung können mit bestimmten TCs verknüpft werden.

[0099] Abb. 4 zeigt ein konzeptionelles Diagramm eines Switches 402, der für die Klassifizierung von Datenverkehr in Übereinstimmung mit den beschriebenen Ausführungsformen konfiguriert ist. Im Beispiel empfängt der Switch 402 Datenverkehr 406, der zwischen einer Quelle und einem Ziel über die Switch-Fabric übertragen werden kann. Wie dargestellt, kann der Datenverkehr 406 von der Vermittlungsstelle 402 in Form von Strömen mit mehreren Datenpaketen empfangen werden. Der Switch 402 verfügt über die Funktionalität, den Datenverkehr 406 auf der Grundlage von definierten Verkehrsklassen 430 zu klassifizieren, die in die Logik des Switches 402 programmiert werden können. Wichtig ist, dass die vom Switch 406 verwendeten Verkehrsklassen 430 geeignet sind, den Verkehr insbesondere anhand von Merkmalen zu kategorisieren, die mit HPC und HPC-Anwendungen zusammenhängen. Die Verkehrsklassen 430 können bestimmten I/0-Mustern zugeordnet werden, so dass eine vorhersehbare, wiederholbare und konfigurierbare Leistung über ein konvergiertes Netzwerk möglich ist. Diese zusammengesetzten Verhaltensweisen würden effektiv Konfigurationen für bestimmte Verkehrsmuster 430 definieren. Eine Reihe von HPC-Merkmalen könnte für die Steuerung der definierten Verkehrsklassen 430 ausgewählt werden. Beispiele für HPC-Merkmale, die verwendet werden können, sind unter anderem:

Verkehrsklassenwahl (Vorrang/Priorität)

- Scavenger/ Jederzeit

Nach bestem Bemühen (Standard)

- Gebinde I/0

Barriere/ Sync

Adaptive Route/ Flussumgehung

- Maximale Latenzzeit= <us>

- Mindestbandbreite = <%> I <Gbps>

- Einspritzrate auf max<%> begrenzen

- Expedited delivery

[0100] Im Allgemeinen lassen sich die Verkehrsklassen 430 gut mit den betrieblichen Anforderungen von HPC-Anwendungen in Einklang bringen. Darüber hinaus bieten diese Klassifizierungen die Möglichkeit, eine relative Anwendungspriorität festzulegen und zwischen zeit- oder geschäftskritischen Anwendungen und anderen Aufträgen zu unterscheiden. In die vom Switch 430 verwendeten Verkehrsklassen 430 können mehrere individuelle Klassen aufgenommen werden. Im Beispiel sind die einzelnen Verkehrsklassen wie folgt dargestellt: Klasse 431 für niedrige Latenzzeiten, Klasse 432 für dedizierten Zugang, Klasse 433 für Massendaten, Klasse 434 für beste Dienste und Klasse 435 für Scavenger. Die dargestellten einzelnen Verkehrsklassen 431-435 dienen jedoch nur zu Diskussionszwecken und sind nicht als Einschränkung zu verstehen. Daher können andere Arten von HPC-bezogenen Verkehrsklassen in Übereinstimmung mit den hier offengelegten Ausführungsformen verwendet werden, wenn dies als angemessen erachtet wird.

[0101] Die Low-Latency-Klasse 431 unterstützt Datenmuster mit niedriger Latenz und geringem Jitter. Niedrige Latenzzeiten können typischerweise durch transaktionalen Datenaustausch, Barrieresynchronisationen und Sammeloperationen verursacht werden. Angegebene maximale Latenzen werden von der Klasse 431 mit niedriger Latenz garantiert. Dienstgarantien in der Klasse 431 mit niedriger Latenz erfordern in der Regel begleitende Bandbreitenbegrenzungen und Beschränkungen der Paketgröße, damit keine übermäßige Bandbreite mit hoher Priorität verbraucht wird.

[0102] Die dedizierte Zugangsklasse 432 bietet eine Kategorie, die mit der höchsten Priorität betrieben wird. So kann die dedizierte Zugangsklasse 432 beispielsweise eine hohe Bandbreitenzuweisung, eine maximale garantierte Latenzzeit und die höchste Planungs- und Arbitrierungspriorität aufweisen. Die dedizierte Zugangsklasse 432 kann absoluten Vorrang vor allen anderen Klassen haben.

[0103] Die Massendatenklasse 433 kann in erster Linie für I/0 verwendet werden und dient dazu, die dauerhafte Datenübertragung von der sonstigen Kommunikation zwischen Anwendungen und Prozessen zu trennen.

[0104] Die Best-Efforts-Klasse 434 kann als gemeinsam genutzte Standardklasse dienen. Die Best-Efforts--Klasse 434 kann den Datenverkehr für eine Reihe von Anwendungen übertragen, die gleichzeitig über dieselbe Netzinfrastruktur ausgeführt werden. Auch wenn die Best-Efforts-Klasse 434 gemeinsam genutzt wird, werden die Netzwerkkapazität und die Ressourcenzuweisung gerecht auf die Anwendungen verteilt.

[0105] Die Scavenger-Klasse 435 kann für Daten verwendet werden, die zwar erwünscht sind, für die aber keine strengen Lieferbedingungen gelten. Ein Beispiel für Datenverkehr, der in die Scavenger-Klasse 435 aufgenommen werden kann, sind Überwachungsdaten, insbesondere für die anwendungsspezifische Überwachung, z. B. mit Leistungstools. Auch die globale Überwachung von Daten, die für den Out-of-Band-Transport zu umfangreich sind, könnte auf diese Weise erfolgen. Die Verwendung der Scavenger-Klasse 435 stellt sicher, dass eine solche Kommunikation nicht die Kommunikation mit „echter“ Arbeit (z. B. mit hoher Priorität) stört.

[0106] Im Allgemeinen kann sich das PHB-Verhalten (Per-Hop Behavior) einer bestimmten Verkehrsklasse, das festlegt, wie Pakete oder Rahmen weitergeleitet werden, an jedem Netzwerkrouter ändern, indem der Code Point eines Pakets neu klassifiziert wird. Logischerweise kann die gesamte Switch-Fabric, die aus vielen miteinander verbundenen Switch-ASICs besteht, als eine einzige logische Netzwerkeinheit betrachtet werden. Als solche kann die Klassifizierung gemäß den Ausführungsformen am Eingang der Netzwerkstruktur durchgeführt werden und wird unverändert zum Ausgang der Netzwerkstruktur übertragen. Somit kann der Switch 402, der die Verkehrsklassifizierung durchführt, ein Ingress in die Switch-Fabric sein.

Fabric Specific Tag für die Verkehrsklassifizierung

[0107] Beim Eintritt in die Fabric kann der Switch 402 ein Paket oder einen Rahmen empfangen und dieses Paket analysieren. Danach wird entweder der DSCP des Headers oder der PCP der VLAN-Felder eines Ethernets als primäre Quelle verwendet, um ein fabrikationsspezifisches Tag (Ftag) zu erzeugen, das die Verkehrsklasse angibt, die dem Paket zugewiesen ist. Wenn DSCP verwendet wird, ist auch eine implizite Zuordnung von DSCP zu PCP definiert. Anschließend kann das Ftag als direkte oder indirekte Zuordnung zu Switch-Ressourcen verwendet werden, z. B. zur Zuweisung von Eingangspufferspeicherplatz und zu Warteschlangen für die Verkehrsgestaltung, in denen die anschließende Verkehrszuteilung vorgenommen wird. Andere Felder des Headers können ebenfalls verwendet werden, um den endgültigen Ftag-Wert zu beeinflussen. Das Ftag kann ein 4-Bit-Feld sein, das letztlich dazu dient, das Verhalten des Pakets beim Durchqueren der gesamten Netzstruktur zu steuern. Wie bereits erwähnt, wird der FTag-Wert berechnet, wenn ein Frame von einem Ethernet-Port in die Fabric eintritt, und bleibt dann unverändert im Fabric-Header erhalten, wenn der Frame die Network Fabric durchläuft.

[0108] In einem Beispiel können die 16 Werte von FTag auf 8 Werte von Shaping Queue (SQ) abgebildet werden, während Ftag durch die Fabric läuft. Der SQ-Wert wird über die Anforderungskreuzschiene an die Age-Warteschlangen im Ausgangsanschluss weitergeleitet. Die Shaping-Funktion steuert die in AGEQ durchgeführte Arbitrierung.

[0109] In der AGEQgibt es eine nachfolgende Abbildung von SQauf Buffer Class (BC). Die BC wird verwendet, um den verfügbaren Platz im Eingangspuffer des Linkpartners aufzuteilen. Diese zusätzliche Zuordnung

ermöglicht eine gewisse Aggregation des Eingangspuffers für Klassen mit niedriger Latenz, die nicht viel Bandbreite und damit wenig Pufferplatz benötigen.

[0110] Da es eine feste Zuordnung von FTag zu SQ und eine anschließende feste Zuordnung von SQ zu BC gibt, gibt es eine implizite feste Zuordnung direkt von FTag zu BC. Die Verwaltungssoftware muss sicherstellen, dass diese Zuordnungen im gesamten Netz für alle aktiven FTags, SQs und BCs vollständig konsistent sind. Es ist möglich, einen FTag zu inaktivieren, damit die Software diese Zuordnungen in einem stark ausgelasteten Netz neu organisieren kann, doch ist dies ein schwerwiegender Vorgang.

[0111] Dieses FTag wird dann dem Fabric-Header hinzugefügt, wenn das Paket von allen Funktionsblöcken aller Switches, die es durchläuft, verarbeitet wird, bevor es wieder auf einer anderen Ethernet-Verbindung austritt. Bei einigen Transportprotokollen kann eine umgekehrte Zuordnung von FTag zurück zu DSCP oder von FTag zurück zu PCP erforderlich sein. In diesem Fall kann die umgekehrte Zuordnung im EEG-Block des endgültigen Switches durchgeführt werden. Beim Austritt aus der Fabric kann das interne Ftag in einen DSCP oder PCP zurückübersetzt werden, wenn der Edge-Port der Netzwerk-Fabric mit einem Gerät verbunden ist, das Portale oder Ethernet unterstützt, wodurch der Eingangscodepunkt Ende-zu-Ende erhalten bleibt.

[0112] Das Weiterleitungsverhalten (oder PHB) der Netzwerkstruktur kann durch das Ftag bestimmt werden. Das Ftag kann zur Implementierung einer Reihe von QoS-Kategorien verwendet werden, die sowohl beobachtbares als auch nicht beobachtbares Netzwerkverhalten unterstützen, mit dem letztendlichen Ziel, eine vorhersehbare, skalierbare und leistungsstarke Anwendungsausführung für eine Reihe von Anwendungsworkloads und Datenverkehr zu bieten. Zu den QoS-Kategorien, die auf dem Ftag und damit auf der vom Ftag angegebenen Verkehrsklasse basieren können, gehören:

- Fabric Routing
- Verkehrstrennung
- Verkehrsgestaltung
- Staumanagement

[0113] Das Routing in der Switch-Fabric kann durch eine in Switch 202 implementierte Fabric-Routing-Funktion (FRF) gesteuert werden. Ein Beispiel für eine FRF-Komponente 500 ist in den **Abb. 5A** und **Abb. 5B** dargestellt. Es versteht sich, dass eine separate Instanz der FRF-Komponente 500 innerhalb der Eingangslogik für jeden Port des Switches 202 implementiert werden kann. Die von der FRF-Komponente 500 getroffenen Routing-Entscheidungen können auf diejenigen Frames angewendet werden, die nicht bereits Teil eines etablierten Flusses sind. Es ist zu beachten, dass die FRF-Komponente 500 nicht unbedingt weiß, ob ein bestimmter Rahmen mit einem Fluss verbunden ist oder nicht, sondern vielmehr eine unabhängige Weiterleitungsentscheidung für jeden an einem Eingangsport präsentierten Rahmen trifft. Die FRF-Komponente 500 kann Filter, Tabellen, Schaltkreise und/oder Logik, wie z. B. Auswahlschaltkreise/Logik, umfassen, um die Weiterleitung von Daten durch eine Switch-Fabric, wie hier beschrieben, zu bewirken. Wie dargestellt, umfasst die FRF-Komponente 500 mindestens: eine minimale Ports-Auswahlkomponente 502 (die eine minimale Tabellenkomponente 502A umfasst), verschiedene Ports-Filter (Filter für zugelassene Ports, Filter für betriebsbereite Ports, Filter für belegte Ports); eine Komponente zur Unterscheidung bevorzugter Ports 502B; Pseudo-Zufallsauswahlkomponenten/-logik 502C; Ausnahmetabellen 504 (einschließlich einer Ausnahmelistentabelle 504A); eine Komponente für betriebsbereite Ports 506, die eine globale Fehlertabelle 506A umfasst; und eine Routing-Algorithmus-Tabelle 508. Wie in **Abb. 5B** dargestellt, kann die FRF-Komponente 500 ferner Folgendes umfassen: eine Komponente zur Auswahl nicht minimaler Ports (510), die eine lokale Komponente zur Auswahl nicht minimaler Ports (510A) und eine globale Komponente zur Auswahl nicht minimaler Ports (510B) umfasst; und eine Ausgangslogikkomponente (512) (die Teil des Ausgangssteuerblocks eines Schalters ist), die eine adaptive Auswahlkomponente oder -logik (512A) umfasst, die wiederum eine Vorspannungskomponente (514) mit einer Vorspanntabelle (514A) enthält. Die FRF-Komponente 500 enthält weitere Komponenten, die hier beschrieben werden.

