(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第3671057号
(P3671057)
(45)発行日　平成17年7月13日(2005.7.13)　(24)登録日　平成17年4月22日(2005.4.22)

(51)Int.Cl.[7]　F I
H04L 12/40　H04L 11/00　320

請求項の数 12　(全 14 頁)

(21)出願番号　特願平9-523666
(86)(22)出願日　平成8年12月4日(1996.12.4)
(65)公表番号　特表平11-501196
(43)公表日　平成11年1月26日(1999.1.26)
(86)国際出願番号　PCT/US1996/019330
(87)国際公開番号　WO1997/023976
(87)国際公開日　平成9年7月3日(1997.7.3)
審査請求日　平成15年12月4日(2003.12.4)
(31)優先権主張番号　08/577,575
(32)優先日　平成7年12月22日(1995.12.22)
(33)優先権主張国　米国(US)

(73)特許権者
ディジタル　イクイップメント　コーポレイション
アメリカ合衆国　マサチューセッツ州　01754　メイナード　パウダーミル　ロード　111
(74)代理人
弁理士　杉村　興作
(74)代理人
弁理士　冨田　典
(74)代理人
弁理士　杉村　純子
(74)代理人
弁理士　徳永　博

最終頁に続く

(54)【発明の名称】ネットワークアダプターにおけるパケットの自動再送信のための方法及び装置

(57)【特許請求の範囲】
【請求項１】
データを送信するために共用リソースに接続され且つ送信バッファを有するアダプターが動作し、パケット送信並びに選択的に過剰衝突状態及びバッファアンダーフローを含む好ましくない送信状態の場合におけるパケット送信の自動再試行を行う方法であって、以下のステップ、即ち
予め定められた第１の量のデータが前記アダプターの送信バッファに格納された後に送信を開始するステップ、及び
過剰衝突状態は前記パケットの選択された送信試行回数の各々が前記パケットに関連して衝突を生ずる時に生起するものとし、送信開始ステップの後に過剰衝突状態が生起したか否かを決定し、過剰衝突状態が生起した場合は過剰衝突カウントを更新し、及び
(i)前記過剰衝突カウントが過剰衝突限界以下の場合は送信を停止し、次に実質的に直ちに、前記パケットの送信を繰り返し、及び
(ii)前記過剰衝突カウントが過剰衝突限界を超えている場合は送信を停止し、前記送信バッファに格納された前記パケットのデータを廃棄するステップ
を含む方法。
【請求項２】
請求項１に記載の方法において、更に、以下のステップ、即ち
生起したいかなる衝突をも検出するステップ、
生起した前記衝突の各々を、衝突カウント限界までカウントするステップ、

前記衝突カウント限界に到達する度に、前記過剰衝突の数を表示するように過剰衝突表示カウントを増すステップ
を含む方法。
【請求項３】
請求項１に記載の方法において、前記共用リソースがコンピュータネットワークに接続され、更に、ネットワークのトラヒック状態に応じて、動的に過剰衝突限界を決定するステップを含む方法。
【請求項４】
請求項１に記載の方法において、更に、送信バッファアンダーフロー状態に続いてパケットの自動再送信を行う複数のステップを含み、このステップは、
前記パケットの送信中に送信バッファアンダーフロー状態が生起した場合、前記パケットの送信を停止し、前記パケットの前記第１の量のデータより多い第２の量のデータが前記送信バッファに格納された後に再び前記パケットの送信を開始するステップ
を含む方法。
【請求項５】
請求項４に記載の方法において、前記共用リソースがコンピュータネットワークに接続され、更に、ネットワークのトラヒック状態に応じて、動的に前記過剰衝突限界及び第２のデータ量を決定するステップを含む方法。
【請求項６】
過剰衝突状態に続くデータのパケットの自動再送信のための方法であり、ネットワークアダプターによって実現される方法であって、以下のステップ、即ち
a)前記パケットの予め定められたバイト数が送信できる状態になった後でパケットの送信を開始するステップ、
b)衝突が検出された場合は衝突カウントを増し、この衝突カウントが予め定められた第１限界を超えない場合は、送信を続けるステップ、
c)衝突カウントが前記第１限界を超えた場合は以下のステップ、即ち
(i)再試行（ＲＮＥＣ）カウントが第２限界を超えた場合は、前記パケットの送信を停止するステップ、及び
(ii)ＲＮＥＣカウントが前記第２限界を超えない場合は、このＲＮＥＣカウントを増し、この増加したＲＮＥＣカウントを用いて再びステップ(a)-(c)を遂行するステップ
を含む方法。
【請求項７】
請求項６に記載の方法において、前記パケットの前記送信開始ステップが、前記衝突カウントに対する前記第１限界と前記ＲＮＥＣカウントに対する前記第２限界との積に等しい回数まで遂行される方法。
【請求項８】
請求項６に記載の方法において、ステップ(b)が、衝突ウィンドウの間で衝突が生起したか否かを決定し、この衝突状態が前記衝突ウィンドウの間に検出された場合のみ、この衝突状態に基づいて前記パケットの送信を停止するステップを含む方法。
【請求項９】
データを送信するために共用リソースに接続され且つ送信バッファを有するアダプターが動作し、パケット送信及びバッファアンダーフローのような好ましくない送信状態の場合におけるパケット送信の自動再試行を行う方法であって、以下のステップ、即ち
a)予め定められた第１の量のデータが前記アダプターの送信バッファに格納された後に送信を開始するステップ、及び
b)送信開始ステップの後、バッファアンダーフロー状態が生起した場合は送信の停止を含むステップを遂行し、次に、第１の量のデータより多い予め定められた第２の量のデータが前記送信バッファに格納された後にのみ再び送信を開始するステップ、及び
c)ステップb)の後実質的に直ちに再び送信を開始し、第１状態は送信の完了を含み且つ第２状態はステップb)が限界に到達するまでに繰り返される回数を含むものとした場合、第

