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

特許第6444988号
(P6444988)

(45)発行日　平成30年12月26日(2018.12.26)　(24)登録日　平成30年12月7日(2018.12.7)

(51)Int.Cl.
G06F 13/00 (2006.01)
H04L 12/805 (2013.01)

F I
G06F 13/00 353C
G06F 13/00 650B
H04L 12/805

請求項の数 12　(全 17 頁)

(21)出願番号　特願2016-508045 (P2016-508045)
(86)(22)出願日　平成26年4月21日 (2014.4.21)
(65)公表番号　特表2016-522478 (P2016-522478A)
(43)公表日　平成28年7月28日 (2016.7.28)
(86)国際出願番号　PCT/EP2014/001052
(87)国際公開番号　WO2014/173521
(87)国際公開日　平成26年10月30日 (2014.10.30)
審査請求日　平成27年10月16日 (2015.10.16)
審判番号　不服2017-3 (P2017-3/J1)
審判請求日　平成29年1月4日 (2017.1.4)
(31)優先権主張番号　1307340.8
(32)優先日　平成25年4月23日 (2013.4.23)
(33)優先権主張国　英国 (GB)

早期審査対象出願

(73)特許権者　513156386
グルロジック　マイクロシステムズ　オーワイ
Ｇｕｒｕｌｏｇｉｃ　Ｍｉｃｒｏｓｙｓｔｅｍｓ　Ｏｙ
フィンランド共和国　２０１００　トゥルク　リンナンカツ　３４
Ｌｉｎｎａｎｋａｔｕ　３４　２０１００　Ｔｕｒｋｕ　ＦＩＮＬＡＮＤ
(74)代理人　100127188
弁理士　川守田　光紀
(72)発明者　カルッカイネン　トゥオマス　ミカエル
フィンランド共和国　ＦＩ－２０３２０　トゥルク　ラウタランカツ　２　Ｂ１７

最終頁に続く

(54)【発明の名称】ＨＴＴＰを利用する通信システム

(57)【特許請求の範囲】

【請求項１】

ＨＴＴＰに準拠する通信をサポートするように動作可能であり、ＨＴＴＰに関連するＧＥＴメソッド及びＰＯＳＴメソッドの組合せを利用することによって、システムの２つのノードの間に双方向リアルタイム通信リンクを確立するように動作可能である通信システムであって、

前記双方向リアルタイム通信リンクを介したデータ交換は、転送符号化を利用して、チャンク様式として実装され、

前記双方向リアルタイム通信リンクを介して伝送されるデータチャンクに関する最大セグメントサイズ（ＭＳＳ）を、通信リンク中で最も小さな最大セグメントサイズに基づいて決定すること、及び、ネーグルアルゴリズムを無効化することを特徴とする、

通信システム。

【請求項２】

前記双方向リアルタイム通信リンクが、ＨＴＴＰと関連するＣＯＮＮＥＣＴメソッドを利用することによってトンネルされるＴＣＰ／ＩＰ及び／又はＵＤＰである、請求項１に記載の通信システム。

【請求項３】

前記双方向リアルタイム通信リンクが、前記双方向リアルタイム通信リンクを提供する受信接続及び送信接続を含み、空のデータチャンクを受信するまで、それら接続がオープンに維持される、請求項１に記載の通信システム。

【請求項４】
前記双方向リアルタイム通信リンクが、前記双方向リアルタイム通信リンクを介して伝送されるデータの暗号化を利用するように動作可能である、請求項１に記載の通信システム。
【請求項５】
前記双方向リアルタイム通信リンクが、グラフィックデータ、画像データ、ビデオデータ、音声データ、テキストデータ、非構造化データのうちの少なくとも１つの通信を提供するように動作可能である、請求項１に記載の通信システム。
【請求項６】
ＨＴＴＰに準拠する通信をサポートするように動作可能である通信システムを介して通信リンクを確立する方法であって、
（ａ）ＨＴＴＰに関連するＧＥＴメソッド及びＰＯＳＴメソッドの組合せを利用することによって、システムの２つのノードの間に双方向リアルタイム通信リンクを確立する通信システムを用いることと；
（ｂ）前記双方向リアルタイム通信リンクを介して、チャンク様式でデータを交換することと；
（ｃ）前記双方向リアルタイム通信リンクを介して伝送されるデータチャンクに関する最大セグメントサイズ（ＭＳＳ）を、通信リンク中で最も小さな最大セグメントサイズに基づいて決定すること、及び、ネーグルアルゴリズムを無効化することと；
を含み、前記交換することは、転送符号化を利用して実装される、方法。
【請求項７】
前記方法は、ＨＴＴＰと関連するＣＯＮＮＥＣＴメソッドを利用することによって双方向通信リンクのＴＣＰ／ＩＰ及び／又はＵＤＰトンネリングすることを含む、請求項６に記載の方法。
【請求項８】
前記双方向リアルタイム通信リンクが、前記双方向リアルタイム通信リンクを提供する受信接続及び送信接続を含み、空のデータチャンクを受信するまで、それら接続がオープンに維持される、請求項６に記載の方法。
【請求項９】
前記双方向リアルタイム通信リンクが、前記双方向リアルタイム通信リンクを介して伝送されるデータの暗号化を利用するように動作可能である、請求項６に記載の方法。
【請求項１０】
前記双方向リアルタイム通信リンクが、グラフィックデータ、画像データ、ビデオデータ、音声データ、テキストデータ、非構造化データのうちの少なくとも１つの通信を提供するように動作可能である、請求項６に記載の方法。
【請求項１１】
装置の処理手段に実行されると、前記装置に、請求項６から１０のいずれかに記載の方法を遂行させるように構成されるプログラム命令を備える、コンピュータプログラム。
【請求項１２】
ＨＴＴＰで表現され、かつＨＴＴＰに従って動作する通信ネットワークのサーバ上で実行可能である、請求項１１に記載のコンピュータプログラム。
【発明の詳細な説明】
【技術分野】
【０００１】
本願は、通信システム、例えば、リアルタイム・ハイパーテキスト・トランスファー・プロトコル（ＨＴＴＰ）を利用することによりグラフィックデータ、画像データ、ビデオデータ、音声データ等の各種デジタルデータを伝送する通信システムに関する。さらに、本願は、上記通信システムを動作させることにより各種データを伝送する方法にも関連する。さらに、本願は、装置の処理手段に実行されると、当該装置に上記方法を遂行させるように構成されるプログラム命令を備えるコンピュータプログラムにも関連する。