[0114] Insbesondere bestimmt die FRF-Komponente 500 mit dem Diskriminator 502B für bevorzugte Ports einen bevorzugten Port, um jeden am Eingangsport präsentierten Frame weiterzuleiten, und zwar auf der Grundlage der Ziel-Fabric-Adresse (DFA) eines empfangenen Frames, des aktuellen Routing-Zustands des Frames (wo sich der Frame auf seinem Pfad befindet und welchen Pfad bzw. welche Pfade er genommen hat, um seinen aktuellen Routing-Zustand zu erreichen), des Routing-Algorithmus und der Konfiguration der

Switch-Fabric und der mit dem Ausgangsport (dem oben erwähnten bevorzugten Port, an den der Frame weitergeleitet werden soll) verbundenen Lastmetriken unter Verwendung von Filtern für belegte Ports.

[0115] Die FRF-Komponente 500 kann eine Routing-Algorithmus-Tabelle 508 enthalten, die als softwarekonfigurierbare Tabelle ausgeführt sein kann, die auf der Grundlage des aktuellen Routing-Zustands des Rahmens gültige Auswahlmöglichkeiten bestimmt. Gültige Entscheidungen sind beispielsweise, ob ein lokaler minimaler, globaler minimaler, lokaler nicht-minimaler oder globaler nicht-minimaler Pfad für den nächsten Hop des Frames gewählt werden darf. Der Routing-Status enthält Informationen wie den VC, auf dem der Rahmen empfangen wurde, und ob er sich in der Quell-, Ziel- oder Zwischengruppe befindet. Die Routing-Algorithmus-Tabelle 508 bestimmt zusammen mit der adaptiven Auswahlfunktion oder -logik 512A (wie unten beschrieben) auch den VC, der für den nächsten Sprung des Rahmens verwendet wird.

[0116] Als Beispiel wird das Frame-Routing mit Unicast-DFAs beschrieben. Es ist jedoch zu beachten, dass die DFA der Routing-Anforderung entweder im Unicast- oder im Multicast-Format vorliegen kann. Das Unicast-Format kann ein 9-Bit-Global-ID-Feld (global_id), ein 5-Bit-Switch-ID-Feld (switch_id) und ein 6-Bit-Endpunkt-ID-Feld (endpoint_id) enthalten. Die globale ID kann eine Gruppe innerhalb des Netzes eindeutig identifizieren. Sie identifiziert insbesondere die letzte Gruppe, an die der Rahmen zugestellt werden muss. Die Switch-ID identifiziert eindeutig einen Switch innerhalb der durch die globale ID identifizierten Gruppe. Das Feld Endpunkt-ID identifiziert zusammen mit der globalen ID und der Switch-ID den Endpunkt, der mit dem Rand des Netzes verbunden ist und an den der Rahmen zugestellt werden soll. Dieses Feld wird einem Port oder einer Reihe von Ports auf dem Switch zugeordnet, der durch die globale ID und die Switch-ID identifiziert wird.

[0117] Das Multicast-Format enthält ein 13-Bit-Multicast-ID-Feld (multicast_id). Dieses Feld wird von der FRF-Komponente 500 auf eine Reihe von Ports am aktuellen Switch abgebildet, an die der Rahmen weitergeleitet werden soll.

[0118] Aus diesen Informationen ermittelt die FRF-Komponente 500 einen aktualisierten Routing-Status für den Frame, der dann in den Frame übertragen wird. Um beispielsweise das Routing in einer Libellen-Topologie zu realisieren, kann der aktuelle Status eines Rahmens aus dem VC des Rahmens (wie oben beschrieben) abgeleitet werden. Basierend auf algorithmischen Switch Fabric Routing-Regeln, die für die Switch Fabric spezifiziert sind (deren Auswahl weiter unten beschrieben wird), bestimmt die FRF-Komponente 500 einen bestimmten VC, der für den nächsten Hop des Frames verwendet wird, um Deadlocks zu vermeiden. Je nachdem, wo sich der Frame auf seinem Weg befindet, z. B. ob er sich in seiner Quellgruppe, in einer Zwischengruppe oder in seiner Zielgruppe befindet, können zusätzliche Routing-Statusinformationen bereitgestellt werden. Es sei darauf hingewiesen, dass die FRF-Komponente 500 eine Port-Filterung durchführt (die weiter unten ausführlicher beschrieben wird), indem sie Filter für zugelassene Ports, Filter für in Betrieb befindliche Ports, Filter für belegte Ports usw. verwendet, um festzustellen, ob ein bevorzugter Port, an den ein Frame weitergeleitet werden soll, derzeit fehlerhaft, besetzt, nicht vorhanden usw. ist.

[0119] Die FRF-Komponente 500 erhält Lastmessungen für ihren Ausgangsanschluss vom Switch 202. Lastinformationen für den Eingangsanschluss der FRF-Komponente 500 werden auch von einem benachbarten Switch empfangen. In einigen Ausführungsformen tauscht die FRF-Komponente 500 ihre Eingangs- und Ausgangsanschluss-Lastinformationen mit allen anderen FRF-Komponenteninstanzen innerhalb eines Switches, z. B. Switch 202, und mit benachbarten Switches in der Switch-Fabric aus. Auf diese Weise kennt jede FRF-Komponenteninstanz jedes Switches in der Switch-Fabric die zusammengefassten Lastinformationen für alle benachbarten Switches.

[0120] Es ist zu beachten, dass die FRF-Komponente 500 das Multicasting von Rahmen unterstützen kann. Wenn ein Multicast-DFA empfangen wird, bestimmt die FRF-Komponente 500 eine Reihe von Anschlüssen, an die der mit dem Multicast-DFA verbundene Rahmen weitergeleitet werden soll. Der Satz von Ports kann durch Zugriff auf eine Nachschlagetabelle bestimmt werden, die softwarekonfigurierte Multicast-Fabric-Adressen auf Ausgangsports abbildet. Dadurch werden Probleme im Zusammenhang mit doppelten Multicast--Rahmenkopien vermieden.

Routing mit Verkehrsklassen

[0121] In **Abb. 6** ist ein Flussdiagramm eines Beispielprozesses 600 zur Implementierung von Routing nach Verkehrsklassen dargestellt. Der Prozess 600 kann von einem Netzwerk-Switch implementiert werden (wie in **Abb. 1** dargestellt). Daher ist der Prozess als eine Reihe von ausführbaren Operationen dargestellt, die in

einem maschinenlesbaren Speichermedium 604 gespeichert sind und von Hardware-Prozessoren 602 in einer Rechnerkomponente 605 ausgeführt werden. Die Hardware-Prozessoren 602 führen den Prozess 600 aus und implementieren damit die hierin offenbarten Techniken.

[0122] Wie bereits beschrieben, bietet ein Switch systemweite Verkehrsklassen und damit die Möglichkeit zur Steuerung pro Verkehrsklasse. Die Kontrollen pro Verkehrsklasse können bestimmen, wie der Verkehr im Netzwerk verwaltet wird. So kann beispielsweise die Weiterleitung eines Pakets und die zugewiesene Bandbreite direkt von der zugewiesenen Verkehrsklasse abhängen. Ein Beispiel dafür ist das Traffic Shaping, das auf Verkehrsklassen basiert. Bei Konflikten um die Netzwerkbandbreite wählen die Arbiters die weiterzuleitenden Pakete auf der Grundlage ihrer Verkehrsklasse und der für diese Klasse verfügbaren Guthaben aus. Außerdem unterstützt das Netz Mindest- und Höchstbandbreiten für jede Verkehrsklasse. Die Möglichkeit der Bandbreitenverwaltung bietet die Möglichkeit, einer Anwendung Netzwerkressourcen sowie CPUs und Speicherbandbreite zuzuweisen.

[0123] Insbesondere umfasst die Steuerung pro Verkehrsklasse eine Routing-Politik, die die Verkehrsklasse eines Pakets bei Routing-Entscheidungen berücksichtigt.

[0124] Der Prozess 600 beginnt mit dem Vorgang 606, bei dem eine Vielzahl von Paketen von einem Eingangsport der Switch-Fabric empfangen werden kann. Beispielsweise kann der Datenverkehr in ein Netzwerk gelangen, wenn verschiedene HPC-Anwendungen über die Switch Fabric mit Zielen kommunizieren. Der Datenverkehr kann an einem Eingangsport in die Switch Fabric eintreten, zum Beispiel als eine Vielzahl von Paketen, die von einem Eingangs-Edge-Switch empfangen werden. Unterschiedliche Anwendungen können unterschiedliche Datentypen erzeugen, wobei die Daten unterschiedliche Merkmale aufweisen. Daher können verschiedene Ströme aus der Vielzahl der in Operation 606 empfangenen Pakete (entsprechend den verschiedenen Datentypen) Merkmale aufweisen, die zu mehreren individuellen Verkehrsklassen gehören. Mit anderen Worten: Der Datenverkehr kann Daten mit unterschiedlichen Verkehrsklassen umfassen.

[0125] Als nächstes wird in Vorgang 608 eine Verkehrsklasse für ein Paket (aus der Vielzahl der Pakete) bestimmt. Operation 608 kann beinhalten, dass der Switch das Paket analysiert und ein Feld mit dem PCP (und/oder DSCP) im Paketkopf analysiert. Der PCP kann ein Hinweis auf die Anwendung sein, die das Paket erzeugt hat, oder eine Charakteristik anzeigen, die mit dem Netzwerkverhalten des Pakets zusammenhängt. In einigen Ausführungsformen werden die Werte (z. B. Bits) des PCP bestimmten Verkehrsklassen zugeordnet, die vom Switch definiert und bekannt sind. So kann der Switch auf der Grundlage der PCP im Header des Pakets anhand dieser Zuordnung die entsprechende Verkehrsklasse für das Paket bestimmen. Die Verkehrsklassen basieren auf HPC-bezogenen Merkmalen und können folgende Klassen umfassen: Klasse für niedrige Latenzzeiten, dedizierte Klasse, Klasse für Massendaten, Best-Effort-Klasse und Scavenger-Klasse. Jede der Verkehrsklassen wird in **Abb. 4** ausführlicher beschrieben und wird aus Gründen der Kürze nicht noch einmal beschrieben. Dementsprechend wird das Paket in Operation 608 einer der oben genannten Verkehrsklassen zugewiesen. Obwohl in Prozess 600 beschrieben wird, dass Pakete klassifiziert werden, ist zu beachten, dass die Verkehrsklassifizierung nicht auf eine paketweise Basis beschränkt ist. Alternativ kann die Verkehrsklassifizierung beispielsweise pro Fluss durchgeführt werden. Darüber hinaus können herkömmliche Verkehrsklassen, die nicht spezifisch für HPC-Merkmale sind, wie Durchsatz, Bandbreitenzuweisung (z. B. datenflussspezifische Merkmale), bei der Zuweisung einer Verkehrsklasse zu einem Paket in Vorgang 608 verwendet werden.

[0126] Wie bereits beschrieben, kann der Eingangs-Switch eine Klassifizierung vornehmen und der Vielzahl von Paketen beim Eintritt in die Switch-Fabric Verkehrsklassen zuweisen (die bis zum Austrittspunkt beibehalten werden). Somit muss der Vorgang 608 nicht an jedem Switch durchgeführt werden, den das Paket in der Fabric erreicht. Nachdem eine Verkehrsklasse für das Paket bestimmt wurde, kann der Prozess 600 mit Operation 610 fortfahren

[0127] Anschließend kann in Operation 610 eine fabrikspezifische Kennzeichnung für das Paket auf der Grundlage der ermittelten Verkehrsklasse erzeugt werden. Die fabrikspezifische Kennzeichnung zeigt die besondere Verkehrsklasse an, die dem Paket zugewiesen wurde. Das fabrikspezifische Tag kann ein Ftag sein, wie oben im Detail beschrieben. In einigen Fällen beinhaltet die Erzeugung des Ftag eine Übersetzung des PCP in den Ftag-Wert. Somit dient das Fabric-spezifische Tag als Markierung im Paket, die innerhalb der Fabric erkennbar ist und zur weiteren Klassifizierung des Pakets in seine entsprechend zugewiesene Verkehrsklasse verwendet wird. Mit anderen Worten: Innerhalb der Netzstruktur wird die Verkehrsklasse des Pakets durch einen FTAG-Wert identifiziert.

