(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号

特許第6169251号
(P6169251)

(45)発行日　平成29年7月26日(2017.7.26)　(24)登録日　平成29年7月7日(2017.7.7)

(51)Int.Cl.			F I		
G06F	13/00	(2006.01)	G06F	13/00	357Z
H04L	12/803	(2013.01)	H04L	12/803	
H04L	12/743	(2013.01)	H04L	12/743	
H04L	12/951	(2013.01)	H04L	12/951	

請求項の数 15　(全 78 頁)

(21)出願番号	特願2016-509083(P2016-509083)
(86)(22)出願日	平成26年4月16日(2014.4.16)
(65)公表番号	特表2016-520904(P2016-520904A)
(43)公表日	平成28年7月14日(2016.7.14)
(86)国際出願番号	PCT/US2014/034426
(87)国際公開番号	WO2014/172499
(87)国際公開日	平成26年10月23日(2014.10.23)
審査請求日	平成27年12月8日(2015.12.8)
(31)優先権主張番号	13/864,157
(32)優先日	平成25年4月16日(2013.4.16)
(33)優先権主張国	米国(US)

(73)特許権者　507303550
アマゾン・テクノロジーズ・インコーポレーテッド
アメリカ合衆国・９８１０８－１２２６・ワシントン州・シアトル・ピイオーボックス・８１２２６
(74)代理人　100064621
弁理士　山川　政樹
(74)代理人　100098394
弁理士　山川　茂樹
(72)発明者　ソレンソン，ザ　サード，ジェームズ・クリストファー
アメリカ合衆国・９８１０９－５２１０・ワシントン州・シアトル・テリー　アヴェニュ　ノース・４１０

最終頁に続く

(54)【発明の名称】分散型ロードバランサにおける非対称パケットフロー

(57)【特許請求の範囲】

【請求項１】

複数のロードバランサノードの少なくとも２つが入口サーバとして構成され、

前記複数のロードバランサノードの少なくとも２つがフロートラッカーノードとして構成される、

前記複数のロードバランサノードと、

複数のサーバノードと、

１つまたは複数の顧客からのパケットフローを、ハッシュ化されたマルチパスルーティング技術に従って、前記入口サーバへと分散させるよう構成されたルータと、

を備えた分散型ロードバランサシステムであり、

顧客のためのパケットフローにおけるパケットを前記ルータから受信し、

前記複数のサーバノードへの前記パケットフローのためのマッピングを前記入口サーバが有しないことを決定し、

前記パケットのソースおよび宛先アドレス情報に適用される一貫したハッシュ関数に従って、前記パケットフローのための少なくとも１つのフロートラッカーノードを決定し、

前記パケットフローのための前記複数のサーバノードの特定の１つへの接続のマッピングを、少なくとも１つのフロートラッカーノードから取得し、

前記特定のサーバノードへの前記パケットフローの１つ以上のパケットを送信する

ように各入口サーバが構成される、

分散型ロードバランサシステム。

【請求項２】
前記パケットフローがトランスミッションコントロールプロトコル（ＴＣＰ）パケットフローである、請求項１に記載の分散型ロードバランサシステム。
【請求項３】
前記複数のロードバランサノードの少なくとも２つが、前記１つまたは複数の顧客への前記サーバノードからの発信パケットを送信するよう構成された出口サーバとして構成され、
前記パケットフローのための前記出口サーバを選択し、
前記パケットフローのための１つまたは複数の発信パケットを、前記選択した出口サーバへと送信する、
ように前記サーバノードが構成され、
前記顧客への前記発信パケットを送信するように、前記出口サーバが構成され、
前記パケットフローのための前記選択された出口サーバが、前記パケットフローのための前記入口サーバとは異なるロードバランサノードである、
請求項１に記載の分散型ロードバランサシステム。
【請求項４】
前記サーバノードへの前記パケットの送信の前に、前記入口サーバが前記１つまたは複数のパケットをユーザデータグラムプロトコル（ＵＤＰ）に従ってカプセル化し、前記出口サーバへの前記発信パケットの送信の前に、前記サーバノードが前記発信パケットをＵＤＰに従ってカプセル化し、前記顧客への前記発信パケットの送信の前に、前記出口サーバが前記発信パケットから前記ＵＤＰカプセル封じを取り外す、請求項３に記載の分散型ロードバランサシステム。
【請求項５】
前記パケットフローのための前記出口サーバを選択し、
前記カプセル化された受信パケットを前記入口サーバから受信し、
前記パケットから前記ＵＤＰカプセル封じを取り外し、前記パケットを前記サーバノード上のサーバへと伝達させ、
前記サーバノード上の前記サーバから前記発信パケットを取得し、
ＵＤＰに従って、前記発信パケットをカプセル化し、
前記カプセル化された発信パケットを前記出口サーバへと送信する
ように構成されるロードバランサモジュールを前記サーバノードが含む、
請求項４に記載の分散型ロードバランサシステム。
【請求項６】
前記パケットフローのための前記複数のサーバノードの特定の１つへの接続のマッピングを前記少なくとも１つのフロートラッカーノードから取得するために、
前記パケットフローのための情報を含むメッセージを、前記入口サーバが前記パケットフローのための１次フロートラッカーへと送信し、
前記パケットフローのための前記情報を含むメッセージを、前記１次フロートラッカーが前記パケットフローのための２次フロートラッカーへと送信し、前記パケットフローのための前記１次および２次フロートラッカーが異なるロードバランサノードであり、
前記２次フロートラッカーが、前記パケットフローのための受信確認を、前記顧客へと送信し、
前記入口サーバが、受信確認パケットを前記顧客から受信し、前記受信確認パケットを前記１次フロートラッカーへと転送し、
前記１次フロートラッカーが、前記パケットフローを受信するための前記サーバノードとして、前記複数のサーバノードの中から前記特定のサーバノードを無作為に選択し、前記特定のサーバノードを示すメッセージを、前記２次フロートラッカーへと送信し、
前記２次フロートラッカーが、同期メッセージを生成して前記生成された同期メッセージを前記特定のサーバノードへと送信し、
前記２次フロートラッカーが、前記パケットフローのための接続情報を前記特定のサー

バノードから受信して前記１次フロートラッカーへの前記接続情報を含むメッセージを送信し、
前記１次フロートラッカーが、前記パケットフローのための前記接続情報を含むメッセージを前記入口サーバへと送信し、前記接続情報が前記パケットフローを前記特定のサーバノードへとマッピングする、
請求項１に記載の分散型ロードバランサシステム。
【請求項７】
前記生成された同期メッセージを前記２次フロートラッカーから受信し、
前記サーバノード上のサーバが接続を受け入れることができることを決定し、
前記生成された同期メッセージに従って同期パケットを生成し、前記同期パケットを前記サーバノード上の前記サーバへと伝達し、
前記サーバノード上の前記サーバによって生成された受信確認パケットを遮断し、
前記接続情報を含むメッセージを前記２次フロートラッカーへと送信する、
ように構成されるロードバランサモジュールを前記サーバノードが含む、
請求項６に記載の分散型ロードバランサシステム。
【請求項８】
顧客からのパケットフローにおけるパケットの受信であって、１つまたは複数の顧客から一貫したハッシュ関数に従って前記複数のロードバランサノードへと前記パケットフローを分散させるルータからのパケットの受信、
前記パケットのソースおよび宛先アドレス情報に適用される一貫したハッシュ関数に従っての、前記パケットフローのためのフロートラッカーノードとしての役割を担うロードバランサノードの決定、
前記パケットフローのための複数のサーバノードの１つへの接続のマッピングの、前記パケットフローのための前記フロートラッカーノードからの取得、
前記マッピングにより示された、前記サーバノードへの前記パケットフローの１つまたは複数のパケットの送信、
を、複数のロードバランサノードのひとつの入口サーバによって実行すること、
を備えた方法。
【請求項９】
前記パケットフローがトランスミッションコントロールプロトコル（ＴＣＰ）パケットフローである、請求項８に記載の方法。
【請求項１０】
前記パケットフローの前記１つまたは複数のパケットの前記サーバノードへの前記送信の前に、ユーザデータグラムプロトコル（ＵＤＰ）に従って前記パケットをカプセル化する前記入口サーバをさらに備えた、請求項８に記載の方法。
【請求項１１】
前記サーバノードによる、前記パケットフローのための出口サーバとしての前記複数のロードバランサノードの１つの選択であり、前記パケットフローのための前記選択した出口サーバが、前記パケットフローのための前記入口サーバとは異なるロードバランサノードである、前記複数のロードバランサノードの１つの選択と、
前記サーバノードによる、前記パケットフローのための１つまたは複数の発信パケットの、前記選択された出口サーバへの送信と、
前記出口サーバによる前記発信パケットの、前記パケットフローの前記顧客への送信と、
をさらに備えた、請求項８に記載の方法。
【請求項１２】
前記発信パケットの前記出口サーバへの前記送信の前に、ユーザデータグラムプロトコル（ＵＤＰ）に従って前記発信パケットをカプセル化する前記サーバノードと、
前記発信パケットの顧客への前記送信の前に、前記発信パケットから前記ＵＤＰカプセル封じを取り外す前記出口サーバと、

をさらに備えた、請求項１１に記載の方法。
【請求項１３】
前記フロートラッカーノードが前記パケットフローのための１次フロートラッカーノードであり、前記一貫したハッシュ関数に従った一貫したハッシュリングにおける次のロードバランサノードが前記パケットフローのための２次フロートラッカーノードである、請求項８に記載の方法。
【請求項１４】
前記入口サーバによる、少なくとも１つのメッセージの前記パケットフローのための前記１次フロートラッカーノードへの送信であり、各メッセージが前記ルータから受信された前記パケットフローのパケットを含む前記送信と、
前記１次フロートラッカーノードによる、前記複数のサーバノードからの前記パケットフローのための前記サーバノードの選択と、
前記１次フロートラッカーノードによる、前記選択されたサーバノードを示すパケットフロー情報の前記２次フロートラッカーノードへの送信と、
前記２次フロートラッカーノードによる、前記サーバノードと前記顧客との通信による、前記パケットフローのための前記選択されたサーバノードへの前記接続の確立の円滑化と、
前記２次フロートラッカーノードによる、前記パケットフローのための接続情報の、前記１次フロートラッカーノードを通じた前記入口サーバへの送信であり、前記接続情報が前記選択されたサーバノードへの前記パケットフローのマッピングを行う前記送信と、
を、前記パケットフローのための複数のサーバノードの１つへの接続のマッピングの、前記パケットフローのための前記フロートラッカーノードからの前記取得が備える、
請求項１３に記載の方法。
【請求項１５】
前記２次フロートラッカーノードによる、前記サーバノード上の前記ロードバランサモジュールへの、生成された同期メッセージの送信と、
前記サーバノード上のサーバが接続を受け入れることができることの決定、
前記生成された同期メッセージに従った同期パケットの生成、
前記同期パケットの、前記サーバノード上の前記サーバへの伝達、
前記サーバノード上の前記サーバにより生成された受信確認パケットの遮断と、
前記接続情報を含むメッセージの、前記２次フロートラッカーノードへの送信
を、前記サーバノード上の前記ロードバランサモジュールにより実行することと、
を、前記サーバノードと前記顧客との通信により、前記パケットフローのための前記選択されたサーバノードへの前記接続の確立を円滑化するロードバランサモジュールを前記サーバノードが含む、
請求項１４に記載の方法。
【発明の詳細な説明】
【背景技術】
【０００１】
従来のロードバランサは通常、複数のネットワークインタフェースコントローラ（ＮＩＣ）を含む単一の専用装置であり、例えばその一部が顧客からのインバウンドトラフィック／顧客へのアウトバウンドトラフィックを処理し、残りがロードバランスされたホスト装置（例、ウェブサーバ等のサーバ）からのアウトバウンドトラフィック／そのようなホスト装置へのインバウンドトラフィックを処理するような、８つのＮＩＣなどを含む。これら従来のロードバランサの帯域またはスループットは通常、顧客側で４０ギガビット毎秒（Ｇｂｐｓ）、サーバ側で４０Ｇｂｐｓの範囲である。クラウドコンピューティングサービスのようなネットワークを活用したアプリケーションやネットワークを活用したサービスの規模および範囲の拡大につれて、データセンターには、数百、または数千ものロードバランスを必要とするホスト装置（例、ウェブサーバ）を格納する可能性が生じる。従来のロードバランサはそのような環境に対応できない。

【０００２】
さらに従来のロードバランサは通常、ホスト装置から収集されたデータに対して最大接続（またはｍａｘ　ｃｏｎｎ）、ラウンドロビン、および／または最小接続（ｌｅａｓｔ　ｃｏｎｎ）等の技術を適用し、どのホスト装置が接続の処理を行うかを選択する。また、従来のロードバランサは通常、ホスト装置によりフロントに配置され、それによって顧客からの接続（例、トランスミッションコントロールプロトコル（ＴＣＰ）接続）を遮断し、ホスト装置およびロードバランサの間で確立されたＴＣＰ接続において顧客のトラフィックをホスト装置へ送信する、ホスト装置へのプロキシの役割を担う。したがってこれら従来のロードバランサの使用時には、ホスト装置と顧客は直接ＴＣＰ接続を通じた通信を行わない。

【図面の簡単な説明】
【０００３】
【図１】少なくともいくつかの実施形態による、分散型ロードバランスシステムの実施例のブロック図である。
【図２】少なくともいくつかの実施形態による、図１の分散型ロードバランサシステムにより実装されてもよいロードバランス方法のハイレベルフローチャートである。
【図３】少なくともいくつかの実施形態による、入口、出口およびフロートラッカーの構成要素を含むロードバランサノードの実施例を示す。
【図４】少なくともいくつかの実施形態による、分散型ロードバランサにおけるルーティングおよびパケットフローを示す。
【図５】少なくともいくつかの実施形態による、エッジルータへの入口ノードの提供を示す。
【図６】少なくともいくつかの実施形態による、マルチパスルーティング方法のフローチャートである。
【図７】少なくともいくつかの実施形態による、非対称パケットフローを図示する。
【図８】少なくともいくつかの実施形態による、分散型ロードバランスシステムにおけるパケットフローを示す。
【図９Ａ】少なくともいくつかの実施形態による、分散型ロードバランスシステムにおいて接続を確立する際のパケットフローのフローチャートを提示する。
【図９Ｂ】少なくともいくつかの実施形態による、分散型ロードバランスシステムにおいて接続を確立する際のパケットフローのフローチャートを提示する。
【図１０Ａ】少なくともいくつかの実施形態による、分散型ロードバランスシステムにおけるパケットフローを示す。
【図１０Ｂ】少なくともいくつかの実施形態による、分散型ロードバランスシステムにおけるパケットフローを示す。
【図１０Ｃ】少なくともいくつかの実施形態による、分散型ロードバランスシステムにおけるパケットフローを示す。
【図１０Ｄ】少なくともいくつかの実施形態による、分散型ロードバランスシステムにおけるパケットフローを示す。
【図１０Ｅ】少なくともいくつかの実施形態による、分散型ロードバランスシステムにおけるパケットフローを示す。
【図１０Ｆ】少なくともいくつかの実施形態による、分散型ロードバランスシステムにおけるパケットフローを示す。
【図１０Ｇ】少なくともいくつかの実施形態による、分散型ロードバランスシステムにおけるパケットフローを示す。
【図１１Ａ】少なくともいくつかの実施形態による、ロードバランサノードの一貫したハッシュリングにおいてメンバーシップに影響を与える処理を示す。
【図１１Ｂ】少なくともいくつかの実施形態による、ロードバランサノードの一貫したハッシュリングにおいてメンバーシップに影響を与える処理を示す。
【図１１Ｃ】少なくともいくつかの実施形態による、ロードバランサノードの一貫したハ

ッシュリングにおいてメンバーシップに影響を与える処理を示す。
【図１１Ｄ】少なくともいくつかの実施形態による、ロードバランサノードの一貫したハッシュリングにおいてメンバーシップに影響を与える処理を示す。
【図１２】少なくともいくつかの実施形態による、ヘルスチェック間隔に従って各ロードバランサノードにより実行されてもよいヘルスチェック方法のハイレベルフローチャートである。
【図１３】少なくともいくつかの実施形態による、別のロードバランサノードからのロードバランサノードのヘルスチェック方法を示す。
【図１４】少なくともいくつかの実施形態による、１つまたは複数の他のロードバランサノードのヘルスチェックを行うロードバランサノードを図示する。
【図１５】少なくともいくつかの実施形態による、サーバノードのヘルスチェックを行うロードバランサノードを示す。
【図１６】少なくともいくつかの実施形態による、ロードバランサノード１１０により維持されてもよい別のノードのヘルス状態を図示する。
【図１７】少なくともいくつかの実施形態による、各ロードバランサノードにより維持されてもよいヘルス情報を示す。
【図１８Ａ】少なくともいくつかの実施形態によるロードバランサノードの故障の処理を示す。
【図１８Ｂ】少なくともいくつかの実施形態によるロードバランサノードの故障の処理を示す。
【図１９Ａ】少なくともいくつかの実施形態による、接続公開技術を図示する。
【図１９Ｂ】少なくともいくつかの実施形態による、接続公開技術を図示する。
【図２０】少なくともいくつかの実施形態による、各ロードバランサモジュールにより実行されてもよい接続公開方法のハイレベルフローチャートである。
【図２１】少なくともいくつかの実施形態による、対象のロードバランサノードへの接続公開パケットにおいて受信されるアクティブな接続情報の分散方法のフローチャートである。
【図２２】少なくともいくつかの実施形態による、対象のロードバランサノードへの接続公開パケットにおいて受信されるアクティブな接続情報の分散の代替方法を示す。
【図２３】少なくともいくつかの実施形態による、ロードバランサノードのソフトウェアスタックアーキテクチャの実施例を示す。
【図２４】実施形態において用いられてもよいコアパケット処理技術の態様を示す。
【図２５】少なくともいくつかの実施形態による、ロードバランサノードにおけるデータフローの処理のためのマルチコアパケットプロセッサの実施例を示す。
【図２６】少なくともいくつかの実施形態による、ロードバランサノードにおけるデータフローの処理のためのマルチコアパケットプロセッサの別の実施例を示す。
【図２７】少なくともいくつかの実施形態による、ロードバランサノード処理による受信パケットの処理を示す。
【図２８】少なくともいくつかの実施形態による、ロードバランサノード処理による発信パケットの処理を示す。
【図２９】少なくともいくつかの実施形態による、本番環境における分散型ロードバランサを含むロードバランスシステムを示す。
【図３０】少なくともいくつかの実施形態による、複数の分散型ロードバランスシステムの構成要素を、単一処理の過程でまたは単一処理として構成、実行することを可能にするメッセージバス機構を組み込む、分散型ロードバランサテストシステムを示す。
【図３１】少なくともいくつかの実施形態による、メッセージバスパケットアダプタおよびパケットパイプラインを示す。
【図３２】少なくともいくつかの実施形態による、メッセージバスパケットアダプタおよびパケットパイプラインを示す。
【図３３Ａ】少なくともいくつかの実施形態による、プロバイダネットワーク環境の実施

例を示す。
【図３３Ｂ】少なくともいくつかの実施形態による、図３３Ａで示されるプロバイダネットワーク環境の実施例における、分散型ロードバランサの実装を示す。
【図３４Ａ】少なくともいくつかの実施形態による、分散型ロードバランサおよびサーバノードの物理ラックの実装の実施例を示す。
【図３４Ｂ】少なくともいくつかの実施形態による、分散型ロードバランサおよびサーバノードの物理ラックの実装の別の実施例を示す。
【図３５】少なくともいくつかの実施形態による、ネットワーク上に１つまたは２つ以上の分散型ロードバランサが実装されるネットワーク環境の実施例を示す。
【図３６】いくつかの実施形態において用いられてもよいコンピュータシステムの実施例を示すブロック図である。
【０００４】
本明細書内ではいくつかの実施形態および例示的な図面のために例として実施形態が記載されているが、実施形態が本明細書内で示される実施形態または図面のみに限られないことを当業者は認識すべきである。本明細書に記載の図面および詳細な説明は、本明細書で開示される特定の形式に実施形態を限定することを意図するものではなく、反対に、添付の請求項で定められる精神および範囲から逸脱することのないすべての修正、均等物および代替物を包括する意図があることを理解すべきである。本明細書で使用される見出しは構成上の目的のためのみであり、説明また請求の範囲を限定するために用いる意図はない。本願を通じて用いられる通り、「してもよい（ｍａｙ）」という語は、必須の意（すなわち、義務）ではなく、むしろ許容の意（すなわち、可能性を示す意味）で用いられる。同様に、「含む（「ｉｎｃｌｕｄｅ」、「ｉｎｃｌｕｄｉｎｇ」および「ｉｎｃｌｕｄｅｓ」）」の語は含むことを意味するが、それに限定されない。
【発明を実施するための形態】
【０００５】
ネットワーク環境における分散型ロードバランス方法およびシステムの様々な実施形態が記載される。分散型ロードバランス方法およびシステムの実施形態は、様々なネットワーク環境における分散型ロードバランサの実施形態によって実装されてもよいように記載される。例として、分散型ロードバランサの実施形態は、図３３Ａおよび３３Ｂに示されるようなプロバイダネットワーク１９００等のローカルネットワーク上のインターネットや宛先、通常はサーバ（例、ウェブサーバ、アプリケーションサーバ、データサーバ等）のような外部ネットワーク上の顧客間のパケットフロー、例えばトランスミッションコントロールプロトコル（ＴＣＰ）技術を用いたパケットフロー等を円滑化し、維持するために用いられてもよい。本明細書に記載の実施形態は主にＴＣＰパケットフローの処理に関するが、実施形態はＴＣＰ以外のデータ通信プロトコルや、パケットフロー処理以外の用途に適用されてもよいことに留意する。
【０００６】
分散型ロードバランサは、特定の顧客と選択したサーバ（例、ウェブサーバ）との間のＴＣＰパケットフローを円滑化し維持する役割を担ってもよい。しかし分散型ロードバランサは、従来のロードバランサにおけるやり方では、顧客からのＴＣＰフローを遮断せず、また、サーバへのプロキシの役割も担わない。代わりに分散型ロードバランサのロードバランサノードが顧客から受信されたＴＣＰパケットを対象のサーバへルーティングし、サーバがそのＴＣＰスタックを用いて顧客へのＴＣＰ接続を管理する。すなわち、サーバが顧客からのＴＣＰパケットフローを遮断する。
【０００７】
また、従来のロードバランサ技術において行われるように、サーバから収集された情報に適用されるロードバランス技術またはアルゴリズムに基づいて、どのサーバが接続要求を送信するかに関して決定を下す１つまたは複数のロードバランサノードの代わりに、ロードバランサノードが新規接続要求を受信するサーバを無作為に選択してもよく、サーバノード上にある分散型ロードバランサの構成要素がそれぞれのサーバの現状の１つまたは

複数の測定基準に基づき、選択したサーバが新規接続要求を受け入れるか拒否するかに関する決定をローカルで下す。したがって、どのサーバが接続要求を受け入れるかに関する決定は、１つまたは複数のロードバランサノードから、接続を処理するサーバノードへと移行する。すなわち、接続要求の送信により近い場所およびタイミングへと決定が移行される。

【０００８】

顧客とサーバ間のパケットフローを円滑化し、維持するため、分散型ロードバランサの実施形態は様々な技術を用いてもよい。またそのような技術は、マルチパスルーティング技術、一貫したハッシュ技術、分散型ハッシュテーブル（ＤＨＴ）技術、境界ゲートウェイプロトコル（ＢＧＰ）技術、メンバーシップトラッキング、ヘルスチェック、接続公開、およびパケットのカプセル化および脱カプセル化を含むが、これらに限定されない。これらは分散型ロードバランスシステムの他の態様と同様に、図面に関連して以下に記載される。

分散型ロードバランスシステム

【０００９】

図１は、少なくともいくつかの実施形態による、分散型ロードバランスシステムの実施例のブロック図である。分散型ロードバランサの実施形態は、例えば図３３Ａおよび３３Ｂで示されるサービスプロバイダのプロバイダネットワーク１９００等のネットワーク１００において実装されてもよい。分散型ロードバランサシステムにおける顧客のパケット処理のハイレベル概観図に示される通り、ネットワーク１００の１つまたは複数の顧客１６０は、例えばインターネット等の外部ネットワーク１５０を通じて、ネットワーク１００の境界ルータ１０２に接続してもよい。境界ルータ１０２は、顧客１６０からの受信パケット（例、ＴＣＰパケット）を分散型ロードバランサの構成要素エッジルータ１０４へとルーティングしてもよい。エッジルータ１０４は受信パケットを、分散型ロードバランサシステムのロードバランサノードレイヤー上のロードバランサ（ＬＢ）ノード１１０へとルーティングする。少なくともいくつかの実施形態において、エッジルータ１０４は例えば等価コストマルチパス（ＥＣＭＰ）ハッシュ技術のようなフローごとにハッシュ化されたマルチパスルーティング技術に従って、ルーティングの決定を行ってもよい。次にロードバランサノード１１０がパケットをカプセル化し（例、ユーザデータグラムプロトコル（ＵＤＰ）に従って）、ネットワーク１００上のネットワークファブリック１２０（例、Ｌ３ネットワーク）を通じて、カプセル化されたパケットをサーバノード１３０上のローカルロードバランサモジュール１３２へとルーティングする。ファブリック１２０は、１つまたは複数のネットワーク装置または構成要素を含んでもよく、そのようなネットワーク装置または構成要素にはスイッチ、ルータ、およびケーブルが含まれるがこれらに限定されない。サーバノード１３０上では、ローカルロードバランサモジュール１３２がパケットを脱カプセル化し、顧客のＴＣＰパケットをサーバ１３４のＴＣＰスタックに送信する。サーバノード１３０上のサーバ１３４はその後そのＴＣＰスタックを利用し、顧客１６０への接続を管理する。

【００１０】

図２は少なくともいくつかの実施形態による、図１の分散型ロードバランサシステムにより実装されてもよいロードバランス方法のハイレベルフローチャートである。分散型ロードバランサシステムの実施形態は、従来のロードバランサ技術において行われるような複数の宛先（例、ウェブサーバ）間の負荷の割り当てに関する困難問題の解決を行わなくてもよい。例えば従来のロードバランサは通常、最大接続、ラウンドロビン、および／または最小接続等の技術やアルゴリズムを用い、どのサーバが接続の処理を行うべきかを選択する。しかしこれら技術には欠点があり、特にロードバランスに関する決定を行うために用いられるデータがほぼすぐに古くなってしまうような分散型システムにおいては良好に行うことが難しい。少なくともいくつかの分散型ロードバランサシステムの実施形態においては、従来のロードバランサにおいて行われるように接続要求を満たすために１つまたは複数のロードバランス技術を用いてサーバノード１３０の選択を試みる代わりに、ロ

ードバランサノードレイヤー上のロードバランサノード１１０が、顧客の接続のための要求を受信するサーバノード１３０を無作為に決定してもよい。そのサーバノード１３０が自身に負荷がかかり過ぎていると見なす場合は、サーバノード１３０はロードバランサノード１１０に接続要求を送信し返し、それによってサーバノード１３０が現在接続を処理できないことをロードバランサノード１１０に伝えてもよい。ロードバランサノードレイヤーはその後、接続要求を受信する別のサーバノード１３０を無作為に決定してもよいし、もしくはその代わりに、要求を行う顧客１６０に対してエラーメッセージを返信し、現在接続を確立できないことを顧客１６０に伝えてもよい。

【００１１】

図２の１０で示すように、分散型ロードバランサシステムのロードバランサノードレイヤーがソースから通信セッション（例、ＴＣＰ接続）のための要求を受信する。ソースは例えば、分散型ロードバランサシステムを実装するネットワーク１００への外部ネットワーク１５０上の顧客１６０であってもよい。少なくともいくつかの実施形態においては、要求はネットワーク１００の境界ルータ１０２で顧客１６０から受信されてもよく、また、エッジルータ１０４にルーティングされてもよい。エッジルータ１０４は例えば顧客１６０からの特定の接続要求のルーティング先となるロードバランサノード１１０を擬似ランダムに選択するフローごとの等価コストマルチパス（ＥＣＭＰ）ハッシュ技術を用いて、受信パケットをロードバランサノードレイヤー上のロードバランサ（ＬＢ）ノード１１０にルーティングする。

【００１２】

２０で示すように、ロードバランサノードレイヤーが宛先ノードを無作為に選択し、選択した宛先ノードに接続要求を転送する。宛先ノードは例えば、ロードバランサによってフロントに配置された複数のサーバノード１３０のうちの１つであってもよい。少なくともいくつかの実施形態において、ロードバランサレイヤー上のロードバランサノード１１０はすべての既知のサーバノード１３０の中から接続要求を受信するサーバノード１３０を無作為に選択してもよい。しかしいくつかの実施形態においては、すべての既知のサーバノード１３０の中から純粋に無作為に選択する以外の方法を用いて接続要求を受信するサーバノード１３０を選択してもよい。例えばいくつかの実施形態においては、サーバノード１３０の無作為な選択の重みづけを行うために、ロードバランサノード１１０によってサーバノード１３０に関する情報が利用されてもよい。実施例として、異なるサーバノード１３０が異なる種類の装置であるかまたは異なるＣＰＵによって構成されているために異なる能力や可能性を有するとロードバランサノード１１０が認識している場合、無作為な選択をサーバノード１３０の１つまたは複数の特定の種類または構成の方に（またはそれを避けるように）偏らせるために情報が用いられてもよい。

【００１３】

３０で示すように、宛先ノードが、通信セッションを受け入れることが可能か決定する。少なくともいくつかの実施形態において、サーバノード１３０上のローカルロードバランサ（ＬＢ）モジュール１３２がそれぞれのサーバ１３４の現状の１つまたは複数の測定基準に基づき、サーバノード１３０上のそれぞれのサーバ１３４が新規接続を受け入れることが可能であるかどうかを決定する。

【００１４】

４０において接続要求が受け入れられる場合は、その後５０で示すように宛先ノードが接続を処理できることを宛先ノードからロードバランサノードレイヤーに伝える。その後６０で示すようにソース（例、顧客１６０）と宛先ノード（例、サーバノード１３０上のサーバ１３４）との間にロードバランサノードレイヤーを通じて通信セッションが構築される。少なくともいくつかの実施形態において、サーバノード１３０上のサーバ１３４がＴＣＰスタックを用いて顧客１６０への接続を管理する。

【００１５】

４０において接続要求が受け入れられない場合は、その後７０で示すように宛先ノードがロードバランサノードレイヤーに通知し、メソッドは要素２０に戻ってもよい。ロード

バランサノードレイヤーはその後２０において別の宛先ノードを無作為に選択してもよいし、もしくはその代わりに、要求を行う顧客１６０に対して現在接続を確立できないことを顧客１６０に伝えてもよい。顧客１６０は、必ずそうするわけではないが、要素１０においてメソッドを再び開始するために接続要求を再提出してもよいことに留意する。

【００１６】

図１を再び参照する。分散型ロードバランサシステムの少なくともいくつかの実施形態は、コモディティハードウェアを用いてネットワーク１００上のエッジルータ１０４で受信された顧客のトラフィックをネットワーク１００上のサーバノード１３０へとルーティングしてもよい。分散型ロードバランサの少なくともいくつかの実施形態は、複数のロードバランサノード１１０を含むロードバランサノードレイヤーを含んでもよい。少なくともいくつかの実施形態において、各ロードバランサノード１１０はロードバランサノードレイヤーにおける複数の役割のうち、１つまたは複数の役割を担ってもよい。これらロードバランサノード１１０の役割には、入口ノード、また出口ノード、そして（所与のパケットフローのための１次フロートラッカーまたは２次フロートラッカーとしての）フロートラッカーノードの役割が含まれてもよい。少なくともいくつかの実施形態において各ロードバランサノード１１０は、ラック搭載型のコモディティコンピューティング装置等の個別のコンピューティング装置として、またはそういった個別のコンピューティング装置において、ロードバランサノードレイヤー上に実装されてもよい。ロードバランサノード１１０は一般的に、特定のパケットフローのための役割のうちただ１つ（しかし可能であれば２つまたは３つ）を担うが、少なくともいくつかの実施形態において、各ロードバランサノード１１０は入口ノード、また出口ノード、そして（所与のパケットフローのための１次フロートラッカーまたは２次フロートラッカーとしての）フロートラッカーノードの３つの役割それぞれを担う。しかし少なくともいくつかの実施形態において、ロードバランサノード１１０は特定のパケットフローのための１次フロートラッカーおよび２次フロートラッカーの両方の役割を担うことはできないことに留意する。その代わりにいくつかの実施形態においては、各ロードバランサノード１１０が３つの役割のうちただ１つを担ってもよい。この実施形態においては、コンピューティング装置の個別の組が、特に入口ノード、出口ノード、およびフロートラッカーノードとしてロードバランサノードレイヤー上で実装されてもよい。

【００１７】

少なくともいくつかの実施形態において、パケットフローのための１次および２次フロートラッカーの決定するために、一貫したハッシュおよび一貫したハッシュリング技術が適用されてもよい。顧客からの各パケットフローは、例えば顧客のＩＰアドレス、顧客用ポート、サーバ（パブリック）ＩＰアドレス、およびサーバポートから成る４タプルによって一意に識別されてもよい。この識別子は、顧客およびパブリックエンドポイントのペアを示すＣＰまたはＣｃＰｐとして略されてもよい。与えられた任意のＴＣＰフロー（またはＣＰペア）に関連するパケットは、エッジルータ１０４からのハッシュ化されたマルチパス（例、ＥＣＭＰ）フロー分散により、入口サーバ１１２として動作するいずれのロードバランサノード１１０上にも表れることができる。パケットが入口ノードとして動作するロードバランサノード１１０に到達する際に、どのロードバランサノード１１０がパケットフローのための状態維持の役割を担うか（すなわち、１次フロートラッカーノード）を入口ノードが決定できるように、一貫したハッシュが用いられる。どのロードバランサノード１１０がパケットフローのための状態維持の役割を担うかを決定するために、ＣＰペアは入口ノードにより一貫したハッシュリング内にハッシュ化されてもよい。一貫したハッシュリング内でパケットフローのためのＣＰペアの一貫したハッシュに従って決定されたノード１１０が、パケットフローのための１次フロートラッカーの役割を担うノード１１０である。少なくともいくつかの実施形態において、一貫したハッシュリングにおける後続ノードがパケットフローのための２次フロートラッカーの役割を担う。

【００１８】

図３は少なくともいくつかの実施形態による、３つの役割すべて（入口、出口、および

フロートラッカー）を実装する構成要素を含む、ロードバランサ（ＬＢ）ノード１１０の実施例を示す。この実施例において構成要素である入口サーバ１１２は、１つまたは複数の顧客からインバウンドＴＣＰパケットを受信し、ＴＣＰパケットをカプセル化されたパケットとして１つまたは複数のサーバに送信する、入口の役割を果たす。構成要素である出口サーバ１１４は、１つまたは複数のサーバからアウトバウンドのカプセル化されたパケットを受信し、脱カプセル化されたＴＣＰパケットを１つまたは複数の顧客に送信する、出口の役割を果たす。構成要素であるフロートラッカー１１６は、１つまたは複数の顧客１６０と１つまたは複数のサーバ１３４との間に確立された１つまたは複数のパケットフローのための１次または２次フロートラッカーの役割を果たす。それぞれの顧客１６０から受信された接続要求に応じて顧客とサーバ１３４の１つとの間のＴＣＰ接続を開始するため、またはパケットフローのためにマッピング情報を取得するため、入口サーバ１１２はまた、ロードバランサノード１１０上のフロートラッカー１１６または別のロードバランサノード１１０上のフロートラッカー１１６と通信してもよい。

ロードバランサノード

【００１９】

図１を再び参照する。少なくともいくつかの実施形態において、ロードバランサノードレイヤーにおけるロードバランサノード１１０は、ネットワーク上の１つまたは複数のルータ１０４から顧客のトラフィック（例えばＴＣＰパケット等のパケット）を受信し、ファブリック１２０上の分散型ロードバランサシステムにより用いられるプロトコル（例、ユーザデータグラムプロトコル（ＵＤＰ））に従ってパケットをカプセル化する。ロードバランサノードレイヤーはその後ファブリック１２０を介してカプセル化されたパケットを宛先サーバノード１３０に転送する。各サーバノード１３０はロードバランサシステムの構成要素であるローカルモジュール１３２を含む。モジュール１３２は本明細書内ではロードバランサモジュールまたは単にＬＢモジュールと称されてもよく、サーバノード１３０上のソフトウェア、ハードウェア、またはそれらの組み合わせにおいて実装されてもよい。各サーバノード１３０において、それぞれのロードバランサモジュール１３２がパケットを脱カプセル化し、通常のＴＣＰ処理のためにＴＣＰパケットをローカルＴＣＰスタックに送信する。少なくともいくつかの実施形態においては、ロードバランサノードレイヤーがすべての顧客サーバのＴＣＰフローのために状態情報を維持してもよい。しかし、ロードバランサノードレイヤー上のロードバランサノード１１０は、ＴＣＰフローに関する一切の解釈を実行することはできない。各フローは、それぞれのサーバノード１３０のサーバ１３４と顧客１６０との間で管理される。分散型ロードバランサシステムはＴＣＰパケットが正確な宛先サーバ１３４に確実に到着するようにする。各サーバノード１３０におけるロードバランサモジュール１３２は、ロードバランサノード１１０から受信した顧客の接続要求に応じてそれぞれのサーバ１３４が新規接続を受け入れるか拒否するかに関する決定を下す。

