(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第3932994号
(P3932994)
(45)発行日　平成19年6月20日(2007.6.20)　(24)登録日　平成19年3月30日(2007.3.30)

(51)Int.Cl.		F I		
H04L 12/46	(2006.01)	H04L	12/46	Z
H04L 12/56	(2006.01)	H04L	12/56	G
G06F 11/20	(2006.01)	G06F	11/20	310E
G06F 15/00	(2006.01)	G06F	15/00	320K
G06F 15/177	(2006.01)	G06F	15/177	Z

請求項の数 24　(全 30 頁)

(21)出願番号　特願2002-183809(P2002-183809)
(22)出願日　平成14年6月25日(2002.6.25)
(65)公開番号　特開2004-32224(P2004-32224A)
(43)公開日　平成16年1月29日(2004.1.29)
審査請求日　平成17年3月10日(2005.3.10)

(73)特許権者　000005108
株式会社日立製作所
東京都千代田区丸の内一丁目６番６号
(74)代理人　100100310
弁理士　井上　学
(72)発明者　馬場　恒彦
東京都国分寺市東恋ケ窪一丁目２８０番地
株式会社日立製作所中央研究所内

審査官　羽岡　さやか

最終頁に続く

(54)【発明の名称】サーバ引継システムおよびその方法

(57)【特許請求の範囲】
【請求項１】
外部から自分に向けた通信パケットと他へ向けた通信パケットの双方が流れるネットワークノードに接続された複数のサーバマシンを含むサーバシステムにおいて、該複数のサーバマシンのひとつである第１のサーバマシンに障害が発生した場合に、待機状態にある第２のサーバマシンが前記第１のサーバマシンの処理を引き継ぐサーバ引継方法であって、
前記第２のサーバマシンは、
前記ネットワークノード上の前記第１のサーバマシンの入力パケットと、出力パケットを監視し、
前記第１のサーバマシンに対して発行されたトランザクションを内部のキューにエンキューし、
切り替えのトリガがあったときに前記第１のサーバマシンに対してトランザクション処理の停止を指示し、
前記キュー上の残る未処理のトランザクションの処理を引き継ぎ、
前記出力パケットの監視中は前記第１のサーバマシンからの出力パケットによりキューに保存したトランザクションに対する前記第１のサーバマシンでの処理終了を知り、処理終了のトランザクションをデキューすることを特徴とするサーバ引継方法。
【請求項２】
外部から自分に向けたトランザクションと他へ向けた通信パケットの双方が流れるネッ

トワークノードに接続された複数のサーバマシンを含むシステムにおいて、該複数のサーバマシンの一つである第１のサーバマシンに障害が発生した場合に、待機状態にある第２のサーバマシンは第１のサーバマシンの処理を引き継ぐサーバ引継方法であって、
前記第２のサーバマシンは、
前記ネットワークノード上の前記第１のサーバマシンの入力パケットと、出力パケットを監視し、
前記第１のサーバマシンに対して発行されたトランザクションを内部のキューに保存し、
前記出力パケットの監視中は、前記第１のサーバマシンからの出力パケットによりキューに保存したトランザクションに対する前記第１のサーバマシンでの処理終了を知り、処理終了のトランザクションをデキューし、
切り替えのトリガがあったときに前記第１のサーバマシンに対してトランザクション処理の停止を指示し、
前記キュー上に残る未処理のトランザクションをアプリケーションに処理させ、処理結果を返送するための応答パケットを作成して前記第１のサーバマシンを装って前記ネットワークノードに出力することを特徴とするサーバ引継方法。
【請求項３】
外部から自分に向けたトランザクションと他へ向けた通信パケットの双方が流れるネットワークノードに接続された複数のサーバマシンを含み、該複数のサーバマシンの一つである第１のサーバマシンに障害が発生した場合に、待機状態にある第２のサーバマシンがトランザクションに対するキュー処理を代行するようにされたサーバシステムにおけるマシン切り替え方法であって、
前記第２のサーバマシンは、
前記第１のサーバマシンのキュー処理を代行している状態では、前記第１のサーバマシンに伝達されるべきパケットを監視し、かつ監視結果により前記第１のサーバが処理すべきトランザクションをエンキューし、
該エンキューされたトランザクションを処理するべき手続に引き渡す処理及び前記手続の結果を返送するための応答パケットを作成し、
前記第１のサーバマシンのアドレスを送信元アドレスとして用いて前記ネットワークノードに出力し、
前記出力した応答パケットによりキューに保存したトランザクションに対する処理終了を知り、処理終了のトランザクションをデキューし、
前記第１のサーバマシンのアドレスを引き継いだ前記第１のサーバマシンの処理を正規に引き継ぐべきマシンへの切り替えのトリガがあったとき第２サーバマシンのトランザクション代行処理を停止し、
前記キュー上に残る未処理トランザクションを前記正規に引き継ぐべきマシンに引き継ぐことを特徴とするサーバ引継方法。
【請求項４】
前記第１のサーバマシンのアドレスを引き継いだ前記正規に引き継ぐべきマシンは前記第１のサーバマシンであることを特徴とする請求項３記載のサーバ引継方法。
【請求項５】
前記第１のサーバマシンのアドレスを引き継いだ前記正規に引き継ぐべきマシンは第２のサーバマシンであることを特徴とする請求項３記載のサーバ引継方法。
【請求項６】
前記第２のサーバマシンは前記トリガがあったとき引き継いだマシンへの入出力パケットを監視し、前記引き継いだマシンに対して発行された入力パケットをエンキューし、
前記出力した応答パケットによりキューに保存したトランザクションに対する処理終了を知り、処理終了のトランザクションをデキューすることを特徴とする請求項３記載のサーバ引継方法。
【請求項７】

前記未処理トランザクションの処理の引継は前記第２のマシンから前記正規に引き継ぐべきマシンへ未処理のキュー内容を複製する手続きを含むことを特徴とする請求項３記載のサーバ引継方法。

【請求項８】

外部から自分に向けたトランザクションと他へ向けた通信パケットの双方が流れるネットワークノードに接続された複数のサーバマシンを含むサーバシステムにおいて、該複数のサーバマシンの一つである第１のサーバマシンに障害が発生した場合に、待機状態にある第２のサーバマシンがトランザクションに対するキュー処理を代行するサーバシステムにおけるサーバ引継方法であって、

前記第２のサーバマシンは、

前記第１のサーバマシンのキュー処理を代行している状態では、前記第１のサーバマシンに伝達されるべきパケットを監視し、かつ監視結果により前記第１のサーバが処理すべきトランザクションをエンキューし、

該エンキューされたトランザクションを処理するべき手続に持ち込み、前記手続の結果を返送するための応答パケットを作成し、前記第１のサーバマシンのアドレスを送信元アドレスとして用いて前記ネットワークノードに出力し、

前記出力した応答パケットによりキューに保存したトランザクションに対する処理終了を知り、処理終了のトランザクションをデキューし、

前記第２のサーバマシンに対する待機状態になるべきマシンの準備が完了したトリガがあったとき前記第２のサーバマシンはトランザクション代行処理を停止し、

前記第２のマシンが前記第１のサーバマシンのアドレスを引き継いでトランザクションの処理を正規に引き継ぐことを特徴とするサーバ引継方法。

【請求項９】

前記待機状態になったマシンは前記第２のマシンへの入出力パケットを監視し、発行されたトランザクションをエンキューすることを特徴とする請求項８記載のサーバ引継方法。

【請求項１０】

前記待機状態になるべきマシンは前記第１のサーバマシンであることを特徴とする請求項８記載のサーバ引継方法。

【請求項１１】

前記待機状態になるべきマシンは第２のサーバマシンであることを特徴とする請求項８記載のサーバ引継方法。

【請求項１２】

前記未処理トランザクションの処理の引継は前記第２のサーバマシンから前記待機状態になったマシンへのキュー内容を複製する手続きを含むことを特徴とする請求項８記載のサーバ引継方法。

【請求項１３】

通信ネットワークにおけるサーバ引継システムであって、

第１ネットワークノードと、

前記第１ネットワークノードと通信可能に接続する第１ネットワークインターフェースカードと、

第１内部キューとを有する第１キューコンピュータと、

前記第１ネットワークノードと通信可能に接続する第２ネットワークインターフェースカードと第２内部キューを有する第２キューコンピュータとを有し、

前記第１キューコンピュータは自身に向けられた未処理トランザクションを前記第１ネットワークノードから受け取り、該未処理トランザクションを前記第１内部キューに処理完了まで保持し、処理完了時に応答トランザクションを前記第１ネットワークノードに返送してトランザクション管理を行い、

前記第２キューコンピュータは前記第１キューコンピュータに向けられた未処理トランザクションを受信して第２内部キューに保持し、前記第１キューコンピュータが障害状態

となった時に該第１キューコンピュータからのトランザクション管理の引継を実行し、
かつ、前記第２ネットワークインターフェースカードに接続された第２パケットスニフィングモジュールと、該第２パケットスニフィングモジュールに接続された第２コネクション情報バッファを更に有し、該第２パケットスニフィングモジュールは前記第２コネクション情報バッファに保存する情報を用いてトランザクションのうち前記第１キューコンピュータに向けられた未処理トランザクションを選別し、該第２パケットスニフィングモジュールにて選別した未処理トランザクションが前記第２内部キューに保持されるようにし、
さらに、前記第１キューコンピュータが障害状態に陥ったことを検知する第２監視モジュールと、前記第２コネクション情報バッファ及び前記第２内部キューに接続される第２スプーフィングモジュールを更に有し、該第２スプーフィングモジュールは、前記第１キューコンピュータが障害状態に陥った場合に、該第１キューコンピュータを装って該第１キューコンピュータが返送すべきパケットを出力し、前記出力したパケットにより前記第２内部キューに保存したトランザクションに対する処理終了を知り、処理終了のトランザクションをデキューすること特徴とするサーバ引継システム。
【請求項１４】
前記第１ネットワークノードと異なるネットワーク層に属する第２ネットワークノードを更に有し、
前記第１キューコンピュータは、前記第２ネットワークノードに接続された第３ネットワークインターフェースカードと、該第３ネットワークインターフェースカードに接続された第３コネクション情報バッファを有し、
前記第２キューコンピュータは、前記第２ネットワークノードに接続された第４ネットワークインターフェースカードと、前記第３コネクション情報バッファに接続された第４コネクション情報バッファを有し、
前記第１ネットワークノードは上位クライアント層の一部を成し、前記第２ネットワークノードは下位アプリケーションサーバ層の一部を成すことを特徴とする請求項１３記載のサーバ引継システム。
【請求項１５】
前記第１ネットワークノードと通信可能に接続された第５ネットワークインターフェースカードと第３内部キューとを含む第３キューコンピュータを更に備え、該第３キューコンピュータは前記第１キューコンピュータに向けられた未処理トランザクションを受信し、前記第３内部キューに保存し、前記出力した応答パケットによりキューに保存したトランザクションに対する処理終了を知り、処理終了のトランザクションをデキューすることを特徴とする請求項１４記載のサーバ引継システム。
【請求項１６】
前記第１キューコンピュータは第１アプリケーションサーバを含み、前記第２キューコンピュータは第２アプリケーションサーバを含むことを特徴とする請求項１３記載のサーバ引継システム。
【請求項１７】
上位ネットワーク層と、下位ネットワーク層と、前記上位・下位ネットワーク層の間を接続する互いに並列な第１、第２のキューコンピュータとを含むネットワーク通信システムにおけるサーバ引継方法であって、
前記第１のキューコンピュータは、前記上位ネットワーク層から向けられた入力トランザクションを受信し、第１のキューにエンキューし、
前記第２のキューコンピュータは、前記入力トランザクションを受信し、第２のキューにエンキューし、
前記第１のキューコンピュータは、前記下位ネットワーク層に前記入力トランザクションを転送し、
前記第２のキューコンピュータは、前記転送された入力トランザクションを監視し、前記転送された入力トランザクションにより前記第２のキューに保存したトランザクション