【背景】
【０００２】
概して、ハイパーテキスト・トランスファー・プロトコル（ＨＴＴＰ）は、現代のインターネットを実装するために広く用いられている。該プロトコルは、分散協調ハイパーメディア情報システムのためのアプリケーションプロトコルである。実装においては、ＨＴＴＰは、ネットワークを定義する論理リンクを用いて該ネットワークを構築するように動作可能である多重線形セットのオブジェクトであり、上記リンクは、しばしば、ノード間のネットワーク関係を定義する「ハイパーリンク」とされる。
【０００３】
ＨＴＴＰは、インターネットに実装されるクライアントサーバモデル等における要求応答プロトコルとして機能するように動作可能である。当該モデルにおいて、任意に、ウェブブラウザが、クライアントを実装するのに用いられ、サーバ上で実行されるソフトウェアアプリケーションがウェブサイトをホストしてもよい。動作においては、所定のクライアントが、ＨＴＴＰ要求メッセージをサーバに送信し、該サーバは、ＨＴＭＬファイル及びその他コンテンツ等のリソースを提供することにより応答し、又は上記クライアントの代わりにデータ処理機能を実行し、若しくは上記クライアントに応答メッセージを返しさえする。上記ウェブブラウザは、様々な方法で実装されることが可能であり、例えばユーザーエージェントとして、ウェブクローラとして、又はインターネット由来データコンテンツにアクセスし、利用し、又は表示する、コンピュータハードウェア上で実行可能なその他任意のソフトウェアとして実装され得る。
【０００４】
ＨＴＴＰは、隣接したネットワーク要素がクライアントとサーバとの間における通信を可能にすることができるように設計されている。インターネットの高トラフィックウェブサイトは、しばしば、上流サーバの代わりにコンテンツを配信し、データ及び／又はサービスの配信に関する応答時間を向上させるように動作可能であるウェブキャッシュサーバを利用する。さらに、プライベートネットワーク境界におけるＨＴＴＰプロキシサーバは、グローバルにルーティング可能なインターネットアドレス無しに、即ち外部サーバを介してメッセージを中継することによって、クライアントのための通信を容易にするために有利に用いられる。
【０００５】
ＨＴＴＰリソースは、ユニフォームリソースロケータ（Ｕｎｉｆｏｒｍ Ｒｅｓｏｕｒｃｅ Ｌｏｃａｔｏｒｓ；ＵＲＬ)とも称されるユニフォームリソースアイデンティファイア（Ｕｎｉｆｏｒｍ Ｒｅｓｏｕｒｃｅ Ｉｄｅｎｔｉｆｉｅｒｓ；ＵＲＩ）を用いることによって所定のネットワーク上で識別され、かつそこに配置される。さらに、ＵＲＩ及びハイパーリンクは、相互に関連付けされたハイパーテキスト文書のウェブを形成することが可能であるハイパーテキスト・マークアップ言語（ＨＴＭＬ）で表現される。
【０００６】
ＨＴＴＰセッションは、一連のネットワーク要求／応答トランザクションによって実装される。例えば、ＨＴＴＰクライアントは、サーバ上の特定のポートに対して伝送制御プロトコル（Ｔｒａｎｓｍｉｓｓｉｏｎ Ｃｏｎｔｒｏｌ Ｐｒｏｔｏｃｏｌ；ＴＣＰ）接続を確立することによって要求を開始する。ＨＴＴＰサーバは、クライアントの要求メッセージをリッスンし、関連するメッセージと共に「ＨＴＴＰ/１．１　２００　ＯＫ」等の状態表示行を返送することによって応答する。この関連するメッセージの本文は、しばしば、要求リソースであるが、エラーメッセージが代わりに返される場合もある。
【０００７】
ＨＴＴＰは、便宜上「動詞」と称される、識別リソースに関して実行され得る所望のアクションを示すメソッドを定義する。その識別リソースとは、例えば、１又は複数のサーバ上に存在する実行可能なオブジェクトに由来するデータファイル又は出力である。ＨＴＴＰ「動詞」としても知られるＨＴＴＰメソッドの例が、表１に提供されている。
【０００８】

【表１】

表１：ＨＴＴＰメソッド（ＨＴＴＰ「動詞」）

「動詞」	詳細
ＧＥＴ（ゲット）	指定したリソースの表示を要求する。「ＧＥＴ」を用いた要求はデータのみ取得するものでなければならない。
ＨＥＡＤ（ヘッド）	ＧＥＴから得られるものと同一である応答を要求するが、応答は如何なる本体も欠いている。即ち、「ＨＥＡＤ」は、しばしば、効率的な態様でメタデータを取得するために利用される。
ＰＯＳＴ（ポスト）	所定のサーバが、ＵＲＬによって識別される所定のウェブリソースの新規下位リソースとして要求に包含されるエンティティを受け入れることを要求する。
ＰＵＴ（プット）	包含されるエンティティを、供給したＵＲＩ（ＵＲＬ）に関して保存することを要求する。ＵＲＩが既存のリソースを参照するものである場合、そのリソースは変更される。
ＤＥＬＥＴＥ（デリート）	指定したリソースの削除を要求する。
ＴＲＡＣＥ（トレース）	結果として、受信要求が所定のクライアントにエコーバックされる。
ＯＰＴＩＯＮＳ（オプションズ）	所定のＵＲＬに関連するサーバによってサポートされるＨＴＴＰメソッドを返す。
ＣＯＮＮＥＣＴ（コネクト）	例えば上述の非暗号化ＨＴＴＰプロキシを介してＴＬＳ及びＳＳＬで暗号化された通信（ＨＴＴＰｓ）を簡素化するために、要求した接続を透過的ＴＣＰ／ＩＰトンネルに変換する。デフォルトでは、ＨＴＴＰ接続は非暗号化されるのに対し、ＨＴＴＰＳ接続は暗号化される。
ＰＡＴＣＨ（パッチ）	所定のリソースに対する部分的な変更の適用を要求する。