【００２０】

少なくともいくつかの実施形態において、分散型ロードバランスシステムは一貫したハッシュ技術を、例えば、どのサーバノード１３０が特定のＴＣＰパケットフローのための役割を担うかについてどの１つまたは複数のロードバランサノード１１０が記憶すべきかなどを決定するために用いてもよい。一貫したハッシュ技術を利用し、ロードバランサノードレイヤーにおけるロードバランサノード１１０は一貫したハッシュリングとして見なされてもよく、ロードバランサノード１１０はリングにおけるメンバーシップのトラッキングを継続し、一貫したハッシュ関数に従って、特定のパケットフローのための役割を果たすリングにおける特定のメンバーを決定してもよい。少なくともいくつかの実施形態においては、顧客１６０とサーバ１３４との間の各パケットフローのトラッキングの役割を担う２つのロードバランサノード１１０が存在する。これらのノード１１０は、１次フロートラッカー（ＰＦＴ）ノードおよび２次フロートラッカー（ＳＦＴ）ノードと称されてもよい。少なくともいくつかの実施形態において、１次フロートラッカーはフローのための一貫したハッシュリングにおける第１のロードバランサノード１１０であり、２次フロ

ートラッカーは一貫したハッシュリングにおける次のまたは後続の、１次フロートラッカーノードとは異なるロードバランサノード１１０である。この場合、１次フロートラッカーノードが故障した際には、その後２次フロートラッカーノードが新規１次フロートラッカーになってもよく、別のロードバランサノード１１０（例、一貫したハッシュリングにおける次のノード１１０）が２次フロートラッカーの役割を担ってもよい。少なくともいくつかの実施形態において、ロードバランサノード１１０は所与のパケットフローのための１次フロートラッカーおよび２次フロートラッカーの両方の役割を担うことはできないことに留意する。一貫したハッシュリングにおけるこのまたは別のメンバーシップの変更については、本明細書にて後述される。少なくともいくつかの実施形態において、ロードバランサの実装のための構成情報（例、現在実装されているロードバランサノード１１０およびサーバノード１３０の１つまたは複数の信頼すべきリスト）は、例えばファブリック１２０を通じてロードバランサノード１１０に接続される１つまたは複数のサーバ装置上で実装されてもよい分散型ロードバランスシステムの構成要素である、構成サービス１２２によって維持されてもよい。

【００２１】

少なくともいくつかの実施形態において、１次および２次フロートラッカーノードとしての役割に加えて、ロードバランサノード１１０は所与のフローのために、他の２つののうち１つの役割を果たしてもよい。すなわち、入口ノードの役割および出口ノードの役割である。パケットフローのための入口ノードは、エッジルータ１０４からそれぞれのパケットフローを受信し、パケットフローを（カプセル化されたパケットとして）ファブリック１２０を通じてサーバノード１３０上の選択されたサーバ１３４に転送する、ロードバランサノード１１０である。入口ノードは、実際の顧客データ（ＴＣＰデータパケット）をそれぞれの宛先サーバノード１３０へと移動させる唯一のロードバランサノード１１０である。入口ノードは、宛先サーバノード１３０上のそれぞれのロードバランサモジュール１３２へのＴＣＰフローのマッピングを維持し、顧客のトラフィックをどのロードバランサモジュール１３２へと転送すべきかを把握する。出口ノードは、ファブリック１２０を通じてサーバノード１３０から受信されたパケットフローのための応答トラフィックを、境界ネットワークを通じてそれぞれの顧客１６０に転送する役割を担うロードバランサノード１１０である。ロードバランサモジュール１３２は、応答パケットをロードバランサプロトコル（例、ＵＤＰ）に従ってサーバ１３４から得た応答パケットをカプセル化し、カプセル化された応答パケットを、ファブリック１２０を通じてフローのためのそれぞれの出口ノードに送信する。出口ノードはステートレスであり、単にデータパケットを脱カプセル化して境界ネットワーク上の応答パケット（例、ＴＣＰパケット）を境界ルータ１０２に送信し、外部ネットワーク１５０を通じてそれぞれの顧客１６０に送達する。

【００２２】

上述のように少なくともいくつかの実施形態において、各ロードバランサノード１１０は異なるパケットフローのための入口ノード、出口ノード、および／またはフロートラッカーノード（１次または２次いずれかのフロートラッカーとして）の役割を果たす。ロードバランサノードレイヤー上の単一のロードバランサノード１１０は、ノードが何のパケットフローを処理しているかに応じて、役割のうちのいずれか１つを担ってもよい。例えば、少なくともいくつかの実施形態において、ロードバランサノード１１０は、１つのパケットフローのために入口ノードの役割を、また別のパケットフローのために１次または２次フロートラッカーの役割を、そしてさらに別のパケットフローのために出口ノードの役割を果たしてもよい。また少なくともいくつかの実施形態においてロードバランサノード１１０は、例えば所与のパケットフローのために入口ノードおよび１次（または２次）フロートラッカーノードの役割をというように、同一のパケットフローのために複数の役割を果たしてもよい。しかし少なくともいくつかの実施形態において、ロードバランサノード１１０は冗長化および回復を目的として、同一のパケットフローのために１次および２次フロートラッカーの両方の役割を担うことはできない。

【００２３】

上記は、各ロードバランサノード１１０が入口サーバ、出口サーバ、およびフロートラッカーの３つの役割のうちいずれの役割を担ってもよい実施形態である。しかしいくつかの実施形態においては、コンピューティング装置の異なるグループが、ロードバランスシステムにおける異なる役割に割り当てられてもよい。例えばいくつかの実施形態においては、個別のコンピューティング装置上でそれぞれ実装される入口ノード、出口ノードおよびフロートラッカーノードの異なる組があってもよい。別の実施例として、いくつかの実施形態においては、コンピューティング装置の別の組が出口ノードの役割のみを担う一方で、コンピューティング装置の組の１つが入口ノードおよびフロートラッカーノードの両方の役割を担ってもよい。

ロードバランサモジュール

【００２４】

上述のように各サーバノード１３０は、ロードバランサシステムの構成要素であるローカルロードバランサモジュール１３２を含む。モジュール１３２は、サーバノード１３０のソフトウェア、ハードウェア、またはそれらの組み合わせにおいて実装されてもよい。少なくともいくつかの実施形態において、サーバノード１３０のロードバランサモジュール１３２は、発信パケットのカプセル化および受信パケットの脱カプセル化、ノード１３０上のサーバ１３４のためのロードバランスに関するローカルでの決定、ならびに接続公開の、３つの主要な役割を果たしてもよい。これら３つの役割は以下に簡単に記され、さらに本明細書で詳細に後述される。

【００２５】

分散型ロードバランスシステムの少なくともいくつかの実施形態は、ＴＣＰ接続を遮断せず、パケットのスプーフィングも行わない。ロードバランサノードレイヤーを通じて送信されるすべてのパケットのソースおよび宛先ＩＰアドレスは、パケットフローに関与するエンドポイント（すなわち、顧客１６０およびサーバ１３４）の実際のＩＰアドレスである。スプーフィングの代わりに、これら実施形態は、ロードバランサノード１１０とサーバノード１３０との間で送信される、例えばＵＤＰパケット等のすべてのパケットを、ファブリック１２０上でカプセル化する。フローのための入口ノードの役割を担うロードバランサノード１１０からサーバノード１３０に到着するパケットフロー内のインバウンドパケットはロードバランサノード１１０によってカプセル化されるため、パケットはノード１３０上のサーバ１３４のために脱カプセル化され、ローカルホストのＴＣＰフローへと方向を変えられる必要がある。ノード１３０上のロードバランサモジュール１３２がこの脱カプセル化を行う。同様に、サーバ１３４からのパケットフローのための発信データパケットは、ロードバランサモジュール１３２によってカプセル化され、ファブリック１２０を通じてパケットフローのための出口ノードの役割を果たすロードバランサノード１１０へと送信される。

【００２６】

少なくともいくつかの実施形態において、サーバノード１３０上のロードバランサモジュール１３２は、それぞれのサーバノード１３０上のサーバ１３４のためのロードバランスに関してローカルで決定を下す。特にサーバノード１３０上のロードバランサモジュール１３２は、それぞれのサーバ１３４が新規ＴＣＰ接続要求の受信に応じて別のＴＣＰフローを受け入れるかどうかの決定を行う。上述のように、ロードバランサノード１１０はロードバランサモジュール１３２に送信されるすべてのパケットをカプセル化するため、ロードバランサモジュール１３２は実際にはＴＣＰ同期（ＳＹＮ）パケットを顧客１６０から受信することはなく、その代わりにロードバランサモジュール１３２はカプセル化プロトコル（例、ＵＤＰ）に従ったフロートラッカー１１６からの接続要求メッセージを、受信する。ロードバランサモジュール１３２はこの接続要求メッセージを受け入れるか拒否することができる。ロードバランサモジュール１３２が接続要求メッセージを受け入れる場合、ロードバランサモジュール１３２はローカルホストに向けたＳＹＮパケットを作成する。ローカルホストが接続を受け入れる場合、これはそれぞれの顧客接続を処理する実際のＴＣＰスタックとなる。

【００２７】
少なくともいくつかの実施形態において、接続要求メッセージを受け入れるべきかどうかに関する決定を下すため、ロードバランサモジュール１３２はサーバノード１３０上の現在のリソース消費に関する１つまたは複数の測定基準を確認し、新規接続の処理のために使用可能なリソースが十分にある場合は、ロードバランサモジュール１３２が接続を受け入れる。少なくともいくつかの実施形態において、ロードバランサモジュール１３２により考慮されてもよいリソースの測定基準は、ＣＰＵ使用率、最新の帯域占有量、および確立された接続の数のうちの１つまたは複数を含んでもよいが、これらに限定されない。いくつかの実施形態においては、これらの測定基準の代わりに、またはこれらの測定基準に加えて他の測定基準が考慮されてもよい。例えばいくつかの実施形態においては、ロードバランサモジュールがサーバ待ち時間（すなわち、要求がサーバ接続バックログに留まる時間）を測定基準として考慮してもよく、また、サーバ待ち時間が閾値を超える場合には接続要求を拒否してもよい。これらのおよび／または他の測定基準を利用して、ロードバランサモジュール１３２はそれぞれのサーバ１３４のために、サーバ１３４が新規パケットフローを受け入れるか拒否するかに関する決定を下すことができる。少なくともいくつかの実施形態において、リソース利用率（例、Ｎ％利用）は個別の、または組み合わせた、また、閾値（例、９０％利用）と比較した測定基準から決定されてもよい。決定されたリソース利用率が閾値以上である場合、または、接続を追加することによりリソース利用率が閾値を超えることになる場合には、その後接続要求が拒否されてもよい。

【００２８】
少なくともいくつかの実施形態において、ロードバランサモジュール１３２は、接続要求メッセージを拒否するべきかどうか決定するために確率論的手法を実装してもよい。上記のようにリソース利用が閾値以上である場合にすべての接続要求を拒否する代わりに、この方法では２つ以上の異なる利用レベルにおいて異なる確率で接続要求を拒否してもよい。例えばリソース利用が８０％の場合にロードバランサモジュール１３２が２０％の確率で接続要求を拒否してもよく、リソース利用が９０％の場合にロードバランサモジュール１３２が２５％の確率で接続要求を拒否してもよく、リソース利用が９５％の場合にロードバランサモジュール１３２が５０％の確率で接続要求を拒否してもよく、そしてリソース利用が９８％以上の場合にロードバランサモジュール１３２がすべての接続要求を拒否してもよい。

【００２９】
少なくともいくつかの実施形態において、各接続要求メッセージには、ロードバランサモジュール１３２が接続要求メッセージを拒否した回数が含まれていてもよい。自身が閾値以上の回数に渡り拒否されたことをロードバランサモジュール１３０により受信された接続要求メッセージが示す場合、パフォーマンス測定基準がサーバノード１３０の接続要求を拒否すべきだと示していても、ロードバランサモジュール１３０は接続を受け入れてもよい。

【００３０】
場合によっては、接続要求メッセージの送信先であるロードバランサモジュール１３２がすべて、接続要求を拒否する可能性もある。少なくともいくつかの実施形態において、接続要求メッセージがロードバランサモジュール１３２同士の間で無期限にバウンスされるのを防ぐために、各接続要求メッセージには生存時間が与えられてもよい。この生存時間の期限が切れると、フロートラッカーノードは要求を遮断し、要求を現在伝達することができないことをそれぞれの顧客１６０に通知してもよい。

【００３１】
少なくともいくつかの実施形態において、サーバノード１３０上のロードバランサモジュール１３２はまた、ロードバランサノード１１０への接続公開を行う。少なくともいくつかの実施形態において、接続公開を行うために、各ロードバランサモジュール１３２は定期的または非定期的に（例、１秒に１回）サーバノード１３０上のルーティングテーブル（例、ネットスタットルーティングテーブル）を確認し、アクティブな接続（ＴＣＰフ

ロー）のリストをロードバランサノード１１０に公開する。所与のパケットフローの存在について通知を受ける必要のあるロードバランサノード１１０は、それぞれのパケットフローのために入口ノードおよび１次および２次フロートラッカーとしての役割を担っているロードバランサノード１１０である。いくつかの実施形態においては、ロードバランサモジュール１３２は一貫したハッシュ技術を用いて、サーバノード１３０上のアクティブなＴＣＰフローについて通知を受ける必要のあるロードバランサノード１１０のリストをフィルターにかけてもよい。例えばロードバランサモジュール１３２は一貫したハッシュリングに従って、どのロードバランサノード１１０が所与のパケットフローのために１次および２次フロートラッカーの役割を担っているかを決定してもよい。いくつかの実施形態においては、ロードバランサモジュール１３２は各パケットフローのためにどのロードバランサノード１１０が最後にデータパケットをロードバランサモジュール１３２に送信したかをトラッキングし、この情報を用いてどのロードバランサノード１１０がパケットフローのための入口ノードの役割を担っているかを決定する。これは入口ノードのみが顧客データをロードバランサモジュール１３２に転送するためである。いくつかの実施形態においてはロードバランサモジュール１３２がその後、パケットフローについて通知を受ける必要があると決定したロードバランサノード１１０の各々のためのメッセージを作成し、メッセージをロードバランサノード１１０へ送信してそれぞれのサーバノード１３０が１つまたは複数の接続を１つまたは複数の顧客１６０に対してまだ維持していることをノード１１０に通知する。このロードバランサモジュール１３２によるロードバランサノード１１０への接続公開はリースのロードバランサノード１１０への延長と見なされてもよい。ロードバランサノード１１０が一定の時間（例、１０秒）内に特定のパケットフローを示す接続公開メッセージを受信しない場合、ロードバランサノード１１０はその後それぞれのパケットフローについて忘れることもできる。

ロードバランサノードへのマルチパスルーティング

【００３２】

図４は少なくともいくつかの実施形態による、分散型ロードバランサにおけるルーティングおよびパケットフローの態様を示す。少なくともいくつかの実施形態において、各入口ノード（入口ノードは図４において入口サーバ１１２として示される）が分散型ロードバランサのために、例えば境界ゲートウェイプロトコル（ＢＧＰ）を通じて１つまたは複数のパブリックエンドポイント（例、ＩＰアドレスおよびポート）をエッジルータ１０４へとルーティングする能力を提供する。少なくともいくつかの実施形態においては、各入口ノードがＢＧＰセッションを通じて自身をエッジルータ１０４に提供するのではなく、１つまたは複数の他の入口ノード、例えば２つの近傍ノードがエッジルータ１０４とのＢＧＰセッションを確立し、図５で示されるように入口ノードを提供してもよい。

【００３３】

従来のロードバランサは通常、単一のパブリックエンドポイントとしての役割しか担うことができない。それに対して、分散型ロードバランサの実施形態は複数のロードバランサノード１１０が単一のパブリックエンドポイントとしての役割を担うことを可能にする。これによりルータの能力に応じて、すべての入口サーバ１１２にルーティングされた単一のパブリックＩＰアドレスが１つまたは複数のエッジルータ１０４を通じて全帯域（例、１６０Ｇｂｐｓ）の処理を行ってもよい構成が可能になる。少なくともいくつかの実施形態においてこれを達成するために、１つまたは複数のエッジルータ１０４がレイヤー４のフローごとにハッシュ化されたマルチパスルーティング技術、例えば等価コストマルチパス（ＥＣＭＰ）ルーティング技術を利用し、各々が同一のパブリックＩＰアドレスを提供する複数の入口サーバ１１２を介してトラフィックを分散してもよい。一般的に、１つまたは複数のエッジルータ１０４のフローハッシュの一部としてフローのためのレイヤー４のソースおよび宛先ポートを用いてすべての入口サーバ１１２に受信パケットを分散することにより、入口サーバ１１２としての役割を担う同一のロードバランサノード１１０へとルーティングされた各接続のためのパケットに、故障したパケットを避けさせてもよい。しかしいくつかの実施形態においては、１つまたは複数のエッジルータ１０４は他の

技術を利用して入口サーバ１１２を通じてトラフィックを分散させてもよいことに留意する。

【００３４】

図４はまた、２つ以上の分散型ロードバランサがネットワーク１００上に実装されてもよいことを示す。２つ以上の分散型ロードバランサは、複数のサーバ１３０をフロントに配置し各々が異なるパブリックＩＰアドレスを提供するそれぞれ独立したロードバランサとして動作してもよく、またはその代わりに、図４に示されるように、２つ以上の分散型ロードバランサがそれぞれ同一のＩＰアドレスを提供してもよく、ハッシュ技術（例、レイヤー４のフローごとにハッシュ化されたマルチパスルーティング技術）は１つまたは複数の境界ルータ１０２において、次にパケットフローをそれぞれの区分する入口サーバ１１２に分散させるエッジルータ１０４からパケットフローを隔てるために用いられてもよい。

【００３５】

図５は少なくともいくつかの実施形態による、エッジルータへの入口ノードの提供のための境界ゲートウェイプロトコル（ＢＧＰ）の利用を示す。この実施例においては、ロードバランサの実装において入口ノード１１０Ａ～１１０Ｄの役割を担う、４つのロードバランサノードがある。エッジルータ１０４は顧客（図示せず）からの受信パケットをロードバランサノード１１０にルーティングする。少なくともいくつかの実施形態において、エッジルータ１０４はレイヤー４のフローごとにハッシュ化されたマルチパスルーティング技術、例えば等価コストマルチパス（ＥＣＭＰ）ルーティング技術に従って、ルーティングに関する決定を下してもよい。

【００３６】

少なくともいくつかの実施形態において、エッジルータ１０４は、ロードバランサの実装において、入口ノード１１０により開始されるセッションを提供する境界ゲートウェイプロトコル（ＢＧＰ）技術を通じて顧客のトラフィックを受信することが現在可能である入口ノード１１０に関して把握する。各入口ノード１１０はＢＧＰを用いて自身をエッジルータ１０４に提供することができる。しかし、ＢＧＰは通常、収束に比較的長い時間がかかる（３秒以上）。各入口ノード１１０がＢＧＰを介して自身を提供するこの技術を用いる場合、入口ノード１１０の故障時には、ネットワーク時間（３秒以上）において、エッジルータ１０４上のＢＧＰセッションがタイムアウトするまでに相当な時間がかかる可能性があり、したがって、エッジルータ１０４が失敗による終了について把握し現在のＴＣＰフローを入口ノード１１０に再度ルーティングするまでには相当な時間がかかる可能性がある。

【００３７】

ＢＧＰの収束問題を回避し、ノード１１０の故障時の回復を早めるため、少なくともいくつかの実施形態において、入口ノード１１０がＢＧＰセッションを介して自身をエッジルータ１０４に提供する代わりに、ロードバランの実装において少なくとも１つの他の入口ノード１１０がＢＧＰを通じてその入口ノード１１０をエッジルータ１０４に提供する役割を担う。例えば図５に示されるいくつかの実施形態においては、所与の入口ノード１１０の左右の近傍入口ノード１１０、例えばノード１１０の番号付きリストにおける左右の近傍ノードや、例えばノード１１０により形成された一貫したハッシュリングが、所与の入口ノード１１０をエッジルータ１０４に提供してもよい。例えば図５では、入口ノード１１０Ａが入口ノード１１０Ｂおよび１１０Ｄを提供し、入口ノード１１０Ｂが入口ノード１１０Ａおよび１１０Ｃを提供し、入口ノード１１０Ｃが入口ノード１１０Ｂおよび１１０Ｄを提供し、そして入口ノード１１０Ｄが入口ノード１１０Ｃおよび１１０Ａを提供する。入口ノード１１０は本明細書にて後述するように、互いのヘルス状態をチェックしゴシップする。記載のヘルスチェック方法を用いて、異常なノードを検知することができ、１秒以内、例えば１００ミリ秒（ｍｓ）以内にノード１１０間に情報を伝播することができる。入口ノード１１０が正常でないと決定された際には、異常なノードを提供する入口ノード１１０は直ちに異常なノード１１０の提供を停止してもよい。少なくともいく

つかの実施形態において、入口ノード１１０がＴＣＰクローズまたはＢＧＰセッションのための類似のメッセージをエッジルータ１０４に送信することによりエッジルータ１０４とのＢＧＰセッションを終了する。したがって、故障したノード１１０により確立されたＢＧＰセッションがノード１１０の故障の検知についてタイムアウトするのを待つ必要なく、故障したノード１１０の代わりに提供を行う他の入口ノード１１０が、ノード１１０の異常の検知時にノード１１０を提供するエッジルータ１０４とのＢＧＰセッションを遮断する時に、エッジルータ１０４が故障したノード１１０を発見することができる。ロードバランサノードの故障の処理については図１８Ａおよび１８Ｂに関連して本明細書にてさらに後述される。

【００３８】

図６は分散型ロードバランスシステムの少なくともいくつかの実施形態による、マルチパスルーティング方法のフローチャートである。９００で示すように、ロードバランサの実装における入口ノード１１０がその近傍ノード１１０をエッジルータ１０４に提供する。少なくともいくつかの実施形態において、入口ノード１１０は一貫したハッシュリングのようなノード１１０の番号付きリストに従って、その近傍ノード１１０を決定してもよい。少なくともいくつかの実施形態において、入口ノード１１０はＢＧＰセッションを用いてその１つまたは複数の近傍ノード１１０をエッジルータ１０４に提供する。それらＢＧＰセッションのうち１つは、提供されるノード１１０の各々のために確立されたエッジルータ１０４へのＢＧＰセッションである。

【００３９】

９０２で示すように、エッジルータ１０４はフローごとにハッシュ化されたマルチパスルーティング技術、例えば等価コストマルチパス（ＥＣＭＰ）ルーティング技術に従って、顧客１６０から受信したトラフィックをアクティブな（提供された）入口ノード１１０に分散させる。少なくともいくつかの実施形態において、エッジルータ１０４はパブリックＩＰアドレスを顧客１６０に公開し、すべての入口ノード１１０が同一のパブリックＩＰアドレスをエッジルータ１０４に提供する。エッジルータレイヤー４のソースおよび宛先ポートをエッジルータ１０４のフローハッシュの一部として用い、受信パケットを入口ノード１１０間に分散させる。一般的に、これによって各接続のためのパケットを同一の入口ノード１１０に分散させる。

【００４０】

９０２で示すように、入口ノードがデータフローを対象のサーバノード１３０に転送する。少なくともいくつかの実施形態において、入口ノード１１０はデータフローのための１次および２次フロートラッカーノードと対話し、データフローを対象のサーバノード１３０へとマッピングする。こうして各入口ノード１１０は、受信されたパケットを対象のサーバノード１３０へと適切に転送するノード１１０を通じ、アクティブなデータフローのマッピングを維持してもよい。

【００４１】

要素９０６～９１０は入口ノード１１０の故障の検知およびそこからの回復に関する。９０６で示すように、例えば本明細書に記載のヘルスチェック技術に従って入口ノード１１０は入口ノード１１０のダウンを検知してもよい。ノード１１０のダウンの検知時には、近傍ノード１１０がそのノード１１０のエッジルータ１０４への提供を停止する。少なくともいくつかの実施形態において、これにはそれぞれのＢＧＰセッションのためのエッジルータ１０４へのＴＣＰクローズの送信が関与する。

【００４２】

９０８で示すように、入口ノード１１０のダウンをＢＧＰセッションの終了を通じて検知する際に、エッジルータ１０４はフローごとにハッシュ化されたマルチパスルーティング技術に従い、顧客１６０の受信トラフィックを残りの入口ノード１１０に再分散させる。したがって、少なくともいくつかのデータフローを入口ノード１１０にルーティングしてもよい。

【００４３】

９１０で示すように、入口ノード１１０は必要に応じてマッピングを回復し、データフローを適切な対象のサーバノードに転送してもよい。入口ノード１１０におけるノード１１０の故障からの回復方法については、本明細書の別の部分でも論じる。１つの実施例として、入口ノード１１０は、そのための現在のマッピングがないパケットの受信の際に、一貫したハッシュリングに従ってデータフローのためのフロートラッカーノードを決定し、フロートラッカーノードからマッピングを回復するために、一貫したハッシュ関数を用いてもよい。

非対称パケットフロー

【００４４】

少なくともいくつかの実施形態において、アウトバウンドトラフィックとインバウンドデータとの比が１より大きい場合に入口ノードの帯域およびＣＰＵの使用率を効率的に利用するために、分散型ロードバランスシステムは図７で示すようにアウトバウンドパケットをサーバノード１３０から複数の出口ノードへと転送する。少なくともいくつかの実施形態において、各接続のために、それぞれのサーバノード１３０上のロードバランサモジュール１３２が顧客エンドポイント／パブリックエンドポイントのタプルをハッシュし、それぞれのアウトバウンドパケットフローのための出口サーバ１１４の役割を担うロードバランサノード１１０を選択するために、一貫したハッシュアルゴリズムを用いる。しかし、いくつかの実施形態においては、接続のために出口サーバ１１４を選択するために、他の方法および／またはデータが用いられてもよい。選択した出口サーバ１１４は必ずではないが、通常は接続のための入口サーバ１１２の役割を担うロードバランサノード１１０とは異なるロードバランサノード１１０である。少なくともいくつかの実施形態において、そのロードバランサノード１１０／出口サーバ１１４に故障がない限りは、特定の接続のためのすべてのアウトバウンドパケットは故障したパケットを回避するために、同一の出口サーバ１１４に転送される。

【００４５】

少なくともいくつかの実施形態において、出口サーバ１１４の選択のためにサーバノード１３０によって用いられる方法およびデータは、１つまたは複数のエッジルータ１０４により行われる入口サーバ１１２の選択に用いられる方法およびデータとは異なってもよい。一般的には異なる方法およびデータの利用により、所与の接続のための出口ノードには、その接続のために入口ノードとして選択されたロードバランサノード１１０とは異なるロードバランサノード１１０が結果的に選択されてもよく、また複数のロードバランサノード１１０が、入口ノードの役割を担う単一のロードバランサノード１１０を通過する接続のための発信トラフィックを処理する出口ノードとして結果的に選択されてもよい。

【００４６】

図７は少なくともいくつかの実施形態による、非対称パケットフローを図示する。外部ネットワーク１５０上の顧客１６０から入口サーバ１１２を通り、サーバノード１３０Ａ、１３０Ｂ、１３０Ｃおよび１３０Ｄのそれぞれに至る接続が少なくとも１つは確立されている。少なくともいくつかの実施形態において、接続のための出口ノードを選択するため、各接続についてそれぞれのサーバノード１３０上のロードバランサモジュール１３２が、顧客エンドポイント／パブリックエンドポイントのタプルをハッシュし、それぞれのアウトバウンドパケットフローのための出口サーバ１１４の役割を担うロードバランサノード１１０を選択するために一貫したハッシュアルゴリズムを利用する。例えばサーバノード１３０Ａが接続のために出口サーバ１１４Ａを選択し、サーバノード１３０Ｂがある接続のために出口サーバ１１４Ａを、また別の接続のために出口サーバ１１４Ｂを選択している。しかし、いくつかの実施形態においては他の方法および／またはデータが接続のための出口ノードの選択に用いられてもよい。

顧客接続を破棄しないロードバランサノードの故障からの回復

【００４７】

どのサーバノード１３０が顧客のトラフィックを受信するべきか決定するためにロードバランサノード１１０は一貫したハッシュを用いることができるが、いくつかの接続は寿

命が長いために、この手法では、新規サーバノード１３０が一貫したハッシュメンバーシップに参加し、入口ロードバランサノード１１０の故障が続いて起こる場合に既存のフローを維持できない可能性がある。この場合、サーバ１３０のための一貫したハッシュリングが異なるメンバーシップを有することになるため、故障したノード１１０からのフローを引き継ぐロードバランサノード１１０はもともと選択したマッピングを決定することができない可能性がある。したがって少なくともいくつかの実施形態においては、接続のためのサーバノード１３０を選択し、選択したサーバノード１３０にパケットをルーティングするために、ロードバランサノード１１０が分散型ハッシュテーブル（ＤＨＴ）技術を用いてもよい。ＤＨＴに従って、サーバノード１３０が特定の接続を受信するために選択された場合に、サーバノード１３０が正常を保ち、サーバノード１３０上のロードバランサモジュール１３２がそのアクティブな接続の状態を（例、接続公開を介して）ＤＨＴに定期的に伝達することでリースを継続的に延長すると仮定すると、ＤＨＴは接続が完了するまでマッピングを保持する。入口ノード１１０の故障はエッジルータ１０４から残りのロードバランサノード１１０へのパケットの分散に影響を及ぼし、その結果、ロードバランサノード１１０は異なる顧客接続の組からトラフィックを受信することになる。しかしＤＨＴはすべてのアクティブな接続をトラッキングするので、アクティブなマッピングのいずれかのリースを取得するためにロードバランサノード１１０がＤＨＴに問い合わせを行うことは可能である。その結果、すべてのロードバランサノード１１０がトラフィックを正しいサーバノード１３０へと渡し、それによって入口ロードバランサノード１１０の故障時であってもアクティブな顧客接続の故障を防ぐ。

分散型ロードバランスシステムにおけるパケットフロー

【００４８】

図８は少なくともいくつかの実施形態による、分散型ロードバランスシステムにおけるパケットフローを示す。図８において実線矢印はＴＣＰデータパケットを表し、破線矢印はＵＤＰデータパケットを表すことに留意する。図８では、入口サーバ１１２が１つまたは複数の顧客１６０からエッジルータ１０４を通じてＴＣＰデータパケットを受信する。ＴＣＰパケットの受信時に、入口サーバ１１２はＴＣＰパケットフローのためのサーバノード１３０へのマッピングを有しているかどうかを判断する。入口サーバ１１２がＴＣＰパケットフローのためのマッピングを有する場合には、サーバ１１２はその後ＴＣＰパケットを（例えばＵＤＰに従って）カプセル化し、カプセル化されたパケットを対象のサーバノード１３０へと送信する。入口サーバ１１２がＴＣＰパケットフローのためのマッピングを有していない場合には、入口サーバ１１２はその後、サーバノード１３０への接続を確立するために、さらに／またはＴＣＰパケットフローのためのマッピングを取得するために、ＴＣＰパケットから抽出されたＴＣＰパケットフローに関する情報を含むＵＤＰメッセージを１次フロートラッカー１１６Ａへと送信してもよい。図９Ａおよび９Ｂならびに図１０Ａ～１０Ｇは、顧客１６０とサーバノード１３０との間の接続の確立方法を示す。サーバノード１３０上のロードバランサモジュール１３２は、サーバノード１３０上の１つまたは複数のＴＣＰ接続のための１つまたは複数の出口サーバ１１４の役割を担う１つまたは複数のロードバランサノード１１０を無作為に選択し、１つまたは複数の出口サーバ１１４を通じてＵＤＰによりカプセル化されたＴＣＰ応答データパケットを１つまたは複数の顧客１６０へと送信する。

【００４９】

図９Ａおよび９Ｂは少なくともいくつかの実施形態による、分散型ロードバランスシステムにおいて接続を確立する際のパケットフローのフローチャートを提示する。図９Ａの２００で示すように、入口サーバ１１２は顧客１６０のＴＣＰパケットを、エッジルータ１０４を通じて受信する。２０２で入口サーバ１１２がＴＣＰフローのためのサーバノード１３０へのマッピングを有する場合、２０４で示すように、入口サーバ１１２がその後ＴＣＰパケットをカプセル化し、それぞれのサーバノード１３０へと送信する。入口サーバ１１２は継続的に、２つ以上の顧客１６０から２つ以上のＴＣＰフローのためのパケットを受信し処理してもよいということに留意する。

【００５０】
２０２において入口サーバ１１２がＴＣＰフローのためのマッピングを有していない場合、パケットは顧客１６０からのＴＣＰ同期（ＳＹＮ）パケットであってもよい。２０６で示すように、ＳＹＮパケットの受信時には、入口サーバ１１２がＳＹＮパケットからデータを抽出し、例えばＵＤＰメッセージにおいて、データを１次フロートラッカー１１６Ａへと転送する。少なくともいくつかの実施形態において、入口サーバ１１２はＴＣＰフローのための１次フロートラッカー１１６Ａおよび／または２次フロートラッカー１１６Ｂを一貫したハッシュ関数に従って決定することができる。２０８では、１次フロートラッカー１１６Ａがデータを例えばハッシュテーブル内に格納し、ＴＣＰ接続のサーバノード１３０側のための最初のＴＣＰシーケンス番号を生成し、データおよびＴＣＰシーケンス番号を２次フロートラッカー１１６Ｂへと転送する。２１０では、２次フロートラッカー１１６Ｂもデータを格納してもよく、少なくともＴＣＰシーケンス番号を含むＳＹＮ／ＡＣＫパケットを生成して顧客１６０に送信する。
【００５１】
２１２で示すように、入口サーバ１１２は顧客１６０からＴＣＰ受信確認（ＡＣＫ）パケットを、エッジルータ１０４を通じて受信する。入口サーバ１１２はこの時点ではＴＣＰフローのためのサーバ１３０ノードへのマッピングは有していないため、２１４で入口サーバ１１２がＡＣＫパケットから抽出されたデータを含むメッセージを１次フロートラッカー１１６Ａへと送信する。２１６で示すように、メッセージの受信時に１次フロートラッカー１１６Ａは格納されたデータに従ってＴＣＰフローを確認し、承認されたＡＣＫパケットからのシーケンス番号（＋１）がＳＹＮ／ＡＣＫにおいて送信された数値に一致することを確認する。１次フロートラッカー１１６Ａはその後、ＴＣＰフローを受信するサーバノード１３０を選択し、選択したサーバノード１３０上のローカルロードバランサモジュール１３２のデータ、ＴＣＰシーケンス番号およびＩＰアドレスを含むメッセージを２次フロートラッカー１１６Ｂに送信する。２１８で示すように、２次フロートラッカー１１６Ｂもまたデータおよびシーケンス番号を確認し、ＳＹＮメッセージを生成し、生成されたＳＹＮメッセージを選択したサーバノード１３０上のローカルロードバランサモジュール１３２へと送信する。メソッドは図９Ｂの要素２２０に続く。
【００５２】
図９Ｂの２２０で示すように、ロードバランサモジュール１３２は生成されたＳＹＮメッセージに応じて、サーバノード１３０の１つまたは複数の測定基準を調べてサーバノード１３０が接続を受け入れることが可能かどうかを決定してもよい。２２２においてサーバノード１３０が接続を現在受け入れることができないとロードバランサモジュール１３２が決定した場合、その後２２４においてロードバランサモジュール１３２が２次フロートラッカー１１６Ｂへの伝達を行う。２次フロートラッカー１１６Ｂは、すでに格納したフローのための情報を削除してもよい。２２６において、２次フロートラッカー１１６Ｂは１次フロートラッカー１１６Ａへの伝達を行う。図９Ａの２１６で示されるように１次フロートラッカー１１６Ａはその後、新規対象のサーバノード１３０を選択し、２次フロートラッカー１１６Ｂへの伝達を行ってもよい。
【００５３】
２２２では、サーバノード１３０が接続を受け入れることが可能であるとロードバランサモジュール１３２が決定した場合、その後図９Ｂの２２８で示すように、ローカルロードバランサモジュール１３２は生成されたＳＹＮからＴＣＰ　ＳＹＮパケットを構成し、ＴＣＰ　ＳＹＮパケットをサーバノード１３０上のサーバ１３４に送信する。ＴＣＰ　ＳＹＮパケットのソースＩＰアドレスに顧客１６０の実際のＩＰアドレスが取り込まれ、それによりサーバ１３４は、顧客１６０への直接ＴＣＰ接続を受信したことを知る。ロードバランサモジュール１３２はＴＣＰフローに関連する詳細を、例えばローカルハッシュテーブルに格納する。２３０で示すように、サーバ１３４はロードバランサモジュール１３２が遮断するＳＹＮ／ＡＣＫパケットで応答する。２３２で示すように、ロードバランサモジュール１３２はその後接続情報を含むメッセージを２次フロートラッカー１１６Ｂに

送信し、接続が受け入れられたことを示す。このメッセージの受信時に、２３４において２次フロートラッカー１１６Ｂがサーバ１３４へのマッピングを記録し、類似のメッセージを１次フロートラッカー１１６Ａに送信し、１次フロートラッカー１１６Ａもまたマッピング情報を記録する。２３６で示すように、その後１次フロートラッカー１１６Ａがマッピングメッセージを入口サーバ１１２に転送する。入口サーバ１１２はそうして顧客１６０からサーバ１３０へのＴＣＰフローのためのマッピングを有する。

【００５４】

２３８において入口サーバ１１２はサーバノード１３０上のロードバランサモジュール１３２へのデータフローのためのいずれかのバッファされたデータパケットをカプセル化し転送する。入口サーバ１１２により顧客１６０から受信されたデータのための追加の受信データパケットはカプセル化され、ロードバランサモジュール１３２へと直接転送され、ロードバランサモジュール１３２がデータパケットを脱カプセル化し、サーバ１３４へと送信する。

【００５５】

２４０において、ロードバランサモジュール１３２はデータフローのための出口サーバ１１４を無作為に選択する。後続のサーバ１３４からのアウトバウンドＴＣＰデータパケットはロードバランサモジュール１３２により遮断され、ＵＤＰに従ってカプセル化され、任意に選択された出口サーバ１１４へと転送される。出口サーバ１１４は発信パケットを脱カプセル化し、ＴＣＰデータパケットを顧客１６０に送信する。