[0128] Danach kann in Operation 612 eine Prüfung durchgeführt werden, um festzustellen, ob irgendwelche Routing-Direktiven für das Paket von der Verkehrsklassifizierung in Bezug auf das fabrikationsspezifische Tag abhängig sind. Wie in den **Abb. 5A-5B** ausführlich beschrieben, wird das Routing durch die FRF eines Switches gesteuert. Daher können die FRF und ihre Strukturen (z. B. Routing-Tabellen) bestimmen, ob Routing-Richtlinien, -Entscheidungen oder -Anweisungen von der fabrikspezifischen Kennzeichnung oder der Verkehrsklasse eines Pakets abhängig sind. Wenn das Routing auf der Verkehrsklasse basiert, geht der Prozess zu Operation 614 über und das Paket wird auf der Grundlage seiner Verkehrsklasse geroutet.

[0129] In Operation 614 kann die FRF eine bestimmte Routing-Richtlinie bestimmen, die der jeweiligen Verkehrsklasse des Pakets (wie durch dessen Ftag angegeben) entspricht. Beispielsweise kann die FTAG als Bias-Wert beim adaptiven Routing verwendet werden. Beim adaptiven Routing kann der Bias-Wert in Abhängigkeit von der Art des Pfades, auf den er angewendet wird (nicht bevorzugt minimal, bevorzugt minimal und nicht minimal), der Verkehrsklasse des Pakets, das geroutet wird, und der Position des Pakets auf seinem Pfad variieren. So können beispielsweise Pakete in einer Verkehrsklasse mit niedriger Latenzzeit stärker auf minimale Pfade ausgerichtet sein als Pakete in anderen Verkehrsklassen.

[0130] Sobald ein Paket in der vorherigen Operation 610 markiert wurde, gibt es eine Reihe von Routing-Steuerungen pro Verkehrsklasse, die im Switch implementiert werden können. Die Verkehrsklasse eines Pakets kann in Vorgang 614 zur Durchführung verwendet werden:

- Routing Bias-Einstellungen. Der FRF unterstützt eine Reihe von verschiedenen Routing Bias, die ausgewählt werden können, um eine optimale Leistung für ein bestimmtes Verkehrsmuster zu erzielen.
- Flussbestellung.
  - Vollständig bestellt für Protokolle, die eine Bestellung erfordern
  - Völlig ungeordnet, was einen nahezu perfekten Lastausgleich für Verkehrsmuster ermöglicht, die niemals eine Überlastung verursachen sollten.
  - Teilweise geordnete Zustellung, bei der die Reihenfolge nicht wichtig ist, aber das Verkehrsmuster eine Überlastung verursachen könnte. In diesem Fall kann der Datenverkehr bis zu dem Punkt, an dem der Zielort eine Überlastung meldet, einen beliebigen Weg zum Ziel nehmen. Er wird dann gezwungen, in der richtigen Reihenfolge zugestellt zu werden. Dies verhindert die seitliche Ausbreitung der Rahmen und die unkontrollierte Nutzung des Eingangspuffers.
- Umleitungskontrolle. Ein langer geordneter Paketfluss kann langsam vorankommen, wenn sich auf seiner Route eine Überlastung abzeichnet. Durch die Umleitung wird der Fluss kurzzeitig angehalten, so dass er einen neuen Weg durch das Netz nehmen kann und die Möglichkeit hat, einen weniger überlasteten Weg zum Ziel zu finden.
- Quiesce-Kontrollen. Dies kann von einem Verwaltungsagenten verwendet werden, um zu garantieren, dass ein neuer Satz von Routenwerten für einen oder mehrere FTag-Werte verwendet wird.
- Anfrage/Antwort-Konfiguration. Einige HPC-Protokolle beruhen auf einem Anfrage- und Antwortprotokoll. In einigen Fällen kann eine Abhängigkeit innerhalb der Netzwerkschnittstellenkarte (NIC) des Servers bestehen, bei der die Annahme eines Anforderungspakets vom Netz von der Fähigkeit der NIC abhängt, ein Antwortpaket zurück in das Netz zu injizieren. Wenn die Pufferressourcen des Netzes vollständig aufgebraucht sind, kann die Fähigkeit des Netzes, ein neues Paket von einer Netzwerkkarte anzunehmen, davon abhängen, ob das Netz in der Lage ist, Pakete in eine andere Netzwerkkarte zurückzuspeisen. Dies ist die klassische Deadlock-Schwachstelle. Die Dienstklassen können als Anfrage oder Antwort zugewiesen werden, und dies bietet Garantien, die die Deadlock-Anfälligkeit aufheben, da die Annahme eines Antwortpakets im Netz niemals von der Lieferung eines Anfragepakets aus dem Netz abhängt.

[0131] In einigen Ausführungsformen kann die Operation 614 zusätzliche Operationen auf der Grundlage der Verkehrsklasse zusätzlich zur (oder anstelle der) Leitweglenkung durchführen, z. B.: Trennung des Datenflusses, geordnete oder ungeordnete Datenlieferung, verlustbehaftete oder verlustfreie Übertragung, Erfassung von Telemetriedaten und Regeln zur Verkehrsgestaltung.

[0132] Zurück zu Vorgang 612: Wenn festgestellt wird, dass es keine Routing-Richtlinien gibt, die pro Verkehrsklasse definiert sind, kann der Vorgang mit Vorgang 616 fortgesetzt werden und die Pakete auf der Grundlage anderer vom Switch verwendeter Routing-Richtlinien geleitet werden. Beispielsweise kann das

Paket auf der Grundlage adaptiver Routing-Techniken geroutet werden, die nicht von der Verkehrsklasse des Pakets abhängig sind.

[0133] Abb. 7 zeigt ein Blockdiagramm eines beispielhaften Computersystems 700, in dem verschiedene der hier beschriebenen Ausführungsformen implementiert werden können. Das Computersystem 700 umfasst einen Bus 702 oder einen anderen Kommunikationsmechanismus zur Übermittlung von Informationen sowie einen oder mehrere Hardware-Prozessoren 704, die zur Verarbeitung von Informationen mit dem Bus 702 verbunden sind. Bei dem/den Hardware-Prozessoren) 704 kann es sich zum Beispiel um einen oder mehrere Allzweck-Mikroprozessoren handeln.

[0134] Das Computersystem 700 umfasst auch einen Hauptspeicher 706, z. B. einen Speicher mit wahlfreiem Zugriff (RAM), einen Cache und/oder andere dynamische Speichergeräte, die mit dem Bus 702 verbunden sind, um Informationen und Anweisungen zu speichern, die vom Prozessor 704 ausgeführt werden sollen. Der Hauptspeicher 706 kann auch zum Speichern von temporären Variablen oder anderen Zwischeninformationen während der Ausführung von Anweisungen verwendet werden, die vom Prozessor 704 ausgeführt werden sollen. Solche Befehle, die in Speichermedien gespeichert sind, auf die der Prozessor 704 zugreifen kann, machen das Computersystem 700 zu einer Spezialmaschine, die so angepasst ist, dass sie die in den Befehlen angegebenen Operationen ausführen kann.

[0135] Das Computersystem 700 umfasst außerdem einen Festwertspeicher (ROM) 708 oder ein anderes statisches Speichergerät, das mit dem Bus 702 verbunden ist, um statische Informationen und Anweisungen für den Prozessor 704 zu speichern. Ein Speichergerät 710, wie z. B. eine Magnetplatte, eine optische Platte oder ein USB-Stick (Flash-Laufwerk) usw., ist vorgesehen und mit dem Bus 702 verbunden, um Informationen und Anweisungen zu speichern.

[0136] Das Computersystem 700 kann über den Bus 702 mit einer Anzeige 712, z. B. einer Flüssigkristallanzeige (LCD) (oder einem Berührungsbildschirm), verbunden sein, um einem Computerbenutzer Informationen anzuzeigen. Ein Eingabegerät 714, einschließlich alphanumerischer und anderer Tasten, ist mit dem Bus 702 gekoppelt, um Informationen und Befehlsauswahlen an den Prozessor 704 zu übermitteln. Eine andere Art von Benutzereingabegerät ist die Cursorsteuerung 716, wie z. B. eine Maus, ein Trackball oder Cursor-Richtungstasten zur Übermittlung von Richtungsinformationen und Befehlsauswahlen an den Prozessor 804 und zur Steuerung der Cursorbewegung auf dem Display 712. In einigen Ausführungsformen können die gleichen Richtungsinformationen und Befehlsauswahlen wie bei der Cursorsteuerung über den Empfang von Berührungen auf einem Touchscreen ohne Cursor implementiert werden.

[0137] Das Computersystem 700 kann ein Benutzerschnittstellenmodul zur Implementierung einer grafischen Benutzeroberfläche enthalten, das in einem Massenspeichergerät als ausführbare Softwarecodes gespeichert werden kann, die von dem/den Computergerät(en) ausgeführt werden. Dieses und andere Module können beispielsweise Komponenten wie Softwarekomponenten, objektorientierte Softwarekomponenten, Klassenkomponenten und Aufgabenkomponenten, Prozesse, Funktionen, Attribute, Prozeduren, Unterprogramme, Segmente von Programmcode, Treiber, Firmware, Mikrocode, Schaltkreise, Daten, Datenbanken, Datenstrukturen, Tabellen, Arrays und Variablen umfassen.

[0138] Im Allgemeinen kann sich das Wort „Komponente“, „Engine“, „System“, „Datenbank“, „Datenspeicher“ und dergleichen, wie es hier verwendet wird, auf eine in Hardware oder Firmware verkörperte Logik oder auf eine Sammlung von Softwareanweisungen beziehen, die möglicherweise Ein- und Ausstiegspunkte haben und in einer Programmiersprache wie z. B. Java, C oder C++ geschrieben sind. Eine Softwarekomponente kann kompiliert und zu einem ausführbaren Programm verknüpft werden, in einer dynamischen Link-Bibliothek installiert werden oder in einer interpretierten Programmiersprache wie BASIC, Perl oder Python geschrieben sein. Es ist klar, dass Softwarekomponenten von anderen Komponenten oder von sich selbst aus aufgerufen werden können und/oder als Reaktion auf erkannte Ereignisse oder Unterbrechungen aufgerufen werden können. Softwarekomponenten, die für die Ausführung auf Computergeräten konfiguriert sind, können auf einem computerlesbaren Medium, wie z. B. einer Compact Disc, einer digitalen Videodisc, einem Flash-Laufwerk, einer Magnetplatte oder einem anderen greifbaren Medium, oder als digitaler Download bereitgestellt werden (und können ursprünglich in einem komprimierten oder installierbaren Format gespeichert sein, das vor der Ausführung eine Installation, Dekomprimierung oder Entschlüsselung erfordert). Ein solcher Softwarecode kann teilweise oder vollständig in einem Speicher des ausführenden Computergeräts zur Ausführung durch das Computergerät gespeichert werden. Softwareanweisungen können in Firmware, wie z. B. einem EPROM, eingebettet sein. Darüber hinaus können die Hardwarekomponenten aus verbunde-

nen Logikeinheiten wie Gattern und Flipflops und/oder aus programmierbaren Einheiten wie programmierbaren Gatteranordnungen oder Prozessoren bestehen.

[0139] Das Computersystem 700 kann die hierin beschriebenen Techniken unter Verwendung von kundenspezifischer festverdrahteter Logik, einem oder mehreren ASICs oder FPGAs, Firmware und/oder Programmlogik implementieren, die in Kombination mit dem Computersystem bewirkt oder programmiert, dass das Computersystem 700 eine Spezialmaschine ist. Gemäß einer Ausführungsform werden die hierin beschriebenen Techniken vom Computersystem 700 als Reaktion auf den/die Prozessor(en) 704 ausgeführt, der/die eine oder mehrere Sequenzen von einem oder mehreren im Hauptspeicher 706 enthaltenen Befehlen ausführt/ausführen. Solche Anweisungen können in den Hauptspeicher 706 von einem anderen Speichermedium, wie z. B. dem Speichergerät 710, eingelesen werden. Die Ausführung der im Hauptspeicher 706 enthaltenen Befehlssequenzen veranlasst den/die Prozessor(en) 704, die hier beschriebenen Prozessschritte durchzuführen. In alternativen Ausführungsformen können festverdrahtete Schaltungen anstelle von oder in Kombination mit Softwareanweisungen verwendet werden.

[0140] Der Begriff „nichtflüchtige Medien“ und ähnliche Begriffe, wie sie hier verwendet werden, beziehen sich auf alle Medien, die Daten und/oder Befehle speichern, die eine Maschine in einer bestimmten Weise arbeiten lassen. Solche nichtflüchtigen Medien können nichtflüchtige Medien und/oder flüchtige Medien umfassen. Zu den nichtflüchtigen Medien gehören beispielsweise optische oder magnetische Festplatten, wie die Speichervorrichtung 710. Zu den flüchtigen Medien gehören dynamische Speicher, wie der Hauptspeicher 706. Zu den gängigen Formen nichtflüchtiger Medien gehören beispielsweise Disketten, flexible Platten, Festplatten, Solid-State-Laufwerke, Magnetbänder oder andere magnetische Datenspeichermedien, CD-ROMs, andere optische Datenspeichermedien, physische Medien mit Lochmustern, RAM, PROM und EPROM, FLASH-EPROM, NVRAM, andere Speicherchips oder -kassetten sowie deren vernetzte Versionen.

