

(11) 特許出願公表番号

特表2013-507023

(P2013-507023A)

(43) 公表日 平成25年2月28日(2013.2.28)

(51) Int.Cl.
H04L 12/70

F I
H O 4 L 12/56

テーマコード (参考)
5K030

審查請求 未請求 予備審查請求 未請求 (全 38 頁)

(21) 出願番号	特願2012-531436 (P2012-531436)
(86) (22) 出願日	平成22年9月30日 (2010. 9. 30)
(85) 翻訳文提出日	平成24年5月18日 (2012. 5. 18)
(86) 国際出願番号	PCT/EP2010/064605
(87) 国際公開番号	W02011/039332
(87) 国際公開日	平成23年4月7日 (2011. 4. 7)
(31) 優先権主張番号	12/571, 018
(32) 優先日	平成21年9月30日 (2009. 9. 30)
(33) 優先権主張国	米国 (US)
(31) 優先権主張番号	12/788, 205
(32) 優先日	平成22年5月26日 (2010. 5. 26)
(33) 優先権主張国	米国 (US)

(71) 出願人 5110000633
エスティー - エリクソン、ソシエテ、アノ
ニム
スイス国ブラン - レ - ズアト、シュマン、
デュ、シャン - デ - フィーユ、39

(74) 代理人 100117787
弁理士 勝沼 宏仁

(74) 代理人 100082991
弁理士 佐藤 泰和

(74) 代理人 100103263
弁理士 川崎 康

(74) 代理人 100107582
弁理士 関根 毅

(74) 代理人 100118843
弁理士 赤岡 明

[最終頁に続く](#)

(54) 【発明の名称】 接続指向の正順デリバリ環境において接続を管理するための方法およびシステム

(57) 【要約】

本開示は、正順デリバリ環境においてクライアントとサーバとの間の接続を確立するためのシステムおよび方法を提供する。本開示のシステムおよび方法は、第1の種類のメッセージをサーバへと送信することによって接続の確立を要求するように構成されたクライアントを含んでおり、サーバが、サーバの接続につながる第2の種類のメッセージをクライアントへと送信することによって接続の確立が可能であることを確認するように構成されている。第1の種類のメッセージにより、第1の最大応答時間としての第1の所定の期間を測定する第1のクライアントタイマが開始され、第2の種類のメッセージまたはデータメッセージの受信によって、第1のクライアントタイマが停止される。接続は、第3の種類のメッセージの送信によって閉じられる。

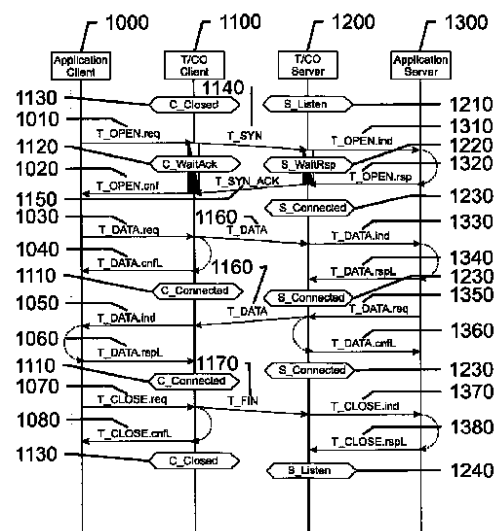


Fig. 1

【特許請求の範囲】**【請求項 1】**

接続指向の正順デリバリ環境における接続を管理するための方法であって、

a) クライアントが第 1 の種類のメッセージをサーバへと送信することによって、接続の確立を要求し、

b) サーバが、サーバの接続につながる第 2 の種類のメッセージをクライアントへと送信することによって、接続の確立が可能であることを確認し、

c) 前記第 1 の種類のメッセージを送信することによって、第 1 の最大応答時間としての第 1 の所定の期間を測定する第 1 のクライアントタイマを始動し、前記第 2 の種類のメッセージまたはデータメッセージを受信することによって、前記第 1 のクライアントタイマが停止され、

10

d) 第 3 の種類のメッセージを送信することによって接続が閉じられる、方法。

【請求項 2】

第 2 の種類のメッセージを受信されることなく前記第 1 の所定の期間が過ぎた場合に、別の第 1 の種類のメッセージが送信される請求項 1 に記載の方法。

【請求項 3】

サーバが接続を引き受けることができない場合に、サーバがエラーメッセージを返す請求項 1 に記載の方法。

【請求項 4】

a) 前記第 2 の種類のメッセージを受信することが、クライアントの接続につながり、

20

b) 前記第 2 の種類のメッセージの受信を確認するための第 4 の種類のメッセージを、該クライアントがサーバに送信する、請求項 1 に記載の方法。

【請求項 5】

前記第 2 の種類のメッセージを送信することによって、第 2 の最大応答時間としての第 2 の所定の期間を測定する第 1 のサーバタイマを始動し、前記第 4 の種類のメッセージを受信することによって、該第 1 のサーバタイマが停止される請求項 4 に記載の方法。

【請求項 6】

サーバが接続を確立できない場合に、サーバがクライアントの非接続につながる第 5 の種類のメッセージをクライアントへと送信する請求項 1 に記載の方法。

【請求項 7】

30

前記第 3 の種類のメッセージの送信により、第 3 の最大応答時間としての第 3 の所定の期間を測定する第 2 のクライアントタイマが始動される請求項 1 に記載の方法。

【請求項 8】

サーバが第 6 の種類のメッセージを送信することによって、前記第 3 の種類のメッセージの受信を確認する請求項 1 に記載の方法。

【請求項 9】

前記第 6 の種類のメッセージの送信により、第 4 の最大応答時間としての第 4 の所定の期間を測定する第 2 のサーバタイマが始動される請求項 8 に記載の方法。

【請求項 10】

クライアントにおける第 6 の種類のメッセージの受信により、前記第 2 のクライアントタイマが停止されるとともに、

40

クライアントが第 4 の種類のメッセージを送信する請求項 7 に記載の方法。

【請求項 11】

サーバによる前記第 3 の種類のメッセージの送信により、第 5 の最大応答時間としての第 3 の所定の期間を測定する第 2 のサーバタイマが始動される請求項 1 に記載の方法。

【請求項 12】

前記第 3 の種類のメッセージを受信するクライアントが第 6 の種類のメッセージを送信することによって、該第 3 の種類のメッセージの受信を確認する請求項 1 に記載の方法。

【請求項 13】

前記第 6 の種類のメッセージの送信により、第 4 の最大応答時間としての第 4 の所定の

50

期間を測定する第2のクライアントタイマが始動される請求項12に記載の方法。

【請求項14】

サーバにおける前記第3の種類のメッセージの受信により、前記第2のサーバタイマが停止される請求項11に記載の方法。

【請求項15】

クライアントとサーバとの間の前記メッセージのうちの少なくとも1つが、ルータを通過し、

該少なくとも1つのメッセージが、リソース予約の要求を含んでおり、

前記ルータが、要求されたリソースを提供できる場合に限って前記少なくとも1つのメッセージを転送する、請求項1に記載の方法。

10

【請求項16】

ルータが前記少なくとも1つのメッセージを転送する場合に、該ルータは該リソース予約の要求に関する接続の確立に関わるクライアント・サーバ・ペアに係る第2の種類のメッセージの受信まで、さらなるリソース予約の要求を一時的に停止する請求項15に記載の方法。

【請求項17】

接続が閉じられる場合に、少なくとも1つのメッセージがリソースの予約を取り消すために使用される請求項15に記載の方法。

【請求項18】

予約されるリソースが帯域幅である請求項15に記載の方法。

20

【請求項19】

クライアントおよびサーバのそれぞれが、それぞれのアプリケーションクライアントとそれぞれのアプリケーションサーバとの間の接続を確立する請求項1に記載の方法。

【請求項20】

接続指向の正順デリバリ環境において接続を管理するためのシステムであって、

サーバと、

クライアントと、

正順デリバリのネットワークと、 を備えており、

前記サーバが、請求項1に記載の方法のうちのサーバに関する動作のいずれかを実行するように構成され、

30

前記クライアントが、請求項1に記載の方法のうちのクライアントに関する動作のいずれかを実行するように構成されているシステム。

【請求項21】

ルータを備えており、

クライアントとサーバとの間の前記メッセージのうちの少なくとも1つが、該ルータを通過しており、

該少なくとも1つのメッセージが、リソース予約の要求を含んでおり、

前記ルータが、要求されたリソースを提供できる場合に限って前記少なくとも1つのメッセージを転送する、請求項20に記載のシステム。

【請求項22】

40

前記ルータが前記少なくとも1つのメッセージを転送する場合に、該ルータは該リソース予約の要求に関する接続の確立に関わるクライアント・サーバ・ペアに係る第2の種類のメッセージの受信まで、さらなるリソース予約の要求を一時的に停止する請求項21に記載のシステム。

【請求項23】

前記サーバおよび前記クライアントの各々が、前記サーバタイマおよび前記クライアントタイマのそれぞれを実現するためのただ1つのタイマを備えている請求項20に記載のシステム。

【請求項24】

サーバにおける前記第3の種類のメッセージの受信により、サーバは第4の種類のメッ

50

セージを送信する請求項 11 に記載の方法。

【請求項 25】

サーバにおける第 6 の種類のメッセージの受信により、サーバは第 4 の種類のメッセージを送信する請求項 11 に記載の方法。

【請求項 26】

前記第 4 の種類のメッセージの受信により、前記第 2 のサーバタイマが停止される請求項 24 または 25 に記載の方法。

【請求項 27】

サーバにおける第 6 の種類のメッセージの受信により、前記第 2 のサーバタイマが停止される請求項 11 に記載の方法。

【請求項 28】

クライアントにおける前記第 3 の種類のメッセージの受信により、前記第 2 のクライアントタイマが停止される請求項 7 に記載の方法。

【請求項 29】

クライアントにおける第 6 の種類のメッセージの受信により、前記第 2 のクライアントタイマが停止される請求項 7 に記載の方法。

【請求項 30】

前記第 4 の種類のメッセージの受信により、前記第 2 のサーバタイマが停止される請求項 10 に記載の方法。

【発明の詳細な説明】

【関連出願および優先権の主張】

【0001】

本出願は、「METHOD AND SYSTEM FOR MANAGING A CONNECTION IN A CONNECTION ORIENTED IN-ORDER DELIVERY ENVIRONMENT」という名称の 2009 年 9 月 30 日付の米国特許出願第 12/571,018 号の部分継続出願である。特許出願第 12/571,018 号は、本出願の譲受人へと譲渡されており、ここでの言及によってあたかも全体が本明細書に記載されているかのように本出願に援用される。本出願は、米国特許出願第 12/571,018 号について米国特許法第 119 条(e)に規定の優先権を主張する。

【技術分野】

【0002】

本開示は、伝達可能なアイテムの正順(in-order)デリバリを特徴とするネットワークによって互いに接続されたマルチコンポーネントシステムのコンポーネント間の通信のための方法およびシステムに関する。

【背景技術】

【0003】

現在開発されている高度に統合されたシステムにおいては、高帯域幅の通信能力が、性能の要件としての前提条件である。さらに、セカンドソースの原理を追求するシステム開発者は、任意のあらゆる製造者から自身の設計のコンポーネントを選択できなければならないと同時に、それらがいかなる問題もなく相互運用されることを必要とする。このことが、該当の分野において活動する複数の製造者によって設立され、コンポーネントおよびコンポーネント間の相互通信についての規格を定める標準化団体の形成につながる。そのような標準化団体の 1 つが、モバイル・インダストリ・プロセッサ・インターフェイス・アライアンス(MIPI (登録商標))である。現時点において、この団体は、モバイルシステムの相互通信の詳細について作業する約 150 の製造者の集まりである。ある程度の情報が、ワールドワイドウェブ上の <http://www.mipi.org/> において入手可能である。

【0004】

コンポーネント間の通信を標準化するために、MIPI (登録商標) アライアンスは、

10

20

30

40

50

モバイルシステムにおいてデバイスを接続するためのシリアル高速リンクとして、UniProSMを定めている。UniProSM規格は、着実に発展中であり、現時点において規格バージョン1.0がリリースされている。この規格の種々のバージョンの特徴についてのある程度の情報が、ワールドワイドウェブ上のインターネット百科事典(<http://en.wikipedia.org/wiki/UniPro>)において入手可能である。

【0005】

相互接続および通信の規格を用意することで、製造者は、自身のシステムの開発においてはるかに柔軟になることができ、種々の要件に良好に適し、種々のベンダによって提供されるコンポーネントを混ぜ合わせ、適合させることができる。UniPro規格またはユニファイドプロトコルは、高速シリアルリンクを利用するチップ対チップのネットワークに向けられている。エラーハンドリング、フロー制御、ルーティング、およびアービトレーションなどの一般的な相互接続の問題を解決する汎用の通信プロトコルとなるように定められている。

【0006】

現時点において、UniProは、接続の確立と同時に、状態およびバッファなどの他のリソースの割り当てを必要とする接続指向の通信を提供している。通常は、接続が、通信に関わるバッファのオーバフローを防止するための、クレジット・エンド・トゥ・エンド・フロー制御を実装する。これが、データの喪失または破損がないことを保証する信頼できるネットワークの使用との組み合わせにおいて、ユーザへの信頼できる通信サービスを保証する。