【００５６】

上記の通り２０２において、入口サーバ１１２が受信されたパケットのＴＣＰフローのためのマッピングを有していない場合、パケットは顧客１６０からのＴＣＰ同期（ＳＹＮ）パケットであってもよい。しかし、パケットはＴＣＰ　ＳＹＮパケットでなくてもよい。例えばロードバランサノード１１０のメンバーシップがロードバランサノード１１０の追加や故障により変更される場合、エッジルータ１０４は入口サーバ１１２がマッピングを有しない入口サーバ１１２へのＴＣＰフローのためのデータパケットのルーティングを開始してもよい。少なくともいくつかの実施形態において、入口サーバ１１２がマッピングを有しないこのようなパケットの受信時には、入口サーバ１１２は、一貫したハッシュリングに従ってＴＣＰフローのための１次フロートラッカー１１６Ａおよび／または２次フロートラッカー１１６Ｂを決定するために一貫したハッシュ関数を用い、マッピングを要求するために１次フロートラッカー１１６Ａあるいは２次フロートラッカー１１６Ｂのいずれかへの伝達を行ってもよい。ＴＣＰフローのためのフロートラッカー１１６へのマッピング受信時には、入口サーバ１１２はマッピングを格納し、ＴＣＰフローのための１つまたは複数のＴＣＰパケットのカプセル化および正しい宛先サーバノード１３０への転送を開始することができる。

ロードバランサノードの説明

【００５７】

少なくともいくつかの実施形態において、おのおののロードバランサノード１１０は３つの役割を有する：

＊ 入口－顧客接続における顧客１６０からのすべての受信パケットの受信、マッピングが把握されている場合のデータパケットのサーバノード１３０へのルーティング、またはマッピングが把握されていない場合のフロートラッカーへの伝達。入口ノードからの発信パケットは（例、ＵＤＰに従って）入口ノードによりカプセル化される。

＊ フロートラッキング－接続状態（例えば、各顧客接続を伝達するためにどのサーバノード１３０／サーバ１３４が割り当てられているか）の継続的な追跡。フロートラッカーも顧客１６０とサーバ１３４との間の接続の確立に参加する。

＊ 出口－サーバ１３４から受信されたアウトバウンドパケットの脱カプセル化および顧客１６０への転送。

【００５８】

少なくともいくつかの実施形態において、顧客－＞サーバのマッピングが把握されてい

る場合に、ロードバランサノード１１０は入口の役割として、サーバ１３４へのパケットの転送をし、マッピングが把握されていない場合に、要求のフロートラッカーへの転送の役割を担う。少なくともいくつかの実施形態において、特定の顧客接続／データフローのための入口ノードの役割を担うロードバランサノード１１０はまた、顧客接続のための１次フロートラッカーまたは２次フロートラッカーのいずれかの役割も担ってもよいが、その両方の役割を果たすことはできない。

【００５９】

少なくともいくつかの実施形態において、ロードバランサノード１１０はフロートラッカーの役割として、すでに確立された接続の顧客－＞サーバのマッピングの維持と同様にいまだ確立されつつある接続状態の維持の役割を担う。１次フロートラッカーおよび２次フロートラッカーと呼ばれる２つのフロートラッカーは個別の各顧客接続に関与する。少なくともいくつかの実施形態において、顧客接続に関するフロートラッカーは一貫したハッシュアルゴリズムを用いて決定されてもよい。フロートラッカーはまた、各新規顧客接続のためのサーバノード１３０の擬似ランダム選択を含むがそれに限定されないロードバランス機能も果たす。選択されたサーバノード１３０上のローカルロードバランサモジュール１３２は、サーバ１３４が接続を処理することができないと決定された場合に接続要求を拒否してもよいことに留意する。この場合、フロートラッカーはその後別のサーバノード１３０を選択し接続要求を他のサーバノード１３０に送信してもよい。少なくともいくつかの実施形態において、所与の接続のための１次フロートラッカーの役割および２次フロートラッカーの役割は異なるロードバランサノード１１０によって果たされる。

【００６０】

少なくともいくつかの実施形態において、ロードバランサノード１１０は出口の役割として、ステートレスであり、サーバノード１３０から受信された受信パケットを脱カプセル化し、いくつかの検証を行い、アウトバウンドＴＣＰデータパケットをそれぞれの顧客１６０に転送する。少なくともいくつかの実施形態において、サーバノード１３０上のロードバランサモジュール１３２は所与の接続のためのロードバランサノード１１０を任意に選択してもよい。

ロードバランサノード一貫したハッシュリングトポロジ

【００６１】

少なくともいくつかの実施形態において、ロードバランサノード１１０は入力キー空間（顧客エンドポイント、パブリックエンドポイント）の一貫したハッシュに基づくリングトポロジを形成する。入力キー空間は利用可能なフロートラッカーノードの間で区分されてもよく、すべてのフロートラッカーノードがそのキー空間に応じた質問への回答の役割を担ってもよい。少なくともいくつかの実施形態において、一貫したハッシュリングにおける後続処理に基づいて（例、２次フロートラッカーノードが１次フロートラッカーノードへの後続ノードであるか、または一貫したハッシュリングにおいて次のノードである）、データは１次および２次フロートラッカーノードに複製されてもよい。フロートラッカーノードが何らかの理由によりダウンした場合、一貫したハッシュリングにおける次のロードバランサノードが故障したノードのキー空間を獲得する。新規フロートラッカーノードが参加する際には、ノードは（例、図１で示すように構成サービス１２２を用いて）そのエンドポイントを登録し、それにより他のロードバランサノードはロードバランサの実装における構成変化および、その結果の一貫したハッシュリングにおける構成変化を把握してもよい。一貫したハッシュリングにおけるフロートラッカーの追加および故障の処理については、図１１Ａ～１１Ｄに関連して詳細に後述される。

入口ノード＜－＞フロートラッカーノードの通信

【００６２】

少なくともいくつかの実施形態において、入口ノードの役割を担うロードバランサノード１１０は、構成サービス１２２からフロートラッカーノードの役割を担うロードバランサノード１１０について把握してもよい。入口ノードは、ロードバランサの実装において変更され、その結果の一貫したハッシュリングにおいても変更されたメンバーシップのた

めの構成サービス１２２を監視してもよい。入口ノードがマッピングを有しない顧客１６０からのパケットを受信する際には、どのフロートラッカーノードがパケットを伝達すべきか決定するために、入口ノードは一貫したハッシュ関数を用いてもよい。少なくともいくつかの実施形態において、ハッシュ関数への入力はパケットからの（顧客エンドポイント、パブリックエンドポイントの）ペアである。少なくともいくつかの実施形態において、入口ノードおよびフロートラッカーノードは、ＵＤＰメッセージを用いた通信を行う。

【００６３】

１次フロートラッカーノードが入口ノードから新規パケットフローのためのメッセージを受信する際には、１次フロートラッカーノードが無作為にＴＣＰシーケンス番号を決定し、別のメッセージを２次フロートラッカーノードに転送する。２次フロートラッカーノードは顧客のためのＴＣＰ　ＳＹＮ／ＡＣＫメッセージを生成する。両方のフロートラッカーが顧客接続のエンドポイントのペアおよびＴＣＰシーケンス番号を記憶し、メモリ圧迫または期限により状態が消去されるまでこの情報を保持する。

【００６４】

ＴＣＰ　ＡＣＫパケットが受信された入口ノードから１次フロートラッカーノードがメッセージを受信する際には、１次フロートラッカーノードが、承認されたＴＣＰシーケンス番号がＳＹＮ／ＡＣＫパケットにおいて送信済みの格納された数値と一致することを検証し、要求を伝達するサーバノード１３０を選択し、メッセージを２次フロートラッカーノードに転送する。２次フロートラッカーノードは、サーバノード１３０でＴＣＰスタックを用いた実際のＴＣＰ接続を開始するためにメッセージを選択されたサーバノード１３０上のロードバランサモジュール１３２へと送信し、その後サーバノード１３０からの確認応答を待つ。

【００６５】

２次フロートラッカーノードがサーバノード１３０上のロードバランサモジュール１３２から接続確認を受信する際に、１次フロートラッカーを通って入口ノードに至るリバースメッセージフローが開始され、それにより両方のフローにおいてサーバノード１３０に関連する情報が格納される。この時点より、入口ノードにおいて受信された追加のＴＣＰデータパケットがサーバノード１３０上のロードバランサモジュール１３２に直接転送される。

ロードバランサモジュール＜－＞ロードバランサノードの通信

【００６６】

少なくともいくつかの実施形態において、すべてのロードバランサモジュール１３２が構成サービス１２２を用いてそのエンドポイントを登録し、ロードバランサノードレイヤー内のメンバーシップ変更のために構成サービス１２２を継続的に監視する。少なくともいくつかの実施形態による、ロードバランサモジュール１３２の関数が以下に記載される：

＊ 接続公開－定期的に（例、１秒に１回）または非定期的に、それぞれのサーバノード１３０上のアクティブな接続（顧客エンドポイント、パブリックエンドポイント）の組を、それら接続のために最後にパケットをロードバランサモジュール１３２へと送信した入口ノードへと同様に、それら接続のための役割を担う１次および２次フロートラッカーノードの両方にも公開する。接続公開関数はそのための役割を担うロードバランサノード１１０において、接続状態のリースを更新する。

＊ ロードバランサレイヤーにおけるメンバーシップ変更の監視。メンバーシップが変更された際に、ロードバランサモジュール１３２はこの変更情報を用いて、接続のための役割を担うロードバランサノードへとアクティブな接続を直ちに送信してもよい。

分散型ロードバランスシステムにおけるパケットフロー－詳細

【００６７】

分散型ロードバランスシステムは複数のロードバランサノード１１０を含んでもよい。少なくともいくつかの実施形態において、分散型ロードバランスシステムにおける各ロードバランサノード１１０はサーバ１３４への顧客１６０の接続のために、フロートラッカ

ーノード、出口ノード、および入口ノードの役割を担ってもよい。分散型ロードバランスシステムはまた、各サーバノード１３０上のロードバランサモジュール１３２を含んでもよい。

【００６８】

図１０Ａ～１０Ｇは、少なくともいくつかの実施形態による、分散型ロードバランスシステムにおけるパケットフローを示す。図１０Ａ～１０Ｇにおいて、ロードバランサノード１１０の間で交換されるパケットと、ロードバランサノード１１０とサーバノード１３０との間で交換されるパケットは、ＵＤＰメッセージまたはＵＤＰによりカプセル化された顧客ＴＣＰパケットのいずれかである。少なくともいくつかの実施形態において、顧客ＴＣＰパケットはネットワーク１００に脱カプセル化された形式で、ロードバランサノード１１０の北側で境界ルータ１０２との間の相互の移行（図１参照）においてのみ存在する。図１０Ａ～１０Ｇの実線矢印はＴＣＰパケットを表し、破線矢印はＵＤＰパケットを表すことに留意する。

【００６９】

少なくともいくつかの実施形態において、単一のロードバランサノード１１０の故障時に、分散型ロードバランスシステムは確立された接続の保持を試みてもよい。少なくともいくつかの実施形態において、１次フロートラッカーノードまたは２次フロートラッカーノードのいずれかの故障時に、接続の顧客－＞サーバのマッピングが残りのフロートラッカーノードにより格納できるように、１次フロートラッカーノードおよび２次フロートラッカーノードにおける接続の詳細の複製を行うことで、これは達成されてもよい。少なくともいくつかの実施形態において、ノードの故障時にはパケットの一部が喪失される可能性があるが、顧客／サーバＴＣＰパケットの再移行により、喪失されたパケットが回復する可能性もある。

【００７０】

顧客からの各ＴＣＰ接続は、ＴＣＰフローとよばれてもよく、顧客のＩＰアドレス、顧客用ポート、サーバ（パブリック）ＩＰアドレス、およびサーバポートから成る４タプルによって一意に識別される。この識別子は、顧客およびパブリックエンドポイントのペアを示すＣＰまたはＣｃＰｐとして略されてもよい。与えられた任意のＴＣＰフロー（またはＣＰペア）に関連するパケットは、上流のエッジルータ１０４からのハッシュ化された等価コストマルチパス（ＥＣＭＰ）フロー分散により、入口サーバ１１２として動作するどのロードバランサノード１１０上にも表れることができる。しかし一般的にＴＣＰフローのためのパケットはＴＣＰフローの方向を変えるリンクやロードバランサノード１１０の故障がない限りは、同一のロードバランサノード１１０に到着し続けてもよい。上流のルータ１０４からＴＣＰフローのためのパケットを受信するロードバランサノード１１０はＴＣＰフローのための入口ノードと称される。

【００７１】

少なくともいくつかの実施形態において、パケットがＴＣＰフローのための入口ノードの役割を担うロードバランサノード１１０に到着した際に、どのロードバランサノード１１０がＴＣＰフローのための状態を含むか（すなわち、フロートラッカーノード）を入口ノードが決定できるように、一貫したハッシュが用いられる。ＴＣＰフローに関する状態の維持の役割をどのロードバランサノード１１０が担うか決定するために、ＣＰペアは入口ノードによって一貫したハッシュリングへとハッシュされてもよい。このノードは、ＴＣＰフローのための１次フロートラッカーの役割を担う。一貫したハッシュリングにおける後続ノードは、ＴＣＰフローのための２次フロートラッカーの役割を担う。

【００７２】

少なくともいくつかの実施形態において、すべてのロードバランサノード１１０は、入口ノード、１次フロートラッカーノード、および２次フロートラッカーノードとしての役割を担ってもよい。ＴＣＰフローのための一貫したハッシュの結果に応じて、ＴＣＰフローのための入口ノードの役割を担うロードバランサノード１１０はまた、ＴＣＰフローのための１次または２次フロートラッカーノードの役割を担ってもよい。しかし、少なくと

もいくつかの実施形態において、異なる物理ロードバランサノード１１０がＴＣＰフローのための１次および２次フロートラッカーの役割を果たす。

接続の確立

【００７３】

図１０Ａを参照すると、顧客１６０からの新規接続は顧客ＴＣＰ同期（ＳＹＮ）パケットによって起こってもよい。ロードバランサノード１１０は実際にはＳＹＮパケットの受信時にサーバノード１３０との接続を確立せず、また、接続を受信するためのサーバノード１３０の選択も直ちには行わない。代わりに、ロードバランサノード１１０は顧客のＳＹＮパケットからの関連データを格納し、まだ未選択のサーバノード１３０のためにＳＹＮ／ＡＣＫパケットを生成する。図１０Ｃを参照すると、顧客１６０がＴＣＰのスリーウェイハンドシェイクにおける第１のＡＣＫパケットで応答すると、ロードバランサノード１１０はサーバノード１３０を選択し、サーバノード１３０のための同等のＳＹＮパケットを生成し、サーバノード１３０を用いて実際のＴＣＰ接続の確立を試みる。

【００７４】

図１０Ａを再度参照すると、ＴＣＰフローのための入口サーバ１１２の役割を担うロードバランサノード１１０における顧客ＳＹＮパケットの受信時には、入口サーバ１１２がＳＹＮパケットからデータフィールドを抽出し、ＴＣＰフローのための１次フロートラッカー１１６Ａにデータを転送する。１次フロートラッカー１１６Ａは例えばハッシュテーブル内にデータを格納し、最初のＴＣＰシーケンス番号（ＴＣＰ接続のサーバ側のための）を生成し、同一のデータを次フロートラッカー１１６Ｂへ転送する。２次フロートラッカー１１６Ｂは、サーバのＴＣＰシーケンス番号を含む顧客１６０のためのＳＹＮ／ＡＣＫパケットを生成する。

【００７５】

図１０Ａでは、入口サーバ１１２、１次フロートラッカー１１６Ａ、２次フロートラッカー１１６Ｂの役割が異なるロードバランサノード１１０により果たされる。しかし場合によっては、ＴＣＰフローのための入口サーバ１１２の役割を担うロードバランサノード１１０は、ＴＣＰフローのための１次フロートラッカー１１６Ａまたは２次フロートラッカー１１６Ｂいずれか（しかし両方ではない）の役割を担う同一のノード１１０であってもよい。パケットフローのための入口サーバ１１２がフローのためのフロートラッカー１１６と同一のノード１１０上にあってもよい理由は、パケットフローのためのフロートラッカー１１６がパケットフローのアドレス情報に適用される一貫したハッシュ関数に従って一貫したハッシュリング上で決定される一方で、エッジルータ１０４がフローごとにハッシュ化されたマルチパスルーティング技術（例、ＥＣＭＰルーティング技術）に従ってフローのための入口サーバ１１２を擬似ランダムに選択するからである。パケットフローのための入口サーバ１１２は、パケットフローのためのフロートラッカー１１６と同一のノード１１０上にある場合、ＳＹＮパケットからのデータは、入口サーバ１１２を実装するノード１１０から他のフロートラッカー１１６ノード１１０へのみ転送されてもよい。例えば図１０Ｂにおいて、１次フロートラッカー１１６ＡはＴＣＰフローのための入口サーバ１１２と同一のロードバランサノード１１０Ａ上にあるが２次フロートラッカー１１６Ｂは異なるロードバランサノード１１０Ｂ上にあり、したがってＳＹＮパケットからのデータがノード１１０Ａから（フロートラッカー１１６Ａによって）ロードバランサノード１１０Ｂ上の２次フロートラッカー１１６Ｂへと転送される

【００７６】

図１０Ｃを参照すると、非ＳＹＮパケットが入口サーバ１１２に到着した際、入口サーバ１１２はどのサーバノード１３０がパケットの転送先となるべきかを把握しているか把握していないかのいずれかである。ＴＣＰフローのための入口サーバ１１２に到着する第１の非ＳＹＮパケットは、ＴＣＰ確認番号フィールドが図１０ＡにおいてＳＹＮ／ＡＣＫパケットが送信されたサーバシーケンス番号（＋１）に一致するＴＣＰスリーウェイハンドシェイクにおける第１のＴＣＰ受信確認（ＡＣＫ）パケット（または後続のデータパケットである可能性もある）であるべきである。入口サーバ１１２がサーバマッピングを有

しない非ＳＹＮパケットを受信した際には、入口サーバ１１２はシーケンス番号といったＡＣＫパケットからの情報を含む、またはその代わりにＡＣＫパケット自体を含むメッセージをＴＣＰフローのための１次フロートラッカー１１６Ａへと転送する。少なくともいくつかの場合に、１次フロートラッカー１１６ＡはＴＣＰフローのために格納されたデータを記憶し、承認されたシーケンス番号（＋１）がＳＹＮ／ＡＣＫパケットにおいて顧客１６０へ送信された数値に一致することを確認する。１次フロートラッカーはその後ＴＣＰフローのためのサーバノード１３０を選択し、ＴＣＰフローのためにすでに格納されたデータ、サーバシーケンス番号、および選択されたサーバノード１３０上のロードバランサモジュール１３２のためのＩＰアドレスを含む別のメッセージを２次フロートラッカー１１６Ｂに転送する。２次フロートラッカー１１６Ｂはサーバシーケンス番号を確認し、情報を記録し、生成されたＳＹＮメッセージを選択されたサーバノード１３０上のロードバランサモジュール１３２に送信する。ＴＣＰフローのＣＰエンドポイントペアはここでロードバランサモジュール１３２／サーバノード１３０へとマッピングされる。サーバノード１３０上のロードバランサモジュール１３２は、２次フロートラッカー１１６Ｂから生成されたＳＹＮメッセージを受信する際に、サーバノード１３０上のサーバ１３４のための正しいＴＣＰ　ＳＹＮパケットを作成する役割を担う。ＳＹＮパケットの増加時には、ソースＩＰアドレスに顧客１６０の実際のＩＰアドレスが取り込まれ、それによってサーバ１３４は顧客１６０から直接ＴＣＰ接続要求を受信したことを把握する。ロードバランサモジュール１３２はＴＣＰフローに関連する詳細を、例えばローカルハッシュテーブル内に格納し、ＴＣＰ　ＳＹＮパケットをサーバ１３４に送信する（例、ＳＹＮパケットをサーバ１３４のＬｉｎｕｘカーネルに注入する）。

【００７７】

図１０Ｃにおいて、入口サーバ１１２、１次フロートラッカー１１６Ａ、および２次フロートラッカー１１６Ｂの役割はそれぞれ異なるロードバランサノード１１０により果たされる。しかし場合によっては、ＴＣＰフローのための入口サーバ１１２の役割を担うロードバランサノード１１０は、ＴＣＰフローのための１次フロートラッカー１１６Ａまたは２次フロートラッカー１１６Ｂの役割を担うノードと同一のノード１１０である（しかし両方ではない）。例えば、図１０Ｄにおいて、１次フロートラッカー１１６Ａが異なるロードバランサノード１１０Ｂ上にある一方で、２次フロートラッカー１１６ＢはＴＣＰフローのための入口サーバ１１２の役割を担う同一のロードバランサノード１１０Ａ上にある。

【００７８】

図１０Ｅを参照すると、サーバ１３４（例、Ｌｉｎｕｘカーネル）はロードバランサモジュール１３２も遮断するＳＹＮ／ＡＣＫパケットで応答する。ＳＹＮ／ＡＣＫパケットは、もともとＳＹＮ／ＡＣＫにおいて２次フロートラッカー１１６Ｂから顧客１６０へと伝達されたもの（図１０Ａを参照）とは異なるＴＣＰシーケンス番号を含んでもよい。ロードバランサモジュール１３２はシーケンス番号デルタを受信および発信データパケットへと適用させる役割を担う。サーバ１３４からのＳＹＮ／ＡＣＫパケットはまた、ロードバランサモジュール１３２から２次フロートラッカー１１６Ｂへと戻るメッセージ（例、ＵＤＰメッセージ）のきっかけとなり、選択したサーバノード１３０／ロードバランサモジュール１３２／サーバ１３４への接続が成功したことを示す。このメッセージの受信時に２次フロートラッカー１１６Ａは、顧客１６０とサーバ１３４との間の顧客およびパブリックエンドポイントのペア（ＣＰ）マッピングを送信されたとおりに記録し、同様にＣＰマッピングを記録する１次フロートラッカー１１６Ａへと類似のメッセージを送信してもよい。１次フロートラッカー１１６Ａはその後入口サーバ１１２へとＣＰマッピングメッセージを転送してもよく、それにより入口サーバ１１２は接続のためにあらゆるバッファされたデータパケットをカプセル化されたデータパケットとしてローカルサーバノード１３０上のロードバランサモジュール１３２へと転送できるようになる。

【００７９】

図１０Ｆを参照すると、接続のためのＣＰマッピングは入口サーバによって把握され、

したがって接続のための入口サーバ１１２により受信された受信ＴＣＰパケットは、（例、ＵＤＰに従って）カプセル化され、カプセル化されたデータパケットとしてサーバノード１３０上のローカルロードバランサモジュール１３２に直接転送されてもよい。ロードバランサモジュール１３２はデータパケットを脱カプセル化し、例えばカーネルのＴＣＰスタック上へとＴＣＰパケットを注入することでＴＣＰパケットをサーバノード１３０上のサーバ１３４へと送信する。サーバ１３４からのアウトバウンドパケットは、サーバノード１３０上のロードバランサモジュール１３２によって遮断され、（例、ＵＤＰに従って）カプセル化され、そしてロードバランサモジュール１３２がこの接続のための出口サーバ１１４として無作為に選択する任意のロードバランサノード１１０へと転送される。出口サーバ１１４はパケットを脱カプセル化し、脱カプセル化されたデータパケットを顧客１１６へと送信する。選択したロードバランサノード１１０の出口関数はステートレスであり、そのため出口サーバの役割を担うロードバランサノード１１０の故障時には、異なるロードバランサノード１１０を接続のための出口サーバ１１４としてとして選択することが可能である。しかし一般的には、接続の維持のための出口サーバ１１４と同一のロードバランサノード１１０が、アウトバウンドパケットの再配置を減少させ、または取り除くために用いられる。

【００８０】

図１０Ｇを参照すると、少なくともいくつかの実施形態において、１次フロートラッカー１１６Ａによって選択されたサーバノード１３０Ａ上のロードバランサモジュール１３２Ａ（図１０Ｃを参照）は、自身に負荷がかかり過ぎていると判断した場合に、２次フロートラッカー１１６Ｂから受信された、生成されたＳＹＮメッセージ（図１０Ｃを参照）を拒否する選択肢を有する。少なくともいくつかの実施形態において、生成されたＳＹＮメッセージは拒否の最大値を許す生存時間（ＴＴＬ）の数値またはカウンターを含む。少なくともいくつかの実施形態において、このＴＴＬ値がゼロに達した場合、ロードバランサモジュール１３２Ａが負荷を制限するために接続の受け入れまたは接続の破棄のいずれかを行ってもよい。ロードバランサモジュール１３２Ａは接続の拒否を決定した場合、ＴＴＬ値をディクリメントし、２次フロートラッカー１１６Ｂに拒否メッセージを送信する。２次フロートラッカー１１６ＢはＣＰマッピングをリセットし、リリースメッセージを同様のことを行う１次フロートラッカー１１６Ａに送信する。１次フロートラッカー１１６Ａは別のサーバノード１３０Ｂ上の新規ロードバランサモジュール１３２Ｂを選択し、２次フロートラッカー１１６Ｂに新規対象のメッセージを返信し、２次フロートラッカー１１６Ｂが新規生成されたＳＹＮメッセージを新規に選択されたロードバランサモジュール１３２Ｂに送信する。パケットの破棄によりこれらのシーケンスが完了しない可能性があることに留意する。しかし、顧客１６０からの再伝達によりロードバランサモジュールの選択処理が１次フロートラッカー１１６Ａにおいて再び行われてもよい。１次フロートラッカー１１６Ａは必ずそうするとは限らないが、生成されたＳＹＮパケットの前回の拒否について把握していない場合、接続のための同一のロードバランサモジュール１３２を選択してもよい。

【００８１】

少なくともいくつかの実施形態において、ＴＴＬカウンターは継続的に接続要求をサーバノード１３０に送るのを阻止してもよく、これは例えばすべてのサーバノード１３０がビジー状態である場合に発生してもよい。少なくともいくつかの実施形態において、ロードバランサモジュール１３２が、それぞれのサーバノード１３０に代わり接続要求を拒否する時は毎回、ロードバランサモジュール１３２がＴＴＬカウンターをディクリメントする。フロートラッカーノード１１６は、ＴＴＬカウンターがゼロではない（すなわちある特定の閾値を超える）限りＴＴＬカウンターを監視してもよく、別のサーバノード１３０を選択し再度試みてもよい。ＴＴＬカウンターがゼロに達する（すなわちある特定の閾値を超える）場合、接続要求は破棄され、その接続のために選択されたサーバノード１３０のうちの１つに接続要求を送信する試みをフロートラッカーノード１１６が再び行うことはない。少なくともいくつかの実施形態において、エラーメッセージがそれぞれの顧客１

６０に送信されてもよい。
【００８２】
少なくともいくつかの実施形態において、分散型ロードバランサシステムは複数のパブリックＩＰアドレスをサポートする。こうして、顧客１６０が同一の顧客用ポート番号から２つの異なるパブリックＩＰアドレスへの２つのＴＣＰ接続を開始することが可能になる。これらのＴＣＰ接続は顧客１６０の観点とは異なるが、内部では分散型ロードバランサが同一のサーバノード１３０への接続をマッピングしてもよく、これにより衝突が起こる。少なくともいくつかの実施形態において、可能性のある衝突を検知し処理するために、ロードバランサモジュール１３２は図１０Ｃおよび１０Ｄで示すように２次フロートラッカー１１６Ｂから生成されたＳＹＮパケットを受信する際に、アドレス情報をアクティブな接続と比較してもよく、この接続が衝突を発生させる場合には、図１０Ｇで示すように接続要求を拒否してもよい。
ロードバランサノードの故障および追加の処理
【００８３】
従来のロードバランサの多くにおいては、ロードバランサの故障時には既存の接続の一部またはすべてが喪失される。少なくともいくつかの実施形態において、単一のロードバランサノード１１０の故障時には、接続が完全に正常に戻るまで顧客およびサーバが接続を通じてパケットの交換を継続できるように、分散型ロードバランスシステムが確立された接続のうち少なくともいくつかを維持してもよい。また分散型ロードバランスシステムは、故障時に確立の処理中であった接続の伝達を継続してもよい。
【００８４】
分散型ロードバランスシステムの少なくともいくつかの実施形態において、単一のロードバランサノード１１０の故障時に既存の顧客接続を回復させることができる故障回復プロトコルが実装されてもよい。しかし複数のロードバランサノード１１０の故障により、顧客接続が喪失される場合がある。少なくともいくつかの実施形態において、顧客１６０とサーバ１３４との間のＴＣＰの再伝達がロードバランサノード１１０の故障後の回復手段として用いられてもよい。
【００８５】
可能性のあるロードバランサノード１１０の故障に加えて、新規ロードバランサノード１１０が分散型ロードバランサシステムに追加されてもよい。これら新規ノード１１０はロードバランサレイヤーに、またそれにより一貫したハッシュリング追加されてもよく、ロードバランサノード１１０の既存の顧客接続に関する役割は、必要に応じて変更に従い調整される。
フロートラッカーノードの故障および追加の処理
【００８６】
少なくともいくつかの実施形態において、各接続が確立される（例、図１０Ａ～１０Ｇを参照）にしたがって、接続状態の情報が１次および２次フロートラッカーと呼ばれる２つのロードバランサノード１１０を通じて渡される。これらは例えば（顧客ＩＰ：ポート、パブリックＩＰ：ポート）タプルをハッシュ関数入力として用いる一貫したハッシュアルゴリズムを利用して決定されてもよい。単一のロードバランサノード１１０の故障時には、パケットを接続のための選択されたサーバノード１３０へと導くために、少なくとも１つの生存するロードバランサノード１１０は一貫したハッシュ関数を通じて継続的にマッピングされてもよく、また接続のために必要な状態情報を含んでもよい。また、ロードバランサノード１１０を一貫したハッシュリングへと追加する場合、接続のための状態情報は適切なフロートラッカーへとリフレッシュされてもよい。
【００８７】
図１１Ａ～１１Ｄは少なくともいくつかの実施形態による、ロードバランサノードの一貫したハッシュリングにおいてメンバーシップに影響を与えるイベントの処理を示す。これらのイベントは、新規１次フロートラッカーノードの追加、新規２次フロートラッカーノードの追加、１次フロートラッカーノードの故障、および２次フロートラッカーノード

の故障を含んでもよいがこれらに限定されない。

【００８８】

図１１Ａは、新規１次フロートラッカーノードの一貫したハッシュリングへの追加処理を示す。図１１Ａの上部列は、１つまたは複数の顧客接続のための１次フロートラッカーとしてのフロートラッカー１１６Ａおよび１つまたは複数の同一の接続のための２次フロートラッカーとしてのフロートラッカーノード１１６Ｂを示す。図１１Ａの下部列においては、新規フロートラッカーノード１１６Ｃが追加され、１つまたは複数の顧客接続のための１次フロートラッカーとなっている。以前は１次フロートラッカーであったフロートラッカーノード１１６Ａは２次フロートラッカーとなり、以前は２次フロートラッカーであったフロートラッカーノード１１６Ｂは、一貫したハッシュリングにおける次のフロートラッカーとなる。フロートラッカー１１６Ａおよび１１６Ｂにより維持された１つまたは複数の顧客接続のための状態情報は新規１次フロートラッカー１１６Ｃに提供されてもよい。また、フロートラッカー１１６Ｂは２次フロートラッカーの役割として以前トラッキングしていた接続を「忘れて」もよい。

【００８９】

図１１Ｂは、新規２次フロートラッカーノードの一貫したハッシュリングへの追加処理を示す。図１１Ｂの上部列は、１つまたは複数の顧客接続のための１次フロートラッカーとしてのフロートラッカー１１６Ａおよび１つまたは複数の同一の接続ための２次フロートラッカーとしてのフロートラッカーノード１１６Ｂを示す。図１１Ｂの下部列においては、新規フロートラッカーノード１１６Ｃが追加され、１つまたは複数の顧客接続のための２次フロートラッカーとなっている。フロートラッカーノード１１６Ａは１つまたは複数の接続のための１次フロートラッカーのままであり、以前は２次フロートラッカーであったフロートラッカーノード１１６Ｂは一貫したハッシュリングにおける次のフロートラッカーとなる。フロートラッカー１１６Ａおよび１１６Ｂによって維持された１つまたは複数の顧客接続のための状態情報は新規２次フロートラッカー１１６Ｃに提供されてもよい。また、フロートラッカー１１６Ｂは２次フロートラッカーの役割として以前トラッキングしていた接続を「忘れて」もよい。

【００９０】

図１１Ｃは、一貫したハッシュリングにおける１次フロートラッカーノードの故障の処理を示す。図１１Ｃの上部列は、１つまたは複数の顧客接続のための１次フロートラッカーとしてのフロートラッカー１１６Ａ、１つまたは複数の同一の接続のための２次フロートラッカーとしてのフロートラッカーノード１１６Ｂ、および一貫したハッシュリングにおける次のフロートラッカーとしてのフロートラッカーノード１１６Ｃを示す。図１１Ｃの下部列においては、１次フロートラッカーノード１１６Ａが故障している。フロートラッカーノード１１６Ｂは１つまたは複数の接続のための１次フロートラッカーとなり、フロートラッカーノード１１６Ｃは１つまたは複数の接続のための２次フロートラッカーとなる。１つまたは複数の顧客接続のための状態情報はフロートラッカー１１６Ｂによって維持され、新規２次フロートラッカー１１６Ｃへと提供されてもよい。

【００９１】

図１１Ｄは一貫したハッシュリングにおける２次フロートラッカーノードの故障の処理を示す。図１１Ｄの上部列は、１つまたは複数の顧客接続のための１次フロートラッカーとしてのフロートラッカー１１６Ａ、１つまたは複数の同一の接続のための２次フロートラッカーとしてのフロートラッカーノード１１６Ｂ、および一貫したハッシュリングにおける次のフロートラッカーとしてのフロートラッカーノード１１６Ｃを示す。図１１Ｄの下部列においては、２次フロートラッカーノード１１６Ｂが故障している。フロートラッカーノード１１６Ａは１つまたは複数の接続のための１次フロートラッカーのままであり、フロートラッカーノード１１６Ｃは１つまたは複数の接続のための２次フロートラッカーとなる。１つまたは複数の顧客接続のための状態情報はフロートラッカー１１６Ｂにより維持され、新規２次フロートラッカー１１６Ｃに提供されてもよい。

【００９２】

少なくともいくつかの実施形態において、サーバノード１３０上のロードバランサモジュール１３２はロードバランサノード１１０への接続公開を行う。少なくともいくつかの実施形態において、接続公開は定期的に（例、１秒に１回）または非定期的に現在の接続状態の情報をサーバノード１３０から、接続のための１次および２次フロートラッカーノード両方への接続マッピングのリフレッシュまたは復元を行うフロートラッカーノードおよび入口ノードの役割を担うロードバランサノード１１０へとプッシュする。少なくともいくつかの実施形態において、ロードバランサモジュール１３２は例えば図１１Ａ～１１Ｄで示される通り、フロートラッカーのメンバーシップ変更を検知してもよい。それに応じてロードバランサモジュール１３２は、メンバーシップが変更された際に接続のために変更されたかもしれない、１次および２次フロートラッカーノードにおける接続のための状態情報を追加するために、接続公開を行ってもよい。接続公開により、複数のロードバランサノードの故障時に少なくともいくつかの確立された接続が回復されてもよいことに留意する。

故障に関連するメッセージフロー

【００９３】

少なくともいくつかの実施形態において、１次および２次フロートラッカーノードの間のプロトコルは、修正または同期機能を含んでもよい。例えば図１１Ａを参照すると、新規１次フロートラッカーノード１１６Ｃが一貫したハッシュリングに参加する際には、新規ノード１１６Ｃがいくつかの数（～１／Ｎ）の接続のための一貫したハッシュのキー空間に要求を出し、エッジルータ１０４からこれらの接続に関連するトラフィックの受信を開始してもよい。しかし新規１次フロートラッカーノード１１６Ｃは接続のために格納された状態を一切有さないため、各パケットに対して、顧客１６０から受信された第１のパケットとして動作してもよい。１次フロートラッカーはＳＹＮデータパケットに応じてサーバのＴＣＰシーケンス番号の生成（例、図１０Ａを参照）および顧客１６０からの第１のＡＣＫパケットに応じてサーバノード１３０の選択（例、図１を参照）の役割を担い、これらの生成された数値は前の１次フロートラッカー（図１１Ａにおけるフロートラッカーノード１１６Ａ）により選択された数値と相違してもよい。しかし少なくともいくつかの実施形態において、一貫したハッシュアルゴリズムは前の１次フロートラッカー（図１１Ａにおけるフロートラッカーノード１１６Ａ）を２次フロートラッカーの役割に割り当て、このフロートラッカーは接続のためのすでに格納された状態をいまだに保持する。したがって少なくともいくつかの実施形態において２次フロートラッカー（図１１Ａにおけるフロートラッカーノード１１６Ａ）は、１次フロートラッカー１１６Ｃから受信された情報における不一致を検知した時、２次フロートラッカーは接続のためのフロートラッカーとしての役割を担う２つのロードバランサノード１１０を同期するために更新メッセージを１次フロートラッカー１１６Ｃへと返信することができる。一貫したハッシュリングのメンバーシップにおいて他の変更がなされた後にフロートラッカーを同期するために類似の方法が用いられてもよい。

ロードバランサモジュールの説明

【００９４】

少なくともいくつかの実施形態において、ロードバランサモジュール１３２は各サーバノード１３０上にある分散型ロードバランサシステムの構成要素である。ロードバランサノード１３２の役割は、ロードバランサノード１１０から受信されたパケットの脱カプセル化および脱カプセル化されたパケットのサーバノード１３０上のサーバ１３４への送信、ならびにサーバ１３４からの発信パケットのカプセル化およびカプセル化されたパケットのロードバランサノード１１０への送信を含むがそれに限定されない。

【００９５】

少なくともいくつかの実施形態において、入口サーバ１１２の役割を担うロードバランサノード１１０からサーバノード１３０上のロードバランサモジュール１３２への受信パケットは、実際の顧客データパケットをカプセル化するステートレスプロトコル（例、ＵＤＰ）パケットである。カプセル化された顧客データパケットは各々、ソースアドレスと