[0141] Nicht-transitorische Medien unterscheiden sich von Übertragungsmedien, können aber in Verbindung mit ihnen verwendet werden. Übertragungsmedien sind an der Übertragung von Informationen zwischen nicht-transitorischen Medien beteiligt. Zu den Übertragungsmedien gehören z. B. Koaxialkabel, Kupfer- und Glasfaserkabel, einschließlich der Drähte, aus denen der Bus 702 besteht. Übertragungsmedien können auch in Form von Schall- oder Lichtwellen auftreten, wie sie bei der Datenkommunikation über Funk und Infrarot erzeugt werden.

[0142] Das Computersystem 700 umfasst auch eine Kommunikationsschnittstelle 718, die mit dem Bus 702 verbunden ist. Die Netzwerkschnittstelle 718 stellt eine Zwei-Wege-Datenkommunikationsverbindung zu einer oder mehreren Netzwerkverbindungen her, die mit einem oder mehreren lokalen Netzwerken verbunden sind. Bei der Kommunikationsschnittstelle 718 kann es sich beispielsweise um eine ISDN-Karte (Integrated Services Digital Network), ein Kabelmodem, ein Satellitenmodem oder ein Modem handeln, um eine Datenkommunikationsverbindung zu einer entsprechenden Art von Telefonleitung herzustellen. Ein weiteres Beispiel: Die Netzwerkschnittstelle 718 kann eine LAN-Karte (Local Area Network) sein, um eine Datenkommunikationsverbindung zu einem kompatiblen LAN (oder einer WAN-Komponente zur Kommunikation mit einem WAN) herzustellen. Es können auch drahtlose Verbindungen implementiert werden. In jeder dieser Implementierungen sendet und empfängt die Netzwerkschnittstelle 718 elektrische, elektromagnetische oder optische Signale, die digitale Datenströme mit verschiedenen Informationstypen übertragen.

[0143] Eine Netzverbindung ermöglicht in der Regel die Datenkommunikation über ein oder mehrere Netze zu anderen Datengeräten. So kann eine Netzverbindung beispielsweise eine Verbindung über ein lokales Netz zu einem Host-Computer oder zu Datengeräten eines Internetdienstanbieters (ISP) herstellen. Der ISP wiederum bietet Datenkommunikationsdienste über das weltweite Paketdatenkommunikationsnetz an, das heute gemeinhin als „Internet“ bezeichnet wird. Sowohl das lokale Netz als auch das Internet verwenden elektrische, elektromagnetische oder optische Signale, die digitale Datenströme übertragen. Die Signale über die verschiedenen Netze und die Signale auf der Netzverbindung und über die Kommunikationsschnittstelle 718, die die digitalen Daten zum und vom Computersystem 700 übertragen, sind Beispiele für Übertragungsmedien.

[0144] Das Computersystem 700 kann über das/die Netzwerk(e), die Netzwerkverbindung und die Kommunikationsschnittstelle 718 Nachrichten senden und Daten, einschließlich Programmcode, empfangen. In dem Internet-Beispiel könnte ein Server einen angeforderten Code für ein Anwendungsprogramm über das Internet, den ISP, das lokale Netzwerk und die Kommunikationsschnittstelle 718 übertragen.

[0145] Der empfangene Code kann vom Prozessor 704 bei seinem Empfang ausgeführt und/oder in der Speichervorrichtung 710 oder einem anderen nichtflüchtigen Speicher zur späteren Ausführung gespeichert werden.

[0146] Jeder der in den vorstehenden Abschnitten beschriebenen Prozesse, Methoden und Algorithmen kann in Code-Komponenten verkörpert und vollständig oder teilweise durch diese automatisiert werden, die von einem oder mehreren Computersystemen oder Computerprozessoren mit Computerhardware ausgeführt werden. Das eine oder die mehreren Computersysteme oder Computerprozessoren können auch so betrieben werden, dass sie die Ausführung der entsprechenden Vorgänge in einer „Cloud Computing"-Umgebung oder als „Software as a Service" (SaaS) unterstützen. Die Prozesse und Algorithmen können teilweise oder vollständig in anwendungsspezifischen Schaltkreisen implementiert sein. Die verschiedenen oben beschriebenen Merkmale und Verfahren können unabhängig voneinander verwendet oder auf verschiedene Weise kombiniert werden. Verschiedene Kombinationen und Unterkombinationen sollen in den Anwendungsbereich dieser Offenlegung fallen, und bestimmte Verfahrens- oder Prozessblöcke können in einigen Implementierungen weggelassen werden. Die hier beschriebenen Methoden und Prozesse sind auch nicht auf eine bestimmte Reihenfolge beschränkt, und die damit verbundenen Blöcke oder Zustände können in anderen geeigneten Reihenfolgen, parallel oder auf andere Weise ausgeführt werden. Blöcke oder Zustände können zu den offengelegten Beispielen hinzugefügt oder aus ihnen entfernt werden. Die Ausführung bestimmter Operationen oder Prozesse kann auf Computersysteme oder Computerprozessoren verteilt werden, die sich nicht nur auf einer einzigen Maschine befinden, sondern über eine Reihe von Maschinen verteilt sind.

[0147] Wie hierin verwendet, kann eine Schaltung in jeder Form von Hardware, Software oder einer Kombination davon implementiert werden. Beispielsweise können ein oder mehrere Prozessoren, Controller, ASICs, PLAs, PALs, CPLDs, FPGAs, logische Komponenten, Software-Routinen oder andere Mechanismen implementiert werden, um eine Schaltung zu bilden. Bei der Implementierung können die verschiedenen hier beschriebenen Schaltungen als diskrete Schaltungen implementiert werden, oder die beschriebenen Funktionen und Merkmale können teilweise oder insgesamt auf eine oder mehrere Schaltungen aufgeteilt werden. Auch wenn verschiedene Merkmale oder Funktionselemente einzeln als separate Schaltungen beschrieben oder beansprucht werden, können diese Merkmale und Funktionen von einer oder mehreren gemeinsamen Schaltungen gemeinsam genutzt werden, und eine solche Beschreibung soll nicht voraussetzen oder implizieren, dass separate Schaltungen erforderlich sind, um diese Merkmale oder Funktionen zu implementieren. Wenn eine Schaltung ganz oder teilweise mit Software implementiert ist, kann diese Software so implementiert werden, dass sie mit einem Computer- oder Verarbeitungssystem arbeitet, das in der Lage ist, die beschriebene Funktionalität auszuführen, wie z. B. das Computersystem 700.

[0148] Wie hierin verwendet, kann der Begriff „oder" sowohl im einschließenden als auch im ausschließenden Sinne verstanden werden. Darüber hinaus ist die Beschreibung von Ressourcen, Vorgängen oder Strukturen im Singular nicht so zu verstehen, dass der Plural ausgeschlossen wird. Bedingte Ausdrücke, wie z. B. „kann", „könnte", „könnte" oder „darf", sollen im Allgemeinen vermitteln, dass bestimmte Ausführungsformen bestimmte Merkmale, Elemente und/oder Schritte einschließen, während andere Ausführungsformen diese nicht einschließen, es sei denn, es ist ausdrücklich etwas anderes angegeben oder im Zusammenhang mit der Verwendung anders zu verstehen.

[0149] Die in diesem Dokument verwendeten Begriffe und Ausdrücke sowie deren Abwandlungen sind, sofern nicht ausdrücklich etwas anderes angegeben ist, nicht als einschränkend, sondern als offen zu verstehen. Adjektive wie „konventionell", „traditionell", „normal", „Standard", „bekannt" und Begriffe mit ähnlicher Bedeutung sind nicht so zu verstehen, dass sie den beschriebenen Gegenstand auf einen bestimmten Zeitraum oder auf einen zu einem bestimmten Zeitpunkt verfügbaren Gegenstand beschränken, sondern sollten so verstanden werden, dass sie konventionelle, traditionelle, normale oder Standardtechnologien umfassen, die jetzt oder zu einem beliebigen Zeitpunkt in der Zukunft verfügbar oder bekannt sein können. Das Vorhandensein erweiternder Wörter und Ausdrücke wie „eine oder mehrere", „mindestens", „aber nicht beschränkt auf" oder ähnliche Ausdrücke in einigen Fällen ist nicht so zu verstehen, dass der engere Fall beabsichtigt oder erforderlich ist, wenn solche erweiternden Ausdrücke nicht vorhanden sind.

ZITATE ENTHALTEN IN DER BESCHREIBUNG

Zitierte Patentliteratur

- US 62/852273 [0002]
- US 62/852203 [0002]
- US 62/852289 [0002]

Patentansprüche

1. Ein Verfahren zur Klassifizierung von Verkehrsdaten, das Folgendes umfasst:
Empfang einer Vielzahl von Paketen an einem Eingangsport einer Switch-Fabric;
Bestimmen einer Verkehrsklassifizierung für mindestens ein Paket der Vielzahl von Paketen, wobei die bestimmte Verkehrsklassifizierung aus einer Gruppe von definierten Verkehrsklassen ausgewählt wird, die sich auf anwendungsdatenspezifische Merkmale für das Netzwerk beziehen; und
Erzeugen eines fabrikspezifischen Flags für das mindestens eine Paket, das die ermittelte Verkehrsklassifizierung angibt.

2. Das Verfahren nach Anspruch 1, wobei die Bestimmung der Verkehrsklassifizierung Folgendes umfasst:
Analysieren des mindestens einen Pakets;
Analysieren eines Precedence Code Point (PCP) oder Differentiated Service Code Point (DSCP) in einem Header des Pakets, wobei der PCP eine dem Paket zugeordnete Anwendung oder eine dem Paket zugeordnete anwendungsdatenspezifische Eigenschaft anzeigt; und
die Zuordnung der analysierten PCP zu einer Verkehrsklasse aus der Gruppe der definierten Verkehrsklassen, um die Verkehrsklassifizierung zu bestimmen.

3. Das Verfahren nach Anspruch 2, wobei es sich bei der Anwendung um eine High Performance Computing (HPC)-Anwendung handelt und die definierten Verkehrsklassen sich auf HPC-spezifische Merkmale beziehen.

4. Das Verfahren nach Anspruch 2, wobei der Eingangsport einer Switch-Fabric mit einem Eingangs-Edge-Switch verbunden ist.

5. Das Verfahren nach Anspruch 4, wobei das fabric-spezifische Tag die Verkehrsklassifizierung des mindestens einen Pakets für andere Switches innerhalb der Switch-Fabric identifiziert.

6. Das Verfahren nach Anspruch 5, wobei das fabrikationsspezifische Tag ein FTAG-Wert ist, der innerhalb der Switch-Fabrik verwendet werden kann, um ein Routing pro Verkehrsklasse für das mindestens eine Paket auf der Grundlage der bestimmten Verkehrsklassifizierung durchzuführen.

7. Das Verfahren nach Anspruch 6, wobei das fabric-spezifische Tag ein FTAG-Wert ist, der innerhalb der Switch-Fabric verwendet werden kann, um eine Aktion für das mindestens eine Paket auf der Grundlage der bestimmten Verkehrsklassifizierung durchzuführen, wobei die Aktion mindestens eines der folgenden Elemente umfasst: Verkehrsformung, Datenstromtrennung, geordnete oder ungeordnete Datenlieferung, verlustbehaftete oder verlustfreie Übertragung und Überlastungsmanagement.

8. Das Verfahren nach Anspruch 7, wobei die Gruppe der definierten Verkehrsklassen Folgendes umfasst: Klasse mit niedriger Latenz, Klasse mit dediziertem Zugang, Klasse mit Massendaten, Klasse mit besten Bemühungen und Scavenger-Klasse.

9. Das Verfahren nach Anspruch 8, wobei sich zusätzliche definierte Verkehrsklassen auf datenübertragungsspezifische Merkmale wie Durchsatz, Bandbreite und Latenz beziehen.

10. Das Verfahren nach Anspruch 9, wobei sich die definierten Verkehrsklassen auf Routing, Ordnung, Umleitung, Ruhezustand, HPC-Protokollkonfiguration und Telemetrie beziehen.