【0007】

UniProの将来の発展において、リアルタイムのトラフィッククラスを提供するバージョンを予測することができるが、レイヤ2の再送信の数を抑える結果となり、データのデリバリそのものについての保証を犠牲にすることによって、パケットのデリバリに関する制限時間を保証できる。なぜならば、レイヤ2の再送信の数を抑えることで、データの断片がデリバリされない可能性をきわめて小さくできるからである。UniProアプリケーションの上位のレイヤが、該当のレポートの受信時に、欠けた断片を処理しなければならない。接続指向の通信にもとづく信頼できるリアルタイムのトラフィッククラスは、接続を開始し、維持し、終了させるためのプロトコルを必要とする。現時点において、スリーウェイハンドシェイクが、伝送制御プロトコルTCPから知られている。詳細は、the transmission control protocol、DARPA Internet program、protocol specification by Information Sciences Institute、University of Southern California、IETF request for comments #793、September 1981に公開されている。しかしながら、TCPは、ネットワークにつきものの低い信頼性に対処しなければならず、したがってそのような不確実性に対処するための予防措置をプロトコルにおいて講じる必要があるため、ユニファイドプロトコルとは大きく異なる。さらに、TCPは、例えばパケットがネットワークにおいてさまざまな経路をとることができるため、データのデリバリに順番がないことを前提としている。したがって、TCPは、接続の管理を保証するために、きわめて大きな続き番号および最大パケット寿命を使用する。しかしながら、UniProは、典型的には最大でも10のノードからなる小さなネットワークにおいて働き、順番どおりの通信を提供する。したがって、使用されるオーバーヘッドを通常のプロトコルよりも少なくすることができ、機能を犠牲にすることなく単純さを達成することができる。

【0008】

さらに、B-ISDN application protocols for access signalling, ITU-T recommendation Q.2931, Feb-1995においてITU-T Q.2931に開示されているAT

M接続設定も当分野において知られている。関連の接続設定は、A T Mにおいてはリファレンスと呼ばれる続き番号を使用するT C Pにおいて使用される機構と同様の機構を使用するが、A T Mも、メッセージのオーバーヘッドを生じさせ、したがって通信チャネルからの帯域幅を必要とする大きな続き番号にもとづいている。

【発明の概要】

【発明が解決しようとする課題】

【0009】

本開示の目的は、接続指向の正順デリバリ環境において接続を管理するための代案の方法およびシステムであって、最小限のメッセージのオーバーヘッドでリソースの適切な割り当てならびに接続の確立および終了を可能にする方法およびシステムを提供することにある。

10

【課題を解決するための手段】

【0010】

この課題は、接続指向の正順デリバリ環境において接続を管理するための方法であって、下記によって解決される。

a) クライアントが第1の種類のメッセージ(1140)をサーバへと送信することによって、接続の確立を要求し、

b) サーバが、サーバの接続につながる第2の種類のメッセージをクライアントへと接続することによって接続の確立が可能であることを確認し、

c) 第1の種類のメッセージを送信することによって、第1の最大応答時間としての第1の所定の期間を測定する第1のクライアントタイマを始動し、第2の種類のメッセージまたはデータメッセージを受信することによって、第1のクライアントタイマが停止され、

20

d) 第3のメッセージを送信することによって接続が閉じられる。

【0011】

さらに本開示は、接続指向の正順デリバリ環境において接続を管理するためのシステムであって、

サーバと、クライアントと、正順デリバリのネットワークと、を備えており、サーバが、上述の方法のうちのサーバに関する動作のいずれかを実行するように構成され、クライアントが、上述の方法のうちのクライアントに関する動作のいずれかを実行するように構成されているシステムを提供する。

30

【0012】

本開示は、サーバ、クライアント、ルータなどにおいて本開示の方法のいずれかを実行するためのコンピュータプログラム製品を含む。

【0013】

さらに本開示は、上述のコンピュータプログラム製品を保存する機械によって読み取ることができる一時的でない信号記憶媒体を含む。例えば、機械によって読み取ることができる一時的でない信号記憶媒体は、C D R O MまたはD V D R O Mなどの光ディスク、磁気テープメモリ、ハードディスクまたはディスクレットなどの磁気ディスクメモリ、U S Bメモリスティックまたはフラッシュメモリなどの半導体メモリなどであってよい。

40

【0014】

本開示の好都合なさらなる実施の形態が、従属請求項に提示される。

【0015】

好都合なことに、本開示による方法は、メッセージの数を最小限にし、サーバによる第1のメッセージの脱落を防止する。このやり方で、たとえ信頼できるネットワークを使用している場合でも、本方法は、別の接続を処理している等により接続に対応するための利用可能なリソースを有さないビジーサーバを保護する。

【0016】

好都合には、本開示による方法のさらなる実施の形態によれば、可能な限り短い時間での接続の確立を保証すると同時に、サーバに十分な処理時間を許しつつ最初の送信におけ

50

る第1の種類のメッセージのサーバにおける脱落への対処を保証するために、第1の所定の期間が過ぎた場合にもう1つの第1の種類のメッセージがサーバへと送信される。

【0017】

好都合には、本開示による方法のさらなる実施の形態によれば、第2の種類のメッセージの受信がクライアントの接続につながる一方で、第4の種類のメッセージの送信が、本方法が信頼性に欠けるネットワークまたはリアルタイムのトラフィッククラスに対応して、確認メッセージが受信された旨をサーバへと知らせることを可能にする。

【0018】

有益には、サーバからクライアントへの第2の種類のメッセージの形式の本開示の方法のさらなる実施の形態による確認メッセージが、このメッセージの喪失から接続の確立を保護するために、第2の所定の期間を測定する第1のサーバタイマによって守られる一方で、確認メッセージがサーバから受信された旨のクライアントからの確認としての第4の種類のメッセージの受信が、第1のサーバタイマを停止させる。このやり方で、接続の確実な確立が、最小限の数のメッセージで保証される。

【0019】

好都合には、本開示による方法のさらなる実施の形態によれば、他の接続への対応または他のアプリケーションの高い処理付加に起因してリソースが利用不可能であるがゆえに接続の確立が不可能である場合に、サーバが第5の種類のメッセージを生成し、クライアントへと送信する。このメッセージが、クライアントにおいて受信されたときに、クライアントが接続を閉じることを好都合に可能にし、サーバが利用不可能であるときにクライアントを所定の状態にする方法を提供する。

【0020】

好都合には、本開示による方法のさらなる実施の形態によれば、第3の所定の期間を測定する第2のクライアントタイマが、第3の種類のメッセージが送信されたときに開始されることで、第3の種類のメッセージへの応答が第3の所定の期間内に受信されない場合に、接続の所定の終了が保証され、適切な対策がとられる。

【0021】

有益には、本開示による方法のさらなる実施の形態によれば、第3の種類のメッセージの受信を確認し、したがって接続の確立および終了について綿密に構造化された取り扱いおよび管理が保証されていることを確認するために、サーバが第3の種類のメッセージの受信時に第6の種類のメッセージをクライアントへと送信する。

【0022】

好都合には、本開示による方法のさらなる実施の形態によれば、信頼性に欠けるネットワークまたはリアルタイム通信のためのトラフィッククラスを有するネットワークがメッセージの伝達に使用される場合に、クライアントとの確実な通信を守るために、第6の種類のメッセージの送信によって第2のサーバタイマが開始される。

【0023】

有益には、本開示の方法のさらなる実施の形態によれば、ひとたびクライアントが第6の種類のメッセージを受信し、したがってサーバが接続を閉じている旨を知ると、第2のクライアントタイマが停止されると同時に、この第6の種類のメッセージの受信を確認するため、ならびにサーバおよびクライアントにおける所定の状態を保証するために、第7の種類のメッセージがサーバへと送信される。

【0024】

好都合には、本開示の方法のさらなる実施の形態によれば、クライアントおよびサーバの間の通信の柔軟性を高めると同時に、用途またはトラフィッククラスに応じて接続のために要求される帯域幅を定めることができるよう、クライアントおよびサーバの間で交換される任意のメッセージが、ルータを通過することができる。同時に、ルータは、受信したメッセージを、要求された帯域幅が利用できる場合に限って転送する。

【0025】

好都合には、本開示による方法のさらなる実施の形態によれば、ルータが要求された帯

10

20

30

40

50

域幅の提供に利用できる場合、クライアントによって指定された (addressed) サーバからの確認メッセージが受信されるまで、他のサーバおよびクライアントからの通信を拒絶しつつ、さらなる帯域幅の割り当てから保護する。このやり方で、帯域幅の割り当てが、存在しうる他の通信相手によるアクセスから保護される。

【0026】

好都合には、本開示による方法のさらなる実施の形態は、アプリケーションクライアントとアプリケーションサーバとの間に確立される接続を介してデータの交換を提供するために、クライアントがクライアントにおいてメッセージを開始させるアプリケーションクライアントによって対処される一方で、サーバがアプリケーションサーバと通信することで、通信の確立のためのメッセージを交換するクライアントおよびサーバによってアプリケーションサーバとアプリケーションクライアントとの間の通信の確立を可能にする。

10

【0027】

好都合には、本開示によるシステムは、本開示による方法の各機能を実行するために、最小限の構成においてサーバ、クライアント、および正順デリバリのネットワークを提供する。

【0028】

好都合には、本開示によるシステムのさらなる実施の形態は、本開示による方法の実施の形態を利用しつつ、柔軟性を高め、クライアントとサーバとの間の通信距離を伸ばすためのルータを提供する。

【0029】

好都合には、本開示のシステムのさらなる実施の形態によれば、通信の際に同時には使用されない確認メッセージが、1つのフォーマットのただ1つのメッセージへと節約され、通信の文脈においてサーバまたはクライアントにおいて正しい機能をもたらす。同様に、同時に動作することがないサーバタイマおよびクライアントタイマも、必要なときに作動させられ、本開示およびその実施の形態の方法によるメッセージ交換において第1または第2のサーバまたはクライアントタイマを実現するただ1つのサーバタイマおよびただ1つのクライアントタイマとして実現することができる。

20

【0030】

以下で、本開示の実施の形態の例を、図面に示した実施例にもとづいてさらに説明する。

30

【図面の簡単な説明】

【0031】

【図1】信頼できるネットワークにおいて生じる典型的なメッセージ交換を示している。

【図2】信頼できるネットワークにおける状態および状態変化の例を説明する状態機械を示している。

【図3】信頼できるネットワークにおけるサーバの状態および状態変化の例を示している。

。

【図4】信頼性に欠けるネットワークにおける例としてのメッセージの流れを示している。

。

【図5】信頼性に欠けるネットワークにおけるクライアントの例としての状態および状態変化を示している。

40

【図6】信頼性に欠けるネットワークにおけるサーバの状態および状態変化の例を示している。

【図7】帯域幅の割り当てを含むサーバ、クライアント、およびルータ間のメッセージの流れの例を示している。

【図8】メッセージがルータを通過するときの信頼性に欠けるネットワークにおけるクライアントとサーバとの間のメッセージの流れの例を示している。

【図9】メッセージの流れにルータを含んでいる信頼性に欠けるネットワークにおけるクライアントの例としての状態および状態変化を示している。

【図10】メッセージの流れにルータを含んでいる信頼性に欠けるネットワークにおける

50

例としてのサーバの状態および状態変化を示している。

【図 1 1】信頼性に欠けるネットワークにおけるクライアントとサーバとの間のメッセージの流れに含まれるルータの例としての状態および状態変化を示している。

【図 1 2】サーバがビジーであるときの信頼性に欠けるネットワークにおけるクライアントとサーバとの間のメッセージ交換の例を示している。

【図 1 3】第 1 の種類のメッセージが失われる場合のクライアントとサーバとの間のメッセージ交換の例を示している。

【図 1 4】第 2 の種類のメッセージが失われる場合のクライアントとサーバとの間のメッセージ交換の例を示している。

【図 1 5】サーバアプリケーションがクラッシュした場合のクライアントとサーバとの間のメッセージの流れの例を示している。

【図 1 6】サーバがビジーである場合の信頼性に欠けるネットワークにおけるクライアントとサーバとの間のメッセージの流れの例を示している。

【図 1 7】プロトコルエラーが生じる場合のクライアントとサーバとの間のメッセージ交換の例を示している。

【図 1 8】ルータを含む信頼性に欠けるネットワークにおいて生じる第 1 のエラーの例を示している。

【図 1 9】経路にルータを含んでいるクライアントとサーバとの間の通信において生じる例としての第 2 のエラーを示している。

【図 2 0】間にルータを有しているクライアントとサーバとの間で生じる通信エラーの第 3 の例を示している。

【図 2 1】ルータを通過するメッセージ交換に関し、クライアントとサーバとの間の通信において生じる通信エラーの第 4 の例を示している。

【図 2 2】信頼性に欠けるネットワークにおける例としてのさらなるメッセージの流れを示している。

【図 2 3】信頼性に欠けるネットワークにおける例としての別のメッセージの流れを示している。

【発明を実施するための形態】

【0032】

本開示を、特定の実施の形態に関して、特定の図面を参照しつつ説明するが、本開示は、これらに限られるわけではなく、特許請求の範囲以外の何ものにも限定されない。説明される図面は、あくまでも概略図であって、本開示を限定するものではない。図面においては、説明の目的のために、一部の構成要素のサイズが誇張され、比例尺では描かれていない。

【0033】

効率をよくする目的で、図 2、図 3、図 5、図 6、図 9、図 10、および図 11 のような状態機械に関する図面の説明の全体を通じて、状態変化のトリガおよびそれによって生じる事象を説明するために、以下の文法的規則が使用される。

【0034】

<トリガ>/<アクション>などの書式が、状態機械における表記に使用される。ここで、<トリガ>は、対応するトランザクションへとつながるトリガとして機能する入力トリガのプレースホルダとして機能する。さらに、<アクション>は、トランザクションに関連付けられた一式の結果としての事象のプレースホルダとして機能する。

【0035】

本明細書および特許請求の範囲において、用語「・・・からなる (comprising)」が使用される場合、他の構成要素または工程が除外されるわけではない。さらに、本明細書および特許請求の範囲において、「第 1 の」、「第 2 の」、「第 3 の」、などといった用語は、類似の構成要素間の区別のために用いられ、必ずしも順番または時系列を表わすために使用されているわけではない。そのように使用される用語が、適切な状況のもとで互いに入れ換え可能であり、本明細書に記載される本開示の実施の形態が、本明細

