

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第5518954号
(P5518954)

(45) 発行日 平成26年6月11日(2014. 6. 11)

(24) 登録日 平成26年4月11日(2014. 4. 11)

(51) Int.Cl.	F I
HO 4 L 12/70 (2013. 01)	HO 4 L 12/70 A
HO 4 L 12/66 (2006. 01)	HO 4 L 12/66 A

請求項の数 16 (全 17 頁)

(21) 出願番号	特願2012-167317 (P2012-167317)	(73) 特許権者	500046438
(22) 出願日	平成24年7月27日(2012. 7. 27)		マイクロソフト コーポレーション
(62) 分割の表示	特願2005-175167 (P2005-175167)		アメリカ合衆国 ワシントン州 9805
	の分割		2-6399 レッドモンド ワン マイ
原出願日	平成17年6月15日(2005. 6. 15)		クロソフト ウェイ
(65) 公開番号	特開2012-257288 (P2012-257288A)	(74) 代理人	100107766
(43) 公開日	平成24年12月27日(2012. 12. 27)		弁理士 伊東 忠重
審査請求日	平成24年7月27日(2012. 7. 27)	(74) 代理人	100070150
(31) 優先権主張番号	10/881, 251		弁理士 伊東 忠彦
(32) 優先日	平成16年6月30日(2004. 6. 30)	(74) 代理人	100091214
(33) 優先権主張国	米国 (US)		弁理士 大貫 進介

最終頁に続く

(54) 【発明の名称】 セッション接続の持続

(57) 【特許請求の範囲】

【請求項 1】

クライアント計算装置とサーバ計算装置との間のセッション接続を持続するためのプロキシ計算装置によって実行される方法であって、

前記クライアント計算装置から前記プロキシ計算装置でハートビートメッセージを受信するステップであって、前記クライアント計算装置およびサーバ計算装置は、セッション開始プロトコルを使用してオープンセッションを有し、かつ、前記ハートビートメッセージは、トランスポート層プロトコルの有効なメッセージであって前記セッション開始プロトコルによって無視される、ステップと、

前記クライアント計算装置とサーバ計算装置との間の前記セッション接続のメッセージクロックをリセットするステップと、

前記プロキシ計算装置から前記サーバ計算装置に前記ハートビートメッセージを送信し、それによって、前記ハートビートメッセージを送信するステップは、前記クライアント計算装置とサーバ計算装置との間の前記セッション接続を維持する、ステップとを含むことを特徴とする方法。

【請求項 2】

前記メッセージクロックをリセットするステップは、前記プロキシ計算装置から前記サーバ計算装置に前記ハートビートメッセージを送信した後に実行することを特徴とする請求項 1 に記載の方法。

【請求項 3】

10

20

前記ハートビートメッセージは、改行およびキャリッジ・リターンの一つまたは複数を含む空白文字を含むことを特徴とする請求項 1 に記載の方法。

【請求項 4】

前記トランスポート層プロトコルは T C P / I P であることを特徴とする請求項 1 に記載の方法。

【請求項 5】

前記メッセージクロックは、前のメッセージが送信または受信されてから経過した時間を測定することを特徴とする請求項 1 に記載の方法。

【請求項 6】

前記メッセージクロックは、データ通信スタックのトランスポート層によって送信または受信されるメッセージに関することを特徴とする請求項 1 に記載の方法。

10

【請求項 7】

前記プロキシ計算装置は、エッジプロキシサーバであることを特徴とする請求項 1 に記載の方法。

【請求項 8】

前記クライアント計算装置はネットワークの外部に有し、かつ、前記サーバ計算装置は前記ネットワーク内に有することを特徴とする請求項 7 に記載の方法。

【請求項 9】

前記ハートビートメッセージを受信するステップ後一定期間を経て、前記クライアント計算装置から前記プロキシ計算装置で第 2 のハートビートメッセージを受信するステップと、

20

前記ハートビートメッセージを受信するステップと前記第 2 のハートビートメッセージを受信するステップとの間のタイムアウト時間が前記一定期間を超えているかを判定するステップと

をさらに含むことを特徴とする請求項 1 に記載の方法。

【請求項 10】

前記一定期間が前記ハートビートメッセージを受信するステップと前記第 2 のハートビートメッセージを受信するステップとの間の前記タイムアウト時間を超えていると判定すると、前記クライアント計算装置と前記サーバ計算装置との間の前記セッション接続をドロップするステップをさらに含むことを特徴とする請求項 9 に記載の方法。

30

【請求項 11】

前記一定期間が前記ハートビートメッセージを受信するステップと前記第 2 のハートビートメッセージを受信するステップとの間の前記タイムアウト時間を超えていないと判定すると、前記メッセージクロックをリセットし、それによって、前記セッション接続を維持するステップをさらに含むことを特徴とする請求項 9 に記載の方法。

【請求項 12】

クライアント計算装置とサーバ計算装置との間に通信可能に接続され、前記クライアント計算装置と前記サーバ計算装置との間のセッション接続を持続するように構成されているプロキシ計算装置であって、

前記クライアント計算装置との接続を通じて、メッセージを受信する前記プロキシ計算装置上で実行可能な第 1 のコンポーネントと、

40

前記受信したメッセージを前記サーバ計算装置に送信する前記プロキシ計算装置上で実行可能な第 2 のコンポーネントと、

前記受信したメッセージがハートビートメッセージであるかを判定する前記プロキシ計算装置であって、前記ハートビートメッセージは、トランスポート層プロトコルの有効なメッセージであってセッション開始プロトコルによって無視され、かつ、前記受信したメッセージがハートビートメッセージであると判定すると、前記クライアント計算装置と前記サーバ計算装置との間の前記セッション接続のメッセージクロックをリセットし、それによって、前記クライアント計算装置と前記サーバ計算装置との間の前記セッション接続を維持する、前記プロキシ計算装置上で実行可能な第 3 のコンポーネントとを含むことを

50

特徴とするプロキシ計算装置。

【請求項 1 3】

前記プロキシ計算装置は、エッジプロキシサーバであることを特徴とする請求項 1 2 に記載のプロキシ計算装置。

【請求項 1 4】

セッション接続を持続するためのサーバ計算装置によって実行される方法であって、
セッション開始プロトコルを使用してクライアント計算装置とのセッションを確立する
ステップであって、前記セッションはセッション接続を有する、ステップと、

プロキシ計算装置を介して、前記クライアント計算装置から、トランスポート層プロト
コルを使用してハートビートメッセージを受信するステップであって、前記セッション開
始プロトコルは、前記トランスポート層プロトコルよりも高いデータ通信スタックの層に
よって用いられ、かつ、前記ハートビートメッセージは、前記トランスポート層プロトコ
ルの有効なメッセージであって前記セッション開始プロトコルによって無視され、前記ハ
ートビートメッセージを受信するステップは、前記サーバ計算装置で前記セッション接続
を維持する、ステップと、