【０００９】

このように、現代のウェブブラウザによって利用されている主な転送プロトコルは、上述のＨＴＴＰであり、幾つかの関連する「エコシステム」、並びにそれらが利用するソフトウェア、特にブラウザソフトウェアアプリケーションは、ＨＴＴＰを用いることなく機能することはできない。上述のとおり、ＨＴＴＰは、送信される要求（表１を参照）に基づくものであり、それら要求への応答では、一般的に、ＨＴＭＬページ、又は画像若しくは音声ストリーム／ファイル等のバイナリデータが、その要求を受信したことに応じて供

給される。

【００１０】

インターネットの複雑さにより、インターネット通信遅延、即ち「待ち時間」が、動作において生じる。このような遅延は、リアルタイム応答が要請されるような双方向（全２重）通信が所望される場合、具体的には殆ど遅延の無いビデオイメージ及び／又は音声の転送及び受信が所望される場合など、よりデータ交換要件が厳しい状況において問題を発生させ得る。インターネットを介した双方向通信は、ボイスオーバインターネットプロトコル（Ｖｏｉｃｅ－ｏｖｅｒ－Ｉｎｔｅｒｎｅｔ－Ｐｒｏｔｏｃｏｌ；ＶｏＩＰ)から、また、例えば、近時ではスカイプソフトウェア等（「Ｓｋｙｐｅ」（スカイプ）は登録商標である。）を用いて提供されるようなインターネットを利用したビデオ会議で知られている。

【００１１】

特定種類の通信ニーズに対処するためには、ウェブサイトhttp://tools.ietf.org/html/rfc6455に記載されるとおり、「ウェブソケット（ＷｅｂＳｏｃｋｅｔｓ）」として知られるプロトコルを利用することが知られている。以下の通信特性がそれによって実現可能である。

（ｉ）ウェブソケットがＨＴＴＰ／ＨＴＴＰＳトンネル内で利用され、この場合、ファイアウォールは、近年、ウェブブラウザ上で一般的に利用されていることから、ポート８０／４４３に対して既に開放されている、並びに

（ｉｉ）ウェブソケットは、全２重接続モードで利用され、１つのＴＣＰ接続のみがリアルタイムで両方向に伝送することが可能であり、つまり、データ配信の方向を変えることによって１つの接続を用いてデータを送受信することができる。

【００１２】

しかしながら、このようなウェブソケットは、ポートに依存し得るものであり、この事は、つまり、望ましくない制限が生じることを意味する。

【摘要】

【００１３】

本願は、改良した様式のＨＴＴＰ通信ネットワークを介した双方向データ通信を提供できる通信システムを提供することを目的とする。

【００１４】

さらに、本願は、ＨＴＴＰ通信ネットワークを介した双方向データ通信を提供する通信システムを動作させる改良法を提供することを目的とする。

【００１５】

本発明の第１の態様によれば、ＨＴＴＰを利用する通信をサポートするように動作可能である通信システムであって、該通信システムは、ＨＴＴＰに関連するＧＥＴメソッド及びＰＯＳＴメソッドの組合せを利用することによって、システムの２つのノードの間に双方向リアルタイム通信リンクを確立するように動作可能であり、かつ上記通信リンクを介したデータ交換が、チャンク様式として実装され、上記通信リンクを介して伝送されるデータチャンクに関する最大セグメントサイズ（ＭＳＳ）が、上記通信リンクをサポートする通信ネットワーク容量に応じて最適化されることを特徴とする、通信システムが提供される。

【００１６】

当該通信システムは、待ち時間が減少したリアルタイム双方向通信を提供することができる点で有利である。

【００１７】

任意に、ＣＯＮＮＥＣＴメソッドは、以下の３つの異なるタイプのシナリオにおいて用いることができる。

（ｉ）接続が、ターゲットにトンネルされる。これは、有利にはデフォルトのシナリオである。

（ｉｉ）接続が、ローカルホストを介してターゲットにトンネルされ、その結果、データが

ローカルサービス内の送信プロセスから転送プロキシプロセスに転送され、転送プロキシプロセスからデータがターゲットに送信される。このようなアプローチは、ウイルス対策ソフトウェアが、データを分析しないようにし、かつデータを不注意にブロックしたり又はさもなければデータに干渉しないようにすることが可能であることから有利である。
（iii）接続は、転送プロキシサーバにトンネルされ、該転送プロキシサーバは、次いで、データをそのターゲットにリダイレクトする。このようなアプローチは、負荷平衡化システム、即ちクライアントによって生じるネットワーク負荷がターゲットに最適な状態で分配されるシステムにおいて利用するのに有利である。例えば、直接接続を介するよりもバックボーンネットワークにおいてデータを送信するのがより早くなる。

【００１８】
任意に、上記通信システムにおいて、上記通信リンクが、上記双方向通信を提供する受信接続及び送信接続を含み、かつ、空のデータチャンク及び／又は空のマルチパートデータブロックを受信するまで、それら接続がオープンに維持される。

【００１９】
任意に、上記通信システムにおいて、上記通信リンクが、該通信リンクを介して伝送されるデータの暗号化を利用するように動作可能である。

【００２０】
任意に、上記通信システムにおいて、上記通信リンクが、グラフィックデータ、画像データ、ビデオデータ、音声データ、非構造化データのうちの少なくとも１つの通信を提供するように動作可能である。

【００２１】
本願の第２の態様によれば、ＨＴＴＰを利用する通信をサポートするように動作可能である通信システムを介して通信リンクを確立する方法であって、
当該方法は、
（ａ）ＨＴＴＰに関連するＧＥＴメソッド及びＰＯＳＴメソッドの組合せを利用することによって、システムの２つのノードの間に双方向リアルタイム通信リンクを確立する通信システムを用いること、
（ｂ）上記通信リンクを介して、チャンク様式でデータを交換すること、並びに
（ｃ）上記通信リンクをサポートする通信ネットワーク容量に応じて、上記通信リンクを介して伝送されるデータチャンクに関する最大セグメントサイズ（ＭＳＳ）を最適化すること、
を含む、方法が提供される。