10

20

30

40

50

書に記載または例示の順序以外の順序でも機能できることを、理解すべきである。

【0036】

図1が、信頼できるネットワークにおける本開示の一実施の形態による接続管理のためのメッセージの流れの例を示している。図面の説明の全体を通して、同じ参照符号が、すべての図面において、同じエンティティに使用され、その繰り返しの説明は、効率をよくする目的で省略される。

【0037】

図1に例示されるとおり、アプリケーションクライアント1000は、信頼できるネットワークを使用して、クライアント1100、サーバ1200、およびアプリケーションサーバ1300と通信する。ネットワークは、単純なリンクであってよく、あるいは1つ以上のルータを含むことができる。しかしながら、簡単にするために、図1においてはいかなるルータも示されていない。アプリケーションクライアント1000は、接続の確立を開始するために、「C__Closed」の状態1130にあるクライアント1100へ、メッセージ1010「T__OPEN. req」を送信する。クライアント1100において、第1の種類のメッセージ1140「T__SYN」が「S__Listen」状態にあるサーバ1200へ送信された時点で、第1のタイマが開始される。サーバはアプリケーションサーバ1300へのメッセージ「T__OPEN. ind」1310を生成し、アプリケーションサーバ1300がデータを取り扱うことができる場合には、サーバへのメッセージ1320「T__OPEN. rsp」にて応答する。サーバ1200は、状態1220「S__WaitRsp」にあるときに第1のサーバタイマを用いて、メッセージ1310および1320の間の時間間隔を測定する。メッセージ1320を受信すると、第2の所定の期間を測定する第1のサーバタイマが停止され、第2の種類のメッセージ1150「T__SYN__Ack」がクライアント1100へと送信され、クライアント1100において、メッセージ1140が生成されて状態1120「C__WaitAck」となったときに開始された第1のクライアントタイマが停止され、メッセージ1020「T__OPEN. cnf」が生成される。今や状態1230「S__Connected」にあるサーバ1200に接続が行われる。今やアプリケーションクライアントがデータ送信の要求1030「T__DATA. req」の送信を開始し、これが確認メッセージ1040「T__DATA. cnfL」によってクライアント1100によって確認される。サーバを「S__Connected」の状態に、クライアントを「T__Connected」の状態に残しつつ、サーバ1200は送信されたデータ1160「T__DATA」を受信し、アプリケーションクライアントからのデータ1330「T__DATA. ind」をアプリケーションサーバへと送信し、アプリケーションサーバはメッセージ1340「T__DATA. rspL」によって応答する。

【0038】

データをメッセージ1350「T__DATA. req」をサーバ1200へと送信することによってアプリケーションサーバから要求することもでき、このメッセージはメッセージ1160「T__DATA」としてクライアント1100へと転送され、そこからメッセージ「T__DATA. ind」1050としてアプリケーションクライアントへとさらに転送され、「C__Connected」の状態1110にあるクライアント1100へのメッセージ1060「T__DATA. rspL」によって応答するよう、アプリケーションクライアントを促す。

【0039】

接続を終わらせるために、アプリケーションクライアントはメッセージ1070「T__CLOSE. req」を送信し、これがクライアント1100によってメッセージ1080「T__CLOSE. cnfL」にて確認されるとともに、クライアント1100はサーバ1200への第3の種類のメッセージ1170「T__FIN」を生成し、サーバ1200はアプリケーション1300へと接続閉鎖1370「T__CLOSE. ind」を知らせることによって応答し、これがサーバ1300によってメッセージ1380「T__CLOSE. rspL」にて確認され、サーバを状態1240「S__Listen」に移行さ

せる。あるいは、接続の閉鎖がメッセージ1070「T_CLOSE.req」にてアプリケーションサーバ1300によって要求され（メッセージ1080「T_CLOSE.cnfL」にて確認される）、メッセージ1370「T_CLOSE.ind」にて示される（メッセージ1380「T_CLOSE.rspL」にて応答される）場合に、メッセージ1170「T_FIN」を送信することによってサーバ1200により、接続が閉鎖されてもよい。一般に、これらの実施の形態において説明したそれぞれのメッセージの流れを、サーバおよびクライアントに関して逆にすることができ、すなわちサーバおよびクライアントを入れ替えることができる。

【0040】

クライアントは状態1130「C_Closed」のままである。たとえ信頼できるネットワークが通信の基盤を形成している場合であっても、メッセージの処理に利用できるリソースが存在しない場合に、第1の種類のメッセージ1140がサーバによって落とされる可能性があるため、メッセージ1140および1150の間のタイマが必要である。随意により、メッセージ1310および1320の間でサーバによって運営されて第2の所定の期間を測定するタイマが、通信に必要とされるアプリケーションサーバのアプリケーションがクラッシュまたは他の理由で利用可能になっていないかどうかを監視するために、利用可能である。

【0041】

便宜上、クライアント1100は、メッセージ1140の送信後に到着する傾向にあるメッセージ1150の受信のためのリソースを提供すべきである。このやり方で、好都合なことに、サーバ1200において、メッセージ1150の適切な伝送を監視するためのタイマが不要である。同様に、クライアント1100および／またはサーバ1200は、メッセージ1170の受信／取り扱いのためのリソースを提供すべきである。このやり方で、メッセージ1170の送信者はメッセージ1170の伝送損失に対する保護のためのタイマを必要としない。

【0042】

図2は、図1に示したメッセージの流れに関係する状態機械を、クライアントがとることができる状態および状態変化の例として示している。

【0043】

本明細書における状態機械の表記をさらに説明するために、例が図2を参照して提示されるが、これは、状態機械を表わす他の図にも同様のやり方で適用可能である。

【0044】

ここで、図2において、状態1130から状態1120へのクライアントの移行2100は、アプリケーションクライアント1000からのT_OPEN.req(my_server)の受信によって引き起こされ、サーバ1200へのT_SYN(to:my_server)の送信およびクライアント1100におけるTimer_SYNの開始にもつながる。これは、上述の表記において、2100:T_OPEN.req(my_server)/T_SYN(to:my_server), start Timer_SYNと称される。

【0045】

他の例として、移行2200のトリガの事象は、Timer_SYNの時間切れであり、やはりサーバ1200へのT_SYN(to:my_server)の送信およびクライアント1100におけるTimer_SYNの再開につながる。これが、2200:timeout Timer_SYN/T_SYN(to:my_server), restart Timer_SYNと称される。

【0046】

状態1130にあるとき、クライアントは、2100:T_OPEN.req(my_server)/T_SYN(to:my_server), start Timer_SYNに対応して、状態1120「C_WaitSynAck」へと移行することができる。クライアントは、2200:timeout Timer_SYN/T_SYN(t

10

20

30

40

50

o:my_server), restart Timer_SYNに対応して状態1120のままである一方で、状態1130への再移行が、2150:T_SYN_ERR(from:my_server)/T_OPEN.cnf(error), stop Timer_SYNに従って生じる。状態1120から状態1110「C_Connected」への移行が、2300:T_SYN_ACK(from:my_server)/T_OPEN.cnf(ok), stop Timer_SYNまたは2300:T_DATA(from:my_server)/T_OPEN.cnf(ok), stop Timer_SYN, T_DATA.indに従って生じる一方で、クライアントは、2400:T_DATA.req/T_DATA(to:my_server)または2400:T_DATA(from:my_server)/T_DATA.indに対応して状態1110にとどまり、2500:T_CLOSE.req()/T_CLOSE.cnf, T_FIN(to:my_server)によって状態1130へと移行する。

【0047】

図3は、図1に示したメッセージの流れの文脈においてサーバがとることができる状態および状態変化の例としての状態機械を示している。

【0048】

状態「S_WaitCloseRspE」3220、状態3420「S_Error」、状態1220「S_WaitOpenRsp」、状態1230「S_Connected」、ならびにさらなる状態3320「S_WaitCloseRsp」および状態1210「S_Listen」が示されている。

【0049】

状態1210から状態1220への移行は、3350:T_SYN(from:my_client)/T_OPEN.ind(my_client), start Timer_Rspによって生じ、状態1220から3420への移行は、3200:timeout Timer_Rsp/stop Timer_Rsp, T_SYN_ERR(to:my_client)によって生じ、状態1220から状態1230への移行は、3300:T_OPEN.rsp()/stop Timer_Rsp, T_SYN_ACK(to:my_client)に対応して生じる。状態3420から状態3220への状態変化は、3100:T_OPEN.rsp()/T_CLOSE.ind()に従って生じる。

【0050】

さらなる移行が、3450:T_FIN(from:my_client)/T_CLOSE.ind()に対応して状態1230から3320へと生じ、3550:T_CLOSE.rsp/T_FIN(to:my_client)によって状態3320から状態1210へと生じる。状態1210は、最初に事象3650:T_LISTEN.req/-によってトリガされる。サーバは、移行3150:T_SYN(from:any_client)/T_SYN_ERR(to:any_client)に従って状態3420にとどまる。サーバは、移行3400:T_DATA.req(data)/T_DATA(to:my_client, data)または3400:T_DATA(from:my_client, data)/T_DATA.ind(data)または3400:T_SYN(from:my_client)/T_SYN_ACK(to:my_client)または3400:T_SYN(from:other_client)/T_SYN_ERR(to:other_client)に従って状態1230にとどまる。サーバは、移行3500:T_DATA.req(data)/T_DATA(to:my_client, data)または3500:T_SYN(from:other_client)/T_SYN_ERR(to:other_client)に従って状態3320にとどまる。

【0051】

例えば、状態3320において、サーバは依然としてデータを送信することができるが

、クライアントからそれ以上のデータは受信しない。

【0052】

図4は、レイヤ2の再送信の数に制限があるネットワークなど、信頼性を欠くネットワークにおいて生じる本開示による実施の形態のメッセージの流れの例を示している。

【0053】

図1との比較において容易に特定できるとおり、交換されるメッセージの多くは同じである。この事実ゆえ、信頼できるネットワークにおいて必要とされるメッセージを、信頼性に欠けるネットワークにおける接続の確立に必要なメッセージから隔てる、メッセージの流れにおける相違に注目する。信頼性に欠けるネットワークの場合には、好ましくは、追加のタイマおよびメッセージが、信頼性の欠如を補償し、かつ、接続の確立を確実にするために、用意される。この場合、クライアント1100が、第4の種類のメッセージ4150「T__ACK__ACK」を生成する。この第4の種類のメッセージは、信頼性に欠けるネットワークの場合に用意され、第2の種類のメッセージ1150の適切な伝達を監視するためのメッセージの流れを実行する。第4の種類のメッセージの適切かつ時機を得た伝達は、サーバ1200が状態7510「S__WaitSynAck」にある間に、サーバ1200において動作する第2のサーバタイマによって判断される。この場合には、クライアントが、第4の種類のメッセージ4150の送信後に状態4110「C__Connected」に移行する。

10

【0054】

このメッセージの流れの他の特徴は、接続の終了にあり、接続の終了の際の第3のメッセージ1170への適切な応答が、クライアント自身が状態4110「C__WaitFinAck」にあるときにタイマによって監視される。ひとたび接続の終了を確認する第6の種類のメッセージ4170「T__FIN__ACK」がサーバ1200から受信されると、第2のクライアントタイマが停止される。

20

【0055】

信頼性に欠けるネットワークにもとづくメッセージの伝達の場合には、リンクがパケットを破棄する可能性がある。これは、メッセージのデリバリ時間を拘束する、例えば0または1の再送信の境界数を有するトラフィッククラスの場合かもしれない。この手順ゆえ、メッセージの一部分または全体が失われることがあり、あるいはメッセージが、ペイロードに既知のエラーがある状態で、届けられることがある。したがって、接続管理メッセージも、処理されずに捨てられる可能性がある。ここで、特に、サーバからの第2の種類のメッセージ1150を、好ましくは保護すべきである。なぜならば、喪失ゆえのエラーの場合、サーバが最初にビジーであり、そして、第2の種類のメッセージ1150を利用できるようになった後に、「S__Connected」の状態1230のままになる可能性があるからである。

30

【0056】

これは、サーバが接続されていると思い込み、接続されておらず、したがってデータの受信の準備ができていないクライアントへ、データの送信を開始するという事例につながりかねない。

【0057】

図5は、図4に示したメッセージの流れに関連したクライアントにおける状態および状態間の移行の例を示す状態機械を示している。

40

【0058】

この場合、この実施の形態に示される、信頼性に欠けるネットワークに存在する状況が、図2に示した信頼できるネットワークにおけるクライアントと比較した相違点として、接続「C__WaitFinAck」の終了の文脈における状態4110の追加を示している。さらなる状態は、1110「C__Connected」、1130「C__Closed」、5120「C__WaitSynAck」、および5650「C__WaitCloseRsp」である。

【0059】

50

状態 1130 から状態 5120 の移行は、5150: T_OPEN.req(my_server) / T_SYN(to: my_server), start Timer_SYN に対応する。状態 5120 から状態 1110 への移行は、5250: T_SYN_ACK(from: my_server) / T_ACK_ACK(to: my_server), T_OPEN.cnf(ok), stop Timer_SYN に対応して生じる。状態 1110 から状態 4110 へは、移行 5450: T_CLOSE.req() / T_FIN(to: my_server), start Timer_FIN に対応する。5300: T_SYN_ERR(from: my_server) / T_FIN(to: my_server), start Timer_FIN に従って、状態 5120 から状態 4110 に達することも可能である。状態 4110 から状態 1130 は、移行 5550: T_FIN_ACK(from: my_server) / T_CLOSE.cnf(), stop Timer_FIN、移行 5550: T_FIN(from: my_server) / T_FIN_ACK(to: my_server), T_CLOSE.cnf(), stop Timer_FIN による。状態 1110 から状態 5420 は、移行 5460: T_FIN(from: my_client) / T_CLOSE.ind() による。状態 5420 から状態 1130 は、移行 5660: T_CLOSE.rsp() / T_FIN_ACK(to: my_client) による。