に対する転送処理終了を知り、前記第２のキューにエンキューされた前記入力トランザクションに対して、インプットフラグを付し、
前記第１のキューコンピュータは、前記下位ネットワーク層から、前記入力トランザクションへの応答として前記第１のキューコンピュータに向けて発せられた出力トランザクションを受信し、前記第１のキューにエンキューし、
前記第２のキューコンピュータは、前記出力トランザクションを受信し、前記第２のキューにもエンキューする手順を有することを特徴とするサーバ引継方法。
【請求項１８】
前記第１のキューコンピュータは、前記出力トランザクションを前記第１のキューから前記上位ネットワーク層に送出し、
前記送出した出力トランザクションと、前記出力トランザクションに対応する入力トランザクションを前記第１のキューから削除し、
前記第２のキューコンピュータは、前記第１のキューコンピュータが送出した出力トランザクションを監視し、前記出力トランザクションにより前記第２のキューに保存した出力トランザクションの処理終了と前記出力トランザクションに対応する入力トランザクションの処理終了とを知り、前記入力トランザクション及び対応するインプットフラグと、前記出力トランザクションとを前記第２のキューからデキューする手順を更に有することを特徴とする請求項１７記載のサーバ引継方法。
【請求項１９】
前記第１、第２のキューのそれぞれはインプットキュー、アウトプットキュー及びインプットフラグを有し、
前記インプットキューは前記上位ネットワーク層から前記第１のキューコンピュータに向けられ、且つ前記第１のキューコンピュータで受信されたインプットトランザクションを含み、
前記アウトプットキューは前記インプットトランザクションに対応して前記下位ネットワーク層から前記第１のキューコンピュータに向けられ、且つ前記第１のキューコンピュータで受信したアウトプットトランザクションを含み、
前記インプットフラグは受信したインプットトランザクションを前記下位ネットワーク層に転送した時に設定されることを特徴とする請求項１７記載のサーバ引継方法。
【請求項２０】
前記第１のキューコンピュータが障害状態に陥ったか否かを前記第２のキューコンピュータにより監視し、障害状態に陥った時に前記第２のキューコンピュータが前記第１のキューコンピュータの代理でトランザクション通信管理を実行する手順を更に有することを特徴とする請求項１９記載のサーバ引継方法。
【請求項２１】
前記トランザクション通信管理は、前記第２のキューコンピュータが前記第２のキューのアウトプットキューの項目に保存されたアウトプットトランザクションの全てを前記上位ネットワーク層に送出する手順を含むことを特徴とする請求項２０記載のサーバ引継方法。
【請求項２２】
前記トランザクション通信管理は、前記第２のキューコンピュータが前記第２のキューのインプットキューの項目に保存されたインプットトランザクションの全てを前記下位ネットワーク層に送出する手順を含むことを特徴とする請求項２０記載のサーバ引継方法。
【請求項２３】
前記トランザクション通信管理は、前記上位ネットワーク層から前記第１のキューコンピュータに向けて発せられた追加のインプットトランザクションを前記第２のキューコンピュータで受信し、前記第１のキューコンピュータのアドレスを用いて前記第１のキューコンピュータを詐称して前記第２のキューコンピュータから前記追加のインプットトランザクションを前記下位ネットワーク層に送出する手順を含むことを特徴とする請求項２０記載のサーバ引継方法。

【請求項２４】
前記トランザクション通信管理は、前記第２のキューコンピュータで前記下位ネットワーク層から前記第１のキューコンピュータに向けて発せられた追加のアウトプットトランザクションを前記第２のキューコンピュータで受信し、前記第１のキューコンピュータのアドレスを用いて前記第１のキューコンピュータを装って前記追加のアウトプットトランザクションを前記上位ネットワーク層に送出する手順を含むことを特徴とする請求項２０記載のサーバ引継方法。
【発明の詳細な説明】
【０００１】
【発明の属する技術分野】
本発明は障害許容性のあるネットワークシステムに関し、特に、アクティブキューをもつコンピュータのソフトウェアもしくはオペレーティングシステムに障害があった時に、リクエストされた処理を引き継ぐことのできるアクティブ／シュードアクティブ／スタンバイキューを持つコンピュータシステムに関する。
【０００２】
【従来の技術】
コンピュータネットワークが日常生活において一般的になるにつれ、コンピュータネットワークを必要とすることが公私双方の活動において劇的に増大してきている。一般的に、典型的なネットワーク環境ではリクエストを処理する複数のアプリケーションサーバ（APサーバ）とそれに対してリクエストを出す複数のクライアントコンピュータ（ユーザ）が存在する。クライアントコンピュータとAPサーバはクライアントサーバ層とAPサーバ層とにそれぞれ存在し、相互通信を行うノードを持つシステムを構成する。
単一のネットワークであっても多種多様なコンピュータやその他の機器が接続されるため、システムではマシン間の通信を実際に行うことが必要不可欠である。各コンピュータや機器からその他のコンピュータへ直接通信をリンクすることはコストがかかるため、ネットワークはある種の通信の共有を行っている。ネットワーク通信手段のうち最も一般的な種類として、ネットワークに接続したマシンごとに個別のアドレスを与え、送信側コンピュータ（センダ）と受信側コンピュータ（レシーバ）のアドレスを含む細分化した単位（パケット）を用い、ネットワークを介して互いに情報を交換している。この方法では、各マシンは自身に向けられた情報を識別し、それに応答する。こうした通信手段の一般的なプロトコルの1つとして、TCP/IP(Transaction Control Protocol / Internet Protocol)がある。
ネットワーク上の各マシンは、ネットワークインターフェースカード（本明細書では以下NICと呼ぶ）を介し、有線あるいは無線の通信手段によって物理的に相互接続されている。NICはそれぞれマシンを一意に識別するためのMAC(マシンアドレス)を有する。ネットワーク上を行き来するTCP/IPパケットはデータだけでなく送信側と受信側コンピュータ双方のIPアドレス（実アドレスやエイリアスアドレス）やMACアドレスといったアドレス情報が含まれている。
【０００３】
NICはネットワーク上を行き来するパケット全てをブラウズし、自身のMAC/IPアドレスを含むパケットを探している。NICは自身のアドレスを送信先として含んでいるパケットを認識すると、そのパケットを受信する。つまり、そのNICのアドレスが含まれないパケットはNICが無視することになる。
NICには全パケットを受信する“プロミスキャスモード(promiscuous mode”がある。プロミスキャスモードでは、NICのMACアドレスが含まれるかどうかを判断せずに全てのパケットをNICが受け取る。こうした全てのプロセスを受け取る処理は“スニフィング（sniffing）”として知られる。この処理はフォルトトレラントシステムを設計する上で有効なものであることもある。
前記のように、あるシステム上を行き来する全てのTCP/IPパケットには送信側とパケット受信予定側との双方のIPアドレスとMACアドレスが含まれる。これらのIP・MACアドレスは

NICがインストールされているコンピュータ上で実行されているネットワークドライバによって付与される。フォルトトレラントシステムの中には、「送信側」のIPアドレスとMACアドレスをNICに設定し、実際にパケットを送信するマシン以外のマシンのIP・MACアドレスを反映させる、つまり送信元アドレスを詐称するものがある。この処理を通じて、NICは「間違った」IP・MACアドレス情報を有するパケットを受信するが、パケットを送信したコンピュータに応答することはない（コンピュータの正しいアドレスを持たないため）。この処理はパケットの“スプーフィング（spoofing）”として知られ、こうしたスプーフされたパケットに対する応答はスプーフされたマシンへと送られることになる。この技術は自身を識別されたくないクラッカーによってしばしば利用される。TCP/IPネットワークを介して、別のマシンと通信するためには、パケットの送信側コンピュータが受信側コンピュータのIPアドレスとMACアドレスとを知る必要がある。NICのMACアドレスとIPアドレスを対応付ける手法はARP（Address Resolution Protocol）として知られる。ホストBのMACアドレスを決定するため、ホストAはホストBのハードウェアアドレスを求めるARPリクエストを送出する。ホストBはネットワーク上のリクエストを検出し、効かれたIPアドレスを持つインタフェースに対するハードウェアアドレスを含むARPリプライを返す。ホストAは後にTCP/IPパケットを送信する時に利用できるようにこのハードウェアアドレスをARPキャッシュに保存する。

Gratuitous ARPとして知られる同様の処理がある。Gratuitous ARPはARPリクエストがなされていない時もARPリプライを送信することである。Gratuitous ARPリプライは通常ハードウェアアドレスをブロードキャストするため、LAN（Local Area Network）上の全ホストがARPリプライを受け取ることになる。Gratuitous ARPリプライを受け取ると、ネットワーク上の全ホストでGratuitous ARPの反映のため、ARPキャッシュの更新が行われる。Gratuitous ARPはIPアドレスとMACアドレスとの関連付けが変更となった時、すなわち、IP引継が生じた時に利用される。Gratuitous ARPにより、変更されたことがシステム上の他のホスト全てに実際に通知され、その結果後の通信には変更が反映されることになる。

こうしたネットワーク通信機構はとても複雑であるため、あるマシン、またははネットワーク上のあるノードが正常に動作しているかを判断するのは難しい事が多い。マシンは同時に複数マシンとの通信を行っていることがあり、その場合はエラーが発生しても認識するのが難しいことがありうる。さらに、エラーがネットワークマシンで発生した時は、他のマシンにこのエラーを知らせなければならないか、あるいは、バックグラウンド処理により、ネットワーク上の他のコンピュータに知らせずにエラーを克服しなければならない。こうした訂正処理は他のコンピュータに対して「透過的」と見なされる。

こうしたエラー訂正やシステム移動の分類の1つとして、フォルトトレラント処理が良く知られている。フォルトトレラント処理における主な問題の一つとして、障害状態に陥ったコンピュータの処理を別のコンピュータに（一時的ないしは永久に）引き継ぐことがある。従来、この引継はある限定された方法か、あるいはユーザやクライアントコンピュータに何らかの変更を加えること（つまり、透過的でない方法）でのみ実現される。

【０００４】

最近のフォルトトレラントシステムの中で、幾つかについて後述する。

特開平11-296396 号公報は現用系ホストと待機系ホストとを含む障害回復システムを開示する。各ホストは搭載NICと接続されている。また、互いに相手のホストのNIC（現用系から待機系ホストのNIC、待機系から現用系ホストのNIC）にも接続されている。もし、系障害が現用系で発生すると、待機系は現用系ホストのNICを（直接接続によって）利用する。これにより、MACアドレスとIPの引継が実現される、

このシステムは追加ハードウェア（ホスト・NIC間の接続）を必要とするため、望ましくない。さらに、障害が現用系ホスト自身ではなく、そのNICに生じた場合には、必須であるMACアドレス引継を実現することが出来なくなる。さらに、このシステムはMACアドレス引継しか行うことができないという限定がある。現用系ホストが障害状態に入っている最中に処理していたトランザクションは全て失われるため、待機系ホストで実行して回復さ

せることは出来ない。
米国特許6,049,825号公報は二重化されたネットワークアダプタを切り替えることで、あるレベルのフォルトトレランスを実現する手法を開示するで。このシステムではGratuitous ARPにより、障害の検出時のNIC切り替えを実現している。Gratuitous ARPの手法は、クライアントに対して透過ではないため、望ましくない。言い換えれば、クライアントコンピュータが障害の発生を認識してしまう。加えて、NIC切り替えはGratuitous ARP処理により生じるため、切り替えが終了するまでには時間がかかることになる。さらに、前述のケースで述べたのと同様な問題として、提供するのはNIC切り替えのみであり、障害発生時に処理中だったトランザクションは全て失われ、クライアントに再送してもらう必要がある。特開2000-056996号公報は互いに通信することのできる複数のキューが複数のクライアントと複数のAPサーバ間に散在する方法を記載する。クライアントのリクエストは第一のキューにエンキューされ、そのリクエストはクライアントが次のリクエストをキューに送信する前に、第二のキューにもエンキューされる。リクエストを処理中に第一のキューで障害が発生した場合、第二のキューがAPサーバに処理をするようにリクエストを送信することができる（第二のキューでの障害に対しては、同様に別のキューへリクエストを複写すればよい）。
まず、この手法は一つ以上のキューに対して全部のリクエストを渡すために大きなオーバヘッドを生じ、データ転送速度に依存する通信システムには十分ではない。次に、この手法では、リクエスト喪失の可能性は軽減できるが、IP引継処理は行われない。さらに、キューに障害が発生した場合、キューシステムとサーバ間の接続が失われ、後々のパケットを再度ルーティングするためにエラーがクライアント側に認識されてしまう。さらにこれら全てがオーバヘッドを生じることになる。
特開平10-135993号公報はMACアドレスとIPアドレスの引継機能をシステムにすることを開示する。システムは複数のMACアドレスを保持できるアドレスバッファを持つ。システムで障害が発生すると、待機系コンピュータがエイリアスIPアドレスを利用したIP引継と、このアドレスバッファを利用したMAC引継とを行う。元の現用系コンピュータ（現在は待機系）のシステムリブート処理では、ネットワーク環境が正常に機能しているかどうかを元のIPアドレスとMACアドレスを用いて、テストする。
このシステムはMACアドレスとIPアドレス引継機能を有するが、障害発生時にホスト上で実行中であったリクエストに対して対応する機能を有していない。また、これらのリクエストは、クライアントが再送を行うまでの間は喪失したままである。さらに、上記のMACアドレスとIPアドレス処理は十分ではない　これらの引継処理が実行に時間を必要とするからである。
特開2000-322350号公報は障害の発生したサーバをバイパスするようにリクエストを再度ルーティングするシステムを開示する。そのシステムはクライアントと複数のAPサーバとの間にスイッチ装置を含んでいる。クライアントからAPサーバへのリクエストは全てこのスイッチ装置を経由するため、APサーバに障害が発生した場合、スイッチ装置が代替サーバにリクエストを再度ルーティングする。この再ルーティングはクライアントに対して透過的である。
まず、ネットワークシステム上に追加のハードウェア層が必要となるために、このシステムは適当ではない。また、システムはAPサーバにおける障害（つまり、スイッチ装置より“下位”の障害）を隠蔽するだけである。それゆえ、スイッチ装置またはその“上位”で発生した障害は対応できず、クライアントに対して透過的でなくなる。さらに、このシステムは障害が発生した時にそのアプリケーションサーバで処理中であるリクエストに対して対応していない。
【０００５】
【発明が解決しようとする課題】
総じて、これらの従来システムは現用系コンピュータの障害発生時における一時的ないしは永久的なIP/MACアドレス引継問題に対し、透過的かつ効果的な解決方法を与えてはいない。あるコンピュータから別のコンピュータへと処理を透過的に移行することと同時に、