を含み、

前記セッションは、前記プロキシ計算装置を通じて、前記クライアント計算装置と前記
サーバ計算装置との間で確立されて維持され、かつ、前記クライアント計算装置から前記
サーバ計算装置で前記ハートビートメッセージを受信するステップの後に、メッセージク
ロックがリセットされる、ステップとを含むことを特徴とする方法。

【請求項 1 5】

前記ハートビートメッセージは、前記セッション開始プロトコルのメッセージが受信さ
れてから一定期間が経過したときに受信され、前記一定期間はタイムアウト値よりも短い
ことを特徴とする請求項 1 4 に記載の方法。

【請求項 1 6】

前記メッセージロックは前記プロキシ計算装置によってリセットされることを特徴と
する請求項 1 4 に記載の方法。

【発明の詳細な説明】

【技術分野】

【0001】

記載の技術は、一般には、データ・ネットワーク通信に関し、より詳細には、セッショ
ン接続の持続 (sustaining) に関する。

【背景技術】

【0002】

コンピューティング装置のユーザは時々、そのコンピューティング装置を使用して、他
のユーザと通信する。一例として、ユーザは、MICROSOFT MESSENGER
（「MESSENGER」）インスタント・メッセージング製品を使用するなどして、タ
イプされたメッセージを送信し、または受信することによって、コンピューティング装置
を使用する別のユーザと通信することができる。別の例では、ユーザは、ボイス・オーバ
・インターネット・プロトコル（「VoIP」：Voice over Internet Protocol）アプリ
ケーションを使用するなどして、マイクに向かって話し掛け、またスピーカまたはハンド
セットで他のユーザの話を聞くことによって、別のユーザと通信することができる。こ
うしたユーザは、パーソナル・コンピュータ、携帯情報端末、携帯電話、VoIP対応電話
など、様々なコンピューティング装置を使用することができる。

【0003】

ユーザへのこうしたタイプの通信サービスを提供するアプリケーションは、プレゼンス
（presence）情報を提供することも必要とすることがある。プレゼンス情報は、ユーザが
コンピューティング装置を使用して通信する能力または意思を伝える。プレゼンス情報は
、コンピューティング装置によって自動的に検出され、またはユーザによって能動的に選
択することができる。コンピューティング装置は、ユーザの「状況（status）」などのプ

プレゼンス情報を自動的に観察し、通信することができる。一例として、ユーザがどのコンピューティング装置にもログオンしていない、またはプレゼンス情報を観察し伝えるアプリケーションを使用（またはそれにログオン）していない場合、ユーザの状況は、「オフライン（off-line）」として自動的に指し示すことができる。ユーザがMESSENGERなど、プレゼンス情報を観察し伝えるアプリケーションを開始し、またはそれにログオンするときに、ユーザ状況は、「オンライン（on-line）」として自動的に指し示すことができる。ユーザがある時間の間、コンピューティング装置上で動作を行わない場合、アプリケーションは、ユーザが「不在」であることを自動的に指し示すことができる。ユーザは、その状態を能動的に選択することもできる。たとえば、ユーザは、「昼食のため外出中である（out for lunch）」または「多忙である（busy）」ことを指し示すことができ、これらは、他のユーザからのメッセージに直ぐに応答することができないことの指標（indication）であるとする事ができる。

10

【0004】

ユーザの状態を決定したいと望むアプリケーションである「ウォッチャー（watcher）」が、自動的にプレゼンス情報を受信するために登録サーバに問い合わせし、またはそれに加入することによって有意義なプレゼンス情報を決定することができるように、複数のコンピューティング装置が、登録サーバ・コンピューティング装置にプレゼンス情報を登録することができる。一例として、MESSENGERは、ユーザがコンピューティング装置上である時間の間操作を行っていないので、したがってユーザのプレゼンス情報は「不在（away）」である可能性があることを登録サーバに指し示すことができる。VOIP対応電話は、ユーザが電話を掛けたので、したがってユーザのプレゼンス情報は「電話中（on the telephone）」である可能性があることを登録サーバに指し示すことができる。ウォッチャーは、登録サーバからこのプレゼンス情報を受信するときに、ユーザ状況は「電話中であり、コンピューティング装置上のメッセージに応答することができない」と決定することができる。しかし、ユーザがコンピューティング装置上で動作を実施し始める場合、ウォッチャーは、ユーザ状況は「電話中であるが、コンピューティング装置上のメッセージに応答することができる」と決定することができる。ユーザが、ユーザ状況が「不在（away）」であり、またVOIP対応電話を使用していることをコンピューティング装置上に能動的に指し示す場合、ウォッチャーは、ユーザ状況は「ローミング中（roaming）」であると決定することができる。

20

30

【0005】

通信アプリケーションは時々、コンピューティング装置間のセッションを確立し管理することが必要である。セッションは、ある時間にわたって発生する、コンピューティング装置間の1組の対話である。一例として、MESSENGERやVOIPなどのリアルタイム通信アプリケーションは、ユーザに代わって通信装置間のセッションを確立する。こうしたアプリケーションは、「セッション開始プロトコル」（「SIP」：Session Initiation Protocol）など、セッションを確立するための様々な機構を使用することが可能である。SIPは、コンピューティング装置が、互いを発見し、コンピューティング装置間でセッションを確立し、変更し、終了するために使用することができるアプリケーション層の制御プロトコルである。

40

【0006】

アプリケーションは、情報を送信しまたは受信するために、SIPを別のプロトコルと共に使用することができる。SIPを他のプロトコルと共に使用することによって、アプリケーションは、セッションを作成し管理し、またセッション中に情報を交換することができる。情報を交換するためにSIPと共に用いられるプロトコルは、情報をメッセージにセグメント化することができる。一例として、VOIPアプリケーションは、長い話をより短いメッセージにセグメント化することができる。セッション中のメッセージ交換は、「ダイアログ（dialog）」と称される。SIPは、ダイアログのメッセージを移送するために、さらに下のレベルの通信層、たとえば伝送制御プロトコル／インターネット・プロトコル（「TCP/IP」：Transmission Control Protocol/Internet Protocol）を

50

使用することができる。このTCP/IPは、一般的に使用されるトランスポートおよびネットワーク層のプロトコルである。

【0007】

伝送制御プロトコル（「TCP」）は、コネクション型（connection-oriented）の信頼できる配信のトランスポート層プロトコルである。TCPは一般に、トランスポート層と記載され、これはアプリケーション層（SIPを使用するアプリケーションなど）とネットワーク層の間のインターフェースを提供する。アプリケーション層は、一般に、データのストリーム（複数バイトのデータなど）を送信または受信する際に、TCP層と通信する。TCPは、このデータストリームを、インターネットプロトコル（「IP」）などの、ネットワーク層で使用されるプロトコルによって運ばれることが可能なセグメントに編成する。こうしたデータ・セグメントは一般に、「パケット」、「フレーム」または「メッセージ」と称される。それぞれのメッセージは一般に、ヘッダおよびペイロードを含む。ヘッダは、メッセージをルーティングし、解釈するのに必要なデータを含む。ペイロードは、送信されまたは受信される実際のデータを含む。他の層と共に、アプリケーション層、トランスポート層およびネットワーク層はまとめて、データ通信スタックと称される。