１及び第２状態の一つが生起するまでの回数だけステップb)及びこのステップc)を繰り返すステップ
を含む方法。
【請求項１０】
請求項９に記載の方法において、前記限界が予め定められた固定値からなる方法。
【請求項１１】
請求項９に記載の方法において、更に前記アダプターによって経験された従前のアンダーフロー状態に基づいて計算される限界を含む方法。
【請求項１２】
データを送信するために共用リソースに接続され且つ送信バッファを有するアダプターが動作し、パケット送信及びバッファアンダーフローの場合におけるパケット送信の自動再試行を行う方法であって、以下のステップ、即ち
a)送信を開始する前に前記送信バッファ中で必要とするバイト数を表示する閾値変数Ｔを第１値Ｎ2に設定するステップ、
b)前記アダプターが共用リソースに送信すべきパケットのデータを得るようにし、このデータを前記送信バッファに格納するステップ、
c)前記パケットのＴバイトのデータが前記送信バッファに格納された後にのみ、前記パケットの前記共用リソースへの送信を開始するステップ、及び
d)前記パケットの送信の開始後、衝突ウィンドウを含む予め定められた時間が満了したか否かを試験し、
(i)衝突ウィンドウが満了した場合は、第１状態はバッファアンダーフロー状態を含み第２状態は送信の完了を含むとして、第１及び第２状態の一つが生起するまで送信を継続し、
(ii)衝突ウィンドウが満了していない場合は、ステップ(d)を繰り返し、
(iii)ステップ(d)(i)でバッファアンダーフロー状態が生起し、且つ、再試行カウントが予め定められた限界に到達した場合は、送信を停止し、前記状態を表示するメッセージ信号を発生し、及び
（iv）ステップ(d)(i)でバッファアンダーフロー状態が生起し、且つ、再試行限界が前記予め定められた限界以下である場合は、ステップ(d)の前記送信を反映するために前記再試行カウントを変更し、閾値変数を前記第１値Ｎ2より大きい第２値Ｎ3に設定し、ステップ(b),(c)及び(d)を繰り返す
ステップを含む方法。
【発明の詳細な説明】
発明の分野
本発明は、イーサネットのようなローカルエリアネットワーク（ＬＡＮ）に関し、更に特別には、ネットワーク上の通信を制御するネットワークアダプターに関する。
本発明の背景
イーサネットは、衝突検出を具えたキャリアセンスマルチプルアクセス（ＣＳＭＡ／ＣＤ）に好適なネットワークアクセスプロトコルを用いるＬＡＮに対して通常用いられる名称である。ＣＳＭＡ／ＣＤプロトコルは、345 East 45th Street, New York, NY, 10017 USAのInstitute of Electrical and Electronics Engineers, Incによって発行されたＡＮＳＩ／ＩＥＥＥ　Ｓｔｄ．８０２．３に定められている。この標準は１０Mbps（メガビット／秒）ＣＳＭＡ／ＣＤチャネル（例えば、ネットワークバス）に適用されるが、本発明はこのようなチャネルに限定されるものではなく、例えば１００Mbpsにおける他のチャネル動作にも適用できることが理解されるべきである。
チャネルアクセスに対するＣＳＭＡ／ＣＤルールの下では、ネットワークの全てのノードが等しいアクセス優先権を持ち、チャネルが空くと同時に送信を開始することができる。メッセージ送信を「希望する」いずれのノードも、送信の開始の前にチャネルが空いていることを確かめるために、最初は「聴く」でなければならない。送信を希望するノードが、パケット間遅延と呼ばれる予め定められた時間、即ち９．６マイクロ秒間他の送信を検

出しない場合に送信を開始する。ノードはメッセージ及び制御情報を、ネットワークシステム上に、予め定められたサイズを持ち普通パケットと呼ばれるブロックとして送出する。

本発明の理解のために、一般的に直面する、過剰衝突及びネットワークアダプターバッファのアンダーフローを含む、潜在的にネットワークの性能を低下させる事象又は状態について説明することが有用と思われる。

Ａ．衝突処理及び過剰衝突

１を超えるノードが同時にチャネル上にパケットを送信すると、その結果、信号が干渉し正しくないものになり得る。このようなチャネル上へのパケットの同時多重送信は「衝突」と呼ばれる。

一般的に、送信を開始したノードが衝突を検出した場合、「衝突ウィンドウ」と呼ばれる予め定められた時間送信を続け、送信を希望する全てのノードが同様に衝突を確実に検出するようにする。衝突を検出した後で衝突ウィンドウの間ノードが送信するものは、一般的には、例えば、ＣＳＭＡ／ＣＤプロトコルに従った「１」と「０」とで変化する３２ビットの「ジャム」と呼ばれる無意味のデータである。全ての他の「アクティブ」ノード即ちデータパケットを送信した全ての他のノードが衝突を検出し、同様に３２ビットの「ジャム」を送信する。続いて衝突を検出した全てのノードが送信を停止する。

衝突に含まれたノードは、一般に「バックオフ」と呼ばれるパケット間遅延に付加遅延を加えた時間、送信を停止する。バックオフは、一般的にはノードによってランダムに選択された種々の長さを持つため、衝突に含まれる他のノードによって導入されたバックオフと異なる。それらのそれぞれのバックオフ時間が経過した後、アクティブノードは、それらのそれぞれのパケットを再び送信するために再試行を行う。バックオフ時間が種々に異なるため、再試行が成功すること即ち、ノードが他の衝突に遭遇しないことがある。

勿論、再送信の最初の試みが成功しないこともあり得る。送信の再試行は、次に、送信が成功するまで、又は予め定められた最大数（例えばＣＳＭＡ／ＣＤプロトコルに従えば１６）の試行が終わり全てが衝突によって停止するまで、バックオフを挟んで繰り返される。送信が成功せずに送信試行の最大数に至った場合は、当該ノードは「過剰衝突」を経験したといわれ、送信のために準備したパケットを廃棄する。過剰衝突状態はネットワークの上位プロトコルレイヤに報告される。これは、上位ネットワークレイヤによる回復は長いタイムアウト（例えば数秒）の後で行われ、実質的なパケット遅延を引き起こすので、システムの効率の低下をもたらす。更に、これは、このようなイベントの処理がプロセッサ時間を消費するので、システムの効率の低下をもたらす。

要するに、再試行と上位プロトコルレイヤの介入との間のバックオフ時間が導入されたことにより、過剰衝突とそれに続く再試行が、関連するパケットの送信のための長い待ち時間を持ち込み、システムの効率を低下させる。

Ｂ．ネットワークアダプター及びバッファアンダーフロー

ネットワークの動作の問題は、ネットワークの個々のノード中に位置する例えばイーサネットアダプターのようなネットワークアダプター中で起きる。ネットワークアダプターは、一般的にはアダプターを含むホストノードとこれが接続されているネットワークシステムとの間の通信を管理する。