障害発生時にキューに入れられたが現用系コンピュータから外部に送信されていないようなリクエストやトランザクションを一つも失わずにこの操作を行うことが満たされる必要がある。 本発明は従来システムの持つ上述の幾つかの問題に対して、対応するものである。

【０００６】

【課題を解決するための手段】

少なくとも一つの典型例において、本発明は現用系コンピュータが障害状態に陥った場合にネットワークトランザクションリクエストの処理を現用系から待機系コンピュータへと透過的に切り替えるシステム及び手段を提供する。切り替えは、現用系コンピュータが再び通常状態に戻るまで待機系コンピュータが一時的に現用系を装ってパケットをスプーフする仮引継処理(pseudo-takeover)と、待機系コンピュータまたは第三のコンピュータが前現用系コンピュータのIP/MACアドレスを引き継いで、新規現用系コンピュータとなる引継処理(full takeover)からなる。

システムは、共有ディスクや直接通信接続またはその他の手段により相互通信接続された現用系と待機系キューコンピュータからなる。現用系コンピュータはAPサーバ層へ、クライアントネットワーク層（“上位層”や“外部”とも表記している）から、もしくはAPサーバ層からクライアントネットワーク層へ送信されるトランザクション全部を保持する現用系キューを含む。同様に、待機系コンピュータは現用系キューと絶えず同期を行う待機系キューを有する。現用系コンピュータが障害状態に陥った場合、待機系コンピュータは障害前の現用系キューと一致した待機系キューを持つため、待機系コンピュータによって現用系コンピュータの機能を引き継ぐことができる。

待機系コンピュータは、現用系と待機系が接続されているネットワーク上の書くノードに転送されるパケットを監視するパケットスニフィングデバイス（スニファー）を各NICに有する。現用系コンピュータからのネットワーク識別情報（例えば、IPアドレス・MACアドレス・シーケンス番号・暗号鍵など）を受信することにより、待機系コンピュータはこれらを用いてどのトランザクションやパケットが現用系に対して送信されたか、どのトランザクションやパケットが現用系から送信されたかを認識する。待機系コンピュータはこれらのパケットを捕まえ、待機系キューにエンキューしたり、エンキュー済みパケットをデキューしたりする。

待機系コンピュータは（現用系コンピュータ障害により）仮現用系モードにある時に現用系コンピュータを装ってパケットを送出することを可能とするパケットスプーフィングデバイスもまた有する。このスプーフされたパケットは待機系（仮現用系）コンピュータではなく現用系コンピュータを示すパケット送信元アドレス情報を含んでいる。仮現用系コンピュータから送出された各スプーフ・パケットは送信先コンピュータからは現用系コンピュータが送信してきたかのように見える。このようにして、待機系コンピュータは仮引継処理を実行する。

待機系コンピュータは完全にIP/MACアドレスを引き継ぐ機能もまた持つ。現用系コンピュータで障害が発生すると、待機系コンピュータは仮現用系モードに移行し、障害中の現用系コンピュータのIP/MACアドレスにリクエストがなされている間は一時的にパケットをスニフおよびスプーフし続ける。これらのアドレスが待機系コンピュータに再割り当てされると、待機系コンピュータは“新規”現用系コンピュータとなる。これと同様の方法で、待機系コンピュータは第三のコンピュータが新規現用系コンピュータとなる場合にも対応することができる。さらに、この（または他の）第三のコンピュータが現用系と同期されたキューを持つことで待機系コンピュータになることもできる。

さらに、システムの別の携帯として、多様な引継処理を補助することが可能である。例えば、一つ以上のAPサーバ（またはAP層にある別マシン）によって、障害状態に陥った“現用系”APサーバが処理していた役割を“待機系”APサーバにより、引き継ぐような現用／待機系設定による構成することができる。こうした引継処理は全て上位のクライアントネットワーク層に対して透過的である。これらのテーマの多様な入れ替えは可能である。

【０００７】

【発明の実施の形態】
本発明に関する図と説明は、本発明を鮮明に理解するために適当な要素を示すために簡単化されており、既知の要素については省略していることを理解されたい。本技術中で従来技術の中には、本発明を実装するために他の要素が望ましく、かつ／または、必要とされると思われるものが幾つかある。しかし、技術中のこれらの要素は既知であり、本発明の理解を容易にするものではないので、ここでは説明しない。以下では、添付の図に関して詳細に説明していく。
本発明は現用系キューコンピュータで発生した障害を透過的に保証するキューコンピュータ引継システムを提供しようとするものである。
【０００８】
以下では、代表的な例として、上位のクライアントネットワーク層と下位のAPサーバネットワーク層との間に存在するキューコンピュータシステムに関するものを取り上げるが、本発明はその範囲内で任意の二者ネットワーク間の相互関係を包括するように適用可能である。
さらに、議論簡便化のため、幾つかの注釈が用いられている。
一つ目として、クライアント層から発信されAP層へ送られるトランザクションや要求をまとめて“インプット”トランザクションと記述する。同様に、AP層から発信され、クライアント層に送信されるトランザクションをまとめて“アウトプット”トランザクションと記述する。一般に、クライアントはインプットトランザクションをAPサーバに対し送信し、何らかの処理を実行させ、返信される結果がAPサーバからクライアントへ戻るアウトプットトランザクションである。
次に、本発明に関するプロセスにおけるステップの中には、コンピュータが別のコンピュータにトランザクションを送信したり、トランザクションを送信することを前提とした何らかのアクションを行ったりするものが幾つかある(すなわち、現用系コンピュータはクライアントにアウトプットトランザクションを送信し、その後、対応するインプットトランザクションをデキューする)。たとえ必ずしもそのことが明記されてない場合でも、送信されたトランザクションが送信元に受信され、肯定応答(ACK)が返って初めて行われるべきアクションが生じる。すなわち、トランザクション通信は標準的なネットワークの挙動に則ったものになっている。もし肯定応答が受け取れない、もしくはネットワークの慣習が満足されない場合は、本発明ではトランザクションはまだ送信が完了していないものと見なす。
図1は本発明の好適な実施例である現用／待機システムモデルの高位システムブロック図を表している。図1のシステムは一般に次の3つのネットワーク層から構成される。(1)複数のクライアントコンピュータを含む上位クライアント層（“外部”）、(2)アプリケーションサーバを含む下位アプリケーション層（APサーバ）、(3)クライアントからAPサーバへのインプットトランザクションと、それに対応したAPサーバからクライアントへのアウトプットトランザクションとの完了を追跡する、つまり、トランザクション管理を行う、キューコンピュータ中間層である。キュー層はクライアントとAPサーバ間のバッファとして動作し、本発明の少なくとも1つの典型によって、フォルトトレラント処理を行う。
図1で、キューコンピュータ層は現用系キューシステムA（または“引継元”や“現用系”コンピュータ）と待機系キューシステムB（または“引継先”や“待機系”コンピュータ）からなる。上位クライアント層と下位APサーバ層との間の通信を実現するために、各キューコンピュータは各層に対するNICを持つものとする。現用系と待機系のコンピュータは共有ディスクや直接ケーブル接続のような通信手段を通じて、互いに通信しあうものとする。
通常のネットワーク操作（つまり、現用コンピュータが障害でない状態）にある間は、インプットトランザクションをクライアント層から現用系キューコンピュータのクライアント側NICに対し現用キューコンピュータの実IPアドレスAまたはエイリアスIPアドレスCをあて先として送信することで、クライアントはAPサーバに要求を行う。現用コンピュータはインプットトランザクションをエンキューし、APサーバ層にある送信先のAPサーバへ

と送信する。受信したインプットトランザクションに対応した要求が処理された後、実IPアドレスXまたはエイリアスIPアドレスZをアドレスとして持つ現用コンピュータに対してAPサーバ層からアウトプットトランザクションを送信することで、APサーバはクライアントに対して応答を送信する。

【０００９】

このアウトプットトランザクションはクライアント層へと送られ、指定されたあて先のクライアントサーバによって受信される。現用系キューコンピュータは送信されたアウトプットトランザクションと最初に受信したインプットトランザクションとを対応させ、双方のトランザクションを現用系キューからデキューする。このデキュープロセスにより、インプットトランザクションがAPサーバから送信されたアウトプットトランザクションに対して適切に応答したことが示される。

エンキューとデキューを行う（以下に詳述）ためには実際には様々な方法があることをここでは明記しておく。しかし、ほとんどのシステムでは、クライアントからのインプットトランザクションは、現用系コンピュータが最初に受信すると同時にエンキューされ、このエンキューされたインプットトランザクションは、クライアントコンピュータからアウトプットトランザクションの配送に対するある種の受信通知を受け取るまでデキューされない。 例えば、ネットワーク接続環境においてトランザクションを受信した際に送信先のコンピュータから受け取るような一般的な“ACK”シグナルがこの承認作業にあたる。

前記の現用系コンピュータ通信が行われると同時に、待機系コンピュータではネットワーク上の全トランザクショントラフィックを監視している。既知のスニフ（監視）手続きを用いることで待機系コンピュータはネットワークのクライアント層とAPサーバ層との両方のパケットを全てスニフする。待機系コンピュータは、クライアントから現用コンピュータ（実IPアドレスAまたはエイリアスIPアドレスC）へのインプットトランザクションや、APサーバから現用コンピュータ（実IPアドレスXまたはエイリアスIPアドレスZ）へのアウトプットトランザクションが送信されたことを認識すると、現用系コンピュータと同様の方法により待機系キューからそのトランザクションをエンキュー／デキューする。もし、現用系コンピュータがクライアント層（アウトプット）とAPサーバ層（インプット）の双方へと送信したトランザクションを待機系コンピュータがスニフしているならば、待機系コンピュータが現用系コンピュータが障害を起こした時に引き継ぐことのできる機能を有することになる。追加の機能については以下で述べる。

キューコンピュータシステムはさらに現用系と待機系キュー間でデータ交換を含む。この通信は何らかの直接的なケーブル接続や共有ディスク、あるいは二つのシステム間での任意の情報共有手段によって行われても良い。二つのキューコンピュータ間で共有される情報の第一は、現用系キューと待機系キュー間のトランザクションキューの一貫性を保証する。例えば、現用系コンピュータが処理を行うようにAPサーバやクライアントコンピュータから、もしくはそれらに送信される要求トランザクションをエンキュー／デキューするときに、待機系コンピュータのキューは更新され、この変更が反映される。待機系コンピュータは現用系コンピュータと同一のキュー情報を保持し、これによって、現用系で障害が発生した際にエンキューされたインプットトランザクションのうち、APサーバ層からの応答（対応するアウトプットトランザクション）が来ていないトランザクションの処理を引き継ぐことができる。キューの同期が行われない場合には、現用系で障害が発生した場合に、幾つかの要求が失われることがある。

現用系と待機系コンピュータはAP状態の更新情報を共有することもある。これにより、AP層の1つないしそれ以上のサーバの情報がどう変化しても各キューコンピュータは認知することができ、その結果、各キューコンピュータは同様の方法で要求を転送することができる。さらに現用系と待機系コンピュータ間で送信された情報を監視するシステムが存在する。これにより、待機系キューコンピュータは現用系コンピュータを監視し、現用系で障害が発生したことを正確に知ることができる。障害が検知されると、待機系コンピュータは障害に対処しようと幾つかの方法を試みる。この方法には、最終的な現用系コンピュータのシステムレベルリブートを含む。リブートと同時に、待機系コンピュータではエン

キューされたが未処理状態にあるインプットトランザクションに対する処理が実行される。

図2は本発明において現用系コンピュータで障害が発生した際の高位のシステムブロック図である。引継処理を理解しやすくするため、現用系コンピュータは引継元として、待機系コンピュータは引継先として図中では記している。図2において、クライアントはインプットトランザクションを上位層からエイリアスIPアドレスCに対して送信するが、リクエストは意図したAPサーバに到着しない。引継元で何らかの障害が発生している。しかし、このインプットトランザクションは引継先コンピュータによりスニフされており、引継先コンピュータ内の待機系キューに保持されている。

引継元と引継先コンピュータ間で行われるシステム監視機能によって、引継先コンピュータは短時間で引継元コンピュータが機能していないことを検出することができる。そして、まず、引継先コンピュータが引継元コンピュータにエイリアスIPアドレス（C、Z）を開放するように要求する。もし、これに対し何の応答もなければ、引継先コンピュータは引継元コンピュータに対してハードウェアリセット（強制リセット）コマンドを送信し、（障害を回復するために）引継元コンピュータを強制的にリブートさせる。

引継先コンピュータはその時、少なくとも二つの状態になる：仮現用系と現用系である。仮現用系モードでは、引継元コンピュータはクライアント層から受け取った（かつ、エンキューした）が、対応するアウトプットトランザクションによるAPサーバより応答がないようなインプットトランザクションの処理を補助する。この処理の補助は、引継先コンピュータは送出パケット内の送信元のIPアドレスとMACアドレスを変更し、引継元コンピュータを装ってパケット送信することで実現され、各パケットは（現在リブート中にある）引継元コンピュータから送信されているかのように見える。これにより、パケットの受信側はパケットが引継元コンピュータから送られたものだと認識され、引継元コンピュータかの障害はシステム上の他のコンピュータ（クライアントやAPサーバ）に対して透過的にみえることになる。この意図的なアドレス変更はパケットの“スプーフィング”として良く知られている。