【００２２】
任意に、上記方法においては、上記通信リンクが、上記双方向通信を提供する受信接続及び伝送接続を含み、かつ、空のデータチャンク及び／又は空のマルチパートデータブロックを受信するまで、それら接続がオープンに維持される。

【００２３】
任意に、上記方法においては、上記通信リンクが、該通信リンクを介して伝送されるデータの暗号化を利用するように動作可能である。

【００２４】
任意に、上記方法においては、上記通信リンクが、グラフィックデータ、画像データ、ビデオデータ、音声データ、非構造化データのうちの少なくとも１つの通信を提供するように動作可能である。

【００２５】
本願の第３の態様によれば、装置の処理手段に実行されることにより、前記装置に、本願の第２の態様による方法を遂行させるように構成されるプログラム命令を備えるコンピュータプログラムが提供される。

【００２６】
任意に、上記コンピュータプログラムは、ＨＴＴＰで表現され、かつＨＴＴＰに従って動作する通信ネットワークのサーバ上で実行可能である。

【００２７】
本発明は、通信システムが、該通信システムにおいて実行されるソフトウェア若しくはハードウェアファイアウォール、並びに／又はウイルス対策ソフトウェアアプリケーションにおいて追加のコンフィギュレーション（設定；構成）が必要とならないように、既知のＨＴＴＰ転送プロトコルを利用することによって、該システムが、非暗号化又は暗号化の双方向全２重通信を提供可能であるという点で有利である。
【００２８】
さらに、本発明は、通信アプリケーションの機能性及び信頼性を向上させることにより、システムに関係する技術的なメンテナンスの問題、例えばデータセキュリティ設定を簡素化するという点で有利である。
【００２９】
本発明の特徴は、添付の特許請求の範囲に規定される発明の要旨から逸脱することなしに、様々な組合せにおいて組合せることが可能であることが理解される。
【００３０】
以下の図面を参照して、本願の実施形態を例としてのみ以下に説明する。
【図面の簡単な説明】
【００３１】
【図１】ＨＴＴＰを利用するように動作可能である通信ネットワークの図である。
【図２】開示される方法のステップのセットの図である。
【図３】開示される方法のステップの別のセットの図である。
【開示される実施形態の説明】
【００３２】
添付の図面において、下線を付した番号は、その下線を付した番号が位置する要素、又はその下線を付した番号が隣接する要素を表すのに用いられる。下線が付されていない番号は、要素にその下線が付されていない番号を結合させる線によって特定されるその要素に関する。番号に下線が付されておらず、関連する矢印が付随する場合、下線が付されていないその番号が、その矢印が指し示す全体要素を特定するのに用いられる。
【００３３】
概して、図１を参照すると、以下に説明されるシステムが存在し、そのシステムの一部は５で全体的に示されており、並びに、利用されるＨＴＴＰの記述がＲＦＣ２６２１、ＲＦＣ２０６８及びＲＦＣ１９４５等の標準に従う様式で、双方向リアルタイム通信のＨＴＴＰに関し、遅延、つまり「待ち時間」を推定することが可能である関連する方法が存在する。通常、ＨＴＴＰは、第１及び第２ノード１０Ａ、１０Ｂの間においてリアルタイム双方向通信を可能にするように設計されておらず、所定のクライアントは、以下のような様式で、リアルタイムデータを送信すると同時にリアルタイムデータを受信することができる。
【００３４】
（ｉ）２つのノード１０Ａ、１０Ｂの間で利用される通信接続２０は、暗号化フォーマットで双方向通信をサポートするように動作可能であり、
（ｉｉ）ウイルス防止ソフトウェア３０が、通信接続２０を介して送信され、かつ受信されるコンテンツに干渉せず、
（ｉｉｉ）ファイアウォール６０は、インターネットトラフィック、即ち「ＷＷＷトラフィック」の全面的遮断が、例えば安全な金融取引に利用されるバンキング接続の状況において、ブロックされないかぎり、ネットワークトラフィックを阻むことができず、並びに
（ｉｖ）ネットワークデバイス、例えばブリッジ及びルータが、通信接続２０を介して伝送されるデータを分析せず、かつ該データに干渉することができない。
【００３５】
本願の実施形態は、以下の特徴を利用することによって機能性（ｉ）から（ｉｖ）に対処することが可能である。
（ａ）２つの相互に異なるタイプであるＧＥＴメソッド及びＰＯＳＴメソッドが用いられ

（表１を参照。）、ＧＥＴメソッドは、通信接続２０を介して受信接続を構築し、ＰＯＳＴメソッドは、通信接続２０を介して送信接続を構築する、
（ｂ）両方の接続が、現代のＨＴＴＰにおいて利用されるＣＯＮＮＥＣＴメソッドを用いてトンネルされ、並びに
（ｃ）以下により詳細に説明されるとおり、「チャンク」又はマルチパート転送符号化の形態が利用される。

【００３６】

好都合なことに、ＨＴＴＰがインターネットセッションに用いられ、ＧＥＴメソッド及びＰＯＳＴメソッドが、相互に独立した様式で利用される。例えば、ＧＥＴメソッドは、ウェブブラウザークライアントのホストとして機能するよう動作し得るウェブサーバからＨＴＭＬコンテンツを要求するために用いられ、ＧＥＴメソッドのための接続は、全ての応答データがホストからクライアントに配信されるまで、オープンのままである。さらに、データがクライアントからホストに配信される以外は、ＰＯＳＴメソッドと同様の接続手順が利用される（表１を参照）。

【００３７】

以下に説明される実施形態においては、通信は、所定のソケットが、例えば上述したウェブソケット等の既知のアプローチとそれら実施形態を区別する半２重様式で用いられる態様で実行される。それら実施形態では、データの送信及び／又は受信が、全２重接続よりも、より効率的である。何故ならば、ネットワークインターフェースカードが、受信と送信との間におけるそれらの入力／出力（Ｉ／Ｏ）状態を切り替える必要がないからである。既知の技術で利用されるこのような切り替えは、システムリソースを消費し、その結果、潜在的な通信速度を低下させる。