10

【0008】

ある接続内のメッセージは、中間のコンピューティング装置を通過する可能性がある。送信側コンピューティング装置のデータ通信スタックのトランスポートまたはネットワーク層が、受信側コンピューティング装置内のデータ通信スタックの対応する層と接続を確立するときに、コンピューティング装置によって交換されるメッセージは、複数の中間コンピューティング装置を通過する可能性がある。たとえば、メッセージは、プロキシ・サーバ、ネットワーク・アドレス・トランスレータ、またはゲートウェイを通過する可能性がある。こうした中間コンピューティング装置は、メッセージを受信し、処理し、転送する。一例として、プロキシ・サーバは、イントラネットに接続されたコンピューティング装置と、インターネットに接続されたコンピューティング装置の間の仲介役であることが可能である。このプロキシ・サーバは、たとえば、イントラネットの外部からの接続要求を無視するなどによって、イントラネットをより安全なものにしようと試み得る。

20

【0009】

中間コンピューティング装置は、ネットワーク・リソースを効率的に使用するために、2つのコンピューティング装置間のアイドル状態の接続を終了させることがある。一例として、イントラネットに接続されたユーザ・コンピューティング装置（「コンピューティング装置A」）上で実行されるMESSENGERまたはVOIPアプリケーションは、インターネットに接続された別のコンピューティング装置（「コンピューティング装置B」）上で実行されるアプリケーションとSIPベースのセッションを確立することができる。このセッションは、コンピューティング装置AとBの間のTCP/IP接続を使用することができる。コンピューティング装置によって交換されるメッセージは、プロキシ・サーバを通過することになる。ある時間の間、接続内でメッセージが交換されない場合、プロキシ・サーバは、ネットワーク・リソースを保存するために、それとコンピューティング装置Bの間の接続を閉じることができる。しかし、コンピューティング装置Aは、プロキシ・サーバとのその接続は開いているので、その接続が閉じていることを認識しないことがある。その結果、コンピューティング装置B上で実行されるアプリケーションは、コンピューティング装置Aのアプリケーション（またはユーザ）がもはやオンライン状態ではなく、または接続を使用していないと、誤って決め込む可能性がある。さらに、プロキシ・サーバが接続されるイントラネットをより安全なものにするために、プロキシ・サーバは、インターネットに接続されたコンピューティング装置など、イントラネットに接続されていない装置からの接続要求を無視することができる。したがって、インターネットに接続されたコンピューティング装置B上で実行するアプリケーションは、その状況を求めて、またはセッションの再確立を要求するために、コンピューティング装置A上で実行されるアプリケーションに問い合わせることが不可能になる可能性がある。

30

40

50

【0010】

中間のコンピューティング装置は、コンピューティング装置が「キープ・アライブ」機構を使用する場合であっても、接続を閉じる可能性がある。コンピューティング装置によっては、キープ・アライブ機構を使用して、メッセージが無いにも拘らず、接続を存続状態に保つ。この機構は、送信側が接続を閉じていないことを指し示すための「キープ・アライブ」メッセージを送信することを伴う。しかし、効率的な中間コンピューティング装置は、キープ・アライブ・メッセージを認識することができ、送信側コンピューティング装置と中間コンピューティング装置の間の接続を存続状態に保ちながら、キープ・アライブ・メッセージ内に示された受信側（インターネットに接続されたコンピューティング装置など）との接続を終了させる可能性がある。

10

【0011】

接続は、アプリケーション層の有効なメッセージを送信することによって持続することができる。一例として、トランスポート層のTCPを使用するSIPアプリケーションは、REGISTERメッセージなどの有効なSIPメッセージを定期的に送信することができる。REGISTERメッセージを使用して、SIPサーバが、メッセージ送信側のユニフォーム・リソース識別子（「URI」：Uniform Resource Identifier）を、送信側によって使用されるコンピューティング装置に関連付けることを可能にすることができる。有効なSIPメッセージが送信したりまたは受信したりすることができない場合、SIPアプリケーションは、そのTCP接続が使用不可能であることを検出することができる。

20

【0012】

SIPは、インターネット提案規格である。その仕様、「RFC3261」は、非特許文献1で入手可能である。イベント通知に関するSIP拡張に関する仕様、「RFC3265」は、非特許文献2で入手可能である。インスタントメッセージング・システム内のプレゼンス情報に関する仕様、「RFC2778」は、非特許文献3で入手可能である。SIP内のプレゼンス情報に関する提案仕様のドラフトは、非特許文献4で入手可能である。その4つのすべての仕様全体を本明細書中で参考として援用する。

【先行技術文献】

【非特許文献】

【0013】

30

【非特許文献1】 <http://www.ietf.org/rfc/rfc3261.txt>【非特許文献2】 <http://www.ietf.org/rfc/rfc3265.txt>【非特許文献3】 <http://www.ietf.org/rfc/rfc2778.txt>【非特許文献4】 <http://www.ietf.org/internet-drafts/draft-ietf-simple-presence-10.txt>

【発明の概要】

【発明が解決しようとする課題】

【0014】

REGISTERの送信、または実際に有効なデータを含む任意の有効なSIPメッセージを送信することに関する問題は、複数のクライアントおよびサーバが接続を持続するために送信することを必要とする場合に、送信することは、計算処理上集中的（computationally intensive）になる可能性があることである。REGISTERメッセージの場合、こうしたメッセージを受信するサーバは、送信側のコンピューティング装置のURIおよびIDを決定するために、メッセージの解析を必要とすることがあり、またおそらくは別のサーバ上にあるデータベース内にデータを格納することをさらに必要とする可能性がある。

40

【0015】

したがって、TCPキープ・アライブに依存したり、またはメッセージに集中的な計算処理を要しないセッション接続を持続することに対する有効な手法は、非常に有用である。

50

【課題を解決するための手段】

【0016】

セッション接続を持続するための技術が、ある実施形態で、提供される。この技術は、タイム・アウト条件のため、メッセージを送信しないことがセッション接続を閉じる原因になる可能性がある場合に、ハートビート・メッセージを送信する。ハートビート・メッセージは、データ通信スタックのより高いレベルのプロトコルによって無視されることになる有効なトランスポート層メッセージである。一例として、この技術は、ペイロード内にキャリッジ・リターンおよび改行（「CRLF」：carriage return and line feed）だけを含むTCPメッセージを送信することができる。TCP層は、CRLFだけを含むメッセージを有効なTCPメッセージであると見なすので、プロキシ・サーバなどの中間コンピューティング装置は、ハートビート・メッセージを「キープ・アライブ」メッセージとして解釈しないことができ、セッション接続を持続することができる。