してそれぞれの顧客１６０のオリジナルの顧客ＩＰ：ポートを、そして宛先アドレスとしてサーバ１３４パブリックＩＰ：ポートを有する。ロードバランサモジュール１３２は顧客データパケットを脱カプセル化し、例えばパケットの方向をローカルホストＴＣＰフローへと変更することで、それぞれのサーバノード１３０上のサーバ１３４へと送信する。

【００９６】

少なくともいくつかの実施形態において、サーバ１３４から出口サーバ１１４の役割を担うロードバランサノード１１０への発信パケットは、発信ＩＰパケットをカプセル化するステートレスプロトコル（例、ＵＤＰ）パケットである。ロードバランサモジュール１３２は発信ＩＰパケットをカプセル化し、ファブリック１２０を通してカプセル化されたパケットを出口サーバ１１４へと送信する。各カプセル化された発信ＩＰパケットは、ソースアドレスとしてサーバ１３４パブリックＩＰ：ポートを、そして宛先アドレスとしてそれぞれの顧客１６０の顧客ＩＰ：ポートを有する。

ロードバランサモジュール機能

【００９７】

少なくともいくつかの実施形態において、サーバノード１３０上のロードバランサモジュール１３２の機能は以下の１つまたは複数を含んでもよいが、それらに限定されない：

＊ １つまたは複数のロードバランサノード１１０からの、例えば顧客１６０への接続を処理する入口サーバ１１２からの、ＵＤＰトンネルの終了。これは入口サーバ１１２から受信された受信顧客データパケットのＵＤＰ脱カプセル化を含む。

＊ 接続のための発信トラフィックを受信する出口サーバ１１４の選択。

＊ それぞれのサーバ１３４上の接続における発信ＩＰパケットの遮断、接続のための発信ＩＰパケットのカプセル化、およびカプセル化されたデータパケットの出口サーバ１１４への送信。

＊ フロートラッカーノード１１６が顧客１６０にＳＹＮ／ＡＣＫを送信する際にシーケンス番号がフロートラッカーノード１１６により生成されたシーケンス番号と整列させるための、受信および発信パケットにおけるシーケンス番号のマングリング。

＊ 例えばそれぞれのサーバ１３４の現在の負荷を示す１つまたは複数の測定基準に基づく、それぞれのサーバ１３４のための接続を受け入れるか拒否するかの決定。

＊ 顧客ＩＰ：ポートアドレスの衝突を回避するためのアクティブな接続がある場合の、同一の顧客ＩＰ：ポートアドレスからそれぞれのサーバ１３４への接続の検知および拒否。

＊ 接続トラッキングおよび接続公開。

ロードバランサモジュールの構成情報

【００９８】

少なくともいくつかの実施形態において、各ロードバランサモジュール１３２は構成のために、以下の情報の組の１つまたは複数を獲得し、ローカルに格納してもよいが、それらに限定されない：ロードバランサノード１１０エンドポイントの組、伝達する有効なパブリックＩＰアドレスの組、およびそれぞれのサーバ１３４が受信接続を受け入れる１つまたは複数のポート番号。少なくともいくつかの実施形態において、図１で示すようにこの情報は、分散型ロードバランサシステムの構成要素である構成サービス１２２から獲得されるか、またはそれへのアクセスか問い合わせにより更新されてもよい。いくつかの実施形態においては、他の情報獲得方法が用いられてもよい。

ロードバランサモジュールのパケット処理

【００９９】

少なくともいくつかの実施形態による、インバウンドトラフィックおよびアウトバウンドトラフィックのためのロードバランサモジュール１３２の動作を以下に記載する。少なくともいくつかの実施形態において、インバウンドデータパケットがロードバランサモジュール１３２により受信される際に、データパケットがＵＤＰパケットから脱カプセル化され、脱カプセル化されたＴＣＰパケットにおける宛先アドレスは構成された有効なパブリックＩＰアドレスの組に対して最初に検証される。一致がない場合、パケットは破棄さ

れるかまたは無視される。少なくともいくつかの実施形態において、シーケンス番号がＳＹＮ／ＡＣＫパケットを顧客１６０に送信したフロートラッカーノード１１６により生成され無作為に選択されたシーケンス番号に一致するように、ロードバランサモジュール１３２はＴＣＰヘッダにおけるシーケンス番号を定数デルタにより調整してもよい。ロードバランサモジュール１３２は［顧客：パブリック］エンドポイントから［顧客／サーバ］エンドポイントへのマッピングを内部状態として記録する。

【０１００】

少なくともいくつかの実施形態において、サーバ１３４からのアウトバウンドＴＣＰデータパケットのために、ロードバランサモジュール１３２はまず内部状態を確認し、パケットが、ロードバランサモジュールが管理しているアクティブな接続のためのものであるかどうかを決定する。そうでない場合、ロードバランサモジュール１３２はただパケットを渡す。そうである場合、ロードバランサモジュール１３２は発信ＴＣＰパケットを例えばＵＤＰに従ってカプセル化し、カプセル化されたパケットをこの接続のための出口サーバ１１４として選択されたロードバランサノード１１０へと転送する。少なくともいくつかの実施形態において、ロードバランサモジュール１３４は発信ＴＣＰパケットにおけるＴＣＰシーケンス番号を定数デルタにより調整して、ＳＹＮ／ＡＣＫパケットを顧客１６０に送信したフロートラッカーノード１１６により生成されたシーケンス番号を整列させてもよい。

接続のトラッキング

【０１０１】

少なくともいくつかの実施形態において、各サーバノード１３０上のロードバランサモジュール１３２はそれぞれのサーバ１３４へのすべてのアクティブな顧客接続のための接続の詳細を含むハッシュテーブルを管理する。少なくともいくつかの実施形態において、ハッシュテーブルのためのキーは（顧客ＩＰ：ポート、パブリックＩＰ：ポート）タプルである。少なくともいくつかの実施形態において、各顧客接続のための接続状態は以下の１つまたは複数を含むが、それらに限定されない：

＊ 顧客ＩＰ：ポート
＊ パブリックＩＰ：ポート
＊ フロートラッカー１１６ノードにより提供される最初のサーバのＴＣＰシーケンス番号。
＊ サーバのＴＣＰシーケンス番号デルタ。
＊ オリジナルの１次フロートラッカーＩＰアドレス。
＊ オリジナルの２次フロートラッカーＩＰアドレス。
＊ 最後に検知された入口サーバ１１２のＩＰアドレス。
＊ このエントリのための有効期限
＊ 最長期間未使用の（ＬＲＵ）／衝突指数。

【０１０２】

少なくともいくつかの実施形態において、各ロードバランサモジュール１３２はすべてのアクティブな顧客接続のための１次および２次フロートラッカーノードへの接続公開メッセージを定期的に生成する。少なくともいくつかの実施形態において、／ｐｒｏｃ／ｎｅｔ／ｔｃｐの内容がスキャンされロードバランサモジュールのハッシュテーブルにおけるアクティブな接続と交差し、Ｌｉｎｕｘカーネルが接続のトラッキングを停止するまでフロートラッカーノードへと継続的に公開される。接続公開については本明細書にて詳細に後述される。

シーケンス番号のマングリング

【０１０３】

上述のように少なくともいくつかの実施形態において、ロードバランサノード１１０はサーバ１３４の代わりに顧客１６０ＳＹＮパケットに応じて、ＳＹＮ／ＡＣＫパケットを生成する。顧客１６０がＡＣＫパケットを送信する後にのみ（ＴＣＰスリーウェイハンドシェイク）ロードバランサモジュール１１０がサーバノード１３０上のロードバランサモ

ジュール１３２へといずれかのデータを送信する。ロードバランサモジュール１３２が最初に顧客接続を確立するよう指示される際は、ロードバランサモジュール１３２がローカルでＳＹＮパケットを作成してサーバノード１３０上のサーバ１３４を用いてＴＣＰ接続を開始し、サーバ１３４に対応するＳＹＮ／ＡＣＫパケットを遮断する。通常、サーバ１３４（例、サーバノード１３０上のＬｉｎｕｘカーネル）がＳＹＮ／ＡＣＫパケットにおいてロードバランサノード１１０から受信された顧客の１つとはまったく異なるＴＣＰシーケンス番号を選択する。こうして少なくともいくつかの実施形態において、ロードバランサモジュール１３２は顧客１６０とサーバ１３４との間のＴＣＰ接続のすべてのパケットにおけるシーケンス番号の補正を行ってもよい。少なくともいくつかの実施形態において、ロードバランサモジュール１３２はロードバランサノード１１０により生成されたシーケンス番号とサーバ１３４により作成されたシーケンス番号との間の差異を計算し、その差異をデルタ値としてＴＣＰ接続のためのハッシュテーブルエントリ内に格納する。受信データパケットが顧客１６０から接続に到着する際には、ＴＣＰヘッダがサーバ１３４により用いられるシーケンス番号と整列しない確認番号を含むため、ロードバランサモジュール１３２はＴＣＰヘッダにおけるシーケンス番号の数値からデルタ値を減算する（例、２つの補数を用いて）。ロードバランサモジュールはまた、サーバ１３４から顧客１３０への接続上のアウトバウンドデータパケットにおけるシーケンス番号にデルタ値を追加する。

分散型ロードバランサシステムにおけるヘルスチェック

【０１０４】

分散型ロードバランサシステムの少なくともいくつかの実施形態において、各ロードバランサノード１１０はロードバランサの実装における正常なメンバー（すなわち、正常なロードバランサノード１１０およびサーバノード１３０）の一貫した見解を、少なくとも以下の理由により要求する：

＊ロードバランス－ロードバランサノード１１０がサーバノード１３０の故障を検知し、顧客のトラフィックを受け入れることができる正常なサーバノード１３０の組において収束する必要がある。

＊分散状態の管理－ロードバランサは複数のロードバランサノード１１０で共有された／複製された状態を有する分散型システムである（例、一貫したハッシュ機構に従って）。顧客のトラフィックを正しく処理するために、各ロードバランサノード１１０は最終的にロードバランサの実装における正常なメンバーの一貫した見解を有する必要がある。

【０１０５】

これを達成するため、分散型ロードバランサシステムの少なくともいくつかの実施形態は、ロードバランサの実装においてノードを監視し、可能な限り迅速に異常なノードを検知するヘルスチェックプロトコルの実施形態を実装してもよい。ヘルスチェックプロトコルはロードバランサの実装においてノード間にヘルス情報を伝播してもよく、正常なノードの組においてノードの収束を可能にする方法を提供してもよい。またヘルスチェックプロトコルは、ロードバランサの実装における正常／異常なノードおよび状態の変化を報告するための機構を提供してもよい。

【０１０６】

少なくともいくつかの実施形態において、ヘルスチェックプロトコルは以下の仮定のうちの１つまたは複数に基づいてもよいが、それらに限定されない：

＊ロードバランサの実装におけるすべてのノードが把握される（すなわち、ヘルスチェックプロトコルは発見を行わなくてもよい）。

＊ノードの故障はすべてフェイルストップである。

＊ノード間のすべてのメッセージはステートレスプロトコル（例、ＵＤＰ）メッセージであり、メッセージは破棄され、遅延させられ、複製され、または破損する可能性がある。メッセージの伝達の保証はない。

【０１０７】

少なくともいくつかの実施形態において、ロードバランサの実装におけるノード（例

、ロードバランサノード１１０またはサーバノード１３０）は以下の条件の下で正常であると見なされてもよい：
＊ノードの内部構成要素はすべてレディ状態である（顧客のトラフィックを処理する準備が完了している）。
＊ノードの受信／発信ネットワークリンクは正常である（少なくともどの顧客のトラフィックを流すかについてのネットワークインタフェースコントローラ（ＮＩＣ）に関しては）。
【０１０８】
図１２は少なくともいくつかの実施形態による、ヘルスチェック間隔に従って各ロードバランサノードにより実行されるヘルスチェック方法のハイレベルフローチャートである。１０００で示すように、各ロードバランサ間隔において、例えば１００ミリ秒毎に、各ロードバランサ（ＬＢ）ノード１１０は少なくとも１つの他のＬＢノード１１０および少なくとも１つのサーバノード１３０のヘルスチェックを行ってもよい。１００２で示すように、ロードバランサノード１１０はヘルスチェックに従って、そのローカルに格納されたヘルス情報を更新してもよい。１００４で示すように、ロードバランサノード１１０はその後、少なくとも１つの他のロードバランサノード１１０を無作為に選択し、そのヘルス情報を選択された１つまたは複数のロードバランサノード１１０へと送信してもよい。少なくともいくつかの実施形態において、ノード１１０はまた、正常なロードバランサノード１１０のリストを１つまたは複数のサーバノード１３０、例えばノード１１０によりヘルスチェックされる１つまたは複数の同一のサーバノード１３０へと送信してもよい。図１２の要素は以下において詳細に説明される。
【０１０９】
ヘルスチェックプロトコルの少なくともいくつかの実施形態において、ロードバランサノード１１０はそのヘルス状態を他のロードバランサノード１１０にアサートしない。その代わりに、１つまたは複数の他のロードバランサノード１１０がそのノード１１０のヘルスチェックを行ってもよい。例えば少なくともいくつかの実施形態において、各ロードバランサノード１１０はヘルスチェックを行う１つまたは複数の他のノード１１０を、定期的または非定期的に無作為に選択してもよい。別の実施例として、少なくともいくつかの実施形態において、１つまたは複数の他のロードバランサノード１１０、例えば一貫したハッシュリング等のノード１１０の番号付きリスト上の所与のロードバランサノード１１０の２つの最近傍ノードはそれぞれ、所与のノード１１０のヘルスチェックを定期的または非定期的に行ってもよい。少なくともいくつかの実施形態において、ノード１１０のヘルスチェックは図２３で示すように、ノード１１０上のＮＩＣ１１１４へと送信されたヘルスｐｉｎｇの利用を含んでもよい。少なくともいくつかの実施形態において、第２のノード１１０が正常であると第１のノード１１０がヘルスチェックを通じて決定する場合、第１のノード１１０は、ロードバランサノード１１０のためのローカルヘルス情報に格納された、第２のノード１１０のためのハートビートカウンターを更新（例、増加）してもよい。第１のノード１１０はそのローカルヘルス情報をロードバランサの実装における１つまたは複数の他のロードバランサノード１１０へと定期的または非定期的に送信してもよく、それら１つまたは複数の他のロードバランサノード１１０はそのローカルヘルス情報を適宜更新（例、第２のノードのためのハートビートカウンターの増加により）し、その更新されたローカルヘルス情報を１つまたは複数の他のノード１１０へと送信してもよい。第２のノード１１０のためのハートビート情報はこうしてロードバランサの実装における他のノード１１０へと伝播されてもよい。第２のノード１１０が正常である限り、第２のノード１１０から到達可能な他のすべてのノード１１０はこのように、第２のノード１１０のハートビートカウンターが一定期間毎に、例えば、１秒に１回または１０秒毎に一回、増加していることを確認すべきである。第２のノード１１０が、そのヘルスチェックを行う１つまたは複数のノード１１０により、異常であると検知された場合、ノード１１０のためのハートビートはヘルスチェックを行うノード１１０により一切送信されず、ある時間閾値の経過後、ロードバランサの実装１１０における他のノード１１０が、問

題のノード１１０が異常である、またはダウンしていると見なす。
【０１１０】
少なくともいくつかの実施形態において、ロードバランサノード１１０はその内部状態の１つまたは複数の態様を確認してもよく、ノード１１０が何らかの理由によるその異常を検知した場合、ノード１１０はそのヘルスチェックを行う他のノード１１０からのヘルスｐｉｎｇに対して応答を停止してもよい。したがって、異常なノード１１０のヘルスチェックを行うノード１１０は、そのノード１１０を以上であると見なしてもよく、そのノード１１０の代わりにハートビートの増加を伝播しなくてもよい。
ヘルスチェックプロトコルの説明
【０１１１】
少なくともいくつかの実施形態において、ヘルスチェックプロトコルはハートビートカウンター技術およびゴシッププロトコル技術を活用してもよい。ヘルスチェックプロトコルは２つの主要部分－ヘルスチェックおよびゴシップ／故障検知を有すると見なされてもよい。
【０１１２】
ヘルスチェック－ロードバランサの実装におけるすべてのロードバランサノード１１０は、実装における１つまたは複数の他のノード１１０のヘルスチェックを定期的または非定期的に行ってもよい。１つまたは複数の他のノードの決定方法は後述される。ヘルスチェックの中心となる概念は、ノード１１０が別のノード１１０のヘルスチェックを行い、他のノード１１０が正常であると決定する場合、そのチェックを行うノード１１０が他のノード１１０のハートビートカウンターを増加させまた伝播することにより、他のノード１１０が正常であるとアサートする。すなわち、ノード１１０はそのヘルス状態を他のノードにアサートせず、その代わりに、１つまたは複数の他のノード１１０がロードバランサの実装における各ノード１１０のヘルス状態をチェックしアサートする。
【０１１３】
ゴシップ／故障検知－少なくともいくつかの実施形態において、ヘルスチェックプロトコルはロードバランサの実装におけるメンバーであるロードバランサノード１１０の間にロードバランサノード１１０のヘルス情報を伝播するゴシッププロトコルを活用してもよい。ゴシッププロトコルは迅速に収束し、分散型ロードバランスシステムの目的に十分な最終的な一貫性を保証する。少なくともいくつかの実施形態において、ゴシッププロトコルの利用により各ロードバランサノード１１０は、ロードバランサの実装における互いのノード１１０のためのハートビートカウンターを、例えばハートビートリストにおいて維持する。各ロードバランサノード１１０は上記のように少なくとも１つの他のロードバランサノード１１０のヘルスチェックを定期的または非定期的に行い、ヘルスチェックを通じてチェックを行ったノード１１０が正常であると決定した際に、ノード１１０のためのハートビートカウンターを増加させる。少なくともいくつかの実施形態において、各ロードバランサノード１１０は定期的または非定期的に、ロードバランサの実装における少なくとも１つの他のノード１１０を現在のハートビートリストの送信先として無作為に選択する。別のノード１１０からハートビートリストを受信した際に、２つのリスト（受信されたリストおよびそのリスト）上の各ノード１１０のための最大のハートビートカウンターを決定し、決定された最大のハートビートカウンターをそのハートビートリストにおいて利用することで、ロードバランサノード１１０は受信されたリストのハートビート情報をそのハートビートリストと組み合わせる。次にこのハートビートリストは別の無作為に選択されたノード１１０へと送信され、選択されたノード１１０がそのハートビートリストの更新等を適宜行う。この技術を用い、各正常なノード１１０のためのハートビート情報は最終的に（例、数秒後に）すべての他のロードバランサの実装におけるロードバランサノード１１０へと伝播される。所与のロードバランサノード１１０のためにハートビートカウンターし続ける限り、それは他のノード１１０により正常であると見なされる。ロードバランサノード１１０のハートビートカウンターがヘルスチェックおよびゴシップ方法により特定の期間中に増加されない場合は、他のロードバランサノード１１０がその後

、異常であると見なされたロードバランサノード１１０上で収束してもよい。
ヘルスチェックを行うロードバランサノード
【０１１４】
少なくともいくつかの実施形態による、別のロードバランサノード１１０により実行されてもよいロードバランサノード１１０のヘルスチェック方法を以下に記載する。図２３に関連し、少なくともいくつかの実施形態において、ノード１１０のために以下の条件のうちの１つまたは複数が決定された場合、ロードバランサノード１１０は正常であると見なされてもよい：
＊ノード１１０のプロセッサの閾値（例、コアパケット処理コード１１０８の閾値）がレディ状態（内部）である。
＊ノード１１０がエッジルータ１０４のＩＰアドレスおよび／またはＭＡＣアドレスを把握している（内部）。
＊ノード１１０のすべての閾値および／またはプロトコルハンドラーがレディ状態である（内部）。
＊北側（エッジルータ１０４／境界ネットワーク）から、また南側（サーバ１３０／本番ネットワーク）からの受信および発信リンクがアクティブである（外部）。
＊ロードバランサの実装において用いられるネットワークインタフェースコントローラ（ＮＩＣ）を通じて、ノード１１０がパケットを受信およびディスパッチすることが可能である。例えば図２３で示される例示的なロードバランサノード１１０の実施形態において、ノード１１０は北向きのＮＩＣ１１１４Ａおよび南向きのＮＩＣ１１１４Ｂを通じてパケットの受信およびディスパッチに成功すべきである。
【０１１５】
１つまたは複数のこれらのヘルス条件が所与のノード１１０に当てはまらない場合、そのノード１１０は正常でないと見なされてもよい。いくつかの実施形態において、上記条件のすべてがノード１１０に当てはまる場合にのみ、ノード１１０は正常であると見なされることに留意する。
【０１１６】
少なくともいくつかの実施形態において、上記ヘルス条件に加えて、図２３においてＮＩＣ１１１４Ｃとして示され、例えば制御プレーン通信のために用いられてもよい、各ロードバランサノード１１０上の第３のＮＩＣもまた、ＮＩＣへパケットを送信し、ＮＩＣからパケットを受信することでヘルスチェックを行うノード１１０によりチェックされてもよく、第３のＮＩＣのチェックが失敗した場合、チェックされているノード１１０は異常であると見なされてもよい。
【０１１７】
図１３は少なくともいくつかの実施形態による、別のロードバランサノードからのロードバランサノードのヘルスチェック方法の実施例を示す。この実施例では、ロードバランサノード１１０Ａはロードバランサノード１１０Ｂのヘルスチェックを行っている。ノード１１０Ａおよび１１０Ｂはそれぞれ、北向きのＮＩＣ（図２３のＮＩＣ１１１４Ａ）および南向きのＮＩＣ（図２３のＮＩＣ１１１４Ｂ）を有する。１では、ノード１１０Ａがパケット（例、ｐｉｎｇパケット）をその北向きのＮＩＣからノード１１０Ｂの北向きのＮＩＣへとエッジルータ１０４を通じて送信する。ノード１１０Ｂはその北向きのＮＩＣにおいてパケットを受信し、上記リストにおいて与えられた条件が満たされた場合、２においてその北向きのＮＩＣからノード１１０Ａの北向きのＮＩＣへとファブリック１２０を通じて応答を送信する。３において、その北向きのＮＩＣで応答を受信した際に、ノード１１０Ａはパケット（例、ｐｉｎｇパケット）をその南向きのＮＩＣからノード１１０Ｂの南向きのＮＩＣへとファブリック１２０を通じて送信する。ノード１１０Ｂはその南向きのＮＩＣにおいてパケットを受信し、上記リストにおいて与えられた条件が満たされた場合、４においてその南向きのＮＩＣからノード１１０Ａの南向きのＮＩＣへとエッジルータ１０４を通じて応答を送信する。その南向きのＮＩＣにおいて応答を受信した際、ノード１１０Ａはノード１１０Ｂを正常であると見なしてノード１１０Ｂのローカルハー

トビートカウンターを増加させ、その後それは上述のようにゴシッププロトコルに従って他のノード１１０へと伝播されてもよい。

【０１１８】

上記の代わりとして、いくつかの実施形態においては、ロードバランサノード１１０Ｂはその南向きのＮＩＣを通じて、その北向きのＮＩＣで受信された、ノード１１０Ａの南向きのＮＩＣへの第１のｐｉｎｇメッセージに応答してもよく、その北向きのＮＩＣを通じて、その南向きのＮＩＣで受信されたノード１１０Ａの北向きのＮＩＣへの第２のｐｉｎｇメッセージに応答してもよい。

【０１１９】

また、いくつかの実施形態においては、ノード１１０Ａはまた、ノード１１０Ｂが正常である場合に、それ自体の第３のＮＩＣからノード１１０Ｂの第３のＮＩＣへとｐｉｎｇを送り、ノード１１０Ｂの第３のＮＩＣからのその第３のＮＩＣ上のｐｉｎｇメッセージへの応答を受信して、（図２３でＮＩＣ１１１４Ｃとして示される）制御プレーン通信のために用いられるノード１１０Ｂの第３のＮＩＣのヘルスチェックを行う。ｐｉｎｇメッセージおよび応答は１つまたは複数の制御プレーン装置１７０、例えばネットワークスイッチを通過してもよい。

【０１２０】

上記のヘルスチェック機構は、すべてのノード１１０ＢのＮＩＣと同様にすべての受信および発信リンクならびに全方向（北、南、および制御プレーンを通じて）のノード１１０Ｂのデータ経路を実行し、顧客パケットもそうするようにｐｉｎｇデータパケットが内部キューをトラバースしノード１１０Ｂのディスパッチを行う時に、ノード１１０Ｂの内部ヘルス状態を検証する。

ロードバランサノードへのヘルスチェックの役割の割り当て

【０１２１】

少なくともいくつかの実施形態において、ロードバランサの実装におけるすべてのロードバランサノード１１０は、例えば構成関数を通じて、および／または図１で示すように構成要素としての構成サービス１２２を通じて、ロードバランサの実装におけるすべての他のロードバランサノード１１０のリスト（例、ソートされたリスト）へのアクセスを有する。少なくともいくつかの実施形態において、各ロードバランサノード１１０は、各ヘルスチェック間隔でヘルスチェックを行うためにリスト上の１つまたは複数の他のノード１１０を無作為に選択し、正常であると決定された場合にハートビートカウンターを増加させてもよい。リストはヘルスチェック機構を通じて現在正常であると見なされているものも異常であると見なされているものも関わらずロードバランサの実装におけるすべてのロードバランサノード１１０を含んでよく、現在異常なノード１１０も正常なノード１１０と同様にリストから無作為に選択されてヘルスチェックを行われてもよいことに留意する。こうして、現在異常なノード１１０は、そのノード１１０のヘルスチェックを行う１つまたは複数のノード１１０により正常であると決定されてもよく、そのハートビートカウンターが増加させられて他のノード１１０へと伝播されてもよく、このようにして異常なノード１１０が正常な状態に戻されてもよい。

【０１２２】

その代わりにいくつかの実施形態においては、各ロードバランサノード１１０はリスト上の１つまたは複数の他のノード１１０のヘルスチェックの役割および、正常であると決定された場合のハートビートカウンターの増加の役割を担ってもよい。例えばいくつかの実施形態においては、各ノード１１０は２つの他のノード、例えばリスト上の「左」（すなわち前）そして「右」（すなわち次）の最近傍ノード１１０の役割を担ってもよい。リストは円環であってもよく、リストの「最後」のノード１１０がリストの「最初」のノード１１０のヘルスチェックの役割を担ってもよく、その逆もまた同様であることに留意する。いくつかの実施形態においては、２つの他のノード１１０が、例えばリスト上の次の２つの最近傍ノードとして他に選択されてもよい。いくつかの実施形態においては、各ノード１１０は、例えば３つまたは４つの他のノード１１０のような、リスト上の３つ以上

のノード１１０のヘルスチェックの役割を担ってもよい。少なくともいくつかの実施形態において、ノード１１０からチェックされている近傍ノード１１０が異常であると決定された場合、ノード１１０はその後、異常な近傍ノード１１０がチェックする役割を担っていたリスト上の少なくとも１つのノードのヘルスチェックの役割を担ってもよい。少なくともいくつかの実施形態において、その近傍ノード１１０（例、「左」および「右」の近傍ノード）のヘルスチェックに加えて、各ロードバランサノード１１０もまた定期的または非定期的に、リング上のノード１１０を無作為に選択し、その無作為に選択されたノード１１０のヘルスチェックを行い、正常であれば無作為なノード１１０のハートビートを増加させ伝播させてもよい。少なくともいくつかの実施形態において、他のノード１１０がすでに正常であるか正常でないかのどちらに見なされているかに関わらず、番号付きリスト上のすべての他のノード１１０が無作為な選択およびヘルスチェックのために考慮される。

【０１２３】

少なくともいくつかの実施形態において、各ノード１１０は１つまたは複数の無作為に選択されたノード１１０の、またはその代わりにその近傍ノード１１０および無作為に選択されたノードのヘルスチェックを通常の間隔で行い、その間隔はヘルスチェック間隔と称されてもよい。例えば、いくつかの実施形態においては、ハートビート間隔は１００ミリ秒であってもよく、それより短い、または長い間隔も利用されてもよい。また少なくともいくつかの実施形態において、各ノード１１０はその現在のハートビートリストを少なくとも１つの他の無作為に選択されたノード１１０に通常の間隔で送信または「ゴシップ」し、それはゴシップと称されてもよい。いくつかの実施形態においては、ヘルスチェック間隔およびゴシップ間隔が同じ長さであってもよいが、必ずしも同じである必要もない。

【０１２４】

図１４は少なくともいくつかの実施形態による、１つまたは複数の他のロードバランサノードのヘルスチェックを行うロードバランサノードを図示する。この実施例においては、ロードバランサの実装における８つのロードバランサノード１１０Ａ－１１０Ｈがある。破線の円は実装におけるすべてのノード１１０の番号付きリストを表す。いくつかの実施形態においては、各ノード１１０は各間隔でヘルスチェックを行う、リスト上の１つまたは複数の他のノード１１０を無作為に選択してもよい。その代わりとして、いくつかの実施形態においては、各ロードバランサノード１１０は番号付きリスト上の１つまたは複数の特定のノード１１０のチェックの役割を担ってもよい。例えばノード１１０Ａは、図１４に示された番号付きリストに従ってその２つの最近傍ノード１１０Ｂおよび１１０Ｈのヘルスチェックの役割を担ってもよい。またロードバランサノードもまた各ヘルスチェック間隔で、番号付きリストから別のノード１１０を無作為に選択してもよい。この実施例で示されるように、ノード１１０Ａもヘルスチェックのためにノード１１０Ｆを無作為に選択している。ゴシップ間隔において、ノード１１０Ａは何か別の正常なノード１１０、例えばノード１１０Ｄを無作為に選択し、その現在のハートビートリストを選択した他のノード１１０へと、例えばＵＤＰメッセージで送信する。ノード１１０は、別のノード１１０からハートビートリストを受信した際に、それ自身のハートビートリストを適宜更新し、次のゴシップ間隔で、ハートビートリストを１つまたは複数の無作為に選択されたノード１１０に伝播してもよい。

サーバノードのヘルスチェック

【０１２５】

上述のようなロードバランサノード１１０のヘルスチェックに加えて、ヘルスチェックプロトコルの実施形態は、それらノード１３０上のロードバランサモジュール１３２およびサーバ１３４を含むサーバノード１３０のヘルスチェックを行ってもよい。少なくともいくつかの実施形態において、以下の条件のうちの１つまたは両方がノード１３０のために決定された場合、サーバノード１３０は正常であると見なされてもよい：

＊ ロードバランサモジュール１３２が正常である。

＊ロスｐｉｎｇ（例、Ｌ７ヘルスｐｉｎｇ）への応答に成功する。

【０１２６】

図１５は少なくともいくつかの実施形態による、サーバノードのヘルスチェックを行うロードバランサノードを示す。少なくともいくつかの実施形態において、ロードバランサの実装におけるすべてのロードバランサノード１１０は、ロードバランサの実装におけるすべてのサーバノード１３０のリストと同様に、すべての他のロードバランサの実装におけるロードバランサノード１１０のリストへのアクセスを有する。１つまたは複数のリストは、例えば構成関数を通じておよび／または図１で示されるように構成要素である構成サービス１２２を通じて取得され、更新されてもよい。少なくともいくつかの実施形態において、図１５で示されるような一貫したハッシュリングを形成するために、サーバノード１３０は正常なロードバランサノード１１０に対して一貫したハッシュを行ってもよい。少なくともいくつかの実施形態において、リング内の各サーバノード１３０はリング内の正常なロードバランサノード１１０によりヘルスチェックを行う。例えば図１５では、サーバノード１３０Ａはロードバランサノード１１０Ａおよび１１０Ｃによりヘルスチェックを行う。これら２つのノード１１０は、一貫したハッシュリングにおけるサーバノード１３０のための第１の（ノード１１０Ａ）および第２の（ノード１１０Ｂ）ヘルスチェックノード１１０と称されてもよい。所与の正常なロードバランサノード１１０は２つ以上のサーバノード１３０のヘルスチェックを行ってもよいことに留意する。例えば図１５において、ロードバランサノード１１０Ａはまた、サーバノード１３０Ｂおよび１３０Ｃのヘルスチェックを行う。また所与のノードバランサノード１１０は、１つまたは複数の他のサーバノード１３０のための第１のヘルスチェックノード１１０および１つまたは複数のサーバノード１３０のための第２のヘルスチェックノード１１０であってもよい。例えば、図１５において、ロードバランサノード１１０Ａはサーバノード１３０Ａおよび１３０Ｂのための第１のヘルスチェッカーノードであり、サーバノード１３０Ｃおよび１３０Ｄのための第２のヘルスチェックッカーノードである。

【０１２７】

少なくともいくつかの実施形態において、ロードバランサノード１１０が故障した場合、一貫したハッシュリングにおけるメンバーシップは変更され、まだ正常でしたがって一貫したハッシュリング上にある１つまたは複数の他のロードバランサノード１１０が故障したノード１１０によってすでにヘルスチェックされたサーバノード１３０のヘルスチェックの役割を担ってもよい。

【０１２８】

少なくともいくつかの実施形態において、正常なノード１１０はそれぞれその割り当てられたサーバノード１３０のヘルスチェックを通常の間隔で行い、それはサーバチェック間隔と称されてもよい。少なくともいくつかの実施形態において、サーバチェック間隔は上述のゴシップ間隔以上の長さであってもよい。

【０１２９】

少なくともいくつかの実施形態において、サーバノード１３０のヘルスチェックを行うために、正常なロードバランサノード１１０（例、図１５のノード１１０Ａ）はサーバノード１３０（例、図１５のサーバノード１３０Ａ）へのヘルスｐｉｎｇメッセージ（例、Ｌ７　ＨＴＴＰヘルスｐｉｎｇメッセージ）を開始する。正常である場合、サーバノード１３０はｐｉｎｇ応答をロードバランサノード１１０に返信する。少なくともいくつかの実施形態において、ｐｉｎｇメッセージはサーバノード１３０上のロードバランサモジュール１３２により受信され処理されるため、成功した場合にはヘルスチェックｐｉｎｇがサーバノード１３０上のモジュール１３２が正常であることを確立する。ｐｉｎｇへの応答時に、ロードバランサノード１１０はサーバノード１３０を正常であると見なし、サーバノード１３０のためのハートビートカウンターを増加させる。

【０１３０】

少なくともいくつかの実施形態において、所与の正常なロードバランサノード１１０によりヘルスチェックを行われたすべてのサーバノード１３０のためのハートビートカウン

ターは、例えばすでにロードバランサノード１１０ハートビートカウンターのために説明された、各ノード１１０がそのハートビートリストを少なくとも１つの他の無作為に選択されたノード１１０へと通常の間隔（ゴシップ間隔）で送信するゴシップ技術に従って他のロードバランサノード１１０へと伝播されてもよく、また受信ノード１１０が２つのリスト上の最大値に従って、それ自体のハートビートリストを更新する。

故障検知およびゴシップ

【０１３１】

少なくともいくつかの実施形態において、上記のロードバランサノード１１０のヘルスチェックおよびサーバノード１３０のヘルスチェックを通じて取得された情報は、すべてのロードバランサノード１１０がロードバランサの実装の一貫した見解を維持できるように、ロードバランサの実装におけるすべてのノード１１０へと伝播される必要があってもよい。上述の通り、少なくともいくつかの実施形態において、ロードバランサノード１１０はゴシッププロトコルに従って、このヘルス情報を交換し伝播するため、またロードバランサノード１１０およびサーバノード１３０の故障を検知するために、互いに通信してもよい。

【０１３２】

少なくともいくつかの実施形態において、各ロードバランサノード１１０は通常の間隔（ゴシップ間隔と称する）で、別のロードバランサノード１１０を無作為に選択し、他のノード１１０にその正常なロードバランサノード１１０およびサーバノード１３０に関する見解をロードバランサノード１１０およびサーバノード１３０のためのハートビートカウンターとともに送信する。ロードバランサノードまたはサーバノード１３０が正常である限り、ノードはそのヘルスチェックにパスし、そのハートビートカウンターは増加し続ける。ノードのためのハートビートカウンターが特定の間隔（故障時間間隔と称されてもよい）で変化しない場合、ノードはその後ロードバランサノード１１０により故障を疑われる。ノードが故障を疑われると、ロードバランサノード１１０はノードが異常であると決定するまで特定の間隔（異常な時間間隔と称されてもよい）で待機してもよい。この異常な時間間隔により、ノードが故障したことをすべてのロードバランサノード１１０が把握するまでロードバランサノード１１０は待機することができる。

【０１３３】

図１６は少なくともいくつかの実施形態による、ロードバランサノード１１０により維持されてもよい別のノード（ロードバランサノード１１０またはサーバノード１３０のいずれか）のヘルス状態またはその見解を図示する。３００で示すように、ロードバランサノード１１０がまず、問題のノードが正常であるとの見解を有すると仮定する。これはノードのためのハートビートカウンターが増加しつつあることを示す。しかし、ノードのハートビートカウンターが３０２で示すように特定の間隔（故障時間間隔）で増加しない場合、ロードバランサノード１１０はその後３０４で示すように、ノードが故障したことを疑う。３０６で示すように、ノードのハートビートカウンターが特定の間隔（異常な時間間隔）で増加しない場合、ロードバランサノード１１０はその後３０８で示すように、ノードを異常であると見なす。しかしノードのためのハートビートカウンターが３１０で示すように、異常な時間間隔が終了する前に増加する場合、ロードバランサノード１１０は再びノードを正常３００であると見なす。同様に３１２で示すように、異常なノードのためのハートビートの増加を受信することでもノードは正常３００であると見なされうる。

【０１３４】

ノードの異常の決定には、本明細書で別に記載するように、異常なノードがロードバランサノード１１０またはサーバノード１３０のどちらであるかによって、また、ロードバランサノード１１０の異常なノードとの関係によって、１つまたは複数のロードバランサノード１１０による異なる行動を含んでもよい。