11. Ein Schalter, bestehend aus:
eine anwendungsspezifische integrierte Schaltung (ASIC) zu:
eine Vielzahl von Paketen an einem Eingangsport des Switches empfangen;
Bestimmen einer Verkehrsklassifizierung für jedes Paket der Vielzahl von Paketen, wobei die bestimmte Verkehrsklassifizierung aus einer Gruppe von definierten Verkehrsklassen ausgewählt wird, die sich auf anwendungsdatenspezifische Merkmale für ein Netzwerk beziehen;
für jedes Paket aus der Vielzahl von Paketen ein gewebespezifisches Flag für das eine Paket erzeugen, das die ermittelte Verkehrsklassifizierung angibt;
festzustellen, ob eine Routing-Richtlinie von der Verkehrsklassifizierung innerhalb der Switch-Fabric abhängig ist; und
als Reaktion auf die Feststellung, dass eine Routing-Richtlinie von der Verkehrsklassifizierung abhängt, ein

klassenbasiertes Routing für die Mehrzahl der Pakete unter Verwendung der jeweiligen strukturspezifischen Klasse durchführen.

12. Der Schalter nach Anspruch 11, mit dem ASIC zum weiteren:
für jedes der Vielzahl von Paketen das Paket analysieren;
für jedes der Vielzahl von Paketen, Analysieren eines Precedence Code Point (PCP) in einem Header des Pakets, wobei der PCP eine dem Paket zugeordnete Anwendung oder ein dem Paket zugeordnetes anwendungsspezifisches Merkmal anzeigt; und
für jedes der Vielzahl von Paketen die analysierte PCP einer Verkehrsklasse aus der Gruppe der definierten Verkehrsklassen zuordnen, um die Verkehrsklassifizierung zu bestimmen.

13. Der Schalter nach Anspruch 12, wobei es sich bei der Anwendung um eine HPC-Anwendung (High Performance Computing) handelt und die definierten Verkehrsklassen sich auf HPC-spezifische Merkmale beziehen.

14. Der Schalter nach Anspruch 13, wobei die Gruppe der definierten Verkehrsklassen Folgendes umfasst: Klasse mit niedriger Latenz, Klasse mit dediziertem Zugang, Klasse mit Massendaten, Klasse mit besten Bemühungen und Scavenger-Klasse.

15. Der Schalter nach Anspruch 12, mit dem ASIC zum weiteren:
Durchführen einer Aktion für die Vielzahl von Paketen auf der Grundlage der Verkehrsklassifizierung, wobei die Aktion mindestens eines der folgenden Elemente umfasst: Verkehrsformung, Datenstromtrennung, geordnete oder ungeordnete Datenlieferung, verlustbehaftete oder verlustfreie Übertragung und Staumanagement.

Es folgen 12 Seiten Zeichnungen

Anhängende Zeichnungen

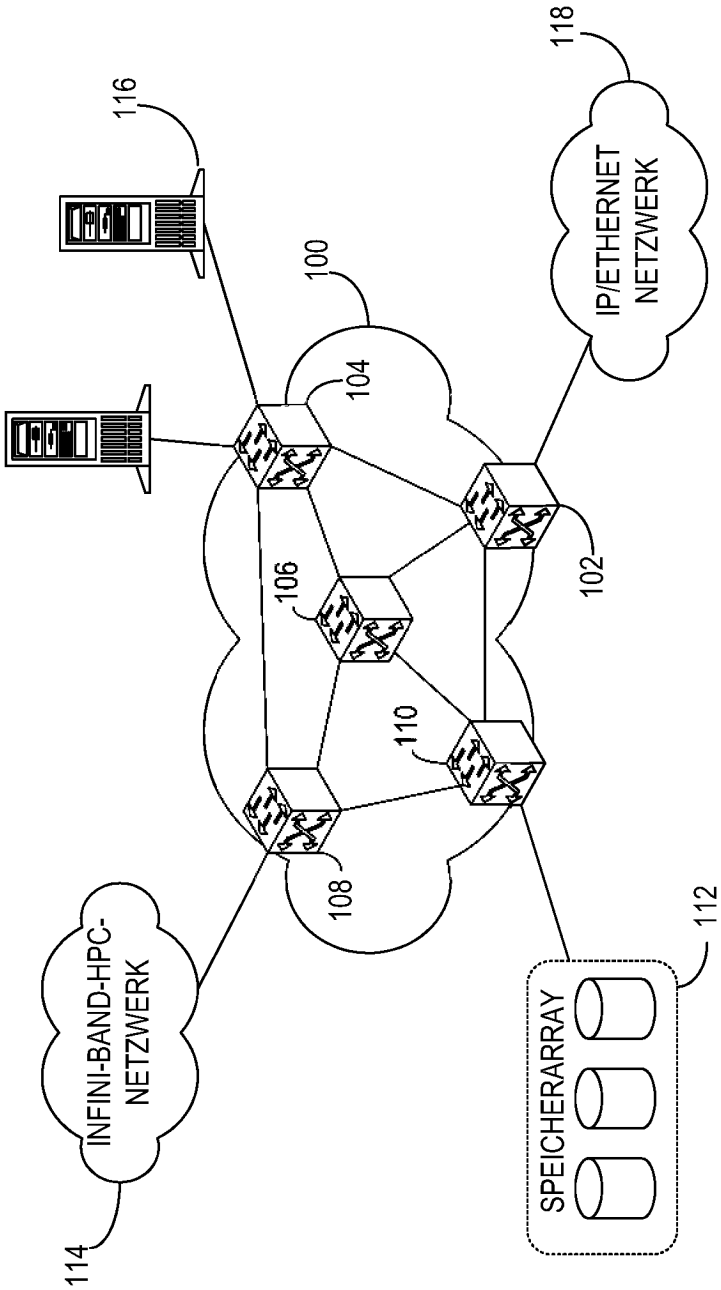


ABB. 1

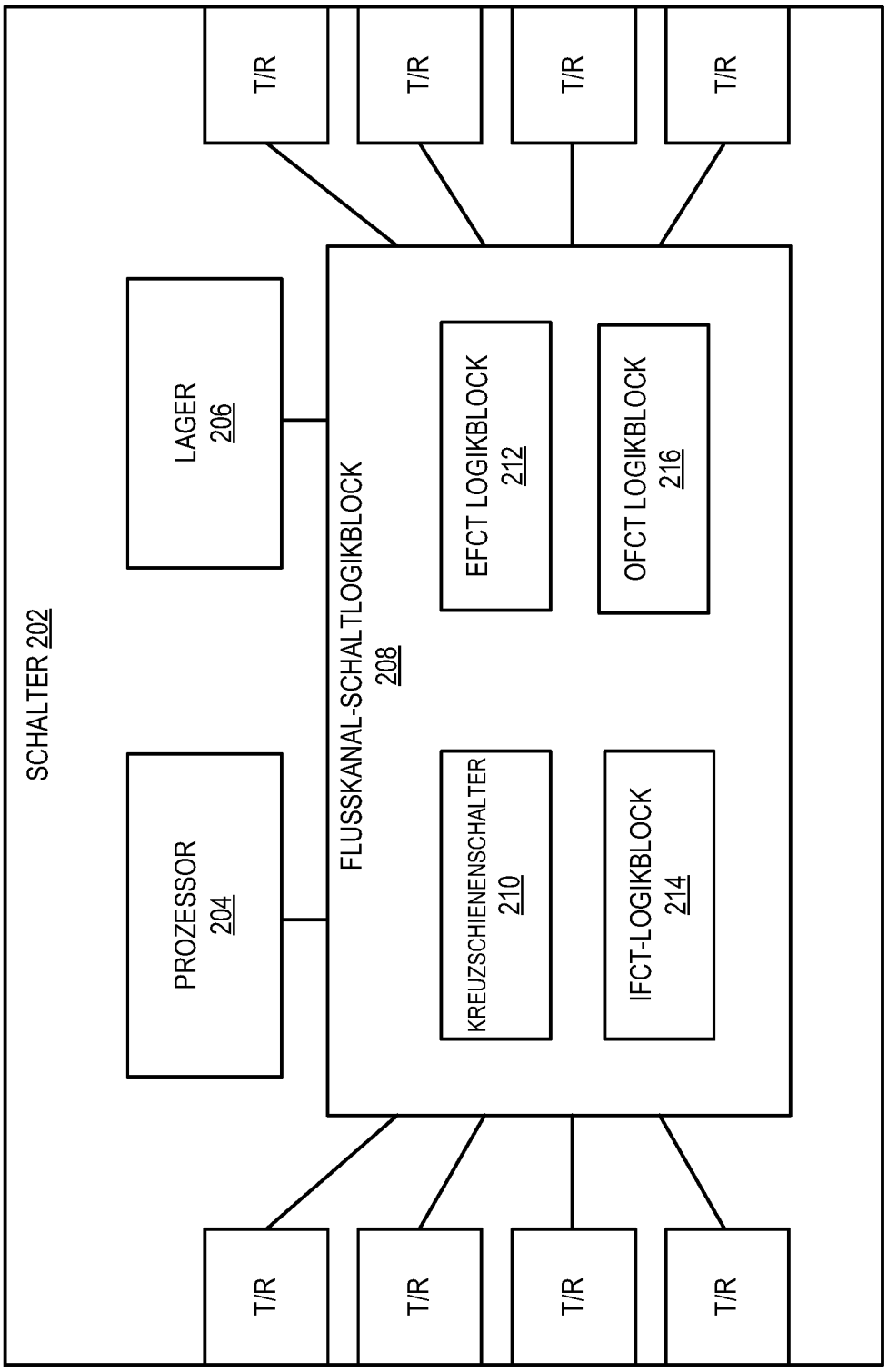

SCHALTER 202
T/R
T/R
T/R
T/R
LAGER 206
FLUSSKANAL-SCHALTLOGIKBLOCK 208
EFCT LOGIKBLOCK 212
OFCT LOGIKBLOCK 216
PROZESSOR 204
KREUZSCHIENENSCHALTER 210
IFCT-LOGIKBLOCK 214
T/R
T/R
T/R
T/R

ABB. 2

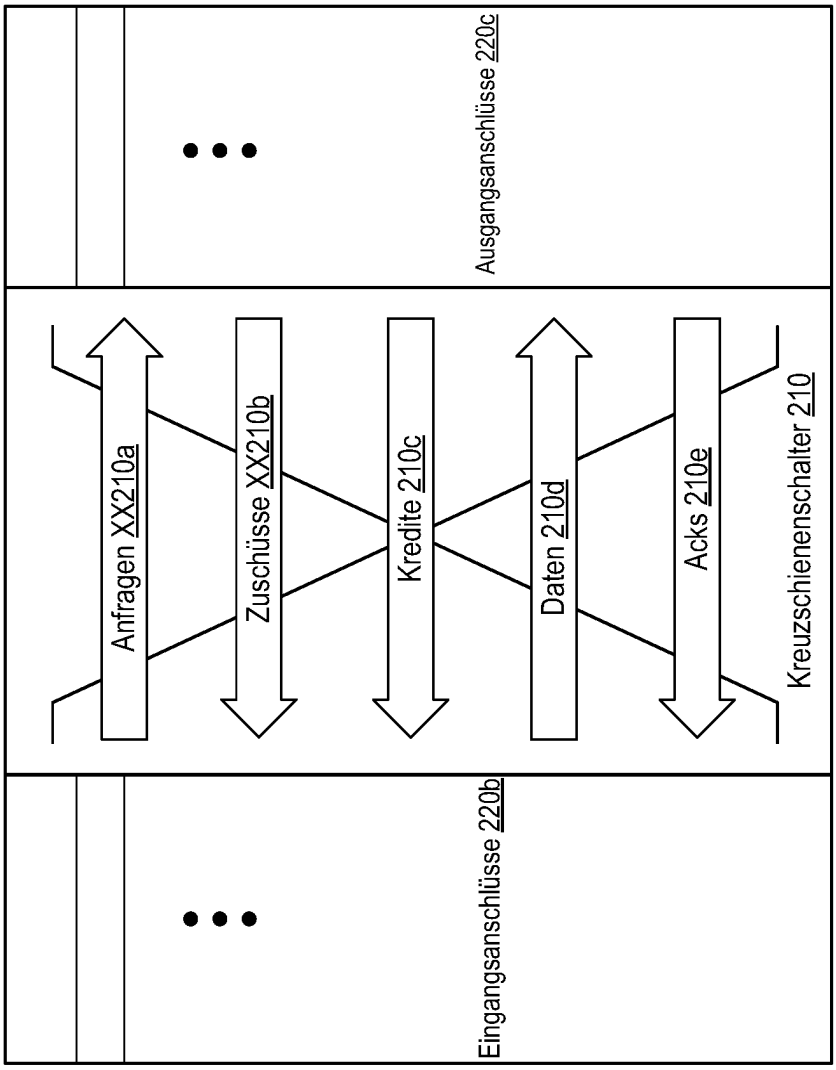

Ausgangsanschlüsse 220c
Anfragen XX210a
Zuschüsse XX210b
Kredite 210c
Daten 210d
Acks 210e
Kreuzschienenschalter 210
Eingangsanschlüsse 220b