10

【図面の簡単な説明】

【0017】

【図1】セッション接続を持続するためのシステムの一実施形態を示すブロック図である。

【図2】図1のネットワークの一実施形態を示すブロック図である。

【図3】コンピューティング装置によって使用されるデータ通信スタックの一実施形態を示すブロック図である。

【図4】send_messageルーチンの一実施形態を示す流れ図である。

20

【図5】sustain_connectionルーチンの一実施形態を示す流れ図である。

【図6】send_heartbeatサブ・ルーチンの一実施形態を示す流れ図である。

【図7】receive_messageサブ・ルーチンの一実施形態を示す流れ図である。

【図8】ある実施形態におけるメッセージ・フローを示すメッセージ流れ図である。

【図9】ある実施形態におけるメッセージ・フローを示すメッセージ流れ図である。

【図10】ある実施形態におけるメッセージ・フローを示すブロック図である。

【発明を実施するための形態】

30

【0018】

セッション接続を持続するための技術が、ある実施形態で、提供される。この技術は、タイム・アウト条件のため、メッセージを送信しないことがセッション接続を閉じる原因になる可能性がある場合に、ハートビート・メッセージを送信する。ハートビート・メッセージは、データ通信スタックのより高いレベルのプロトコルによって無視されることになる空でないペイロードを収容する有効なトランスポート層メッセージである。一例として、この技術は、ペイロード内にキャリッジ・リターンおよび改行（「CRLF」）だけを含むTCPメッセージを送信することができる。TCP層は、CRLFだけを含むメッセージを有効なTCPメッセージであると見なすので、プロキシ・サーバなどの中間コンピューティング装置は、メッセージが送信された、または受信された接続に関するメッセージ・クロックを再設定することができる。メッセージ・クロックは、接続内でメッセージが最後に送信されまたは受信されてから経過した時間量を指し示す。ある接続についてのメッセージ・クロックが閾値の時間量が経過したことを示す（たとえばタイム・アウト状態が発生したことを指し示す）場合、中間コンピューティング装置は、接続を閉じる可能性がある。メッセージ・クロックはメッセージが送信されまたは受信されるときに再設定（reset）されるので、閾値の時間量が経過する前に別のメッセージが送信されまたは受信されないという場合を除いては、接続は閉じられ得ない。受信側コンピューティング装置のTCP層が受信されたハートビート・メッセージをデータ通信スタックのより上の層に転送した後、その上位層は、ハートビート・メッセージを無視することができる。一例として、別な例では、SIPは、スペース、タブ、キャリッジ・リターン、改行またはC

40

50

R L Fなどの空白 (white space) だけを含む、有効な S I Pメッセージを無視することができる。その上位層は、C R L F以外の情報を含むメッセージを受信する場合には、無視することができない情報を含むものとして、こうしたメッセージを解釈しようと試みることができる。様々な実施形態において、ハートビートは、キャリッジ・リターン、改行、あるいはトランスポート層では有効であるが、データ通信スタックのより上の層のプロトコルによっては無視される任意のデータだけを含むことができる。したがって、S I Pによっては無視されるハートビート・メッセージなどの有効なトランスポート層メッセージを送信することによって、この技術は、トランスポート層より上のデータ通信スタック層ではコンピューティング装置間で情報が交換されない場合にも、セッション接続を持続することができる。無視されるが有効であるメッセージの種類については、プロトコルの仕様または定義内で定めることが可能である。

10

【0019】

ある実施形態では、トランスポート層より上の層によって無視されるメッセージは、キャリッジ・リターン、改行、スペース、タブまたは任意の空白文字などを一般に含むことが可能である。ある実施形態では、空白文字だけを含むこうしたメッセージは、キャリッジ・リターンまたは改行で終了することができる。過度な計算上の負担をもたらさずに接続を持続するためにメッセージ内で使用することができる文字は、接続を閉じることができるプロトコル (T C Pなど) より上で使用されるプロトコルを分析することによって決定することができる。

【0020】

20

次に図面に移ると、図1は、セッション接続を持続するためのシステムの一実施形態を示すブロック図である。このシステムは、1つまたは複数のクライアント102、1つまたは複数のネットワーク104、およびクライアントと前記ネットワークを接続するためのネットワーク、たとえばインターネット114を含む可能性がある。クライアントは、ネットワーク接続108を介してインターネットに接続することができる。ネットワーク104は、様々な形のデータ通信リンク110を使用して、インターネットに接続することができる。クライアントは、コンピュータ、携帯電話、またはインターネットに接続された他の装置などであることが可能である。ネットワークは、イントラネットなどであることが可能である。クライアントは、私設のデータ通信リンク (private data communications links) 112を使用して、ネットワークに直接に接続されることもできる。同様に、ネットワークは、私設のデータ通信リンク112を使用して、他のネットワークに直接に接続することもできる。インターネットは、様々な他のコンピューティング装置 (図示せず) に接続されることもできる。本明細書で述べる接続は、物理接続、無線接続、また実際には任意の形のデータ通信ネットワーク接続である可能性がある。

30

【0021】

図2は、図1のネットワークの一実施形態を示すブロック図である。ネットワーク200は、1つまたは複数のクライアント202、サーバ204およびエッジ・プロキシ・サーバ212を含む。このネットワークは、他の装置 (図示せず) をも含むことができる。このネットワークの図示された装置 (および図示されていない装置) は、このネットワークに関連するデータ通信ネットワーク (図示せず) を介して相互接続することができる。クライアント202およびサーバ204は、S I Pメッセージの交換を含めて、様々な動作を実施することができる。エッジ・プロキシ・サーバ212は、ネットワーク200 (「内部」ネットワーク) と、インターネット (図示せず) などの「外部」ネットワークの両方に接続される。クライアントがこのネットワークの外部のサーバとのセッションを開くことを望む場合、エッジ・プロキシは、たとえば、セッションが開かれるべきか、またどの通信リンクが使用されるべきかを決定することができる。

40

【0022】

図3は、図1および2のコンピューティング装置によって使用されるデータ通信スタックの一実施形態を示すブロック図である。コンピューティング装置は、互いに通信するために、1つまたは複数のデータ通信スタックを使用することができる。図示する例では、

50

コンピューティング装置 300 は、こうしたコンピューティング装置に関連するデータ通信スタックを使用して、コンピューティング装置 303 と通信する。コンピューティング装置 301 および 302 は、その関連するデータ通信スタックを使用して、コンピューティング装置 300 と 303 の間でメッセージを受信し、転送することができる。