【0060】

移行 5100: T_FIN_ACK(from: any_server) / -、移行 5100: T_FIN(from: any_server) / T_FIN_ACK(to: any_client)、移行 5200: T_FIN_ACK(from: any_server) / -、移行 5200: T_FIN(from: any_server) / T_FIN_ACK(to: any_client)、移行 5200: timeout Timer_SYN / T_SYN(to: my_server), restart Timer_SYN、移行 5400: T_DATA.req / T_DATA(to: my_server)、移行 5400: T_DATA(from: my_server) / T_DATA.ind、移行 5400: T_FIN_ACK(from: other_server) / -、移行 5400: T_FIN(from: other_server) / T_FIN_ACK(to: other_client)、移行 5400: T_SYN_ACK(from: my_server) / T_ACK_ACK(to: my_server)、ならびに移行 5500: T_SYN_ACK(from: my_server) / T_FIN(to: my_server)、移行 5500: T_SYN_ERR(from: my_server) / T_FIN(to: my_server)、移行 5500: T_FIN_ACK(from: other_server) / -、移行 5500: T_FIN(from: other_server) / T_FIN_ACK(to: other_client)、移行 5500: timeout Timer_FIN / T_FIN(to: my_server), restart Timer_FIN、移行 5650: T_FIN_ACK(from: other_server) / -、および移行 5650: T_FIN(from: other_server) / T_FIN_ACK(to: other_client) のそれぞれの場合、それぞれの状態 1130、5120、1110、および 4110 が維持される。

【0061】

状態 5120 の場合において、第 2 の種類のメッセージ 1150、あるいはエラーメッセージが受信される場合、それらは第 3 の種類のメッセージ 1170 によって応答される。この場合、第 2 のクライアントタイマが再開される。この場合には、両方のクライアントタイマが、例えば相互に排他的であり、したがって両方の機能をとる単一のタイマとして実現することができる。

【0062】

図 6 が、図 4 に示したとおりの信頼性に欠けるネットワークの通信状況におけるサーバ 1200 の状態および状態間の移行の例を示す状態機械を示している。

10

20

30

40

50

【0063】

ここで、信頼できるネットワークを利用するサーバの状況との比較において、やはり第4の種類のメッセージ4150がクライアント1100からの確認として受信される。同様に、第6の種類のメッセージ4170が発せられる。

【0064】

図6において、状態6550「S_Error」および状態1240「S_Listen」ならびに状態6220「S_WaitOpenRsp」および状態6620「S_WaitSynAck」とともに、状態6520「S_WaitCloseRspE」が示されている。さらに、状態1230「S_Connected」、状態6420「S_WaitCloseRsp」、および状態4112「S_WaitFinAck」も示されている。

10

【0065】

状態1240は、6800:T_LISTEN.req/ーによって開始され、6750:T_FIN(from:any_client)/T_FIN_ACK(to:any_client)または6750:T_FIN_ACK(from:any_client)/ーの場合に維持される。そこからの状態6220への移行は、6850:T_SYN(from:my_client)/T_OPEN.ind(my_client), start_Timer_Rspに従って生じ、状態6220は、6300:T_FIN(from:other_client)/T_FIN_ACK(to:other_client)または6300:T_FIN_ACK(from:other_client)/ーの場合に維持される。そこから、6250:timeout_Timer_Rsp/stop_Timer_Rsp, T_SYN_ERR(to:my_client)の場合に状態6550に到達でき、状態6550は、6150:T_SYN(from:any_client)/T_SYN_ERR(to:any_client)、6150:T_FIN(from:any_client)/T_FIN_ACK(to:any_client)、または6150:T_FIN_ACK(from:any_client)/ーに対応して維持される。この状態6550から、6200:T_OPEN.rsp()/T_CLOSE.ind()によって状態6520に到達でき、状態6520は、6100:T_SYN(from:any_client)/T_SYN_ERR(to:any_client)、6100:T_FIN(from:any_client)/T_FIN_ACK(to:any_client)、または6100:T_FIN_ACK(from:any_client)/ーに従って維持される。

20

30

【0066】

状態6220から状態6620への別の移行が、6350:T_OPEN.rsp()/stop_Timer_Rsp, T_SYN_ACK(to:my_client), start_Timer_ACKに対応して生じる。それぞれの状態が、移行6400:T_SYN(from:my_client)/T_SYN_ACK(to:my_client)、移行6400:T_SYN(from:other_client)/T_SYN_ERR(to:other_client)、移行6400:T_FIN(from:other_client)/T_FIN_ACK(to:other_client)、移行6400:T_FIN_ACK(from:other_client)/ー、または移行6400:timeout_Timer_ACK/T_SYN_ACK(to:my_client), restart_Timer_ACKに対応して維持され、ここから6450:T_ACK_ACK(from:my_client)/stop_Timer_ACKまたは6450:T_DATA(from:my_client, data)/stop_Timer_ACK, T_DATA.ind(data)に従って状態1230が到達され、状態1230が、トランザクション6555:T_DATA.req(data)/T_DATA(to:my_client, data)、トランザクション6555:T_DATA(from:my_client, data)/T_DATA.ind(data)、トランザクション6555:T_SYN_ACK(fr

40

50

om:other_client) / -、トランザクション6555:T_SYN(from:other_client) / T_SYN_ERR(to:other_client)、トランザクション6555:T_FIN(from:other_client) / T_FIN_ACK(to:other_client)、または移行6555:T_FIN_ACK(from:other_client) / -によって維持される。ここから、状態6420が6600:T_FIN(from:my_client) / T_CLOSE.ind()に対応してとられ、移行6650:T_DATA.req(data) / T_DATA(to:my_client, data)、移行6650:T_FIN(from:my_client) / -、移行6650:T_SYN(from:other_client) / T_SYN_ERR(to:other_client)、
10 移行6650:T_FIN(from:other_client) / T_FIN_ACK(to:other_client)、または移行6650:T_FIN_ACK(from:other_client) / -に従って維持される。次いで、6700:T_CLOSE.rsp / T_FIN_ACK(to:my_client)に対応する初期状態1240への移行が生じることで、この状態を離れることができる。

【0067】

状態6555から状態6640への移行は、6610:T_CLOSE.req() / T_FIN(to:my_client), start Timer_FINに対応する。状態4112から状態1240への移行は、6645:T_FIN_ACK(from:my_client) / T_CLOSE.cnf(), stop Timer_FIN
20 および6645:T_FIN(from:my_client) / T_FIN_ACK(to:my_client), T_CLOSE.cnf(), stop Timer_FINに対応する。状態4112は、移行6640:T_ACK_ACK(from:my_client) / T_FIN(to:my_client)、移行6640:T_SYN(from:other_client) / T_SYN_ERR(to:other_client)、移行6640:T_FIN(from:other_client) / T_FIN_ACK(to:other_client)、移行6640:T_FIN_ACK(from:other_client) / -、timeout Timer_FIN / T_FIN(to:my_client), restart Timer_FIN
30、またはT_DATA(from:my_client, data) / -に従って維持される。

【0068】

状態6620から状態1240への他の移行は、6500:T_FIN(from:my_client) / T_FIN_ACK(to:my_client)に対応して生じる。

【0069】

図において、用語「my_client」がアプリケーションクライアントを指す一方で、用語「my_server」はアプリケーションサーバを指す。

【0070】

図7が、クライアントとサーバとの間で交換されるメッセージがルータによって転送される本開示による方法の実施の形態の例を示している。この方法は、接続の確立の前に、接続のための帯域幅の予約および利用可能な帯域幅の確認を、好都合に適用する。このメッセージの図と、図4の信頼できるネットワークにおける接続管理を示したメッセージの流れを示す図との間の主たる相違点は、ルータがクライアント1100とサーバ1200との間に存在する場合に、ルータが図4においてクライアント1100とサーバ1200との間で交換されるメッセージを転送し、好都合には帯域幅の割り当てまたは帯域幅の見積りを実行し、それぞれの接続に要求される帯域幅の利用可能性を確認することにある。詳しくは、ここではルータ7500が新規であり、それぞれの状態7510「R_Ready」および7530「R_Busy」をとる。さらに、先の図において説明したメッセージ書式と比較して、第1の種類のメッセージのメッセージ書式が帯域幅の要求を含んで
40
50

おり、したがって参照番号7140「T__SYN(bw)」によって指し示されている。また、第3の種類のメッセージのメッセージ書式が、今や好ましくは帯域幅の要求を含んでおり、したがって別の参照番号7170「T__FIN(bw)」によって指し示されている。ここで、クライアント1000とサーバ1100との間のルータ7500が、例えば1または複数の帯域幅パラメータを使用し、リンクの帯域幅の一部がすでに他の接続のために予約されている可能性に鑑みて、要求された帯域幅がリンクの能力に適合するか否かを見積り、帯域幅の予約に成功した場合に、第1の種類のメッセージを転送し、そうでない場合には、例えばクライアント1000またはサーバ1100へと通信されるエラーを生成する。さらに、好都合には、ルータが、ルータがビジーであることを示す状態7530に入り、この状態においては、他の接続のための他の帯域幅の予約の要求を受け付けない。

10

【0071】

ひとたび、ルータが第2の種類のメッセージ（同一ペアのクライアントおよびサーバに対して、サーバ1100からクライアント1000へ転送される確認であって、すべての含まれるルータが要求された帯域幅の予約に成功したことを確認するもの）を識別すると、レディ状態7510へと戻る。「R__Busy」状態7530は、例えば第1の種類のメッセージ7140について生じうる再送信を排除するために必要とされ、ルータにおける帯域幅の予約の二重の更新を好都合に防止する。さらに、先の図との比較において、接続は第3の種類のメッセージ7170によって終了させられる。ここで、第3の種類のメッセージは、例えばクライアント1000および第3の種類のメッセージ7170を発するサーバ1100に保存された第1の種類のメッセージ7140と同じ帯域幅パラメータを含んでいる。第3の種類のメッセージ7170の受信の結果として、ルータ7500は、自身の予約済みの帯域幅を減らす。メッセージの交換が信頼できるネットワークにもとづいて行われる場合、この動作が失敗する可能性はない。したがって第3の種類のメッセージ7170が失われる可能性はなく、したがってこの場合には、好ましくはこのメッセージの適切な取り扱いを確認するためのタイマは不要である。ルータにおける見積りのプロセスおよび保存のプロセスが、四角い囲み7520によって示されている。

20

【0072】

図8が、メッセージの交換が信頼性に欠けるネットワークまたはリアルタイムの通信のためのトラフィッククラスをサポートしているネットワークにおいて行われる場合の本開示による方法の実施の形態のメッセージの流れの別の例を示している。

30

【0073】

図4に示したメッセージの流れと比べて、やはり先のメッセージの流れと同様に、このメッセージの流れは、クライアント1000とサーバ1100との間で交換されるメッセージを転送するためのルータ7500を備えている。すでに説明したように、信頼性に欠けるネットワークをサポートする図4におけるメッセージの流れをさらに詳しく述べるとき、好ましくはクライアント1000とサーバ1100との間で交換されるメッセージのより密接な監視およびルータ7500における予防措置が必要である。

【0074】

ここで、やはり第4の種類のメッセージ4150が、接続の終了の過程において必要とされる。理由は、3つのメッセージを必要とするあらゆるルータの更新の要件にある。3つのメッセージとは、第1の種類のメッセージ7140および第3の種類のメッセージ7170などの帯域幅を運ぶ通信を初期化する1つのメッセージと、メッセージ1150およびメッセージ8150「T__FIN__ACK」などの第1の種類のメッセージを確認する第2のメッセージと、メッセージ4150などの帯域幅の変更を委ねるための第3のメッセージと、である。この場合において、ルータがビジーであり、あるいは帯域幅の変更を支持できない場合に、ルータ7500は「R__Busy」を示す7530へと自身の状態を変化させ、空きの帯域幅が不十分である場合に、ルータはさらに後述されるように「S__Error」メッセージを生成する。ここで、帯域幅の割り当ておよび見積りは、ルータ7500における四角囲み8550によってさらに示されている。

40

50

【0075】

図9が、クライアント側から観察される図7のメッセージフローに関する状態機械の例として状態および状態変化を示している。

【0076】

ここで、以下の状態7120、1110、4110、および1130が可能である。

【0077】

状態1130から状態7120への状態移行が、9150:T_OPEN.req(my_server, bw) / conn_bw=bw, T_SYN(to:my_server, bw), start Timer_SYNに対応して生じる一方で、状態7120から状態4110への状態移行は、9300:T_SYN_ERR(from:my_server) / T_FIN(to:my_server, conn_bw)に対応して可能である。他方で、クライアントについて、状態7120から状態1110への移行が、9250:T_SYN_ACK(from:my_server) / T_ACK_ACK(to:my_server), T_OPEN.cnf(ok), stop Timer_SYNに従って生じる。おそらくは状態1110および状態4110の間のさらなる移行が、9400:T_CLOSE.req() / T_FIN(to:my_server, conn_bw), start Timer_FINに対応してさらに存在する。移行9100:T_FIN_ACK(from:any_server) / T_ACK_ACK(to:any_server)、移行9200:T_FIN_ACK(from:any_server) / T_ACK_ACK(to:any_server)、移行9200:timeout Timer_SYN / T_SYN(to:my_server, conn_bw), restart Timer_SYN、移行9350:T_DATA.req / T_DATA(to:my_server)、移行9350:T_DATA(from:my_server) / T_DATA.ind、移行9350:T_FIN_ACK(from:other_server) / T_ACK_ACK(to:other_server)、移行9350:T_SYN_ACK(from:my_server) / T_ACK_ACK(to:my_server)、および移行9450:T_SYN_ACK(from:my_server) / T_FIN(to:my_server, bw)、移行9450:T_SYN_NAC(from:my_server) / T_FIN(to:my_server, bw)、移行9450:T_FIN_ACK(from:other_server) / T_ACK_ACK(to:other_server)、移行9450:timeout Timer_FIN / T_FIN(to:my_server, conn_bw), restart Timer_FINの場合、それぞれの状態(それぞれ、1130、7120、9350、および9450)が維持される。