ロードバランサノードのデータ

【０１３５】

少なくともいくつかの実施形態において、各ロードバランサノード１１０はロードバラ

ンサの実装の状態に関するデータを維持してもよい。少なくともいくつかの実施形態において、このデータは各ロードバランサノード１１０上の、正常なロードバランサノードリスト、疑わしいロードバランサノードリスト、およびハートビートリストを含むがそれらに限定されない、１つまたは複数のデータ構成において維持されてもよい。図１７は正常なロードバランサノードリスト３２０、疑わしいロードバランサノードリスト３２２、異常なロードバランサノードリスト３２４、およびロードバランサノードのハートビートリスト３２６を維持するロードバランサノード１１０の実施例を示す。

【０１３６】

少なくともいくつかの実施形態において各ロードバランサノード１１０は、例えばどのノード１１０が正常であるか、またそれに従ってどのノード１１０がゴシッププロトコルに参加するかを決定するために用いられてもよい、正常なロードバランサノード１１０のリストである正常なロードバランサノードリスト３２０を維持してもよい。リスト３２０上のノード１１０のみがゴシッププロトコルを通じたロードバランサ情報の伝播に関与し、リスト３２０上のノード１１０のみが一貫したハッシュリング内にあると見なされ、このリスト上のノード１１０のみがサーバノード１３０のヘルスチェックを行う。ノード１１０はこのリスト３２０から、そのハートビート情報の送信先となる別のノード１１０を無作為に選択してもよい。またハートビートカウンターは、正常なロードバランサノードリスト３２０上に現在あるノード１１０のためのみに交換される。少なくともいくつかの実施形態において、ノードＮがロードバランサノード１１０によるヘルスチェックにパスする場合、またはロードバランサノード１１０がリスト３２０上のどれか他のロードバランサノード１１０からノードＮに関するゴシップメッセージを受信する場合に、ロードバランサノードＮは別のロードバランサノード１１０の正常なロードバランサノードリスト３２０に追加されることができる。

【０１３７】

少なくともいくつかの実施形態において、各ロードバランサノード１１０は、ハートビートカウンター（ハートビートリスト３２６を参照）が特定の間隔（故障時間間隔と称されてもよい）で増加しなかったロードバランサノードのリストである疑わしいロードバランサノードリスト３２２を維持してもよい。ロードバランサノードＥがロードバランサノード１１０の疑わしいロードバランサノードリスト３２２上にある場合、ロードバランサノード１１０はその後ノードＥについてゴシップしない。正常なリスト３２０上の他のどれかのロードバランサノード１１０が、ノード１１０のハートビートリスト３２６上でノードＥのためのカウンターより高いハートビートカウンターとともに、ロードバランサノード１１０にノードＥについてゴシップする場合、ノードＥはその後疑わしいリスト３２２から正常なリスト３２０へと移行される。ノードＥが特定の間隔（異常な時間間隔と称されてもよい）でロードバランサノード１１０の疑わしいリスト３２２上に留まる場合、ノードＥはロードバランサノード１１０により異常であると見なされ、異常なノードリスト３２４上に移行される。異常なノードリスト３２４上のノード１１０（この実施例のノードＧ）は、ノードＧがノード１１０によるヘルスチェックにパスした際、またはノードＧのための更新されたハートビートカウンターを別のノード１１０から受信した際に、ロードバランサノード１１０の正常なノードリスト３２０へと移行されてもよい。

【０１３８】

少なくともいくつかの実施形態において、各ロードバランサノード１１０はすべての既知のロードバランサノード１１０のためのハートビートリスト３２６を維持してもよい。各ノード１１０のために、このリスト３２６はハートビートカウンターおよび、ハートビートカウンターが最後に変更された時を示すタイムスタンプを含んでもよい。

【０１３９】

少なくともいくつかの実施形態において、各ロードバランサノード１１０はまた、図１７に図示されていないすべての既知のサーバノードのためのハートビートリストを維持してもよい。このリストはロードバランサノードのハートビートリスト３２６に類似していてもよい。いくつかの実施形態においては、２つのリストが組み合わされてもよい。少な

くともいくつかの実施形態において、サーバノード１３０のためのハートビート情報は、例えばゴシッププロトコルに従って、ロードバランサノード１１０のためのハートビート情報とともにあるいはそれに加えて、ロードバランサノード１１０間に伝播されてもよい。

【０１４０】

図１７は４つの個別のリストを示すが、リストは２つ以上が単一のリストに組み合わせられてもよいことに留意する。例えば、いくつかの実施形態においては、すべてのノード１１０の単一のリストが各ロードバランサノード１１０上に維持されてもよく、ビットフラッグまたは他のデータ構成は各ノードが現在正常、疑わしい、または異常のどれであるかを示すために用いられてもよい。

サーバノードデータ

【０１４１】

少なくともいくつかの実施形態において、ノード１３０上のサーバノード１３０およびローカルロードバランサモジュール１３２はロードバランサノード１１０を含むゴシッププロトコルに参加しない。ロードバランサノード１１０は、ロードバランサノードヘルスチェック方法により取得された他のロードバランサノード１１０に関するハートビート情報およびサーバノードヘルスチェック方法により取得されたサーバノード１３０に関するハートビート情報を、自身らの間でのみゴシップする（特に、各ロードバランサノード１１０はその正常なロードバランサノードリスト３２０上に現在あるノードにのみゴシップする）。

【０１４２】

しかし各サーバノード１３０／ロードバランサモジュール１３２は、サーバノード１３０が、サーバノード１３０が顧客のトラフィックを転送する先となるロードバランサノード１１０（特に、出口ノード）を決定し、どのロードバランサノードが接続公開情報の送信先となるかを決定できるように、ロードバランサの実装における正常なロードバランサノード１１０に関する情報を必要とする場合がある。少なくともいくつかの実施形態において、この情報をサーバノード１３０に提供するために、ロードバランサノード１１０は現在正常なロードバランサノード１１０（例、図１７の正常なロードバランサノードリスト３２０）を特定する情報を用いて、定期的または非定期的にサーバノード１３０を更新してもよい。少なくともいくつかの実施形態において、所与のサーバノード１３０のヘルスチェックの役割を担うロードバランサノード１１０（図１５を参照）は、現在正常なロードバランサノードを特定する情報をサーバ１３０へと提供する役割を担う。例えば図１５を参照すると、ロードバランサノード１１０Ａはその正常なロードバランサノードリスト３２０をサーバノード１３０Ａ、１３０Ｂ、１３０Ｃおよび１３０Ｄへと送信してもよく、ロードバランサノード１１０Ｂはその正常なロードバランサノードリスト３２０をサーバノード１３０Ｃ、１３０Ｄ、および１３０Ｅ等へと送信してもよい。

ロードバランサノードの故障の処理

【０１４３】

図１８Ａおよび１８Ｂは少なくともいくつかの実施形態による、ロードバランサノードの故障の処理を示す。図１８Ａはロードバランサの実装の実施例を示す。現在のロードバランサの実装において、４つのロードバランサノード１１０Ａ～１１０Ｄがある。エッジルータ１０４は顧客（図示せず）からの受信パケットをロードバランサノード１１０へとルーティングする。少なくともいくつかの実施形態において、エッジルータ１０４は、レイヤー４のフローごとにハッシュ化されたマルチパスルーティング技術、例えば等価コストマルチパス（ＥＣＭＰ）ルーティング技術に従って、ルーティングの決定を行ってもよい。少なくともいくつかの実施形態においてエッジルータ１０４は、ロードバランサノード１１０の提供、例えばロードバランサノード１１０によって開始された境界ゲートウェイプロトコル（ＢＧＰ）技術セッションを介した提供を通じて顧客のトラフィックを受信するために、ロードバランサの実装において現在利用可能なロードバランサノード１１０について把握する。しかし、少なくともいくつかの実施形態において、ＢＧＰセッション

を通じて自身をエッジルータ１０４に提供するロードバランサノード１１０の代わりに、ロードバランサの実装における少なくとも１つの他のノード１１０が、ＢＧＰを通じたノード１１０のエッジルータ１０４へ提供の役割を担う。例えば図１８Ａで示されるようにいくつかの実施形態においては、所与のノード１１０の左および右の近傍ノード１１０が、所与のノード１１０をエッジルータ１０４に提供する。例えばロードバランサノード１１０Ａはノード１１０Ｂおよび１１０Ｄを提供し、ロードバランサノード１１０Ｂはノード１１０Ａおよび１１０Ｃを提供し、ロードバランサノード１１０Ｃはノード１１０Ｂおよび１１０Ｄを提供する。

【０１４４】

図１８Ａの実施例において示されるように、各ロードバランサノード１１０はまた、例えば１つまたは複数の無作為に選択されたノード１１０、ロードバランサノードの番号付きリストにより決定された１つまたは複数の近傍ノード１１０、あるいは１つまたは複数の近傍ノードおよび１つまたは複数の無作為に選択されたノードなどの１つまたは複数の他のロードバランサノード１１０のヘルスチェックを定期的に行う。また各ロードバランサノード１１０は少なくとも１つのサーバノード１３０のヘルスチェックを定期的に行ってもよく、その正常なロードバランサノード１１０のリストをそれがヘルスチェックを行う１つまたは複数のサーバノードへと送信してもよい。ロードバランサノード１１０およびサーバノード１３０のためのヘルス情報は、例えばゴシッププロトコルに従ってノード１１０間に伝播されてもよい。

【０１４５】

図１８Ｂは、図１８Ａのロードバランサの実装の実施例における、単一のロードバランサノード１１０の故障の処理を示す。この実施例において、ロードバランサノード１１０Ｂは何らかの理由で故障している。例えば、ノード１１０Ａおよび１１０Ｃはノード１１０Ｂのヘルスチェックを行ってもよく、両方ともノード１１０Ｂがそのヘルスチェックに失敗していることを検知してもよい。したがって、ノード１１０Ａおよび１１０Ｃはノード１１０Ｂのためのハートビートカウンターを増加させない。ノード１１０Ａおよび１１０Ｂの両方からのハートビート情報は、ゴシッププロトコルに従って、他の正常なロードバランサノード１１０（この実施例においては、唯一の他のロードバランサノードはノード１１０Ｄである）へと伝播される。すべての正常なロードバランサノード１１０（この実施例においては、ノード１１０Ａ、１１０Ｃ、および１１０Ｄ）がノード１１０Ｂの故障において収束すると直ちに、以下のイベントの１つまたは複数が発生してもよいがそれらに限定されない。これらのイベントは必ずしもこの順序で発生するわけではないことに留意する。

＊ノード１１０Ａおよび１１０Ｃはエッジルータ１０４へのノード１１０Ｂの提供を停止する。少なくともいくつかの実施形態においてこれは、ノード１１０がエッジルータ１０４を用いてノード１１０Ｂを提供するために確立したＢＧＰセッションの終了に関連する。各ノード１１０は、提供を行う互いのノード１１０のためにエッジルータ１０４を用いて個別のＢＧＰセッションを確立し、そのためノード１１０ＢのためのＢＧＰセッションの終了は提供された他のノード１１０に影響を与えないことに留意する。少なくともいくつかの実施形態においてノード１１０は、ＢＧＰセッションのためのＴＣＰクローズまたは類似のメッセージをエッジルータ１０４に送信することにより、エッジルータ１０４を用いてＢＧＰセッションを終了する。

＊ノード１１０Ｂがどのノードからも提供されなくなったことを検知すると、それを受けてエッジルータ１０４が顧客データパケットのノード１１０Ｂへのルーティングを停止する。エッジルータ１０４はまた、顧客から残りの正常なロードバランサノード１１０、特にノード１１０上の入口サーバ１１２へのパケットフローを再分散するために、マルチパス（例、ＥＣＭＰ）ハッシュを調整する。入口サーバ１１２が顧客－＞サーバマッピングを有しない、入口サーバ１１２へとルーティングされたいずれかのパケットフローのために、マッピングが顧客－＞サーバ接続のためのフロートラッカーノードから取得されてもよく、またはその代わりに新規顧客－＞サーバ接続が図１０Ａ～１０Ｇで示される技術

に従って確立されてもよい。
＊ノード１１０Ａおよび１１０Ｃはそれぞれ、互いを提供するためのエッジルータ１０４へのＢＧＰセッションを開始してもよい。ノード１１０Ａおよび１１０Ｃは両方ともノード１１０Ｂと同様にロードバランサノード１１０Ｄによってエッジルータ１０４へと提供されるため、ノード１１０Ｂが故障時にノード１１０Ａおよび１１０Ｂのエッジルータ１０４への提供を停止するかもしれないという事実が、エッジルータ１０４からのこれら２つのノード１１０へのパケットのルーティングに停止を引き起こさないことに留意する。
＊少なくともいくつかの実施形態において、ノード１１０Ａおよび１１０Ｃはこの時点では近傍ノード１１０であるため、互いのヘルスチェックの役割を担ってもよい。ノード１１０Ｂは異常であると見なされてもなお、１つまたは複数の他のノード１１０により無作為なヘルスチェックを行われてもよいことに留意する。
＊１つまたは複数の残りの正常なロードバランサノード１１０は、以前はノード１１０Ｂによりトラッキングされていたフローのトラッキング接続の役割を担ってもよい。例えばノード１１０Ｃおよび／またはノード１１０Ｄは、図１１Ｃおよび１１Ｄで示されるように、ノード１１０Ｂが１次または２次フロートラッカーの役割を担っていた１つまたは複数の接続のための１次または２次フロートラッカーとしての役割を引き継いでもよい。
＊１つまたは複数の残りの正常なロードバランサノード１１０は、ノード１１０Ｂによりすでにヘルスチェックを行われたサーバノード１３０のヘルスチェックの役割を担ってもよい。サーバノード１３０は正常なロードバランサノードリスト（この時点ではノード１１０Ｂを含まない）を用いて残りのロードバランサノード１１０により更新される。例えば図１８Ｂにおいて、ロードバランサノード１１０Ａはサーバノード１３０Ｃのヘルスチェックおよび更新を開始し、ロードバランサノード１１０Ｃはサーバノード１３０Ｂのヘルスチェックおよび更新を開始する。
＊エッジルータ１０４上で、故障したノード１１０ＢからのＢＧＰセッションが最終的にタイムアウトする。その代わりにエッジルータ１０４は、ノード１１０Ｂの故障の認識時にＢＧＰセッションを遮断してもよい。
【０１４６】
２つのロードバランサノード１１０が同時に、またはほぼ同時に故障する可能性がある。２つの故障したロードバランサノードが互いに隣接していない場合、故障は独立となり、個別の単一のノード１１０の故障として図１８Ｂで示す方法に従って処理されてもよい。しかし、２つの故障したノードが互いに隣接し（例、図１８Ａにおけるノード１１０Ｂおよび１１０Ｃ、その後直ちにすべての正常なロードバランサノード１１０（この実施例では、ノード１１０Ａおよび１１０Ｄ）が故障を検知し、故障上で収束する場合に、以下のイベントの１つまたは複数が発生してもよいがそれらに限定されない。これらのイベントは必ずしもこの順序で発生するわけではないことに留意する。
＊ノード１１０Ａはノード１１０Ｂのためのエッジルータ１０４へのＢＧＰセッションを終了する。
＊ノード１１０Ｄはノード１１０Ｃのためのエッジルータ１０４へのＢＧＰセッションを終了する。
＊ノード１１０Ａおよび１１０Ｄはエッジルータ１０４を用いた互いを提供するＢＧＰセッションを開始する。
＊ノード１１０Ａおよび１１０Ｄは互いのヘルスチェックを開始する。ノード１１０Ａおよび１１０Ｄはまた、故障したノード１１０のヘルスチェックを継続してもよいことに留意する。
＊残りの正常なノード１１０は正常なロードバランサノードリストを用いてサーバノード１３０を更新する。
＊トラフィックはエッジルータ１０４からノード１１０Ｂおよび／またはノード１１０Ｃへと継続して流れてもよい。これは、これら２つのノード１１０がエッジルータ１０４への互いの提供を継続してもよいからである。しかしこれらのＢＧＰセッションは最終的

にタイムアウトし、エッジルータ１０４はフローを残りの提供されたノード１１０へと適宜、再分散させる。

＊ノード１１０Ｂおよび１１０Ｃは、ノード１１０Ｂおよび１１０Ｃがいまだに正常であると判断する場合に、ノード１１０Ａおよび１１０Ｄをそれぞれ提供する、エッジルータ１０４を用いたＢＧＰセッションを終了してもよい。

接続公開

【０１４７】

図１を再度参照すると、少なくともいくつかの実施形態において、ロードバランサの実装におけるロードバランサノード１１０がサーバ１３０への顧客ＴＣＰ接続のための状態情報を維持する。この状態情報によりロードバランサノード１１０が、顧客の受信トラフィックをエッジルータ１０４からＴＣＰ接続の役割を担うサーバノード１３０へとルーティングできる。サーバノード１３０上のロードバランサモジュール１３２はそれぞれのサーバ１３４へのアクティブなＴＣＰ接続のリストを維持する。接続公開は、サーバノード１３０上のロードバランサモジュール１３２が、それを通じてロードバランサノード１１０へのアクティブな顧客ＴＣＰ接続のリストを公開してもよい機構である。少なくともいくつかの実施形態において、接続公開データパケットはロードモジュール１３２によりロードバランサノード１１０へと、接続公開間隔と称されてもよい通常の間隔で形成され、公開される。

【０１４８】

少なくともいくつかの実施形態において、ロードバランサノード１１０により維持された接続状態情報はキャッシュの形式として見なされてもよく、特定の接続のための状態情報の維持はその接続のためのロードバランサノード１１０上のリースの維持とみなされてもよい。キャッシュエントリが更新されない限りは、ロードバランサノード１１０はデータフローを処理するサーバノード１３０への顧客データフローをルーティングすることができない可能性がある。接続公開機構はロードバランサノード１１０上でサーバノード１３０からの現在の接続状態情報を用いて、定期的にキャッシュを、そしてその結果リースを更新し、こうしてＴＣＰデータパケットを顧客１６０から適切なサーバノード１３０へと流し続ける。顧客１６０がサーバ１３４へのＴＣＰ接続を終了する際、その接続に関連するサーバノード１３０上のロードバランサモジュール１３２はそのアクティブな接続のリストからのリストを破棄し、したがって接続公開機構を通したＴＣＰ接続を公開することはなくなる。こうして、その接続に関連するロードバランサノード１１０（特に、接続のための入口サーバ１１２ならびに１次および２次フロートラッカー１１６）上の、その接続のための接続状態情報（キャッシュエントリまたはエントリ）は更新されなくなり、接続はロードバランサノード１１０により破棄される。少なくともいくつかの実施形態において、接続のためのキャッシュエントリまたはエントリは、他のどれかのアクティブな接続によりメモリを要求されるまで、ロードバランサノード１１０上のキャッシュを保持してもよい。

【０１４９】

したがって接続公開機構は定期的または非定期的に、入口サーバ１１２ならびに１次および２次フロートラッカー１１６上で接続リースを延長し、顧客のトラフィックを流れさせる。また、接続公開機構は少なくともいくつかのロードバランサノード１１０の故障からの回復に貢献してもよい。顧客接続のための状態情報を保持する１つまたは複数のロードバランサノード１１０が故障した場合、接続公開によって残りのロードバランサノード１１０へと提供されたアクティブな接続情報は場合によっては接続の回復に用いられてもよい。

【０１５０】

接続公開機構を用いて、サーバ１３４と顧客１６０との間の接続状態のためのサーバノード１３０は信頼すべきソースとなる。またサーバ１３４への接続の終了は、サーバノード１３０上のロードバランサモジュール１３２およびロードバランサノード１１０によって受動的に処理される。サーバノード１３０とロードバランサノード１１０との間にハン

ドシェイクは要求されない。すなわちロードバランサモジュール１３２は、特定の接続が終了したことを積極的にノードに通知するためにメッセージをロードバランサノード１１０に送信する必要がない。サーバ１３４が接続を終了する際、サーバ１３４は接続のための内部状態をクリアーする。ロードバランサモジュール１３２はサーバ１３４の内部状態を用いて、接続公開パケットを組み込む。接続がサーバ１３４の内部状態からなくなるため、接続はロードバランサノード１１０へと公開されない。ロードバランサノード１１０上の接続のためのリースはこうして終了し、ロードバランサノード１１０は接続に関して受動的に忘れる。接続に用いられたロードバランサノード１１０のキャッシュ内のメモリはその後必要に応じて、他の接続のために用いられることが可能である。

【０１５１】

いくつかの実施形態においては、ロードバランサノード１１０により維持された接続のためのリースにはキャッシュ内の接続のためのタイムスタンプのエントリが関係してもよい。接続のリースが接続公開パケットにより更新された際に、タイムスタンプは更新されてもよい。サーバノード１３０上のロードバランサモジュール１３２によって接続が公開されなくなったために接続のリースが更新されない場合には、タイムスタンプはその後更新されなくなる。少なくともいくつかの実施形態において、遅延ガベージコレクション方法が用いられ、メモリが必要になるまで接続のためのエントリがキャッシュ内に保持されてもよい。例えば少なくともいくつかの実施形態において、キャッシュエントリ内のタイムスタンプがリース更新時間閾値と比較されてもよい。キャッシュエントリのためのタイムスタンプが閾値より遅い時間である場合、エントリはその後古くなり再利用されてもよい。しかしいくつかの実施形態においては、古いエントリは積極的にガベージコレクションされてもよい。

接続公開受信者

【０１５２】

少なくともいくつかの実施形態において、各顧客ＴＣＰ接続のために接続状態を維持する３つのロードバランサノード１１０がある－入口サーバ１１２の役割を担うノード１１０、１次フロートラッカー１１６の役割を担うノード１１０、および２次フロートラッカー１１６の役割を担うノードである。一貫したハッシュリングにおいて１次フロートラッカー１１６ノードおよびその後続ノードを見つけるためのＴＣＰフローに対する一貫したハッシュ関数を適用することにより、所与のＴＣＰフローのために、例えばロードバランサノード１１０によって１次および２次フロートラッカー１１６を決定することができる。ＴＣＰフローのための入口サーバ１１２の役を担うロードバランサノード１１０は、エッジルータ１０４の内部マルチパス（例、ＥＣＭＰ）ハッシュ関数に基づき、エッジルータ１０４からそのフローのためのトラフィックを受信するノード１１０である。ノード１１０の故障または追加がある場合には、入口サーバ１１２の役割を担うロードバランサノード１１０は多くのアクティブなＴＣＰフローのために変更されてもよく、少なくともいくつかのアクティブなＴＣＰフローのためのフロートラッカーの役割を担うロードバランサノード１１０が変更されてもよい（例、図１１Ａ～１１Ｄを参照）。サーバノード１３０上のサーバ１３２へのすべてのＴＣＰフローのために、そのサーバノード１３０上のロードバランサモジュール１３２は状態情報を維持し、入口サーバ１１２がロードバランサノード１１０からのトラフィックを受信することから、どのロードバランサノード１１０がそのＴＣＰフローのための入口サーバ１１２であるかを示す。しかし少なくともいくつかの実施形態において、ロードバランサモジュール１３２は使用された一貫したハッシュ関数を把握していなくてもよいため、どのロードバランサノード１１０がＴＣＰフローのための１次および２次フロートラッカーの役割を担っているかを、ロードバランサモジュール１３２は把握しなくてよく、また決定できなくてもよい。すなわち、少なくともいくつかの実施形態において、ロードバランサモジュール１３２は一貫したハッシュを行わない。

アクティブな接続情報の公開

【０１５３】

図１９Ａおよび１９Ｂは少なくともいくつかの実施形態による、接続公開技術を図示する。図１９Ａはアクティブな接続情報をロードバランサノードに公開するロードバランサ（ＬＢ）モジュールを示す。少なくともいくつかの実施形態において、各ロードバランサモジュール１３２はサーバノード１３０上の各アクティブなＴＣＰフローのための情報を収集し、接続公開パケットを形成する。所与のＴＣＰフローのための情報は、フローのための入口サーバ１１２を担うロードバランサノード１１０を特定する情報を含む。接続公開パケットがレディ状態である際（例、接続公開間隔に達した際）、ロードバランサモジュール１３２はロードバランサノード１１０を、例えばすでに述べたようにサーバノード１３０のヘルスチェックを行うロードバランサノード１１０からサーバノード１３０へと定期的に送信される正常なロードバランサノード１１０のリストから、無作為に選択する。ロードバランサモジュール１３２はその後接続公開パケットを選択したノード１１０へと送信する。例えば図１９Ａでは、ロードバランサモジュール１３２Ａがある接続公開パケットをロードバランサノード１１０Ａに送信し、後ほど別の接続公開パケットをロードバランサノード１１０Ｂに送信する。

【０１５４】

図２０は少なくともいくつかの実施形態による、各ロードバランサモジュール１３２により実行されてもよい接続公開方法のハイレベルフローチャートである。５００で示すようにロードバランサ（ＬＢ）モジュール１３２は、それぞれのサーバノード１３０上のすべてのアクティブなＴＣＰフローのための接続公開エントリを作成する。少なくともいくつかの実施形態において、ロードバランサモジュール１３２はサーバノード１３０上のサーバ１３４が処理するアクティブなＴＣＰ接続の組を、例えばサーバノード１３０上の／ｐｒｏｃ／ｎｅｔ／ｔｃｐから取得する。すべてのアクティブなＴＣＰ接続のために、ロードバランサモジュール１３２は（例、ローカルで維持されたアクティブな接続のテーブル内で）ＴＣＰフローのための入口サーバ１１２の役割を担うロードバランサノード１１０を検索し、接続のためのＴＣＰタプル（例、顧客のＩＰアドレス、顧客用ポート、サーバ（パブリック）ＩＰアドレス、およびサーバポートから成る４タプル）および接続のための入口サーバ１１２を示す接続公開エントリを作成する。各ロードバランサモジュール１３２は、接続のために受信されたパケットを最後に送信したロードバランサノード１１０を示す、各アクティブなＴＣＰ接続のための情報を維持すること、また、この情報が各アクティブな接続のための入口ノード１１０を特定するために、ロードバランサモジュール１３２によって用いられてもよいことに留意する。

【０１５５】

５０２で示すように、ロードバランサモジュール１３２は接続公開パケット（１つまたは複数の接続公開エントリを含み、各アクティブなＴＣＰ接続毎に１つのエントリ）の送信先となるロードバランサノード１１０を無作為に選択する。少なくともいくつかの実施形態において、接続公開パケットの送信準備が完了しているとロードバランサモジュール１３２が決定した際に、ロードバランサモジュール１１０は無作為に選択されてもよい。少なくともいくつかの実施形態において、この決定は接続公開間隔に従って下される。非限定的な実施例として、接続公開間隔は１００ミリ秒（ｍｓ）、または１秒であってもよい。少なくともいくつかの実施形態において、ロードバランサノード１１０の内の１つからすでに受信された正常なロードバランサノード１１０のリストから、ロードバランサモジュール１１０が選択される。５０４で示すように、ロードバランサモジュールはその後接続公開パケットを、選択されたロードバランサノード１１０に公開する。少なくともいくつかの実施形態において、接続公開パケットは例えばＵＤＰパケットといったステートレスパケットである。いくつかの実施形態において接続公開パケットは、対象のロードバランサノード１１０へのパケットの送信前に圧縮されてもよい。少なくともいくつかの実施形態において接続公開情報は、２つ以上のパケットにおける対象のロードバランサノード１１０へと送信されてもよい。

【０１５６】

要素５０４から要素５００へと戻る矢印が示すように、ロードバランサモジュール１３

２は継続的に接続公開データパケットを作成し、無作為なノード１１０を選択し、そしてパケットを選択したノードへと送信してもよい。上述のように、ロードバランサノード１１０がロードバランサノード１１０上の接続リースを維持するための現在のアクティブな接続情報を用いて比較的定期的にリフレッシュされるように、これは接続公開間隔に従って行われてもよい。

【０１５７】

少なくともいくつかの実施形態において、接続公開データパケットはロードバランサモジュールによってロードバランサノード１１０へと無作為に分散されるため、接続公開パケットを受信するロードバランサノード１１０は、接続公開データパケットにおけるアクティブな接続情報の接続のための正しい入口／１次／２次ノード１１０への分散の役割を担う。図１９Ｂならびに図２１および２２は、少なくともいくつかの実施形態において用いられてもよいアクティブな接続情報の分散方法を示す。

【０１５８】

図１９Ｂは少なくともいくつかの実施形態による、ロードバランサノード１１０間のアクティブな接続情報の分散を示す。ロードバランサノード１１０がロードバランサモジュール１３２から接続公開パケットを受信する際、ロードバランサノード１１０は、フローのための入口ノードならびに１次および２次フロートラッカーノードを決定するために、その中に示された各ＴＣＰフローのための情報を分析してもよい。ロードバランサノード１１０が、フローのためのそれら役割のうち１つを担っている場合、ロードバランサノード１１０はフローのための情報を（例、その状態情報のキャッシュの更新により）消費する。少なくともいくつかの実施形態において、ロードバランサノード１１０はまたフローのための情報を、フローのための他の役割を担っている１つまたは複数の他のノード１１０へと送信される１つまたは複数のパケットフローに盛り込んでもよい。接続公開パケットにより示される残りのフローのために、ロードバランサノード１１０はアクティブな接続情報を２つ以上のよりデータ量の少ないパケットに分割して各パケットを１つまたは複数の他のロードバランサノード１１０に送信する。例えば少なくともいくつかの実施形態において、１つまたは複数のフローのためにアクティブな接続情報を含むパケットは、１つまたは複数のフローのための入口サーバ１１２、１次フロートラッカー１１６Ａ、および２次フロートラッカー１１６Ｂの役割を担うロードバランサノード１１０に送信されてもよい。

【０１５９】

図２１は、少なくともいくつかの実施形態による、対象のロードバランサノード１１０への接続公開パケットにおいて受信されるアクティブな接続情報の分散方法のフローチャートである。５２０で示すように、ロードバランサノード１１０はロードバランサモジュール１３２から接続公開パケットを受信する。ロードバランサモジュール１３２は、例えば図１９Ａおよび２０に関連して上述したようにパケットを生成し、パケットを受信するためのロードバランサノード１１０を選択した。接続公開パケットは、受信されたパケットの送信元であるサーバノード１３０を特定する情報（例、サーバノード１３０上のロードバランサモジュール１３２のＩＰアドレス）およびアクティブなＴＣＰ接続を特定するエントリのリスト（例、各接続のための顧客のＩＰアドレス、顧客用ポート、サーバ（パブリック）ＩＰアドレス、およびサーバポートから成る４タプル）を含んでもよい。

【０１６０】

図２１の要素５２２～５３０において、ロードバランサモジュール１１０は受信された接続公開パケットで示されるアクティブなＴＣＰ接続情報を繰り返し処理する。５２２で示すようにロードバランサノード１１０は、それぞれのＴＣＰフローのための入口ノード１１０ならびに１次および２次フロートラッカーノード１１０を決定するために、パケット内の次のＴＣＰフローのためのエントリを分析する。少なくともいくつかの実施形態において、ロードバランサノード１１０は接続公開エントリから入口ノード１１０を特定する。少なくともいくつかの実施形態において、ＴＣＰフローのための１次および２次フロートラッカーノード１１０は一貫したハッシュ関数に従って決定されてもよい。５２４に

おいてロードバランサノード１１０が検証中のＴＣＰフローのための役割のうちの１つを担う場合、その後５２６においてロードバランサノード１１０がフローのための情報を、例えばその状態情報のキャッシュの更新によって消費する。５２８で示すように、ロードバランサノード１１０はＴＣＰフローのための接続公開エントリを、構成中であり別のロードバランサノード１１０に送信される予定のパケットに追加してもよい。５３０において接続公開パケット内にフローのための接続公開エントリがさらにある場合は、その後メソッドは５２２へと戻り、次のエントリの処理を行う。そうでなければロードバランサノードは、それぞれがオリジナルの接続公開パケットからの接続公開エントリのサブセットを含む１つまたは複数の新規構成されたパケットを、５３２で示すようにパケットのための対象のロードバランサノード１１０へと送信する。少なくともいくつかの実施形態において、対象のロードバランサノード１１０に送信されるパケットは、例えばＵＤＰデータパケットといったステートレスパケットである。いくつかの実施形態において、パケットを対象のロードバランサノード１１０に送信する前に、パケットは圧縮されてもよい。

【０１６１】

このように少なくともいくつかの実施形態において、図２１の要素５２２～５２８ではフロートラッカーノード１１０が、受信された接続公開パケット内の接続公開エントリから５２２で決定された情報に従って他のノード１１０のうちの特定の１つへとそれぞれ送信されることになる１つまたは複数のパケット（例、ＵＤＰパケット）を構築する。少なくともいくつかの実施形態において、別のノード１１０に送信されるパケットは、対象のノード１１０が入口ノード１１０、１次フロートラッカーノード１１０、または２次フロートラッカーノード１１０としての役割を担うＴＣＰフローのためのエントリを含む。いくつかの実施形態においては、所与のロードバランサノード１１０がＴＣＰフローのための入口および１次フロートラッカーノードの両方の役割を担ってもよく、またはＴＣＰフローのための入口および２次フロートラッカーノードの両方の役割を担ってもよいことに留意する。

【０１６２】

図２２は少なくともいくつかの実施形態による、対象のロードバランサノード１１０への接続公開パケットにおいて受信されるアクティブな接続情報の分散の代替方法を示す。５５０で示すように、ロードバランサノード１１０はロードバランサモジュール１３２から接続公開パケットを受信する。この方法では、５５２で示すように、ロードバランサモジュール１１０上の処理によってパケット内の接続公開エントリの分析が行われ、受信されたパケットの１つまたは複数のよりデータ量の少ないパケットへの分割が適宜行われる。ロードバランサモジュール１１０はこの処理の間、ローカルでフロー情報を消費しない。接続公開パケットが１つまたは複数のパケットに分割されると、パケットがその後５５４～５６０で示すように処理される。５５４においてパケットのための対象のノード１１０がこのロードバランサノード１１０である場合、ロードバランサノード１１０はその後５５６で示すようにローカルでパケットを消費する。そうでなければパケットは対象のロードバランサノード１１０へと送信される。５６０において処理すべきパケットがさらにあれば、その後メソッドは５５４に戻る。そうでなければメソッドは完了する。

【０１６３】

このようにしてロードバランサモジュール１３２から接続公開パケットを受信するロードバランサノード１１０は、接続公開パケットを特定の他のロードバランサノード１１０に特有の２つ以上のよりデータ量の少ないパケットに分割し、ロードバランサノード１１０により現在処理中であるいずれかのＴＣＰフローのためのフロー情報を内部で消費しながらパケットを適宜分散してもよい。その間、他のロードバランサノード１１０もまた、接続公開パケットをロードバランサモジュール１３２から受信し、接続公開エントリを複数のよりデータ量の少ないパケットに分割し、そしてよりデータ量の少ないパケット対象のノード１１０に送信して、ノード１１０間にアクティブな接続情報を分散してもよい。

接続公開トリガ

【０１６４】

少なくともいくつかの実施形態において、接続公開はロードバランサモジュール１３２上で１つまたは複数の異なるイベントによってトリガされてもよい。上述のようにいくつかの実施形態において、接続公開パケットはロードバランサノード１１０上のＴＣＰ接続のためのリースを更新するべく、接続公開間隔、例えば１００ｍｓまたは１秒間隔に従って生成され、無作為に選択されたロードバランサノード１１０へと送信されてもよい。いくつかの実施形態においては、ロードバランサノード１１０のメンバーシップにおける変更は、即時の接続公開イベントをトリガしてもよい。少なくともいくつかの実施形態において、ロードバランサモジュール１３２はそれぞれのサーバノード１３０のヘルスチェックを行うロードバランサノード１１０の１つより送信された正常なロードバランサノード１１０のリストから変更について学習してもよい。リストに従った変更（削除または追加のいずれか）の検知時には、変更に影響されたＴＣＰ接続がロードバランサノード１１０によって迅速に回復できるように、ロードバランサモジュール１３２は接続公開パケットを生成しロードバランサノード１１０へと送信してもよい。

パケットループの阻止

【０１６５】

接続公開パケットの処理の間にロードバランサレイヤーのメンバーシップが変更された場合、接続公開パケットループが発生してもよい。第１のノード１１０はロードバランサモジュール１３２から接続公開パケットを受信し、よりデータ量の少ないパケットを第２のノード１１０へと送信してもよい。しかしメンバーシップが変更された場合には、パケットが第１のノード１１０へ移行するべきであると第２のノード１１０が決定してもよく、その結果パケットを第１のノード１１０へと転送してもよい。少なくともいくつかの実施形態において、このループの発生を阻止するためにロードバランサモジュール１３２から受信された接続公開パケットと、ロードバランサノード１１０から受信されたそれらとで異なるポート番号が用いられてもよく、またロードバランサノード１１０は他のロードバランサノード１１０から受信された接続公開パケットの再分散を行わない。

接続公開パケット分散の代替方法

【０１６６】

上記の接続公開方法において、ロードバランサモジュール１３２は接続公開パケットの送信先であるロードバランサノード１１０を無作為に選択する。しかしいくつかの実施形態においては、ロードバランサノード１１０の選択に他の方法が用いられてもよい。例えばいくつかの実施形態において、ロードバランサノード１３２は、１つまたは複数のアクティブなＴＣＰフローの処理を行う特定の入口ノード１１０をそれぞれ対象とする１つまたは複数の接続公開データパケットを構築してもよく、また、１つまたは複数のパケットを１つまたは複数の対象の入口ノード１１０へと送信してもよい。１つまたは複数の入口ノード１１０はその後、アクティブな接続情報を接続のための１次および２次フロートラッカーへと再分散する。別の実施例として、いくつかの実施形態においては、接続公開パケットの単一の、無作為に選択されたノード１１０への送信の代わりに、各接続公開パケットがロードバランサモジュール１３２によって２つ以上の正常なノード１１０へと、またはすべての正常なノード１１０へと送信されてもよい。