【００３８】

以下に説明の実施形態においては、ソケットは、受信モード又は送信モードの何れにおいても、ＨＴＴＰのＧＥＴメソッド及びＰＯＳＴメソッドのみの初期化の後に利用される。その結果、用いられるネットワークアダプターは、半２重状態においてのみ動作する必要があり、それによってネットワークインフラストラクチャ及びデバイスリソースを節約することができる。何故ならば、接続は、ＨＴＴＰ　ＧＥＴメソッド及び／又はＰＯＳＴメソッドのヘッダがネゴシエートされた後、その接続が終了するまで、送信モード又は受信モードの何れかにおいてのみ作動するからである。さらに、他の利益も生じ、例えば、ファイアウォール及びルータ、即ちハブ及びスイッチが受ける切り替えの負荷も少なく、従って、１つの全２重接続しか使用しない近年の既知の全２重通信アプローチと同じような速さではそれらに障害は起きないだろう。従って、以下に記載の実施形態は、例えば上記のウェブソケットよりもリソース効率がかなり良い。

【００３９】

上記既知のウェブソケットは、識別されていない接続タイプに属し、従って切断されているのでファイアウォールによって簡単に分析することができ、関連する接続がトンネルされるか否かに関わらず、それらの使用は回避され、又は制限される。以下に説明の実施形態においては、ＨＴＴＰプロトコルによるＧＥＴ接続又はＰＯＳＴ接続はＨＴＴＰプロトコルに従って機能し、そのためファイアウォールは、これらメソッドを利用する通信を制限又は回避することができない。

【００４０】

以下に説明の実施形態においては、ＴＣＰよりも実質３倍は速いと推定されるＵＤＰプロトコルが有利に利用される。任意に、それら実施形態では、ピアツーピア（Ｐ２Ｐ）接続を用いることができ、これによってアプリケーションレベルの通信が達成可能となる。

【００４１】

本明細書に記載の実施形態は、既知のＨＴＴＰ実装がＧＥＴメソッド及びＰＯＳＴメソッドの間に如何なるリンクも欠いているという点で、それら既知のＨＴＴＰ実装とは区別される。即ち、対照的に、本明細書に記載の実施形態では、リアルタイム全２重データ通信を提供するために、新しい様式で融合させたＧＥＴメソッド及びＰＯＳＴメソッドが利

用される。言及される全２重データ通信は、１つの受信接続及び１つの送信接続を用いることによって実装される。１つの受信接続又は１つの送信接続は、１つの半２重接続モード又は１つの全２重接続モードを用い得る。

【００４２】

以下、実施形態が、伝送制御プロトコル（Ｔｒａｎｓｐｏｒｔ Ｃｏｎｔｒｏｌ Ｐｒｏｔｏｃｏｌ；ＴＣＰ）に基づいて説明されるが、代わりにユーザーデータグラムプロトコル（Ｕｓｅｒ Ｄａｔａｇｒａｍ Ｐｒｏｔｏｃｏｌ；ＵＤＰ）が利用され得ることが理解される。ＵＤＰもＴＣＰも基本的なインターネットプロトコル（ＩＰ）に依るものであり、ＵＤＰデータグラムもＴＣＰセグメントもＩＰパケットで送信されるものであるが、ＵＤＰは、ネットワークアドレス変換（ｎｅｔｗｏｒｋ ａｄｄｒｅｓｓ ｔｒａｎｓｌａｔｉｏｎ；ＮＡＴ）トラバース手法を用いることによって、ローカルエリアネットワーク（ＬＡＮ）内だけでなく外部インターネットにおいてもアプリケーション間におけるピアツーピア通信の達成を可能にする無接続プロトコルであるという点で区別される。このようなアプローチを利用することによって、システム５においてサーバを介してデータを転送する必要を回避することができ、その結果、通信ネットワーク容量を顕著に節約することが可能となる。システム５においてＵＤＰを用いることから生じる更なる利益は、そのネットワーク通信容量の使用において、ＴＣＰよりも実質３倍効率が良いことである。何故ならば、ＵＤＰは、制御プロトコルではないからである。さらに、例えばシステム５を実装するために用いられる、ＩＰｖ４及びＩＰｖ６の両方の通信ネットワークにおいてバイトで測定されるＭＳＳ容量は、より大きいものとなる。何故ならば、ＵＤＰヘッダは、対応するＴＣＰヘッダよりも小さいからである。

【００４３】

ＧＥＴ接続及びＰＯＳＴ接続の両方に対してＴＣＰを用いることが、以下に説明されているが、任意に、これら接続のうちの一方のみがＴＣＰを用い、これら接続の他方はＵＤＰを用いることが理解される。さらに、ＧＥＴ接続もＰＯＳＴ接続もＵＤＰを利用し得ることが理解される。

【００４４】

送信側又は受信側におけるデータはまた、本発明の要旨から逸脱することなく、回線交換データからＩＰに基づくデータへと、またそれに相応してＩＰに基づくデータから回線交換データへと変化し得ることも理解される。

【００４５】

第１の例示実施形態においては、図２を参照すると、以下のとおり一連のステップが実行される。

【００４６】

ステップ１（Ｓ１）：クライアントからデータへの接続は、一意のストリーム識別情報（ＩＤ）を生成し、このＩＤは、ＧＥＴメソッドとＰＯＳＴメソッドとをペアにするのに利用され、その結果、そのデータ接続を実装するのに利用されるサーバは、ＧＥＴメソッド及びＰＯＳＴメソッドのペアが同一のクライアントに属することを認識する。利用される該ＩＤは、以下に、より詳細に説明する。しかしながら、ＧＥＴメソッド及びＰＯＳＴメソッドは、一意のストリーム識別情報（ＩＤ）が、送信接続及び受信接続を組み合わせるのに用いられる場合に本発明を限定するものではないことが理解される。そのストリームＩＤの基本的な目的は、サーバでクライアントの送信接続及び受信接続をバインドすることにあるとしても、それは、同時に、そのクライアントを認証し、かつ識別するためにも用いられ得る。この事は、サーバが、次いで、それらの処理が継続する前に、有害であり、誤りであり、及び／又は未識別の接続を破棄することが可能であることを意味する。このような機能性によれば、サーバを保護し、かつ未識別の接続要求及び不必要な計算よって生じるサーバ負荷を低減／削減することが可能になる。言い換えると、これにより、システムが、リソースを節約することが可能となり、これにより、エネルギーを節約することと、サーバ設備、特に負荷分散システムにおいて必要となるサーバ数を低減することの利益を提供できる。