【0078】

ここで、帯域幅の予約が接続の管理に追加される場合、好ましくは必要な帯域幅の要件が、パラメータとしてメッセージ1010に与えられる。例えば、帯域幅は、例えばリンクの使用割合などの代案の代わりに生の帯域幅にて表現されてよく、例えばクライアントからサーバへの方向および逆のサーバからクライアントへの方向の各々について異なる値を含むことができる。さらに、より手の込んだ帯域幅の表現も可能である。例えば、ハードな保証が提供される専用の帯域幅およびソフトな保証しか提供されない共有の帯域幅である。さらに、この帯域幅または一式の帯域幅パラメータが、第1の種類のメッセージ7140および第3の種類のメッセージ7170にも加えられる。帯域幅は、例えばメッセージ1010の受信時に接続の帯域幅として保存され、その後に第1の種類のメッセージ7140および第3の種類のメッセージ7170の両方に使用される。第4の種類のメッセージ4150が、ルータ7500における正しい帯域幅の更新を保証する。改善として、第3の種類のメッセージ7170の帯域幅パラメータが0である場合、第4の種類のメッセージ4150を省略することができる。

【0079】

10

20

30

40

50

図10が、図8のメッセージの流れにおいて説明した本開示による方法の実施の形態においてサーバがとることができる状態および状態変化を示す状態機械の例を示している。

【0080】

この状態機械が、以下の状態および状態変化、すなわち10010「S__WaitCloseRsp」、10020「S__Error」、1240、10030「S__WaitOpenRsp」、7510、8560、1230、および10040「S__WaitCloseRsp」を有している。

【0081】

状態1240が10100:T__LISTEN.req/ーによって開始され、状態10030への移行が、10150:T__SYN(from:my_client,bw) /T__OPEN.ind(my_client),start Timer__Rspに対応して生じる。状態10030から状態10020への移行が、10200:timeout Timer__Rsp/stop Timer__Rsp,T__SYN__ERR(to:my_client)によって生じ、ここから状態10010への移行が、10300:T__OPEN.rsp()/T__CLOSE.ind()に対応して生じる。

【0082】

さらに、状態10030および状態7510の間の状態移行が、10450:T__OPEN.rsp()/stop Timer__Rsp,T__SYN__ACK(to:my_client),start Timer__ACKに対応して生じる。

【0083】

さらに、そこから状態1230への移行が、移行10600:T__ACK__ACK(from:my_client)/stop Timer__ACKおよび移行10600:T__DATA(from:my_client,data)/stop Timer__ACK,T__DATA.ind(data)に従って可能であり、状態7510から状態8560へと、10550:T__FIN(from:my_client,bw)/T__FIN__ACK(to:my_client),start Timer__ACKに従って可能である。さらに、状態1230から状態10040への状態移行が、10700:T__FIN(from:my_client,bw)/T__CLOSE.ind()によって生じる。そこから、状態8560へのさらなる状態移行が、10800:T__CLOSE.rsp/T__FIN__ACK(to:my_client),start Timer__ACKに従って可能であり、再び出発点へと状態8560から状態1240への移行が、10850:T__ACK__ACK(from:my_client)/stop Timer__ACKに従って可能である。

【0084】

それぞれの状態(それぞれ、10010、10020、1240、10030、7510、1230、および10040)が、移行10350:T__SYN(from:any_client,bw)/T__SYN__ERR(to:any_client)、移行10350:T__ACK__ACK(from:any_client)/ー、移行10250:T__SYN(from:any_client,bw)/T__SYN__ERR(to:any_client)、移行10250:T__ACK__ACK(from:other_client)/ー、移行10900:T__ACK__ACK(from:any_client)/ー、移行10400:T__ACK__ACK(from:other_client)/ー、移行10500:T__SYN(from:my_client,bw)/T__SYN__ACK(to:my_client)、移行10500:T__SYN(from:other_client,bw)/T__SYN__ERR(to:other_client)、移行10500:T__ACK__ACK(from:other_client)/ー、移行10500:timeout Timer__ACK/T__SYN__ACK(to:my_client),restart Timer__ACK、移行10650:T__DATA.req(data)/T__DATA(to:my_client,data)、移行10650:T__DATA(from:my_client,data)

)/T_DATA.ind(data), 移行10650:T_SYN_ACK(from:other_client)/-, 移行10650:T_SYN(from:other_client,bw)/T_SYN_ERR(to:other_client)、移行10650:T_ACK_ACK(from:other_client)/-、および移行10750:T_DATA.req(data)/T_DATA(to:my_client,data)、移行10750:T_FIN(from:my_client,bw)/-、移行10750:T_SYN(from:other_client,bw)/T_SYN_ERR(to:other_client)、移行10750:T_ACK_ACK(from:other_client)/-に対応して維持される。また、状態10010から状態1240への移行が、10950:T_CLOSE.rsp/-に従って生じる。

10

【0085】

さらに、例えばサーバが、第1の種類のメッセージ7140および第3の種類のメッセージ7170について、帯域幅パラメータをさらに受信する。状態8560が、好ましくはサーバによる第4の種類のメッセージ4150の待機を生じさせるために、導入される。好ましくは、第6の種類のメッセージ4170の送信時に開始され、時間切れの場合に再送信を生じさせるタイマによって、第6の種類のメッセージ4170も確実な送信に保護される。サーバにおけるすべてのタイマは、並行して動作することがないため、好ましくは1つのタイマにて好都合に実現することができる。

【0086】

20

図11が、図8に示して説明した本開示による方法の実施の形態においてルータがとることができる状態および状態変化についての状態機械の例を示している。

【0087】

以下の状態、すなわち7510、11540「R_Busy_Fin」、および11530「R_Busy_Syn」が、例えば可能である。

【0088】

状態7510の開始が、11100:R_bw=0, err_flag=falseに従って生じる。ここから、状態11540への状態変化が、11400:T_FIN(from:client,to:server,bw)/R_client=client, R_server=server, R_bw-=bw, T_FIN(from:client,to:server,bw)に対応して生じる。状態11540から戻る移行が、11500:T_ACK_ACK(from:R_client,to:R_server)/err_flag=false, T_ACK_ACK(from:R_client,to:R_server)に対応して生じることができる。また、状態7510から状態11530への状態変化が、移行11200:T_SYN(from:client,to:server,bw), (R_bw+bw) ≤ MAX_BW/R_client=client, R_server=server, R_bw+=bw, T_SYN(from:client,to:server,bw)、または移行11200:T_SYN(from:client,to:server,bw), (R_bw+bw) > MAX_BW/R_client=client, R_server=server, err_flag=true, T_SYN_ERR(from:client,to:server)に対応して生じることができ、状態11530から出発点への状態変化が、11350:T_ACK_ACK(from:R_client,to:R_server)/err_flag=false, T_ACK_ACK(from:R_client,to:R_server)に対応して生じることができる。状態11530および11540の間の他の状態変化が、11300:T_FIN(from:client,to:server,bw)/T_FIN(from:client,to:server,bw)に対応して生じる。

30

40

【0089】

それぞれの状態（それぞれ、7510、11540、および11530）は、移行11

50

150: T_DATA (from: node1, to: node2) / T_DATA (from: node1, to: node2)、移行11150: OTHER_MSG (from: node1, to: node2, ...) / OTHER_MSG (from: node1, to: node2, ...)、移行11450: T_FIN_ACK (from: R_server, to: R_client) / T_FIN_ACK (from: R_server, to: R_client)、移行11450: T_FIN (from: R_client, to: R_server, bw) / T_FIN (from: R_client, to: R_server, bw)、移行11450: T_FIN (from: other_client, to: other_server, bw) / -、移行11450: T_DATA (from: node1, to: node2) / T_DATA (from: node1, to: node2)、または移行11450: OTHER_MSG (from: node1, to: node2, ...) / OTHER_MSG (from: node1, to: node2, ...)、ならびに移行11250: T_SYN_ACK (from: server, to: client) / T_SYN_ACK (from: server, to: client)、移行11250: T_SYN_ERR (from: server, to: client) / T_SYN_ERR (from: server, to: client)、移行11250: T_SYN (from: R_client, to: R_server, bw) && (err_flag == false) / T_SYN (from: R_client, to: R_server, bw)、移行11250: T_SYN (from: R_client, to: R_server, bw) && (err_flag == true) / T_SYN_ERR (from: R_client, to: R_server)、移行11250: T_SYN (from: other_client, to: other_server, bw) / -、移行11250: T_DATA (from: node1, to: node2) / T_DATA (from: node1, to: node2)、または移行11250: OTHER_MSG (from: node1, to: node2, ...) / OTHER_MSG (from: node1, to: node2, ...)に従って維持される。

【0090】

例えば、ルータは、接続の開放および閉鎖のプロセス間の状態を集めなければならない。デフォルトの状態は、ルータがレディである状態7510であり、そこからパケットをそれぞれの宛て先へと転送する。しかしながら、ルータが第1の種類のメッセージ7140または第3の種類のメッセージ7170を受信する場合、好ましくはルータの帯域幅の予約が更新され、これを行う際に、ルータが状態11530および11540へとそれぞれ移行する。ルータが初期の状態7510にあり、割り当てるための十分な帯域幅を有するメッセージ7140を受信する場合、それは帯域幅を更新し、メッセージをサーバへと転送する。他方で、利用できる帯域幅が不十分である場合、ルータは、帯域幅の更新を防止するためにerror_flagを発し、メッセージ7140の転送の代わりにエラーメッセージがサーバへと設定される。好ましくは、error_flagを接続を閉じるときに帯域幅の更新を防止するために使用することができる。いずれの場合も、接続の確立に関わるクライアントおよびサーバのペアが、再送信に起因する多数の帯域幅の更新を防止するために、安全であるのが望ましい。状態7510にあってルータがメッセージ7170を受信する場合、やはりクライアントおよびサーバのペアは、メッセージ7140に関して安全であり、ルータの帯域幅が好ましくはbwによって増やされる。

【0091】

状態11530において、保存されたクライアント・サーバ・ペアに適合する第4の種類のメッセージ4150が受信された場合、error_flagが除去され、ルータが状態7510へと移行する。保存されたクライアント・サーバ・ペアに適合するメッセージ7170が受信された場合、ルータは状態11540へと移行する。好ましくは、メッセージ4150およびメッセージ4170を含むすべてのメッセージが転送される。保存されたクライアント・サーバ・ペアについてメッセージ7140が受信される場合と、e

`error_flag`が設定される場合は、例外である。状態11540にあるとき、保存されたクライアント・サーバ・ペアに適合するメッセージ4150が受信された場合、`error_flag`が除去され、ルータが状態7510に移行する。この状態において、メッセージ4150を含むすべてのメッセージが転送される。好ましくは、接続の開放／閉鎖のたびに、ルータは、アドレスおよびポートで構成され得るクライアントおよびサーバの身元を参照する接続を特定する状態を保存する必要がある。例えば、最も単純なルータの実施例は、図11に示されるように、ただ1つの接続の身元を保存することができる。あるいは、ルータがいくつかの接続の身元を保存することができ、その場合には、保存することができる接続の身元ごとに図11に示したとおりの1つの状態機械を使用することができる。この場合、接続の身元に保存することができるメッセージ7140または7170が受信されるたびに、ルータが、図11に示した状態および状態変化に従事する。ルータが接続の身元を保存できない場合、好ましくは、ルータは、メッセージ7140または7170を破棄し、それ以上の行為を行わない。

10

【0092】

以下の図12～図21は、本開示によるシステムによって同じやり方で取り扱うことができる本開示の方法の上述の実施の形態の方法によるエラーおよびそれらエラーの取り扱いの種々の例を示している。

【0093】

図12が、図1のさらなる詳細における本開示による方法の実施の形態において生じるエラーの事例の例を示している。

20

【0094】

この場合、例えばサーバ1200がビジーである可能性があり、したがって現時点においてサーバ1200が別の接続に対応していることを示す状態1230にあるため、第1の種類のメッセージ1140を転送できない可能性がある。この場合、アプリケーションサーバ1300との接続を確立する代わりに、エラーメッセージ12150「T__SYN__ERR」が生成され、このエラーメッセージがクライアント1100に受信されたとき、クライアントが、クライアントが接続されると状態1110ではなく、接続が閉じられる状態1130へと移行する。

【0095】

図13および図14は、喪失または破棄の可能性があるメッセージの伝送を監視するためのタイマの使用を示している。

30

【0096】

図13は、図1においてさらに詳しく検討した本開示による方法の実施の形態においてエラーが生じる可能性がある事例を示している。この場合、第1の種類のメッセージが失われ(13140)、結果としてサーバ1200によって受信または処理されることがなく、サーバ1200が状態1210のままとなる可能性がある。しかしながら、この場合に、第1の所定の期間を測定する第1のクライアントタイマが時間切れとなり、クライアント1100とサーバ1200との間の接続を確立するために、第1の種類のメッセージ1140の再送信を生じさせる。第1の所定の期間は、好ましくは、サーバ1200に第2の種類のメッセージ1150を生成するための十分な処理時間ならびに両方向の十分な伝送時間を許すようなやり方で決められる。第2の種類のメッセージ1150がクライアント1100によって受信されるとき、さらなる取り扱いは、図1において検討したとおりである。

40

【0097】

図14は、さらに詳しく図4に関して検討した本開示による方法の実施の形態において生じる可能性がある別のエラーおよびその取り扱いを説明している。この場合、参照符号14150で示された第2の種類のメッセージが失われる。この場合にも、図13と同様に参照符号13010によって示される時間切れが生じ、第1の種類のメッセージ1140の再送信を生じさせる。これが、サーバ1200に到着したこのメッセージが検出され、初めて受信された第1の種類のメッセージ1140によって引き起こされる状態121

50

0 から 1 2 2 0 への状態変化ゆえに、このメッセージ 1 1 4 0 が再送信され、このメッセージのソースが最初であることをサーバが特定することを可能にする。サーバに、再送信されたメッセージ 1 1 4 0 を処理することなく破棄し、接続された状態 1 2 3 0 へと状態を変化させ、確認のための第 2 の種類のメッセージ 1 1 5 0 を送信させる。