例えば、ネットワークアダプターは、ネットワークシステムに接続されたディスク記憶装置とホストノードのメインメモリーとの間のデータの移動を管理する。ディスク記憶装置から上りのデータが受信される場合は、ネットワークアダプターがそのデータをホストノードのメインメモリーに転送し、次の処理を待つことができる。転送を行うために、ネットワークアダプターは、ホストノードの構成部分が接続されているシステムバスにアクセスしなければならない。同様に、ネットワークアダプターは、データがメインメモリーからシステムバス上に取出された後、ネットワークに下りのデータを転送する。

ネットワークシステム及びシステムバスは共用リソースと考えることができ、種々の方向へのデータ転送を実現するためのアクセスを仲介することが要求される。従って、ネットワークアダプターを介するデータ転送は、共用リソースの制御を行うことを必要とする。

これは、一般的に、(i)共用リソースに対するアクセスのための要求とそれに続くアクセスの許可、及び(ii)予め定められたサイズのデータの、共用リソースへの又は共用リソースからの「バースト」を必要とする。共用リソースはホストノードの他の部分のための動作を行うために占拠されることがあるので、共用リソースを常に直ちに使用できるとは限らない。共用リソースへのアクセスの要求の発行とその共用リソースへのアクセスの許可との間の時間は、リソースの「待ち時間」を構成する。一般的に、待ち時間は、例えばネットワークシステム又はシステムバスのスループット及びそれらに接続されたユニット（例えば、ホストの構成部分）の数のような、リソースの特性に応じてシステム毎に変わる。

変化し得る待ち時間を調整するため、ネットワークアダプターは、一般的に受信及び送信の両パス中にバッファメモリーを具える。受信バッファは、ネットワークシステムとシステムバスとの間の受信パスに沿って設置され、例えばシステムバスのような共用リソースへのアクセスが得られてそれへの送信を開始することができるまで、入来データを一時的に格納する。

送信バッファは、システムバスとネットワークシステムとの間の送信パスに沿って設置され、下りのデータを一時的に格納する。ネットワークシステムへのアクセスが許可されると、ネットワークに対し、一般的に予め定められた一定のレートで、例えばパケットのようなデータブロックが送信されなければならない。通常、例えばシステムバスへのアクセスがまだ許可されない時であっても、送信バッファにより、送出されるべき下りのデータを当該レートの定常的な流れとすることができる。

或る環境の下において、バッファアンダーフローと呼ばれる問題が生起する。バッファアンダーフローは、送信バッファ中に要求された送信レートを維持するために充分なデータが存在せず、且つ、このデータ不足を取り除くために、例えばシステムバスのような共用リソースに対するアクセスが得られない場合に起きる。

バッファアンダーフローは更に、例えばシステムバスのような共用リソースから送信バッファに入るデータのレートが、ネットワークシステム向けのバッファ中に存在するデータのレートより低い場合に起きる。この現象は、長いシステムバス待ち時間を持つノードで起こり得る。この現象は、システムバス上の機能の「バースト的」性質のために、システムバスの平均スループットがネットワークシステムの最大スループットより大きい場合にさえも起きる。そのシステムが通常は長い待ち時間を持たない場合でさえ、アダプターが長いシステムバス待ち時間に遭遇する時間間隔が存在するのである。

アンダーフロー状態が生起した場合、一般的にはそのノードはパケット送信を停止し、パケットの未送信部分を廃棄し、そのノードのプロセッサに中断信号を送る。この中断は、アンダーフロー状態のために送信がアボートされたことを表示する。上位ネットワークプロトコルレイヤは、アボートされたパケットの「受信アクノリッジ」を見ないので、結局タイムアウトになり、上述のように、この状態から回復するための動作を開始する。これらのための上位ネットワークプロトコルレイヤの介入は、前述のように、時間を消費し（プロセッサ時間に関して）システムの効率の低下を招き、（パケット遅延に関して）ネットワークの効率の低下を招く。

バッファアンダーフローを避けるため、既知のネットワークアダプターは、「送信閾値」と呼ばれる予め定められた量のデータが送信バッファに入った後にのみ送信を開始する。送信閾値は、送信されるべきパケットの全バイト数、又は、例えば、１５１８バイトパケットのうちの１２８バイトが既に送信バッファ中にあれば、閾値に到達したというように、実質的にバイト数として設定することができる。不幸にもこの方法は送信の開始時点で送信遅延を加える。これは、少なくとも、獲得され送信バッファに格納されるべきデータの量のために必要とされる時間の長さに等しい。これは、「内部送信遅延」と呼ばれる。この遅延は、ベンチマークドライバーがアダプター上で走行する時に得られる結果を悪くし、この遅延がパケット間ギャップより大きい場合には、明らかに特にネットワークのスループットに影響を与える。

本発明の要約

本発明は、過剰衝突状態又はバッファアンダーフロー状態のような好ましくない送信状態の時に、共用リソースに接続され且つ送信バッファを有するコンピュータネットワークアダプターによって実行される、パケットの自動再送信のための方法に存在する。この場合、そのパケットに関して以前に遭遇した衝突の数が過剰衝突限界以下の場合は、衝突の後実質的に直ちに続いて（バックオフなしに）再送信が行われるように設計される。本質的に、この方法は、生起する全ての衝突の検出及び各検出された衝突の衝突計数限界までのカウントを含む。衝突計数限界に到達する毎に、この方法においては、過剰衝突の数を表示する「過剰」衝突カウントを増す。この過剰衝突カウントが限界を超えた場合は、この方法においては、パケットの再送信を停止し、アダプターの送信バッファに格納されているパケットのデータを廃棄する。

過剰衝突限界は一定値とすることができ、また、ネットワークのトラヒック状態に対応して動的に決定することもできる。

本発明は、送信バッファアンダーフローにより不成功の送信試行に含まれるパケットの自動再送信のために、コンピュータネットワークアダプターで実行される方法に存在する。この場合、送信閾値が、初期の送信試行についての比較的小さい値から再送信のためのより大きい値まで増加される。この方法は、一般的に、(a)アンダーフロー状態が発生した場合に、送信バッファを停止し且つ送信されたパケットのフラグメントを受信端で「ラント」パケットとして廃棄するステップ、及び(b)更に高い送信閾値を用いて、Ｎ回の再試行の間に遅延又はバックオフの挿入なしに、選択された試行の数（「エントリー番号」即ち「Ｎ」）まで、実質的に直ちにパケットの他の送信を再試行するステップを含む。