【００４７】
ステップ２（Ｓ２）：クライアントは、次いで、サーバに対して、例えばそのデフォルトポート「８０」で、２つのＴＣＰ／ＩＰ接続を確立し、その後、クライアントは、ＣＯＮＮＥＣＴメソッドに関連するヘッダを送信する。動作において、ＣＯＮＮＥＣＴメソッドは、例えば、大抵は上述の非暗号化プロキシを介してＴＬＳ及びＳＳＬ暗号化通信（ＨＴＴＰ）を簡素化するために、要求データ接続を透明的ＴＣＰ／ＩＰトンネルに変換する。
【００４８】
ステップ１及び２を実装する場合は、様々な形態の暗号化が任意に利用され、例えば、ＳＳＬ１．０、ＳＳＬ２．０、ＳＳＬ３．０、ＴＬＳ１．０、ＴＬＳ１．１、ＴＬＳ１．２、又は同種の暗号化が利用される。しかしながら、上記トンネルは、異なる「エコシステム」の間において安全な通信を確保するためには、有利には透過的である。さらに、悪意のある攻撃又は干渉から保護されるハードウェアを利用することもまた有利である。本願の実施形態を実施するために利用される、このような透過的なトンネル接続は、ハッカー、敵意のあるソフトウェア、ウイルス対策ソフトウェア、ファイアウォールソフトウェア、又はデータトラフィックを監視し、かつ分析するように動作可能であるデバイス及び／若しくはソフトウェアが、トンネル接続を介して伝送されるデータに干渉することを回避することができる。
【００４９】
ステップ３（Ｓ３）：通信トンネルに関し利用される受信又は送信の接続に依存して、ＧＥＴメソッド又はＰＯＳＴメソッドのヘッダは、送信され、かつ受信され続ける。そのヘッダは、通信トンネルによって提供される所定の通信セッションに必要な情報を含む。さらに、該ヘッダは、有利には、従来型のデータ構造を利用するものであるが、以下のパラメータを含むものである。
【００５０】
（ｉ）結合／リンクされた接続に関する情報のストリームＩＤ種類、並びに
（ｉｉ）チャンク又はマルチパートのフォーマットとしての転送暗号化。
【００５１】
ヘッダに含まれる情報は、データが個々のデータブロックとして転送及び受信されることを確実にする。有利には、データの最大セグメントサイズ（ＭＳＳ）は、データチャンク又はマルチパートデータブロックのヘッダに用いられるバイト量を考慮し、通信トンネルをサポートするネットワーク容量に対して最適化され、その結果、データを転送及び受信する際に、バイトが喪失することがなく、これにより、信頼性のある、安全なデータ交換が提供される。
【００５２】
このようなネットワークの最適化は、例えば、接続されるクライアントデバイスをサーバに連結するネットワークからサーバに最大転送単位（Ｍａｘｉｍｕｍ Ｔｒａｎｓｆｅｒ Ｕｎｉｔ；ＭＴＵ）値を要求することによって実施される。それによって、通信ネットワークにおいて最弱通信リンクを識別することが可能であり、その後、その最弱通信リンクに関連するクライアントデバイスへの送信に関する最大セグメントサイズ（ＭＳＳ）が、その最弱通信リンクに対応し得る割合で設定される。このＭＳＳ値は、任意に、サーバによって、システムの他のクライアントデバイスに伝送される。このようなネットワーク最適化は、有利には、以下のステップを有する方法を用いて実施される。
【００５３】
ステップＡ：システムは、サーバをクライアントデバイスに連結する最弱通信リンクを決定する。例えば、所定のデータリンクに関するＭＴＵ値は１５００バイトである。このＭＴＵ値から、所定の数値のＴＣＰヘッダバイト、即ち４０バイトを減算すると、１４６０バイトが利用可能となる。これら１４６０バイトはＭＳＳに対応する。
【００５４】
ステップＢ：システムは、最弱通信リンクのＭＳＳを利用することによって所定のセッションに関するＭＳＳを決定する。

【００５５】
ステップＣ：任意に、システムにおいて利用されるネーグルアルゴリズム（Ｎａｇｌｅ ａｌｇｏｒｉｔｈｍ）は、システム内の輻輳制御を防止するために無効化されるが、即ち、これは、システムのソケットでＴＣＰ＿ＮＯＤＥＬＡＹオプションを設定することによって達成され、これはネーグルアルゴリズムを無効化するものである。このようなネーグルアルゴリズムの無効化が望ましい。何故ならば、ネーグルアルゴリズムは、対応するデータパケットが送信される前に、特定量バイトのデータが送信キューに追加されるまで待機するからである。ネーグルアルゴリズムが無効化された場合、システムは、上述のとおり、システムによってのみ決定されたサイズのデータパケットを送信することができる。
【００５６】
ステップ４（Ｓ４）：ＨＴＴＰ要求ヘッダが一旦送信され、対応して正常な応答がサーバから受信されると、２重のデータ受信及び送信が次いで開始される。これにより、サーバとの２つの接続、つまり受信接続及び送信接続が正常に活率され、これら接続は、空のデータチャンク又は空のマルチパートデータブロックが受信されるまでオープン状態で維持される。
【００５７】
次に、２つの例示実施形態がＨＴＴＰコードによって説明される。
【００５８】
実施例１：クライアントとサーバとの間に単純なトンネル受信接続を構築するために実行される際に動作するＨＴＴＰコードが提供され、ＩＰアドレス１９２．１６８．０．１０１を有するピアが、ＩＰアドレス１９２．１６８．０．１００を有するホストに接続する。ＨＴＴＰコードにおいては、「ＧＥＴ」メソッドと「ＣＯＮＮＥＣＴ」メソッドの両方が使用され、それと共にチャンク転送符号化が指定されていることが分かる。
【００５９】