【0098】

図 1 5 は、図 1 においてさらに詳しく検討した本開示による方法の実施の形態に関して、メッセージの交換において生じる別のエラーの事例の例を説明している。この場合に、アプリケーションサーバ 1 3 0 0 におけるアプリケーションのクラッシュの場合に何が生じるのかが検討される。ここで、サーバ 1 2 0 0 がメッセージ 1 3 1 0 を送信しているが、クラッシュしたアプリケーション 1 3 2 0 から応答を受信することはない。これは、第 2 の所定の期間を測定する第 1 のサーバタイムにおけるタイムアウト 1 5 2 5 0 につながるが、第 2 の所定の期間の大きさを決める際には、両方向のメッセージの伝達ならびにアプリケーションサーバがメッセージ 1 3 1 0 を処理して応答 1 3 2 0 を生成するための時間が考慮されるべきである。タイムアウト 1 5 2 5 0 に起因する第 1 のサーバタイムの時間切れが、エラーを示す状態 6 5 5 0 を生成し、エラーメッセージ 1 5 1 5 0 をクライアントへと送信する。第 1 の種類のメッセージ 1 1 4 0 のさらなる送信が、サーバ 1 2 0 0 からのエラーメッセージ 1 5 1 4 0 につながり、エラーメッセージ 1 5 1 5 0 または 1 5 1 4 0 に応答したアプリケーションクライアントへの対応するメッセージ 1 5 0 1 0 につながる。どちらの場合も、クライアントは接続を閉じ、状態 1 1 3 0 に移行する。

【0099】

アプリケーションサーバがメッセージ 1 3 2 0 T__OPEN. r s p によってサーバに応答する場合、サーバが、メッセージ 1 3 7 0 T__CLOSE. i n d を使用して接続が実際に閉じられた旨をアプリケーションサーバに通知し、接続確立の要求の受信の準備ができた状態 1 2 1 0 S__L i s t e n へと戻る。

【0100】

図 1 6 は、図 4 においてメッセージの流れを詳しく検討した本開示による方法の実施の形態においてエラーをどのように取り扱うことができるのかについての例を示している。

【0101】

ここでもやはり、信頼できるネットワークにおける事例においてすでに検討したように、サーバ 1 1 0 0 がビジーであり、要求を受けて接続を確立することができない。したがって、クライアント 1 1 0 0 へと知らせるべきエラーメッセージ 1 2 1 5 0 が生成される。これにより、第 3 の種類のメッセージ 1 1 5 0 を発する接続閉鎖の動作がクライアント 1 1 0 0 において生じ、この第 3 の種類のメッセージ 1 1 5 0 が、サーバ 1 2 0 0 から発せられる第 6 の種類のメッセージ 4 1 7 0 によって確認される。次いで、クライアントが状態 1 1 3 0 に移行する。

【0102】

図 1 7 は、図 4 においてさらに詳しく検討したようにメッセージの交換が信頼性に欠けるネットワークにもとづく本開示による方法の実施の形態において生じるエラーの事例を示している。信頼性に欠けるネットワークにおいては、あらゆるメッセージに破損または喪失の可能性があるため、第 2 の種類のメッセージ 1 1 5 0 も、適切な取り扱いに関して監視されるのが望ましい。そうでない場合、喪失により、サーバは最初のビジーにおける「S__C o n n e c t e d」状態 1 2 3 0 のままであるかもしれず、しかし一方、接続 1 2 1 5 0 を確立するリクエストのエラーメッセージに応答することが、参照符号 1 7 1 5 0 で表される他のクライアントから切断し、状態 1 2 1 0 で再び新たな接続のために利用可能となる。しかしながら、エラーメッセージのクライアント 1 1 0 0 への到達が遅すぎ、第 1 の種類のメッセージ 1 1 4 0 の再送信を引き起こしたタイムアウト 1 3 0 1 0 の後、第 1 の種類のメッセージ 1 1 4 0 により、サーバがアプリケーションサーバ 1 3 0 0 との接続を開くとともにデータ 1 1 6 0 の送信を開始し、データ 1 1 6 0 は、第 2 の種類のメッセージ 1 4 1 5 0 が失われたときに、プロトコルエラー 1 7 1 6 0 を生成する。

【0103】

図18～図21は、図8および関連の説明においてさらに詳しく検討および図示した本開示による方法の実施の形態において生じる別のエラーの事例を示している。

【0104】

ここでは、ルータ7500がビジーであり、あるいは帯域幅を利用できないがゆえに帯域幅の変更の要求に応えることができず、この状況ゆえのエラーメッセージを生成する。

【0105】

図18が、その状況を取り扱う第1の事例を示している。第1の種類のメッセージ7140を受信すると、ルータ21500が、21520において帯域幅の割り当てを拒絶し、エラーフラグを設定する。これが、ルータ7500へのエラーメッセージ20140の送信につながり、これがクライアント1100へと転送される。ルータ7500が、20540においてエラーフラグを設定する。他方で、エラーメッセージ20540を受信したクライアントは、接続の終了を開始し、エラーによる終了メッセージ20170「T__FIN__ERR」を送信する。これが、ルータ21500へと転送され、確認メッセージ4170が生成され、確認メッセージ4170がクライアント1100に到着すると、エラーメッセージ12020「T__OPEN. err」の生成が引き起こされる。

【0106】

図19は、図18において検討したものと同一エラーの事例の例を示しているが、メッセージの流れおよび取り扱いがわずかに変わっている。第1の種類のメッセージ7140の受信時に、ルータ21500が19520においてerror__flagを設定せず、クライアントへのエラーメッセージ21140において帯域幅情報を送信することがない。このように、先の例とは対照的に、クライアント1100が、図8において述べたとおりの通常の接続終了を辿ることができる。

【0107】

この場合、ルータは、エラーの事例19530「R__BusyERR」において異なる状態に入る。通常の接続の終了は、利用可能な場合に帯域幅の割り当てを解放する。図19に示した例の場合には、メッセージの取り扱いがより容易であり、送信を必要とするメッセージの量がより少ない。しかしながら、図18および19に示したどちらの場合も、ルータを、好ましくはエラーメッセージ21140および20140を発するときにサーバの一部の機能を実行するように構成する必要がある。

【0108】

そのようなルータの変更を避けるために、考えられる代案は、エラーの取り扱いをサーバ1200へと転送することであり、これが図20および図21の例で検討される。

【0109】

図20は、図8において検討した本開示による方法の実施の形態においてルータがビジーであってメッセージの交換が信頼性に欠けるネットワークにもとづくエラーの事例の取り扱いの例を示している。

【0110】

ここで、図18において検討した事例と同様に、ひとたび転送された第1の種類のメッセージ7140を受信しかつ利用可能な帯域幅を有していない場合、ルータがerror__flagを設定し、この場合には新しいエラーメッセージ20140をサーバ1200へと転送し、サーバ1200が帯域幅パラメータを含むエラーメッセージ20140を生成する。サーバによるさらなる追加の仕事は、エラーがクライアント1100によって第7の種類のメッセージ4150にて発せられたときに接続の終了へと応答することである。

【0111】

さらに図21が、図19に関連して検討したエラーの取り扱いの例を示しているが、この場合にはエラーの取り扱いがサーバ1200において行われる。変更として、図19における手順が、エラーメッセージ21140がサーバ1200へと転送されて、サーバ1200がエラーメッセージ21140の送信および接続の終了の取り扱いを担当するようなやり方で変更される。

【0112】

信頼性に欠けるネットワークに関して、ルータにおける帯域幅の更新について考えられる代案は、それを7140の代わりの第2の種類のメッセージ1150および／または7170の代わりのメッセージ4170によって行うことである。ここで、第1のメッセージ7140および／または7170が受信されたとき、クライアント／サーバペアを上述のように保存でき、さらにフラグERR__flagを設定することができる。しかしながら、この場合に、帯域幅パラメータがメッセージ1150および4170によって運ばれる。エラーのこの場合に、メッセージ1150および／または4170による帯域幅の更新が行われていないため、エラーメッセージ21140はいかなる帯域幅パラメータも運ばず、帯域幅なしに接続の終了につながる。したがって、メッセージ4150を、このエラーの場合にさらに最適化することができる。

10

【0113】

図22が、同様の文脈において接続がメッセージ1070「T_CLOSE.req」を送信しているアプリケーションクライアント1000によって閉じられる場合を示している図4に加えて、別の選択肢を例示している。あるいは、接続を、メッセージ1070「T_CLOSE.req」を送信するアプリケーションサーバ1300によって閉じることにもできる。この場合にも、結果としての工程は同じである。T/COサーバ1200が、メッセージ1070「T_CLOSE.req」を受信したときに、メッセージ1170「T_FIN」を送信し、タイマを開始させ、状態4112「S_WaitFinAck」へと入る。T/COクライアント1100が、メッセージ1170「T_FIN」を受信したときに、メッセージ1370「T_CLOSE.ind」をアプリケーションクライアントへと発し、アプリケーションクライアントが、T/COクライアント1100へのメッセージ4380「T_CLOSE.rsp」で応答し、T/COクライアント1100が、T/COサーバへとメッセージ4170「T_FIN_ACK」を順に送信し、状態1130「C_Closed」へと入る。T/COサーバは、メッセージ4170「T_FIN_ACK」を受信したとき、自身のタイマを停止させ、メッセージ4080「T_CLOSE.cnf」をアプリケーションサーバ1300へと送信し、状態1240「S_Listen」へと入る。

20

【0114】

図23が、同様の文脈における図4に加えて、接続の閉鎖がT/COクライアント1100およびT/COサーバ1200の両方において同時に要求される事例を示している。この場合には、T/COクライアント1100およびT/COサーバ1200からのメッセージ1070「T_FIN」がネットワークにおいて交差する。メッセージ1070「T_FIN」がT/COクライアント1100において受信されるとき、T/COクライアント1100は、タイマを停止させ、T/COサーバ1200へとメッセージ4170「T_FIN_ACK」を送信し、状態1130「C_Closed」へと移行する。メッセージ1070「T_FIN」がT/COサーバ1200において受信されるとき、T/COサーバ1200は、タイマを停止させ、T/COクライアント1100へとメッセージ4170「T_FIN_ACK」を送信し、状態1240「S_Listen」へと移行する。

30

40

【0115】

さらに本開示は、サーバおよび／またはクライアントにおいて具現化されるようなシステムの機能の任意のいずれかを、ハードウェア、コンピュータソフトウェア、または両者の組み合わせとして実現できることを含む。システム（例えば、クライアントおよび／またはサーバ）は、汎用のプロセッサ、デジタル信号プロセッサ（DSP）、特定用途向け集積回路（ASIC）、フィールド・プログラマブル・ゲート・アレイ（FPGA）または他のプログラマブルな論理デバイス、ディスクリートなゲートまたはトランジスタ論理回路、ディスクリートハードウェア部品、あるいは本明細書に記載の機能を実行するように設計された任意の組み合わせを含むことができる。汎用プロセッサは、マイクロプロセッサまたは状態機械であってよい。プロセッサを、例えばDSPとマイクロプロセッサと

50

の組み合わせ、複数のマイクロプロセッサ、DSPコアが組み合わせられた1つ以上のマイクロプロセッサ、あるいは任意の他のそのような構成など、演算装置の組み合わせとして実現することもできる。クライアントおよび／またはサーバにおいて使用されるプロセッサが、本開示のいずれかの方法を実行するように構成される。

【0116】

さらに本開示は、任意の種類の演算装置上で実行されるように構成され、例えばサーバおよび／またはクライアントにおいて使用されるように構成され、すなわち処理エンジンを含んでいる装置において実行されるように構成されたコード部分を含んでいるコンピュータプログラム製品を含む。コンピュータプログラム製品のソフトウェアコードが、演算装置上で実行されたときに、第1の種類のメッセージをサーバへと送信することによって接続の確立を要求する機能をクライアントにおいて提供する。サーバにおいては、ソフトウェアが、演算装置上で実行されたときに、サーバの接続につながる第2の種類のメッセージをクライアントへと送信することによって接続を確立する能力を裏付ける。ソフトウェアは、好ましくは、第1の種類のメッセージの送信によって第1の最大応答時間としての第1の所定の期間を測定する第1のクライアントタイマが開始されるように構成され、好ましくは第2の種類のメッセージまたはデータメッセージの受信をもって第1のクライアントタイマを停止させるように構成される。さらにソフトウェアは、好ましくは、第3の種類のメッセージの送信によって接続が閉じられるように構成される。

10

【0117】

さらにソフトウェアは、好ましくは、第2の種類のメッセージが受信されることなく第1の所定の期間が過ぎた場合に別の第1の種類のメッセージが送信されるように構成される。さらにソフトウェアは、好ましくは、サーバが接続を引き受けることができない場合にエラーメッセージを返すように構成される。

20

【0118】

さらにソフトウェアは、好ましくは、

a) 第2の種類のメッセージの受信が、クライアントの接続につながり、

b) クライアントが第2の種類のメッセージの受信を確認するためにサーバへと第4の種類のメッセージを送信する

ように構成される。

【0119】

さらにソフトウェアは、好ましくは、第2の種類のメッセージの送信によって第2の最大応答時間としての第2の所定の期間を測定する第1のサーバタイマが開始され、第4の種類のメッセージの受信によって第1のサーバタイマが停止されるように構成される。

30

【0120】

さらにソフトウェアは、好ましくは、サーバが接続を確立できない場合にクライアントの非接続につながる第5の種類のメッセージをクライアントへと送信するように構成することができる。

【0121】

さらにソフトウェアは、好ましくは、第3の種類のメッセージの送信によって第3の最大応答時間としての第3の所定の期間を測定する第2のクライアントタイマが開始されるように構成される。

40

【0122】

さらにソフトウェアは、好ましくは、サーバが第6の種類のメッセージを送信することによって第3の種類のメッセージの受信を確認するように構成される。

【0123】

さらにソフトウェアは、好ましくは、第6の種類のメッセージの送信によって第4の最大応答時間としての第4の所定の期間を測定する第2のサーバタイマが開始されるように構成される。