ABB. 3A

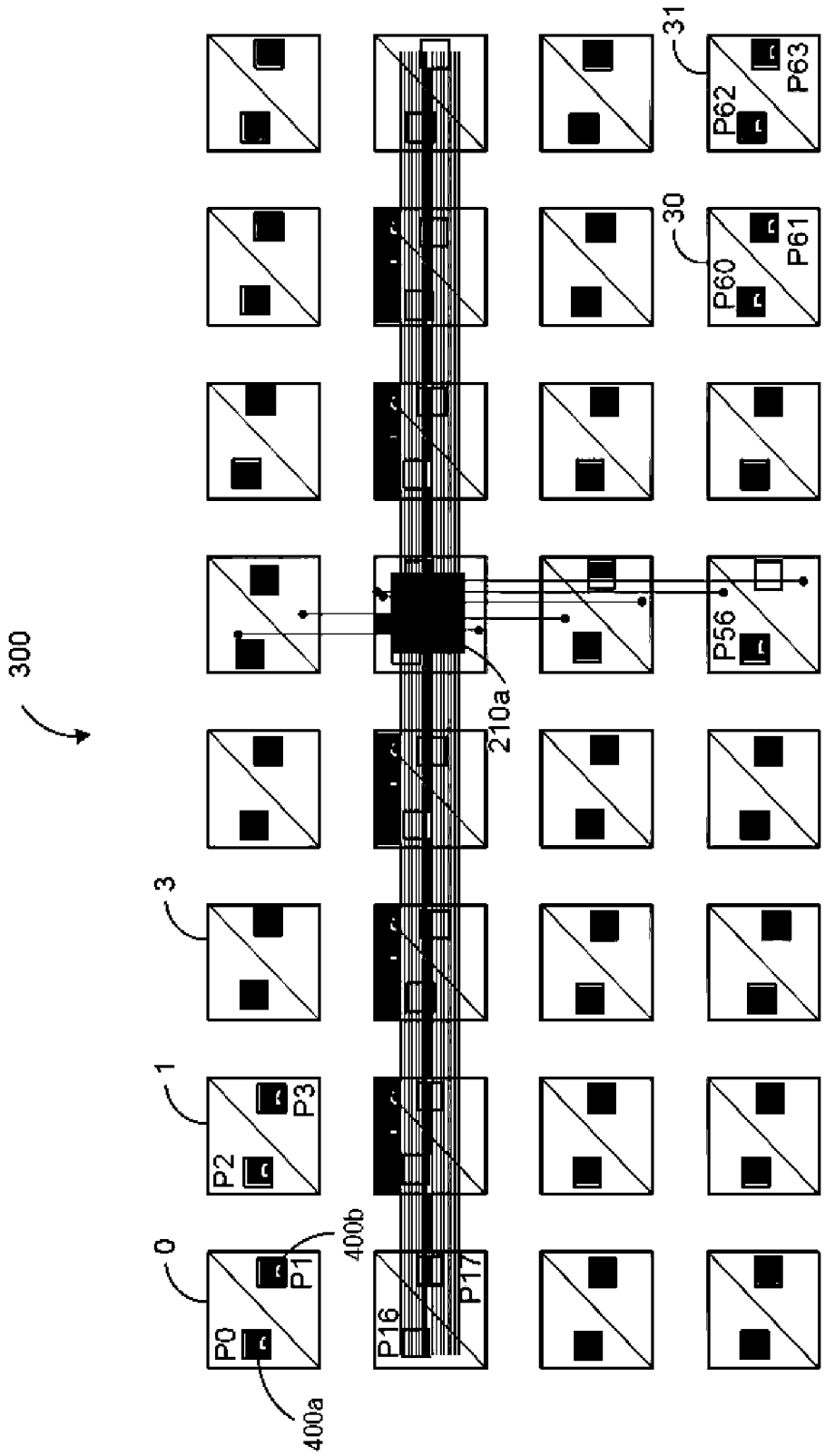

300
0
1
3
P0
P1
P2
P3
400a
400b
P16
P17
210a
P56
30
P60
P61
31
P62
P63

ABB. 3B

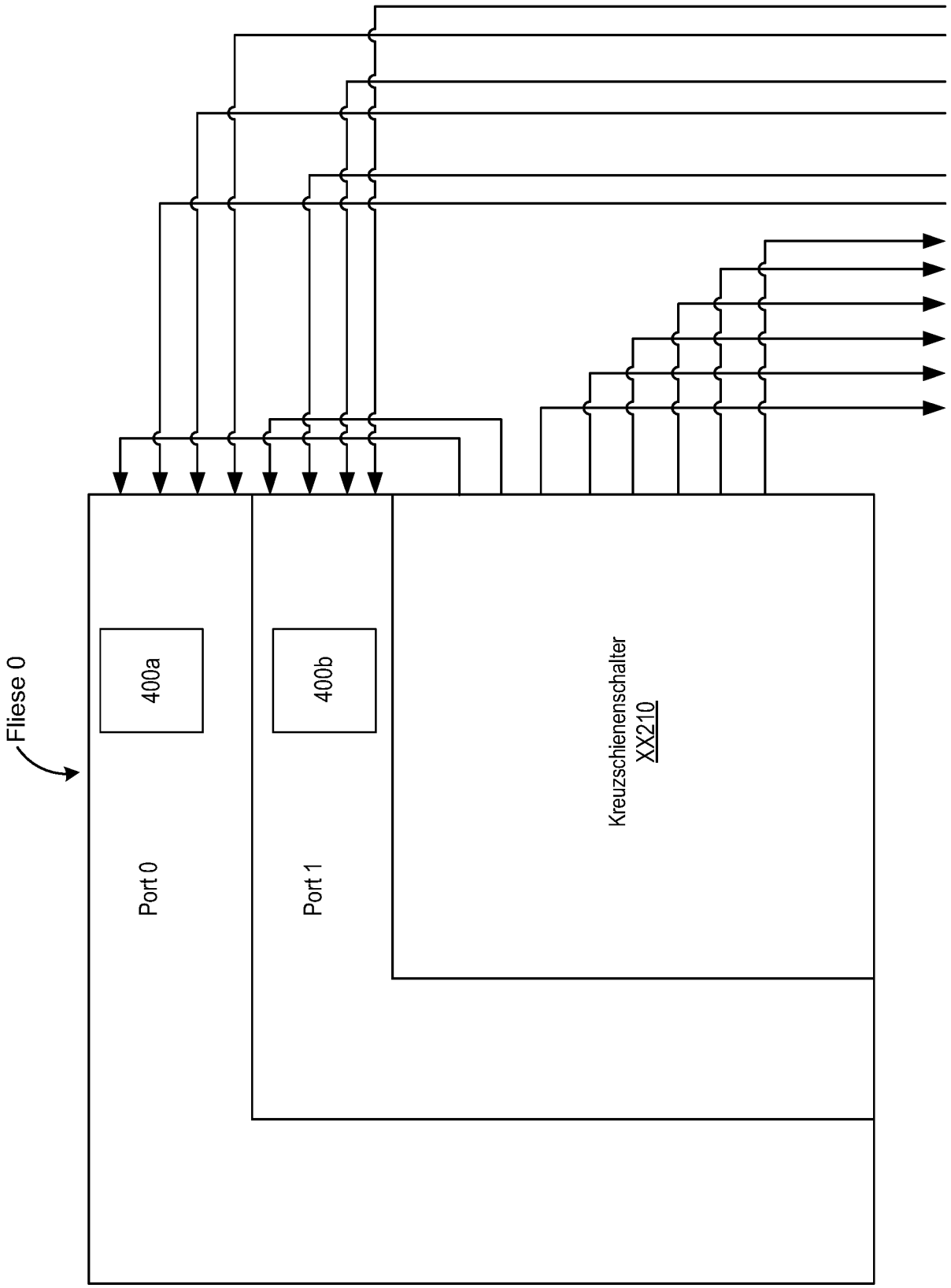

Fliese 0
400a
400b
Port 0
Port 1
Kreuzschienenschalter
XX210

ABB. 3C

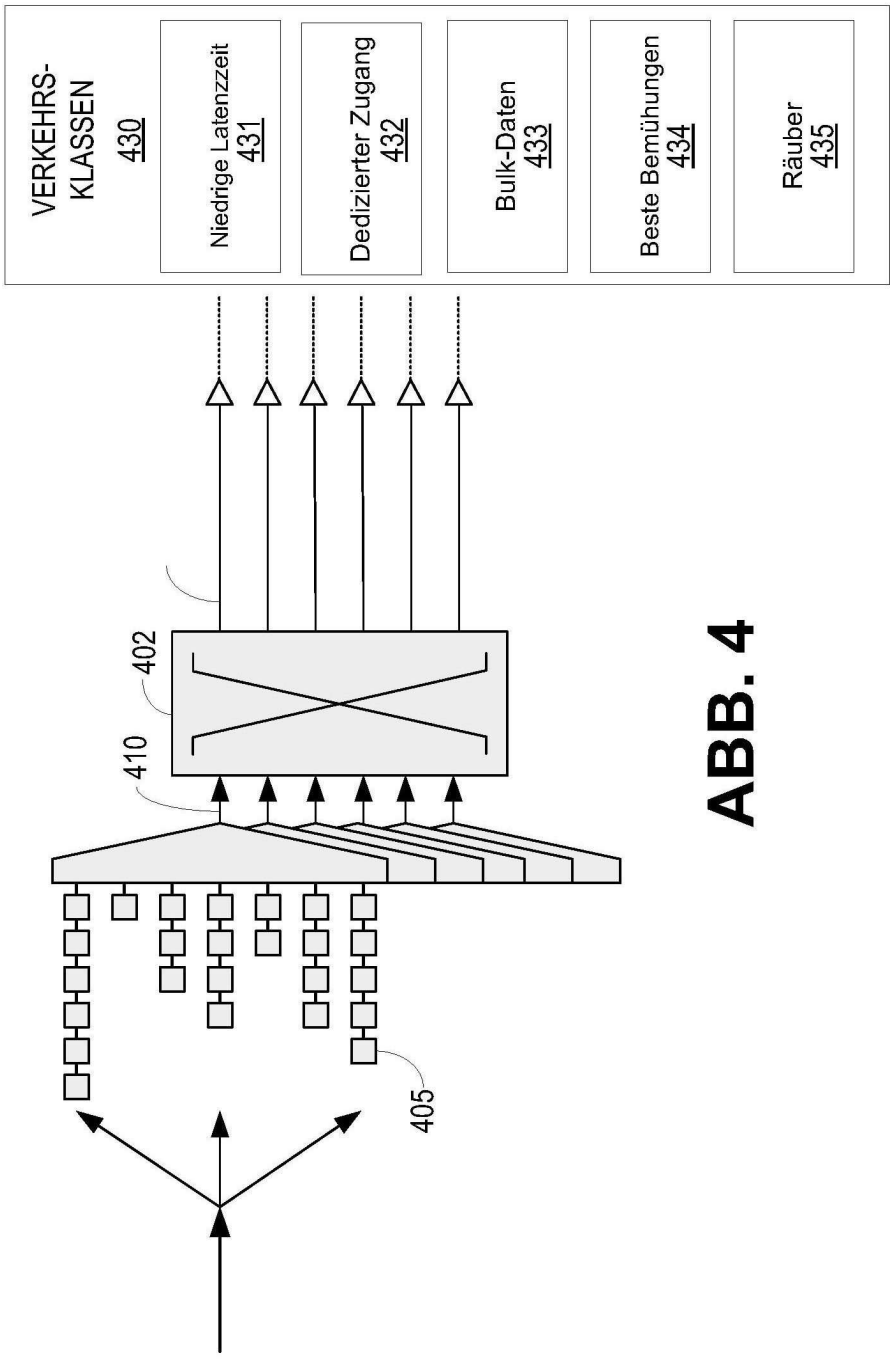

VERKEHRS-KLASSEN 430
Niedrige Latenzzeit 431
Dedizierter Zugang 432
Bulk-Daten 433
Beste Bemühungen 434
Räuber 435
402
410
405

ABB. 4

Abbildung179 FRF-Blockdiagramm der obersten Ebene (Teil A)