<connect>

<send> CONNECT 192.168.0.100:80 HTTP/1.0 \r\n

<send> Mozilla/5.0 (Windows NT 5.0) Gurulogic \r\n

<send> \r\n

<send> GET /readstream? streamid=12345&param1=value1&param2=value2 HTTP/1.1 \r\n

<send> Host: 192.168.0.100 \r\n

<send> Transfer-Coding: chunked \r\n

<send> User-Agent : Mozilla/5.0 (Windows NT 5.0) Gurulogic \r\n

<send> \r\n

<recv> HTTP/1.1 200 OK \r\n

<recv> 5AD\r\n

<recv> 1453 bytes of data… \r\n

<recv> 5AD\r\n

<recv> 1453 bytes of data… \r\n

…

<recv> 5AD\r\n

<recv> 1453 bytes of data… \r\n

<recv> 0 \r\n

<disconnect from 192.168.0.100>

【００６０】

実施例２：クライアントとサーバとの間に単純なトンネル送信接続を構築するために実行される際に動作するＨＴＴＰコードが提供され、ＩＰアドレス１９２．１６８．０．１０１を有するピアが、対応するＩＰアドレス１９２．１６８．０．１００を有するホストに接続される。ＨＴＴＰコードにおいては、「ＰＯＳＴ」メソッドと「ＣＯＮＮＥＣＴ」メソッドの両方が使用され、それと共にチャンク転送符号化が指定されていることが分かる。

【００６１】

<connect to 192.168.0.100>

<send> CONNECT 192.168.0.100:80 HTTP/1.0 \r\n

<send> Mozilla/5.0 (Windows NT 5.0) Gurulogic \r\n

<send> \r\n

<send> POST /writestream?streamid=12345&param1=value1&param2=value2 HTTP/1.1 \r\n

<send> Host: 192.168.0.100 \r\n

<send> Transfer-Coding: chunked \r\n

<send> User-Agent : Mozilla/5.0 (Windows NT 5.0) Gurulogic \r\n

<send> \r\n

<send> 5AD\r\n

<send> 1453 bytes of data... \r\n

<send> 5AD\r\n

<send> 1453 bytes of data... \r\n

...

<send> 5AD\r\n

<send> 1453 bytes of data... \r\n

<send> 0 \r\n

<recv> HTTP/1.1 200 OK \r\n

【００６２】

これら２つの実施例１及び２においては、ＭＳＳは１４６０バイトであると想定されるので、実際上、最適化チャンクに関するデータサイズは、１４５３バイトとなる。最適化チャンクサイズは、方程式１において与えられる式を用いることによって、システムにおいて算出される。

【００６３】

ＭＳＳ＝（チャンク開始ヘッダ）－（チャンク終了ヘッダ）・・・方程式１

【００６４】

チャンク開始ヘッダは、例えば１６進表記の実際のデータチャンクの長さと、通常はキャリッジ・リターン（ＣＲ）とラインフィード（Ｌｆ）の両方である１又は複数の改行文字とからなる。チャンク終了ヘッダは、改行文字の終了に類似し、それら改行文字がデータチャンクを完成させる。

【００６５】

次に図２を参照すると、ステップ３（Ｓ３）、即ちＣＯＮＮＥＣＴメソッドを利用することによって接続トンネルを確立することは、図３に提供されるとおり、任意に省略される。トンネルを要しない場合には、接続トンネルは省略される。従って、通信が利用されない場合には、ステップ１、２及び４のみが利用される。さらに、図２に関しては、接続トンネルが、ＧＥＴ接続又はＰＯＳＴ接続に対してのみ構築され、即ち複数のノード間で非対称トンネル通信構成が構築され得る。つまり、任意に、通信トンネルは、ＧＥＴ接続又はＰＯＳＴ接続に対してのみ用いられる。

【００６６】

実施例３：ＭＳＳ最適化は、所定のデータチャンクによって提供される所定のペイロードにのみ依存する。何故ならば、対応するｈｔｔｐチャンクヘッダは、処理におけるその時点で既に除去されているのに対し、データブロックのペイロードは１００％であるからである。ここで、このようなＭＳＳ最適化は、基本的に、以下の概念に基づくものである。即ち、最大転送単位（ＭＴＵ）は、個別の送信バーストであり、それ自体は、層が渡すことができる最大プロトコルデータ単位であり、例えば１５００バイトである。また、ＭＳＳ（最大セグメントサイズ）は、ＭＴＵ－プロトコルヘッダと同等のデータサイズを有する。本願による技術の実施形態においては、ＭＳＳは、ちょうど、問題とされるネットワークの最弱通信リンクが送信し得るバイト量のデータを保有する。従って、本願による技術が用いられる場合には、データはより小さいパケットに分割されず、その結果、例えばＷｉＦｉ（ワイファイ)ネットワーク等において衝突及びパケット損失が生じることは少なくなる。

【００６７】

ＭＳＳ最適化の例は、以下のとおりである。

【００６８】

【表２】

クライアント１	クライアント１とクライアント２との間におけるオペレータ	クライアント２
（ネットワークのＭＴＵ１５００バイト）	（最弱通信リンクのＭＴＵ６００バイト）	（ネットワークのＭＴＵ１３００バイト）

【００６９】

接続構築の開始：

【００７０】

ＩＣＭＰピング（ＩＣＭＰ－ｐｉｎｇ）は、ネットワークをテストするために送信され、クライアント１とクライアント２との間の通信が、ＭＴＵ＞６００の場合に回避されることが検出される。従って、ＭＴＵは６００バイトに設定され、この事は、ＴＣＰヘッダの４０バイトが除外された後、つまり考慮された後のＭＳＳは５６０バイトであることを意味する。ＵＤＰプロトコルのヘッダはより小さく、従って、ＵＤＰが用いられる場合、ペイロードは相応してより大きくなる。