ロードバランサノードアーキテクチャ

【０１６７】

図２３は少なくともいくつかの実施形態によるロードバランサノード１１０のためのソフトウェアスタックアーキテクチャの実施例を示し、限定的な意図を持たない。このソフトウェアスタックアーキテクチャの実施例においては、Ｊａｖａ Ｎａｔｉｖｅ Ｉｎｔｅｒｆａｃｅ（ＪＮＩ（商標））１１０４技術を用いて、ロードバランササーバネイティブコード１１０６およびコアパケット処理コード１１０８、例えばＩｎｔｅｌ（商標）Ｄａｔａｐｌａｎｅ Ｄｅｖｅｌｏｐｍｅｎｔ Ｋｉｔ（ＤＰＤＫ）技術コードを含んでもよいネイティブコードのレイヤーを管理する、単一のＪａｖａ（商標）技術処理１１０２内でロードバランサノード１１０が動作する。ネイティブコードは２つのネットワークインタフェースコントローラ（ＮＩＣ１１１４Ａおよび１１１４Ｂ）へのインターフェー

スとなってもよい。第１のＮＩＣ（ＮＩＣ１１１４Ａ）は「北」、すなわちエッジルータ１０４向きに面してもよい。第２のＮＩＣ（ＮＩＣ１１１４Ｂ）は「南」、すなわちサーバノード１３０向きに面してもよい。少なくともいくつかの実施形態において、ＮＩＣ１１１４Ａおよび１１１４ＢはＴＣＰスタックを維持しなくてもよい。したがって少なくともいくつかの実施形態は、ロードバランサノード１１０が制御プレーンを通じた処理との通信を行うことができ、またその逆も可能になるようにＴＣＰ接続をサポートする第３のＮＩＣ１１１４Ｃを含んでもよい。その代わりに、いくつかの実施形態においては、第１の、北向きのＮＩＣ１１１４Ａおよび第２の、南向きのＮＩＣ１１１Ｂのみがロードバランサノード１１０において実装されてもよく、また第２の、南向きのＮＩＣ１１１４ＢがＴＣＰスタックを実装してもよい。ロードバランサノード１１０はＴＣＰスタックを通じて制御プレーンを通じた処理との通信を行ってもよい。ロードバランサノード１１０はまた、オペレーティングシステム（ＯＳ）技術ソフトウェア１１１２、例えば、Ｌｉｎｕｘ（商標）カーネル、およびＯＳ技術ソフトウェア１１１２上のＪａｖａ　Ｖｉｒｔｕａｌ　Ｍａｃｈｉｎｅ（ＪＶＭ（商標））技術ソフトウェア１１１０レイヤーそしてＪＮＩ１１０４技術を含む。

【０１６８】

少なくともいくつかの実施形態において、分散型ロードバランスシステム内のロードバランサノード１１０はそれぞれ多くのデータフローを高パケットレートで同時に処理しなければならない場合がある。少なくともいくつかの実施形態において、要求レベルのスループットを達成するために、ロードバランサノード１１０が高性能パケット処理のためのＩｎｔｅｌ（商標）　Ｄａｔａｐｌａｎｅ　Ｄｅｖｅｌｏｐｍｅｎｔ　Ｋｉｔ（ＤＰＤＫ）技術を活用してもよい。ＤＰＤＫ技術により、ユーザースペースプログラムがネットワークインタフェースコントローラ（ＮＩＣ）から直接パケットを読み取ること／ネットワークインタフェースコントローラ（ＮＩＣ）へと直接パケットを書き込むことが可能になり、またＤＰＤＫ技術がＬｉｎｕｘカーネルネットワークスタックの多くのレイヤーを（Ｌｉｎｕｓ　ｉｘｇｂｅ基本ＮＩＣドライバを除いて）バイパスする。パケット処理のためのＤＰＤＫ手法は、ビジーループにおいてＮＩＣハードウェアに直接ポーリングを行う専用ＣＰＵコアのための割り込みハンドラーを利用した入力を拒否する。この手法は、専用ＣＰＵコアをビジーループにおいて継続的に動作させることで、熱出力の増加と引き換えにはるかに高いパケットレートを可能にする。ＤＰＤＫ技術はまた、ＣＰＵコア管理、ロックフリーのキュー、メモリプールおよび同期プリミティブを含むパケット処理ツールも提供する。図２４で示されるように、ＤＰＤＫ技術においては、専用ＣＰＵコア６００が各特定のタスクのために用いられてもよく、また非停止キュー６０２を用いて、あるＣＰＵコア６００Ａから別のＣＰＵコア６００Ｂへと作業が渡される。

【０１６９】

ＤＰＤＫキュー６０２は高速の２のべき乗リングバッファを用いて実装されてもよく、単一および複数の生産者／消費者バリアントのサポートを行ってもよい。複数の生産者／消費者バリアントは、アクセスを同期するためにコンペアアンドスワップ（ＣＡＳ）ループを含むため、真の意味でロックフリーではない。バッファへのポインタのみが読み取られキュー６０２に書き込まれるように、すべてのパケットバッファメモリはメモリプール内に事前割振りされてもよい。メモリプールはキューとして実装されてもよく、メモリチャネルやメモリランクをまたいでメモリを分散させるため最適化されてもよく、非一様性メモリアクセス（ＮＵＭＡ）により最適化された割り当てをサポートしてもよい。少なくともいくつかの実施形態において、パケットバッファにはバッファコピーを要求せずに外部ネットワークレイヤーヘッダーを追加／取り除くことができるカプセル化／脱カプセルの動作をサポートするために、各パケットバッファのヘッドルームおよびテールルームに過剰な割り当てを行うＭｂｕｆパラダイムのような方法を用いてもよい。

【０１７０】

ロードバランサノード１１０の少なくともいくつかの実施形態において、ＤＰＤＫ技術を活用するコアパケット処理アーキテクチャが実装されてもよい。各ロードバランサノー

ド１１０はコアパケット処理アーキテクチャに従って、少なくとも１つの実装されたマルチコアパケットプロセッサを含んでもよい。コアパケット処理アーキテクチャはパケットフローのための、マルチコアパケットプロセッサのキューおよびコアを通じた単一の生産者／単一の消費者パラダイムを用いてもよい。このパラダイムでは、各キューはただ１つのコアへの入力を行い、各コアはただ１つのコアへの出力を互いに行い、パケットを与える。また、マルチコアパケットプロセッサにおいてコアにより用いられたメモリは共有されない。各コアにはそれ自体の別のメモリ領域がある。したがって、コア間にはメモリやキューの共有はなく、メモリやキューの競合もなく、メモリやキューにオーナーシップの要求（ＲＦＯ）やコンペアアンドスワップ（ＣＡＳ）等の機構の共有の必要もない。図２５および２６にはコアパケット処理アーキテクチャに従って実装されるマルチコアパケットプロセッサの実施例が示される。

【０１７１】

図２５は少なくともいくつかの実施形態による、データフローの処理のためにＤＰＤＫ技術を活用するコアパケット処理アーキテクチャに従って実装されたマルチコアパケットプロセッサの実施例を示す。コアパケット処理アーキテクチャは単一の生産者／単一の消費者パラダイムに従って、マルチコアパケットプロセッサとして実装されてもよい。少なくともいくつかの実施形態において、図２３に示されるように、ロードバランサノード１１０はそれぞれ２つのネットワークインタフェースコントローラ（ＮＩＣ）を有する－境界ネットワーク／エッジルータ１０４に面する北向きのＮＩＣ１１１４Ａおよび本番ネットワーク／サーバノード１３０に面する南向きのＮＩＣ１１１４Ｂである。少なくともいくつかの実施形態において、ＮＩＣ１１１４は１０ＧｐｂｓＮＩＣであってもよい。ロードバランサノード１１０を通過するパケットの大部分がこれら２つのＮＩＣ（ＮＩＣ１１１４Ａまたは１１１４Ｂのいずれか）で受信され、処理され（例、カプセル化され、あるいは脱カプセル化され）、一方のＮＩＣ（ＮＩＣ１１１４Ｂまたは１１１４Ａのいずれか）に伝達される。

【０１７２】

図２５を参照すると、少なくともいくつかの実施形態においてロードバランサノード１１０は２つのＣＰＵコア、受信（ＲＸ）コア６１０および伝達（ＴＸ）コア６３０を各ＮＩＣ１１１４のためにスピンアップする。ロードバランサノード１１０も両方のＮＩＣ１１１４のための両方向のパケットの処理を行ういくつかの作業者コア６２０を行う。この実施例では４つの作業者コア６２０Ａ～６２０Ｄが用いられる。受信コア６１０は入力キューからの受信パケットのバッチを、それらがＮＩＣ１１１４に到着する際に読み取り、各パケットのための作業の大部分を行う作業者コア６２０へとパケットを分散させ、各受信コア６１０はパケットを各作業者コア６２０のためのそれぞれの作業者入力キュー６１２に与える。少なくともいくつかの実施形態において、受信コア６１０は各受信パケットにおいてレイヤー４「フローハッシュ」技術（上述のエッジルータ１０４により用いられてもよいフローごとにハッシュ化されたマルチパスルーティング技術に類似）を実行し、いずれの特定の顧客接続（ＩＰアドレスおよびポートにより区別される）も同一の作業者コア６２０によって確実に処理されるようにしながらパケットを作業者コア６２０へと分散させてもよい。これは、各作業者コア６２０が常に同一のパケットのサブセットを見てもよいということを意味し、ロックを要求されないように作業者コア６２０が管理する状態データにおける競合を除去する。受信されたパケットへのポインタは作業者コア６２０が継続的に新規入力のために監視する作業者キュー６２２間に分散されてもよい。作業者コア６２０は各接続のための状態の管理の役割を担い（例えば、割り当てられたサーバノード１３０）、パケットをアウトバウンドキュー６３２の１つに転送する前のパケット上のＵＤＰのカプセル化または脱カプセル化を行ってもよい。伝達コア６３０は作業者コア６２０アウトバウンドキュー６３２を通って循環し、出力パケットを、キュー６３２に現れるときに対応するＮＩＣ１１１４に書き込む。

【０１７３】

図２６は少なくともいくつかの実施形態による、データフローの処理のためにＤＰＤＫ

技術を活用するコアパケット処理アーキテクチャに従って実装されたマルチコアパケットプロセッサの別の実施例を示す。コアパケット処理アーキテクチャは単一の生産者／単一の消費者パラダイムに従って、マルチコアパケットプロセッサとして実装されてもよい。少なくともいくつかの実施形態において、高スループット顧客ＴＣＰフローの処理に加えて、ロードバランサノード１１０上のＤＰＤＫコアアーキテクチャもまたＡＲＰ、ＤＨＣＰ、およびＢＧＰ等の他のプロトコルのための、北および南向きのＮＩＣ１１１４上でのパケットの送受信に用いられてもよい。図２６で示す実施形態においては、作業者コア６２０Ａはこれら他のプロトコルのためのパケットの処理専用である。これらパケットの処理は一般的に顧客ＴＣＰフローよりも低速で行われるため、この作業者コア６２０Ａは「低速の」作業者コアと称されてもよく、一方顧客ＴＣＰフローのみを処理する他の作業者コア６２０Ｂ～６２０Ｄは高速の作業者コアと称されてもよい。北向きのおよび南向きのＮＩＣ１１１４において受信パケットを処理するそれぞれ受信コア６１０Ａおよび６１０Ｂは、低速の作業者コア６２０Ａにより処理される予定のパケットを特定し、パケットを低速の作業者コア６２０Ａのための入力キュー６２２へと導いてもよい。低速の作業者コア６２０Ａはまた、Ｊａｖａ／ＪＮＩにより生成されたパケットのための入力キュー６２２、およびＪａｖａ／ＪＮＩへの出力パケットのための出力キュー６３４を監視してもよい。低速の作業者コア６２０Ａはまた、低速の作業者コア６２０Ａが高速の作業者コア６２０Ｂ～６２０Ｄのそれぞれへとパケット例えば接続公開データパケットを送信できるように、高速の作業者コア６２０Ｂ～６２０Ｄのそれぞれのための入力キュー６２２への出力を行ってもよい。低速の作業者コア６２０Ａはまた伝達コア６３０Ａおよび６３０Ｂのそれぞれに流れ込むアウトバウンドキュー６３２を有する。

【０１７４】

少なくともいくつかの実施形態において、各高速の作業者コア６２０Ｂ～６２０Ｄの第３の入力キュー６２２は低速の作業者コア６２０Ａからの出力キューである。少なくともいくつかの実施形態において、この第３の入力キュー６２２は例えば、それぞれが接続状態情報を含む接続公開パケットの受信および処理のために、高速の作業者キュー６２０Ｂ～６２０Ｄによって用いられてもよい。これら接続公開パケットの少なくともいくつかのためには、伝達コア６３０への出力はなくてもよい。代わりに、データパケットにおける接続状態情報が、例えばそれぞれの高速の作業者コア６２０が維持する１つまたは複数のパケットフローのための格納された状態の更新により、高速の作業者コア６２０から消費されてもよい。こうして、高速の作業者コア６２０Ｂ～６２０Ｄへの入力を行う低速の作業者コア６２０Ａからの出力キューが、入力キュー６２２以外の、受信コア６１０から直接の、高速の作業者コアの格納された状態の更新のためのパスを提供してもよい。

【０１７５】

少なくともいくつかの実施形態において、図２５および２６のマルチコアパケットプロセッサは受信パケットをフィルターにかけ、有効なパケットのみを処理し、出力してもよい。例えば、少なくともいくつかの実施形態において、受信コア６１０は作業者コア６２０のいずれにもサポートされていないプロトコルのパケットをフィルターにかけて取り除いてもよく、その結果作業者コア６２０へとパケットを送信しなくてもよい。少なくともいくつかの実施形態において、作業者コア６２０はパケットの処理時に、それぞれがまずそれぞれの作業者入力キュー６２２から読み取られたパケットを分析し、パケットがさらなる処理のために受け入れられるべきか、また伝達コア６３０への出力を行うべきかを決定してもよく、また単に受け入れられた伝達コア６３０へのパケットの処理および出力を完了してもよい。受け入れられなかったパケットは破棄されてもよい。例えば作業者コア６２０は各パケットのアドレス情報を確認し、ロードバランスされている有効なアドレスを対象とするパケットの受け入れのみを行い、他のパケットは破棄してもよい。

境界ゲートウェイプロトコル（ＢＧＰ）データの処理

【０１７６】

少なくともいくつかの実施形態において、コアアーキテクチャの内部および外部の、ＢＧＰ顧客に関連するパケットフローは以下のように処理されてもよい。ＮＩＣ１１１４Ａ

および１１１４ＢはＬｉｎｕｘカーネルに向かわないので、エッジルータ１０４へのＴＣＰ接続は図２６で示されるようにコアアーキテクチャにより遮断され、出力キュー６３４を通じてＢＧＰパケットをＪａｖａスペースへと渡す低速の作業者コア６２２Ａによって処理される。これらＴＣＰデータパケットはＢＧＰ顧客へと伝達される前にロードバランサノード１１０上の１つまたは複数のモジュールによりさらに処理される。この処理にはＴＣＰ接続を管理し、パケットをＴＣＰストリームへと効率的に変換するためのＬｉｎｕｘカーネルによる処理が含まれる。この設計により、標準ＪａｖａＴＣＰソケットライブラリを用いたＢＧＰ顧客の書き込みが可能になる。

【０１７７】

図２７は少なくともいくつかの実施形態による、ロードバランサ（ＬＢ）ノード処理６５０による受信ＢＧＰ　ＴＣＰデータパケットの処理を示す。エッジルータ１０４からのパケットが北向きのＮＩＣ６４０に到着し、受信コア６５２のための入力キュー６４０に入る。受信コア６５２はキュー６４０からのパケットを読み取り、パケットをＢＧＰパケットとして特定し、パケットを低速の作業者コア６５６のための入力キュー６５４上に配置する。低速の作業者コア６５６はパケットを検証し、それをＪＮＩ出力キュー６５８上に配置する。ＪＮＩパケット受信装置６６０はキュー６５８からのパケットを、ＪＮＩを介して読み取り、ソース／宛先アドレスをマングルし、パケットをｒａｗソケット６４４に書き込む。Ｌｉｎｕｘカーネル６４６は生のパケットを受信し、ＴＣＰプロトコルに従ってそれを処理し、ペイロードデータをＴＣＰソケットＩｎｐｕｔＳｔｒｅａｍへと追加する。パケットからのデータはその後ＢＧＰ顧客６６２内のＪａｖａＴＣＰソケットに伝達される。

【０１７８】

図２８は少なくともいくつかの実施形態による、ロードバランサ（ＬＢ）ノード処理６５０による、発信ＢＧＰ　ＴＣＰデータパケットの処理を示す。ＢＧＰ顧客６６２はデータをＬｉｎｕｘカーネル６４６のＪａｖａＴＣＰソケットへと書き込む。Ｌｉｎｕｘカーネル６４６はＴＣＰプロトコルに従ってデータを処理し、データを１つまたは複数のＴＣＰパケットへと変換する。少なくともいくつかの実施形態において、１つまたは複数のＴＣＰパケットは１２７．ｘ．ｘ．ｘ　ＩＰテーブルルールに適合する。１つまたは複数のＴＣＰパケットは出力キュー６４８、例えばＮｅｔｆｉｌｔｅｒ　ＬＯＣＡＬ_ＯＵＴキュー上に配置される。ＪＮＩを通じてキュー６４８を監視するＪＮＩパケット受信装置６７０のＪａｖａスレッドは１つまたは複数のＴＣＰパケットを受信し、各ＮＦ_ＳＴＯＬＥＮに印を付けてカーネル６４６にそれらを忘れさせる。Ｊａｖａスレッドはソース／宛先アドレスをマングルし、１つまたは複数のパケットを低速の作業者コア６５６のためのＪＮＩ入力キュー６７２へとＪＮＩを通じて追加する。低速の作業者コア６５６はそのＪＮＩ入力キュー６７２から１つまたは複数のＴＣＰパケットを受信してパケットを北向きのＮＩＣ６４０伝達コア６６６のためのアウトバウンドキュー６６４上に配置する。伝達コア６６６はその入力キュー６６４から１つまたは複数のＴＣＰパケットを読み取り、それらを北向きのＮＩＣ６４０に書き込む。ＴＣＰパケットは、ＮＩＣ６４０によってエッジルータに送られる。

分散型ロードバランサのシミュレーションおよびテスティング

【０１７９】

本明細書に記載のロードバランサは、多くの独立した構成要素（例、ルータ、ロードバランサノード、ロードバランサモジュール、等）の対話を要求する分散型システムである。ノードの故障、メッセージの破棄、および遅延等のシナリオのシミュレーションと同様に、分散型構成要素、ロジック、およびプロトコルのテスティングを行うための、分散型ロードバランサを単一処理において動作できるようにするテストシステムの実施形態について記載する。単一処理においては、複雑なネットワークトポロジ（例、本番ネットワーク）において、コードを複数のホストに展開する必要なく対話のテストを行うことができる。これを達成するための、メッセージバスと称される、複数のロードバランサ構成要素を単一処理内でまたは単一処理として構成させて実行させることができるソフトウェア機

構について記載する。単一処理は単一のホストシステムにおいて実行されてもよい。メッセージバス機構により、分散型ロードバランサシステムの単一処理としてのテストが、例えばロードバランサ構成要素（例、ロードバランサノードおよびロードバランサモジュール）にとっては実際の本番ネットワークで動作しているかのような単一のホストシステムにおいて可能になる。

【０１８０】

メッセージバスは、分散型ロードバランサが単一処理として動作することを可能にするフレームワークを提供する。処理における１つまたは複数のメッセージバスレイヤーのそれぞれが分散型ロードバランサの構成要素の間のネットワーク（例、Ｅｔｈｅｒｎｅｔ（登録商標））セグメントのシミュレーションを行う。分散型ロードバランサシステムのソフトウェア構成要素は、構成要素をメッセージバス環境内で動作させるために特別な方法で書かれる必要はない。代わりに、分散型ロードバランサシステムの構成要素が生成するパケットを遮断し、パケットを本物の物理ネットワークの代わりにメッセージバスレイヤーに提供された模擬ネットワーク内に導き、パケットを対象の構成要素へと伝達する構成要素（メッセージバスＮＩＣまたはパケットアダプタと称されてもよい）をメッセージバスフレームワークが提供する。メッセージバスレイヤーは、構成要素間の通信のための１つまたは複数のＴＣＰ／ＩＰスタックを実装しない。その代わりに、メッセージバスレイヤーはホストシステムのオペレーティングシステム（ＯＳ）とのインターフェースを行い、ホストシステムのＴＣＰ／ＩＰスタックを用いる。メッセージバスレイヤーはＯＳにより提供されたＴＣＰ／ＩＰスタックを、顧客およびサーバが予期するＴＣＰストリームの、メッセージバスが遮断し伝達する個別のパケットからの、そしてそういった個別のパケットへの変換のために活用する。

【０１８１】

少なくともいくつかの実施形態において、メッセージバスとのインターフェースを行うために、ロードバランサ構成要素は少なくとも１つのメッセージバスネットワークインタフェースコントローラ（ＮＩＣ）を与えられていてもよい。各ＮＩＣは有効なメディアアクセス制御（ＭＡＣ）アドレスを有しており、それにより物理ネットワークとのやり取りの代わりに、メッセージバス模擬ネットワーク環境へとパケットを送信し、またそこからパケットを受信する。メッセージバスＮＩＣは物理ネットワークの代わりにメッセージバスに付随する仮想ネットワークインタフェースコントローラである。メッセージバスを通じて通信する必要のある各ロードバランサ構成要素は少なくとも１つのメッセージバスＮＩＣを要求する。メッセージバスＮＩＣはメッセージバスへのパイプライン出口の役割と、構成要素へのパイプライン入口の役割を担う。構成要素は複数のメッセージバスネットワークインターフェースを各メッセージバスＮＩＣへとインスタンス化することができる。

【０１８２】

メッセージバスネットワークインターフェースはメッセージバスＮＩＣを通じてメッセージバスに付随する構成要素のための機構である。メッセージバスネットワークインターフェースはＬｉｎｕｘ技術におけるインターフェース構成（ｉｆｃｏｎｆｉｇ）インターフェースと同義であってもよく、違いとしては、メッセージバスネットワークインターフェースが物理ネットワークの代わりにメッセージバスに付随する点である。メッセージバスネットワークインターフェースはＩＰアドレスを有し、メッセージバスＮＩＣの最上段にある。メッセージバスネットワークインターフェースは、構成要素によってメッセージバスからのパケットの受信のために用いられることができるパケットソースインターフェースと、構成要素によってメッセージバスへのパケットの発信のために用いられることができるパケットシンクインターフェースを公開する。

【０１８３】

各ロードバランサノードは、パケットソースおよびパケットシンクインターフェースの実装を通じて伝達され送信される個別のネットワークパケットを処理する。メッセージバス環境における動作時には、これらのインターフェースは、レイヤー２Ｅｔｈｅｒｎｅｔ

ヘッダを追加または削除するメッセージバスネットワークインターフェースによって実装される。（ロードバランサノードのためには、これはカーネルネットワークスタックによって実行されることになる）。図２９に示す本番環境においては、パケットソースおよびパケットシンクインターフェースの実装は、実際のネットワークインターフェースにおいてパケットを受信し伝達する。図３０で示されるメッセージバス環境において、パケットソースおよびパケットシンクインターフェースの実装がメッセージバスレイヤーまたはレイヤーからパケットを受信し、メッセージバスレイヤーまたはレイヤーへとパケットを伝達する。

【０１８４】

単純化のために、メッセージバスＮＩＣおよびメッセージバスインターフェースはメッセージバスパケットアダプタ、または単にパケットアダプタと総称されてもよい。例、図３１および３２を参照。

【０１８５】

図２９は少なくともいくつかの実施形態による、本番環境において分散型ロードバランサ７００を含むロードバランスシステムを示す。ロードバランサ７００はこの説明では単純化されている。ロードバランサ７００は外部ネットワーク７４０上の顧客７４２へと、ロードバランサ７００を実装するデータセンター等のネットワークインストールの境界ルータ７０２を通じて接続してもよい。ロードバランサ７００はいくつかの種類の構成要素を含む－少なくとも１つのエッジルータ７０４、２つ以上のロードバランサ（ＬＢ）ノード７１０、それぞれが個別のサーバノード（図示せず）上で実装された２つ以上のロードバランサ（ＬＢ）モジュール７３２、ファブリック７２０を形成するルータやスイッチのような１つまたは複数のネットワーク構成要素、また、少なくともいくつかの実施形態において構成サービス７２２。少なくともいくつかの実施形態において、ロードバランサ７００の各構成要素は、ラック搭載型のコモディティコンピューティング装置等の個別のコンピューティング装置上で実装されてもよい。

【０１８６】

図３０は少なくともいくつかの実施形態による、複数の分散型ロードバランスシステムの構成要素を単一処理内でまたは単一処理として構成させて実行させることができるメッセージバス機構を組み込む分散型ロードバランサテストシステム８００を示す。図２９に示されるロードバランサ７００において、各ロードバランサソフトウェア構成要素は、個別のコンピューティング装置上でインストールされ、実行される（例、ロードバランサノード７１０上にロードバランサソフトウェア、また、サーバノード上にロードバランサモジュール７３２）。これらロードバランサソフトウェア構成要素を単一処理において実行させるため、各ロードバランサソフトウェア構成要素（図３０においてロードバランサ（ＬＢ）ノード８１０およびロードバランサ（ＬＢ）モジュール８３２として示される）は、ロードバランサソフトウェア構成要素を出入りするパケットがまた、物理ネットワーク上で送受信される代わりに、メッセージバス機構を通じて遮断され、ルーティングされるように、構成要素のネットワーク接続性を抽出するコードを含んでもよい。

【０１８７】

少なくともいくつかの実施形態において、分散型ロードバランサテストシステム８００上では、メッセージバス機構は１つまたは複数の構成要素間の通信のためのＴＣＰスタックを実装しない。その代わりに、メッセージバス機構はホストシステムのオペレーティングシステム（ＯＳ）とのインターフェースを行い、ホストシステムのＴＣＰスタックを用いる。少なくともいくつかの実施形態において、メッセージバス機能は、ホストシステムのＯＳのカーネル（例、Ｌｉｎｕｘカーネル）と、カーネルの機能であるＩＰテーブルを通じて、ユーザーレイヤーの下で結びついている。メッセージバス機能はカーネルレベルでＩＰテーブルに接続され、パケットを遮断し、ルーティングのためのメッセージバス処理へとパケットを送信する。

【０１８８】

図３０において模擬エッジルータ８６２および模擬ファブリック８６４で示されるよう

に、物理ネットワーク構成要素（例、図２９のエッジルータ７０４およびファブリック７２０）の機能は、顧客８６０、サーバ８３４、構成サービス８６６も可能であるように、ソフトウェアにおいてシミュレーションされてもよい。しかし、少なくともいくつかの実施形態において、模擬サーバ８３４ではなく実物が分散型ロードバランサテストシステム８００において用いられてもよいことに留意する。図３０のメッセージバスレイヤー８５０が物理ネットワークインフラストラクチャの代わりとなる。したがって、ロードバランサソフトウェア構成要素（ロードバランサノード８１０およびロードバランサモジュール８３２）は、図２９で示すように本番ネットワーク環境で実行していないことを認識しないままロードバランサテストシステム８００上で動作してもよい。

【０１８９】

いくつかの構成要素（例えば模擬ルータ）は、ネットワークセグメントをシミュレーションする異なるメッセージバスレイヤー８５０とパケットの送受信を行うために、２つ以上のメッセージバスレイヤー８５０に接続されてもよい。

【０１９０】

分散型ロードバランステストシステム８００のメッセージバスレイヤー８５０において実装されるメッセージバス機構は、ネットワークセグメントの「ワイヤ」をシミュレーションする。少なくともいくつかの実施形態において、メッセージバス機構は構成要素のＭＡＣアドレスに基づいて、パケットを分散型ロードバランステストシステム８００内の宛先構成要素に伝達する。こうして、各ロードバランサソフトウェア構成要素（ロードバランサノード８１０およびロードバランサモジュール８３２）は、ロードバランサソフトウェア構成要素が、分散型ロードバランステストシステム８００において他の構成要素から送信されたパケットを受信できるように、ＭＡＣアドレスを１つまたは複数の接続されているメッセージバスレイヤー８５０に提供する。

メッセージバスパケットアダプタ

【０１９１】

図３１および３２は少なくともいくつかの実施形態による、メッセージバスパケットアダプタを示す。少なくともいくつかの実施形態において、各ロードバランサ（ＬＢ）ソフトウェア構成要素はパケットソースおよびパケットシンクインターフェースの実装を通じて伝達され送信される個々のネットワークデータパケットを処理する。図３１を参照すると、これらインターフェース（パケットソースインターフェース８６２およびパケットシンクインターフェース８６４として示される）は分散型ロードバランステストシステム８００上で動作している際に、メッセージバスレイヤー８５０と、カーネルネットワークスタックによって実行されることになるソフトウェア構成要素８７０のためのレイヤー２Ｅｔｈｅｒｎｅｔヘッダを追加または削除するロードバランサソフトウェア構成要素８７０との間のパケットアダプタ８６０によって実装されてもよい。図２９で示されるような本番環境では、ロードバランサソフトウェア構成要素のためのパケットソースおよびパケットシンクの実装が、構成要素が実装される物理装置の実際のネットワークインターフェース上で、パケットを受信し伝達する。

【０１９２】

図３１を参照すると、少なくともいくつかの実施形態において、ロードバランサソフトウェア構成要素８７０がパケットを伝達する際に、パケットを構成要素の入力キューに追加することで最終的にパケットを宛先構成要素に伝達するために、パケットシンクインターフェース８６４の送信パケット方法を呼び出す実行スレッドがパケットアダプタ８６０内とメッセージバスレイヤー８５０内の関数チェーンをトラバースする。少なくともいくつかの実施形態において、ロードバランサソフトウェア構成要素８７０がパケットを受信する際に、ロードバランサソフトウェア構成要素８７０がパケットソースインターフェース８６２の受信パケット方法を呼び出し、その入力キューからパケットを読み取る。少なくともいくつかの実施形態において、メッセージバス機構はパケットの伝達のためにいかなる追加のスレッドも要求しない。

メッセージバスパケットパイプライン

【０１９３】
図３２を参照すると、少なくともいくつかの実施形態において、パケットソースインターフェース８６２およびパケットシンクインターフェース８６４のメッセージバス８５０側がパケットパイプラインの特徴を提供する。ロードバランサソフトウェア構成要素８７０がパケットシンクインターフェース８６４を通じてパケットを送信する際、パケットデータはメッセージバスレイヤー８５０に達する前に、段階のシリーズ（パケットパイプライン８８０）をトラバースしてもよい。これらの段階はパケットを修正し、パケットを破棄し、パケットを複製し、パケットを遅延させる等してもよい。パケットがパケットパイプライン８８０をトラバースしメッセージバスレイヤー８５０が宛先構成要素８７０を選択すると、パケットが宛先構成要素８７０の入力キューに追加される前に、宛先構成要素８７０に関連するパイプライン段階の第２のシリーズ（パケットパイプライン８８２）もまたトラバースされてもよい。

プロバイダネットワーク環境の実施例

【０１９４】
このセクションでは、分散型ロードバランス方法および機器の実施形態が実装されてもよいプロバイダネットワーク環境の実施例を説明する。しかし、これらプロバイダネットワーク環境の実施例は制限を意図するものではない。

【０１９５】
図３３Ａは少なくともいくつかの実施形態による、プロバイダネットワーク環境の実施例を示す。プロバイダネットワーク１９００は、顧客に仮想化リソースのインスタンス１９１２のアクセス、購入、レンタル、またはその他の取得を可能にする、１つまたは複数の仮想化サービス１９１０を通じて顧客にリソース仮想化を提供してもよい。仮想化リソースのインスタンス１９１２には、１つまたは複数のデータセンター内のプロバイダネットワークまたはネットワーク内の装置に実装された計算およびストレージリソースが含まれるが、それらに限定されない。プライベートＩＰアドレス１９１６はリソースのインスタンス１９１２に関連してもよい。プライベートＩＰアドレスは、プロバイダネットワーク１９００上のリソースのインスタンス１９１２の内部ネットワークアドレスである。いくつかの実施形態においては、プロバイダネットワーク１９００はまた、顧客がプロバイダ１９００から取得してもよい、パブリックＩＰアドレス１９１４および／またはパブリックＩＰアドレスの範囲（例、インターネット　Ｐｒｏｔｏｃｏｌバージョン４（ＩＰｖ４）またはインターネット　Ｐｒｏｔｏｃｏｌバージョン６（ＩＰｖ６）アドレス）を提供してもよい。

【０１９６】
従来から、仮想化サービス１９１０を通じたプロバイダネットワーク１９００により、サービスプロバイダの顧客（例、顧客ネットワーク１９５０Ａを運用する顧客）が顧客に割り当てられた少なくともいくつかのパブリックＩＰアドレス１９１４を顧客に割り当てられた特定のリソースのインスタンス１９１２と動的に結びつけることができる。また、プロバイダネットワーク１９００により顧客は、顧客に割り当て済みのある仮想化されたコンピューティングリソースのインスタンス１９１２にすでにマッピングされていたパブリックＩＰアドレス１９１４を、同様に顧客に割り当てられた別の仮想化されたコンピューティングリソースのインスタンス１９１２へと再度マッピングすることができる。サービスプロバイダから提供された、仮想化されたコンピューティングリソースのインスタンス１９１２およびパブリックＩＰアドレス１９１４を用いて、顧客ネットワーク１９５０Ａの運用者などのサービスプロバイダの顧客は例えば、インターネットのような中間ネットワーク１９４０上で、顧客特有のアプリケーションを実装し、顧客のアプリケーションを提示してもよい。中間ネットワーク１９４０上の他のネットワークエンティティ１９２０はその後顧客ネットワーク１９５０Ａによって公開された宛先パブリックＩＰアドレス１９１４へのトラフィックを生成してもよい。トラフィックはサービスプロバイダデータセンターへとルーティングされ、データセンターで、ネットワーク基盤を通じて、現在宛先パブリックＩＰアドレス１９１４にマッピングされている仮想化されたコンピューティ

ングリソースのインスタンス１９１２のプライベートＩＰアドレス１９１６へとルーティングされる。同様に、仮想化されたコンピューティングリソースのインスタンス１９１２からの応答トラフィックはネットワーク基盤を通じて中間ネットワーク１９４０に戻り、ソースエンティティ１９２０へとルーティングされてもよい。

【０１９７】

プライベートＩＰアドレスは、本明細書で用いられている通り、プロバイダネットワーク内のリソースのインスタンスの内部ネットワークアドレスを参照する。プライベートＩＰアドレスはプロバイダネットワーク内でのみルーティング可能である。外部のプロバイダネットワークから始まるネットワークトラフィックはプライベートＩＰアドレスに直接ルーティングされない。代わりにトラフィックは、リソースのインスタンスへとマッピングされたパブリックＩＰアドレスを用いる。プロバイダネットワークは、パブリックＩＰアドレスからプライベートＩＰアドレスへのマッピングやその逆を行うための、ネットワークアドレス変換（ＮＡＴ）または類似の機能を提供するネットワーク装置または機器を含んでもよい。

【０１９８】

パブリックＩＰアドレスは、本明細書で用いられている通り、サービスプロバイダまたは顧客のいずれかによってリソースのインスタンスに割り当てられた、インターネット上でルーティング可能なネットワークアドレスである。パブリックＩＰアドレスへとルーティングされたトラフィックは例えば１：１ネットワークアドレス変換（ＮＡＴ）を通じて変換され、リソースのインスタンスのそれぞれのプライベートＩＰアドレスへと転送される。

【０１９９】

いくつかのパブリックＩＰアドレスは、プロバイダネットワークインフラストラクチャから特定のリソースのインスタンスへと割り当てられてもよい。これらのパブリックＩＰアドレスは標準パブリックＩＰアドレス、または単に標準ＩＰアドレスと称されてもよい。少なくともいくつかの実施形態において、標準ＩＰアドレスのリソースのインスタンスのプライベートＩＰアドレスへのマッピングは、すべてのリソースのインスタンスタイプ向けのデフォルトの起動構成である。

【０２００】

少なくともいくつかのパブリックＩＰアドレスは、プロバイダネットワーク１９００の顧客に割り振られるか、またはそのような顧客により取得されてもよい。顧客はその後割り振られたパブリックＩＰアドレスを、顧客に割り振られた特定のリソースのインスタンスに割り当ててもよい。これらのパブリックＩＰアドレスは、顧客パブリックＩＰアドレス、または単に顧客のＩＰアドレスと称されてもよい。標準ＩＰアドレスと同様にプロバイダネットワーク１９００によりリソースのインスタンスに割り当てられる代わりに、顧客のＩＰアドレスは、例えばサービスプロバイダから提供されたＡＰＩを通じて、顧客によりリソースのインスタンスに割り当てられてもよい。標準ＩＰアドレスとは異なり、顧客のＩＰアドレスは顧客アカウントに割り当てられるものであり、必要に応じてまたは希望があれば、他のリソースのインスタンスへのそれぞれの顧客による再度マッピングすることが可能である。顧客のＩＰアドレスは顧客のアカウントに関連し、特定のリソースのインスタンスには関連せず、顧客がそのＩＰアドレスのリリースを選択するまでは、自身でそれを管理する。従来の固定ＩＰアドレスとは異なり、顧客のＩＰアドレスにより、顧客のパブリックＩＰアドレスを顧客のアカウントに関連するいずれかのリソースのインスタンスへと再度マッピングすることで、顧客はリソースのインスタンスまたはアベイラビリティーゾーンの故障をマスクすることができる。顧客のＩＰアドレスにより、例えば、顧客のリソースのインスタンスまたはソフトウェアの問題について顧客のＩＰアドレスを置き換えのリソースのインスタンスへと再度マッピングすることで、解決に向け動くことができる。