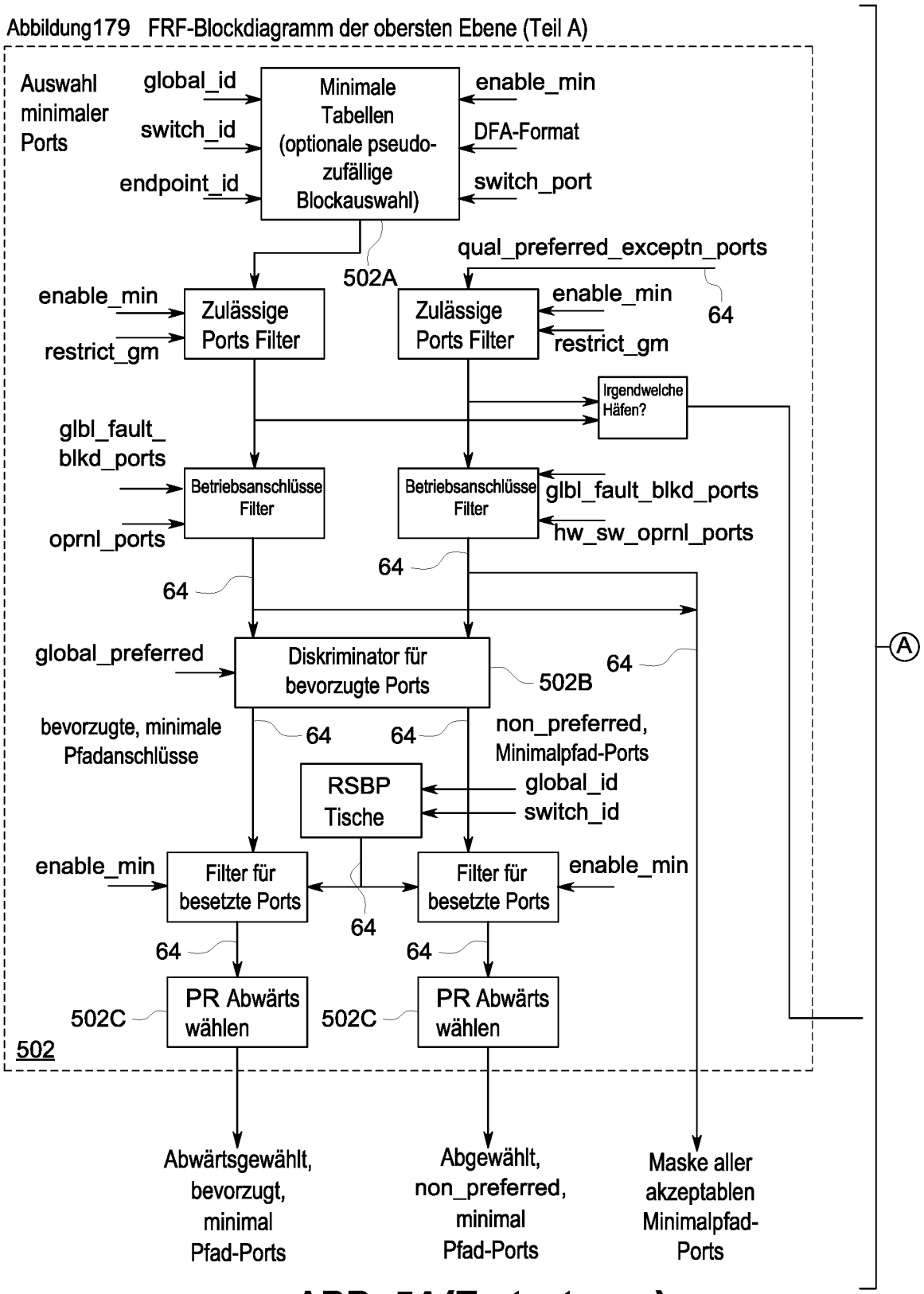


ABB. 5A(Fortsetzung)

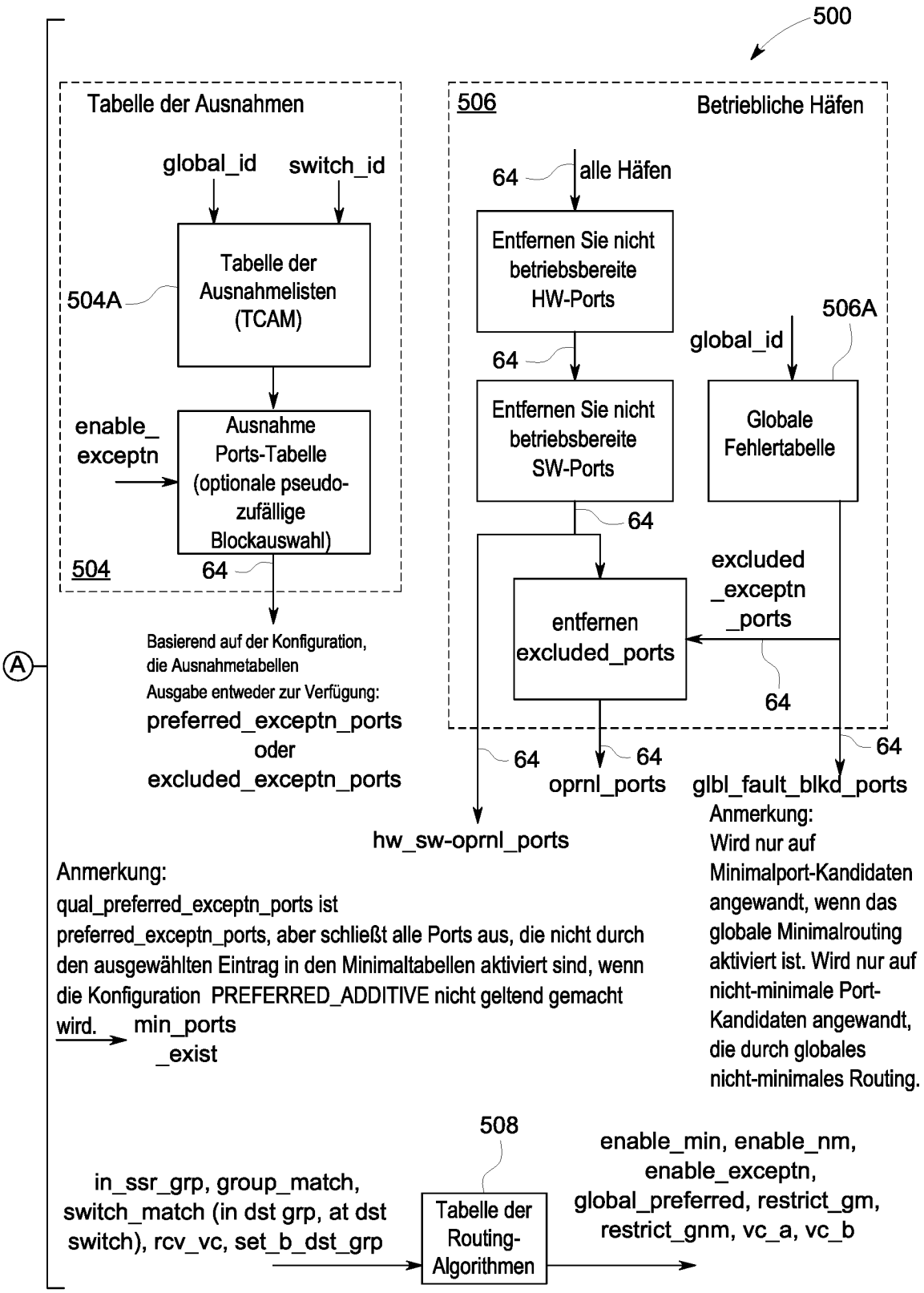

500
Tabelle der Ausnahmen
global_id
switch_id
Tabelle der Ausnahmelisten (TCAM)
504A
enable_ exceptn
Ausnahme Ports-Tabelle (optionale pseudo-zufällige Blockauswahl)
504
64
Basierend auf der Konfiguration, die Ausnahmetabellen
Ausgabe entweder zur Verfügung:
preferred_exceptn_ports
oder
excluded_exceptn_ports
A
506
Betriebliche Häfen
64
alle Häfen
Entfernen Sie nicht betriebsbereite HW-Ports
64
Entfernen Sie nicht betriebsbereite SW-Ports
64
506A
global_id
Globale Fehlertabelle
entfernen excluded_ports
excluded _exceptn _ports
64
64
64
64
64
hw_sw-oprnl_ports
oprnl_ports
glbl_fault_blkd_ports
Anmerkung:
Wird nur auf Minimalport-Kandidaten angewandt, wenn das globale Minimalrouting aktiviert ist. Wird nur auf nicht-minimale Port-Kandidaten angewandt, die durch globales nicht-minimales Routing.
Anmerkung:
qual_preferred_exceptn_ports ist preferred_exceptn_ports, aber schließt alle Ports aus, die nicht durch den ausgewählten Eintrag in den Minimaltabellen aktiviert sind, wenn die Konfiguration PREFERRED_ADDITIVE nicht geltend gemacht wird.
min_ports _exist
508
in_ssr_grp, group_match, switch_match (in dst grp, at dst switch), rcv_vc, set_b_dst_grp
Tabelle der Routing-Algorithmen
enable_min, enable_nm, enable_exceptn, global_preferred, restrict_gm, restrict_gnm, vc_a, vc_b

ABB. 5A

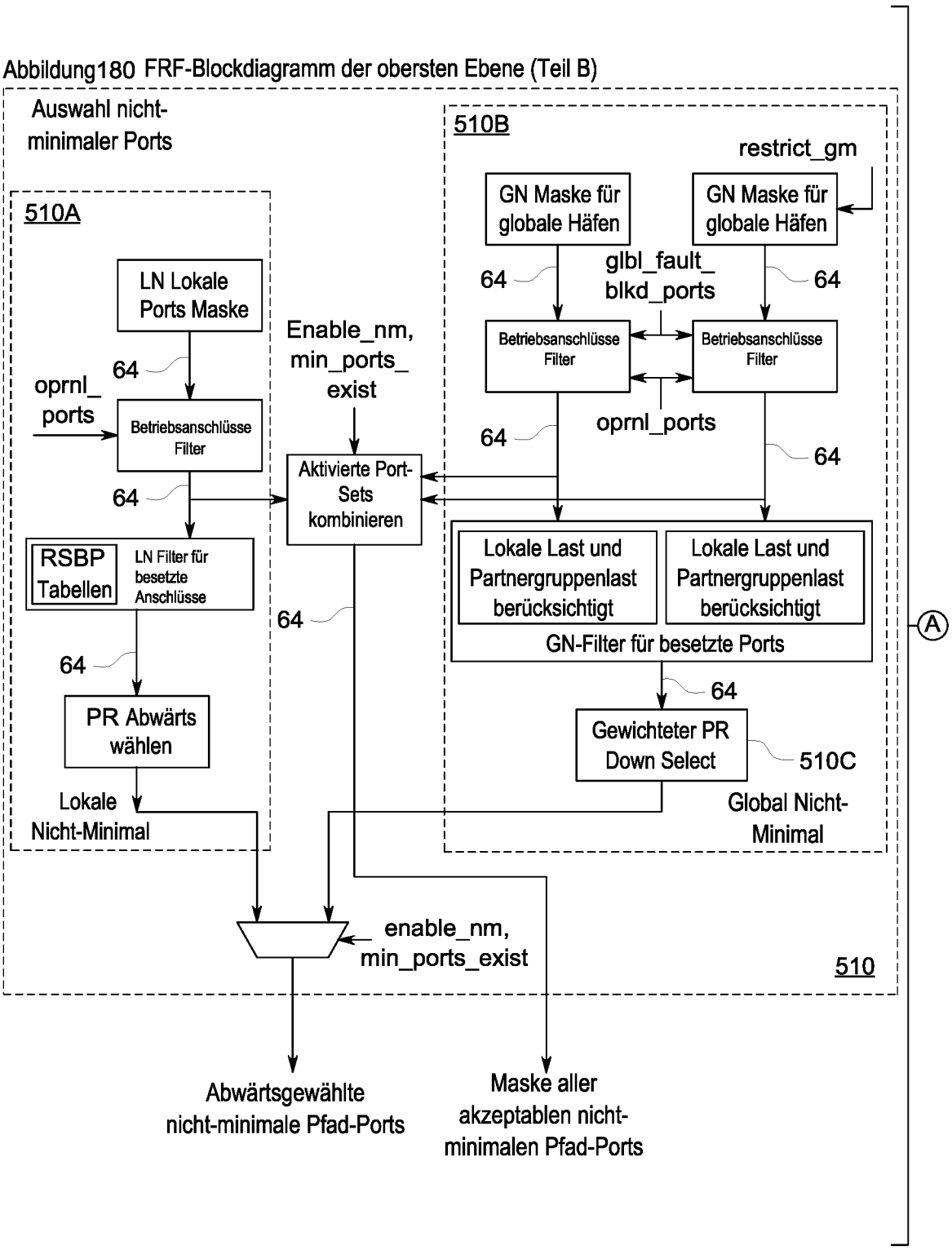


FIG. 5B Fortsetzung)