【００７１】

次いで、クライアント１は、６つのパートに分割される３０００バイトのパケットをクライアント２に送信する。このような分割は単純であり、有利には以下の式に従い実装される。全バイト量が、ネットワークにおける最小ＭＴＵ－開始及び終了チャンクヘッダで除算される。即ち、３０００／（５６０－（５＋２））＝５．４２パケットであり、これは、その他データが送信のために待ち行列に入れられない限り、最も近い整数単位のパケットに四捨五入される。

【００７２】

パケット１：５６０バイトが送信され、そのうちペイロードは５５３バイトである。
パケット２：５６０バイトが送信され、そのうちペイロードは５５３バイトである。
パケット３：５６０バイトが送信され、そのうちペイロードは５５３バイトである。
パケット４：５６０バイトが送信され、そのうちペイロードは５５３バイトである。
パケット５：５６０バイトが送信され、そのうちペイロードは５５３バイトである。
パケット６：５６０バイトが送信され、そのうちペイロードは２３５バイトである。

【００７３】

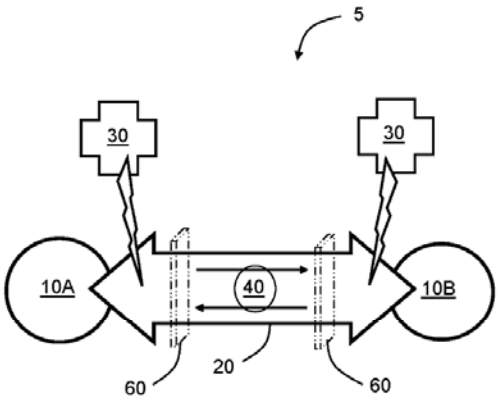
パケットが、分割されることなく、つまり１つの３０００バイトのパケットとして直接

送信された場合には、パケットは、ネットワーク上のオペレータのデバイスによって分割され、即ち断片化され、これは、時間を要するであろうし、また、問題を生じさせる可能性もあり、さらに欠損パケットを再送信することが必要になる可能性も生じるであろう。そして、これら全ての事が、送信側が、受信側の不安定なネットワークによって遅延が生じるために、新規パケットを送信する前に待機しなければならないことに繋がるであろう。

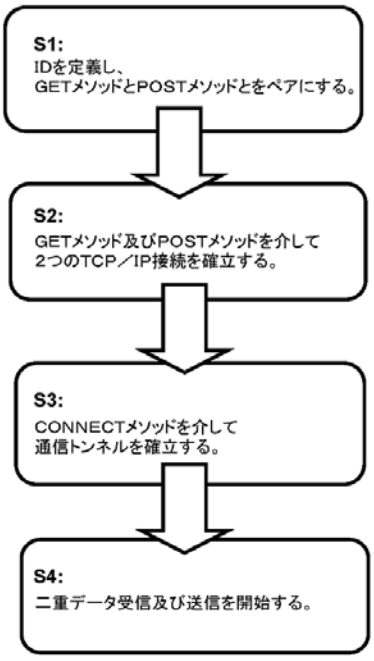
【００７４】

上記に記載した発明の実施形態に対する変更は、添付の特許請求の範囲によって定義される発明の要旨から逸脱することのない限り可能である。特許請求の範囲に図面の参照符号が記載される場合、それは請求項の理解を補助するためのものであり、請求項に係る発明を限定するものとして解釈してはならない。

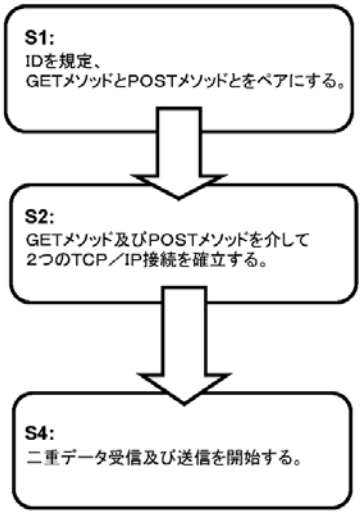
【図１】



【図２】



【図３】


S1:
IDを規定、
GETメソッドとPOSTメソッドとをペアにする。
S2:
GETメソッド及びPOSTメソッドを介して
2つのTCP／IP接続を確立する。
S4:
二重データ受信及び送信を開始する。

フロントページの続き

(72)発明者 ハッカライネン ヴァルッテリ
フィンランド共和国 ＦＩ－２０７２０ トゥルク クルマクヤ １Ｂ アー・エス ２
(72)発明者 カレヴォ オッシ
フィンランド共和国 ＦＩ－３７８００ アカー ケトゥンハンタ １

合議体
審判長 吉田 隆之
審判官 中野 浩昌
審判官 富澤 哲生

(56)参考文献 特開２００１－８６１６３（ＪＰ，Ａ）
特開２００３－１８２１６（ＪＰ，Ａ）
特開２００３－２４４１９４（ＪＰ，Ａ）
特開２００２－９４５６７（ＪＰ，Ａ）
Ｒ．Ｆｉｅｌｄｉｎｇ ｅｔ ａｌ．，”Ｈｙｐｅｒｔｅｘｔ Ｔｒａｎｓｆｅｒ Ｐｒｏｔｏｃｏｌ －－ ＨＴＴＰ／１．１”，ＲＦＣ２６１６，１９９９．６，<ＵＲＬ>ｈｔｔｐｓ：／／ｔｏｏｌｓ．ｉｅｔｆ．ｏｒｇ／ｒｆｃ／ｒｆｃ２６１６．ｔｘｔ
「ほんとうに使えるコマンドはコレ！ ネットワークコマンドランキングベスト１０ 第1位 ＰＩＮＧ」，Ｗｉｎｄｏｗｓ Ｓｅｒｖｅｒ Ｗｏｒｌｄ，日本，２００７．０１．０１発行，Ｖｏｌ．１２，Ｎｏ．１，ｐｐ４８－５２

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｇ０６Ｆ
Ｈ０４Ｌ