初期の送信試行のためには、アダプターは、送信バッファに格納されるべきパケットについては小さいバイト数だけ、即ち、送信閾値について小さい値（例えば、４バイト）を要求することが望ましい。バッファアンダーフロー状態が生起した後には、パケットの実質的に大部分が送信のために送信バッファにエントリーした後でのみ、このアダプターは本発明による再試行を試みる。この要求により、度々初期のアンダーフロー状態の原因になるピーク待ち時間を克服するようにする。いずれかの再試行が成功すると、アダプターは、送信失敗を通知するための中断を発行する必要がなくなる。アルゴリズムは、アンダーフローパケットの送信の再試行をカウントすることができ、そのカウントをネットワーク又はシステムに対して、監視のために報告することができる。

更に特別には、共用リソースにデータを送信するために共用リソースに接続されたアダプターの動作について、本発明は、バッファアンダーフローが生起した場合に、パケット送信及びパケット送信の自動再試行を行うための方法を提供する。本発明による方法は、(A)予め定められた第１の量のデータが前記アダプターの送信バッファに格納された後に送信を開始するステップ、(B)次にバッファアンダーフロー状態が生起した場合は送信を停止し、第１の量のデータより多い予め定められた第２の量のデータが送信バッファに格納された後にのみ再び送信を開始するステップ、及び(C)ステップ(B)の後実質的に直ちに、再び送信を開始し、パケットの送信の完了又は再試行の数即ちカウントに対して予め選択された限界のいずれかに到達するまでの回数、ステップ(B)及びこのステップ（C)をこの順序で繰り返すステップを含む。後者の場合、この方法では送信バッファ中のパケットのデータを廃棄し、アンダーフロー状態を表示する信号を発生する。

以下に説明する方法は、バッファアンダーフロー状態に続いて、及び過剰衝突状態に続いて、パケットの自動再送信を行うことができる方法である。この観点によれば、パケットの送信の間に送信バッファアンダーフロー状態が生起した場合は、この方法においては、パケットの送信を停止し、パケットの第２の量のデータが送信バッファに格納された後にのみ再びそのパケットの送信を開始する。第２のデータ量は、アンダーフロー状態を引き起こした送信が試行される前に送信バッファに格納することが必要とされたデータ量より大きい。この方法では、ネットワークのトラヒック状態に対応して、過剰衝突限界及び第２のデータ量の両者を動的に決定することができる。これに代えて、過剰衝突限界及び第２のデータ量を予め一定値に定めておくことができる。

更に特別には、本発明によって提供される方法は、ネットワークアダプターにより、過剰

衝突状態に続いてデータのパケットの自動再送信を実行することができる。パケットの送信は、予め定められたバイト数のパケットが送信可能になった後に開始される。衝突が検出されると、衝突カウントを増し、衝突カウントが、本質的に「過剰衝突」を決める例えば１６のような予め定められた第１限界を超えていない場合は、他の送信の試行が実行される。（勿論、アンダーフロー状態のような他の状態が生起した場合には送信を中断することができるが、このような他の状態は、この場合の範囲に入っていない。）衝突カウントが第１限界を超えた場合は、この方法は複数のステップを実行する。第１に、再試行（ＲＮＥＣ）カウントが第２限界を超えた場合はパケットの送信を停止し、第２に、ＲＮＥＣカウントが第２限界を超えない場合はＲＮＥＣカウントを増し、この方法は増加されたＲＮＣＥカウントを用いて再び前のステップを実行する。このため、本発明の送信開始ステップは、第１限界と第２限界との積、即ち、衝突カウントの最大値とＲＮＥＣカウントの最大値との積に等しい回数まで実行される。

従って、本発明は、再試行間のバックオフタイムを導入する必要性を除去し、上位プロトコルレイヤの大部分の介入を除去するという方法で過剰衝突状態を取扱うための優れたメカニズムを提供する。従って、本発明は、複雑なパケットの送信のための長い待ち時間がなく、従来の技術の説明の中で上述したようなシステムの効率を低下させることのない、自動送信を提供する。

【図面の簡単な説明】

本発明の前記及び他の目的、利点及び特徴は、添付図面を用いて行われる以下の本発明の詳細な説明を参照することにより、更に容易に理解されるであろう。図面中、

図１は、本発明の実施例によるネットワークシステムに接続されるノードを表すブロック図であり、

図２は、本発明の実施例による図１のアダプターにより実行することができるアンダーフロー状態に続く自動再送信のためのアルゴリズムのフローチャートであり、且つ

図３は、本発明の実施例による図１のアダプターにより実行することができる過剰衝突回復のためのアルゴリズムのフローチャートである。

好ましい実施例の説明

図１は、ＬＡＮ又はディスク記憶装置のようなネットワーク12に接続された、例えばコンピュータシステムであるノード10を示す。ノード10はネットワークアダプター14、メインメモリー16、ＣＰＵ18、及び、全て例えばＰＣＩバスのようなシステムバス22によって相互に接続された端子のような他の周辺ユニット20を含む。アダプター14は、望ましくは双方向パス24を通してシステムバス22に接続され、望ましくは双方向パス26を通してネットワークシステム12に接続される。ネットワークシステム12は、ノード10のアダプター14と、例えば同様の構成でそれに接続されている他のノード27のアダプター（図示せず）との間の、例えば高帯域幅、半二重データ通信を受容するためのバストポロジーを実現することが望ましい。これに代えて、アダプター14は、ネットワークシステム12に接続されたノード10,27間の全二重通信を実現する構成であってもよい。

受信動作の間は、データはネットワーク12からアダプター14に転送され、結局はメインメモリー16に転送されて処理を待つ。アダプター14とメインメモリー16との間のデータ転送は、システムバス22上の直接メモリーアクセス（ＤＭＡ）転送によって遂行される。ＤＭＡ転送は、システムバスへのアクセスの要求とそれに続くアクセスの許可、及び、予め定められたサイズのデータの「バースト」を含む。送信動作の間は、メインメモリー16から取出されたデータがアダプター14からネットワークシステム12に転送される。ネットワークシステム12とアダプター14との間のデータの転送は、例えば、個々のパケットについて固定レートで行われる。

システムバス22は、アダプター14によりメインメモリー16に対して読出し及び書込み転送を遂行するためのアクセスに、常に直ちに利用できるとは限らない。バス22がそれに接続された他のユニットを含む他の動作を遂行するために占拠されることがあり、また、例えばユニット20のような他の部分が読出し又は書込みメモリー転送を行うことがある。そのため、システムバス22は（メインメモリー16と同様に）共用リソースと考えられ、システ