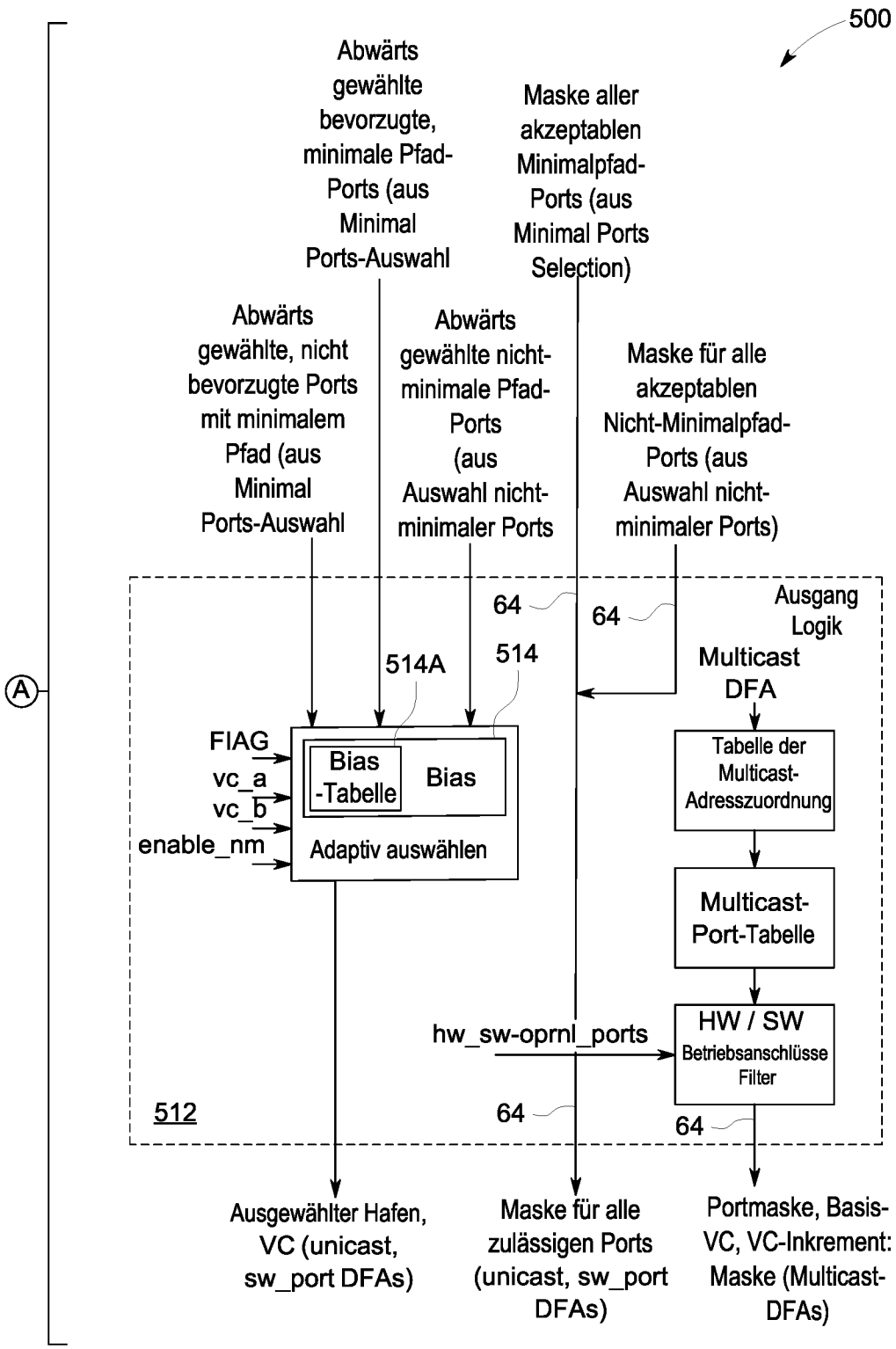

500
Abwärts gewählte bevorzugte, minimale Pfad-Ports (aus Minimal Ports-Auswahl
Maske aller akzeptablen Minimalpfad-Ports (aus Minimal Ports Selection)
Abwärts gewählte, nicht bevorzugte Ports mit minimalem Pfad (aus Minimal Ports-Auswahl
Abwärts gewählte nicht-minimale Pfad-Ports (aus Auswahl nicht-minimaler Ports
Maske für alle akzeptablen Nicht-Minimalpfad-Ports (aus Auswahl nicht-minimaler Ports)
64
64
Ausgang Logik
514A
514
Multicast DFA
FIAG
vc_a
vc_b
enable_nm
Bias -Tabelle
Bias
Adaptiv auswählen
Tabelle der Multicast-Adresszuordnung
Multicast-Port-Tabelle
HW / SW Betriebsanschlüsse Filter
hw_sw-oprnl_ports
512
64
64
A
Ausgewählter Hafen, VC (unicast, sw_port DFAs)
Maske für alle zulässigen Ports (unicast, sw_port DFAs)
Portmaske, Basis-VC, VC-Inkrement: Maske (Multicast-DFAs)

ABB. 5B

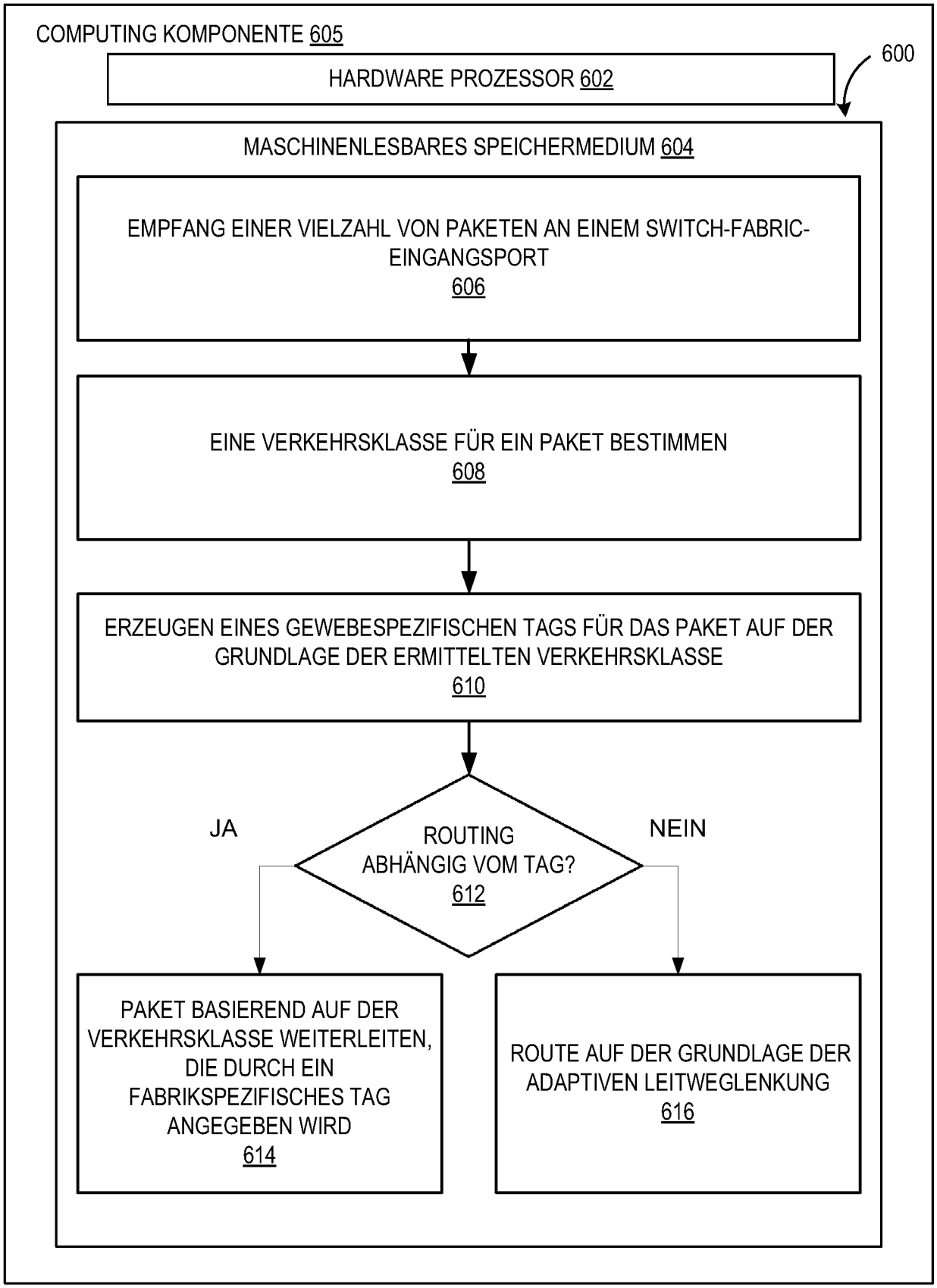

COMPUTING KOMPONENTE 605
600
HARDWARE PROZESSOR 602
MASCHINENLESBARES SPEICHERMEDIUM 604
EMPFANG EINER VIELZAHL VON PAKETEN AN EINEM SWITCH-FABRIC-EINGANGSPORT
606
EINE VERKEHRSKLASSE FÜR EIN PAKET BESTIMMEN
608
ERZEUGEN EINES GEWEBESPEZIFISCHEN TAGS FÜR DAS PAKET AUF DER GRUNDLAGE DER ERMITTELTEN VERKEHRSKLASSE
610
ROUTING ABHÄNGIG VOM TAG?
612
JA
NEIN
PAKET BASIEREND AUF DER VERKEHRSKLASSE WEITERLEITEN, DIE DURCH EIN FABRIKSPEZIFISCHES TAG ANGEGEBEN WIRD
614
ROUTE AUF DER GRUNDLAGE DER ADAPTIVEN LEITWEGLENKUNG
616

ABB. 6

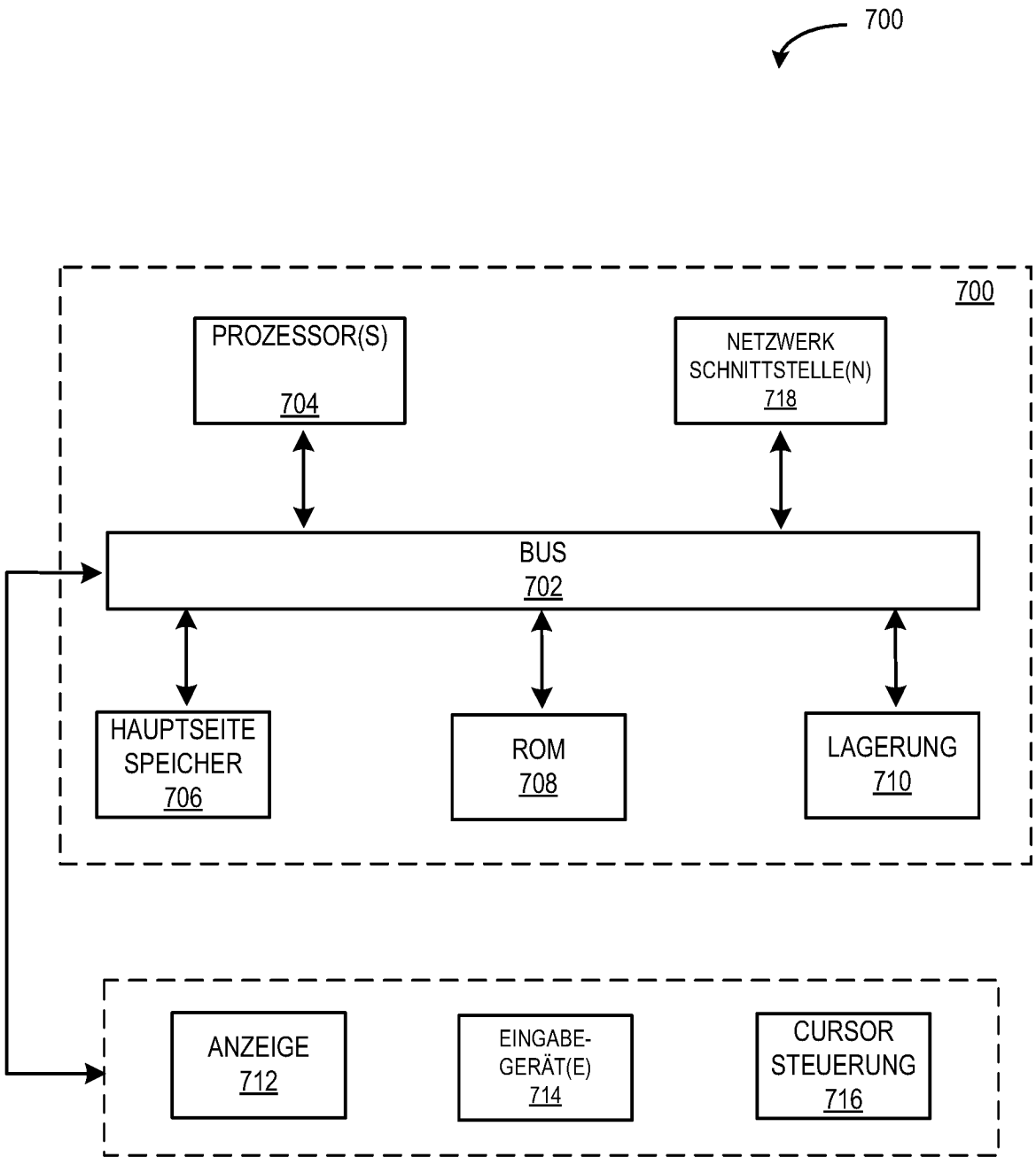


ABB. 7