【0124】

さらにソフトウェアは、好ましくは、クライアントにおける第6の種類のメッセージの

50

受信によって第2のクライアントタイマが停止され、クライアントによる第4の種類のメッセージの送信が生じるように構成される。

【0125】

さらにソフトウェアは、好ましくは、サーバによる第3の種類のメッセージの送信によって第3の最大応答時間としての第3の所定の期間を測定する第2のサーバタイマが開始されるように構成される。

【0126】

さらにソフトウェアは、好ましくは、第3の種類のメッセージを受信するクライアントが、第3の種類のメッセージの受信を第6の種類のメッセージを送信することによって確認するように構成される。

10

【0127】

さらにソフトウェアは、好ましくは、第6の種類のメッセージの送信によって第4の最大応答時間としての第4の所定の期間を測定する第2のクライアントタイマが開始されるように構成される。

【0128】

さらにソフトウェアは、好ましくは、サーバにおける第6の種類のメッセージの受信によって第2のサーバタイマが停止され、クライアントによる第4の種類のメッセージの送信が生じるように構成される。

【0129】

さらにソフトウェアは、好ましくは、クライアントとサーバとの間のメッセージのうちの少なくとも1つがルータを通過でき、この少なくとも1つのメッセージが、リソース予約の要求を含んでおり、ルータが要求されたリソースを提供できる場合に限ってこの少なくとも1つのメッセージを転送する。

20

【0130】

さらにソフトウェアは、好ましくは、ルータが少なくとも1つのメッセージを転送する場合に、このリソース予約の要求に関する接続の確立に関わるクライアント・サーバ・ペアに関係する第2の種類のメッセージの受信まで、さらなるリソース予約の要求を一時的に停止するように構成される。

【0131】

さらにソフトウェアは、好ましくは、接続が閉じられる場合にリソースの予約を取り消すために少なくとも1つのメッセージが使用されるように構成される。

30

【0132】

さらにソフトウェアは、好ましくは、クライアントおよびサーバのそれぞれが、それぞれのアプリケーションクライアントとそれぞれのアプリケーションサーバとの間の接続を確立するように構成される。

【0133】

さらに本開示は、上述のコンピュータプログラム製品を保存する機械によって読み取ることができる一時的でない信号記憶媒体を含む。例えば、機械によって読み取ることができる一時的でない信号記憶媒体は、CDROMまたはDVDROMなどの光ディスク、磁気テープメモリ、ハードディスクまたはディスクなどの磁気ディスクメモリ、USBメモリスティックまたはフラッシュメモリなどの半導体メモリ、などであってよい。

40

【図 2】

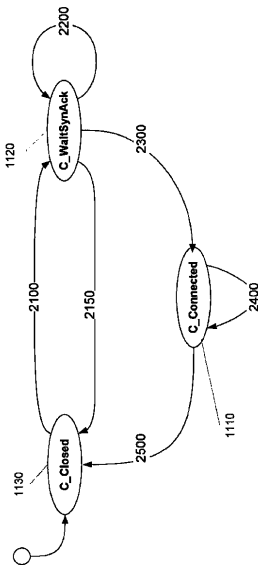


Fig. 2

【 図 4 】

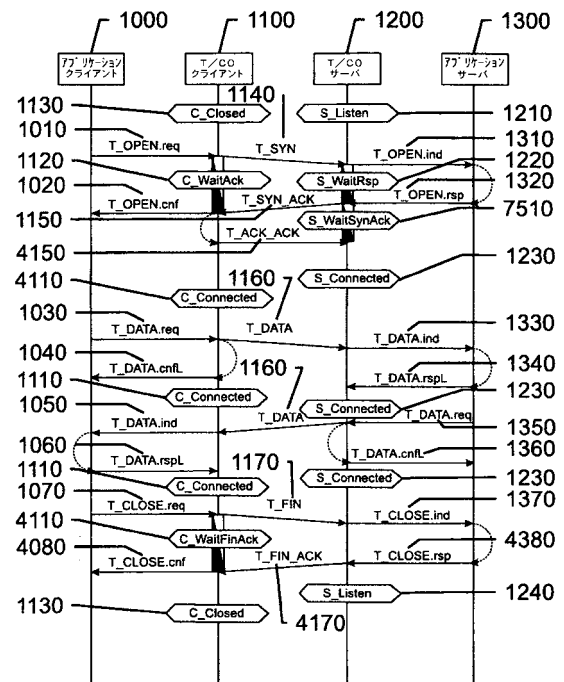


Fig. 3

【 図 5 】

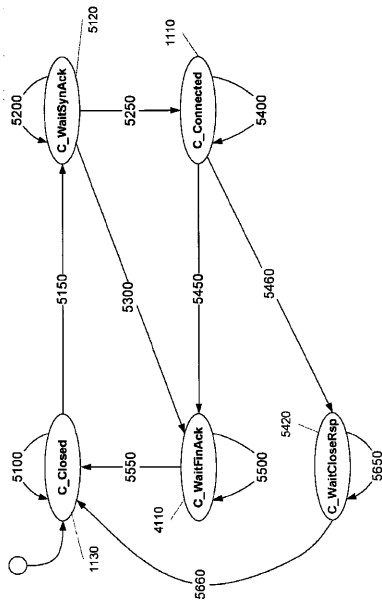


Fig. 5

【図 6】

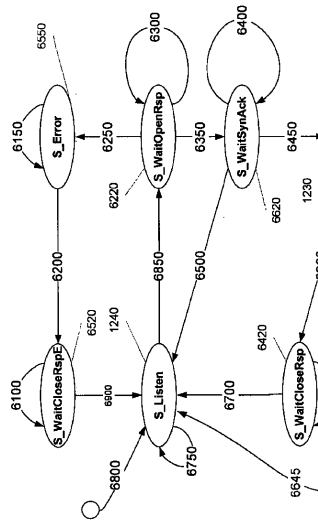
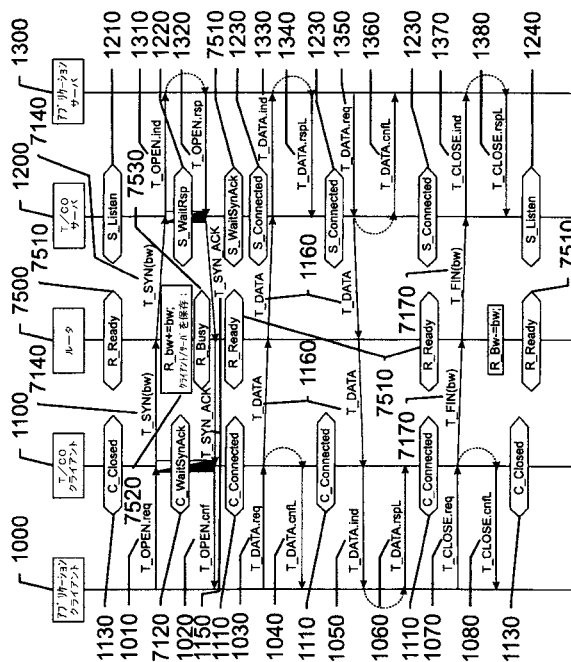
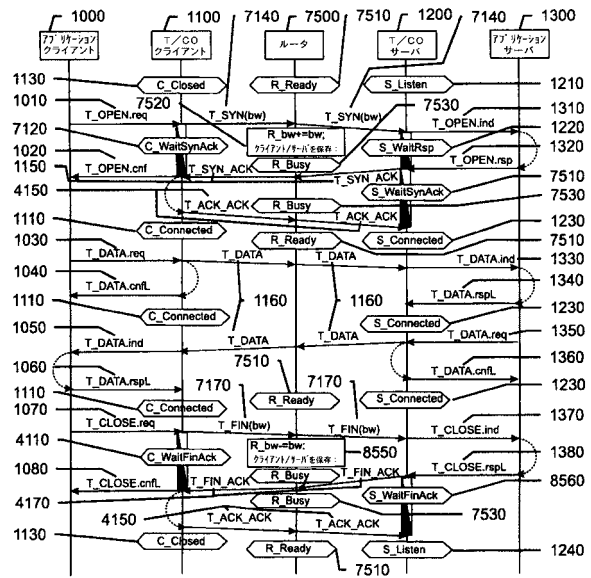


Fig. 6

【图 7】



【图 8】



【図 9】

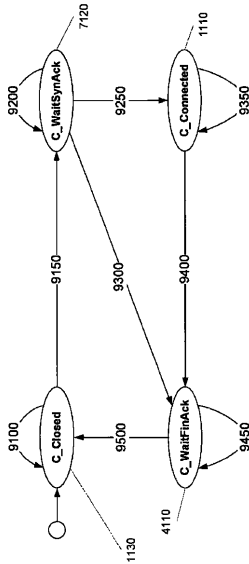


Fig. 9

【図 10】

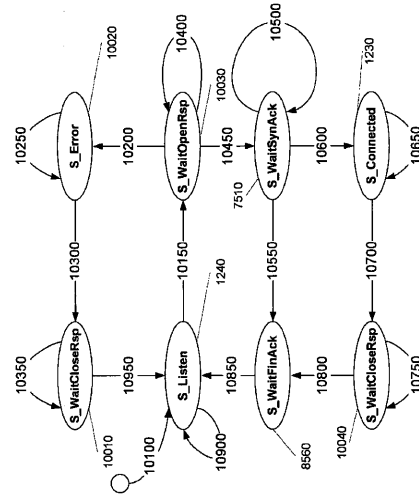


Fig. 10

【図 11】

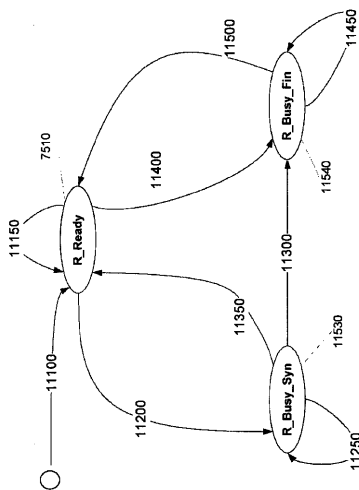
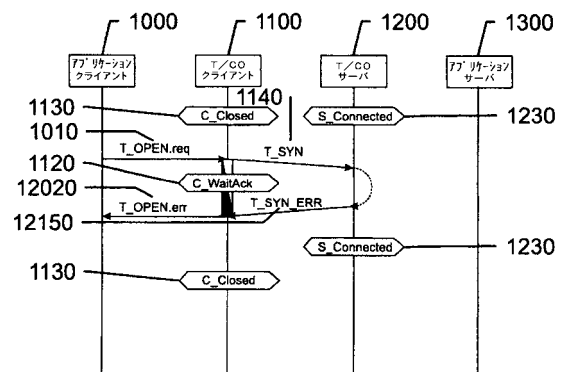
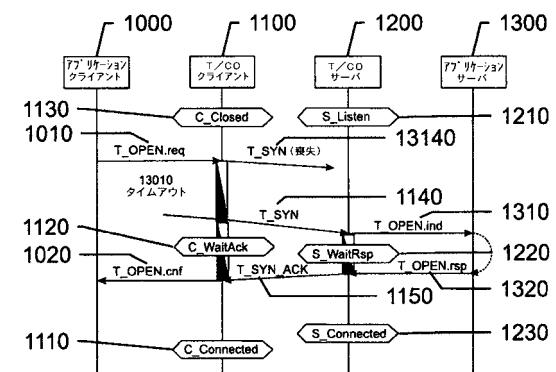