ムバス22へのアクセス要求の発行とバスへのアクセスの許可との間の時間は、システムバス22の「待ち時間」を構成する。
バスの待ち時間に対応するため、アダプター14は、バッファメモリー、好ましくはＦＩＦＯ（「ファーストイン、ファーストアウト」）送信バッファ30及びＦＩＦＯ受信バッファ32を具える。線26を通ってネットワークシステム12からアダプター14に入ったデータは、受信ステートマシン36の制御の下で、ネットワークインタフェース34を通って受信バッファ32に到達する。受信バッファ32は、アダプター14がシステムバス22へのアクセスを得るまでデータを一時的に格納し、アクセスを得た時に受信ＤＭＡ（直接メモリーアクセス）モジュール37が、受信ＦＩＦＯバッファ32中のデータを「読出し」、メインメモリー16に書込み転送を行う。この後、データは、システムバスインタフェース38及びシステムバス22を通ってメインメモリー16に書込まれる。
ネットワークシステム12に送信されるべきデータは、送信ＤＭＡモジュール39により、メインメモリー16から、システムバス22にアクセスしているシステムバスインタフェース38上に読出され、このデータは送信バッファ30に置かれる。アダプター14がネットワークシステム12へのアクセスを獲得すると、データは、送信ステートマシン40の制御の下にネットワークインタフェース34に送られ、次にネットワークシステム12に送られる。アダプター14の各部分は内部バス35で相互に接続されている。
ネットワークインタフェース34は、アダプター14がネットワークシステム12との通信を行うために必要なタイミング及び電気的特性に確実に合致するために必要な、通常のデータパスの論理的及び物理的コネクションを含む。例えばネットワークインタフェース34は、例えばイーサネットのような適当なネットワーク12のネットワークプロトコルを実行するための媒体アクセス制御装置（「ＭＡＣ」）を具えることができる。
システムバスインタフェース38は、アダプター14がシステムバス22上で通信を行うために必要なタイミング及び電気的特性に確実に合致するために必要な、通常のデータパスの論理的及び物理的コネクションを含む。例えばシステムバス22は、例えばＰＣＩバスであってもよいし、インタフェース38は、例えばＰＣＩバスインタフェースあってもよい。
アダプター14を通る双方向のデータの流れは、インタフェース38の調整機能によって制御される。この調整機能は、ＤＭＡマシン37、39に代わって、通常の調整プロセスのとおりにシステムバス22の制御のための調整を行う。受信及び送信ステートマシン36、40を具えているので、調整モジュール38は、抵抗器及び直列論理回路として構成された結合論理（図示せず）を含むことが望ましい。
ＤＭＡマシン37及び39は、送信及び受信バッファ30、32を通る、ネットワークとシステムバス22との間のデータの双方向の流れを管理する。特に、送信ＤＭＡマシン39は、システムバス上の読出し転送と共に、メインメモリー16からの下りデータバーストの転送を開始する。これらの下りデータバーストは、一時的に送信バッファ30に格納され、続いて送信ステートマシン40により、ネットワークシステム12上に送信される。受信ステートマシン36は、ネットワークシステム12からの上りデータバーストを管理して一時的に受信バッファ32にデータを格納し、受信ＤＭＡステートマシン37は、バスインタフェース38及びシステムバス22を介してメモリー16にデータを移動させる。ステートマシン36、40によって遂行される機能の例は、データをバッファに格納した後又はそれに先立って、ネットワークシステム12への及びそれからのデータのビットストリームをバイト幅のワードに変換することである。
受信バッファ32に受信されたデータについての予め定められた閾値レベルに基づいて、及び、システムバス22に対する獲得アクセスに基づいて、受信ステートマシン36は、予め定められたサイズの上りデータのバーストを、メインメモリー16に向けた書込み転送として、システムバス22上への転送を開始する。データのバーストのサイズは、システムバスの特性によって変わり得る。システムバス22上のバーストの間に転送されるデータの量は、一般的に、例えばパケットとしてネットワークシステム12上に転送されるデータのブロックより少ない。受信ＤＭＡマシン37及び送信ＤＭＡマシン39は、バス22上のデータの一つのバーストの各転送について、システムバス22に対するアクセスを調整する必要がある。