【０２０１】

図３３Ｂは少なくともいくつかの実施形態による、図３３Ａに示されるような、プロバ

イダネットワーク環境の実施例における分散型ロードバランサの実装を示す。プロバイダネットワーク１９００は、例えば仮想ストレージサービスのようなサービス１９１０を顧客１９６０に提供してもよい。顧客１９６０は、例えばサービス１９１０への１つまたは複数のＡＰＩを通じて、サービス１９１０にアクセスし、プロバイダネットワーク１９００の本番ネットワーク部分における複数のサーバノード１９９０上に実装されたリソース（例、ストレージリソースまたは計算リソース）の利用形態を得てもよい。サーバノード１９９０はそれぞれ、ローカルロードバランサ（ＬＢ）モジュール１９９２と同様に、サーバ（図示せず）、例えばウェブサーバまたはアプリケーションサーバを実装してもよい。１つまたは複数の分散型ロードバランサ１９８０は、境界ネットワークと本番ネットワークとの間のロードバランサレイヤーにおいて実装されてもよい。境界ルータ１９７０は、顧客１９６０からのパケットフロー上のパケット（例、ＴＣＰデータパケット）をインターネット等の中間ネットワーク１９４０を通じて受信し、パケットを１つまたは複数の分散型ロードバランサ１９８０の１つまたは複数のエッジルータへと境界ネットワークを通じて転送してもよい。データパケットは、１つまたは複数の分散型ロードバランサ１９８０の１つまたは複数のエッジルータによって公開されたパブリックＩＰアドレスを対象としてもよい。各分散型ロードバランサ１９８０のエッジルータは、それぞれの分散型ロードバランサ１９８０のロードバランサノード間にパケットフローを分散させてもよい。少なくともいくつかの実施形態において、入口ノードの役割を担う各ロードバランサノードはエッジルータに同一のパブリックＩＰアドレスを提供し、エッジルータはフローごとにハッシュ化されたマルチパスルーティング技術、例えば等価コストマルチパス（ＥＣＭＰ）ハッシュ技術に従って、顧客１９６０からのパケットフローを入口サーバ間に分散させる。パケットフローのための対象のサーバノード１９９０を決定するため、また、サーバと顧客１９６０との間の接続を円滑化するために、ロードバランサノードは本明細書に記載の接続プロトコルを用いてもよい。接続が確立されると、フロートラッカーノードが接続のための状態を維持する一方で、入口ノードがフローのために受信されたパケットをカプセル化して本番ネットワーク上の対象のサーバノード１９９０へと発信する。サーバノード１９９０上のロードバランサモジュール１９９２は、サーバノード１９６０上のそれぞれのサーバが接続を受け入れるかどうかについて決断を下してもよい。ロードバランサモジュールは入口ノードからのパケットを受信して脱カプセル化し、脱カプセル化されたデータパケット（例、ＴＣＰデータパケット）をサーバノード１９９０上のそれぞれのサーバへと送信する。ロードバランサモジュール１９９２はまた、パケットフローのための出口ノードとしてロードバランサノードを選択し、フローのための発信パケットをカプセル化し、選択した出口ノードに本番ネットワークを通じて送信してもよい。次に出口ノードがパケットを脱カプセル化し、脱カプセル化されたパケットを、それぞれの顧客１９６０への伝達のための境界ネットワークへと送信する。

【０２０２】

図３４Ａは少なくともいくつかの実施形態による、分散型ロードバランサおよびサーバノードの物理ラックの実装の実施例を示すが、制限を意図するものではない。少なくともいくつかの実施形態において、分散型ロードバランサ上の様々な構成要素が、ラック搭載型のコモディティコンピューティング装置上で、またはラック搭載型のコモディティコンピューティング装置として実装されてもよい。ラック１９０は、それぞれがロードバランサノードの役割を担う複数のコンピューティング装置（ＬＢノード１１０Ａ～１１０Ｆ）および、それぞれがサーバノードの役割を担う複数のコンピューティング装置（サーバノード１３０Ａ～１３０Ｌ）を含んでもよい。ラック１９０はまた、少なくとも１つのエッジルータ１０４、ファブリック１２０を形成する１つまたは複数のラック搭載型ネットワーク装置（ルータ、スイッチ等）、および１つまたは複数の他の構成要素１８０（他のネットワーク装置、パッチパネル、電源、冷却システム、バス等）を含む。図３３Ａおよび３３Ｂのプロバイダネットワーク１９００を実装するデータセンターまたはセンター等のネットワーク１００のインストールは、１つまたは複数のラック１９０を含んでもよい。

【０２０３】

図３４Ｂは少なくともいくつかの実施形態による、分散型ロードバランサおよびサーバノードの物理ラックの実装の別の実施例を示すが、制限を意図するものではない。図３４Ｂはスロット搭載型コンピューティング装置、例えばブレードサーバとして実装された、ラック１９０上のＬＢノード１１０およびサーバノード１３０を示す。

【０２０４】

図３５は少なくともいくつかの実施形態による、ネットワーク上で１つまたは２つ以上の分散型ロードバランサが実装されてもよく、別途実装されたサーバノードを持つネットワーク環境の実施例を示す。この実施例では、２つの分散型ロードバランサ１９８０Ａおよび１９８０Ｂが示される。分散型ロードバランサ１９８０はそれぞれ境界ネットワークを通じて顧客１９６０からのパケットフローを受信し、複数のサーバノード１９９０間にパケットフローを分散させるために本明細書に記載のロードバランス方法を実行してもよい。いくつかの実装においては、各分散型ロードバランサ１９８０は図３４Ａおよび３４Ｂで示されるラック１９０に類似のラック実装であってもよいが、ロードバランサラック内にインストールされたサーバノードは含まれない。サーバノード１９９０は、データセンター内の１つまたは複数の個別のラック上にインストールされたブレードサーバ等のラック搭載型コンピューティング装置であってもよい。いくつかの実装においては、サーバノード１９９０は、プロバイダネットワークにより提供され、それぞれが異なる１つまたは複数の分散型ロードバランサ１９８０によりフロントに配置された、２つ以上の異なるサービスを実装してもよい。

例示的システム

【０２０５】

少なくともいくつかの実施形態において、本明細書に記載の分散型ロードバランス方法および機器の一部またはすべてを実装するサーバは、図３６に示されるコンピュータシステム２０００のような、１つまたは複数のコンピュータアクセス可能な媒体を含む、またはそのような媒体にアクセスするよう構成されている汎用コンピュータシステムを含んでもよい。示された実施形態においては、コンピュータシステム２０００は、入力／出力（Ｉ／Ｏ）インターフェース２０３０を通じてシステムメモリ２０２０に接続された１つまたは複数のプロセッサ２０１０を含む。コンピュータシステム２０００は、Ｉ／Ｏインターフェース２０３０に接続されたネットワークインターフェース２０４０をさらに含む。

【０２０６】

様々な実施形態において、コンピュータシステム２０００は、１つのプロセッサ２０１０を含むユニプロセッサシステム、または、いくつかのプロセッサ２０１０（例、２つ、４つ、８つ、または別の適切な数）を含むマルチプロセッサシステムであってもよい。プロセッサ２０１０は、命令を実行する能力があるいずれかの適切なプロセッサであってもよい。例えば様々な実施形態において、プロセッサ２０１０は、ｘ８６、ＰｏｗｅｒＰＣ、ＳＰＡＲＣ、またはＭＩＰＳＩＳＡ等の各種の命令セットアーキテクチャ（ＩＳＡ）、または他のいずれかの適切なＩＳＡを実装する汎用または埋め込みプロセッサであってもよい。マルチプロセッサシステムにおいては、プロセッサ２０１０のそれぞれが一般に同一のＩＳＡを実装してもよいが、必ずしもそうである必要はない。

【０２０７】

システムメモリ２０２０は、１つまたは複数のプロセッサ２０１０によりアクセス可能な命令およびデータを格納するよう構成されてもよい。様々な実施形態において、システムメモリ２０２０は、スタティックランダムアクセスメモリ（ＳＲＡＭ）、シンクロナスダイナミックＲＡＭ（ＳＤＲＡＭ）、不揮発性／フラッシュ型メモリ、またはその他各種メモリ等のいずれかの適切なメモリ技術を用いて実装されてもよい。示された実施形態においては、ロードバランス方法および機器のために上述された方法、技術、およびデータ等の、１つまたは複数の所望の関数を実装するプログラム命令およびデータが、システムメモリ２０２０内にコード２０２４およびデータ２０２６として格納されていることが示される。

【０２０８】

１つの実施形態において、Ｉ／Ｏインターフェース２０３０は、プロセッサ２０１０、システムメモリ２０２０、およびネットワークインターフェース２０４０または他の周辺インターフェースを含む装置内のいずれかの周辺装置の間のＩ／Ｏトラフィックを調整するよう構成されてもよい。いくつかの実施形態においては、１つの構成要素（例、システムメモリ２０２０）からのデータ信号を、別の構成要素（例、プロセッサ２０１０）による利用に適したフォーマットへと変換するために、Ｉ／Ｏインターフェース２０３０はいかなる必要なプロトコル、タイミングまたは他のデータ媒体変換を行ってもよい。いくつかの実施形態においては、Ｉ／Ｏインターフェース２０３０は例えば、周辺構成要素相互接続（ＰＣＩ）バス標準またはユニバーサルシリアルバス（ＵＳＢ）標準のバリアント等の様々な種類の周辺バスに付随する装置のためのサポートを含んでもよい。いくつかの実施形態においては、Ｉ／Ｏインターフェース２０３０の関数は、例えばノースブリッジおよびサウスブリッジ等の、２つ以上の個別の構成要素へと分割されてもよい。また、いくつかの実施形態においては、システムメモリ２０２０へのインターフェース等の、Ｉ／Ｏインターフェース２０３０の一部またはすべての機能がプロセッサ２０１０に直接組み込まれてもよい。

【０２０９】

ネットワークインターフェース２０４０は、コンピュータシステム２０００と、例えば図１～３５で示すような他のコンピュータシステムまたは装置等の、１つまたは複数のネットワーク２０５０に付随する他の装置２０６０との間でのデータ交換が可能であるように構成されてもよい。様々な実施形態において、ネットワークインターフェース２０４０は、例えばＥｔｈｅｒｎｅｔネットワークのような種類の、いずれかの適切な有線または無線の一般的なデータネットワークを通じた通信をサポートしてもよい。また、ネットワークインターフェース２０４０は、アナログ音声ネットワークまたはデジタルファイバ通信ネットワークのような遠隔通信／電話網を通した通信、Ｆｉｂｒｅ　Ｃｈａｎｎｅｌ　ＳＡＮ等のストレージエリアネットワークを通じた通信、またはその他いずれかの適切な種類のネットワークおよび／またはプロトコルを通じた通信をサポートしてもよい。

【０２１０】

いくつかの実施形態においてシステムメモリ２０２０は、分散型ロードバランスシステムの実施形態を実装するための図１～３５について上述したように、プログラム命令およびデータを格納するよう構成されたコンピュータアクセス可能な媒体の１つの実施形態であってもよい。しかし他の実施形態においては、異なる種類のコンピュータアクセス可能な媒体において、プログラム命令および／またはデータが受信され、送信され、または格納されてもよい。一般的に、コンピュータアクセス可能な媒体は、磁気または光学式媒体等の非一時的記憶媒体またはメモリ媒体を含んでもよい。例、Ｉ／Ｏインターフェース２０３０を通じてコンピュータシステム２０００に接続されるディスクまたはＤＶＤ／ＣＤ。コンピュータアクセス可能な非一時的記憶媒体はまた、コンピュータシステム２０００のいくつかの実施形態においてシステムメモリ２０２０または別の種類のメモリとして含まれてもよいＲＡＭ（例えば、ＳＤＲＡＭ、ＤＤＲＳＤＲＡＭ、ＲＤＲＡＭ、ＳＲＡＭ等）、ＲＯＭ等のあらゆる揮発性または不揮発性媒体を含んでもよい。さらにコンピュータアクセス可能な媒体は、ネットワークインターフェース２０４０を通じて実装されてもよいネットワークおよび／または無線接続等の通信媒体を通じて伝達される電気、電磁、またはデジタル信号等の伝送媒体または信号を含んでもよい。

【０２１１】

本開示の実施形態は、以下の節を考慮して説明することができる。

１．複数のロードバランサノードの少なくとも２つが入口サーバとして構成され、

前記複数のロードバランサノードの少なくとも２つがフロートラッカーノードとして構成される、

前記複数のロードバランサノードと、

複数のサーバノードと、

１つまたは複数の顧客からのパケットフローを、ハッシュ化されたマルチパスルーティ

ング技術に従って、前記入口サーバへと分散させるよう構成されたルータと、
を備えた分散型ロードバランサシステムであり、
顧客のためのパケットフローにおけるパケットを前記ルータから受信し、
前記複数のサーバノードへの前記パケットフローのためのマッピングを前記入口サーバが有しないことを決定し、
前記パケットのソースおよび宛先アドレス情報に適用される一貫したハッシュ関数に従って、前記パケットフローのための少なくとも１つのフロートラッカーノードを決定し、
前記パケットフローのための前記複数のサーバノードの特定の１つへの接続のマッピングを、少なくとも１つのフロートラッカーノードから取得し、
前記特定のサーバノードへの前記パケットフローの１つ以上のパケットを送信する
ように各入口サーバが構成される、
分散型ロードバランサシステム。
２．前記パケットフローがトランスミッションコントロールプロトコル（ＴＣＰ）パケットフローである、第１節に記載の分散型ロードバランサシステム。
３．前記複数のロードバランサノードの少なくとも２つが、前記１つまたは複数の顧客への前記サーバノードからの発信パケットを送信するよう構成された出口サーバとして構成され、
前記パケットフローのための前記出口サーバを選択し、
前記パケットフローのための１つまたは複数の発信パケットを、前記選択した出口サーバへと送信する、
ように前記サーバノードが構成され、
前記顧客への前記発信パケットを送信するように、前記出口サーバが構成され、
前記パケットフローのための前記選択された出口サーバが、前記パケットフローのための前記入口サーバとは異なるロードバランサノードである、
第１節に記載の分散型ロードバランサシステム。
４．前記サーバノードへの前記パケットの送信の前に、前記入口サーバが前記１つまたは複数のパケットをユーザデータグラムプロトコル（ＵＤＰ）に従ってカプセル化し、前記出口サーバへの前記発信パケットの送信の前に、前記サーバノードが前記発信パケットをＵＤＰに従ってカプセル化し、前記顧客への前記発信パケットの送信の前に、前記出口サーバが前記発信パケットから前記ＵＤＰカプセル封じを取り外す、第３節に記載の分散型ロードバランサシステム。
５．前記パケットフローのための前記出口サーバを選択し、
前記カプセル化された受信パケットを前記入口サーバから受信し、
前記パケットから前記ＵＤＰカプセル封じを取り外し、前記パケットを前記サーバノード上のサーバへと伝達させ、
前記サーバノード上の前記サーバから前記発信パケットを取得し、
ＵＤＰに従って、前記発信パケットをカプセル化し、
前記カプセル化された発信パケットを前記出口サーバへと送信する
ように構成されるロードバランサモジュールを前記サーバノードが含む、
第４節に記載の分散型ロードバランサシステム。
６．前記パケットフローのための前記複数のサーバノードの特定の１つへの接続のマッピングを前記少なくとも１つのフロートラッカーノードから取得するために、
前記パケットフローのための情報を含むメッセージを、前記入口サーバが前記パケットフローのための１次フロートラッカーへと送信し、
前記パケットフローのための前記情報を含むメッセージを、前記１次フロートラッカーが前記パケットフローのための２次フロートラッカーへと送信し、前記パケットフローのための前記１次および２次フロートラッカーが異なるロードバランサノードであり、
前記２次フロートラッカーが、前記パケットフローのための受信確認を、前記顧客へと送信し、
前記入口サーバが、受信確認パケットを前記顧客から受信し、前記受信確認パケットを

前記１次フロートラッカーへと転送し、
前記１次フロートラッカーが、前記パケットフローを受信するための前記サーバノードとして、前記複数のサーバノードの中から前記特定のサーバノードを無作為に選択し、前記特定のサーバノードを示すメッセージを、前記２次フロートラッカーへと送信し、
前記２次フロートラッカーが、同期メッセージを生成して前記生成された同期メッセージを前記特定のサーバノードへと送信し、
前記２次フロートラッカーが、前記パケットフローのための接続情報を前記特定のサーバノードから受信して前記１次フロートラッカーへの前記接続情報を含むメッセージを送信し、
前記１次フロートラッカーが、前記パケットフローのための前記接続情報を含むメッセージを前記入口サーバへと送信し、前記接続情報が前記パケットフローを前記特定のサーバノードへとマッピングする、
第１節に記載の分散型ロードバランサシステム。
７．前記生成された同期メッセージを前記２次フロートラッカーから受信し、
前記サーバノード上のサーバが接続を受け入れることができることを決定し、
前記生成された同期メッセージに従って同期パケットを生成し、前記同期パケットを前記サーバノード上の前記サーバへと伝達し、
前記サーバノード上の前記サーバによって生成された受信確認パケットを遮断し、
前記接続情報を含むメッセージを前記２次フロートラッカーへと送信する、
ように構成されるロードバランサモジュールを前記サーバノードが含む、
第６節に記載の分散型ロードバランサシステム。
８．顧客からのパケットフローにおけるパケットの受信であって、１つまたは複数の顧客から一貫したハッシュ関数に従って前記複数のロードバランサノードへと前記パケットフローを分散させるルータからのパケットの受信、
前記パケットのソースおよび宛先アドレス情報に適用される一貫したハッシュ関数に従って、前記パケットフローのためのフロートラッカーノードとしての役割を担うロードバランサノードの決定、
前記パケットフローのための複数のサーバノードの１つへの接続のマッピングの、前記パケットフローのための前記フロートラッカーノードからの取得、
前記マッピングにより示された、前記サーバノードへの前記パケットフローの１つまたは複数のパケットの送信、
を、複数のロードバランサノードのひとつの入口サーバによって実行すること、
を備えた方法。
９．前記パケットフローがトランスミッションコントロールプロトコル（ＴＣＰ）パケットフローである、第８節に記載の方法。
１０．前記パケットフローの前記１つまたは複数のパケットの前記サーバノードへの前記送信の前に、ユーザデータグラムプロトコル（ＵＤＰ）に従って前記パケットをカプセル化する前記入口サーバをさらに備えた、第８節に記載の方法。
１１．前記サーバノードによる、前記パケットフローのための出口サーバとしての前記複数のロードバランサノードの１つの選択であり、前記パケットフローのための前記選択した出口サーバが、前記パケットフローのための前記入口サーバとは異なるロードバランサノードである、前記複数のロードバランサノードの１つの選択と、
前記サーバノードによる、前記パケットフローのための１つまたは複数の発信パケットの、前記選択された出口サーバへの送信と、
前記出口サーバによる前記発信パケットの、前記パケットフローの前記顧客への送信と、
をさらに備えた、第８節に記載の方法。
１２．前記発信パケットの前記出口サーバへの前記送信の前に、ユーザデータグラムプロトコル（ＵＤＰ）に従って前記発信パケットをカプセル化する前記サーバノードと、
前記発信パケットの顧客への前記送信の前に、前記発信パケットから前記ＵＤＰカプセ

ル封じを取り外す前記出口サーバと、
をさらに備えた、第１１節に記載の方法。
１３．前記フロートラッカーノードが前記パケットフローのための１次フロートラッカーノードであり、前記一貫したハッシュ関数に従った一貫したハッシュリングにおける次のロードバランサノードが前記パケットフローのための２次フロートラッカーノードである、第８節に記載の方法。
１４．前記入口サーバによる、少なくとも１つのメッセージの前記パケットフローのための前記１次フロートラッカーノードへの送信であり、各メッセージが前記ルータから受信された前記パケットフローのパケットを含む前記送信と、
前記１次フロートラッカーノードによる、前記複数のサーバノードからの前記パケットフローのための前記サーバノードの選択と、
前記１次フロートラッカーノードによる、前記選択されたサーバノードを示すパケットフロー情報の前記２次フロートラッカーノードへの送信と、
前記２次フロートラッカーノードによる、前記サーバノードと前記顧客との通信による、前記パケットフローのための前記選択されたサーバノードへの前記接続の確立の円滑化と、
前記２次フロートラッカーノードによる、前記パケットフローのための接続情報の、前記１次フロートラッカーノードを通じた前記入口サーバへの送信であり、前記接続情報が前記選択されたサーバノードへの前記パケットフローのマッピングを行う前記送信と、
を、前記パケットフローのための複数のサーバノードの１つへの接続のマッピングの、前記パケットフローのための前記フロートラッカーノードからの前記取得が備える、
第１３節に記載の方法。
１５．前記２次フロートラッカーノードによる、前記サーバノード上の前記ロードバランサモジュールへの、生成された同期メッセージの送信と、
前記サーバノード上のサーバが接続を受け入れることができることの決定、
前記生成された同期メッセージに従った同期パケットの生成、
前記同期パケットの、前記サーバノード上の前記サーバへの伝達、
前記サーバノード上の前記サーバにより生成された受信確認パケットの遮断と、
前記接続情報を含むメッセージの、前記２次フロートラッカーノードへの送信
を、前記サーバノード上の前記ロードバランサモジュールにより実行すること、
を、前記サーバノードと前記顧客との通信により、前記パケットフローのための前記選択されたサーバノードへの前記接続の確立を円滑化するロードバランサモジュールを前記サーバノードが含む、
第１４節に記載の方法。
１６．１つまたは複数の顧客からのパケットフローを一貫したハッシュ関数に従って前記複数のロードバランサノードへと分散させるルータからパケットが受信されるように、顧客のためのパケットフロー内の前記パケットを受信し、
前記パケットのソースおよび宛先アドレス情報に適用される一貫したハッシュ関数に従って、前記パケットフローのためのフロートラッカーノードとしての役割を担う、複数のロードバランサノードの１つを決定し、
前記パケットフローのための複数のサーバノードの１つへの接続のマッピングを、前記パケットフローのための前記フロートラッカーノードから取得し、
前記パケットフローの１つまたは複数のパケットを、前記マッピングにより示された前記サーバノードへと送信する、
ように各入口サーバが構成された、
複数のロードバランサノードのおのおのに入口サーバおよびフロートラッカーを実装するためにコンピュータにより実行可能なプログラム命令を格納するコンピュータアクセス可能な非一時的記憶媒体。
１７．前記パケットフローのための出口サーバとして前記複数のロードバランサノードの１つを選択し、前記パケットフローのための選択した出口サーバが前記パケットフロー

のための入口サーバとは異なるロードバランサノードであり、
前記パケットフローのための１つまたは複数の発信パケットを前記選択された出口サーバへと送信する、
ロードバランサモジュールから受信されたパケットフロー上の発信パケットを、前記パケットフローの顧客へと送信するように各出口サーバが構成される、
ように各ロードバランサモジュールが構成され、
それぞれのロードバランサノード上に出口サーバ、複数の前記サーバノードのそれぞれの上にロードバランサモジュールを実装するために前記プログラム命令がさらにコンピュータにより実行可能である、第１６節に記載のコンピュータアクセス可能な非一時的記憶媒体。

１８．各入口サーバがさらに、前記パケットフローの１つまたは複数のデータパケットの前記サーバノードへの前記送信の前に、ユーザデータグラムプロトコル（ＵＤＰ）に従って前記パケットをカプセル化するよう構成され、
入口サーバから受信した前記パケットから前記ＵＤＰカプセル封じを取り外し、前記パケットを前記それぞれのサーバノード上のサーバへと伝達し、
前記それぞれのサーバノード上の前記サーバからの前記発信パケットを遮断し、
出口サーバへの前記発信パケットの前記送信の前に、ＵＤＰに従って前記発信パケットをカプセル化する、
ように各ロードバランサモジュールがさらに構成され、
前記パケットフローの顧客への前記発信パケットの送信の前に、前記発信パケットから前記ＵＤＰカプセル封じを取り外すように、前記各出口サーバがさらに構成される、
第１７節に記載のコンピュータアクセス可能な非一時的記憶媒体。

１９．フロートラッカーノードが前記パケットフローのための１次フロートラッカーノードであり、一貫したハッシュ関数に従った一貫したハッシュリング内の次のロードバランサノードが、前記パケットフローのための２次フロートラッカーノードであり、前記パケットフローのため前記フロートラッカーノードから、前記パケットフローのための複数の１つへの接続のマッピングを取得するために、
前記入口サーバが少なくとも１つのメッセージを前記パケットフローのための前記１次フロートラッカーノードへと送信するよう構成され、各メッセージが前記ルータから受信された前記パケットフローのパケットを含み、
前記パケットフローのための前記サーバノードを前記複数のサーバノードから選択し、
前記選択されたサーバノードを示すパケットフロー情報を、前記２次フロートラッカーノードへと送信する、
ように、前記１次フロートラッカーノードが構成され、
前記サーバノードと前記顧客との間の通信により、前記パケットフローのための前記選択したサーバノードへの前記接続の確立を円滑化し、
前記パケットフローのための接続情報を前記入口サーバへと前記１次フロートラッカーノードを通じて送信し、前記接続情報が前記選択されたサーバノードへの前記パケットフローのマッピングを行う、
ように、前記２次フロートラッカーノードが構成される、
第１６節に記載のコンピュータアクセス可能な非一時的記憶媒体。

２０．前記プログラム命令がさらにコンピュータにより実行可能であり、複数のサーバノードのそれぞれの上にロードバランサモジュールを実装するため、また、前記サーバと前記顧客との間の通信により、前記パケットフローのための前記選択されたサーバノードへの前記接続の確立を円滑化するために、
前記２次フロートラッカーノードが生成された同期メッセージを前記サーバノード上の前記ロードバランサモジュールへと送信するよう構成され、
前記サーバノード上の前記サーバが接続を受け入れることができることを決定し、
前記生成された同期メッセージに従って同期パケットを生成し、
同期パケットを前記サーバノード上の前記サーバへと伝達し、

前記サーバノード上の前記サーバにより生成された受信確認パケットを遮断し、
前記接続情報を含むメッセージを前記２次フロートラッカーノードに送信する、
ように、前記サーバノード上の前記ロードバランサモジュールが構成される、
第１９節に記載のコンピュータアクセス可能な非一時的記憶媒体。
結論

【０２１２】

様々な実施形態は、上記のコンピュータアクセス可能な媒体に関する説明に従って実装された命令および／またはデータの受信、送信または格納をさらに含んでもよい。一般的に、コンピュータアクセス可能な媒体は、ネットワークおよび／または無線接続により伝達される通信媒体を介した電気、電磁、またはデジタル信号等の伝送媒体または信号と同様に、磁気または光学式媒体等の記憶媒体またはメモリ媒体、例、ディスクまたはＤＶＤ／ＣＤ-ＲＯＭ、ＲＡＭ（例えば、ＳＤＲＡＭ、ＤＤＲ、ＲＤＲＡＭ、ＳＲＡＭ等）、ＲＯＭ等の揮発性または不揮発性媒体を含んでもよい。

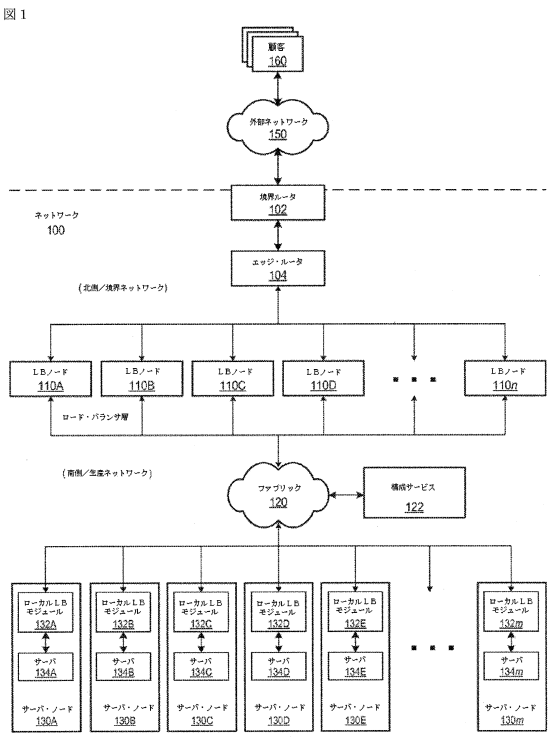
【０２１３】

図面に示され、本明細書に記載された様々な方法は、方法の例示的な実施形態を表す。方法はソフトウェア、ハードウェア、またはそれらの組み合わせにおいて実装されてもよい。方法の順序は変更されてもよく、また、様々な要素が追加され、再整理され、組み合わされ、省略され、修正される等してもよい。

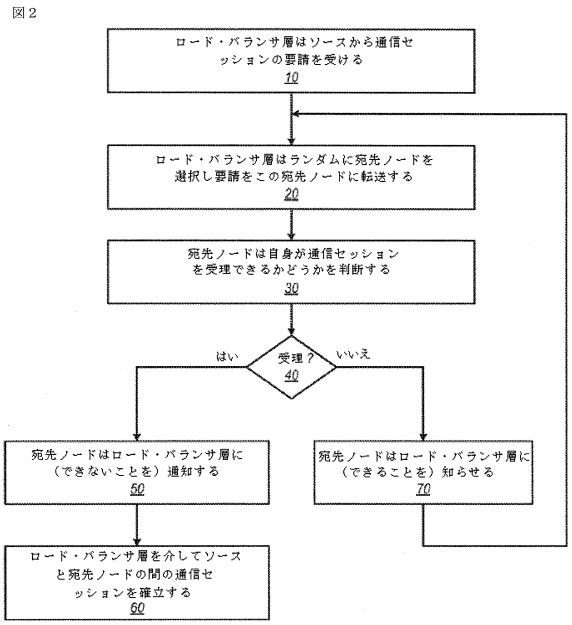
【０２１４】

当業者には明らかであるように、本開示を利用して様々な修正や変更が加えられてもよい。本開示はそのようなすべての修正や変更を包含することを意図しており、したがって、上記記載は制限的な意味ではなく説明的な意味を持つと見なされるべきである。