引継先コンピュータはシステム上の全パケットをスニフし、（リブート中の）引継元コンピュータに向けられたものを保存し続ける。これらスニフ・パケットはエンキューされ、引継先コンピュータは、引継元コンピュータのIPアドレスやMACアドレスを実際に獲得すること無しに、実際に引継元コンピュータと同様の処理を行う。引継元コンピュータがリブートに成功し、IPアドレスC、Zの制御を再び獲得した場合、引継先コンピュータは（引継元コンピュータで障害が発生した時より後に変更されている）残りのキュー情報を引継元コンピュータへと送信し戻す。引継元コンピュータは引継元を装ってパケットをスプーフするのを中止し、再び待機状態へと移行する。

これとは別に、引継先コンピュータが現用系モードに移行し、引継元コンピュータのIP/MACアドレスを実際に引き継ぐこともある。例えば、引継先コンピュータが、引継元コンピュータで障害が発生したことを認識し、IP/MACアドレスを開放させるため引継元コンピュータを強制リブートさせた場合、引継先コンピュータが仮現用系モードに移行し、引継元コンピュータを装ってトランザクションをスプーフし始める。しかし、引継元コンピュータがリブートされて、現用系モードに復活するのを待つのではなく、引継先コンピュータは引継元コンピュータに元々割り当てられていた（実またはエイリアス）IPアドレスとMACアドレスの使用をシステムに対して要求する。引継先コンピュータは引継元コンピュータが以前使っていたIPアドレスとMACアドレスを受け取ると、現用系モードへと移行し、（現用系キューとなっている）そのキューの中のリクエストを処理しつづける。実際に、他のシステムのユーザ（クライアントやAPサーバ）から透過的な引継を行うことによって、引継先コンピュータは完全に引継元コンピュータに取って代わる。

前現用系の引継元コンピュータがリブートされると、現現用系の引継先コンピュータは引継元コンピュータにリクエストキューを全部送信し、引継元コンピュータが待機系モードに移行する。このようにして、引継元もしくは引継先コンピュータはシステムでの役割が根本的に入れ替わる。今度は、前者の引継元コンピュータがパケットをスニフし、現在の

現用系キューコンピュータで障害が発生するのを待つ。障害が起きると、上述と全く同様の処理が行われる　二つのコンピュータの役割が交替しただけである。
引継先コンピュータの三番目の選択肢として、新たなコンピュータを利用することがある。この場合、引継先コンピュータが仮現用モードに移行し、引継元コンピュータを装ってトランザクションのスプーフを開始した後、引継先コンピュータが第三の新たなコンピュータへ現用系を完全に引き継ぐことができる。この第三のコンピュータは引継元コンピュータのIP/MACアドレスの使用を引き受ける。その後、引継先コンピュータは待機系モードに移行し、新たな現用系コンピュータに障害が生じるのを待つ。同様にして、（前現用系の）引継元コンピュータは上述のようにして、待機系モードへ移行することもある。待機系コンピュータとして利用されるキューコンピュータの数に制限はない。従来手法の限界を良く理解するものとして、図3に現用／待機系キューコンピュータを用いた従来の障害系切り替えの高位ブロック図を示す。図3で、障害が引継元コンピュータで発生すると、引継先コンピュータは引継元コンピュータのIPアドレスCを獲得する要求を出し、その結果、獲得する。それから、引継先コンピュータはGratuitous ARPを送り、たとえIPアドレスが同じであっても、前現用系と新たな現用系コンピュータとは一致しないため、クライアントに対して再送を要求することがある。クライアントコンピュータは、元のリクエストトランザクションがタイムアウトするか、Gratuitous ARPを受け取るかすると、元のリクエストトランザクションを再送する。この時、明らかにクライアントコンピュータは引継元コンピュータで障害が起きたことを認識し、クライアントコンピュータではこの障害によって生じる問題を回復するために追加の処理（例えば、トランザクションの再送や、ARPキャッシュの更新）を行う必要がある。この障害系切り替えはクライアントコンピュータに対し、透過ではなく、それ故、十分なものではない。
図4は従来の障害系切り替えと本発明の少なくとも一つの実施例とを比較した一般的なタイミング図である。図は引継元コンピュータで障害が発生した時を始点とし、上から下に向かって時間順に表されている。図4の左側は、従来方法により、障害が引継元で発生したことを次の二つのうちどちらかの方法で認識しているのを示している：インプットトランザクション送信に対する応答を受け取らずに前もって設定された時間だけ待つ（コネクションタイムアウト）場合、現用系コンピュータのIPアドレスの開放と要求を問い合わせる場合である。新しいMACアドレスをクライアントコンピュータに認識させるために、現用系コンピュータはGratuitous ARPをクライアントに送出し、これによりクライアントがそのMACアドレスキャッシュを更新することができる。更新がなされると、クライアントはシステムの障害によって喪失されたリクエストを再送することが必要がとなり、こうしてIPとMACアドレスがすっかり変更されたことが上位のクライアント層によって認識される（透過的でない）。
図4の右側は、本発明の実施例におけるタイミングを表している。引継元コンピュータの状態は引継先コンピュータにより絶えず監視されている為、引継先コンピュータは従来のGratuitous ARPのシステムよりも高速に障害を検知することができる。今回、引継先コンピュータは仮現用系モードに移行し、システム上で対象となるリクエストのスニフとエンキューを続けると同時に、引継元コンピュータに前もって保持されていたリクエストに対するスプーフを開始する。仮現用系モードは現用系コンピュータがリブートするまで継続され、また、場合によっては、引継先コンピュータが障害を発生した引継元コンピュータのIPアドレスとMACアドレスを要求し、その先のある段階で現用系モードに以降することがある。もし、完全な引継が求められてない場合は、引継元コンピュータは制御を（リブート後の）引継元コンピュータへと戻し、これにより引継元コンピュータは再び現用系モードに、引継先コンピュータは再び待機系モードとなる。
どちらの場合でも。仮現用系モードの間、引継先コンピュータはインプットとアウトプットトランザクションの処理を行っている。従来手法では、クライアントコンピュータがこれまでのインプットトランザクションを再送するまで、こうした処理は行なわれない。それゆえ、“時間的利得”部分の矢印が本発明を適用することにより、従来手法に対する時間の削減幅となる。実際に、従来手法ではGratuitous ARPによる通知後にクライアントが

最初のリクエストを再送するまでかかるのに対し、本発明では仮現用系処理が開始されると同時に障害を回避することができる。
図5は本発明の低位ブロック図である。このブロック図の構成要素について最初に説明し、次に図を参照しながら、本発明の様々な実施例の説明を行う。さらに、様々な処理フローチャートを参照しながら、発明の追加的実施例と機能を述べる。
図5は本発明における引継元505と引継先510との両方のコンピュータのシステムブロック図を表している。引継元コンピュータ505と引継先コンピュータ510はともに上位クライアントネットワーク層（外部）512と下位APサーバ層514とそれぞれハブ516、518を介して接続されている。これらのネットワーク層にアクセスするために引継元コンピュータ505と引継先コンピュータ510はそれぞれ二つのNICを持つ。各コンピュータの第一のNICは上位層（520、550）に接続され、第二のNICは下位層（522、552）に接続されている。
引継元コンピュータ505は初期状態及び通常キュー操作時において、現用系キューコンピュータになる。クライアント側NIC520は上位層512からAPサーバ514に向かって送信されたインプットトランザクションを受信する。このインプットトランザクションの応答として下位APサーバからクライアントに向かって送信された対応するアウトプットトランザクションについても同様の処理を行う。二つの引継元コンピュータのNIC 520、522はそれぞれのNICに対し（IP/MACアドレス双方が）正しく割り当てられたパケットを受信し、対応する。
このNIC 520、522はそれぞれ、現用系コンピュータが上位クライアント層512や下位APサーバ層514に対する正規のトランザクションを生成するのに必要となる情報を保持するコネクション情報バッファ540、542と接続されている。この現用系バッファ情報にはNICの実IPアドレスもしくはエイリアスIPアドレス、MACアドレス、ポート番号、シーケンス番号といった種類の情報が含まれている。この情報は現用系コンピュータで稼動中のNICやオペレーティングシステムから展開される。待機系コンピュータ510でのスニフやスプーフを可能にするため、こうしたコネクション情報バッファはそれぞれ、待機系（引継先）コンピュータに存在する同様のバッファ564、566に接続されている。このバッファについては以下で述べる。
現用系コンピュータ505は、待機系コンピュータ510で現用系コンピュータ505が障害状態に陥ったかどうかを検知可能にする機能を持つ監視モジュール546も有する。できれば、現用系監視モジュール546は待機系監視モジュール580に対し、現用系コンピュータの処理が正常実行されていることを伝える信号を一定間隔で送信する。障害が発生すると、現用系コンピュータ505から待機系コンピュータ510への、この“OK”信号送信が中断され、これにより、待機系コンピュータは現用系コンピュータが障害にあったことを判断することができる。さらに、監視モジュール546は、現用系コンピュータに障害が発生した場合に待機系コンピュータから現用系コンピュータへのリセットコマンド発行（これにより現用系マシンがリブートされる）を可能とするために現用系コンピュータに対するハードウェアリセット機能を持つ。
現用系コンピュータ505は、上位クライアント層512から送信されており、APサーバ層514からアウトプットトランザクションによる応答をまだ受け取る必要のあるインプットトランザクションを全て保持している内部の現用系キュー530と接続された（分割ないしは結合されることもある）エンキュー／デキュー用のキューモジュール524も有する。さらに特に、現用系キュー530は以下の三つの主要構成要素を含むことがある：エンキューされたクライアントからのインプットトランザクションを保持するインプットキュー532、APサーバ514から受信済みであるが、クライアント層512への転送が完了していないアウトプットトランザクションを保持するアウトプットキュー534、受信したインプットトランザクションがAP層に送信済みであることを記録しておくインプットフラッグキュー536である。キュー530は従来のメモリ構造、パイプライン、ラッチやフリップフロップ、ディスクといった電子的情報を保持することの出来る任意の構造をとってもよい。
まとめると、現用系キュー530は以下のような処理を行う：(1)受信したインプットトランザクションをインプットキュー532にエンキューする；(2)受信したインプットトランザク

ションをAP層514へ送信し、インプットトランザクションの配送に成功したことを示すようにインプットフラグキュー536にフラグをセットする；(3) 現用系コンピュータ505は。受信したインプットトランザクションの応答としてAPコンピュータ514から送信されたアウトプットトランザクションを受信し、現用系キュー530のアウトプットキュー部534に保存する；そして、現用系コンピュータは受信したアウトプットトランザクションを適当なクライアントコンピュータへと送信し、このインプット／アウトプットトランザクションと、
対応するインプットフラグの全てをデキューする。
トランザクションのエンキューやデキューに関する情報は、キュー処理されたトランザクション自身と同様に、待機系コンピュータ510にコピー・転送するために現用系キュー情報バッファ544に送信される。これにより、現用系と待機系コンピュータ双方における内部キューが常に互いに同期をする（同一データを保持する）ことが可能となる。もし、同期処理に障害が発生した場合、キュー情報バッファ544は待機系コンピュータの情報を更新する機能を第一に持ち、これにより、現用系キュー530と待機系キュー544は確実に同期される。キュー530、554の利用方法と相互作用につては以下で詳細に述べる。
待機系キューコンピュータ510の内部構造は、スニフ機能、スプーフ機能、引継機能を追加機能として持つが、現用系コンピュータ505と同様である。まず、待機系コンピュータ510はネットワークの上位クライアント層と接続されたNIC 550とネットワークの下位APサーバ層と接続されたNIC 552も有する。さらに、これらのネットワークカード550、552はプロミスキャスモードに設定でき、これにより、ネットワーク上に現れる全てのパケットがこれら二つのNICのうち一つにより受信される。各NIC 550、552は、それぞれのNICとコネクション情報バッファ564、566の双方に接続されたスニファーモジュール570、572とも接続されている。コネクション情報バッファ564、566では、上で説明したように、現用系を装って正しいパケットを送信するのに必要となる、現用系コンピュータ505に関する情報が絶えず更新されている。コネクション情報バッファは互いにあらゆる情報を転送しあう。この情報のうち、実際に静的なものは初期起動処理時に転送可能であり、システム操作により動的に更新されるものはコネクションバッファが継続的に互いに送信しあう。これにより、現用系・待機系コンピュータはパケット通信に関して同期される。
待機系コンピュータ510のクライアント側にあるスニファーモジュール570は、コネクション情報バッファ564中の情報を用い、NIC 550が受信したパケット全部から、現用系コンピュータに向けられたパケット・待機系コンピュータに向けられたパケット、（システムの別の構成要素に向けられた）無視しても良いパケットとを選別する。このスニファーモジュール570は外部512から受信した入力パケットと同様、現用系コンピュータが外部に向かって送信したパケットもスニフすることもある。現用系コンピュータに向かって送信されている、スニフされたインプットトランザクションは待機系コンピュータ510内のエンキュー／デキューモジュール562に送信され、モジュールはクライアント層から受信したこれらのインプットトランザクションを待機系キュー554に保存する。同様に、待機系コンピュータ510はエンキューされたインプットトランザクションの応答としてアウトプットトランザクションが現用系コンピュータからクライアント層に送信されたことを検知すると、待機系コンピュータは受信した元々のインプットトランザクションを待機系キュー554からデキューする。このスニフ処理により、現用系キュー530と待機系キュー554は常に同一の情報を保持することができるが、キュー情報バッファ546、568はキューの一貫性を保証するために断続的なキュー同期処理中は利用されることもある。
同様の方法で、第二のNIC 552は待機系コンピュータ510上の第二のコネクション情報バッファ566に接続された第二のスニファーモジュール572を持つ。このNIC 552もまた、プロミスキャスモードに設定することで、下位APサーバ層側の全パケットがNICで受信される。それから、スニファーモジュール572は接続されたコネクション情報バッファ566を利用し、アウトプットトランザクションとしてAPサーバ層から現用系コンピュータに送信されたパケットを認識できる。これらのAPサーバ層のアウトプットトランザクションは上位クライアント層からのインプットトランザクションに応答し、満たしたものであるため、上