【0023】

コンピューティング装置 300 は、アプリケーション層 304、トランスポート層 306、ネットワーク層 308、データリンク層 310 および物理層 312 を含むデータ通信スタックを有する。アプリケーション層は、SIP アプリケーションおよび他のアプリケーションを含むことができる。SIP アプリケーションは、トランスポート層と通信するコンポーネントまたは層を含むことができる。SIP アプリケーションに属するこの通信層は、ハートビート・メッセージを送信しまたは受信することができる。コンピューティング装置 301、302 および 303 は、類似のデータ通信スタックを有することになる。具体的には、コンピューティング装置 303 は、アプリケーション層 314、トランスポート層 316、ネットワーク層 318、データリンク層 320 および物理層 322 を含むデータ通信スタックを有することになる。こうしたアプリケーション層の一部またはすべては、SIP アプリケーションをさらに含むことができる。

【0024】

2つのコンピューティング装置が接続される場合、そのデータ通信スタックの各層は、互いの間の論理的または物理的な接続を有することができる。一例として、コンピューティング装置 300 がコンピューティング装置 301 および 302 を介してコンピューティング装置 303 に接続される場合、物理層 312 は、物理接続 324 を介してコンピューティング装置 301 の物理層に接続されることができ、このコンピューティング装置 301 は、物理接続を介してコンピューティング装置 302 の物理層に接続されることができ、またこのコンピューティング装置 302 は、物理接続を介してコンピューティング装置 303 の物理層に接続されることができる。コンピューティング装置はそれぞれ、たとえばモデム、ネットワーク・インターフェース・カード、またはインターネットなどの他の接続を介して互いに接続することができる。インターネットは、電話回線、マイクロ波リンク、ならびにメッセージのための複数のルートを共に形成する他の形の有線および無線接続のなど、様々な物理接続で構成される。実際に、インターネットの初期の目標は、送信者がネットワーク障害に拘らず受信者と通信できるようにすることであった。コンピューティング装置 300 がコンピューティング装置 303 にデータを送信するときに、そのデータは、インターネットを介してそれぞれ異なる物理ルートを移動することが可能である。

【0025】

データリンク層 310 および 320、ネットワーク層 308 および 318、トランスポート層 306 および 316、ならびにアプリケーション層 304 および 314 などのデータ通信スタックの高位層は、論理接続 326、328、330 および 332 などの論理接続によって接続することができる。データ通信スタックの2つの層が論理接続を有し、データを交換する場合、そのデータは、送信側コンピューティング装置のデータ通信スタックの「下方」、そのデータ通信スタックの物理層に送られ、物理接続を渡り、次いで、受信側コンピューティング装置のデータ通信スタックを「情報に」送ることになる。一例として、コンピューティング装置 300 の SIP アプリケーション 304 がコンピューティング装置 303 の SIP アプリケーション 314 と通信する場合、SIP アプリケーション 304 はトランスポート層 306 にデータを伝え、このトランスポート層はネットワーク層 308 にデータを伝え、このネットワーク層はデータリンク層 310 にデータを伝え、このデータリンク層は、物理層 312 にデータを伝えることになる。物理層 312 は、物理接続 324 を介してデータを通信することができ、最終的には物理層 322 にデータを伝えることになる。物理層 322 は、それが受信するデータをデータリンク層 320 に伝え、このデータリンク層は、ネットワーク層 318 にデータを伝え、このネットワーク層はトランスポート層 316 にデータを伝え、このトランスポート層は、最終的には S I

10

20

30

40

50

Pアプリケーション314にデータを伝えることができる。

【0026】

コンピューティング装置301および302は、その物理接続上でデータを受信するときに、そのデータ通信スタックの上方にデータを送ることになる。データ通信スタックのアプリケーション層より低いレベルは、受信データをそれより上の層に送信せずに、それを処理することができる可能性がある。一例として、SIPアプリケーション304は、SIPアプリケーション314にSIPメッセージを送信するときに、SIPメッセージに対応するデータをトランスポート層306に送ることになる。両方ともが物理層でデータを受信するコンピューティング装置301および302のデータ通信スタックは、そのデータ通信スタックの下から上に、そのそれぞれのトランスポート層に受信データを渡すことになる。データ内に示された宛先がコンピューティング装置303であることを受信データから認識するトランスポート層は、（その対応する下位レベルを介して）トランスポート層316に単にデータを転送することになる。したがって、コンピューティング装置301および302のアプリケーション/SIP層は、メッセージを処理しまたはメッセージを見ることさえ不要であることになる。

10

【0027】

ある時間期間にわたって、コンピューティング装置301または302のトランスポート層によってデータが受信されない場合、コンピューティング装置は、コンピューティング装置300と303の間のSIPセッションに対応する接続を閉じることになる。一例として、コンピューティング装置301は、コンピューティング装置300またはコンピューティング装置302から接続に関するメッセージを受信しなかった場合、その接続はもはや不要であると見なし、その接続を閉じることになる。したがって、コンピューティング装置301と303の間のさらなるSIPメッセージは、前に開いていたコンピューティング装置301および302を通過する接続を通過することが不可能になることになる。

20

【0028】

図4は、send_messageルーチンの一実施形態を示す流れ図である。このルーチンは、メッセージ送信する、SIPアプリケーションのコンポーネントなどの、アプリケーション層コンポーネントによって実施されることになる。send_messageルーチンは、あるコンピューティング装置上で実行されるSIPアプリケーションなどによって、それが別のコンピューティング装置上で実行されるSIPアプリケーションにメッセージを送信するときに使用することができる。このルーチンは、ブロック402で開始し、パラメータとしてメッセージの指標(indication)を受信する。ブロック404で、ルーチンは、クロック値を、たとえばそれを0に設定するなどによって、再設定する。クロック値は、メッセージが最後に受信されてから経過した時間の指標である。この値は、接続が閉じられるべきかどうかを決定するために、システムのコンポーネントによってチェックすることができる。ブロック406で、ブロック402で受信されたメッセージをトランスポート層に送信する。このルーチンは、メッセージを送信するトランスポート層によって公開されたアプリケーション・プログラム・インターフェースを使用することができる。ブロック408で、ルーチンは、その呼出し側に戻る。

30

40

【0029】

図5は、sustain_connectionルーチンの一実施形態を示す流れ図である。このルーチンは、接続を持続するために、SIPアプリケーションのコンポーネントなどのアプリケーション層コンポーネントによって実施される。接続は、メッセージが最後に受信されてから経過した時間量を示すクロック値がタイム・アウト値を超える場合などに、持続される必要があることになる。適切なタイム・アウト値は、トランスポート層のAPIを使用してトランスポート層の属性を決定するなどによって決定されることができ、またはシステムの管理者によって定義することができる。ルーチンは、ブロック502で開始する。ブロック504で、ルーチンは、クロック値がほぼタイム・アウト値であるかどうかを判定する。クロック値がタイム・アウト値を超える場合、接続が既に閉じ