ここで説明されていないステートマシン36、40、ＤＭＡマシン37、39、及びインタフェースモジュール38の構成及び動作については、当業者にとって明らかであり、ここに示されたようなもの以外は一般的なものである。
図２は、送信ステートマシン40で実行される、送信バッファアンダーフロー状態の後における自動再送信のためのアルゴリズム100を示す。
ブロック110では、送信ステートマシン40は非アクティブ、即ちアイドルモードにある。
ブロック110は送信のための新しいパケットをフェッチする。ブロック110は更に、「NO__OF__RETRIES」と呼ばれる変数及び閾値を表す変数「Ｔ」を含む変数を初期化する。NO__OF__RETRIESは、送信ステートマシン40が試行できる再試行の組の数を表す。閾値「Ｔ」は、送信の開始が許可される前に送信バッファ30で利用できるものとして要求されるバイト数を表す。
これらの変数NO__OF__RETRIES及びＴの両者は、アルゴリズム100が実行される特定のアダプターの個々の必要性を考慮して、プログラマブルである値を持つことが望ましい。本発明は、例えばNO__OF__RETRIESの値に、値１の限界を設けることができる。これは、アンダーフロー状態が生起した時はいつでも、送信ステートマシンが１組の追加の再送信を行うようにすることである。
変数Ｔは、最初は値Ｎ1に設定することができ、アダプター14の動作中は例えば三つの値Ｎ1、Ｎ2又はＮ3のいずれかであると推定できるとする。ここで、Ｎ1は中間の値、Ｎ2はＮ1以下の値、及びＮ3はＮ1以上の値である（Ｎ2≦Ｎ1≦Ｎ3）。Ｎ1は例えば７２バイトである。Ｎ2は例えば、送信バッファ30中の最小アドレス可能ブロックデータ（即ち、一つのロングワード）に含まれるバイト数であり、従って送信ステートマシン40がそこからフェッチすることができる最小バイト数である。従ってＮ2は４バイトのデータ程度であり得る。Ｎ3はパケットの実質部分に等しく、例えば１２８バイトである。
ブロック112は、閾値制御ビットが例えば論理HIGHに設定されているか否か、及び、アンダーフローフラブが例えば論理LOWにクリアされているか否かの両者を試験する。閾値制御ビットがセットされておりアンダーフローフラグがクリアされていると、ブロック114で閾値Ｔを初期値Ｎ1からＮ2に減らす。即ち、送信を開始するために送信バッファ30中に必要なバイト数を減らす。アンダーフローフラグは、アンダーフローが生起したか否かを示す。これがセットされると、以下に説明するように、アルゴリズム100がＴの値を可能な最大値、即ちＮ3に変える。
ブロック114の後、又はブロック112の試験の結果が否定の（即ち閾値制御ビットがクリアされたか又はアンダーフローフラグがセットされた）場合、ブロック116で、閾値Ｔに到達するまで、即ち送信バッファ30が適用されるＴの値によって決まる送信可能なバイト数を持つまで、送信ステートマシンを待たせる。次に、ブロック118で送信を開始する。
ブロック120では、衝突ウィンドウが満了したか否かを試験する。衝突ウィンドウはアダプターがネットワーク12上での衝突を監視する時間であり、時にはネットワークアキジションタイムと呼ばれる。衝突ウィンドウは、衝突ウィンドウインジケーターと呼ばれる他の変数によって特定され、一般的にネットワークの仕様によって決まるシステムパラメータを表す。１０Mbpsイーサネットワークにおいては、衝突ウィンドウインジケーターは５１．２マイクロ秒のネットワークアキジションタイムを表す。
衝突ウィンドウインジケーターによって表される衝突ウィンドウが満了していない場合は、ブロック122において、アンダーフロー状態が生起したことを示すアンダーフローフラグがセットされているか否かを試験する。上述のように、アンダーフロー状態は、送信バッファ30がネットワークのために必要とされる送信レートに合致するに充分なパケットデータを持っていない時に生起する。アンダーフローフラグがセットされていない場合は、アルゴリズム100はブロック120に戻って衝突ウィンドウが満了したか否かを試験する。
一方、ブロック122の試験においてアンダーフロー状態が生起したことが表示されると、ブロック124において、NO__OF__RETRIESが限界Ｒ、即ち１より少ないか否かを試験する。言い換えると、衝突ウィンドウの間にアンダーフロー状態が生起し、再試行の回数がＲより少ない場合は、ブロック124の試験は「ＹＥＳ」であり、逆の場合は、ブロック124の試

験が「ＮＯ」である。
ブロック124の試験の結果が肯定の場合は、アルゴリズム100はブロック126に進む。このブロックでは、NO__OF__RETRIESを例えば１だけ増加させる。このように、ブロック126を最初に通過した後はＲ＝１である。
ブロック128では、アンダーフローフラグがセットされ、アンダーフロー状態のパケットを再送信するための計画を進める。ブロック130では、アンダーフローフラグのセットに応答して閾値を値Ｎ3に増加させる。続いて、アルゴリズム100はブロック112に戻る。
ブロック120の試験により衝突ウィンドウが満了したことが示されると、ブロック132では、送信ステートマシンがパケットの送信を続ける。
試験間隔と呼ばれる予め選択された時間の後、ブロック134ではアンダーフロー状態を試験し、衝突ウィンドウが満了した後にアンダーフローが生起したか否かを決定する。そのようなアンダーフローが生起した場合は、ブロック136でアンダーフロー中断を宣言し、次にブロック138でアンダーフローフラグをクリアする。ブロック138に続いてアルゴリズム100はブロック110に戻る。ブロック110では、送信ステートマシン40がアイドルモードを再開し、このブロックに関して上述のように、変数がリセットされる。
ブロック134の試験においてアンダーフロー状態が生起していないことが表示された場合は、次にブロック140において、最終送信パケットが送られたか否かを試験することにより、パケット送信が完了したか否かを試験する。送信が完了していない場合は、アルゴリズム100はブロック132に戻り、ここで送信が継続される。一方、送信が完了した場合は、アルゴリズム100はブロック138に進み、アンダーフローフラグがクリアされ、上述のように、次にブロック110に進む。
従って、アルゴリズム100は、送信バッファ30に対してデータのアンダーフローが生起した場合における自動再送信のための優れた技術を具え、システム効率を改善することができ、データの待ち時間を低減することができる。
アルゴリズム100は、衝突ウィンドウ間隔における送信の間にアンダーフローが生起した場合にのみ、「自動再試行」を実行することが明確にされるべきである。本発明は、衝突ウィンドウ間隔の間及び後の両者を含む送信期間中のいずれの時においても「自動再試行」を実行するように、更に一般的に実施することができる。
図３は過剰衝突回復のためのアルゴリズム200を示す。アルゴリズム200は、バッファアンダーフローに対する自動送信に関するアルゴリズム100とは別個に送信ステートマシン40によって実行される。即ち、当業者には明らかなように、二つのアルゴリズム100，200は相互に統合することができる。アルゴリズム200はブロック210でスタートする。ここでは送信のための新しいパケットを待ち、以下に説明するように、アルゴリズム200で用いられる特別の変数「ＲＮＥＣ」及び「Ｔ」を初期化する。
ブロック212では閾値Ｔに到達したか否かを試験する。これは、ブロック116に関して上述した方法と同じ方法で行われることが望ましい。この場合、閾値ＴはＮ1に等しいこともあり得るし又は異なる値にセットされることもあり得る。閾値Ｔに到達していない場合は、アルゴリズム200は単に閾値に達するまでブロック212の試験を繰り返す。ブロック212では更に変数COLLISION__COUNTをゼロに初期化する。
他方、閾値Ｔに到達した場合は、ブロック214では送信ステートマシン40がデータの送信を始める。ブロック215では、ブロック120に関して利点として上述したように、衝突ウィンドウインジケーターを試験することにより、衝突ウィンドウが満了したか否かを試験する。衝突ウィンドウが満了した場合は、アルゴリズム200は、ブロック216で送信を完了した後ブロック210に戻り、新しいパケットに対して前述のステップを繰り返す。衝突ウィンドウが満了していない場合は、ブロック217で、送信に含まれたパケットに関して衝突が生起したか否かを検出する。ブロック217で衝突が検出されない場合は、アルゴリズム200はブロック215に戻る。
関連するパケットに関して衝突が検出される度に、ブロック218でCOLLISION__COUNTと呼ばれる変数を増し、ブロック219でCOLLISION__COUNTの値を試験する。COLLISION__COUNTの値が予め選択された限界、例えば適用する標準によって設定された１６に到達した場合