位クライアント層からの対応するインプットトランザクションはキューから取り除くことが出来る。それ故、この第二のスニファーモジュール572もまた待機系キュー554に接続されたエンキュー／デキュー用モジュール562に接続されている。エンキューされているインプットトランザクションはアウトプットトランザクションにより満たされるため、これらのインプットトランザクションは現用系コンピュータ505上の現用系キュー530からデキューされる。同様にして、これらのAPサーバ層からアウトプットトランザクションは待機系コンピュータによりスニフされるため、スニファーモジュール572によりエンキュー／デキュー用モジュール562は待機系キュー554から対応するインプットトランザクションを取り除く。さらに、現用系と待機系キューの同期はキュー情報バッファ568、544によって保証される。

加えて、現用系コンピュータが（クライアントの代わりに）APサーバに送信したインプットトランザクションもまた、待機系コンピュータ上510のAPサーバ側のスニファー572によりスニフされる。現用系コンピュータから送られたこれらのインプットトランザクションをスニフすることにより、待機系コンピュータでは待機系コンピュータのクライアント側にあるスニファー570によりスニフされたエンキュー済インプットトランザクションが現用系コンピュータによりAPサーバ層に結果として送信されたことを確認することができる。この確認は（クライアントからの）リクエストトランザクションが現用系コンピュータに受信されたものの、APサーバ層にリクエストを送信する前に現用系コンピュータに障害が発生してしまうような状況に対応するために重要な処理である。これに関して以下でさらに述べる。

まとめると、待機系キュー554は上述の現用系キュー530と同様の働きをする。代表的な処理は次の通りである；(1)クライアントから現用系コンピュータに送られたインプットトランザクションはクライアント側のスニファー570によりスニフされ、待機系キュー554中のインプットキュー556にエンキューされる；(2)現用系コンピュータは受信したインプットトランザクションをAP層へ転送し、待機系コンピュータのAP側のスニファー572がこのトランザクションをスニフし、待機系キュー554中の対応するインプットフラグ560をセットする；(3)APサーバは受信したインプットトランザクションに対して応答し現用系コンピュータにアウトプットトランザクションを送信する。このアウトプットトランザクションはAP側の待機系スニファー572によりスニフされ、待機系キューのアウトプットキュー558にエンキューされる；(4)現用系コンピュータはこの受信したアウトプットトランザクションを転送し、このトランザクションはクライアント側のスニファーモジュール570でスニフされ、待機系キューのインプットキュー、アウトプットキューおよびインプットフラグキュー中のインプット／アウトプットトランザクションに関連した情報がデキュー（消去）される。

上で簡潔に述べたとおり、待機系コンピュータ510の監視モジュール580は現用系コンピュータが正常動作している（障害がない）ことを示す“OK”シグナルを一定間隔で現用系コンピュータの監視モジュール546から受信する。現用系コンピュータ505においてオペレーティングシステムやアプリケーション障害が発生すると、待機系コンピュータ上の監視モジュール580は現用系コンピュータにハードウェアリセットコマンドを送信し、スニファーモジュール570、576とスプーフィングモジュール572、576に対して、待機系キュー554から送信する必要のあるパケットをスプーフし始めるように通知することで“仮現用系”モードへと移行する。

キュー同期機能により、（現在）仮現用系コンピュータ510上の待機系キュー554は現用系キュー530と同一になっている。パケットスニフィングモジュール570。572は上述と同様の方法で、送信されなければならないキュー済みのリクエストをスプーフィングモジュール574、576に受け渡し、そのパケットはあたかも現用系コンピュータのアドレスから送信されたかようにして送出される。特に、仮現用コンピュータ上のNICのIPアドレスとMACアドレスは現用系コンピュータのIPアドレスとMACアドレスを受け継ぐ。この情報は各NICに接続されたコネクション情報バッファ546、566からパケットスプーフィングモジュール574、576に渡される。

さらにとりわけ、インプットトランザクションは上位クライアント層からスニフかつ受信され、エンキューされると、これらのトランザクションもまたパケットスプーフィングモジュール576に転送され、APサーバ層に送出される。送信先のAPサーバがこのパケットを受け取ると、現用系コンピュータから来たかのように見え、APサーバは現用系コンピュータに向けて返答のアウトプットトランザクションを返す。アウトプットトランザクションはシステム上のAPサーバ側にあるNIC 552によって受信されると、スニファーモジュール572がパケットを受け、待機系キュー554から対応するインプットトランザクションをデキューし、さらにシステム上のクライアント側にあるパケットスプーフィングモジュール574によって、クライアント層へとスプーフされたパケットが送出される。そして、その送信先であるクライアントコンピュータからはアウトプットトランザクションが現用系コンピュータから来たかのように見え、それ故、クライアントコンピュータがさらに応答することがある。

上述のように、引継先（待機系）コンピュータは現用系コンピュータキュー530の内部に保持されていたインプットトランザクションに対するトランザクション管理処理を“続行する”ことができ、同時に現用系コンピュータがリブート中に到着した新たなインプット／アウトプットトランザクションを受け取ることもでき、全てがクライアントコンピュータ（さらにはAPサーバ）に対して完全に透過的である。クライアントに関する限りでは、現用系コンピュータは何の障害も起こしてないように見える。

現用系コンピュータはリブートに成功し、再び各NICのMAC/IPアドレスを制御可能になると、現用系コンピュータは“OK”信号（または他のトリガ）を仮現用系（待機系）コンピュータ510の監視モジュール580に対して送信する。その後、仮現用系コンピュータの監視モジュール580はスニファーモジュール570、572とパケットスプーフィングモジュール574、576に対して、現用系コンピュータが自身宛のトランザクションを処理することが可能になったためトランザクションをもうスプーフしなくて良い、ということを通知する。待機系キューは現用系コンピュータがリブートしている間にほとんど大部分が違うものとなっているため、キューの同期機能によって、仮現用系コンピュータキュー554、キュー情報バッファ568から、現用系コンピュータ505の同一モジュール530、544へとキュー情報の転送が行われる。それから、現用系コンピュータではリブート中に行われたパケット処理に関する最新情報が“更新され”、仮現用系コンピュータは再び、通常の待機系モードへと切り替わることができるようになる。現用系コンピュータのアプリケーションやオペレーティングシステム上でさらなる障害が発生した場合は、待機系コンピュータでは再び、障害検知と現用系コンピュータを一時的に引き継ぐことによって、全部の処理が再び全て開始されることになる。

現用系コンピュータの障害に対応して、待機系コンピュータが仮現用実行状態になる処理について上で述べた。本システムは待機系／仮現用系コンピュータが現用系コンピュータに対して従属である（制御を止めて、現用系コンピュータに戻すことが常にできる）という点において特徴的である。本発明の追加の典型例として、二つのシステムキューコンピュータが、それぞれが現用系／待機系モード双方になることのできるという点でより同等である。

本例では、待機系コンピュータはネットワークノード上の全パケットをスニフし続けており、インプット／アウトプットトランザクションのどちらが現用系コンピュータに送信され、待機系コンピュータキューに保存すべきなのかを判断する。待機系コンピュータの監視モジュールが（現用系コンピュータから待機系コンピュータへの“OK”信号送信が中断されることで）現用系コンピュータで障害が発生したことを検知すると、待機系コンピュータは現用系コンピュータにIP/MACアドレスを開放するように要求し、現用系コンピュータをリブートするようにリセットコマンドを送信する。その後、待機系コンピュータは仮現用系モード（上述）へと移行し、コネクション情報バッファに保存されたNIC情報に基づいてパケットをスプーフする。この処理は上述の通り確実に行われる。

しかしながら、現用系コンピュータ上の障害が検知されるとすぐに、待機系（今は仮現用系）コンピュータは現用系コンピュータのリブートによって解放されている現用系コンピ

ュータのIPアドレスとMACアドレスの取得要求をネットワーク上のホストに送出する。しばらくして、仮現用系コンピュータはs枚の現用系NICのIPアドレスとMACアドレスを2枚の仮現用系NIC上で利用するために、その使用権を獲得する。この時、（以前の）現用系コンピュータに向けられたパケットは、前現用系コンピュータのアドレス情報を取得した仮現用系コンピュータによって今や受信される。それゆえ、仮現用系コンピュータは完全に現用系コードへと移行することができ、パケットのスプーフを行ったり、ネットワーク通信全てをスプーフしたりする必要はない。その代わりに（以前の）待機系コンピュータが（以前の）現用系コンピュータの実行処理を今や完全に引き継いでいる。 この完全な引継処理は、上位クライアント層と下位APサーバ層の双方に対して透過的である。

上述の待機系から現用系への遷移に付け加えると、（以前の）現用系コンピュータ505はリブートされている。リブートによって、このコンピュータ505では各NICに対して新しいIP/MACアドレスが割り当てられる。

新しい現用系キューコンピュータ510に将来的に引継処理が生じた場合を補助するため、前現用系キューコンピュータ505は待機系モードへと移行し、元々の待機系コンピュータ510が以前に担っていた役割を引き受ける。

上述の待機系から現用系への遷移に付け加えると、（以前の）現用系コンピュータ505はリブートされている。リブートによって、このコンピュータ505では各NICに対して新しいIP/MACアドレスが割り当てられる。新しい現用系キューコンピュータ510に将来的に引継処理が生じた場合を補助するため、前現用系キューコンピュータ505は待機系モードへと移行し、元々の待機系コンピュータ510が以前に担っていた役割を引き受ける。

新しい現用系コンピュータ510で障害が発生すると、上述の処理で再び現用系と待機系コンピュータとが交替する。このように、二つのキューコンピュータは、障害が発生すると各キューコンピュータが他方のコンピュータの実行処理を引き継ぐという点でより同等である。

本発明のさらなる実施例として、キューコンピュータシステムの一部として第三またはそれ以上のコンピュータがある場合を挙げられる。 一般に、内部のトランザクションキューが現用系キューとの一貫性を保たれる限りにおいては、任意の数の待機系コンピュータが適用可能である。これにより、その時の現用系コンピュータが障害状態に陥った時には、全てのキューコンピュータが互いに引き継ぎあうことができる。または、計画の容易さの点でいえば、各キューコンピュータは障害になるまで使用され、障害が生じたコンピュータがリブートされないこともある。これにより、ほぼ無限の組み合わせが可能である。

さらに、待機系コンピュータは第三のコンピュータによる現用系コンピュータの完全な引継を実現することもできる。例えば、現用系コンピュータが障害状態になると、待機系コンピュータは仮現用系モードに移行し、現用系コンピュータを装ってパケットをスプーフする。この時、現用系コンピュータが使用していたIP/MACアドレスを第三のコンピュータが取得してもよい。これらのIP/MACアドレスの制御を引き取ることで、この第三のコンピュータは現用系状態に移行し、（例えば、第三のコンピュータが仮現用系コンピュータにトリガをおくることで）仮現用系コンピュータは再び待機系状態へと戻る。この第三のコンピュータは通常動作の間は現用系キューと一貫性が保持された内部キューを持つため、現用系コンピュータの障害に対応して容易に引継を行うことが出来る。一方。変形例として、待機系コンピュータが（現用系コンピュータが障害状態にあることを意味する）仮現用系モードにある時のみ待機系コンピュータが第三のコンピュータにキュー情報を送ってもよい。そうすれば、第三のコンピュータが現用系状態へ移行したときに、第三のコンピュータは新しい現用系キュー情報としてこのキュー情報を利用することができる。

図6は図5で示したキューコンピュータシステムをさらに低位にしたシステム概要を表している。図中のモジュールの方向性は図5中にあるシステムブロック図に対応している。本発明におけるこの実施例では、各現用系と待機系コンピュータ上に複数のNICを有する。つまり、その一つは上位クライアント層に接続され、もう一つは下位APサーバ層に接続されている。図6の上側が待機系コンピュータを表し、図6の下側が現用系コンピュータを表している。