50

られていることがある。結果として、タイム・アウト値の前のある時間期間が選択されるべきである。クロック値がほぼタイム・アウト値である場合、ルーチンは、ブロック506に続く。そうでない場合、ルーチンは、ブロック508に続く。ブロック506で、ルーチンは、`send_heartbeat`サブ・ルーチンを呼び出し、これは、接続を持続することができるメッセージを送る。`send_heartbeat`サブ・ルーチンについて、図6に関して以下でさらに述べる。ブロック508で、ルーチンは、その呼出し側に戻る。

【0030】

`sustain_connection`ルーチンは、メッセージ送信または受信するスレッドとは別個のスレッドのループ内で繰り返し呼び出されることになる。あるいは、`sustain_connection`ルーチンは、タイム・アウト前の指定された時間間隔でトリガされるクロック・イベントに応答して呼び出すこともできる。

10

【0031】

あるいは、`sustain_connection`ルーチンは単に、クロック値に関係なく、タイム・アウト値より小さいある間隔で、ハートビート・メッセージを送信することができる。一例として、1分間活動が無かった後にプロキシ・サーバがタイム・アウトになる場合、このルーチンは、メッセージがほんの数秒前に送信されまたは受信された場合にも、30秒ごとにハートビート・メッセージを送信することができる。

【0032】

図6は、`send_heartbeat`サブ・ルーチンの一実施形態を示す流れ図である。このサブ・ルーチンは、ハートビート・メッセージを送信する、SIPアプリケーションのコンポーネントなどの、アプリケーション層コンポーネントによって実施される。ハートビート・メッセージは、ある時間にわたって接続内でメッセージが送信されない場合に閉じる可能性のある接続を持続するために送信することができる。サブ・ルーチンは、ブロック602で開始し、パラメータとしてセッションの指標(indication)を受信する。ボックス604で、サブ・ルーチンは、ハートビート・メッセージを作成する。ハートビート・メッセージは、データ通信スタックのある層によっては無視されることができ、データ通信スタックの別の層によっては無視されることがないメッセージである。一例として、ハートビート・メッセージは、開いたSIPセッションを有する2つコンピューティング装置間でメッセージを処理する別のコンピューティング装置(プロキシ・サーバなど)によってTCP接続が閉じられる恐れがある場合に、そのTCP接続を持続するために、SIPアプリケーションによって送信することができる。ハートビート・メッセージは、たとえばCRLF文字だけを含むことができる。あるいは、ハートビート・メッセージは、CRLFで終了する1つまたは複数のスペースおよびタブを含むことができる。ハートビート・メッセージは、CRLFを使用するのではなく、キャリッジ・リターンまたは改行文字を使用することもできる。TCP層にとって、こうしたメッセージは、メッセージが送信されまたは受信される際に、クロック値を再設定する有効なメッセージであることになる。しかし、SIPアプリケーションは、こうしたメッセージを無視することができる。したがって、こうしたハートビート・メッセージを送信することによって、TCP接続は開かれた状態で保たれることができ、したがって、SIPプレゼンス情報は正確なままであり、有用でない情報が、SIPアプリケーションに提示されることはない。代替実施形態では、ハートビート・メッセージ内で、単にキャリッジ・リターンまたは改行文字が送信されることになる。

20

30

40

【0033】

ブロック606で、サブ・ルーチンは、`send_message`サブ・ルーチンを呼び出し、それに、作成されたハートビート・メッセージをパラメータとして渡す。`send_message`サブ・ルーチンについて、図4に関連して上記で述べている。ブロック608で、サブ・ルーチンは、その呼出し側に戻る。

【0034】

図7は、`receive_message`サブ・ルーチンの一実施形態を示す流れ図で

50

ある。このサブ・ルーチンは、SIPアプリケーションなど、受信側コンピューティング装置のアプリケーション層コンポーネントによって実施されることになる。このサブ・ルーチンは、ブロック702で開始し、パラメータとしてメッセージの指標(indication)を受信する。ブロック704で、サブ・ルーチンは、受信されたメッセージがハートビート・メッセージかどうか決定する。サブ・ルーチンは、メッセージの内容が単にキャリッジ・リターン、改行またはその両方かどうかをチェックすることによって、受信されたメッセージがハートビート・メッセージかどうかを決定しようと試みることになる。あるいは、receive_messageサブ・ルーチンは、ハートビート・メッセージを識別すると分かっている、メッセージ内の他の任意の、情報を探すようにカスタマイズすることができる。受信されたメッセージがハートビート・メッセージである場合、サブ・ルーチンは、ブロック706に続く。そうでない場合、サブ・ルーチンは、ブロック708に続く。ブロック706で、サブ・ルーチンは、メッセージに肯定応答することができる。一般に、トランスポート層が、メッセージに肯定応答する。しかし、場合によっては、アプリケーション層コンポーネント層が、メッセージに肯定応答する必要がある場合がある。このステップはサブ・ルーチンを実施するアプリケーション層コンポーネントに応じて任意選択であることができるので、ブロック706は、破線で示されている。ブロック708で、サブ・ルーチンは、SIPアプリケーションなどにメッセージを転送することになる。ブロック710で、サブ・ルーチンは、その呼出し側に戻る。

【0035】

メッセージを受信すると、サブ・ルーチンは、クロック値(図示せず)を再設定することになる。このクロック値は、接続を持続するためにハートビート・メッセージが送信される必要があるかどうか決定するために使用することができる。

【0036】

図8は、ある実施形態におけるメッセージの流れを示すメッセージ流れ図である。図示する実施形態では、コンピューティング装置1が、コンピューティング装置2とSIPセッションを確立している。セッション内で交換されるメッセージは、プロキシ・サーバを通過する。一例として、 $t=0$ でコンピューティング装置1がメッセージ1を送信するときに、メッセージはプロキシ・サーバを通過し、 $t=1$ で、コンピューティング装置2が、メッセージを受信する。同様に、 $t=2$ でコンピューティング装置2がメッセージ2を送信するときに、メッセージは、 $t=$ でコンピューティング装置1がそれを受信する前に、プロキシ・サーバを通過する。プロキシ・サーバは、メッセージを転送しまたは受信するときはいつでも、クロックを再設定することができる。一例として、プロキシ・サーバは、メッセージ1を転送するときに、クロック値を0に再設定する。同様に、プロキシ・サーバは、メッセージ2を受信するときに、クロック値を0に再設定する。クロック値は、たとえばシステム・クロックに応答して、自動的に増分されることになる。クロック値がタイム・アウト値を超える場合、プロキシ・サーバは、そのセッションと関連する接続を閉じることになる。図示する例では、コンピューティング装置1は、クロック値がタイム・アウト値を超えた後は、メッセージ3を送信することが不可能となることになる。