【図１】

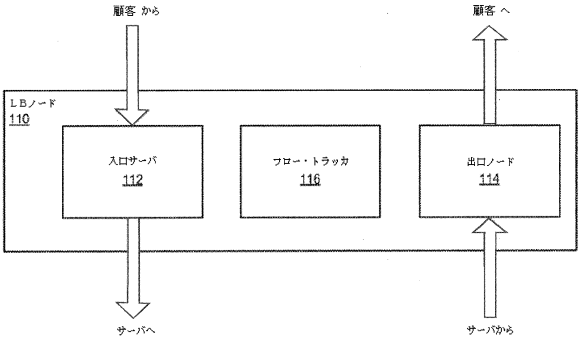


【図２】



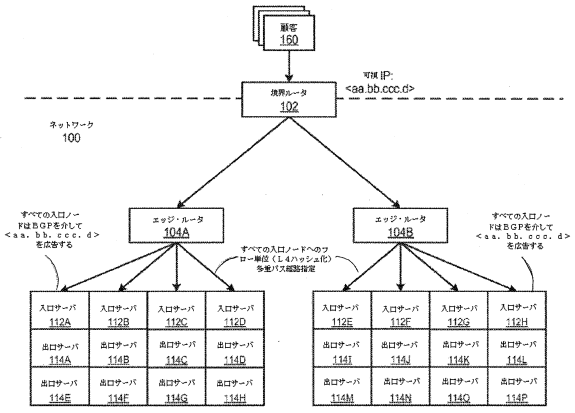
【図３】

図３



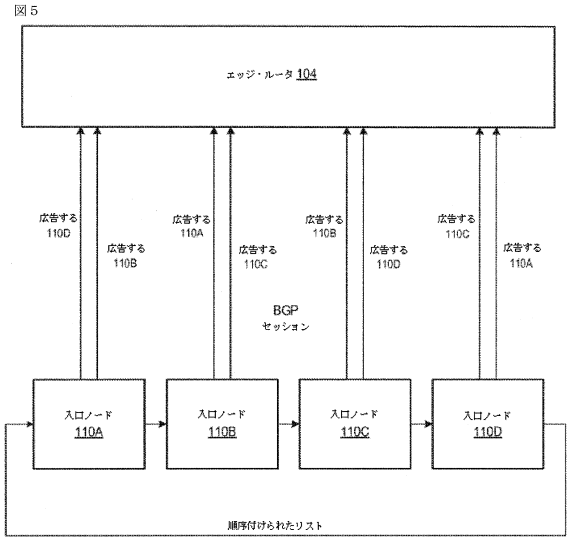
【図４】

図４



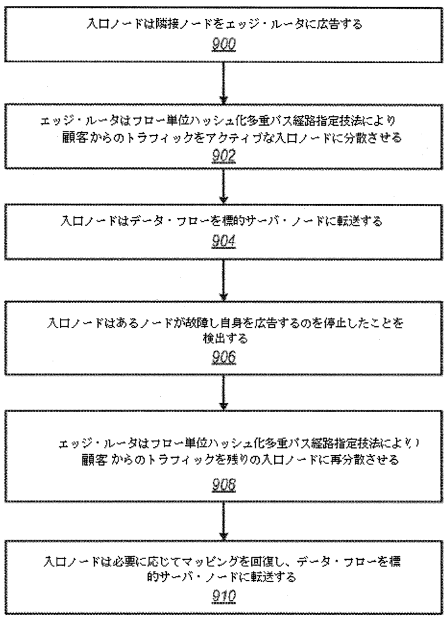
【図５】

図５



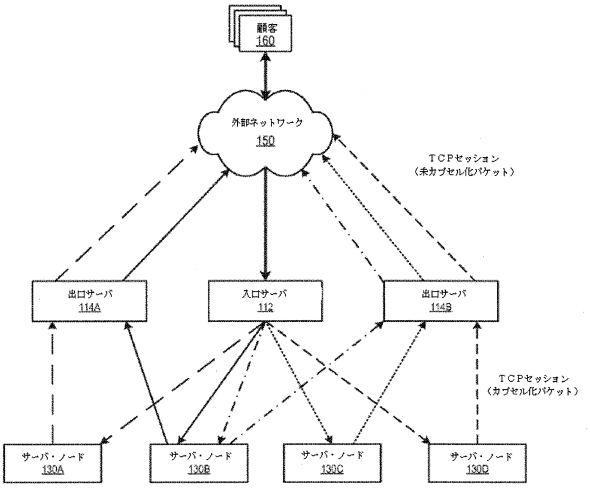
【図６】

図６



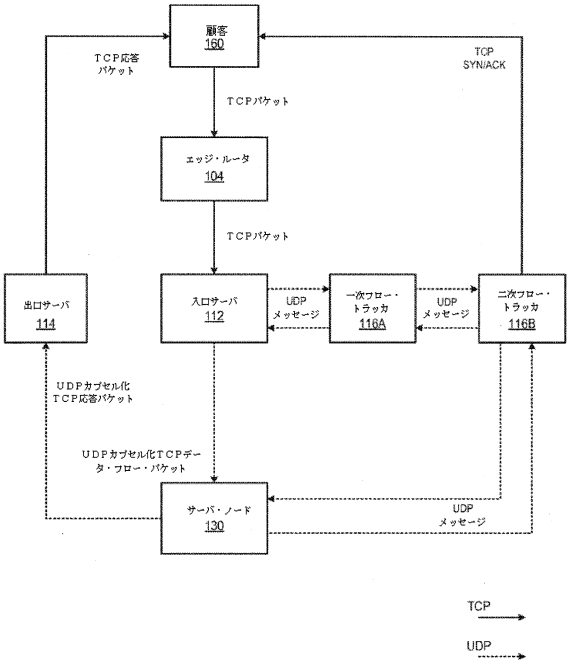
【図７】

図７



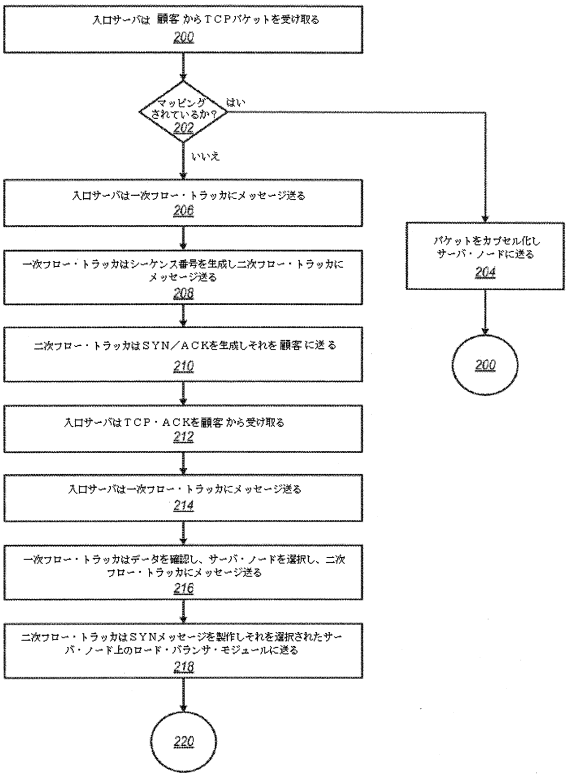
【図８】

図８



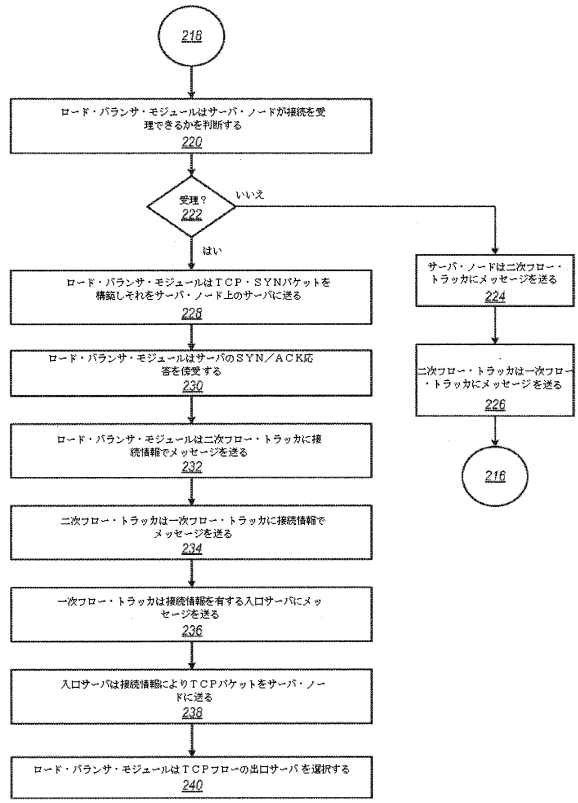
【図９Ａ】

図９Ａ



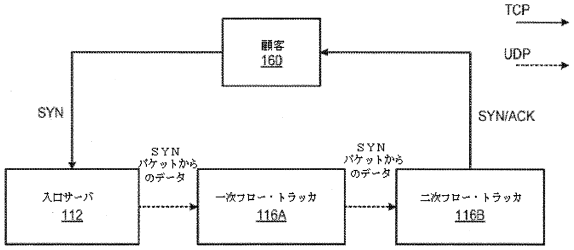
【図９Ｂ】

図９Ｂ



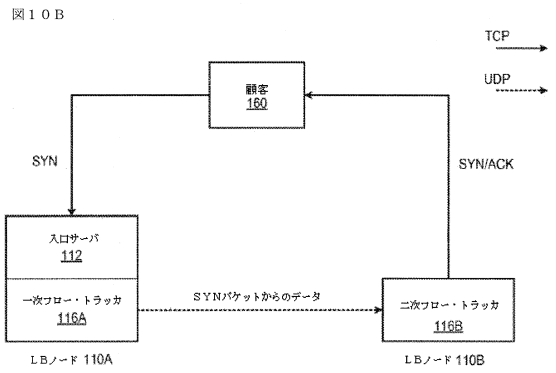
【図１０Ａ】

図１０Ａ

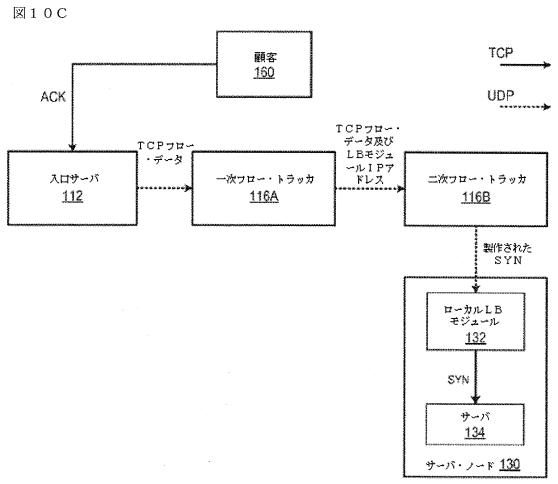


【図１０Ｂ】

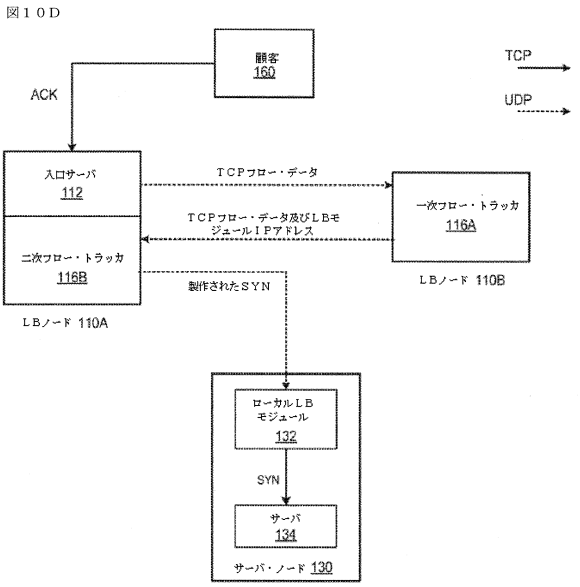
図１０Ｂ



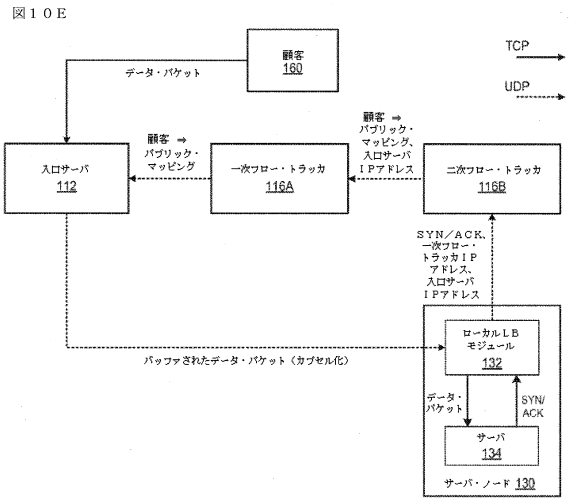
【図１０Ｃ】



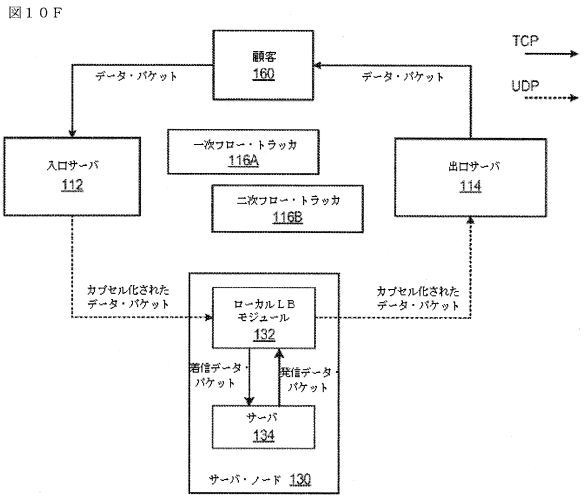
【図１０Ｄ】



【図１０Ｅ】

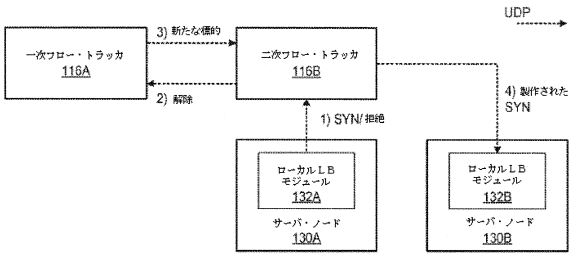


【図１０Ｆ】



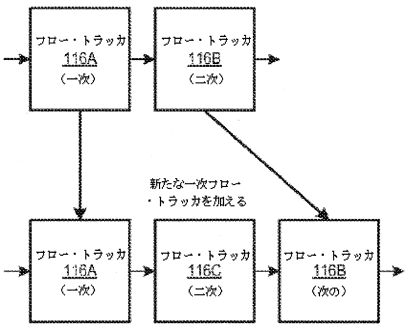
【図１０Ｇ】

図１０Ｇ



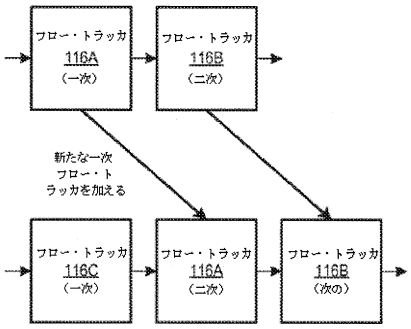
【図１１Ｂ】

図１１Ｂ



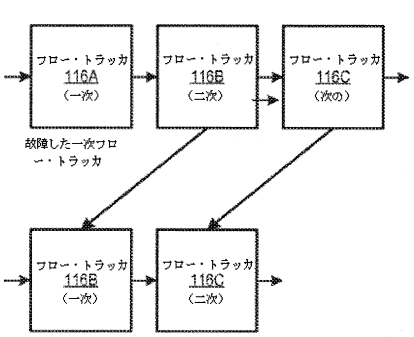
【図１１Ａ】

図１１Ａ



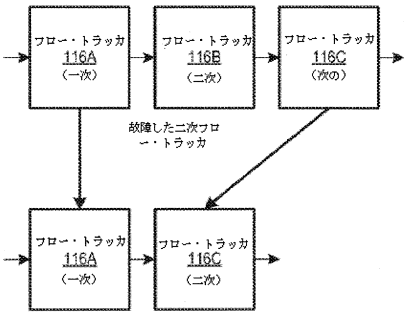
【図１１Ｃ】

図１１Ｃ



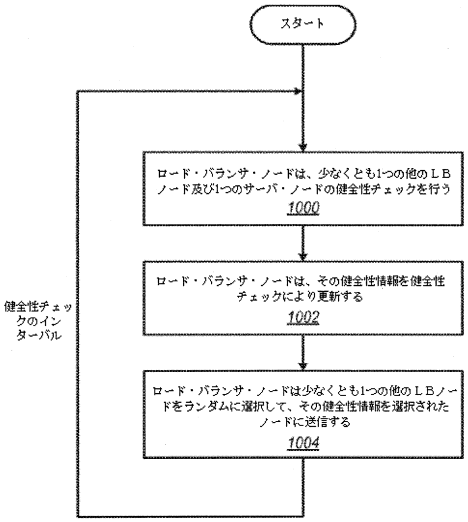
【図１１Ｄ】

図１１Ｄ



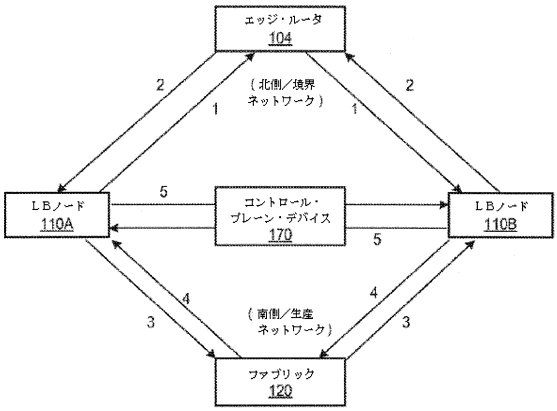
【図１２】

図１２



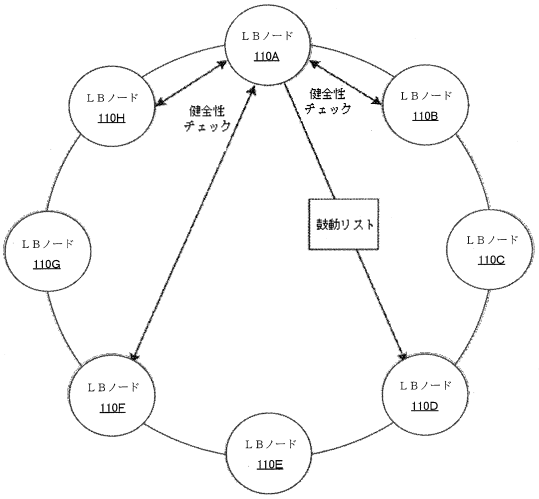
【図１３】

図１３



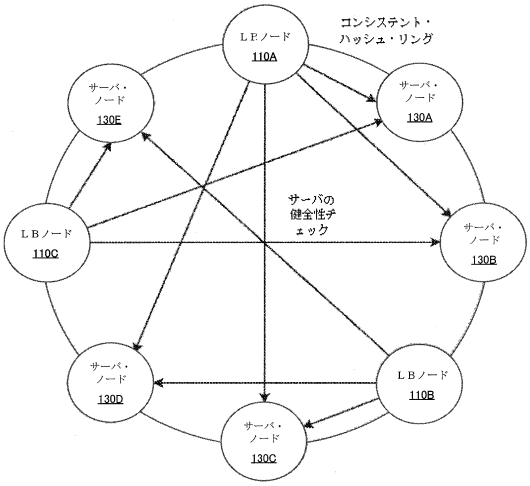
【図１４】

図１４



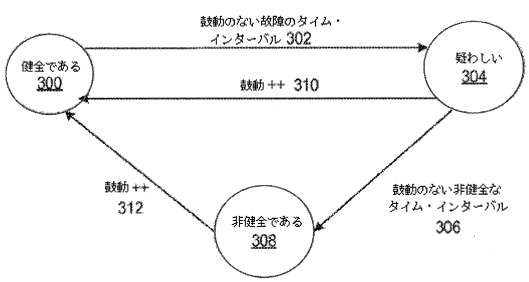
【図１５】

図１５



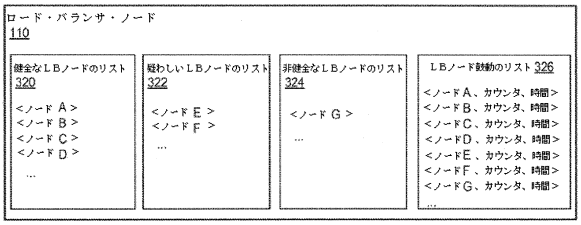
【図１６】

図１６



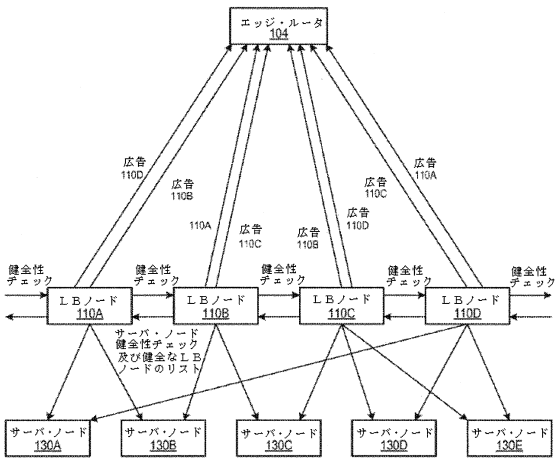
【図１７】

図１７



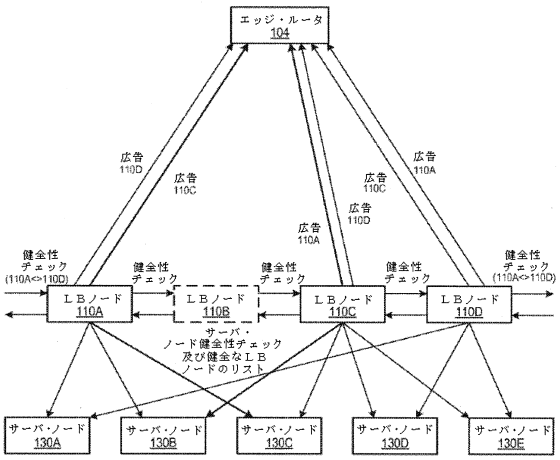
【図１８Ａ】

図１８Ａ



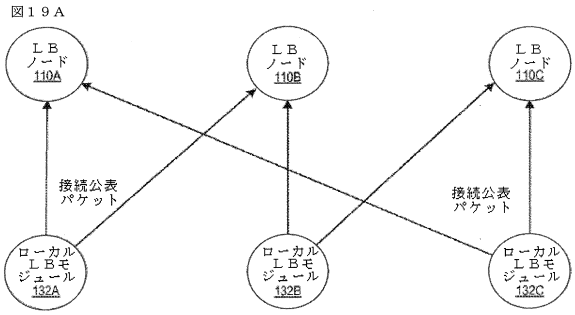
【図１８Ｂ】

図１８Ｂ



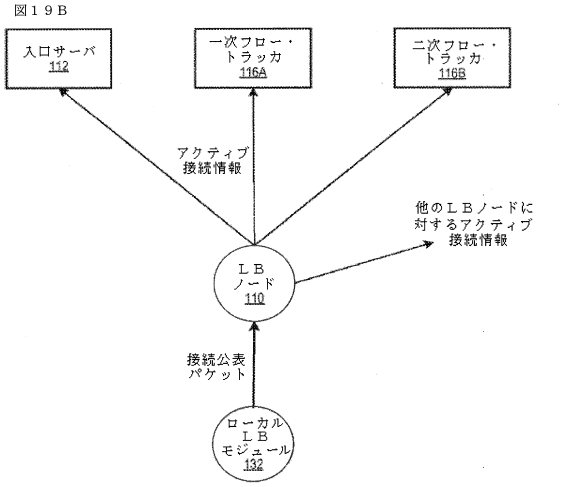
【図１９Ａ】

図１９Ａ



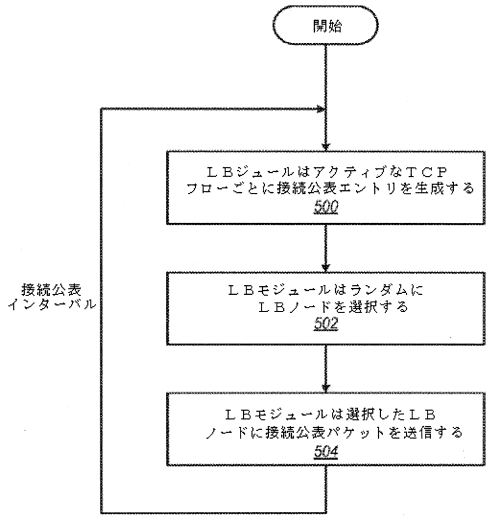
【図１９Ｂ】

図１９Ｂ



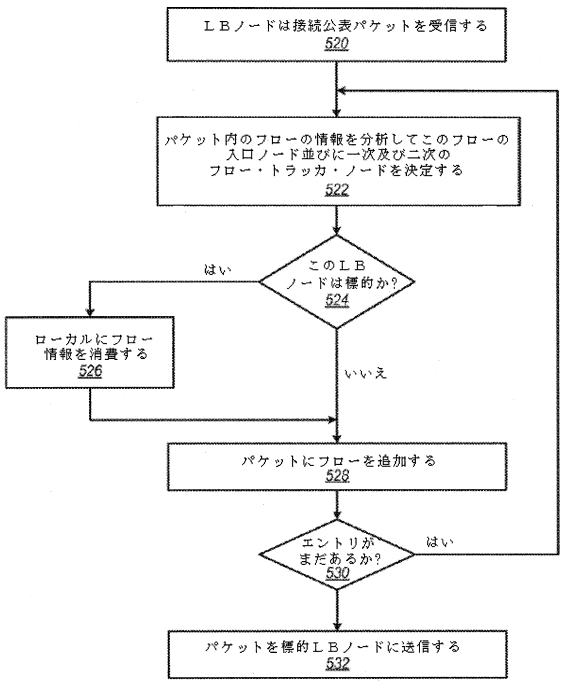
【図２０】

図２０



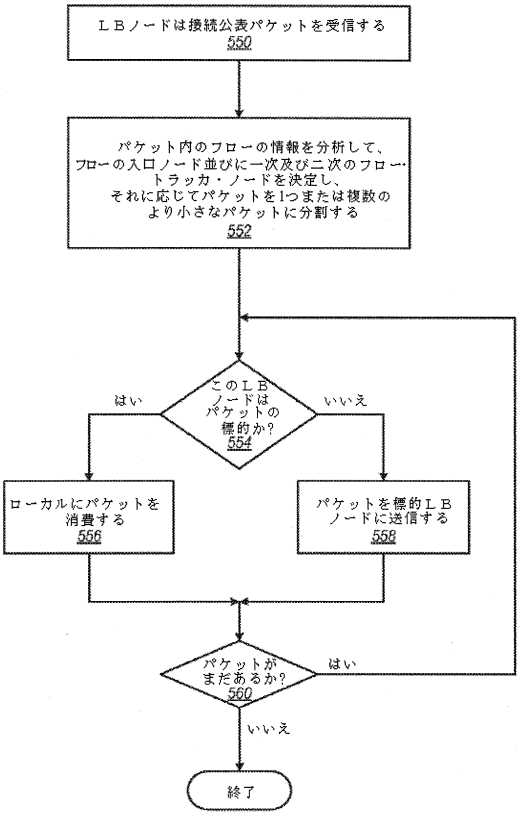
【図２１】

図２１



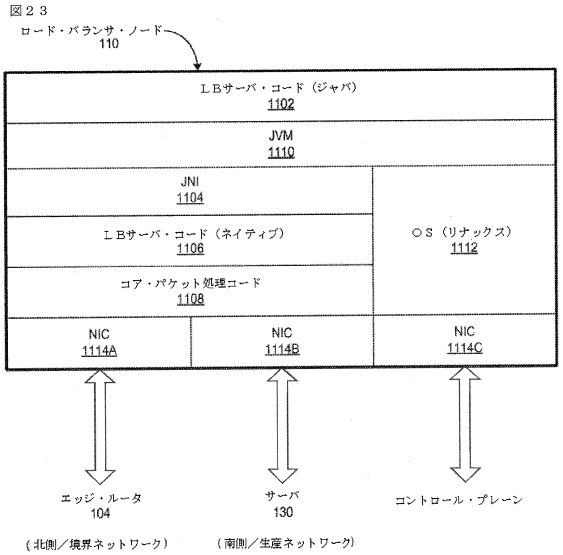
【図２２】

図２２



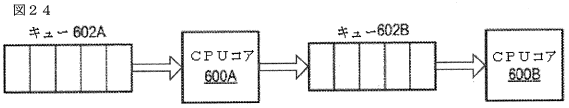
【図２３】

図２３



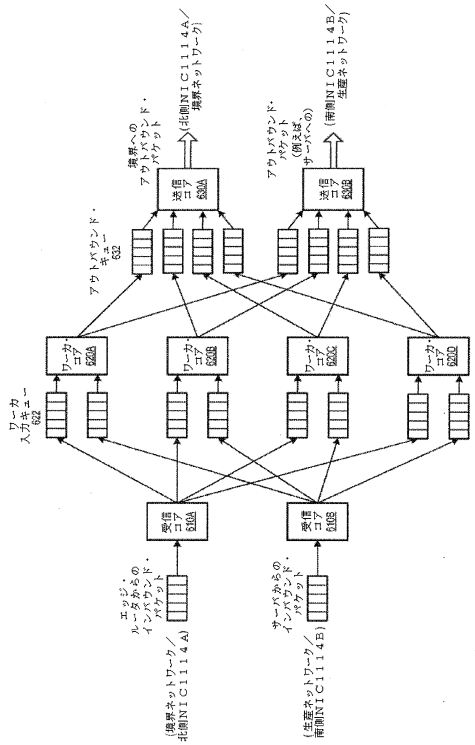
【図２４】

図２４



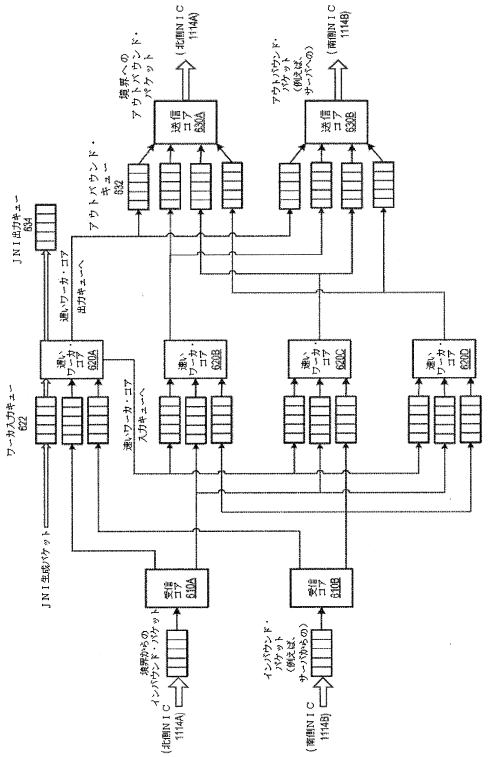
【図２５】

図２５



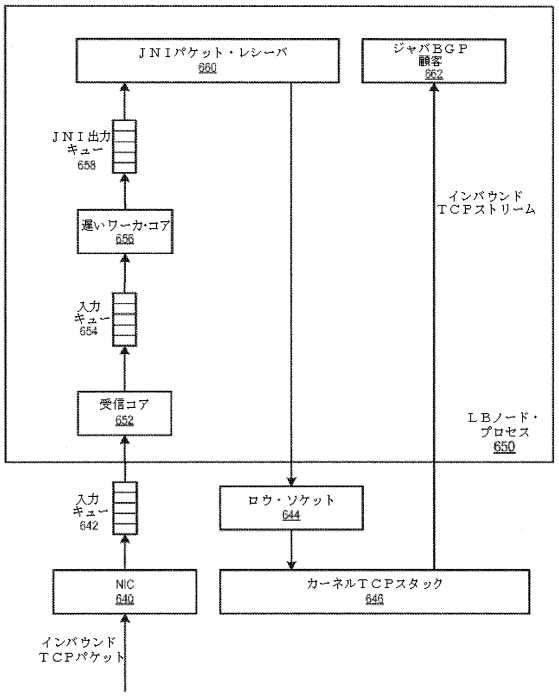
【図２６】

図２６



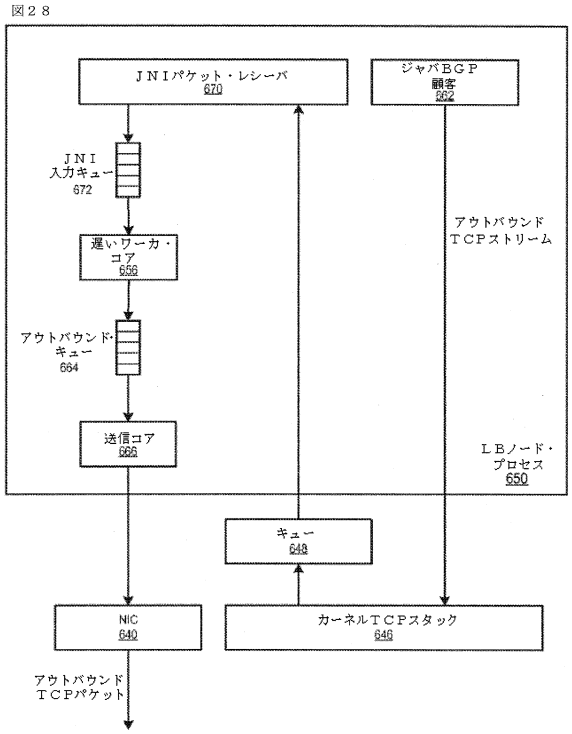
【図２７】

図２７



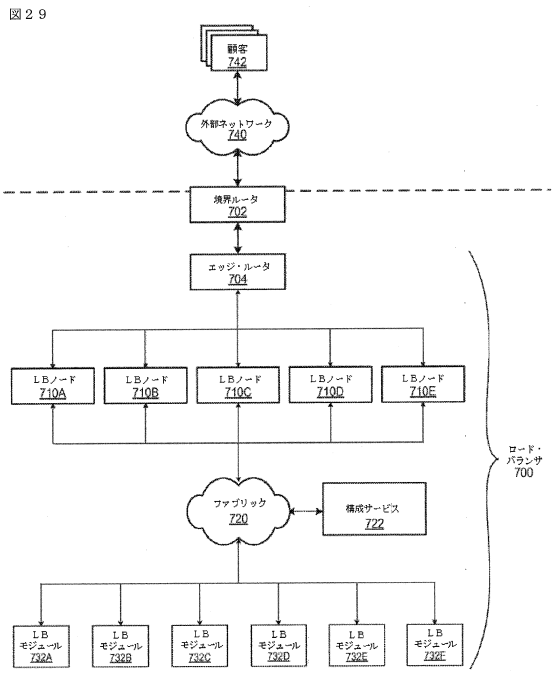
【図２８】

図２８



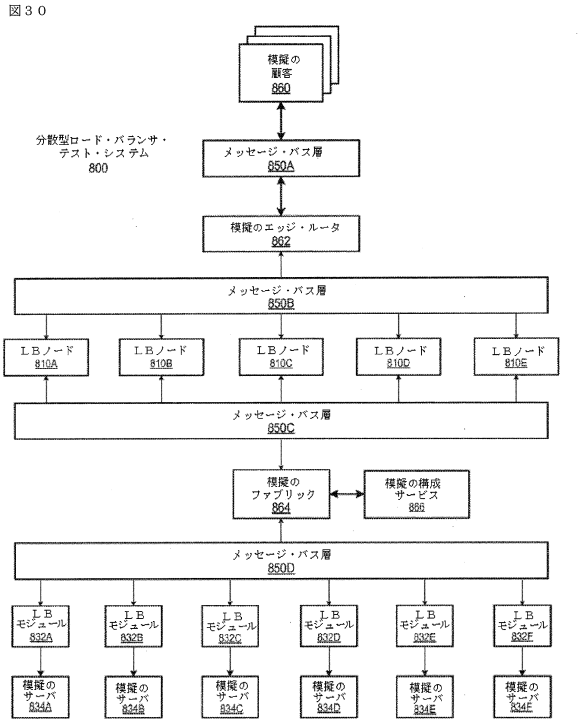
【図２９】

図２９



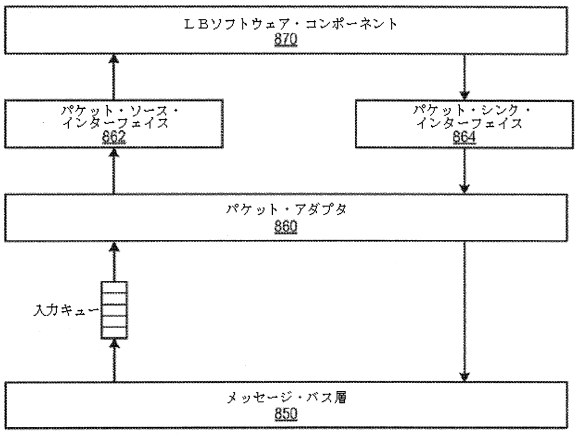
【図３０】

図３０



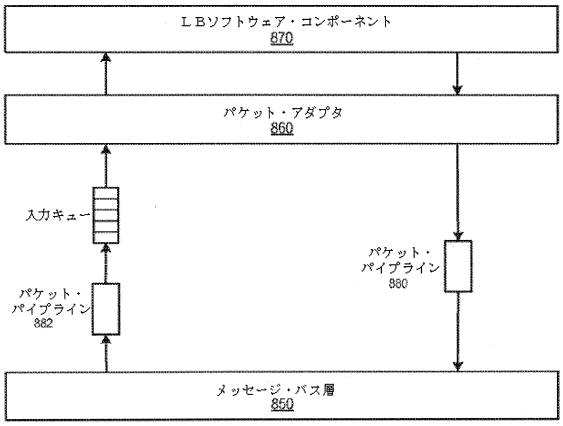
【図３１】

図３１



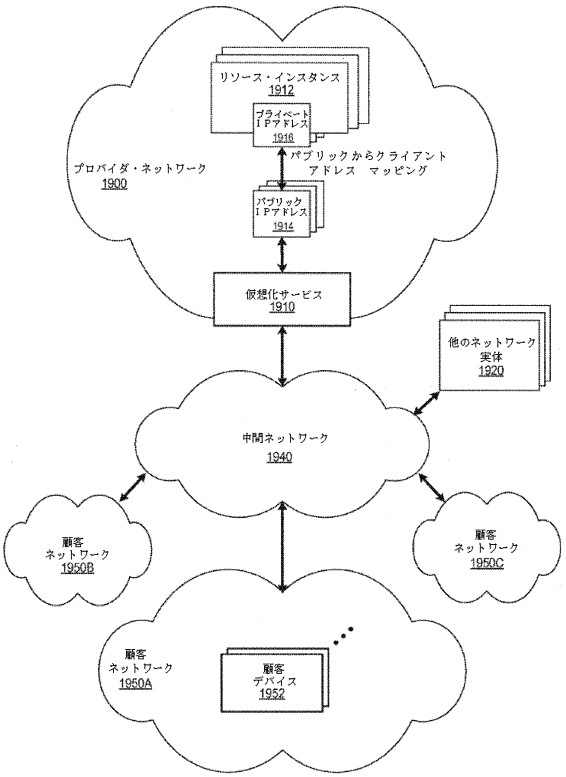
【図３２】

図３２



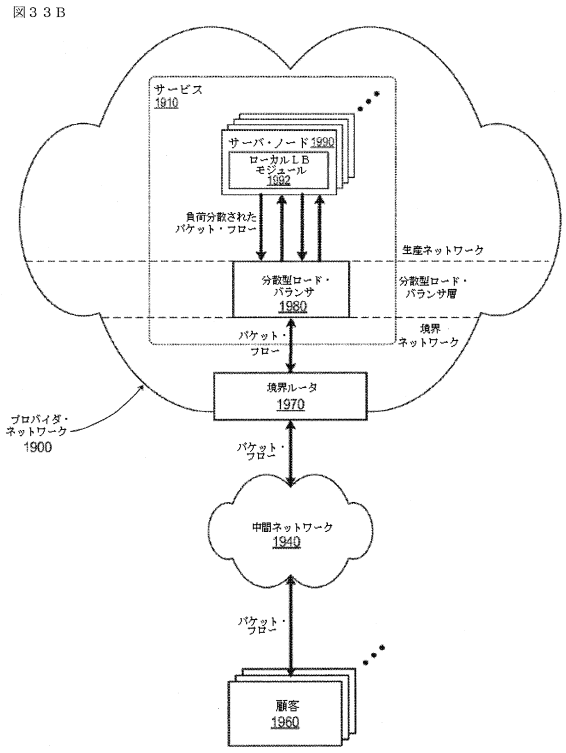
【図３３Ａ】

図３３Ａ



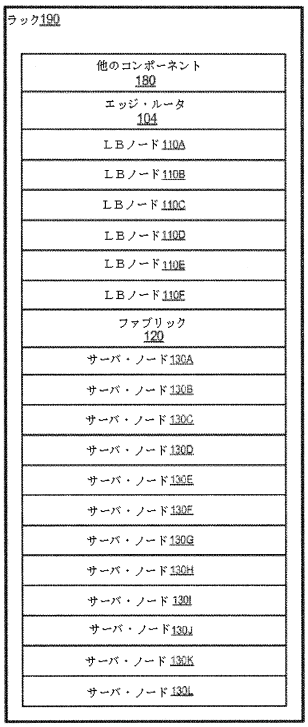
【図３３Ｂ】

図３３Ｂ



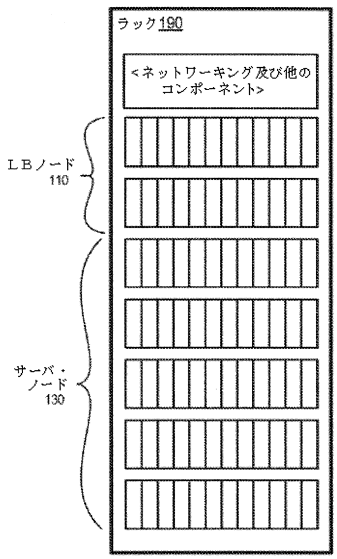
【図３４Ａ】

図３４Ａ



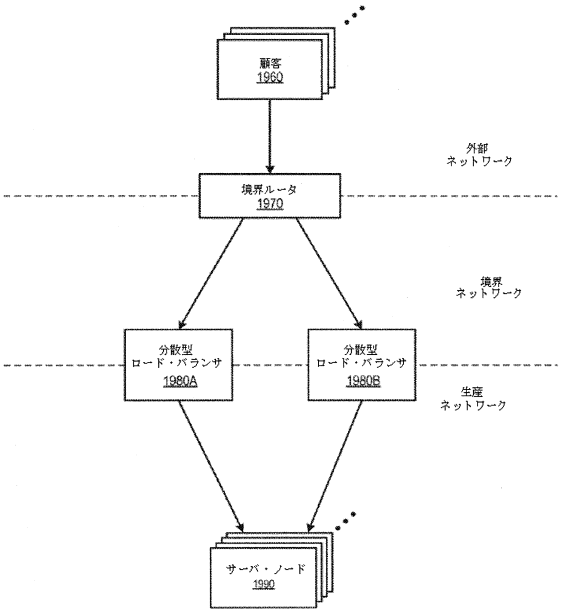
【図３４Ｂ】

図３４Ｂ



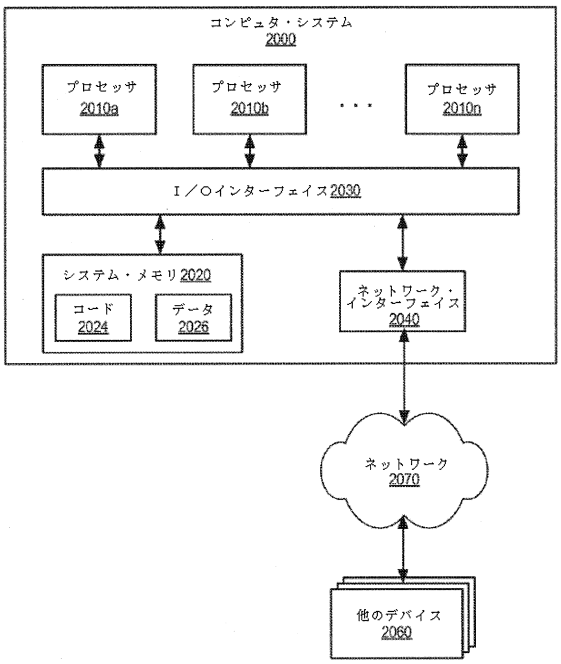
【図３５】

図３５



【図３６】

図３６



フロントページの続き

(72)発明者　ローレンス，ダグラス・スチュワート
アメリカ合衆国・９８１０９－５２１０・ワシントン州・シアトル・テリー　アヴェニュ　ノース・４１０
(72)発明者　スリニヴァサン，ヴェンカトラガヴァン
アメリカ合衆国・９８１０９－５２１０・ワシントン州・シアトル・テリー　アヴェニュ　ノース・４１０
(72)発明者　ヴァイジャ，アクシャイ・スハス
アメリカ合衆国・９８１０９－５２１０・ワシントン州・シアトル・テリー　アヴェニュ　ノース・４１０
(72)発明者　チャン，ファン
アメリカ合衆国・９８１０９－５２１０・ワシントン州・シアトル・テリー　アヴェニュ　ノース・４１０

審査官　寺谷　大亮

(56)参考文献　特開２００５－０２５７５６（ＪＰ，Ａ）
特開２００３－１３１９６１（ＪＰ，Ａ）
特表２０１０－５３１０２０（ＪＰ，Ａ）
特開２０００－０２９８３１（ＪＰ，Ａ）
特開２０１２－０７４９２８（ＪＰ，Ａ）
米国特許出願公開第２０１０／００３６９０３（ＵＳ，Ａ１）

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｇ０６Ｆ　１３／００
Ｈ０４Ｌ　１２／７４３
Ｈ０４Ｌ　１２／８０３
Ｈ０４Ｌ　１２／９５１
Ｇ０６Ｆ　　９／４６