図6の中央に、現用系と待機系コンピュータ双方の内部キューが示されている。上で述べたように、これらのキューは上位クライアント層と下位APサーバ層から到着するリクエストやパケットを、処理するために送出されるまで保存している。こうしたキューはフラグ情報ももっており、現用系コンピュータ障害中におけるリクエストの処理を補助している。様々なNICからこれらのキューのそれぞれに、送られ、もしくは逆にこれらのキューからNICに送出されるパケットは、順番にパケット列に入り（エンキューされ）、もしくはパケット列から取り除かれる（デキューされる）というパイプライン処理で扱われる。実線部分は現用系コンピュータ上の現用系キューへの、もしくは現用系キューからのトランザクションの代表的な流れを示しており、点線部分は待機系コンピュータ上の待機系キューへの、もしくは待機系キューからのトランザクションの代表的な流れを示している。ここでもまた二つの内部キュー状態が同一データを保持しているかどうかを決定するキュー一貫性チェックと同期化ブロックがある。もし、データが同一でない場合は、キューの一貫性チェック及び同期化ブロックによって、現用系コンピュータから待機系コンピュータへとキュー情報と保持されたパケットとを転送することで、その一貫性を保証することができる。

コネクション情報バッファは現用系と待機系コンピュータ双方に存在するデータを保存し伝達する構成要素として示されている。この構成要素は互いに通信しあうように接続されているため、現用系コンピュータのコネクションバッファ情報（例えば、MACアドレス、IPアドレス、シーケンス番号、暗号鍵）は待機系コンピュータと共有されてもよい。双方で共有されたこの情報は待機系コンピュータがスニフし、自身の内部キューに保存すべきパケットがどれであるかを決定し、それと同時に現用系コンピュータの障害時には待機系コンピュータがパケットをスプーフすることを可能にする。

図7は図5におけるキューコンピュータを簡略化したものである。図7では、APサーバの機能（例えば、トランザクションの処理）はキューコンピュータ自身に組み込まれている。モジュールの方向性と参照番号は図5と対応しており、相違点は今述べた点だけである。APサーバ714、718はキューコンピュータ710、705に含まれるため、ネットワーク上にAPサーバ層は存在しない。クライアントコンピュータ712からのインプットトランザクションは現用系コンピュータ705に送信され、現用系コンピュータ内のAPサーバ718で処理される。処理後、現用系コンピュータ705中のAPサーバ718は受信したインプットトランザクションに対する応答として対応するアウトプットトランザクションを返す。従って、現用系トランザクションキュー730と待機系トランザクションキュー754はここに示すように単純化されている。現用系コンピュータ705に対し、受信したインプットトランザクションは現用系キュー730のインプットキュー732にエンキューされ、処理のためAPサーバ732に送信される。

インプットトランザクションがAPサーバ718へ無事に送信されると、現用系キュー730でインプットフラグ736がセットされる。同様に、この到着したパケットは待機系コンピュータのスニファーモジュール770によってスニフされ、そのインプットトランザクションは待機系キュー754にエンキューされる。待機系コンピュータ710のインプットフラグ760は（スニフするべき現用系コンピュータとAPサーバ間の外部トランザクションは存在しないため）無視してもよいし、インプットトランザクションがAPサーバ718に渡されたときに現用系コンピュータ705と待機系コンピュータ710の間でのある種の通信をすることによってインプットフラグ760をセットしても良い。

現用系コンピュータ705内のAPサーバ718はクライアント712に対してアウトプットトランザクションを返すとき、そのトランザクションは現用系コンピュータのアウトプットキュー734に保持されてから、クライアントコンピュータへと送信される。送信が成功したら、そのトランザクションに対する現用系キュー730のインプット／アウトプット／フラグはデキューされる（消去される）。同様に、アウトプットトランザクションは待機系コンピュータ710のアウトプットキュー758に内部的にエンキューされても良いし、待機系コンピュータではこのステップを行わなくても良い。一方、待機系コンピュータのスニファーモジュール770がクライアントコンピュータ712に送信されているアウトプットトランザク

ションをスニフした時には、待機系コンピュータ705は待機系キュー754からこのトランザクションをデキューする。
現用系コンピュータ705の障害時には、仮引継と完全引継処理がすでに述べたのと同様の方法によって保証される。本実施例で行われる処理は図5に応じて簡略化されている。従って、その方法に関する情報についてはこれ以上ここではとりあげない。
本発明の様々な実施例をより一層明確に理解してもらうため、ここでは代表的なシステム動作方法を詳細に記した一連のフローチャートに従って議論を進めていく。フローチャートは現用系と待機系コンピュータの初期化からシステムの通常動作時を介し、仮引継、完全引継を経て、障害発生後の現用系のリブートにいたるまでの処理全体を網羅している。代表例についてのみ述べるが、このフローチャートは上述のシステム要素に触れている。
図8は現用系と待機系コンピュータ双方における代表的な起動処理のフローチャートである。起動処理中で一番重要な処理は現用系と待機系コンピュータのトランザクションキューの生成と同期である。現用系と待機系コンピュータ間で転送される情報量は起動処理を行うためにむしろできるだけ制限されることもある。これにより、全トランザクションデータは最初には転送されないこともある。この起動処理は初期のシステム電源投入時に行われるだけでなく、キューコンピュータの一つに障害が発生した時に行われるハードウェアないしはソフトウェアリブートの処理中にも行われる。
図8の処理は現用系と待機系コンピュータ双方でトランザクションキューの生成と初期化を始める。その後、現用系と待機系キューコンピュータで方法が分かれる。現用系コンピュータは、現用系コンピュータに障害が発生した時に待機系コンピュータが現用系コンピュータに対するネットワークパケットをスニフしたり、スプーフしたりするのに必要となる情報を待機系コンピュータに送る。特に、現用系コンピュータは自身のNICのIP/MACアドレス、シーケンス番号や暗号鍵を待機系コンピュータに送信する。待機系コンピュータはこの情報を受信し、（利用されるNICの数によって）一つないしはそれ以上のコネクション情報バッファへと保存する。
待機系コンピュータは次に、図9の右側に書かれているように、システムのAPサーバ側の全パケット（例えば、現用系コンピュータのNICのIPアドレスやMACアドレスに基づいたもの）を“スニフ”するために、初期送信処理を実行する。図8の起動処理を続けると、キュー同期処理のサブ処理が現用系キューコンピュータで開始され、現用系キュー情報を待機系キューコンピュータに送信する。現在の待機系キューはこの受け取った現用系キュー情報を互いに一致しているかを確認するために比較を行う。もし待機系キューが受信した現用系キューと一致してない場合、二つのキューを同期させることで待機系キューが修正される。
もし待機系キューが受信した現用系キューと一致している場合（または、待機系キューが変更された後に）、待機系キュー情報は現用系コンピュータに対し、キュー同期化の再チェックのため現用系に送信し返される。現用系コンピュータでは、二つのキューが互いに一致しているかを決定するためにその受信した待機系キュー情報を現用系キューに対し確認する。もし、受信した待機系キューと現用系キューとが一致しない場合は、現用系コンピュータキューは二つのキュー間で一貫性保証を行い修正される。このようにやりとりされる処理を図8では“キュー同期化”として記している。この処理はこの後の図でも繰り返し行われるが、明確化のため、キュー同期化としてだけ記してある。
二つのキューが一致している場合、または現用系キューが更新された場合、現用系コンピュータは現用系キュー内の情報に基づいてクライアント層とAPサーバ層に最初のトランザクションを送信する。例えば、上述したように、現用系キューは、クライアントから受信したインプットトランザクションを持つインプットキュー、受信したインプットトランザクションが適当なAPサーバへと転送されたことを示すインプットフラグ、現用系キューのインプットトランザクションに対応したアウトプット（リプライ）トランザクションが記されるアウトプットキューとを持つ。
上記キュー動作指針により、この初期化処理では、現用系キューは現用系コンピュータキュー内のアウトプットトランザクションがあるかどうかの確認から開始する。存在する場

合、そのトランザクションを適当なクライアントコンピュータ（外部）へ送信し、クライアントコンピュータからの配送承認を受信したときにそのトランザクションに対するインプット／アウトプット／フラグの情報はデキューされる（消去される）。次に、現用系キュー内の対応するインプットフラグがセットされていないインプットトランザクションを適当なAP側コンピュータに送信する。AP側からの受信承認後、今送信したインプットトランザクションに対応するフラグが現用系コンピュータのインプットキューにつけられる。多分、待機系コンピュータのAP側スニファーモジュールはこうしたトランザクションをスニフし、同様のデキュー操作やフラグセット操作を行っている。

その後、現用系コンピュータはパケット（インプットトランザクション）送信手段とパケット（アウトプットトランザクション）受信手段を、それぞれ図9と図10で示されているように実行していく。同様に、待機系コンピュータは図10で概略が示されている手法によってパケットを受信（スニフ）する。

図9は現用系コンピュータと待機系コンピュータが通常（待機系）動作時に行う基本的な送信処理を表している。送信操作は、 AP側からクライアント側へアウトプットトランザクションをそれ以前に受信したインプットトランザクションへの応答として送信することを意味している。現用系と待機系コンピュータの図9の手段は、現用系ではトランザクションの受信・送信・デキュー（図9の左側）、待機系ではこうしたトランザクションのスニフ・同様の手段での待機系キューの更新（図9の右側）というように、互いに関連して動作する。

現用系コンピュータ側では、APサーバ層からクライアント層へと送信されたアウトプットトランザクションが現用系コンピュータにより受信され、一時的に以前に受信した対応するインプットトランザクションと対応する場所の現用系キューのアウトプットキューに保持される。同時に、待機系コンピュータはAPサーバから現用系コンピュータへ（クライアントコンピュータへ送信するために）送信される全アウトプットトランザクションと、現用系キューコンピュータから適当なAPコンピュータへと送信される全インプットトランザクションとをスニフし始める。上述の受信されたアウトプットトランザクションは待機系コンピュータのAP側スニファーモジュールでスニフされ、このアウトプットトランザクションは待機系キューの適当なアウトプットキューに保存される。

その後。現用系と待機系コンピュータは、現用系キュー情報を待機系コンピュータへと送信し、待機系キュー情報と比較するキュー同期化処理（上述）を実行する。待機系キューが受信した現用系キュー情報と一致しない場合は、待機系キュー情報を更新する。それから、待機系キュー情報は現用系コンピュータへと送信され、現用系キュー情報は待機系キューと同期化される。

同期化の後、現用系コンピュータは適当なクライアントコンピュータに（現用系キューのアウトプットキューに保存された）アウトプットトランザクションを送信する。受信承認後、このトランザクションに関するインプット／アウトプット／フラグ情報の組はデキューされる。同様に、待機系コンピュータのクライアント側スニファーモジュールは現用系キューコンピュータからクライアントコンピュータへ送信されたアウトプットトランザクションをスニフし、待機系キューの同トランザクションをデキューする。キュー同期化が再度行われる（最初は待機系コンピュータで、次に現用系コンピュータで）。

この時、送信操作は完結し、待機系コンピュータは監視ブロックを使って、現用系コンピュータが“生存”または障害（例えば、待機系コンピュータへの“OK”信号の送信が中断される）状態に未だあるかどうかを決定する。もし現用系コンピュータが依然として動作中であれば、待機系コンピュータの送信処理は図9の先頭へとループし、待機系コンピュータは現用系コンピュータによって送信されるパケットをスニフし続ける。どんなキュー同期化処理が行われていても、このスニフ処理は実行される。もし待機系コンピュータが現用系コンピュータの障害を検知した時には、待機系コンピュータは、図11を用いて以降で述べるように仮現用系モードへと移行する。

図10はシステムが通常（待機系）動作時に現用系コンピュータによって行われる基本的受信処理（図10の左側）と待機系コンピュータによって行われる基本的受信処理（図10の右

側）とを示している。受信処理はクライアントからAPサーバへのインプットトランザクションの受け渡しを表している。まず、クライアント層（外部）から現用系コンピュータに正しく向けられたインプットトランザクションを現用系コンピュータが受け取り、現用系キューのインプットキュー部にエンキューする。同時に、待機系コンピュータのクライアント側スニファーモジュールはこのインプットトランザクションをスニフし、待機系キューにそのトランザクションを保存する（エンキューする）。上述のように、このスニフ処理は待機系コンピュータのクライアント側NICがプロミスキャスモードにされ、クライアント側のネットワークパケット全部を受信するようになることで開始される。これらのパケットはコネクション情報バッファにある現用系コンピュータ情報（例えば、現用系NICのMAC/IPアドレス）を用いて、パケットスニファーモジュールによりフィルタリングされる。

キュー同期化は（上述のように）現用系キュー情報が待機系キューに送信され、キューの一貫性をチェックし、一致しない場合は待機系キューを修正することによって、実行される。それから、同期化を完了するために、全く逆の処理で待機系キュー情報は現用系キューに送られることもある。高負荷時のスニフ処理は待機系でのパケットの受信ミスを起こすことがありうるため、できるなら現用系キュー情報が優先される。

同期化の後、現用系コンピュータは受信したインプットトランザクションを処理してもらうためAPサーバ層（または付属のAPサーバ機能部）へと送出し、（受信承認の後）送信したインプットトランザクションに対応する現用系のインプットフラグをセットする。このインプットフラグは受信したインプットトランザクションがAP層に問題なく送信されたことを表している。この処理と同時に、待機系コンピュータのAP側スニファーモジュールはAP層に送出されたインプットトランザクションをスニフし、待機系キューの対応するフラグをセットする。これにより、現用系と待機系キューは同一の情報を保持することができる。

その後、現用系と待機系のキューの同一性を保証するため、二番目のキュー同期処理が行われる。この同期化はまず待機系コンピュータから行われ、その後、待機系コンピュータで行われ、これによって現用系キューにあるデータに優先度が与えられる。