【0037】

図9は、ある実施形態におけるメッセージの流れを示すメッセージ流れ図である。図示する実施形態では、図8で示し、上記で説明したように、コンピューティング装置1および2は、時点0と3の間にメッセージ1および2を交換している。 $t=4$ で、コンピューティング装置1は、間もなくタイム・アウトになる(図中に破線で示す)ことを認識すると、 $t=5$ として示すタイム・アウトになる前に、コンピューティング装置2にハートビート・メッセージを送信することになる。ハートビート・メッセージを送信すると、プロキシ・サーバはクロック値を再設定することができ、したがって、プロキシ・サーバが接続を閉じてないので、コンピューティング装置1は、時点6で、メッセージ3を送信することが可能であることになる。プロキシ・サーバのタイム・アウトの前にハートビート・メッセージを送信することによって、システムは、コンピューティング装置1と2の間で接続を持続することができる。ハートビート・メッセージを送信するコンピューティング

装置は、ハートビート・メッセージの受信側のクロック値もまたタイム・アウト値を超過するまでまだ十分な時間があるうちに、ハートビート・メッセージの送信を試みる必要があることになる。

【0038】

図10は、ある実施形態におけるメッセージの流れを示すブロック図である。図示する実施形態では、送信側コンピューティング装置1000は、受信側コンピューティング装置1002とのSIPセッションを開いている。送信側コンピューティング装置がそのアプリケーション/SIP層1004を介してSIPメッセージ1を送信するときに、トランスポート層1006は、そのメッセージを2つのセグメントS1およびS2にセグメント化することができる。タイム・アウトを回避するために、送信側コンピューティング装置は、ハートビート・メッセージS3を送信することができる。ハートビート・メッセージは、キャリッジ・リターン、改行、その両方、またはアプリケーションによっては無視されることになえるハートビート・メッセージを示す他の情報を含むことができる。次いで、受信側コンピューティング装置は、トランスポート層1016によってトランスポート・メッセージS4、S5およびS6としてセグメント化されたSIPメッセージ2を送信することができる。セグメント化されたメッセージは、様々なサイズのものである可能性がある。

10

【0039】

ある実施形態では、コンピューティング装置は、一部の接続でハートビート・メッセージを送信することができ、その接続を持続するために、他の接続を含む別の機構を使用することができる。一例として、コンピューティング装置は、それ自体の「ホーム」サーバ、またはそれに接続された他のコンピューティング装置にメッセージを送信するときに、そのホーム・サーバを、そのコンピューティング装置の接続にハートビート・メッセージに関連付け、またその関連付けの指標を格納することが可能であるので、ハートビート・メッセージを使用することができる。そうすることによって、サーバは、後に接続が失われる場合に、そのコンピューティング装置のプレゼンス情報を効率的に更新することを可能とすることができる。コンピューティング装置は、そうすることが有利である可能性がある場合には、他の接続でREGISTERメッセージを送信することができる。一例として、コンピューティング装置は、中間装置がハートビート・メッセージを転送しない場合に、REGISTERメッセージを送信することができる。

20

30

【0040】

本明細書で論じた特定の実施例は、SIPおよびTCPを使用することに言及しているが、代替実施形態は、他の等価または類似のプロトコルを使用することができる。

【0041】

セッション接続を持続するための技術が実装されるコンピューティング装置は、中央処理装置、メモリ、入力装置（キーボードおよびポインティング装置など）、出力装置（表示装置など）、および記憶装置（ディスク・ドライブなど）を含むことができる。メモリおよび記憶装置は、セキュリティ・システムを実装する命令を収容することが可能なコンピュータ読取り可能媒体である。さらに、そのデータ構造体およびメッセージ構造体は、格納したり、あるいは通信リンク上の信号などのデータ伝送媒体を介して送信したりすることができる。インターネット、ローカル・エリア・ネットワーク、広域ネットワークまたはポイント・ツー・ポイント・ダイヤルアップ接続など、様々な通信リンクを使用することができる。

40

【0042】

図1に、セッション接続を持続するための技術を実装することが可能な適切なオペレーティング環境の一例を示す。このオペレーティング環境は、適切なオペレーティング環境の一例にすぎず、本システムの使用または機能の範囲についてのどんな制限をも示唆するものではない。使用に適す可能性のある他の周知のコンピューティング・システム、環境および構成には、パーソナル・コンピュータ、サーバ・コンピュータ、「スマート」携帯電話を含むハンドヘルドまたはラップトップ装置、マルチ・プロセッサ・システム、マイ

50

クロ・プロセッサ・ベースのシステム、プログラマブルな家電、ネットワーク P C、ミニ・コンピュータ、メインフレーム・コンピュータ、上記システムまたは装置のいずれかを
含む分散コンピューティング環境などが含まれる。

【0043】

セッション接続を持続するための技術について、1つまたは複数のコンピュータまたは
他の装置によって実行されるプログラム・モジュールなど、コンピュータ実行可能命令の
一般的な文脈で述べることができる。一般に、プログラム・モジュールは、特定のタスク
を実施し、または特定の抽象データ型を実装するルーチン、プログラム、オブジェクト、
コンポーネント、データ構造体などを含む。一般に、プログラム・モジュールの機能は、
様々な実施形態で望まれるように、組み合わせられることも、分散されることもある。

10

【0044】

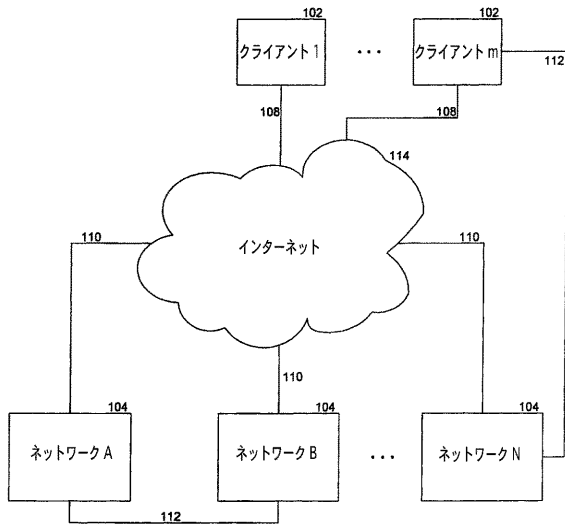
説明するため、本発明の具体的な実施形態について本明細書で述べたが、上記内容から
、本発明の精神および範囲から逸脱せずに様々な修正を行うことができることが理解され
よう。したがって、本発明は、添付の特許請求の範囲によって以外は限定されない。