は、ブロック219で、過剰回数の衝突が生起したことが表示される。他方、COLLISION_COUNTが１６未満である限り、ブロック219の試験は、過剰回数の衝突は生起していないことを表示する。

過剰衝突状態が生起していないことを表示するブロック219の否定の結果に続いて、ブロック220では、好ましくはアルゴリズム100のブロック140に関する方法と同じ方法により、送信か終了したか否かを試験する。送信が終了した場合は、アルゴリズム200はブロック214に戻り、同一パケットの送信を再開する。言い換えると、アルゴリズム200では、例えば（１６回までの）ＣＳＭＡ／ＣＤ通常送信再試行を行う。ブロック220でまだ送信が終了していないことが表示されると、ブロック221では、ブロック220の試験が繰り返される前に、ビットタイムのオーダだけ待機する。「ビットタイム」はチャネルを通して１ビット送信するのに必要な時間であり、例えば、１００Mbpsで動作するチャネルについては１０ナノ秒（又は１０Mbpsチャネルについては１００ナノ秒）である。ブロック221の「待機」期間の間、「本発明の背景」の項で前述したように、ノードはジャムを送信する。

本発明によれば、ブロック219で過剰衝突状態が生起したことが表示された場合は、ブロック222-228の後、「過剰衝突に対する再試行回数」（ＲＮＥＣ）と呼ばれる回数の間、衝突が検出されず送信が成功するまで、又は、予め選択されたＲＮＥＣの限界に到達するまでの間、反復してブロック212-220のステップが繰り返される。例えば、COLLISION_COUNT限界が１６であり、ＲＮＥＣ限界が２の場合は、アルゴリズム200はブロック212-220を３２回まで繰り返す（１６回の繰り返しを２回繰り返す）。繰り返された衝突に関連するパケットの送信が成功せずにＲＮＥＣ限界に到達した場合は、上述の従来技術と同様に、このアルゴリズムはパケットを廃棄し、ブロック210に戻り、新しいパケットを待つ。

更に特別な場合、ブロック219の試験により過剰回数の衝突が生起したことが表示された場合は、ブロック222でＲＮＥＣ限界を獲得し、ブロック223でＲＮＥＣが予め設定された限界より小さいか否かを試験する。ブロック223の試験においてＲＮＥＣ限界に到達したことが示された場合は、上述のように、アルゴリズム200は送信終了後ブロック210に戻る。ブロック223の試験においてＲＮＥＣ限界に到達していないことが示された場合は、ブロック224でＲＮＥＣカウントを例えば１だけ増す。次に、ブロック226により、ネットワークインタフェース34が、そのパケットを初めて送信されるべき新しいパケットのように取扱う。ブロック228では、過剰衝突に関連するパケットを再送信のために再び待つ。続いてアルゴリズム200はブロック212に戻る。

従って、アルゴリズム200は、過剰衝突回復のための優れた技術を具え、システム効率を改善することができ、データ待ち時間を低減することができる。ＲＮＥＣ限界は予めプログラムすることができ、又はトラヒック状態に応じて動的に決定することもできる。言い換えれば、ネットワークが過密のピークの期間を経験する時は、例えば、存在し得る高度の衝突状態に悪化させないように、アダプター14がＲＮＥＣ限界を減らすことができる。他方、他の時刻において、次のパケットの送信を成功させる可能性を増すために、及び、送信が成功するまでこのパケットの送信のために上位レイヤプロトコルが再スケジューリングを行う必要性を除去することによってパケット待ち時間を減らすために、ＲＮＥＣ限界を増加させることができる。ブロック222では、予め設定されたＲＮＥＣ限界を獲得するか、又は例えばネットワークステータスパケットから得られたネットワークの状態情報に基づいてそれを計算する。

<u>他の特徴及び実施例</u>

上記においては本発明の実施例を説明した。当業者にとっては、ここになされた開示から種々の他の特徴及び実施例が明らかである。例えば、上述の実施例では、パケットデータが送信バッファ中で上書きされない限り何時でも、アンダーフロー又は過剰衝突に対する再試行を行うことができるが、本発明においては、送信衝突ウィンドウの間にアンダーフローが起きた場合にのみ実行できるようにすることができる。

更に、送信の成功の可能性を最適化するため、アンダーフロー又は衝突の履歴情報、ネットワークトラヒック状態、及び／又はバス負荷状態に基づいて再試行の最大数を計算するアルゴリズムを用いて本発明を実施することが可能である。

この点について、特に送信バッファアンダーフロー状態を考えることとする。アンダーフローは、ホストシステムバスのピーク待ち時間が大きく、及び／又はシステムバスのスループットが小さい時に起きる。従ってアンダーフローは主としてホストＣＰＵシステムバスに依存する。他方、ネットワークが輻輳し、Ｒが高い値に設定されると、ネットワークに対する不必要な干渉が始まる。このため、或る種のアプリケーションにおいては、ネットワークの高度の輻輳及び／又は高度のホストＣＰＵバスの負荷の期間中に、Ｒについて、その値を減らすために（予め設定された一定値に代えて）動的に計算された値を用いることが合理的である。このため、ブロック124では、限界Ｒについて固定値を用いずに、NO__OF__RETRIESの試験においてその値を用いる前の上述の履歴情報に基づいてＲを計算するとよい。

再試行が不成功に終わる状態を防ぐために、予め定められた不成功の再試行の回数の後、次の再試行の前に計算された期間の遅延を挿入するような、バックオフスケジュールを計算するためのアルゴリズムを用いることも可能である。

アンダーフロー又は過剰衝突について、再試行の開始時刻は、更に、送信バッファに直接メモリーアクセスした関連パケットのバイト数の関数である。この関数は、（例えば、アンダーフローについては、パケットの少なくとも１６０バイトがバッファ中に含まれるまで待つように）プログラムされた値とすることができ、又は履歴からの学習できるようにすることもでき、又は再試行回数の関数であってもよい。

このように、本発明は、パケットの損失を最小にすることができ、上位ネットワークレイヤの介入なしにアンダーフロー又は過剰衝突を回復させることができる。その結果、本発明は、一時的に発生するシステムバスの大きな待ち時間のためにアンダーフロー又は過剰衝突が生起した場合においても、パケットを送出するための、素早く簡潔な方法を提供する。本発明は更に、上位ネットワークレイヤの介入を避けることによって、平均パケット遅延を低減することができる。更に、本発明は、アンダーフローを避けるために従来技術では必要であった一般的に大きい量のパケットの受信を待つ必要がないので、送信バッファへのパケット転送のスタートから送信のスタートまでの遅延時間を一層短くすることができる。