同期化の後、待機系コンピュータは監視ブロックによって現用系コンピュータが障害状態にあるかを検知する。そして、もし障害状態ならば、図12を用いて以下に記すパケットスプーフィング処理を開始する。もし、現用系コンピュータが正常に動作中であるならば、図10の待機系コンピュータ処理はループし、次のネットワークパケットのスニフを行う。

図11は（図9から遷移して）待機系コンピュータが待機系状態から仮現用系状態に移行し、スニフ処理によってパケットを受信し続ける処理を示している。この処理は仮現用系モードにおける送信処理と考えられる。（上述の監視機能を利用して）現用系コンピュータで障害が発生したことを検知すると、待機系コンピュータは現用系コンピュータに強制リセット信号を送ることで、仮現用系コンピュータとなる。この強制リセット信号は現用系コンピュータをリブート処理に移行させるハードウェアまたはシステムボードリセット信号であり、これにより少なくとも一時的にIP/MACアドレスを開放させる。従って、待機系コンピュータは現用系コンピュータに向けられているパケットをスニフし、これらのパケットへの応答をスプーフする必要がある。

パケットスニフ処理を継続している一方で、（現状、仮現用モードにある）待機系コンピュータは（現状、障害状態にある）現用系コンピュータに代わってパケットをスプーフするのに必要となる情報を受け取る。この情報には現用系コンピュータのIP/MACアドレス、あらゆる暗号鍵、シーケンス番号、その他の適当な情報が含まれる。こうした情報の多くは最初の起動処理時に受け取っても良いし、残りの必要な情報は（現用系コンピュータとの）共有ディスクやその他の装置を介して受け取っても良い。

その後、どのパケットを送出する必要があるかを決定するために待機系キューのチェックを行う。すでに述べたように、インプットトランザクションのうち対応したインプットフラグがセットされていないトランザクションは全て、クライアントから受信したがAP側への転送が無事行われていないインプットトランザクションを意味している。従って、キュ

ーされたインプットトランザクションが待機系キュー内にインプットフラグを持たない場合は、これらのトランザクションはパケットの送信元として現用系コンピュータのアドレス情報を利用したスプーフィング処理を介して、APへ送信される必要がある。送信された後、待機系キューの適当なインプットフラグがセットされる。
キュー同期化の後、待機系コンピュータはクライアントへのアウトプットトランザクションをスプーフするループ処理に移行する。まず、AP側スニファーモジュールが現用系コンピュータに送られてきたアウトプットパケットを全てスニフする。これらの受信したアウトプットトランザクションはそのアウトプットトランザクションに対応する以前に受信したインプットトランザクションに対応した場所の待機系キューのアウトプットキュー部にエンキューされる。そして、現用系コンピュータのアドレス情報を用いてスプーフされたパケットが生成され、適当なクライアントコンピュータに対してそのスプーフされたパケットが送信される。クライアントからの配送承認を受信し、今送信したインプット／アウトプットトランザクションに対応するインプット／アウトプット／フラグのデータが待機系キューからデキューされる。
それから、待機系コンピュータは完全なIP引継をすべきかを判断し、もしそうであるなら、図13を用いて以下に説明するIP引継処理に移行する。もし、そうでない場合は、仮現用系送信手続きがループされ、AP側のスニファーモジュールにより次のアウトプットトランザクションがスニフされるまで待つ。
図12は（図10から）仮現用系状態へ移行した後に待機系コンピュータで行われる仮現用系受信処理を示している。この仮現用系受信処理により、（現状、仮現用系モードにある）待機系コンピュータは現用系コンピュータを装ってインプットパケットをスプーフすることが出来るようになる。現用系コンピュータに強制リセット信号の送信や、（現状、リブート中の）現用系コンピュータを装ってパケットをスプーフするのに必要なアドレス情報を受信が図11の送信処理でまだ行われていない場合は、これらの処理を行うことで、そのプロセスが待機系コンピュータで開始される。
それから、待機系コンピュータは待機系キューのアウトプットキュー部にトランザクションがあるかどうか待機系キューをチェックする。もしあるならば、これらのキューされたトランザクションはAP側から受信したが、適当なクライアントコンピュータへ送信がされていないアウトプットトランザクションを意味する。従って、待機系コンピュータはこれらのトランザクションをクライアント側に送信し、クライアント側からの承認を受信した後、この送信されたアウトプットトランザクションに対応するインプット／アウトプット／インプットフラグのデータを待機系キューからデキューする。
その後、待機系コンピュータはクライアント側から受信したインプットトランザクションをエンキューし、適当なAPコンピュータへと転送するループ処理へと移行する。まず、クライアント側スニファーモジュールは（現用系コンピュータに向けられた）クライアントコンピュータから到着したインプットトランザクションをスニフする。そして、待機系コンピュータは、待機系コンピュータにすでに受信された現用系キュー情報からこのインプットトランザクションに対し（現用系コンピュータによって送信されたように見える）スプーフされたパケットを生成する。それから、待機系コンピュータはこのスプーフされたパケットをネットワークのAP側に送信し、インプットトランザクションがAP側へ無事送信されたことを反映して、待機系キューの適当なインプットフラグをセットする。
このパケットスプーフィング処理の後、待機系コンピュータはIP引継を行うべきかどうかを決定する（図13）。もしIP引継処理を行う必要がないならば、処理はクライアント側からの次のインプットトランザクションを受信するためにループする。そして、次のスプーフされたパケットが準備される。図11、図12の仮現用系処理は待機系コンピュータが仮現用系モードにある限り（例えば、現用系コンピュータがリブートし、自身のIP/MACアドレスの制御を再び取り戻し、トランザクションの転送を開始するまで）は継続される。
上記の仮現用系送信処理（図11）や受信処理（図12）の間に、もし現用系コンピュータがリブートに成功したならば、待機系コンピュータは待機系コンピュータ情報を現用系コンピュータに送信し、それによって現用系コンピュータは現状のキュー情報をもつことにな

る。それから、現用系コンピュータはインプットとアウトプットリクエストを通常通り処理し始め、現用系と待機系コンピュータは図9、図10の通常動作のフローチャートへと移行する。

上記の仮現用系送信処理（図11）や受信処理（図12）の間に、もし待機系コンピュータが現用系コンピュータを完全に引き継ぐことが出来ることを検知したならば、図13の処理が適用される。図13にあるように、（現状、仮現用系モードにある）待機系コンピュータは現用系の制御を自分自身ないしは別のコンピュータのどちらかに移すことが出来る。ネットワークAP側とクライアント側でのスニフ処理を介して、現用系コンピュータに向けられたパケットを受信し続けて、引継処理は開始される。一方で、待機系は新しい現用系コンピュータそのものになるのか、あるいは待機系コンピュータが別のコンピュータを現用系コンピュータにすることができるのかどうかを検知する。

待機系コンピュータが現用系コンピュータそのものになることを検知した場合（図13左側）には、待機系コンピュータはスニフ操作は続けるが、新たに受信したインプットトランザクションをAP側に送信することを中断する。従って、トランザクションはキューに入れられることになるが、待機系コンピュータが現用系コンピュータとなるまで、それらのトランザクションは処理を実行するために送出されることはない。さらに、（仮現用系モードにある）待機系コンピュータは（現状、障害中の）現用系コンピュータに代わってパケットをスプーフすることを中断する。代わりに待機系コンピュータは現用系コンピュータとして起動処理を行い、（現状、現用系キューとなっている）自身のキューを新たに待機系となる別のコンピュータにあるキューと同期化させる（図8参照）。この新しい待機系コンピュータは以前の現用系コンピュータであってもよいし、上記のように、追加されるコンピュータであってもよい。 図8による初期化の後、以前の待機系コンピュータは現状では現用系コンピュータとなる。

待機系コンピュータが追加のコンピュータが新たな現用系コンピュータとなることを検知した場合（図13右側）には、待機系コンピュータはパケットのスニフを続行するが、システムのAP側への新たなインプットトランザクションの送信を中断する（これにより、システムのAP側から新たなアウトプットトランザクションを受信することもなくなる）。その後、待機系コンピュータは待機系コンピュータとして起動処理を実行し、これにより、そのキューは現在現用系コンピュータとなっている追加のコンピュータ上の現用系キューと同期化される。この追加のコンピュータもまたこの処理全体の間待機系モードであり、これにより追加の（新たに現用系となる）コンピュータのキューは待機系コンピュータのキューと同一の情報を保持している。または、仮現用系モードにある待機系コンピュータが現用系コンピュータとなるべきコンピュータにキュー情報を送ることもある。このようにして、本発明では、待機系コンピュータを任意の数だけ適用でき、現用系コンピュータのうち個々が障害状態になった時に、その一つの現用系コンピュータから別のコンピュータへ継続的にかつ透過的な引継を実現することができる。

上述のように、本発明を適用することで現用系コンピュータの透過的引継を実現可能となる例として、多くの変形例が存在する。こうした例の一つとして、選択されているAPサーバに障害が発生した場合には、本発明における一般的な引継処理を適用し、局所的にAPサーバの引継を行うことができる。例えば、図5では元々のAPサーバ514とAP層のネットワークノード518の双方に接続された第二の“待機系”APサーバ582が示されている。このAPサーバ582は第一のAPサーバ514の実行状態を監視し、第一のAPサーバの機能を追従することもある。

待機系APサーバ582は第一のAPサーバ514の障害を検知すると、障害発生時に第一のAPサーバ上にあった現状のトランザクションを完了させるために必要となる全ての情報を持っていても良い。コネクションバッファ情報によって、待機系APサーバ582は第一のAPサーバ514を装って第一のAPサーバがリブート中はパケットをスプーフすることができる。従って、この局所的な引継を可能にするために、待機系APサーバ582はトランザクションのスニフ機能とスプーフ機能を持つ。この引継は上述のキューの引継がクライアント層に対して透過であるのと同様に、キューコンピュータ505、510に対して透過的である。第一のAPサ

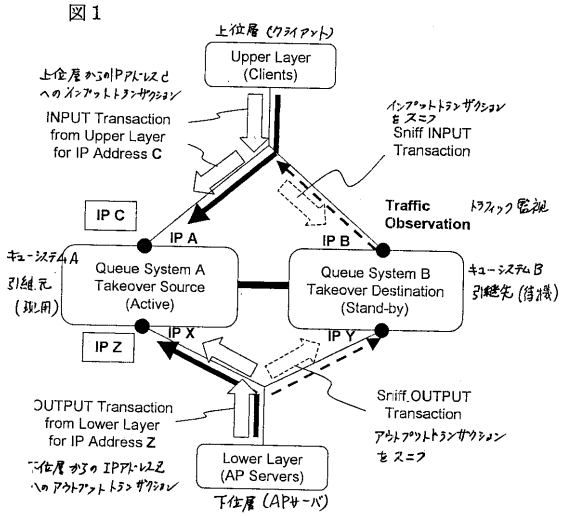
ーバ514のリブート後は、第一のAPサーバが再び現用系APサーバになるか、予備となる。これらの機能全般は上述の事項に平行して動作する。
また、待機系APサーバ582はキューコンピュータ505、510に対してGratuitous ARPを送信し、（APサーバが障害により切り替わったため）APサーバのアドレスが変更されたことを知らせることもある。待機系APサーバ582は障害検知に応じて、第一のAPサーバ514をリブートさせることもある。全体として、多様な違った手法を適用することができるかもしれず、また、こうした手法のうちのそれぞれが組み合わせて使用される場合や、上述の現用／待機系キューコンピュータの方向性とは完全に独立に適用されることもありうる。
図7は、この局所的なAPサーバの引継方法を一つのNICを用いて実現する場合を示している。さらに、待機系APサーバ782は存在しているAPサーバ714とキューコンピュータ710のトランザクションキュー（またはエンキュー／デキューモジュール762）に接続されている。この実施例では、本キューコンピュータシステムに適用するために多少簡単化された方法によって、上述のAPサーバ引継手法のうちどれか一部を適用することができるかもしれない。この局所的なAP引継はまたキューコンピュータ引継手法の一部ないし組み合わせが適用されることもある。
以上に、特定アプリケーションの特定実施態様に関して発明が記述されたが、当該技術者であれば、ここでの説明に照らして、発明の精神を逸脱したり発明の範囲を越えないで、実施態様に追加や変更を加えることが可能である。加えて、ここでの図や説明は単に本発明の理解を促す説明のためだけに提供されており、本発明の範囲を制限するものではない。

【図面の簡単な説明】
【図１】本発明による、現用／待機キューシステムモデルの高位のシステムブロック図である。
【図２】引継元コンピュータで障害が発生した場合の現用／待機キューシステムモデルの高位のシステムブロック図である。
【図３】現用／待機キューコンピュータを用いた従来の障害引継システムの高位ブロック図である。
【図４】従来の障害引継システムと本発明とを比較したタイミング図である。
【図５】本発明によるキューコンピュータシステムの低位のブロック図である。
【図６】APサーバとキューコンピュータが1つになったキューコンピュータシステムの簡易版である。
【図７】本発明によるキューコンピュータシステムの低位のブロック図である。
【図８】現用／待機系コンピュータの初期化プロセスのフローチャートである。
【図９】現用／待機系コンピュータが通常システム動作時のSEND手続きを表すフローチャートである。
【図１０】現用／待機系コンピュータが通常システム動作時のRECEIVE手続きを表すフローチャートである。
【図１１】待機（仮現用）系コンピュータが仮引継処理時のRECEIVE手続きを表すフローチャートである。
【図１２】待機（仮現用）系コンピュータが仮引継処理時のSEND手続きを表すフローチャートである。
【図１３】待機系コンピュータと新たなコンピュータとの引継手続きを表すフローチャートである。
【符号の説明】
505・・引継元(現用系)コンピュータ
510・・引継先(待機系)コンピュータ
512・・上位クライアントネットワーク層（外部）
514・・下位APサーバ層
516、518・・ハブ
520、522、550、552・・NIC