Fig. 11

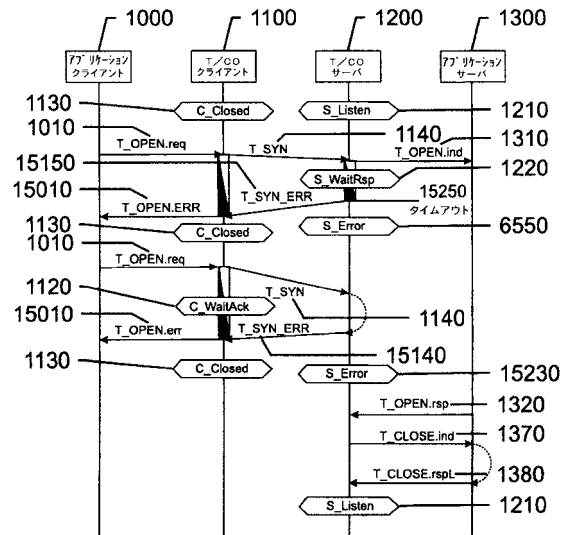
【図 12】



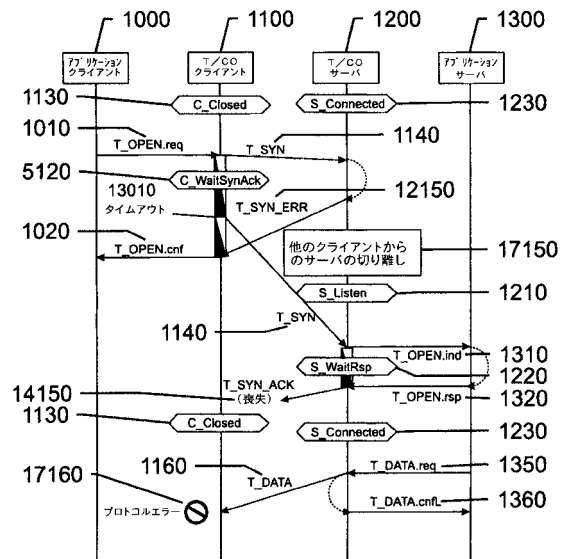
【図 13】



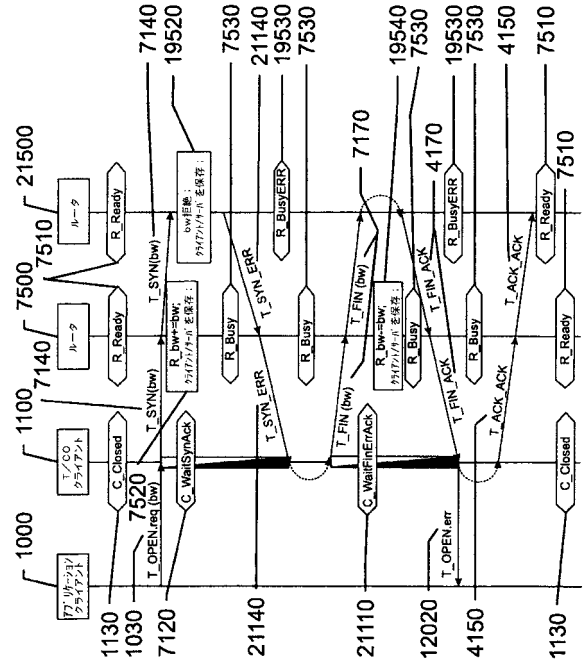
【 図 1 5 】



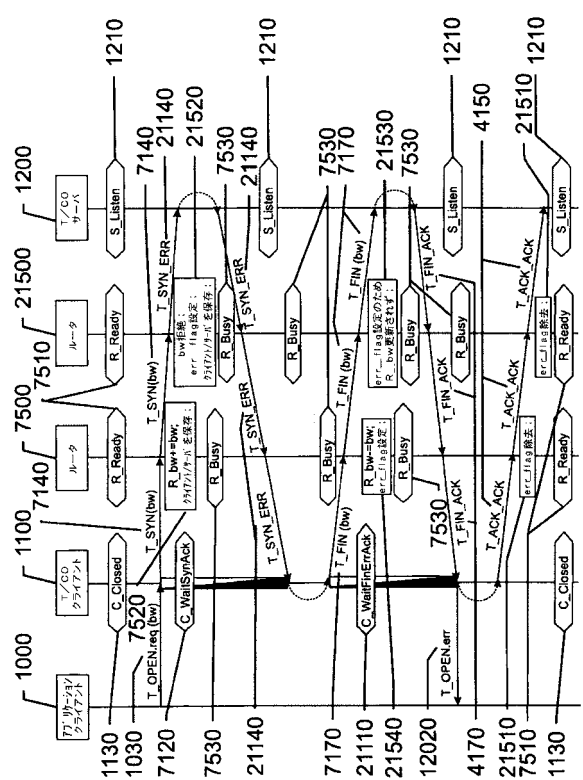
【図 17】



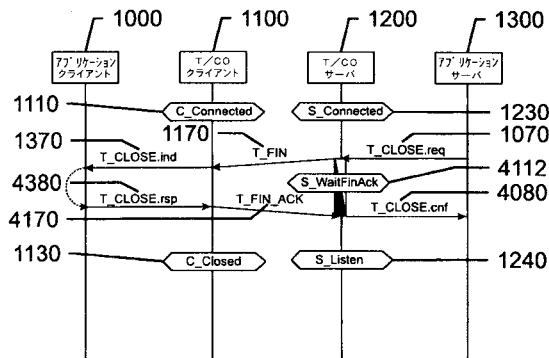
【 図 1 9 】



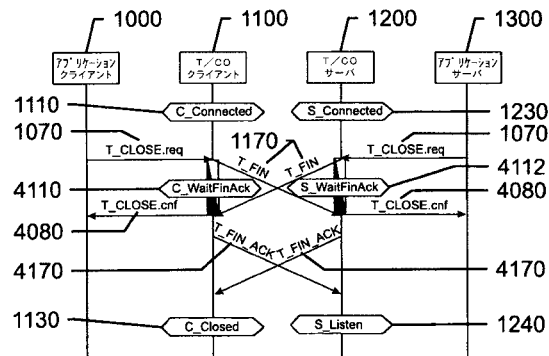
【 図 2 1 】



【図 2 2】



【図 2 3】



【手続補正書】

【提出日】平成24年5月30日(2012.5.30)

【手続補正 1】

【補正対象書類名】特許請求の範囲

【補正対象項目名】全文

【補正方法】変更

【補正の内容】

【特許請求の範囲】

【請求項 1】

接続指向の正順デリバリ環境における接続を管理するための方法であって、

a) クライアントが第 1 の種類のメッセージをサーバへと送信することによって、接続の確立を要求し、

b) サーバが、サーバの接続につながる第 2 の種類のメッセージをクライアントへと送信することによって、接続の確立が可能であることを確認し、

c) 前記第 1 の種類のメッセージを送信することによって、第 1 の最大応答時間としての第 1 の所定の期間を測定する第 1 のクライアントタイマを始動し、前記第 2 の種類のメッセージまたはデータメッセージを受信することによって、前記第 1 のクライアントタイマが停止され、

d) 第 3 の種類のメッセージを送信することによって接続が閉じられる、方法。

【請求項 2】

第 2 の種類のメッセージが受信されることなく前記第 1 の所定の期間が過ぎた場合に、別の第 1 の種類のメッセージが送信される請求項 1 に記載の方法。

【請求項 3】

サーバが接続を引き受けることができない場合に、サーバがエラーメッセージを返す請求項 1 に記載の方法。

【請求項 4】

a) 前記第 2 の種類のメッセージを受信することが、クライアントの接続につながり、
b) 前記第 2 の種類のメッセージの受信を確認するための第 4 の種類のメッセージを、
該クライアントがサーバに送信する、請求項 1 に記載の方法。

【請求項 5】

前記第 2 の種類のメッセージを送信することによって、第 2 の最大応答時間としての第 2 の所定の期間を測定する第 1 のサーバタイマを始動し、前記第 4 の種類のメッセージを受信することによって、該第 1 のサーバタイマが停止される請求項 4 に記載の方法。

【請求項 6】

サーバが接続を確立できない場合に、サーバがクライアントの非接続につながる第 5 の種類のメッセージをクライアントへと送信する請求項 1 に記載の方法。

【請求項 7】

前記第 3 の種類のメッセージの送信により、第 3 の最大応答時間としての第 3 の所定の期間を測定する第 2 のクライアントタイマが始動される請求項 1 に記載の方法。

【請求項 8】

サーバが第 6 の種類のメッセージを送信することによって、前記第 3 の種類のメッセージの受信を確認する請求項 1 に記載の方法。

【請求項 9】

前記第 6 の種類のメッセージの送信により、第 4 の最大応答時間としての第 4 の所定の期間を測定する第 2 のサーバタイマが始動される請求項 8 に記載の方法。

【請求項 10】

クライアントにおける第 6 の種類のメッセージの受信により、前記第 2 のクライアントタイマが停止されるとともに、

クライアントが第 4 の種類のメッセージを送信する請求項 7 に記載の方法。

【請求項 11】

サーバによる前記第 3 の種類のメッセージの送信により、第 5 の最大応答時間としての第 3 の所定の期間を測定する第 2 のサーバタイマが始動される請求項 1 に記載の方法。

【請求項 12】

前記第 3 の種類のメッセージを受信するクライアントが第 6 の種類のメッセージを送信することによって、該第 3 の種類のメッセージの受信を確認する請求項 1 に記載の方法。

【請求項 13】

前記第 6 の種類のメッセージの送信により、第 4 の最大応答時間としての第 4 の所定の期間を測定する第 2 のクライアントタイマが始動される請求項 12 に記載の方法。

【請求項 14】

サーバにおける前記第 3 の種類のメッセージの受信により、前記第 2 のサーバタイマが停止される請求項 11 に記載の方法。

【請求項 15】

クライアントとサーバとの間の前記メッセージのうちの少なくとも 1 つが、ルータを通過し、

該少なくとも 1 つのメッセージが、リソース予約の要求を含んでおり、

前記ルータが、要求されたリソースを提供できる場合に限って前記少なくとも 1 つのメッセージを転送する、請求項 1 に記載の方法。

【請求項 16】

ルータが前記少なくとも 1 つのメッセージを転送する場合に、該ルータは該リソース予約の要求に関する接続の確立に関わるクライアント・サーバ・ペアに関係する第 2 の種類のメッセージの受信まで、さらなるリソース予約の要求を一時的に停止する請求項 15 に記載の方法。

【請求項 17】

接続が閉じられる場合に、少なくとも 1 つのメッセージがリソースの予約を取り消すために使用される請求項 15 に記載の方法。

【請求項 18】

予約されるリソースが帯域幅である請求項 15 に記載の方法。

【請求項 19】

クライアントおよびサーバのそれぞれが、それぞれのアプリケーションクライアントとそれぞれのアプリケーションサーバとの間の接続を確立する請求項 1 に記載の方法。

【請求項 20】

接続指向の正順デリバリ環境において接続を管理するためのシステムであって、
サーバと、
クライアントと、

正順デリバリのネットワークと、 を備えており、

前記サーバが、請求項 1 に記載の方法のうちのサーバに関する動作のいずれかを実行するように構成され、

前記クライアントが、請求項 1 に記載の方法のうちのクライアントに関する動作のいずれかを実行するように構成されているシステム。

【請求項 21】

ルータを備えており、

クライアントとサーバとの間の前記メッセージのうちの少なくとも 1 つが、該ルータを通過しており、

該少なくとも 1 つのメッセージが、リソース予約の要求を含んでおり、

前記ルータが、要求されたリソースを提供できる場合に限り前記少なくとも 1 つのメッセージを転送する、請求項 20 に記載のシステム。

【請求項 22】

前記ルータが前記少なくとも 1 つのメッセージを転送する場合に、該ルータは該リソース予約の要求に関する接続の確立に関わるクライアント・サーバ・ペアに係る第 2 の種類のメッセージの受信まで、さらなるリソース予約の要求を一時的に停止する請求項 21 に記載のシステム。

【請求項 23】

サーバにおける前記第 3 の種類のメッセージの受信により、サーバは第 4 の種類のメッセージを送信する請求項 11 に記載の方法。

【請求項 24】

サーバにおける第 6 の種類のメッセージの受信により、サーバは第 4 の種類のメッセージを送信する請求項 11 に記載の方法。

【請求項 25】

前記第 4 の種類のメッセージの受信により、前記第 2 のサーバタイマが停止される請求項 23 または 24 に記載の方法。

【請求項 26】

サーバにおける第 6 の種類のメッセージの受信により、前記第 2 のサーバタイマが停止される請求項 11 に記載の方法。

【請求項 27】

クライアントにおける前記第 3 の種類のメッセージの受信により、前記第 2 のクライアントタイマが停止される請求項 7 に記載の方法。

【請求項 28】

クライアントにおける第 6 の種類のメッセージの受信により、前記第 2 のクライアントタイマが停止される請求項 7 に記載の方法。

【請求項 29】

前記第 4 の種類のメッセージの受信により、前記第 2 のサーバタイマが停止される請求項 10 に記載の方法。

【国際調査報告】

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2010/064605

A. CLASSIFICATION OF SUBJECT MATTER

INV. H04L29/06
ADD.

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

EPO-Internal, INSPEC, COMPENDEX

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	Despo Galataki: "Design & Verification of UniPro Protocols for Mobile Phones", Thesis, Vrije Universiteit Amsterdam, 1 July 2009 (2009-07-01), pages 1-86, XP002617612, Amsterdam, The Netherlands Retrieved from the Internet: URL: http://www.cs.vu.nl/~tcs/mt/galataki.pdf [retrieved on 2011-01-14] Chapters 4 and 5 ----- -/-	1-30

☒ Further documents are listed in the continuation of Box C.

☐ See patent family annex.

* Special categories of cited documents:

- *A* document defining the general state of the art which is not considered to be of particular relevance
- *E* earlier document but published on or after the international filing date
- *L* document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
- *O* document referring to an oral disclosure, use, exhibition or other means
- *P* document published prior to the international filing date but later than the priority date claimed

- *T* later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
- *X* document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
- *Y* document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.
- *G* document member of the same patent family

Date of the actual completion of the international search

20 January 2011

Date of mailing of the international search report

28/01/2011

Name and mailing address of the ISA/
European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040,
Fax: (+31-70) 340-3016

Authorized officer

Kesting, Volker

INTERNATIONAL SEARCH REPORT

International application No

PCT/EP2010/064605

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	XIONG BING ET AL: "An Efficient TCP Flow State Management Algorithm in High-Speed Network", INFORMATION ENGINEERING AND ELECTRONIC COMMERCE, 2009. IEEC '09. INTERNATIONAL SYMPOSIUM ON, IEEE, PISCATAWAY, NJ, USA, 16 May 2009 (2009-05-16), pages 106-110, XP031497561, ISBN: 978-0-7695-3686-6 page 107, right-hand column, lines 23-33 -----	7-13,30
A	DEFENSE ADVANCED RESEARCH PROJECTS AGENCY INFORMATION SCIENCES INSTITUTE UNIVERSITY OF SOUTHERN CALIFORNIA 4676 ADMIRALTY WAY MARI: "TRANSMISSION CONTROL PROTOCOL DARPA INTERNET PROGRAM PROTOCOL SPECIFICATION prepared for Information Processing Techniques Office 1400 Wilson Boulevard 22209 by ; rfc793.txt", IETF STANDARD, INTERNET ENGINEERING TASK FORCE, IETF, CH, 1 September 1981 (1981-09-01), XP015006775, ISSN: 0000-0003 sections 3.2, 3.4 -----	7-14,24, 28,29

フロントページの続き

(81)指定国 AP(BW, GH, GM, KE, LR, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), EA(AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), EP(AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, SE, SI, SK, SM, TR), OA(BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG), AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PE, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW

(74)代理人 100167933

弁理士 松野 知紘

(72)発明者 アンドレイ、ラドゥレスク

オランダ国5655、アーカー、アインドーフエン、ハルテルスタイン、30

(72)発明者 デスポ、ガラタキ

キプロス共和国ニコシア、ストロボロス、キラス、11

Fターム(参考) 5K030 GA19 HA08 HB15 LA02 LB01 LC05