本発明の特定の実施例について説明したが、この説明は、制限する意味に解釈されるべきではない。当業者には、この説明から、本発明の開示された実施例及び他の実施例の種々の変更が明らかであろう。従って、添付の請求の範囲がこのような変更又は実施例を本発明の真の範囲の中に入れるようにカバーすることが意図されている。

【図１】

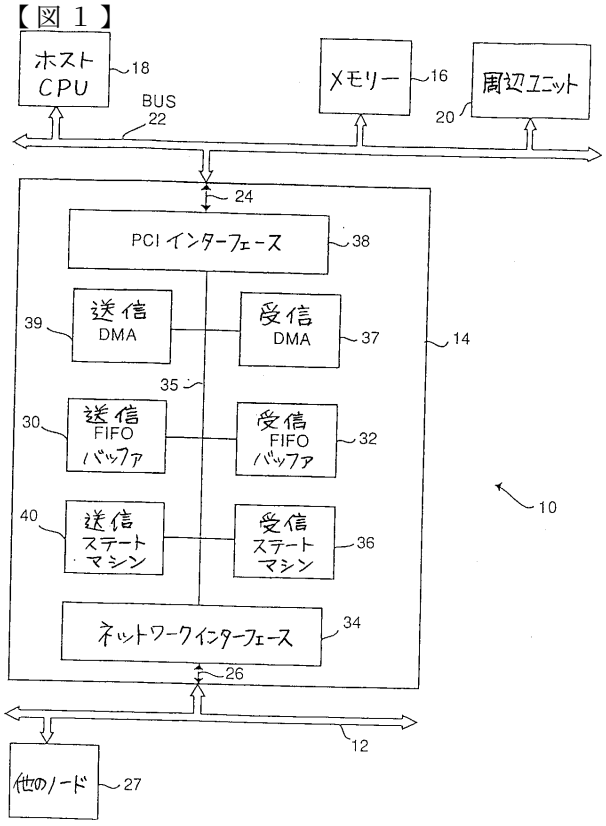

Fig. 1

【図２】

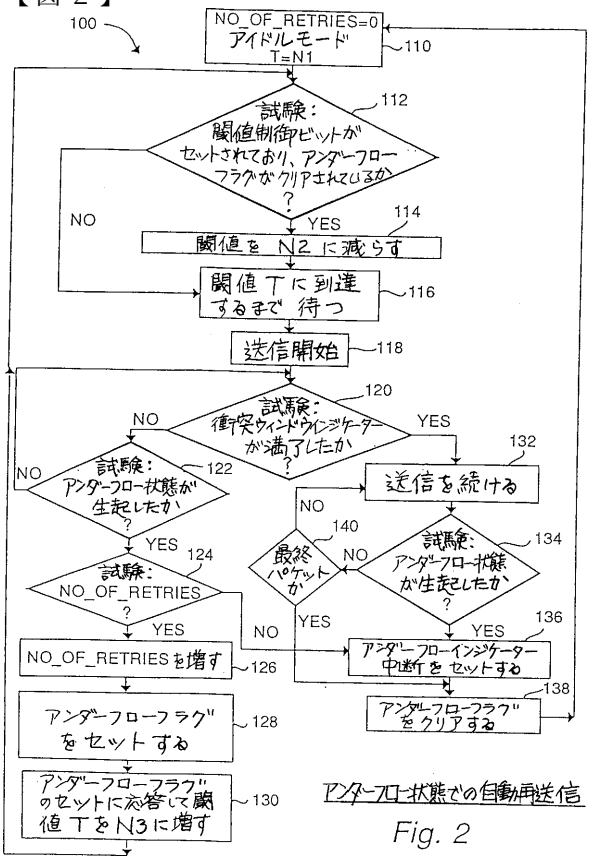

Fig. 2

【図３】

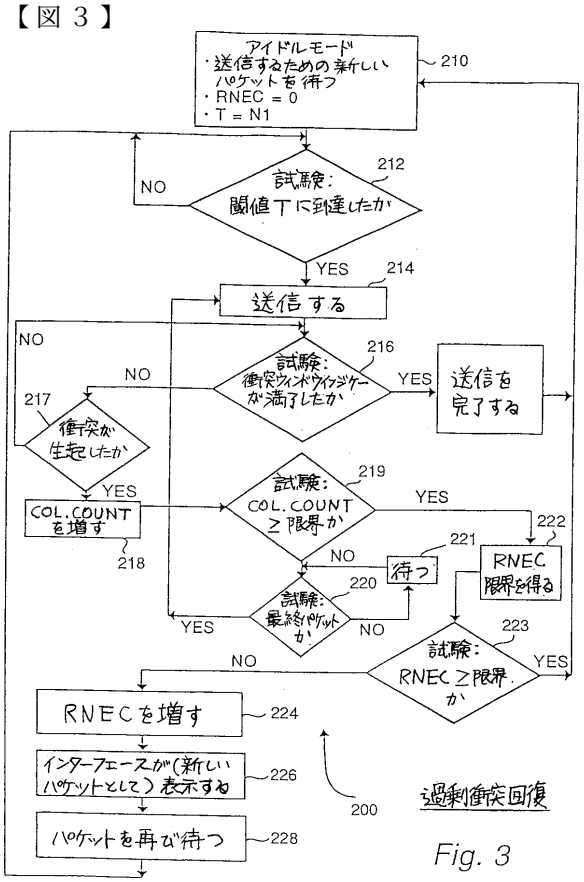

Fig. 3

フロントページの続き

(74)代理人
弁理士　高見　和明
(74)代理人
弁理士　梅本　政夫
(72)発明者　ポール　ギデオン
イスラエル国　９３９０３　ギロ　エルサレム　ボセム　ストリート　１６／９
(72)発明者　ヴェルシメル　アヴィダッド
イスラエル国　９５４２７　エルサレム　レインズ　６　ストリート（番地なし）
(72)発明者　ベンーマイケル　シモニ
イスラエル国　９０９１７　ギファット　ゼーフ　ミップ　ストリート　１３／３
(72)発明者　ハウィ　ウィリアム
アメリカ合衆国　マサチューセッツ州　０１４６３　ペッペレル　インディペンデント　ロード　１６

審査官　矢頭　尚之

(58)調査した分野(Int.Cl.[7], ＤＢ名)
H04L 12/40