524、562・・エンキュー／デキューモジュール
530、562・・キュー
540、542、564、566・・コネクション情報バッファ
544、580・・監視モジュール
570、572・・スニフモジュール
574、576・・スプーフモジュール。

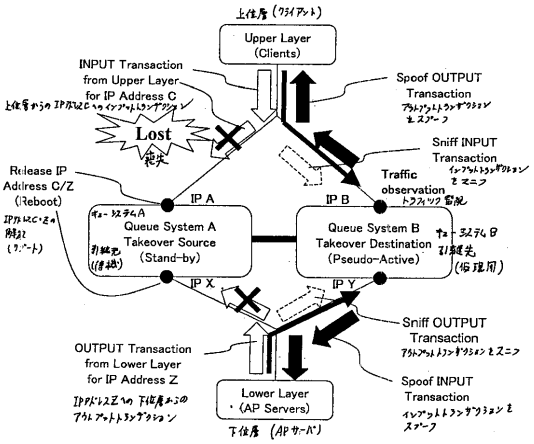
【図１】

図1



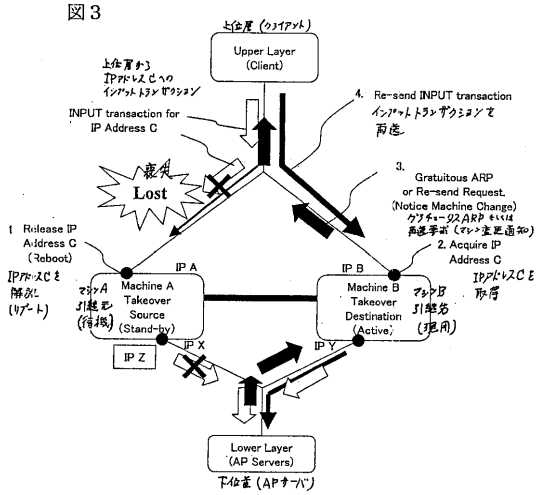
【図２】

図2



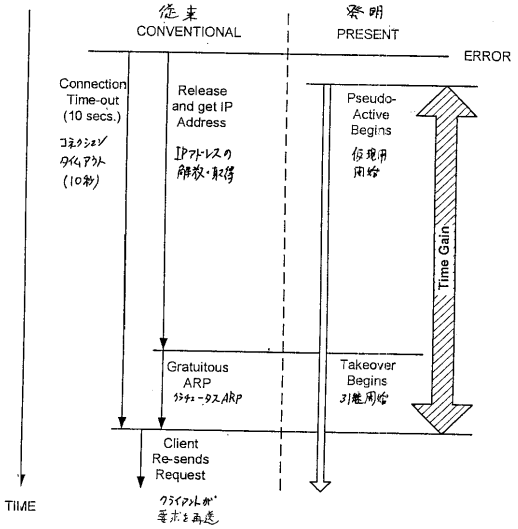
【図３】

図3



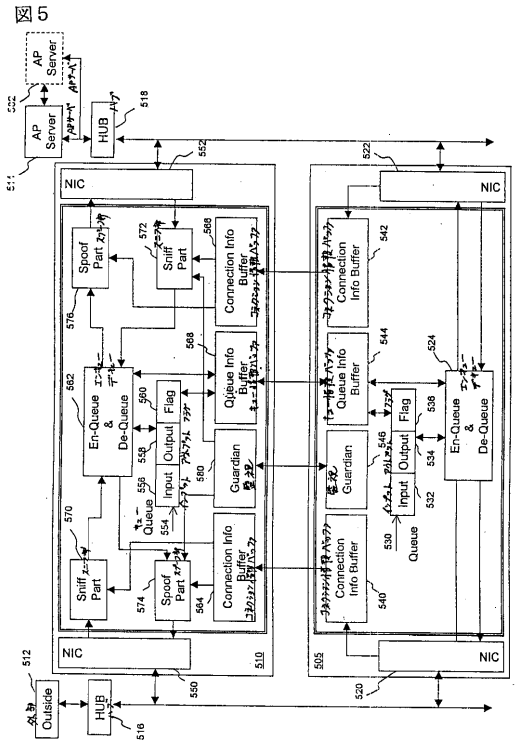
【図４】

図4



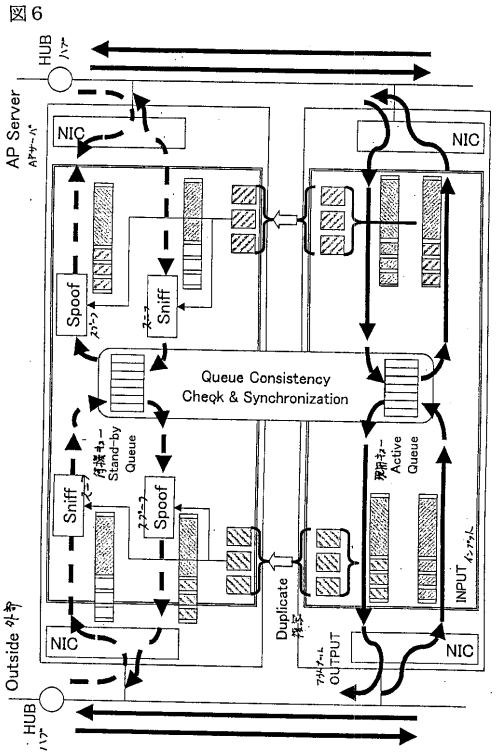
【図５】

図5



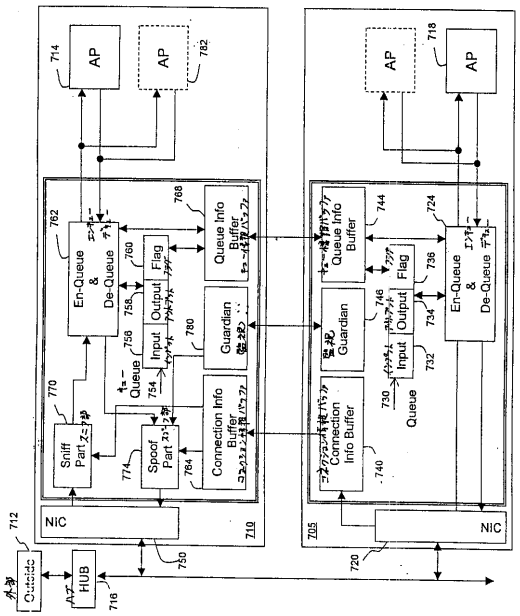
【図６】

図6



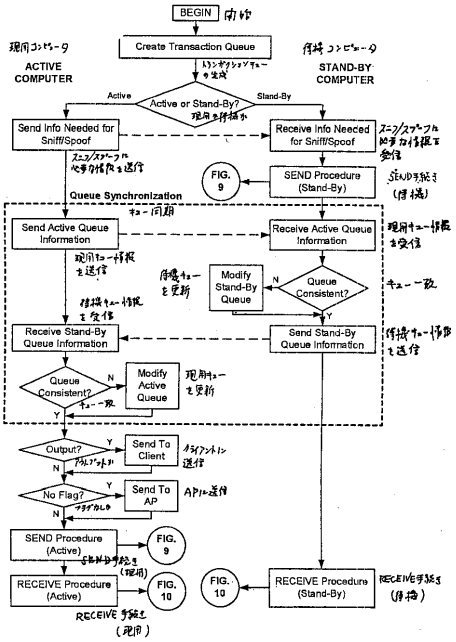
【図７】

図7



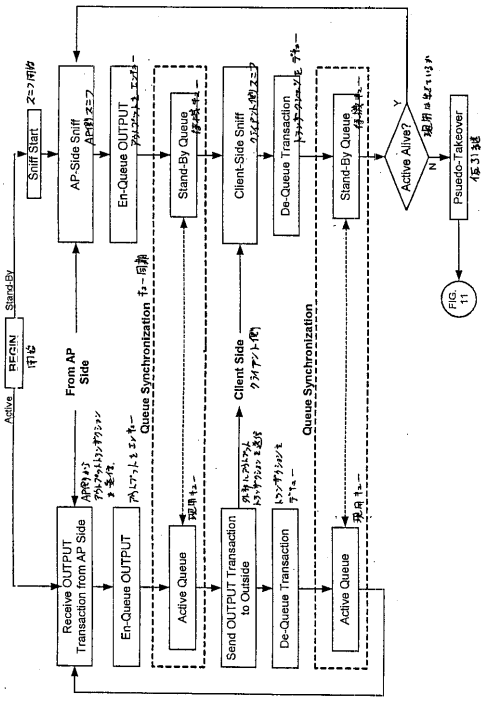
【図８】

図8



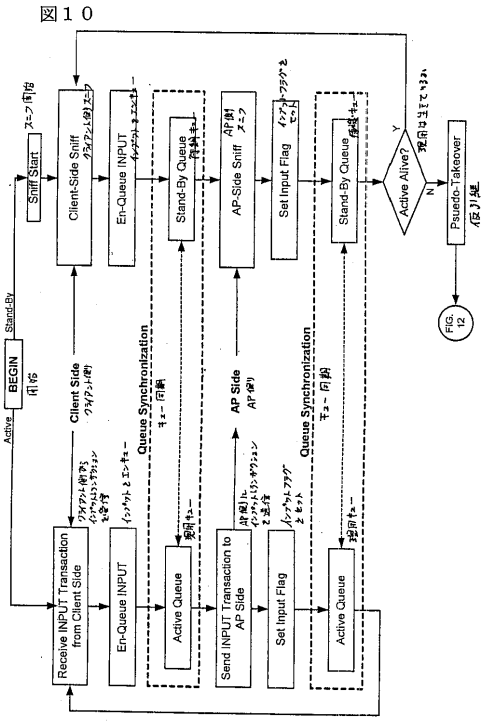
【図９】

図9



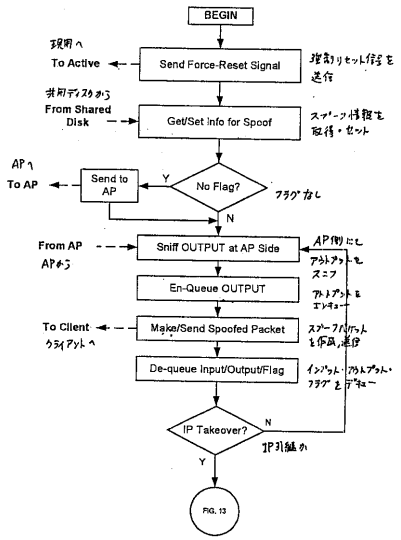
【図１０】

図10



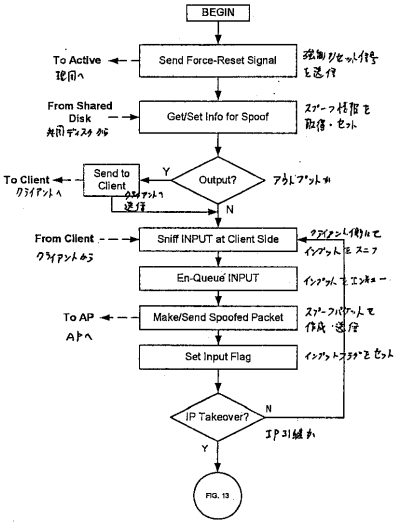
【図１１】

図１１



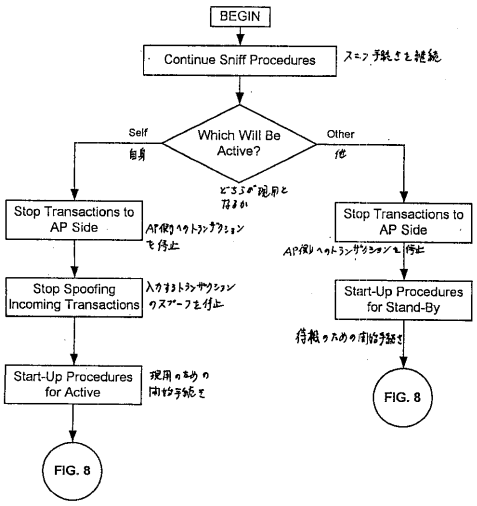
【図１２】

図１２



【図１３】

図１３



フロントページの続き

(56)参考文献 特開平０８－００６９１０（ＪＰ，Ａ）
特開２００１－０２２７１８（ＪＰ，Ａ）
特開２０００－０９２０７９（ＪＰ，Ａ）
特開平１０－１９０７１２（ＪＰ，Ａ）
特開２０００－０５６９９６（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.，ＤＢ名)
H04L 12/00-12/28
H04L 12/44-12/66
G06F 13/00
G06F 15/16-15/177