【符号の説明】

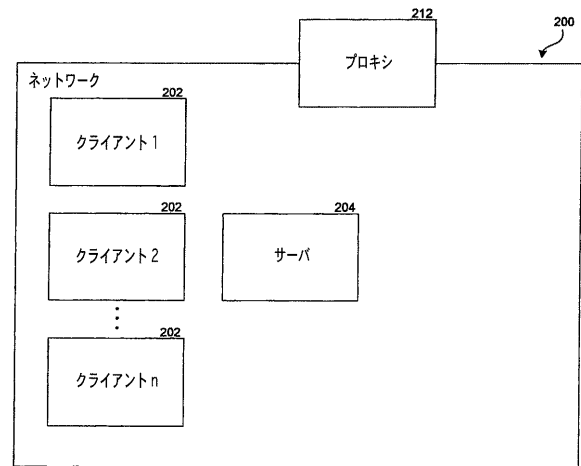
【0045】

102	クライアント	
104	ネットワーク	
108	ネットワーク接続	
110	データ通信リンク	20
112	プライベート・データ通信リンク	
114	インターネット	
200	ネットワーク	
202	クライアント	
204	サーバ	
212	プロキシ	
300～303	コンピューティング装置	
304	アプリケーション／S I P	
306	トランスポート	
308	ネットワーク	30
310	データリンク	
312	物理	
314	アプリケーション／S I P	
316	トランスポート	
318	ネットワーク	
320	データリンク	
322	物理	
1000	送信側コンピューティング装置	
1002	受信側コンピューティング装置	
1004、1014	アプリケーション／S I P（層）	40
1006、1016	トランスポート（層）	
1008、1018	ネットワーク（層）	
1010、1020	データリンク（層）	
1012、1022	物理（層）	

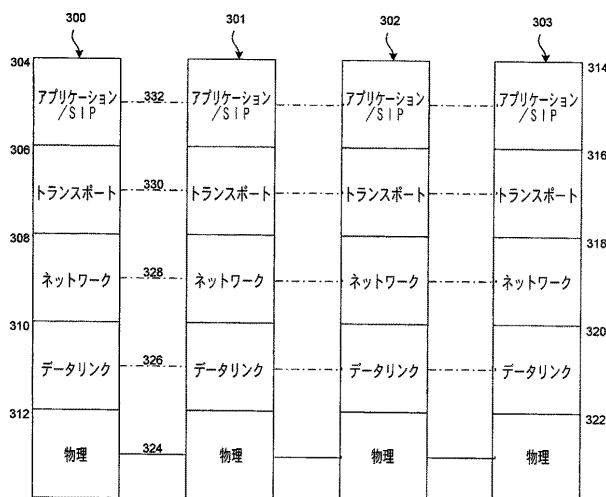
【図 1】



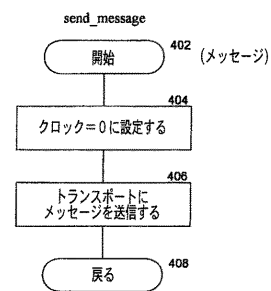
【図 2】



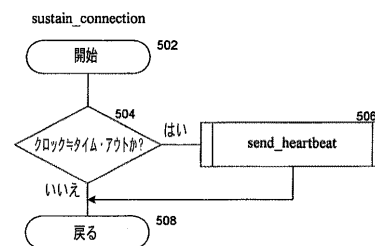
【図 3】



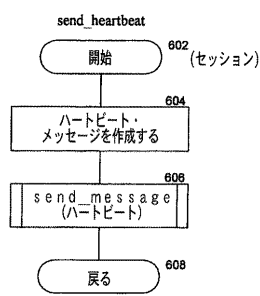
【図 4】



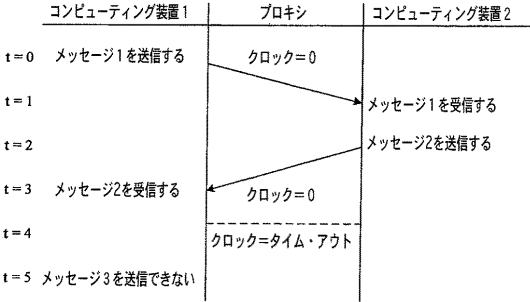
【図 5】



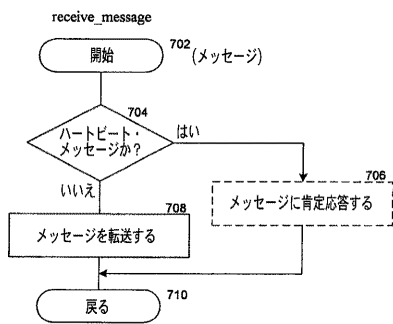
【図 6】



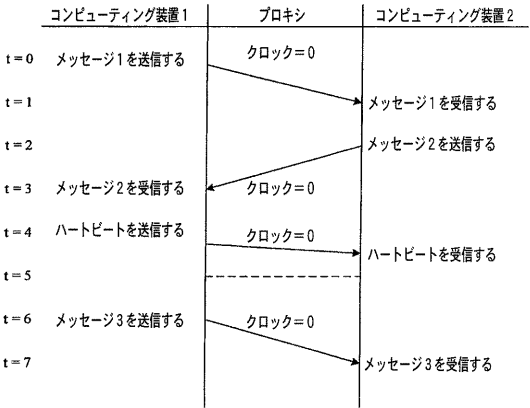
【図 8】



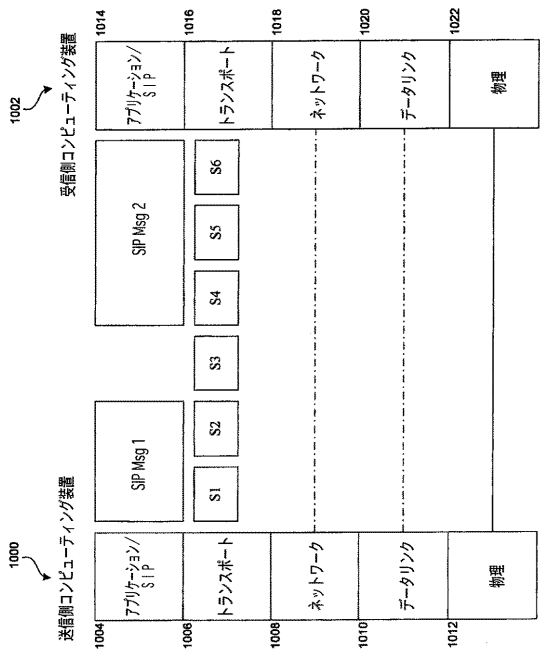
【図 7】



【図 9】



【図 10】



フロントページの続き

- (72)発明者 ジェームス エバン アンダーリー
アメリカ合衆国 98052 ワシントン州 レッドモンド ワン マイクロソフト ウェイ マ
イクロソフト コーポレーション内
- (72)発明者 マーク マーカーヤン
アメリカ合衆国 98052 ワシントン州 レッドモンド ワン マイクロソフト ウェイ マ
イクロソフト コーポレーション内

審査官 衣嶋 文彦

- (56)参考文献 特開2002-141938 (JP, A)
特表2004-537900 (JP, A)
国際公開第2004/012086 (WO, A1)
特表2003-502913 (JP, A)
特開2000-261548 (JP, A)
山居正幸, イーサネットとTCP/IP 第2章 TCP/IPコア・プロトコル, OPEN DESIGN, CQ出版株式
会社, 1997年 5月 1日, No.3, 第7版, p.18-54
- (58)調査した分野(Int.Cl., DB名)
H04L 12/00~12/955